**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**🙞🕮🙜**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

****

**BÁO CÁO HỌC PHẦN**

**Một số công nghệ phát triển phần mềm**

***ĐỀ TÀI***

**Phát triển ứng dụng với RESTful**

**Spring Boot**

|  |  |  |
| --- | --- | --- |
| GVHD | : | TS. Hà Mạnh Đào |
| Nhóm | : | 18 |
| Mã lớp | : | 20214IT6024001 |

*Hà Nội, 2022*

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**🙞🕮🙜**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

****

**BÁO CÁO HỌC PHẦN**

**Một số công nghệ phát triển phần mềm**

***ĐỀ TÀI***

**Phát triển ứng dụng với RESTful**

**Spring Boot**

|  |  |  |
| --- | --- | --- |
| GVHD | : | TS. Hà Mạnh Đào |
| Nhóm | : | 18 |
| Mã lớp | : | 20214IT6024001 |
| Sinh viên thực hiện | : | 1. Chu Văn Chiến - 2019602055 2. Đào Thị Thuỳ Dương - 2019604914 3. Nguyễn Thế Tiến Dũng -2109602418 |

*Hà Nội, 2022*

# *LỜI CẢM ƠN*

*Lời đầu tiên, nhóm 18 xin gửi lời cảm ơn chân thành tới thầy Hà Mạnh Đào. Trong quá trình học tập và thực hiện đề tài này, chúng em đã nhận được sự quan tâm giúp đỡ, hướng dẫn tận tình, tâm huyết của thầy. Những gì chúng em nhận được không chỉ dừng lại ở kiến thức môn học mà nhiều hơn thế đó là những lời khuyên, chia sẻ thực tế từ thầy. Chính nhờ phương pháp dạy học của thầy mà chúng em có cơ hội khám phá và phát huy khả năng của bản thân. Những buổi thuyết trình chính là cơ hội tuyệt vời giúp chúng em rèn luyện sự tự tin, kỹ năng giao tiếp, làm việc nhóm,... Đây cũng chính là hành trang quan trọng giúp chúng em tự tin bước chân vào môi trường làm việc thực tế.*

*Để hoàn thành được đề tài này, nhóm chúng em đã cùng nhau nghiên cứu, thảo luận, áp dụng những kiến thức được học trên lớp cùng với các nguồn tài liệu trên Internet và cả những trải nghiệm của bản thân. Chúng em rất mong sẽ nhận được những lời nhận xét, góp ý từ thầy cô và bạn đọc để đề tài này có thể hoàn thiện hơn nữa .*

*Một lần nữa, chúng em xin chân thành cảm ơn!*

***Nhóm sinh viên thực hiện.***

MỤC LỤC

[*LỜI CẢM ƠN* 1](#_Toc113521003)

[DANH MỤC HÌNH ẢNH 3](#_Toc113521004)

[DANH MỤC BẢNG BIỂU 4](#_Toc113521005)

[Chương 1. Tổng quan 6](#_Toc113521006)

[1.1. Tên đề tài 6](#_Toc113521007)

[1.2. Lý do chọn đề tài 6](#_Toc113521008)

[1.3. Giải pháp 6](#_Toc113521009)

[1.4. Công cụ, môi trường IDE 7](#_Toc113521010)

[Chương 2. Kiến thức cơ sở 11](#_Toc113521011)

[2.1. Web service 11](#_Toc113521012)

[2.2. Giao thức HTTP 14](#_Toc113521013)

[2.3. RESTful 16](#_Toc113521014)

[2.4. Spring Boot 17](#_Toc113521015)

[2.5. Kỹ thuật lập trình với Spring Boot 20](#_Toc113521016)

[Chương 3. Phân tích, thiết kế hệ thống 22](#_Toc113521017)

[3.1. Khảo sát yêu cầu người dùng 22](#_Toc113521018)

[3.2. Phân tích thiết kế hướng đối tượng 26](#_Toc113521019)

[3.3. Thiết kế UI/UX 61](#_Toc113521020)

[Chương 4. Cài đặt, thử nghiệm và kết luận 63](#_Toc113521021)

[4.1. Cài đặt 63](#_Toc113521022)

[4.2. Cấu trúc chương trình 64](#_Toc113521023)

[4.3. Kết quả nghiên cứu 65](#_Toc113521024)

[KẾT LUẬN 74](#_Toc113521025)

[TÀI LIỆU THAM KHẢO 75](#_Toc113521026)

DANH MỤC HÌNH ẢNH

[Hình 3. 1 Sơ đồ Use Case tổng quát 27](#_Toc113521033)

[Hình 3. 2 Biểu đồ phân rã chức năng đăng nhập 27](#_Toc113521034)

[Hình 3. 3 Biểu đồ phân rã chức năng quản lý danh mục 29](#_Toc113521035)

[Hình 3. 4 Phân rã chức năng quản lý nhân viên 33](#_Toc113521036)

[Hình 3. 5 Biểu đồ phần rã chức năng quản lý nhà cung cấp 36](#_Toc113521037)

[Hình 3. 6 Phân rã chức năng quản lý đơn hàng 39](#_Toc113521038)

[Hình 3. 7 Biểu đồ phân rã chức năng quản lý sản phẩm 41](#_Toc113521039)

[Hình 3. 8 Biểu đồ phân rã chức năng sửa thông tin cá nhân 44](#_Toc113521040)

[Hình 3. 9 Biều đồ hoạt động chức năng đăng nhập 46](#_Toc113521041)

[Hình 3. 10 Biểu đồ hoạt động chức năng quản lý nhà cung cấp 47](#_Toc113521042)

[Hình 3. 11 Biểu đồ hoạt động chức năng quản lý khách hàng 47](#_Toc113521043)

[Hình 3. 12 Biểu đồ hoạt động chức năng quản lý đơn hàng 48](#_Toc113521044)

[Hình 3. 13 Biều đồ hoạt động chức năng xem chi tiết đơn hàng 48](#_Toc113521045)

[Hình 3. 14 Biểu đồ hoạt động quản lý người dùng 49](#_Toc113521046)

[Hình 3. 15 Biều đồ hoạt động chức năng tìm kiếm 49](#_Toc113521047)

[Hình 3. 16 Biều đồ hoạt động chức năng sửa thông tin cá nhân 50](#_Toc113521048)

[Hình 3. 17 Biều đồ hoạt động chức năng quản lý danh mục 50](#_Toc113521049)

[Hình 3. 18 Biều đồ hoạt động chức năng quản lý sản phẩm 51](#_Toc113521050)

[Hình 3. 19 Biểu đồ tuần tự chức năng đăng nhập 51](#_Toc113521051)

[Hình 3. 20 Biều đồ tuần tự chức năng quản lý nhà cung cấp 52](#_Toc113521052)

[Hình 3. 21 Biều đồ tuần tự chức năng quản lý đơn hàng 53](#_Toc113521053)

[Hình 3. 22 Biểu đồ tuần tự chức năng xem chi tiết đơn hàng 54](#_Toc113521054)

[Hình 3. 23 Biểu đồ tuần tự chức năng quản lý nhân viên 55](#_Toc113521055)

[Hình 3. 24 Biểu đồ tuần tự chức năng tìm kiếm 56](#_Toc113521056)

[Hình 3. 25 Biểu đồ tuần tự chức năng sửa thông tin cá nhân 56](#_Toc113521057)

[Hình 3. 26 Biểu đồ tuần tự chức năng quản lý danh mục 57](#_Toc113521058)

[Hình 3. 27 Biểu đồ tuần tự chức năng quản lý sản phẩm 58](#_Toc113521059)

[Hình 3. 28 Biểu đồ ERD 61](#_Toc113521060)

[Hình 3. 29 Giao diện đăng nhập 61](#_Toc113521061)

[Hình 3. 30 Giao diện thống kê doanh thu 62](#_Toc113521062)

[Hình 3. 31 Giao diện cập nhật thông tin cá nhân 62](#_Toc113521063)

[Hình 3. 32 Giao diện thêm nhân viên 63](#_Toc113521064)

[Hình 4. 1 Trang web spring initializr 64](#_Toc112158250)

[Hình 4. 2 Giao diện thêm các dependencies 64](#_Toc112158251)

[Hình 4. 3 Cấu trúc chương trình 65](#_Toc112158252)

[Hình 4. 4 Giao diện đăng nhập 66](#_Toc112158253)

[Hình 4. 5 Giao diện thống kê doanh thu 66](#_Toc112158254)

[Hình 4. 6 Giao diện thống kê sản phẩm và nhân viên 67](#_Toc112158255)

[Hình 4. 7 Giao diện cập nhật thông tin cá nhân 67](#_Toc112158256)

[Hình 4. 8 Giao diện thêm nhà cung cấp 69](#_Toc112158257)

[Hình 4. 9 Giao diện hiển thị danh sách nhà cung cấp 70](#_Toc112158258)

[Hình 4. 10 Giao diện thêm danh mục 70](#_Toc112158259)

[Hình 4. 11 Giao diện hiển thị danh mục 71](#_Toc112158260)

[Hình 4. 12 Giao diện nhập kho 71](#_Toc112158261)

[Hình 4. 13 Giao diện danh sách các sản phẩm nhập kho 72](#_Toc112158262)

[Hình 4. 14 Giao diện xuất kho 72](#_Toc112158263)

[Hình 4. 15 Giao diện hiển thị danh sách phiếu xuất 73](#_Toc112158264)

[Hình 4. 16 Giao diện chi tiết phiếu xuất 73](#_Toc112158265)

DANH MỤC BẢNG BIỂU

[Bảng 3. 1 Actor bài toán 26](#_Toc113432164)

[Bảng 3. 2 Kịch bản chức năng đăng nhập 27](#_Toc113432165)

[Bảng 3. 3 Kịch bản thêm danh mục 29](#_Toc113432166)

[Bảng 3. 4 Kịch bản sửa danh mục 30](#_Toc113432167)

[Bảng 3. 5 Kịch bản xoá danh mục 31](#_Toc113432168)

[Bảng 3. 6 Kịch bản tìm kiếm 31](#_Toc113432169)

[Bảng 3. 7 Kịch bản thêm người dùng 33](#_Toc113432170)

[Bảng 3. 8 Kịch bản sửa người dùng 34](#_Toc113432171)

[Bảng 3. 9 Kịch bản xoá người dùng 35](#_Toc113432172)

[Bảng 3. 10 Kịch bản thêm nhà cung cấp 36](#_Toc113432173)

[Bảng 3. 11 Kịch bản sửa nhà cung cấp 37](#_Toc113432174)

[Bảng 3. 12 Kịch bản xoá nhà cung cấp 38](#_Toc113432175)

[Bảng 3. 13 Kịch bản thêm đơn hàng 39](#_Toc113432176)

[Bảng 3. 14 Kịch bản xem chi tiết đơn hàng 40](#_Toc113432177)

[Bảng 3. 15 Kịch bản thêm sản phẩm 41](#_Toc113432178)

[Bảng 3. 16 Kịch bản sửa sản phẩm 42](#_Toc113432179)

[Bảng 3. 17 Kịch bản xoá sản phẩm 43](#_Toc113432180)

[Bảng 3. 18 Thống kê doanh thu 44](#_Toc113432181)

[Bảng 3. 19 Thống kê sản phẩm 45](#_Toc113432182)

[Bảng 3. 20 Kịch bản thống kê doanh thu theo nhân viên 45](#_Toc113432183)

[Bảng 3. 21 Bảng hóa đơn 58](#_Toc113432184)

[Bảng 3. 22 Bảng chỉ tiết hóa đơn 59](#_Toc113432185)

[Bảng 3. 23 Bảng nhà cung cấp 59](#_Toc113432186)

[Bảng 3. 24 Bảng người dùng 59](#_Toc113432187)

[Bảng 3. 25 Bảng quyền 60](#_Toc113432188)

[Bảng 3. 26 Bảng danh mục 60](#_Toc113432189)

[Bảng 3. 27 Bảng sản phẩm 60](#_Toc113432190)

## Tổng quan

### Tên đề tài

Phát triển ứng dụng xây dựng website quản lý kho hàng với RESTful Spring Boot

### Lý do chọn đề tài

Do nhu cầu đòi hỏi ngày càng cao của xã hội cũng như của con người ngày càng cao, do đó để đáp ứng được yêu cầu của khách hàng thì phải luôn đổi mới và hoàn thiện hơn. Hiện nay do xã hội phát triển không ngừng và kéo theo là sự lớn mạnh của các doanh nghiệp, của các tổ chức, chính vì vậy mà việc quản lí sao cho chặt chẽ là nhu cầu cấp thiết các doanh nghiệp. Lĩnh vực quản lí kho hàng trong các doanh nghiệp hiện nay đang có nhu cầu rất lớn vì số lượng hàng hoá được nhập và xuất trong một thời điểm là rất lớn và đòi hỏi chính xác, đồng thời thời gian cũng là yếu tố quan trọng. Thực tế hiện nay có rất nhiều doanh nghiệp lớn quản lí hàng hoá bằng phương pháp thủ công và bán thủ công. Nếu có một công cụ quản lí kho hàng đáp ứng được các yêu cầu của nghiệp vụ cũng như đòi hỏi của của công việc thì đây sẽ là một thị trường lớn đầy tiềm năng phát triển.

Quản lý kho hàng là một công việc khá quan trọng đòi hỏi bộ phận quản lý phải thường xuyên cập nhật thông tin về hàng hoá nhập xuất. Quản lý kho một cách hiệu quả không đơn giản nếu chỉ sử dụng phương pháp quản lý kho truyền thống.

### Giải pháp

Việc ứng dụng CNTT đã mang lại bước đột phá mới cho công tác quản lý kho, giúp doanh nghiệp nắm bắt được thông tin về hàng hóa, vật tư, nguyên vật liệu và sản phẩm một cách chính xác kịp thời. Từ đó, người quản lý doanh nghiệp có thể đưa ra các kế hoạch và quyết định đúng đắn, giảm chi phí và tăng khả năng cạnh tranh, nâng cao kết quả hoạt động sản xuất kinh doanh. Để xử lý thông tin nhanh chóng, đầy đủ và chính xác và có hiệu quả thì ngành công nghệ thông tin đã phát triển tương đối phổ biến trên thế giới nói chung và Việt Nam nói riêng và đã cung cấp cho chúng ta những biện pháp cần thiết đó. Đã qua thời tính toán, sắp xếp và lưu trữ thông tin thủ công truyền thống mang nặng tính chất lạc hậu, lỗi thời. Công nghệ thông tin đã đi vào các ngành với một phương thức hoạt động hoàn toàn mới mẻ, sáng tạo và nhanh chóng mà không mất đi sự chính xác. Đặc biệt, nó đã đánh dấu một bước ngoặt trong việc áp dụng tin học vào trong hệ thống quản lý. Để đáp ứng nhu cầu ngày càng cao của khách hàng, doanh nghiệp phải đáp ứng đúng những mong muốn của khách hàng về mặt hàng và thời gian họ cần. Doanh nghiệp cần đạt được những yếu tố mới mang lại sự hoàn thiện trong khả năng bán hàng, đó là khả năng cung cấp dịch vụ nhanh. Với những lí do trên, nhóm em đã chọn đề tài: “***Phát triển ứng dụng xây dựng website quản lý kho hàng với RESTful Spring Boot***”

### Công cụ, môi trường IDE

#### Visual Studio Code

Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

* + - * Một số tính năng của Visual Studio Code
* **Hỗ trợ nhiều ngôn ngữ lập trình**

Visual Studio Code hỗ trợ nhiều ngôn ngữ lập trình như C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript, … Vì vậy, nó dễ dàng phát hiện và đưa ra thông báo nếu chương chương trình có lỗi.

* **Hỗ trợ đa nền tảng**

Các trình viết code thông thường chỉ được sử dụng hoặc cho Windows hoặc Linux hoặc Mac Systems. Nhưng Visual Studio Code có thể hoạt động tốt trên cả ba nền tảng trên.

* **Cung cấp kho tiện ích mở rộng**

Trong trường hợp lập trình viên muốn sử dụng một ngôn ngữ lập trình không nằm trong số các ngôn ngữ Visual Studio hỗ trợ, họ có thể tải xuống tiện ích mở rộng. Điều này vẫn sẽ không làm giảm hiệu năng của phần mềm, bởi vì phần mở rộng này hoạt động như một chương trình độc lập.

* **Kho lưu trữ an toàn**

Đi kèm với sự phát triển của lập trình là nhu cầu về lưu trữ an toàn. Với Visual Studio Code, người dùng có thể hoàn toàn yên tâm vì nó dễ dàng kết nối với Git hoặc bất kỳ kho lưu trữ hiện có nào.

* **Hỗ trợ web**

Visual Studio Code hỗ trợ nhiều ứng dụng web. Ngoài ra, nó cũng có một trình soạn thảo và thiết kế website.

* **Lưu trữ dữ liệu dạng phân cấp**

Phần lớn tệp lưu trữ đoạn mã đều được đặt trong các thư mục tương tự nhau. Ngoài ra, Visual Studio Code còn cung cấp các thư mục cho một số tệp đặc biệt quan trọng.

* **Hỗ trợ viết Code**

Một số đoạn code có thể thay đổi chút ít để thuận tiện cho người dùng. Visual Studio Code sẽ đề xuất cho lập trình viên các tùy chọn thay thế nếu có.

* **Hỗ trợ thiết bị đầu cuối**

Visual Studio Code có tích hợp thiết bị đầu cuối, giúp người dùng khỏi phải chuyển đổi giữa hai màn hình hoặc trở về thư mục gốc khi thực hiện các thao tác.

* **Màn hình đa nhiệm**

Người dùng Visual Studio Code có thể mở cùng lúc nhiều tệp tin và thư mục – mặc dù chúng không hề liên quan với nhau.

* **Intellisense**

Hầu hết các trình viết mã đều có tính năng nhắc mã Intellisense, nhưng ít chương trình nào chuyên nghiệp bằng Visual Studio Code. Nó có thể phát hiện nếu bất kỳ đoạn mã nào không đầy đủ. Thậm chí, khi lập trình viên quên không khai báo biến, Intellisense sẽ tự động giúp họ bổ sung các cú pháp còn thiếu.

* **Hỗ trợ Git**

Visual Studio Code hỗ trợ kéo hoặc sao chép mã trực tiếp từ GitHub. Mã này sau đó có thể được thay đổi và lưu lại trên phần mềm.

* **Bình luận**

Việc để lại nhận xét giúp người dùng dễ dàng nhớ công việc cần hoàn thành.

* Ưu nhược điểm của Visual Studio Code

**Ưu điểm:**

* Phát triển rất tích cực với sự hỗ trợ của Microsoft. Các tài liệu chính thức được duy trì tốt.
* Có sự hỗ trợ rất tích cực của cộng đồng với tất cả các plugin bạn cần. Nếu gửi một lỗi trên repo GitHub, bạn thường sẽ nhận được phản hồi trong vòng 4 ngày.
* Đối với một ứng dụng electron, tốc độ của VSCode rất cao, gần như tương đương với ST3 được xây dựng nguyên bản

**Nhược điểm:**

* Tương tự như tất cả các ứng dụng electron khác, bộ nhớ và mức sử dụng pin của VSCode khá tệ.
* Không có git merge, điều mà nhiều người mong đợi, vì ST3 và Atom có ​​khả năng làm điều đó.

Các phím tắt mặc định không có ý nghĩa và người dùng phải cấu hình lại gần như tất cả chúng.

#### Phần mềm Inteliji IDEA

IntelliJ IDEA cung cấp trình soạn thảo thông minh, trình phân tích mã và tập hợp mạnh mẽ của refactorings hỗ trợ một loạt các ngôn ngữ lập trình, các khuôn khổ và công nghệ, và đã sẵn sàng để sử dụng.

Phần mềm IntelliJ IDEA, sản phẩm nổi tiếng của JetBrains đã nhận được rất nhiều giải thưởng. Phần mềm được thiết kế để cải tiến năng suất cho các nhà phát triển. IntelliJ IDEA cung cấp trình soạn thảo thông minh, trình phân tích mã và tập hợp mạnh mẽ của refactorings hỗ trợ một loạt các ngôn ngữ lập trình, các khuôn khổ và công nghệ, và đã sẵn sàng để sử dụng.

* Ưu nhược điểm của IntelliJ IDEA
* Ưu điểm:
  + Hỗ trợ nhiều hệ thống, đồng thời hỗ trợ mở rộng database editor và UML designer
  + Cho các tính năng chạy thử UI, bảo vệ code, tích hợp Git
  + Hỗ trợ nhiều công cụ phát triển và gỡ lỗi cho hầu hết những servers
  + Trình soạn thảo văn bản thông minh cho ngôn ngữ laaji trình Java, CSS, HTML
  + Tích hợp kiểm soát các phiên bản
* Nhược điểm: Có quá nhiều phím tắt cần nhớ, UI giao tiếp cũng có những vấn đề chưa được cải thiện. IntelliJ có thể hoàn hảo với một lập trình viên chuyên nghiệp, nhưng chắc chắn sẽ không phù hợp với người mới bắt đầu.

#### Microsoft SQL Server

##### Khái niệm

SQL Server hay còn gọi là Microsoft SQL Server, viết tắt là MS SQL Server. Đây là một phần mềm được phát triển bởi Microsoft dùng để lưu trữ dữ liệu dựa trên chuẩn RDBMS, và cũng là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS)

##### Mục đích của việc sử dụng Microsoft SQL Server

Sử dụng bất kì một hệ quản trị CSDL nào cũng để lưu trữ dữ liệu và SQL cũng không ngoại lệ. Tuy các tính năng giúp việc sử dụng hiệu quả hơn như:

- Cho phép tạo nhiều cơ sở dữ liệu.

- Duy trì lưu trữ bền vững.

- Bảo mật cao.

- Phân tích dữ liệu bằng SSAS – SQL Server Analys Services.

- Tạo được báo cáo bằng SSRS – SQL Server Reporting Services.

- Thực hiện quá trình ETL (Extract – Transform – Load) bằng SISS – SQL Server Intergration Services.

##### Ưu, nhược điểm của Microsoft SQL Server

* Ưu điểm
  + Có thể cài nhiều phiên bản MS SQL khác nhau trên cùng một máy tính.
  + Duy trì riêng biệt các môi trường sản xuất, phát triển, thử nghiệm.
  + Giảm thiểu các vấn đề tạm thời trên cơ sở dữ liệu.
  + Tách biệt các đặc quyền bảo mật.
  + Duy trì máy chủ dự phòng.
* Nhược điểm:
  + SQL Server chỉ chạy trên hệ điều hành Windows.
  + Cần thanh toán phí license để chạy nhiều CSDL (database)

## Kiến thức cơ sở

### Web service

#### Khái niệm

Có thể là một ứng dụng, nguồn dữ liệu, hoặc nền tảng công nghệ có khả năng truy cập qua mạng Internet với giao thức HTTP(s)

WS giúp giao tiếp giữa các nèn tảng phần mềm với nhau không quan tâm đến ngôn ngữ và loại thiết bị của nhau:

* Truy cập qua mạng Internet, Giao thức HTTP(s)
* Định dạng dữ liệu phổ biến là JSON, XML, …
* Giúp phát triển các hệ thống độc lập
* 2 loại phổ biến SOAP và REST

Web service là một hệ thống được các chuyên gia thiết kế lên nhằm mục đích thông qua mạng Internet để hỗ trợ khả năng tương tác giữa nhiều ứng dụng trên máy tính khác nhau, và được XML mô tả giao diện chung và thực hiện chức năng gắn kết. Web service đáp ứng nhu cầu người dùng bằng cách đưa ra các thông tin mà họ yêu cầu. Địa chỉ URL dùng để xác định tài nguyên phần mềm.

Là một lớp giữa hai máy tính, giúp hai máy tính có thể tương tác với nhau qua mạng. Nói cách khác , Web service cho phép một chương trình máy tính có thể nói chuyện với một trang web thay vì người dùng tự dùng trình duyệt để truy cập trang web.

Mô hình: client gửi tin nhắn đến server và server hồi âm lại tin nhắn đó nhờ có web service. Web service hiện nay đa số giao tiếp qua cơ chế HTTP , nhưng format dữ liệu khi gửi và nhận thì hoàn toàn khác nhau

Là một giải pháp hữu ích trong việc tích hợp các cơ sở dữ liệu(database) của các đối tác kinh doanh trong ngành thương mại điện tử đặc biệt với doanh nghiệp B2B

#### Đặc điểm

* Có tính độc lập: Không phải cài đặt bất cứ một thành phần nào khẳng định tính độc lập của nó.
* Khả năng tự mô tả: Thông qua tài liệu WSDL giao diện của web service được xuất bản. Và sử dụng dịch vụ các client chỉ cần biết cấu trúc và nội dung của thông điệp yêu cầu đáp ứng của web service nhờ tài liệu WSDL
* Truy cập hệ thống: Sử dụng giao thức chuẩn của Web, Web service sử dụng WSDL mô tả dịch vụ và được xác định bằng UDDI registry dùng định djang của SOAP để thực hiện lệnh gọi.
* Độc lập về ngôn ngữ, nền tảng, giao thức. Có cơ sở là tiêu chuẩn mở XML. Vì thế ngay cả trong những môi trường khác nhau thì với web service,client và service cũng có thể liên kết được với nhau

**Một số đặc điểm cơ bản khác :**

* Web services dựa trên chuẩn mở thường là : XML, SOAP, WSDL và UDDI
* Cho phép client và server tương tác được với nhau mặc dù trong những môi trường khác nhau
* Dạng mở và dựa vào các tiêu chuẩn XML và HTTP là nền tảng kỹ thuật của web service
* Sử dụng UDDI và WSDL khiến web service rất linh động, có thể mô tả và phát triển tự động hóa.
* Dạng modular
* Được xây dựng trên nền tảng những công nghệ đã được chấp nhận
* Web service có thể công bố và gọi thực hiện qua mạng
* Được sử dụng nhiều trong những lĩnh vực khác nhau của cuộc sống như:

+ Tìm kiếm các thông tin về các khách sạn ở các thành phố hoặc trung tâm để liên hệ

+ Dịch vụ chọn lọc và phân loại itn tức

+ Ứng dụng đại lý du lịch, hiện thị danh sach đĩa nhạc

+ Thông tin thương mại gồm nhiều nội dung, nhiều mục tin như: Dự báo thời tiết, thông tin sức khỏe…..

#### Ưu điểm, nhược điểm

Ưu điểm

* Cho phép chương trình được viết bằng các ngôn ngữ khác nhau trên các nền tảng khác nhau giao tiếp được với nhau dựa trên một nền tảng tiêu chuẩn
* Sử dụng đơn giản chỉ có URL
* Làm việc với các giao thức chuẩn Web như XML. HTTP và TCP/IP
* Sự an toàn của máy chủ cơ sở dữ liệu luôn được bảo mật một cách chắc chắn
* Web service làm giảm giá thành cho việc tích hợp các hệ thống khác nhau

Nhược điểm

* Phụ thuộc vào tốc độ đường truyền internet
* Web service thiếu cơ chế khôi phục đủ tin cậy để đảm bảo giao dịch được khôi phục lại trạng thái ban dầu trong trường hợp xảy ra sự cố
* Số lượng các ứng dụng cộng tác cùng hoạt động sẽ ảnh hưởng tới hiệu suất tối ưu của Web service
* Tải trọng
* Đòi hỏi kết nối thông qua khá nhiều máy chủ trung gian cho nên tốc độ của hạ tầng mạng và các yếu tố liên quan tới hệ thống rõ ràng có vai trò quan trọng góp phần cải thiện hiệu năng của toàn bộ các ứng dụng WS

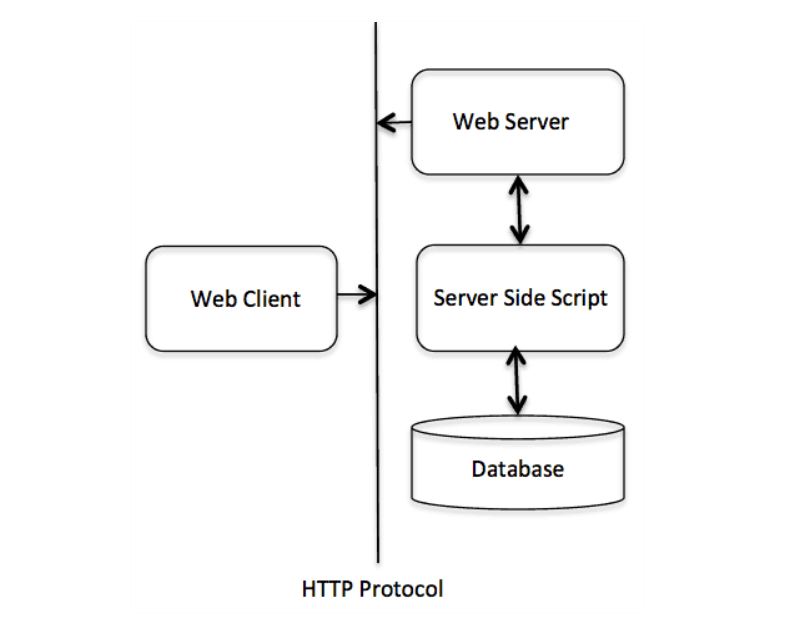
### Giao thức HTTP

[HTTP](https://quantrimang.com/hoc-http) là giao thức cấp độ ứng dụng cho các hệ thống thông tin phân phối, cộng tác, đa phương tiện, là nền tảng cho giao tiếp dữ liệu cho WWW (ví dụ: Internet) từ 1990. HTTP là một giao thức chung và không trạng thái (stateless) mà có thể được sử dụng cho các mục đích khác cũng như các sự co giãn của các phương thức yêu cầu, các code lỗi và Header của nó.

Về cơ bản, HTTP là một giao thức giao tiếp trên cơ sở [TCP/IP](https://quantrimang.com/kien-thuc-ve-giao-thuc-mang-tcp-ip-48), mà được sử dụng để phân phối dữ liệu (các tệp [HTML](https://quantrimang.com/html), các file ảnh, …) trên WWW. Cổng mặc định là TCP 80, những các cổng khác cũng có thể được sử dụng. Nó cung cấp một cách được tiêu chuẩn hóa cho các máy tính để giao tiếp với nhau. Chi tiết kỹ thuật HTTP xác định cách mà dữ liệu yêu cầu của Client sẽ được xây dựng và được gửi tới Server, và cách để Server phản hồi các yêu cầu này.

#### Các đặc trưng cơ bản của HTTP

* **HTTP là giao thức connectionless (kết nối không liên tục):** Client của HTTP, ví dụ: một trình duyệt khởi tạo một yêu cầu HTTP và sau đó một yêu cầu được tạo ra, Client ngắt kết nối từ Server và đợi cho một phản hồi. Server xử lý yêu cầu và thiết lập lại sự kết nối với Client để gửi phản hồi trở lại
* **HTTP là một phương tiện độc lập**: Nó nghĩa là, bất kỳ loại dữ liệu nào cũng có thể được gửi bởi HTTP miễn là Server và Client biết cách để kiểm soát nội dung dữ liệu. Nó được yêu cầu cho Client cũng như Server để xác định kiểu nội dung bởi sử dụng kiểu MIME thích hợp.
* **HTTP là stateless**: Như đã được đề cập ở trên, HTTP là connectionless và nó một kết quả trực tiếp là HTTP trở thành một giao thức Stateless. Server và Client biết về nhau chi trong một yêu cầu hiện tại. Sau đó, cả hai chúng nó quên tất cả về nhau. Do bản chất của giao thức, cả Client và các trình duyệt có thể giữ lại thông tin giữa các yêu cầu khác nhau giữa các trang web

Cấu trúc cơ bản: Cấu trúc rất đơn giản của một ứng dụng web

Giao thức HTTP là một giao thức Yêu cầu/Phản hồi dựa trên cấu trúc Client/Server, nơi mà các trình duyệt web, các thiết bị tìm kiếm,… hoạt động như các Client, và các Server web hoạt động như một Server.

**Client**

Client gửi một yêu cầu tới Server theo mẫu của một phương thức yêu cầu, URI, và phiên bản giao thức, được theo bởi một thông báo MIME chứa các bộ chỉnh sửa yêu cầu, thông tin Client, và nội dung đối tượng có thể qua một kết nối TCP/IP.

**Server**

Server phản hồi với một dòng trạng thái, bao gồm phiên bản giao thức của thông báo và một code thành công hoặc lỗi, theo sau bởi một thông báo MIME chứa thông tin Server, thông tin thực thể đa phương tiện và nội dung đối tượng có thể.

#### Phương thức HTTP

Bốn phương thức HTTP sau đây thường được sử dụng trong kiến ​​trúc dựa trên REST.

* **GET** - Cung cấp quyền truy cập chỉ đọc vào một tài nguyên.
* **POST** - Được sử dụng để tạo một tài nguyên mới.
* **DELETE** - Được sử dụng để xóa một tài nguyên.
* **PUT** - Được sử dụng để cập nhật tài nguyên hiện có hoặc tạo tài nguyên mới.

### RESTful

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

#### Diễn giải các thành phần

**API** (**A**pplication **P**rogramming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như [JSON](https://topdev.vn/blog/json-la-gi/) hay XML.

**REST** (**Re**presentational **S**tate **T**ransfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.

#### RESTful hoạt động như nào



REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là **CRUD** tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

### Spring Boot

1. Giới thiệu về Spring

- Spring là một Java framework cung cấp một cơ sở hạ tầng toàn diện hỗ trợ các lập trình viên phát triển ứng dụng Java.

- Nó được chia thành nhiều module, mỗi module làm một chức năng, ví dụ Spring Core, Web, Data access, AOP,... giúp các lập trình viên giảm thiểu đáng kể thời gian phát triển ứng dụng.

- Spring được xây dựng dựa trên 2 khái niệm nền tảng là Dependency injection và AOP (Aspect Oriented Programming).

Graphical user interface

Description automatically generated

- Nhược điểm của Spring là việc cấu hình (config) dự án quá phức tạp.

=> Spring Boot ra đời giúp chúng ta loại bỏ các bước cấu hình phức tạp mà Spring bắt buộc.

A picture containing graphical user interface

Description automatically generated

1. Giới thiệu về Spring boot

- Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh, giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển bussiness cho ứng dụng.

- Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring, nó không yêu cầu cấu hình XML

- Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho developer.

- Có các tính năng của Spring Framework.

- Version sử dụng: 2.6.6

1. Đặc điểm của Spring boot

- Spring Boot cho phép chúng ta cấu hình từ bên ngoài (externalize), do đó một ứng dụng có thể chạy trên nhiều môi trường khác nhau.

- Chúng ta có thể sử dụng file YAML, file properties, các biến môi trường và tham số command-line để thực hiện externalize configuration.

- Các dependency ‘starter’ giúp đơn giản hoá cấu hình và xây dựng ứng dụng.

- Dễ dàng deploy vì nó đã nhúng server container (Tomcat, Jetty hoặc Undertow) vào ứng dụng để tránh phức tạp khi triển khai lên môi trường production.

- Hỗ trợ cấu hình tự động cho chức năng Spring – bất cứ khi nào có thể

- Cung cấp các số liệu, kiểm tra và cho phép cấu hình ứng dụng từ bên ngoài.

1. Ưu, nhược điểm của Spring boot

* Ưu điểm:
  + Có các tính năng của Spring Framework.
  + Đơn giản hóa cấu hình và xây dựng được các ứng dụng độc lập có khả năng chạy bằng java-jar (cho cả java web) nhờ các dependency starter.
  + Dễ dàng triển khai vì nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) để tránh những khó khăn khi triển khai lên môi trường production do đó không cần phải tải file WAR.
  + Cấu hình ít, hỗ trợ tự động cấu hình bất cứ khi nào có thể (Giảm thời gian viết code, tăng năng suất, không cần cấu hình XML)
  + Cung cấp nhiều plugin, số liệu, cấu hình ứng dụng từ bên ngoài.

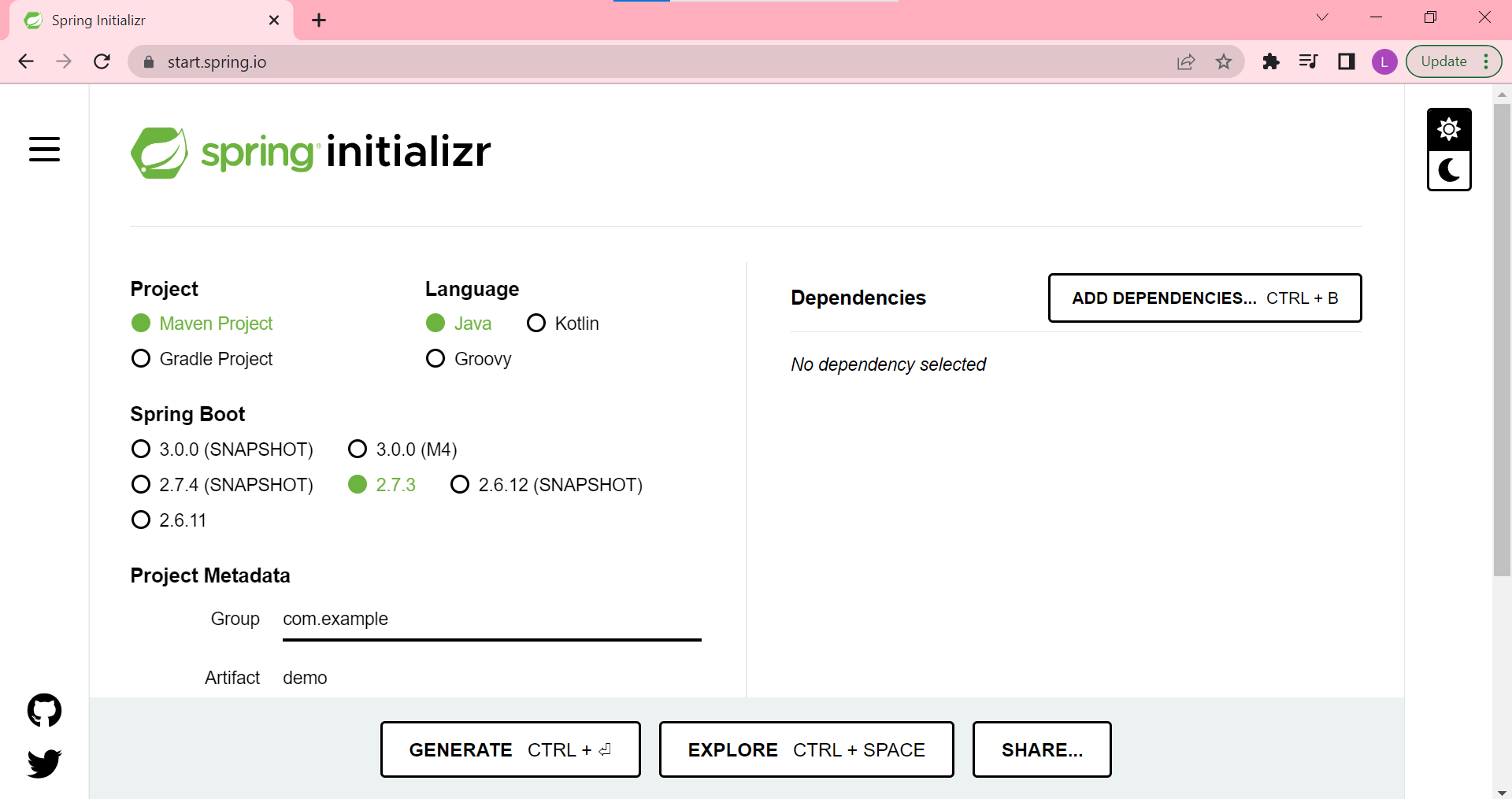
- Các lợi ích mà Spring Boot mang lại

* + Giúp tạo được ứng dụng độc lập dựa trên Spring, có thể tự chạy được java–jar
  + Xây dựng ứng dụng Spring độc lập (stand-alone) bao gồm gần như mọi thứ: REST API, WebSocket, Web, Streaming …
  + Tích hợp các loại mô-đun khác nhau trong các loại Spring như Spring Data, Spring MVC, Spring Security, Spring JDBC, Spring ORM…
  + Phát triển các ứng dụng Spring dựa trên các loại ngôn ngữ khác như Java hoặc Groovy.
  + Hỗ trợ nhiều IDE như Spring Tool Suite, IntelliJ IDEA, NetBeans hoặc cũng có thể cấu hình đơn giản và dùng Sublime Text để phát triển luôn.
* Nhược điểm:
  + Thiếu kiểm soát: Do style cố định, Spring Boot tạo ra nhiều phụ thuộc không được sử dụng dẫn đến kích thước tệp triển khai lớn.
  + Quá trình chuyển đổi dự án Spring cũ hoặc hiện có thành các ứng dụng Spring Boot nhiều khó khăn và tốn thời gian.
  + Không thích hợp cho các dự án quy mô lớn: Hoạt động liên tục với các microservices, theo nhiều nhà phát triển, Spring Boot không phù hợp để xây dựng các ứng dụng nguyên khối.
  + Bất kỳ sự cố nào với Máy ảo Java đều ảnh hưởng đến mã đã viết của phần mềm. Nói một cách đơn giản khi JVM không thành công, chương trình cũng bị lỗi.

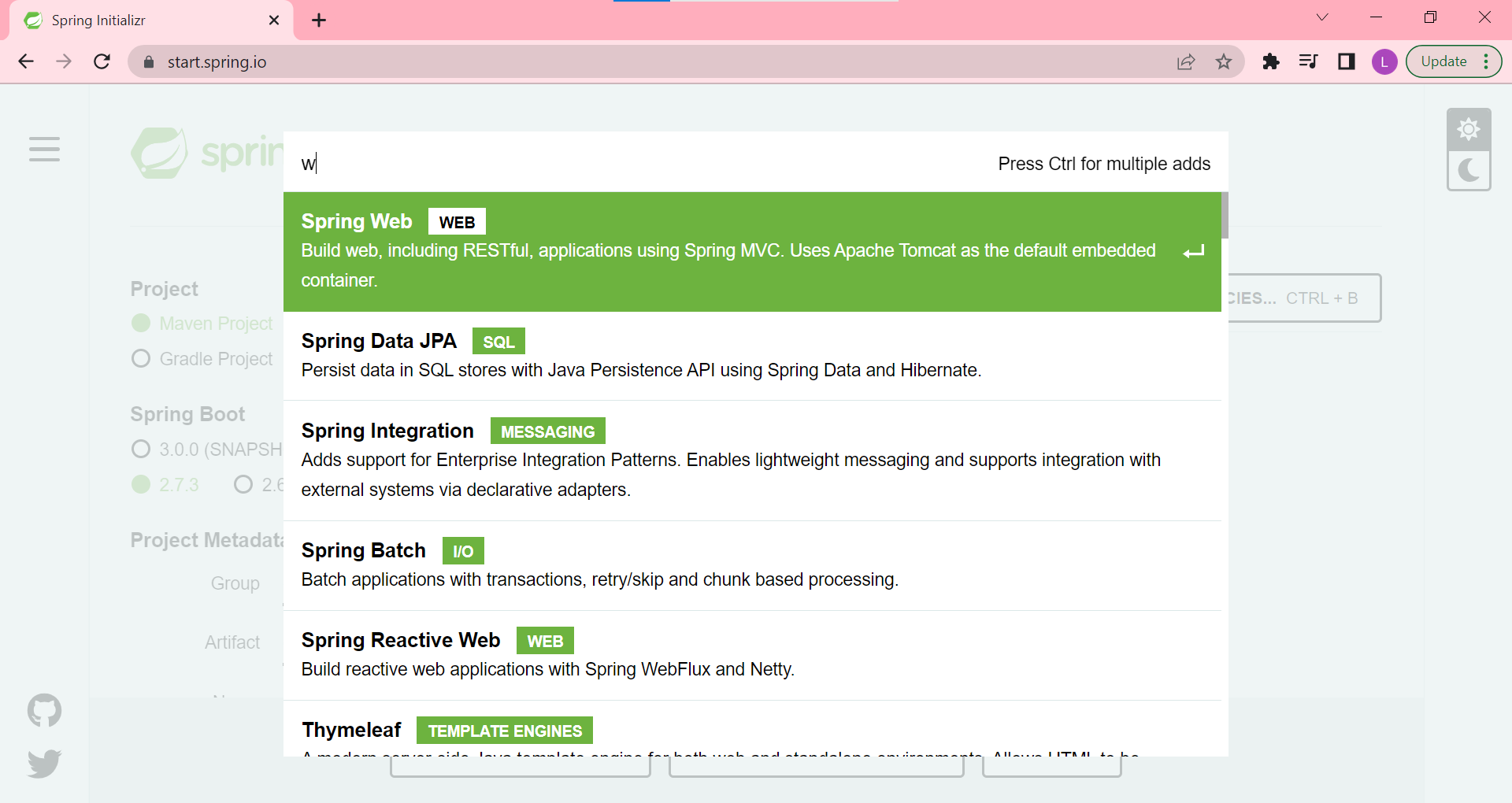
### Kỹ thuật lập trình với Spring Boot

Các bước cài đặt 1 project Spring Boot:

* Bước 1: Truy cập đường dẫn <https://start.spring.io/>



Hình 3. Trang web spring initializr

* Bước 2: Chọn loại project, ngôn ngữ, phiên bản Spring Boot, …
* Bước 3: Ấn “Add dependencies” để thêm các thư viện cần thiết.

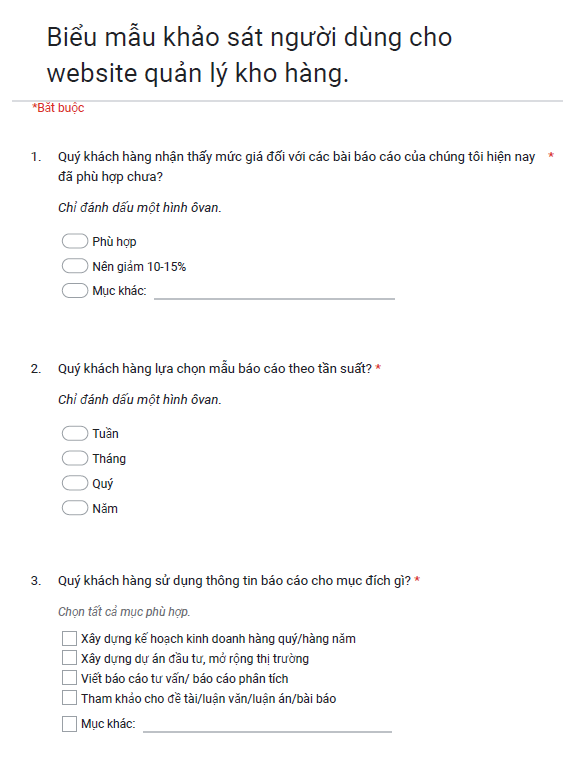
Hình 3. Giao diện thêm các dependencies

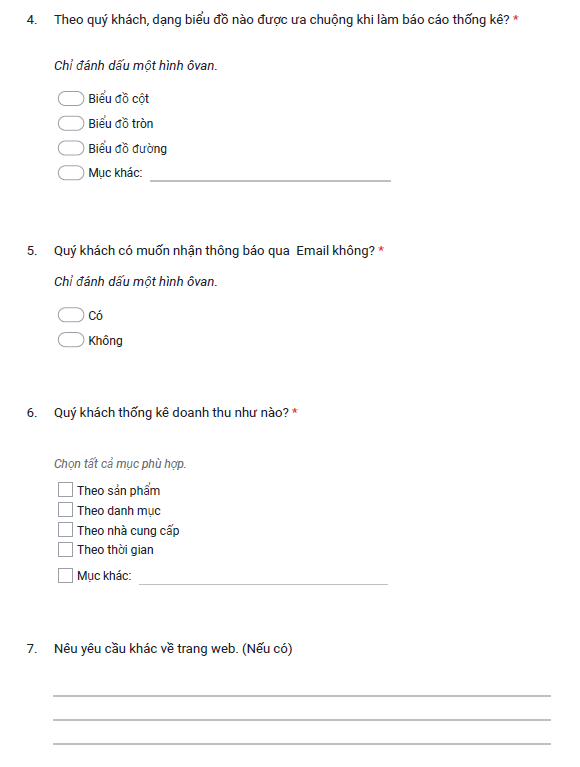
* Bước 4: Chọn “Generate” để lưu project về máy.

## Phân tích, thiết kế hệ thống

### Khảo sát yêu cầu người dùng

#### Khảo sát qua google form

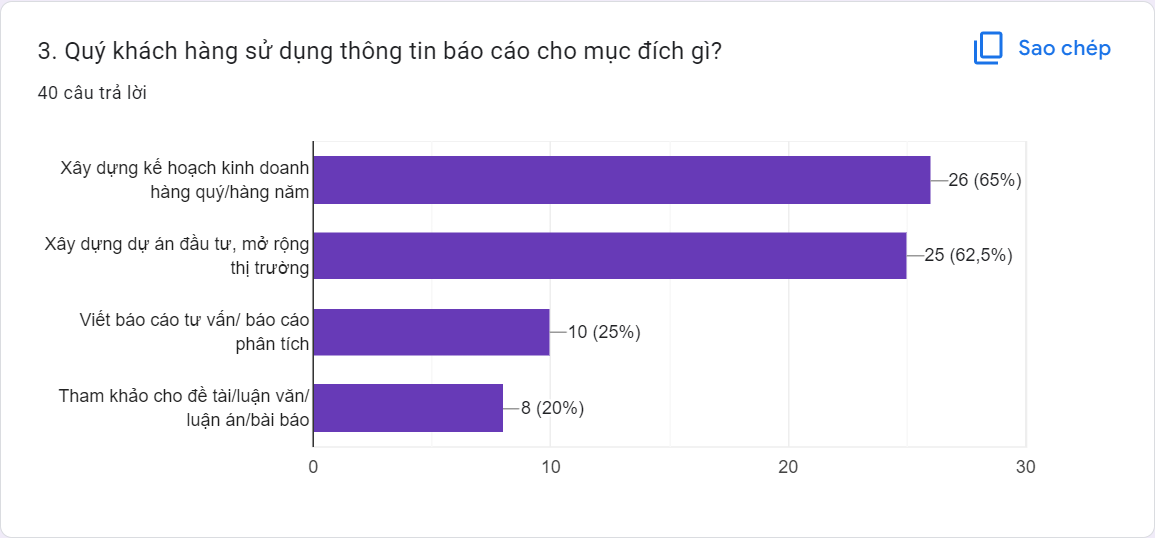


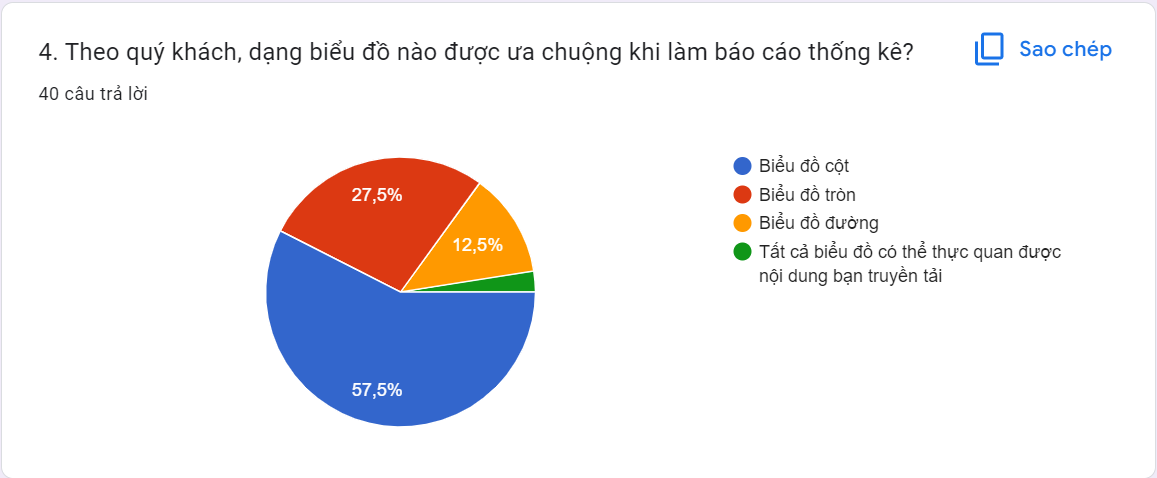


Thống kê kết quả khảo sát:

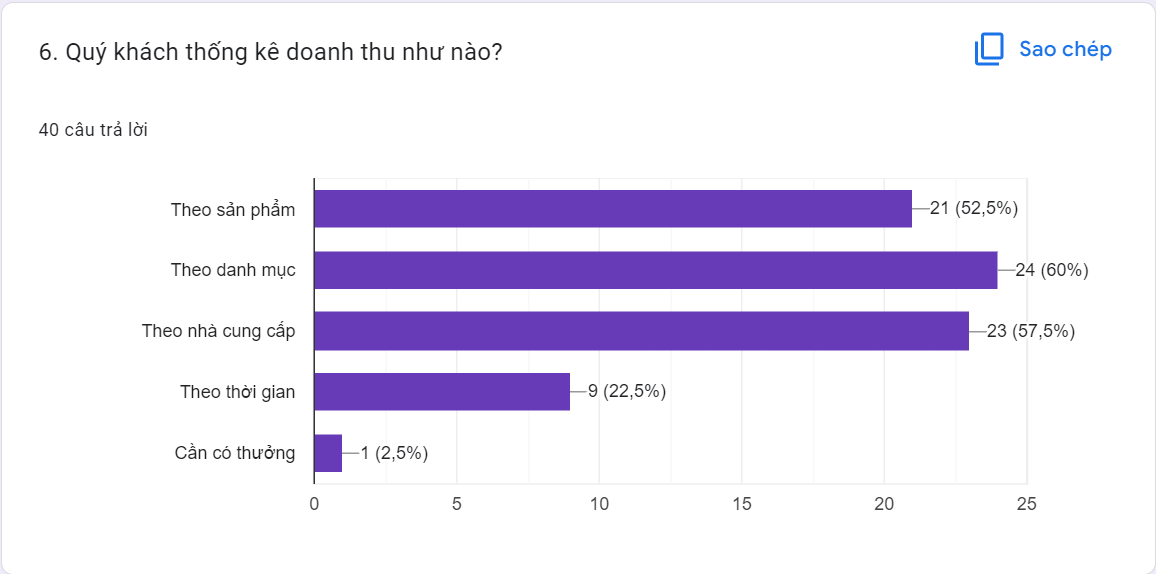














### Phân tích thiết kế hướng đối tượng

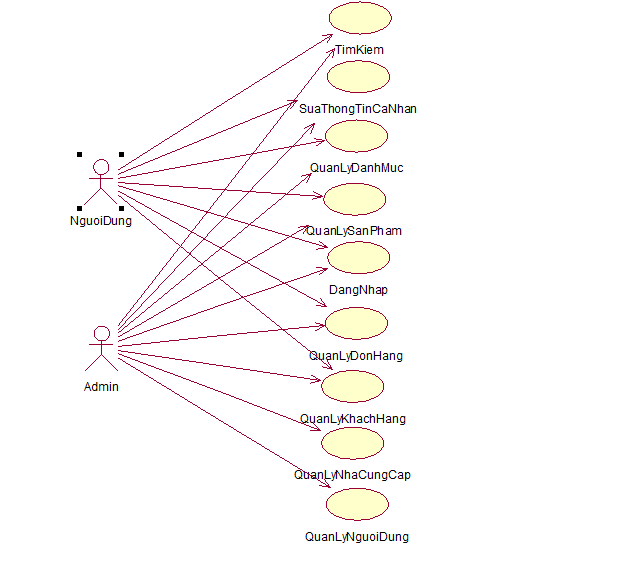
#### Usecase

Xác định actor bài toán

Bảng 3. Actor bài toán

|  |  |  |
| --- | --- | --- |
| STT | Actor | Diễn giải |
| 1 | Admin | Là người quản lý, điều hành toàn bộ công ty. Thực hiện phân quyền, quản lý cấp cao cho các danh mục khác. |
| 2 | User | Là nhân viên trong công ty, giao tiếp với phần mềm để bán hàng, tra cứu thông tin cá nhân. |

Sơ đồ Use Case Tổng quát:



Hình 3. Sơ đồ Use Case tổng quát

Biểu đồ phân rã chức năng của hệ thống:

* Biểu đồ Use case phân rã chức năng đăng nhập

Diagram

Description automatically generated

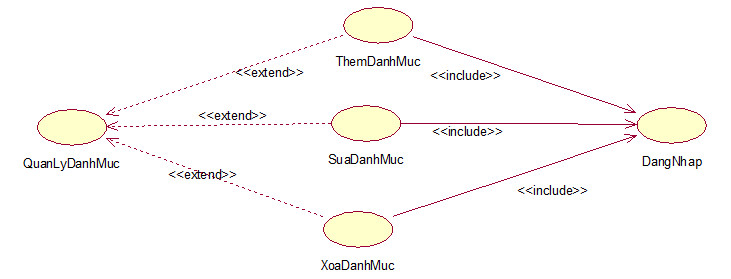
Hình 3. Biểu đồ phân rã chức năng đăng nhập

* + Kịch bản chức năng đăng nhập

Bảng 3. Kịch bản chức năng đăng nhập

|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm đăng nhập** |
| Tác nhân chính | Admin (Quản lý), nhân viên |
| Tiền điều kiện | Admin, nhân viên đã đăng kí tài khoản |
| Đảm bảo tối thiểu | Nhập đúng Username và password |
| Đảm bảo thành công | Dữ liệu đăng nhập tồn tại trong SQL |
| Kích hoạt | Admin chọn chức năng đăng nhập |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị màn hình đăng nhập 2. Admin, Nhân viên điền Username, password 3. Admin, Nhân viên ấn nút đăng nhập 4. Hệ thống kiểm tra ký tự nhập 5. Hệ thống truy vấn cơ sở dữ liệu 6. Hệ thống trả về kết quả đăng nhập cho phép người dùng đăng nhập thành công | |
| Ngoại lệ:   1. Hệ thống báo ký tự nhập không đủ 2. Hệ thống yêu cầu người dùng nhập đầy đủ thông tin 3. Hệ thống báo thông tin đăng nhập sai | |

* Biểu đồ use case phân rã chức năng quản lý danh mục



Hình 3. Biểu đồ phân rã chức năng quản lý danh mục

* Kịch bản Use case Thêm danh mục

Bảng 3. Kịch bản thêm danh mục

|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm danh mục** |
| Tác nhân chính | Admin (Quản lý) |
| Tiền điều kiện | Tác nhân đã đăng nhập tài khoản admin, đang ở form quản lý danh mục |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Thông tin danh mục được thêm thành công vào cơ sở dữ liệu |
| Kích hoạt | Admin chọn chức năng thêm danh mục |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form thêm mới danh mục và yêu cầu điền thông tin danh mục. 2. Tài khoản admin nhập thông tin danh mục và ấn “Thêm” 3. Hệ thống kiểm tra thông tin danh mục vừa thêm vào cơ sở dữ liệu 4. Hệ thống thông báo đã nhập thành công | |
| Ngoại lệ:   1. Hệ thống báo danh mục đã có trong cơ sở dữ liệu 2. Hệ thông yêu cầu nhập lại thông tin 3. Admin nhập lại thông tin danh mục 4. Hệ thống báo thông tin danh mục không hợp lệ 5. Hệ thống yêu cầu nhập lại thông tin 6. Admin nhập lại thông tin danh mục | |

* Kịch bản Use case Sửa danh mục

Bảng 3. Kịch bản sửa danh mục

|  |  |
| --- | --- |
| **Tên Use Case** | **Sửa thông tin danh mục** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Admin đã đăng nhập tài khoản và đang ở form quản lý danh mục |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Dữ liệu danh mục được cập nhật tồn tại trong SQL và không bị trùng lặp ID |
| Kích hoạt | Admin chọn “Lưu thông tin” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form danh sách danh mục 2. Admin chọn danh mục cần sửa 3. Admin thay đổi các thông tin danh mục sản phẩm và chọn nút cập nhật 4. Hệ thống truy vấn cơ sở dữ liệu và thông báo kết quả cập nhật thành công | |
| Ngoại lệ   1. Hệ thống thông báo nhân viên không tồn tại | |

* Kịch bản Use case Xóa danh mục

Bảng 3. Kịch bản xoá danh mục

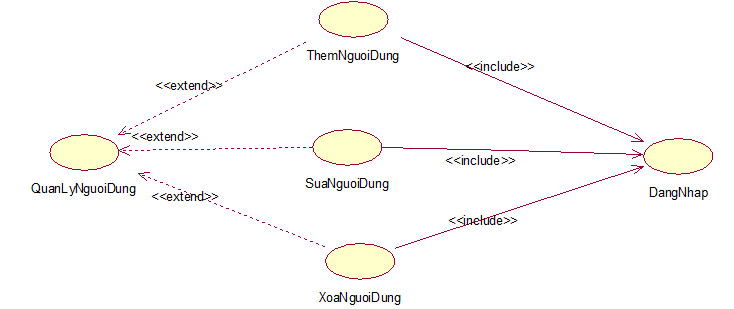
|  |  |
| --- | --- |
| **Tên Use Case** | **Xoá danh mục** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Tác nhân đã đăng nhập ở tài khoản admin, đang ở trong form quản lý |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại ở bước trước |
| Đảm bảo thành công | Dữ liệu danh mục được xoá tồn tại trong SQL |
| Kích hoạt | Admin ấn vào nút “Xoá danh mục” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách danh mục 2. Admin chọn bất kỳ nhân viên nào cần xoá và chọn chút “Xoá” 3. Hệ thống hiển thị pop up xác nhận việc xoá danh mục 4. Admin chọn “Đồng ý” 5. Hệ thống truy vấn cơ sở dữ liệu và thông báo kết quả xoá thành công | |
| Ngoại lệ: Không có | |

* Kịch bản Use Case tìm kiếm

Bảng 3. Kịch bản tìm kiếm

|  |  |
| --- | --- |
| **Tên Use Case** | **Tìm kiếm** |
| Tác nhân chính | Admin (Quản lý), Nhân viên () |
| Tiền điều kiện | Tác nhân đã đăng nhập tài khoản admin, user, đang ở form quản lý danh mục |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Thông tin danh mục được tìm kiếm thành công vào cơ sở dữ liệu |
| Kích hoạt | Admin chọn chức năng thêm danh mục |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form tìm kiếm danh mục và yêu cầu điền thông tin danh mục. 2. Tài khoản nhập tên danh mục và ấn “tìm kiếm” 3. Hệ thống tìm kiếm thông tin thành công và hiển thị ra ngoài màn hình | |
| Ngoại lệ:   1. Hệ thống không tìm thấy thông tin của danh mục, hiển thị ra thông báo | |

* Biểu đồ use case phân rã chức năng quản lý nhân viên



Hình 3. Phân rã chức năng quản lý nhân viên

* Kịch bản Use case Thêm người dùng

Bảng 3. Kịch bản thêm người dùng

|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm nhân viên** |
| Tác nhân chính | Admin (Quản lý nhân sự) |
| Tiền điều kiện | Tác nhân đã đăng nhập tài khoản admin, đang ở form quản lý nhân viên |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Thông tin nhân viên được thêm thành công vào cơ sở dữ liệu |
| Kích hoạt | Admin chọn chức năng thêm nhân viên |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form thêm mới nhân viên và yêu cầu điền thông tin nhân viên. 2. Tài khoản admin nhập thông tin nhân viên và ấn “Thêm” 3. Hệ thống kiểm tra thông tin nhân viên vừ thêm vào cơ sở dữ liệu 4. Hệ thống thông báo đã nhập thành công | |
| Ngoại lệ:   1. Hệ thống báo nhân viên đã có trong cơ sở dữ liệu 2. Hệ thông yêu cầu nhập lại thông tin 3. Admin nhập lại thông tin nhân viên 4. Hệ thống báo thông tin nhân viên không hợp lệ 5. Hệ thống yêu cầu nhập lại thông tin 6. Admin nhập lại thông tin nhân viên | |

* Kịch bản Use case Sửa người dùng

Bảng 3. Kịch bản sửa người dùng

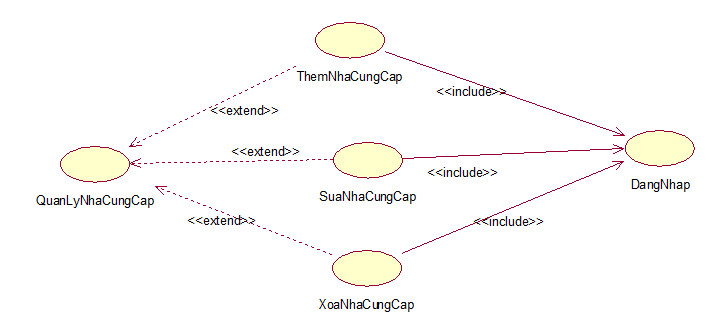
|  |  |
| --- | --- |
| **Tên Use Case** | **Sửa thông tin nhân viên** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Admin đã đăng nhập tài khoản và đang ở form quản lý nhân viên |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Dữ liệu nhân viên được cập nhật tồn tại trong SQL và không bị trùng lặp ID |
| Kích hoạt | Admin chọn “Lưu thông tin” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form danh sách nhân viên 2. Admin chọn nhân viên cần sửa 3. Admin thay đổi các thông tin danh mục sản phẩm và chọn nút cập nhật 4. Hệ thống truy vấn cơ sở dữ liệu và thông báo kết quả cập nhật thành công | |
| Ngoại lệ   1. Hệ thống thông báo nhân viên không tồn tại | |

* Kịch bản Use case Xóa người dùng

Bảng 3. Kịch bản xoá người dùng

|  |  |
| --- | --- |
| **Tên Use Case** | **Xoá nhân viên** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Tác nhân đã đăng nhập ở tài khoản admin, đang ở trong form quản lý |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại ở bước trước |
| Đảm bảo thành công | Dữ liệu nhân viên được xoá tồn tại trong SQL |
| Kích hoạt | Admin ấn vào nút “Xoá nhân viên” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách nhân viên 2. Admin chọn bất kỳ nhân viên nào cần xoá và chọn chút “Xoá” 3. Hệ thống hiển thị pop up xác nhận việc xoá nhân viên 4. Admin chọn “Đồng ý” 5. Hệ thống truy vấn cơ sở dữ liệu và thông báo kết quả xoá thành công | |
| Ngoại lệ: Không có | |

* Biểu đồ use case phân rã chức năng quản lý nhà cung cấp



Hình 3. Biểu đồ phần rã chức năng quản lý nhà cung cấp

* Kịch bản Use case Thêm nhà cung cấp

Bảng 3. Kịch bản thêm nhà cung cấp

|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm nhà cung cấp** |
| Tác nhân chính | Admin (Quản lý) |
| Tiền điều kiện | Tác nhân đã đăng nhập tài khoản admin, đang ở form quản lý nhà cung cấp |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Thông tin danh mục được thêm thành công vào cơ sở dữ liệu |
| Kích hoạt | Admin chọn chức năng thêm nhà cung cấp |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form thêm mới sản phẩm và yêu cầu điền thông tin nhà cung cấp. 2. Tài khoản admin nhập thông tin nhà cung cấp và ấn “Thêm” 3. Hệ thống kiểm tra thông tin nhà cung cấp vừa thêm vào cơ sở dữ liệu 4. Hệ thống thông báo đã nhập thành công | |
| Ngoại lệ:   1. Hệ thống báo nhà cung cấp đã có trong cơ sở dữ liệu 2. Hệ thông yêu cầu nhập lại thông tin 3. Admin nhập lại thông tin nhà cung cấp 4. Hệ thống báo thông tin nhà cung cấp không hợp lệ 5. Hệ thống yêu cầu nhập lại thông tin 6. Admin nhập lại thông tin nhà cung cấp | |

* Kịch bản Use case Sửa nhà cung cấp

Bảng 3. Kịch bản sửa nhà cung cấp

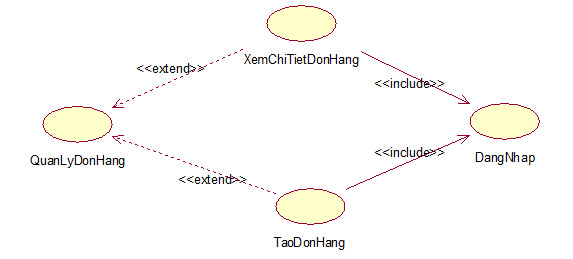
|  |  |
| --- | --- |
| **Tên Use Case** | **Sửa thông tin nhà cung cấp** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Admin đã đăng nhập tài khoản và đang ở form quản lý nhà cung cấp |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Dữ liệu nhà cung cấp được cập nhật tồn tại trong SQL và không bị trùng lặp ID |
| Kích hoạt | Admin chọn “Lưu thông tin” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form danh sách nhà cung cấp 2. Admin chọn nhà cung cấp cần sửa 3. Admin thay đổi các thông tin nhà cung cấp và chọn nút cập nhật 4. Hệ thống truy vấn cơ sở dữ liệu và thông báo kết quả cập nhật thành công | |
| Ngoại lệ   1. Hệ thống thông báo nhà cung cấp không tồn tại | |

* Kịch bản Use case Xóa nhà cung cấp

Bảng 3. Kịch bản xoá nhà cung cấp

|  |  |
| --- | --- |
| **Tên Use Case** | **Xoá nhà cung cấp** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Tác nhân đã đăng nhập ở tài khoản admin, đang ở trong form quản lý |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại ở bước trước |
| Đảm bảo thành công | Dữ liệu nhà cung cấp được xoá tồn tại trong SQL |
| Kích hoạt | Admin ấn vào nút “Xoá nhà cung cấp” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách nhà cung cấp 2. Admin chọn bất kỳ nhà cung cấp nào cần xoá và chọn chút “Xoá” 3. Hệ thống hiển thị pop up xác nhận việc xoá nhà cung cấp 4. Admin chọn “Đồng ý” 5. Hệ thống truy vấn cơ sở dữ liệu và thông báo kết quả xoá thành công | |
| Ngoại lệ: Không có | |

* Biểu đồ use case phân rã chức năng quản lý đơn hàng



Hình 3. 6 Phân rã chức năng quản lý đơn hàng

* Kịch bản Use case Thêm đơn hàng

Bảng 3. Kịch bản thêm đơn hàng

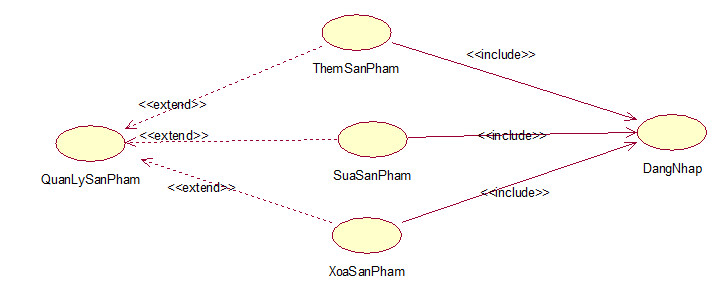
|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm hoá đơn** |
| Tác nhân chính | Admin (Quản lý) |
| Tiền điều kiện | Tác nhân đã đăng nhập tài khoản admin, hoặc tài khoản user, đang ở form tạo hoá đơn, số lượng sản phẩm ở trong kho phải lớn hơn số lượng hoa quả ở hoá đơn |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Thông tin khách hàng được thêm thành công vào cơ sở dữ liệu |
| Kích hoạt | Admin hoặc nhân viên chọn chức năng tạo hoá đơn |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form tạo hoá đơn và yêu cầu điền thông tin hoá đơn 2. Tài khoản admin nhập thông tin hoá đơn và ấn “Thêm” 3. Hệ thống kiểm tra thông tin hoá đơn vừa thêm vào cơ sở dữ liệu 4. Hệ thống thông báo đã nhập thành công | |
| Ngoại lệ:   1. Hệ thống báo sản phẩm không tồn tại trong kho 2. Hệ thông yêu cầu nhập lại thông tin 3. Admin nhập lại thông tin sản phẩm 4. Hệ thống báo số lượng sản phẩm lớn hơn số lượng ở trong kho 5. Hệ thống yêu cầu nhập lại thông tin 6. Admin nhập lại thông tin hoá đơn | |

* Kịch bản Use case Xem chi tiết đơn hàng

Bảng 3. Kịch bản xem chi tiết đơn hàng

|  |  |
| --- | --- |
| **Tên Use Case** | **Xem chi tiết hoá đơn** |
| Tác nhân chính | Admin, Nhân viên |
| Tiền điều kiện | Tác nhân đã đăng nhập ở tài khoản admin hoặc nhân viên, đang ở trong form quản lý |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại ở bước trước |
| Đảm bảo thành công | Dữ liệu hoá đơn hiện ra ở màn hình |
| Kích hoạt | Admin ấn vào nút “Xem chi tiết hoá đơn” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách hoá đơn 2. Admin chọn bất kỳ hoá đơn nào và ấn vào nút “Xem chi tiết” | |
| Ngoại lệ: Không có | |

* Biểu đồ use case phân rã chức năng quản lý sản phẩm



Hình 3. Biểu đồ phân rã chức năng quản lý sản phẩm

* Kịch bản Use case Thêm sản phẩm

Bảng 3. Kịch bản thêm sản phẩm

|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm sản phẩm** |
| Tác nhân chính | Admin (Quản lý) |
| Tiền điều kiện | Tác nhân đã đăng nhập tài khoản admin, đang ở form quản lý sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Thông tin danh mục được thêm thành công vào cơ sở dữ liệu |
| Kích hoạt | Admin chọn chức năng thêm sản phẩm |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form thêm mới sản phẩm và yêu cầu điền thông tin sản phẩm. 2. Tài khoản admin nhập thông tin sản phẩm và ấn “Thêm” 3. Hệ thống kiểm tra thông tin sản phẩm vừa thêm vào cơ sở dữ liệu 4. Hệ thống thông báo đã nhập thành công | |
| Ngoại lệ:   1. Hệ thống báo sản phẩm đã có trong cơ sở dữ liệu 2. Hệ thông yêu cầu nhập lại thông tin 3. Admin nhập lại thông tin sản phẩm 4. Hệ thống báo thông tin sản phẩm không hợp lệ 5. Hệ thống yêu cầu nhập lại thông tin 6. Admin nhập lại thông tin sản phẩm | |

* Kịch bản Use case Sửa sản phẩm

Bảng 3. Kịch bản sửa sản phẩm

|  |  |
| --- | --- |
| **Tên Use Case** | **Sửa thông tin sản phẩm** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Admin đã đăng nhập tài khoản và đang ở form quản lý sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại bước trước |
| Đảm bảo thành công | Dữ liệu danh mục được cập nhật tồn tại trong SQL và không bị trùng lặp ID |
| Kích hoạt | Admin chọn “Lưu thông tin” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị form danh sách sản phẩm 2. Admin chọn nhà cung cấp cần sửa 3. Admin thay đổi các thông tin danh mục sản phẩm và chọn nút cập nhật 4. Hệ thống truy vấn cơ sở dữ liệu và thông báo kết quả cập nhật thành công | |
| Ngoại lệ   1. Hệ thống thông báo sản phẩm không tồn tại | |

* Kịch bản Use case Xóa sản phẩm

Bảng 3. Kịch bản xoá sản phẩm

|  |  |
| --- | --- |
| **Tên Use Case** | **Xoá sản phẩm** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Tác nhân đã đăng nhập ở tài khoản admin, đang ở trong form quản lý |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại ở bước trước |
| Đảm bảo thành công | Dữ liệu danh mục được xoá tồn tại trong SQL |
| Kích hoạt | Admin ấn vào nút “Xoá sản phẩm” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách sản phẩm 2. Admin chọn bất kỳ nhân viên nào cần xoá và chọn chút “Xoá” 3. Hệ thống hiển thị pop up xác nhận việc xoá sản phẩm 4. Admin chọn “Đồng ý” 5. Hệ thống truy vấn cơ sở dữ liệu và thông báo kết quả xoá thành công | |
| Ngoại lệ: Không có | |

* Biểu đồ use case phân rã chức năng sửa thông tin cá nhân



Hình 3. Biểu đồ phân rã chức năng sửa thông tin cá nhân

* Biểu đồ Use case phân rã chức năng thống kê
* Kịch bản Use case thống kê doanh thu

Bảng 2. Thống kê doanh thu

|  |  |
| --- | --- |
| **Tên Use Case** | **Thống kê doanh thu** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Tác nhân đã đăng nhập ở tài khoản admin, đang ở trong form quản lý |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại ở bước trước |
| Đảm bảo thành công | Hiện ra biểu đồ thống kê doanh thu |
| Kích hoạt | Admin ấn vào nút “Xác nhận” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị biểu đồ doanh thu | |
| Ngoại lệ: Không có | |

* Kịch bản Use case thống kê sản phẩm

Bảng 3. Thống kê sản phẩm

|  |  |
| --- | --- |
| **Tên Use Case** | **Thống kê sản phẩm** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Tác nhân đã đăng nhập ở tài khoản admin, đang ở trong form quản lý |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại ở bước trước |
| Đảm bảo thành công | Hiện ra biểu đồ thống kê sản phẩm |
| Kích hoạt | Admin ấn vào nút “Xác nhận” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị biểu đồ sản phẩm | |
| Ngoại lệ: Không có | |

* Kịch bản Use case thống kê doanh thu theo nhân viên

Bảng 3. Kịch bản thống kê doanh thu theo nhân viên

|  |  |
| --- | --- |
| **Tên Use Case** | **Thống kê nhân viên** |
| Tác nhân chính | Admin |
| Tiền điều kiện | Tác nhân đã đăng nhập ở tài khoản admin, đang ở trong form quản lý |
| Đảm bảo tối thiểu | Hệ thống loại bỏ các thông tin đã thêm và quay lại ở bước trước |
| Đảm bảo thành công | Hiện ra biểu đồ thống kê nhân viên |
| Kích hoạt | Admin ấn vào nút “Xác nhận” |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị biểu đồ nhân viên | |
| Ngoại lệ: Không có | |

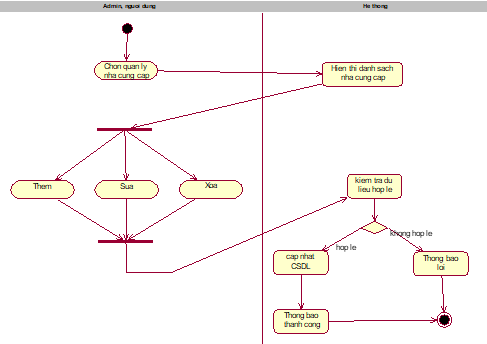
#### Biểu đồ hoạt động chức năng của hệ thống

* Biểu đồ hoạt động chức năng đăng nhập



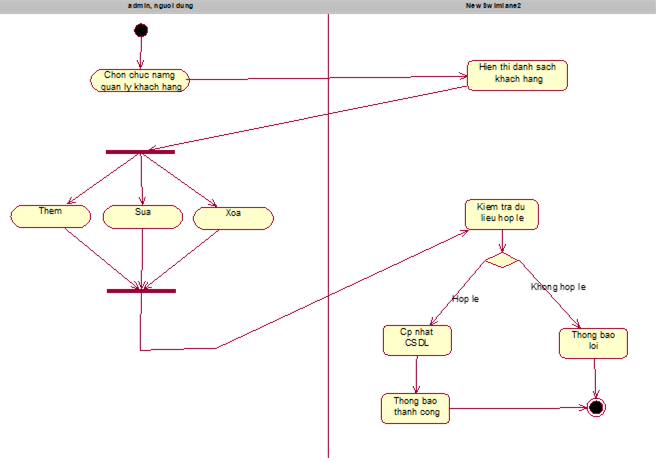
Hình 3. Biều đồ hoạt động chức năng đăng nhập

* Biểu đồ hoạt động chức năng quản lý nhà cung cấp



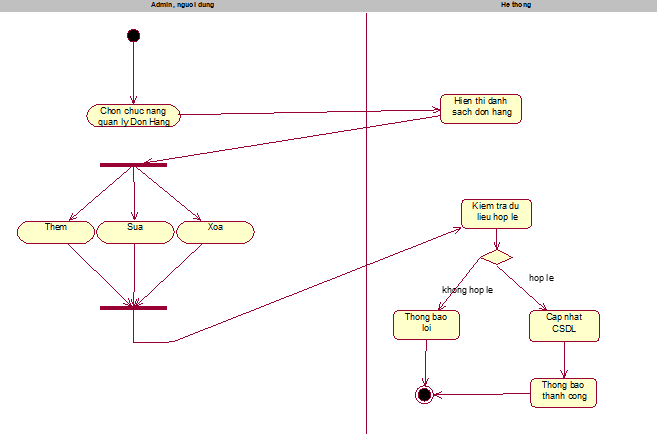
Hình 3. 10 Biểu đồ hoạt động chức năng quản lý nhà cung cấp

* Biểu đồ hoạt động chức năng quản lý khách hàng



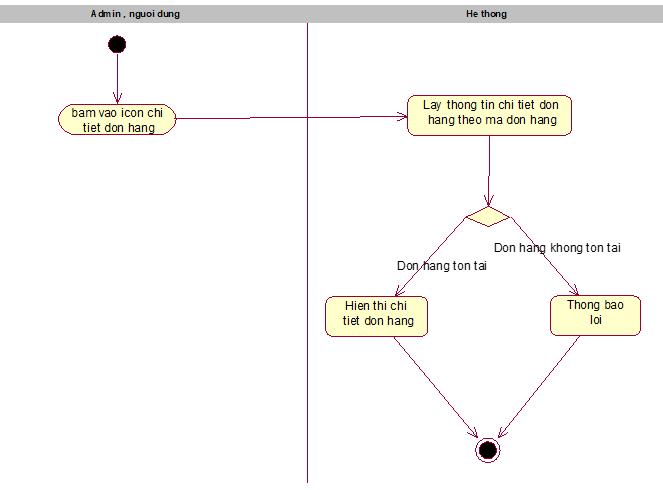
Hình 3. Biểu đồ hoạt động chức năng quản lý khách hàng

* Biểu đồ hoạt động chức năng quản lý đơn hàng



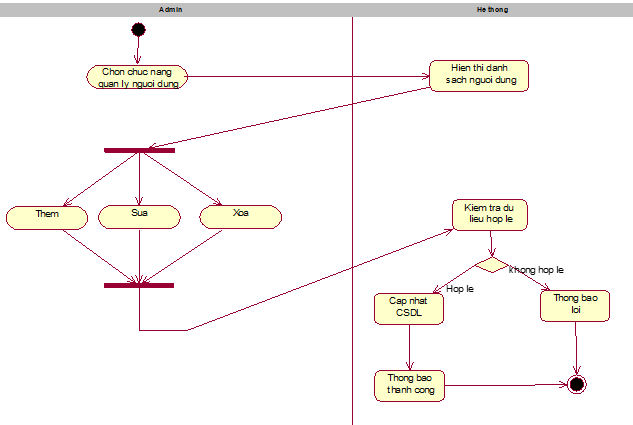
Hình 3. Biểu đồ hoạt động chức năng quản lý đơn hàng

* Biểu đồ hoạt động chức năng xem chi tiết đơn hàng



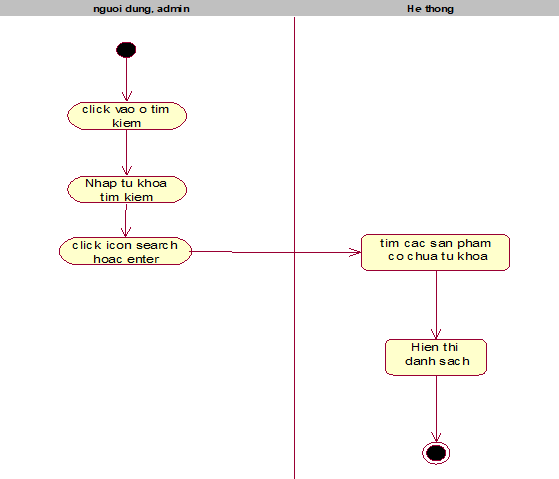
Hình 3. Biều đồ hoạt động chức năng xem chi tiết đơn hàng

* Biểu đồ hoạt động chức năng quản lý người dùng



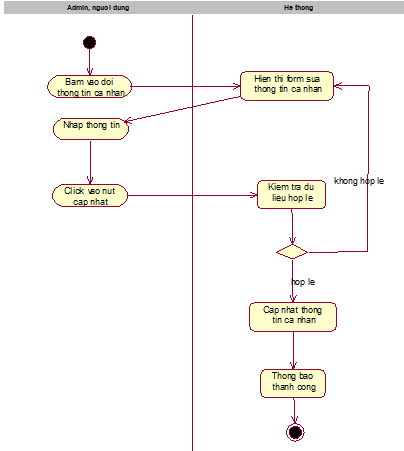
Hình 3. Biểu đồ hoạt động quản lý người dùng

* Biểu đồ hoạt động chức năng tìm kiếm



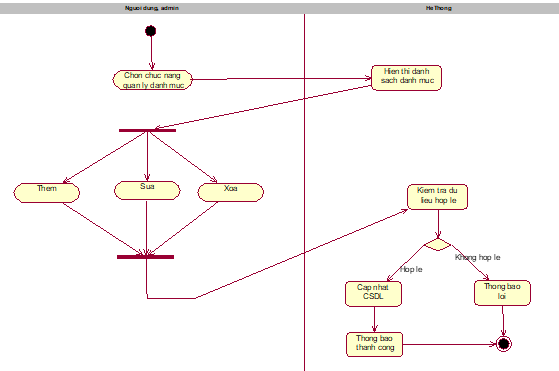
Hình 3. Biều đồ hoạt động chức năng tìm kiếm

* Biểu đồ hoạt động chức năng sửa thông tin cá nhân



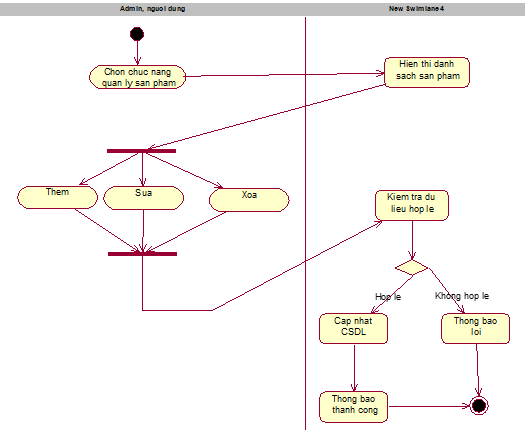
Hình 3. Biều đồ hoạt động chức năng sửa thông tin cá nhân

* Biểu đồ hoạt động chức năng quản lý danh mục



Hình 3. Biều đồ hoạt động chức năng quản lý danh mục

* Biểu đồ hoạt động chức năng quản lý sản phẩm



Hình 3. Biều đồ hoạt động chức năng quản lý sản phẩm

#### Biểu đồ tuần tự chức năng của hệ thống

* Biểu đồ tuần tự chức năng đăng nhập



Hình 3. Biểu đồ tuần tự chức năng đăng nhập

* Biểu đồ tuần tự chức năng quản lý nhà cung cấp

Hình 3. Biều đồ tuần tự chức năng quản lý nhà cung cấp

* Biểu đồ tuần tự chức năng quản lý đơn hàng



Hình 3. Biều đồ tuần tự chức năng quản lý đơn hàng

* Biểu đồ tuần tự chức năng xem chi tiết đơn hàng



Hình 3. Biểu đồ tuần tự chức năng xem chi tiết đơn hàng

* Biểu đồ tuần tự chức năng quản lý nhân viên



Hình 3. Biểu đồ tuần tự chức năng quản lý nhân viên

* Biểu đồ tuần tự chức năng tìm kiếm



Hình 3. Biểu đồ tuần tự chức năng tìm kiếm

* Biểu đồ tuần tự chức năng sửa thông tin cá nhân



Hình 3. Biểu đồ tuần tự chức năng sửa thông tin cá nhân

* Biểu đồ tuần tự chức năng quản lý danh mục

Hình 3. Biểu đồ tuần tự chức năng quản lý danh mục

* Biểu đồ tuần tự chức năng quản lý sản phẩm



Hình 3. Biểu đồ tuần tự chức năng quản lý sản phẩm

#### Danh mục bảng CSDL

* Bảng hoá đơn

Bảng 3. Bảng hóa đơn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| Id\_bill | Integer |  | PK | Id hóa đơn |
| Code | Char | 50 |  | Mã hóa đơn |
| Created\_at | DateTime |  |  | Thời gian tạo |
| Updated\_at | DateTime |  |  | Thời gian cập nhật |
| Status | TinyInt |  |  | Trạng thái |
| Id\_user | Integer |  | FK | Id người dùng |
| Id\_customer | Integer |  | FK | Id khách hàng |

* Bảng chi tiết hoá đơn

Bảng 3. Bảng chỉ tiết hóa đơn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| Id\_bill | Integer |  | FK | Id hóa đơn |
| Id\_fruit | Integer |  | FK | Id sản phẩm |
| Amount | Integer |  |  | Số lượng |

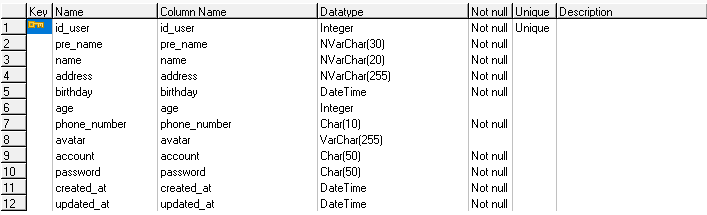
* Bảng nhà cung cấp

Bảng 3. Bảng nhà cung cấp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| Id\_manufacture | Integer |  | PK | Id nhà cung cấp |
| Name | Nvarchar | 50 |  | Tên nhà cung cấp |
| Created\_at | DateTime |  |  | Thời gian tạo |
| Updated\_at | DateTime |  |  | Thời gian cập nhật |
| Phone\_number | Char | 10 |  | Số điện thoại |
| Address | Nvarchar | 255 |  | Địa chỉ |

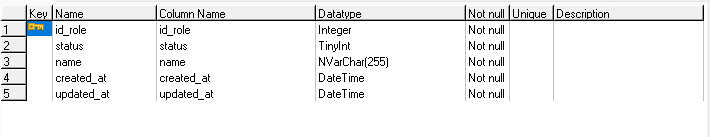
* Bảng người dùng

Bảng 3. Bảng người dùng



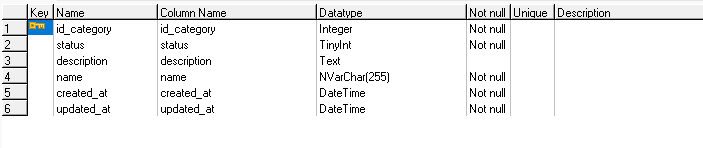
* Bảng quyền

Bảng 3. Bảng quyền



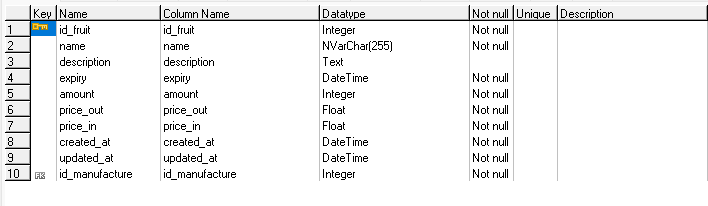
* Bảng danh mục

Bảng 3. Bảng danh mục

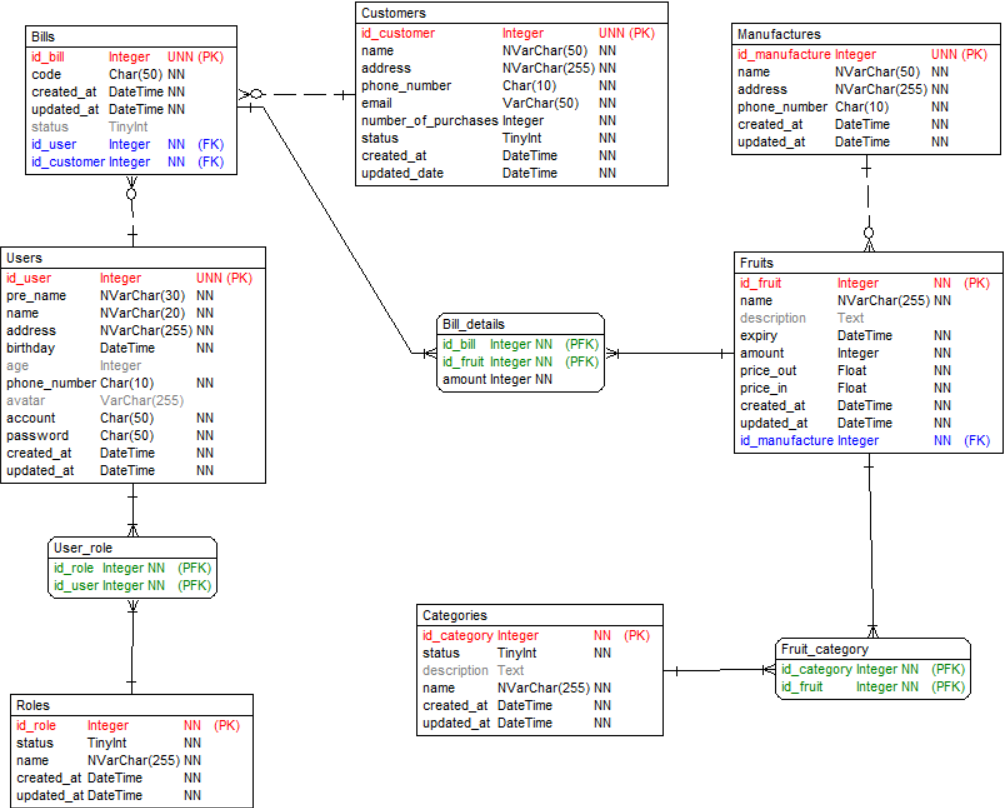


* Bảng sản phẩm

Bảng 3. Bảng sản phẩm



#### Biểu đồ ERD



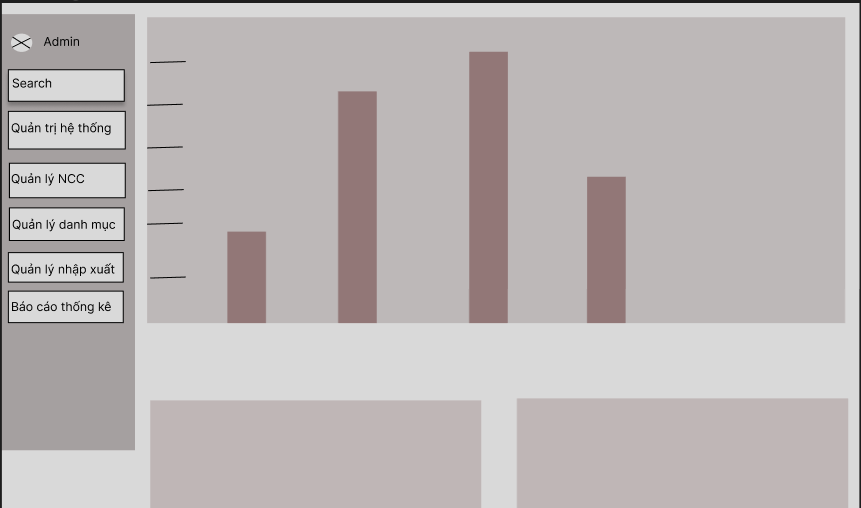
Hình 3. Biểu đồ ERD

### Thiết kế UI/UX

Sau đây là một vài cửa sổ giao diện nhóm em sử dụng Figma để thiết kế:



Hình 3. Giao diện đăng nhập



Hình 3. Giao diện thống kê doanh thu



Hình 3. Giao diện cập nhật thông tin cá nhân



Hình 3. Giao diện thêm nhân viên

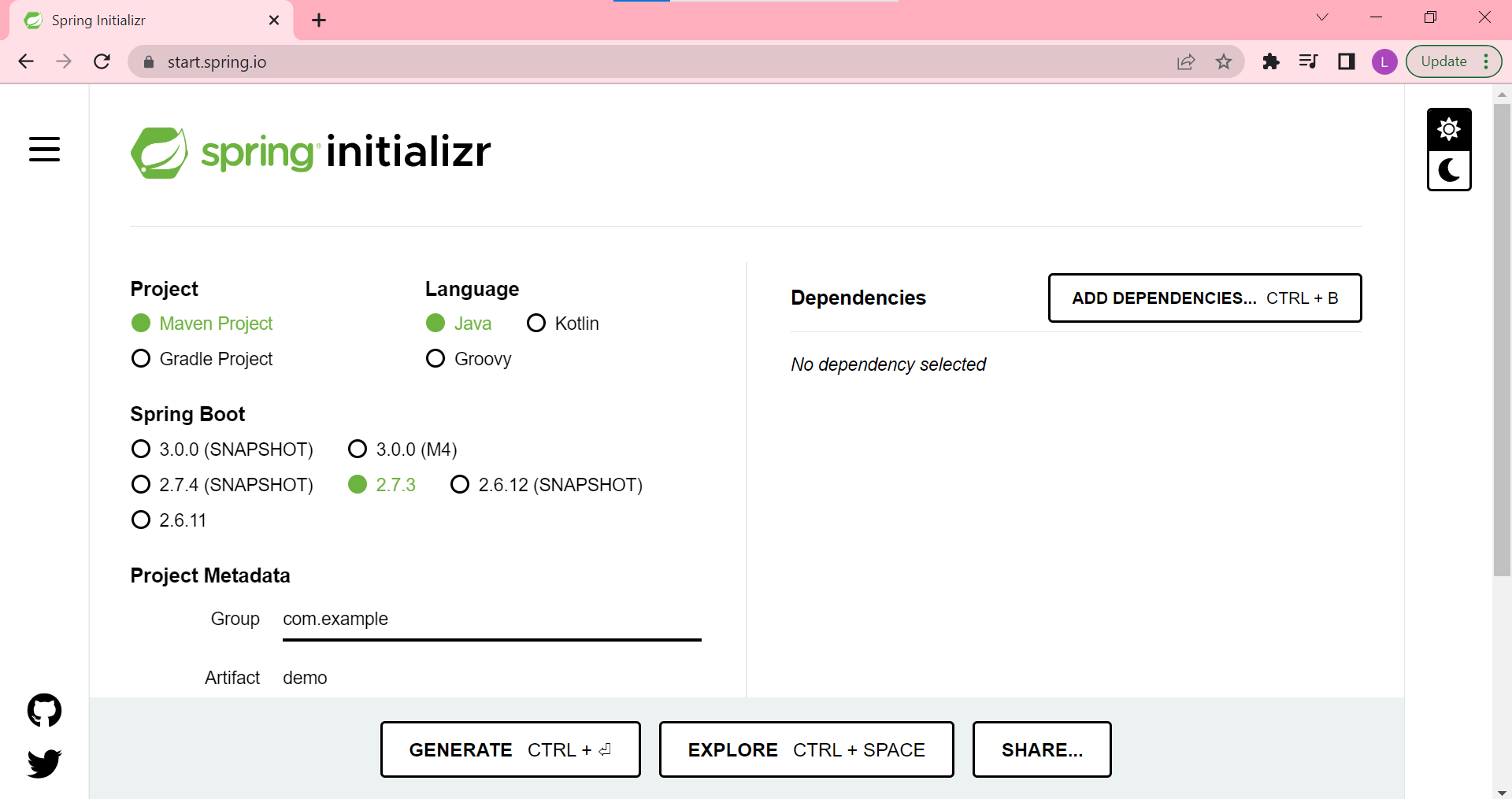
<https://www.figma.com/file/JDStB4mIjuioEbijOwDj9j/GiaoDienPMNM?node-id=0%3A1>

## Cài đặt, thử nghiệm và kết luận

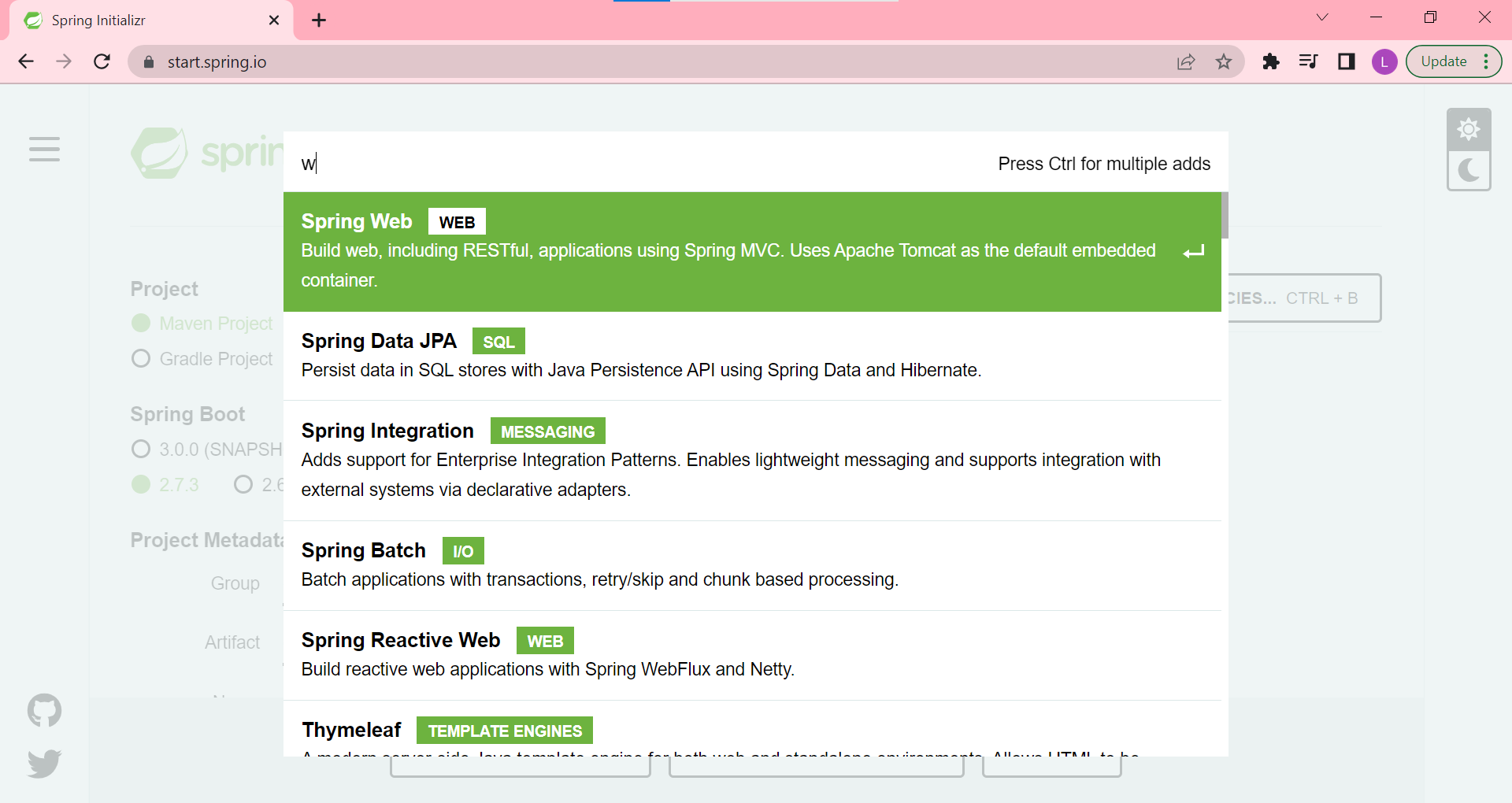
### Cài đặt

Các bước cài đặt 1 project Spring Boot:

* Bước 1: Truy cập đường dẫn <https://start.spring.io/>



Hình 4. 1 Trang web spring initializr

* Bước 2: Chọn loại project, ngôn ngữ, phiên bản Spring Boot, …
* Bước 3: Ấn “Add dependencies” để thêm các thư viện cần thiết.

Hình 4. 2 Giao diện thêm các dependencies

* Bước 4: Chọn “Generate” để lưu project về máy.

### Cấu trúc chương trình

Phần mềm được phát triển trên IDE InteliJ và Visual Studio Code dựa trên nền tảng Java Spring Boot và hệ quản trị cơ sở dữ liệu MS SQL Server

Text

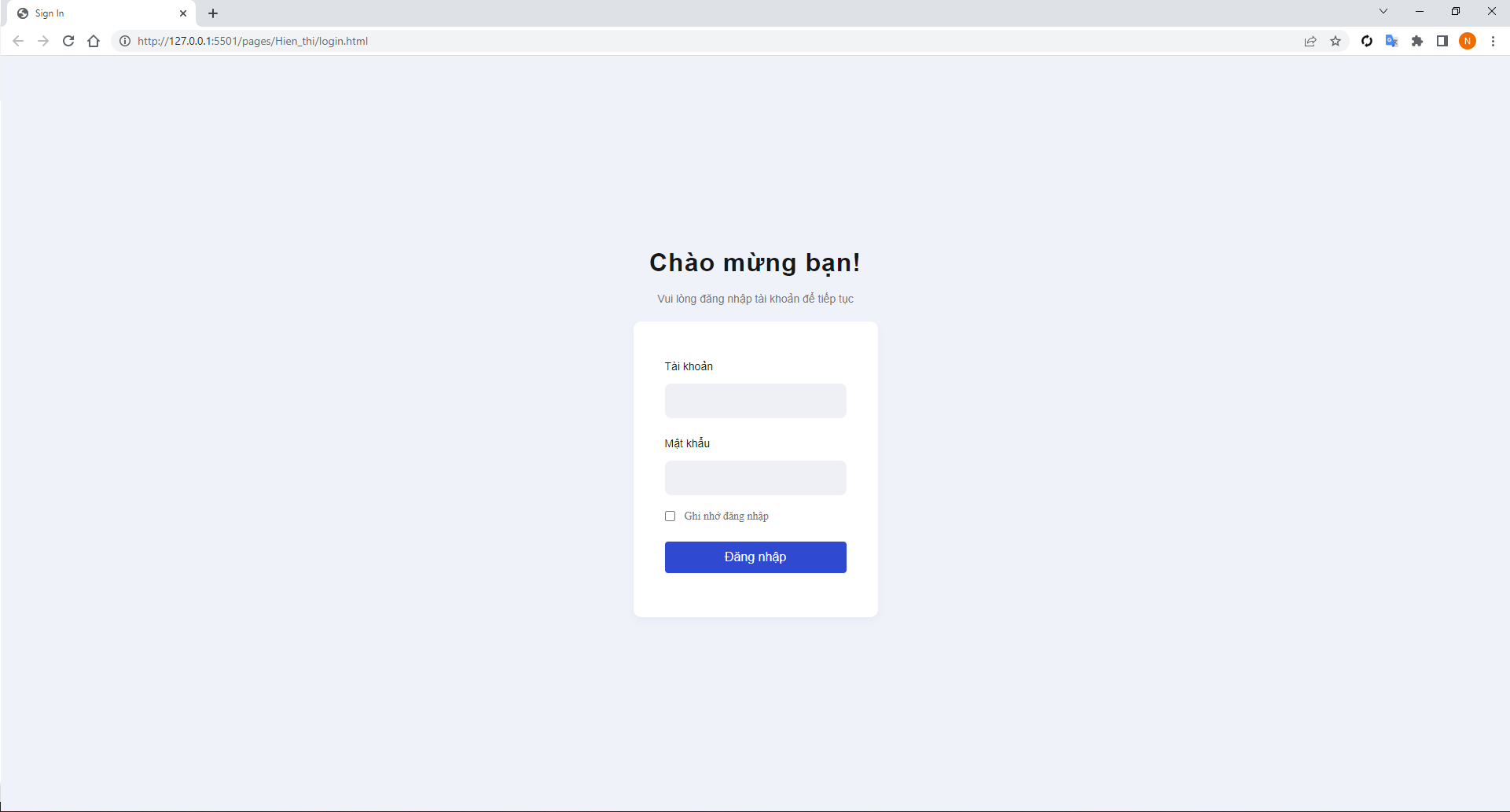
Description automatically generated

Hình 4. 3 Cấu trúc chương trình

### Kết quả nghiên cứu

#### Giao diện đăng nhập

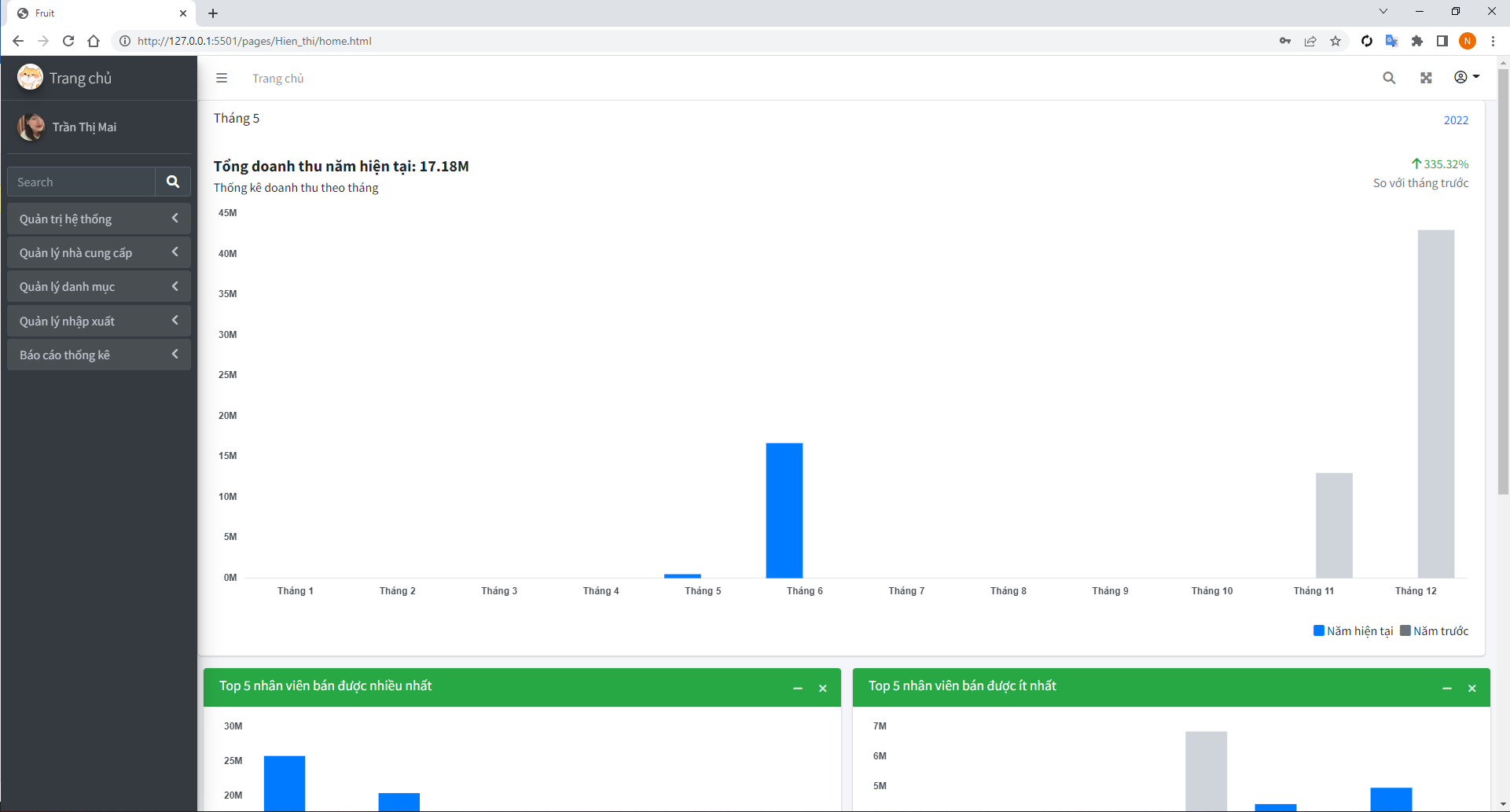
Giao diện hiện ra khi vừa vào hệ thống. Tại đây người dùng phải nhập tài khoản, mật khẩu. Hệ thống sẽ tiến hành phân quyền tại đây. Với người dùng là admin thì sẽ có đầy đủ các chức năng, người dùng là nhân viên nhập kho thì sẽ được nhập kho, người dùng là nhân viên xuất kho thì màn hình sẽ có chức năng xuất kho.

****

Hình 4. 4 Giao diện đăng nhập

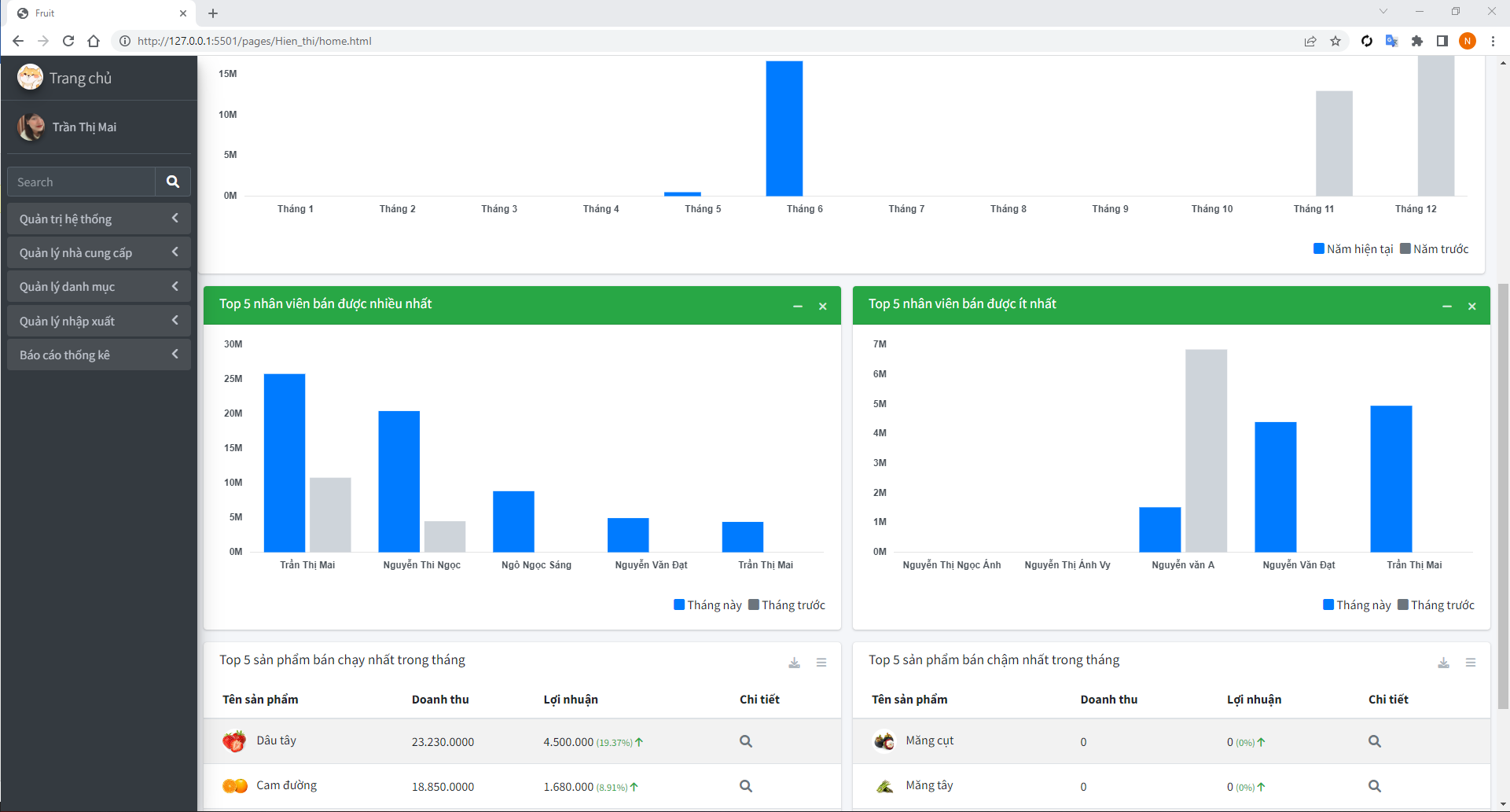
#### Giao diện thống kê

Đây là giao diện của admin, admin có thể xem đầy đủ các thống kê doanh thu theo tháng.



Hình 4. 5 Giao diện thống kê doanh thu

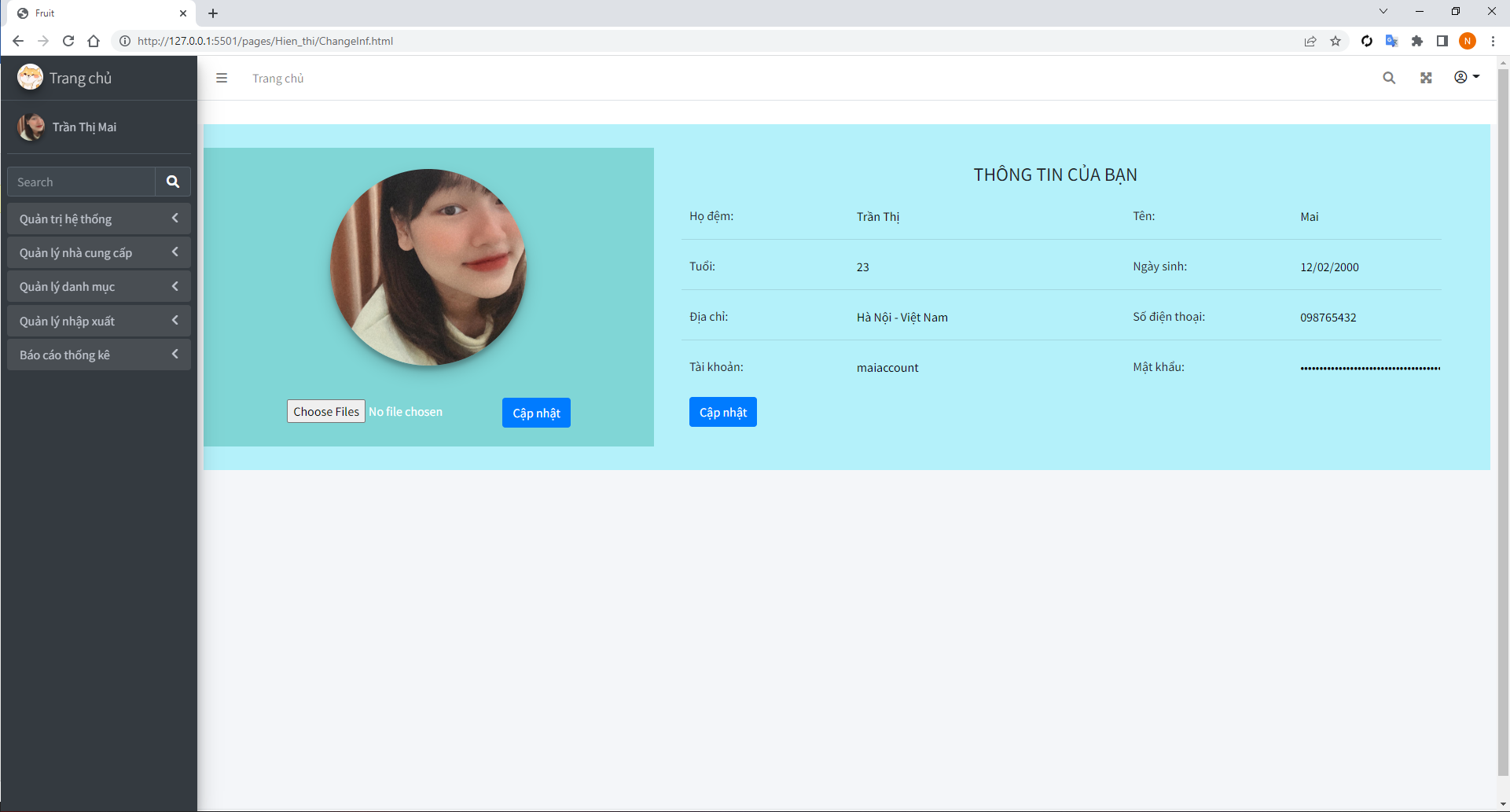
Đây là giao diện, thống kê các mặt hàng bán chạy nhất, thống kê các nhân viên tích cực nhất, thống kê các mặt hàng bị bán chậm nhất, hoặc bị tồn kho, các nhân viên bán được ít hàng nhất, …



Hình 4. 6 Giao diện thống kê sản phẩm và nhân viên

#### Giao diện cập nhật thông tin cá nhân

Đây là giao diện giúp người dùng sửa được thông tin cá nhân của mình. Thông tin sẽ được lưu ở cơ sở dữ liệu và được quản lý bởi người quản lý(admin)

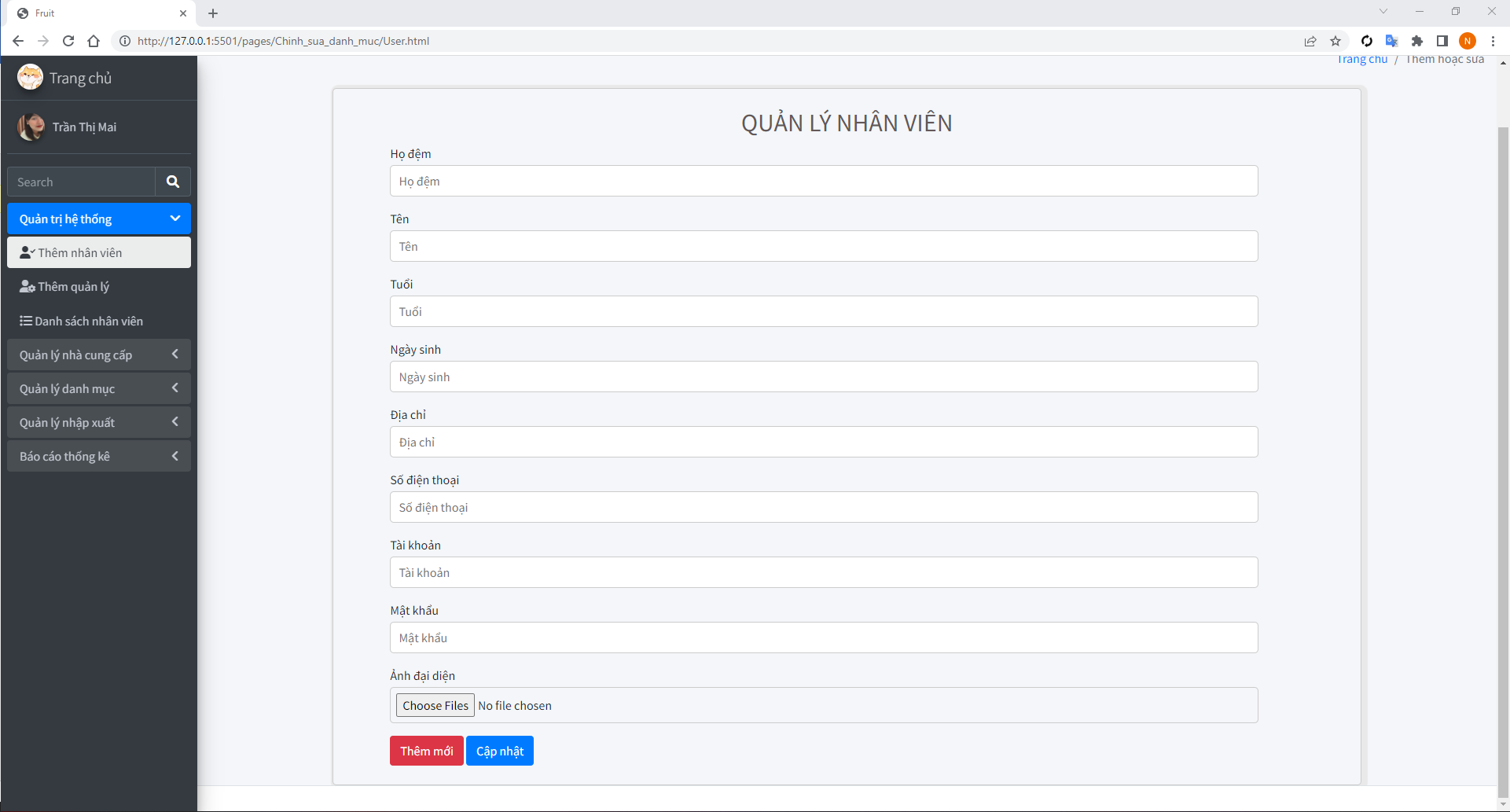


Hình 4. 7 Giao diện cập nhật thông tin cá nhân

#### Giao diện quản trị hệ thống:

* Thêm nhân viên

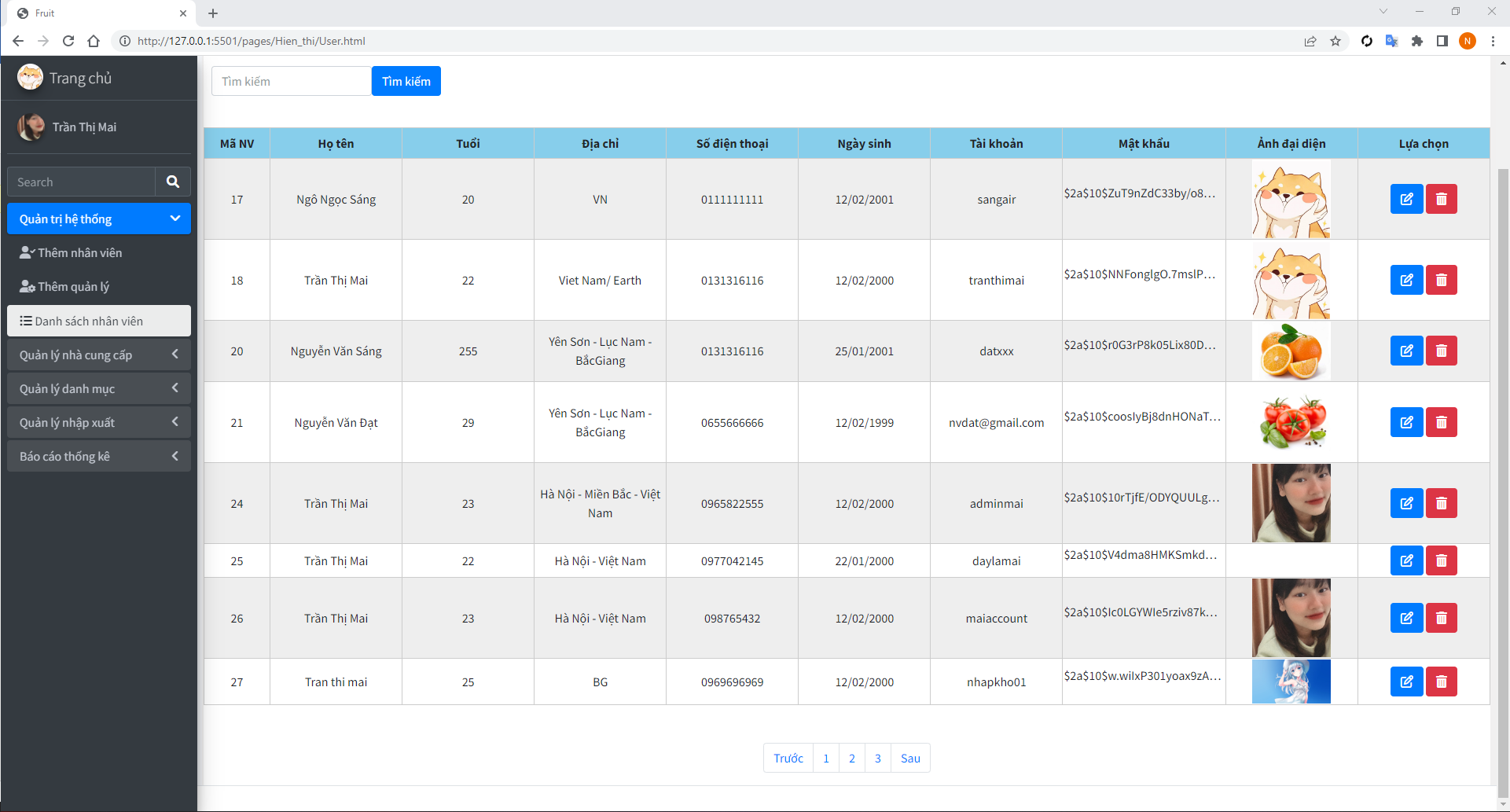
Giao diện thêm nhân viên, đây là chức năng của admin



Hình 4.8 Giao diện thêm nhân viên

* Hiển thị danh sách nhân viên

Đây là giao diện hiển thị danh sách nhân viên, chỉ có admin mới xem được.

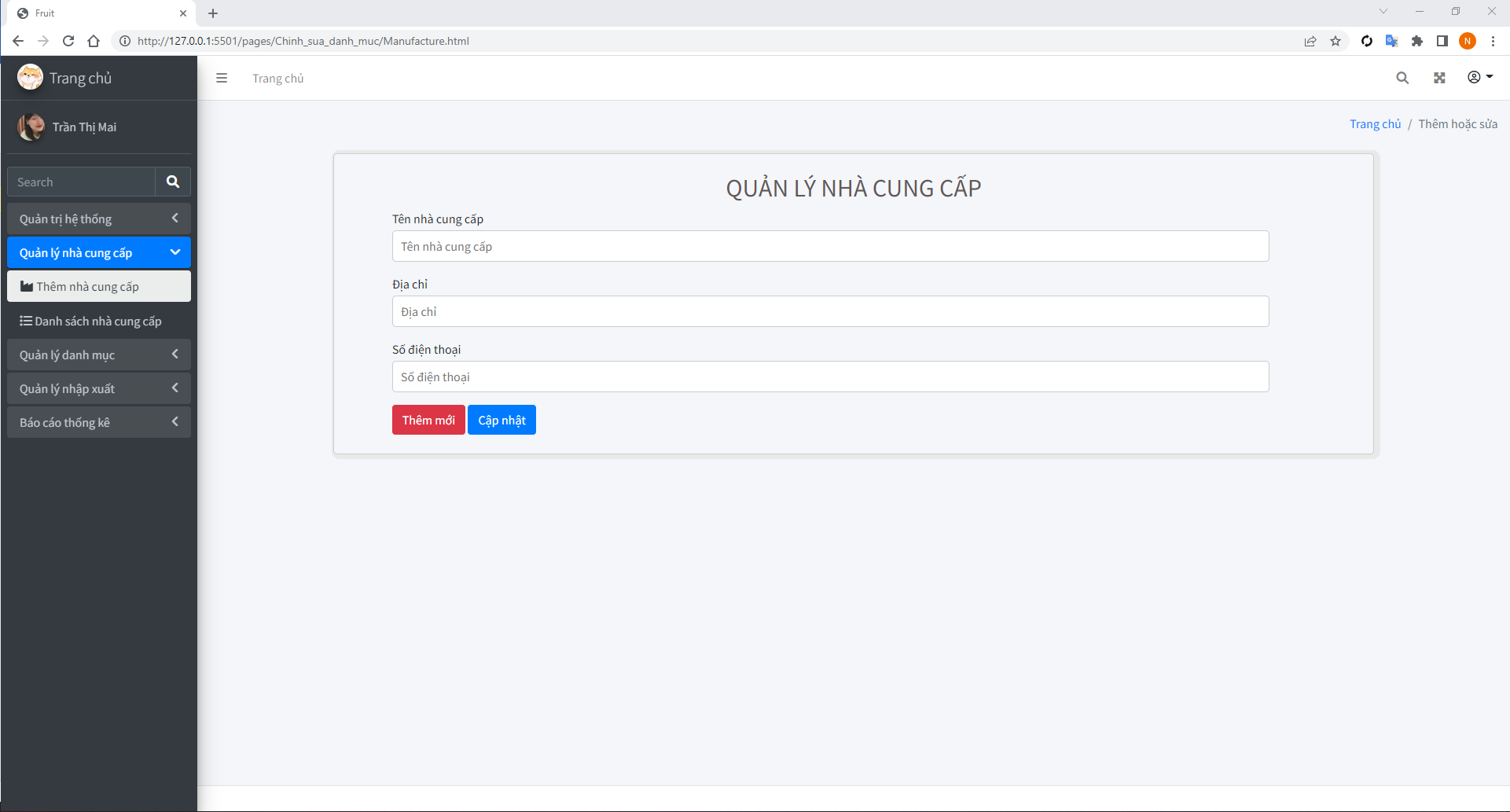


Hình 4.9 Giao diện hiển thị danh sách nhân viên

#### Giao diện quản lý nhà cung cấp:

* Thêm nhà cung cấp

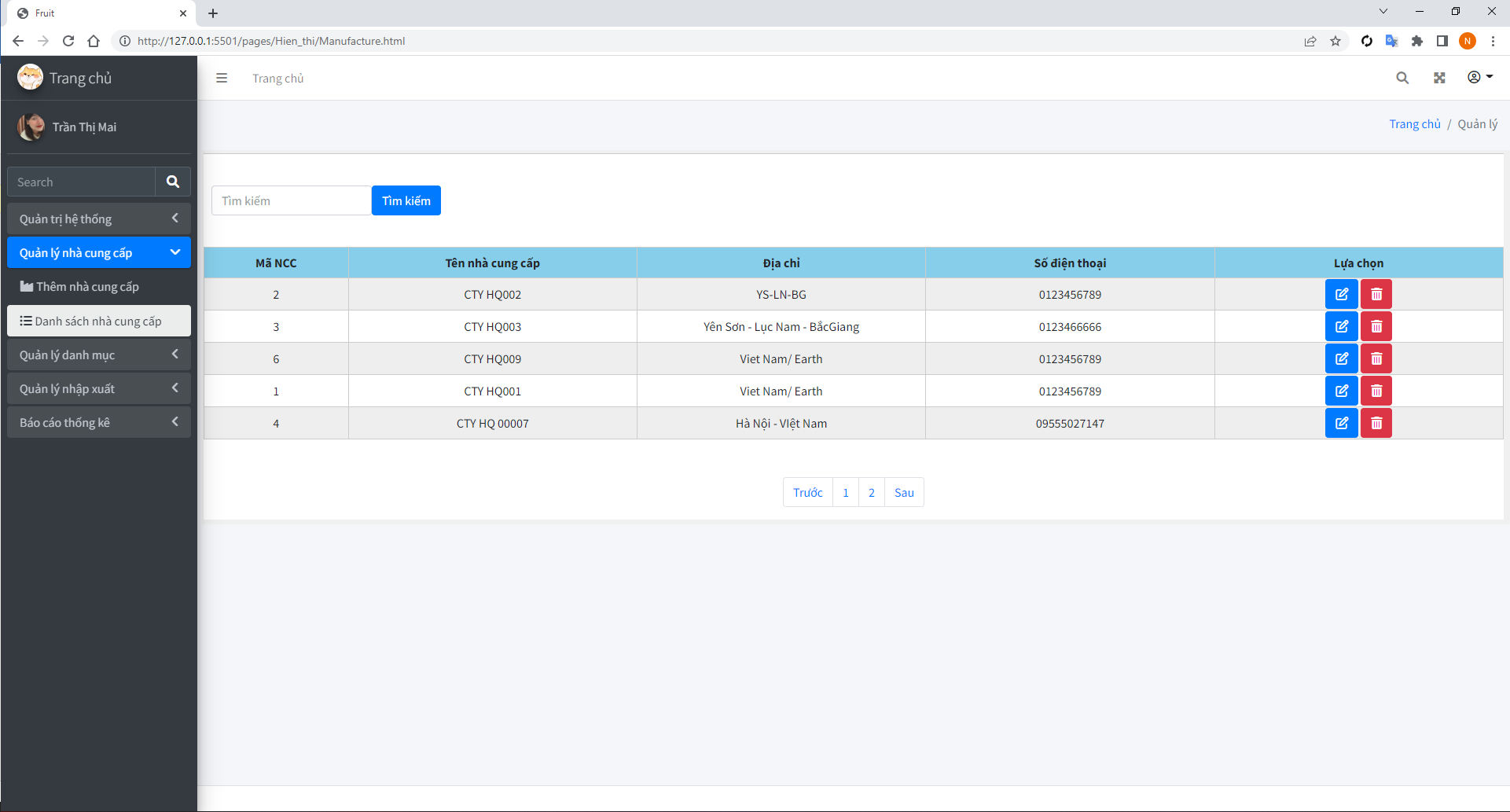
Giao diện thêm nhà cung cấp. Nhà cung cấp chỉ được thêm khi nhà cung cấp có uy tín, đã kí hợp đồng với người quản lý, sau đó người quản lý sẽ thêm nhà cung cấp vào danh sách



Hình 4. 10 Giao diện thêm nhà cung cấp

* Hiển thị danh sách nhà cung cấp

Giao diện hiển thị danh sách nhà cung cấp. Hiển thị các nhà cung cấp đã liên kết với công ty.

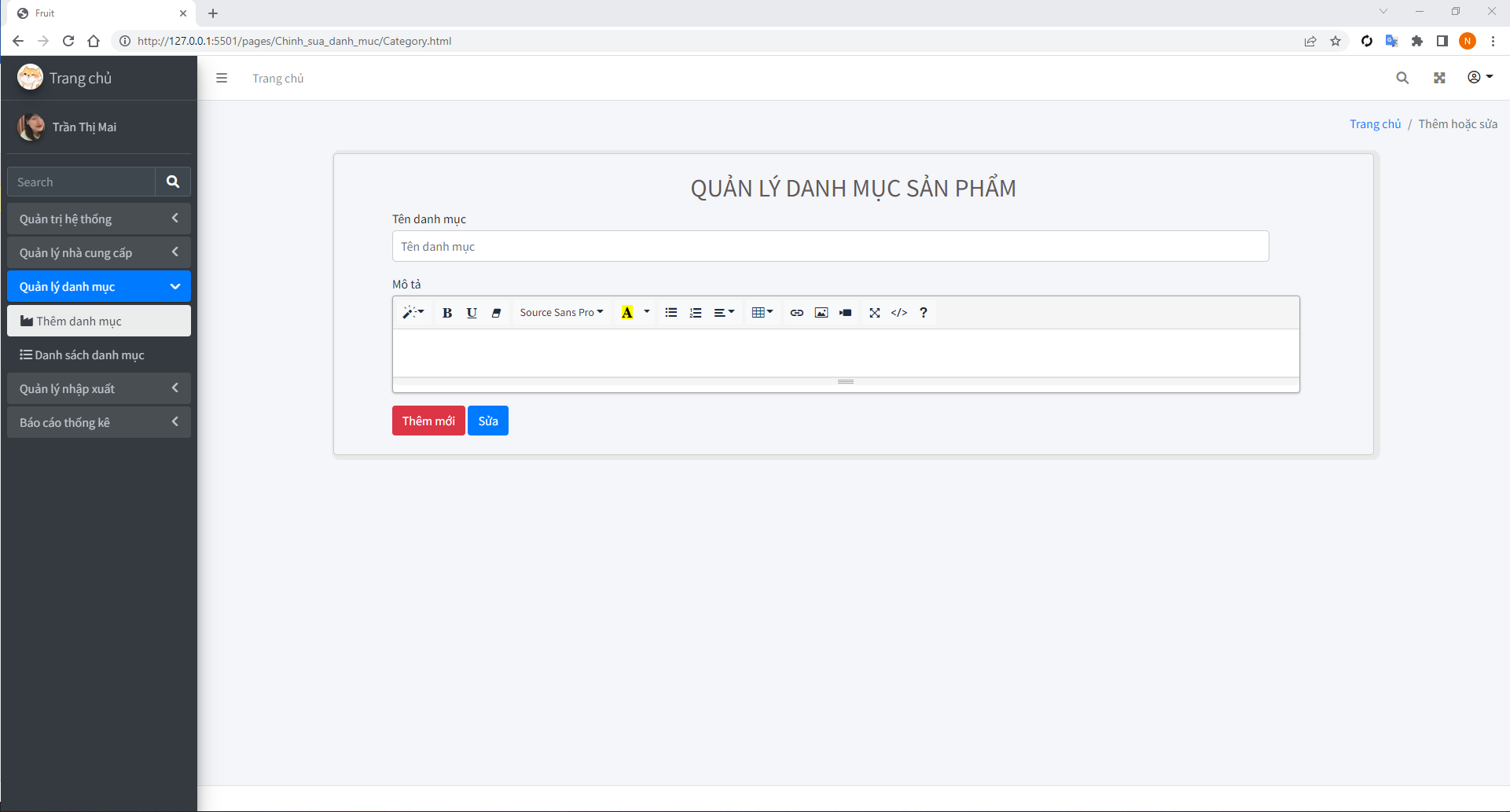


Hình 4. 11 Giao diện hiển thị danh sách nhà cung cấp

#### Giao diện quản lý danh mục

* Thêm danh mục

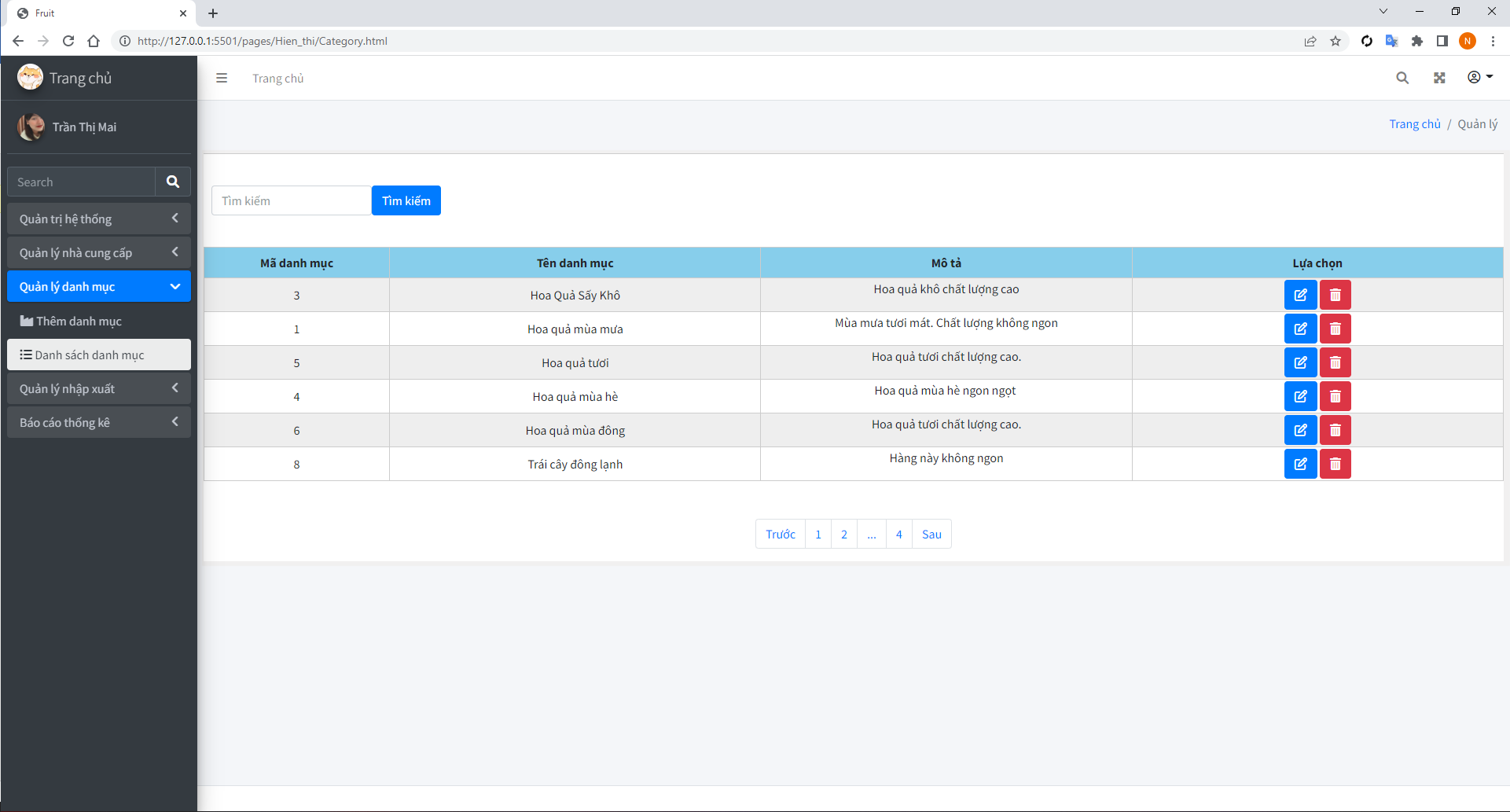
Giao diện thêm danh mục. Danh mục này có thể được thêm mởi admin hoặc là nhân viên. Nhưng nhân viên chỉ được thêm khi đã có sự đồng ý của admin.



Hình 4. 12 Giao diện thêm danh mục

* Hiển thị danh sách danh mục

Giao diện hiển thị toàn bộ danh mục, có thể sửa hoặc xoá bất kì danh mục nào khi không cần thiết, nhân viên có thể xoá khi có sự đồng ý của admin

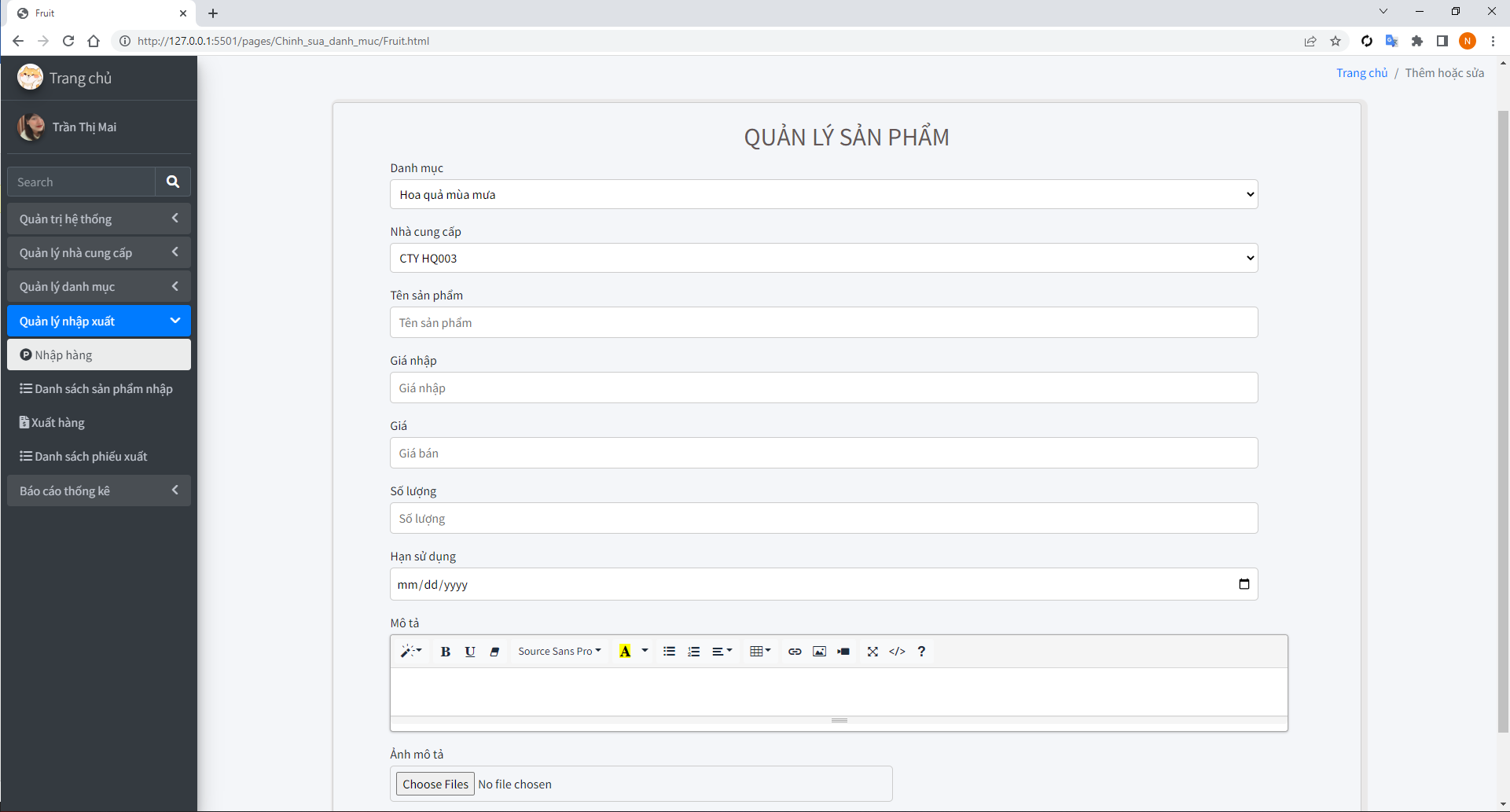


Hình 4. 13 Giao diện hiển thị danh mục

#### Giao diện quản lý nhập xuất:

* Nhập hàng

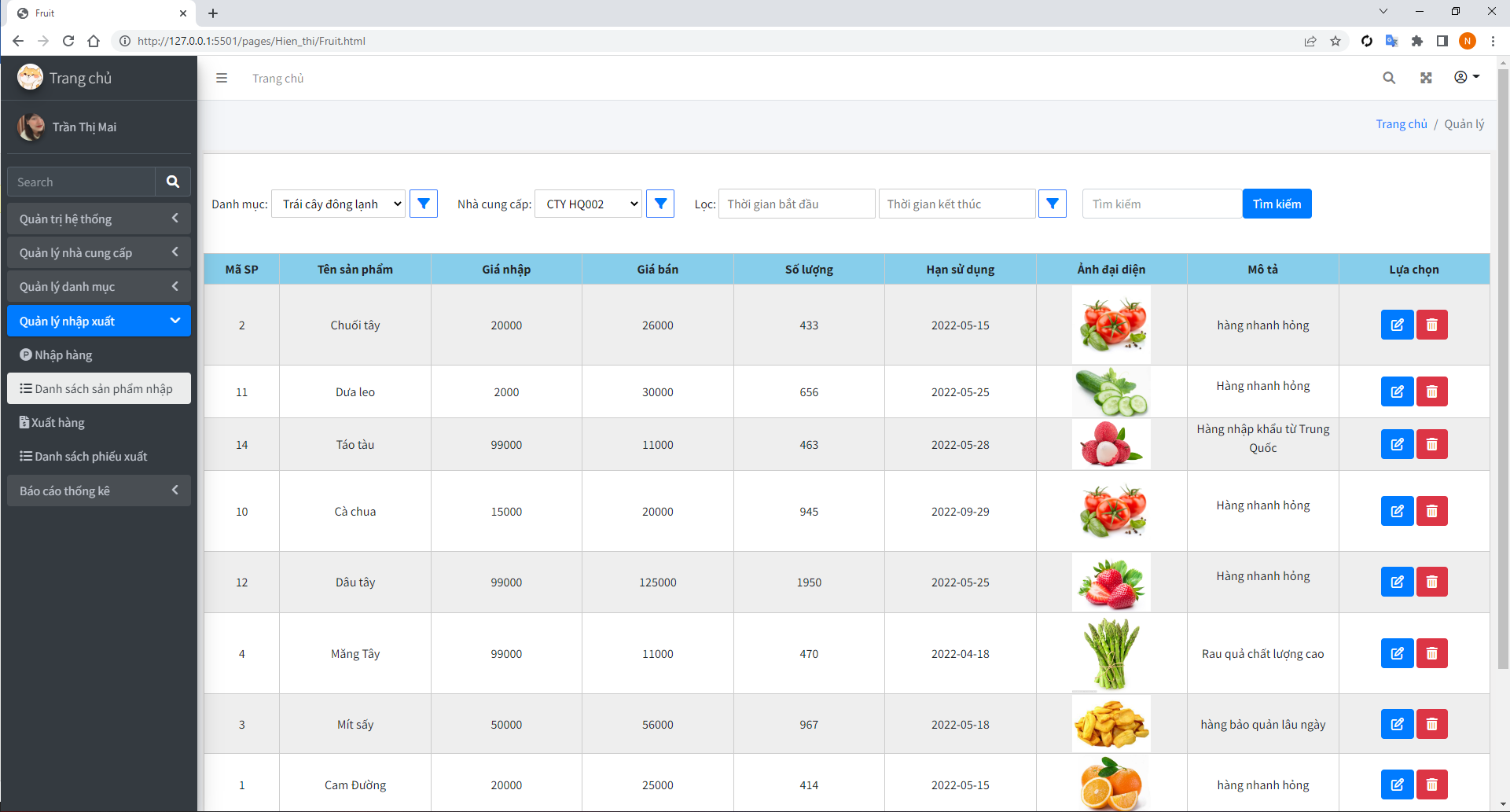
Giao diện nhập hàng vào kho. Đây là chức năng của nhân viên nhập hàng. Nhân viên nhập hàng. Mỗi lần nhập hàng sẽ đều báo về mail của admin với thông tin sản phẩm.



Hình 4. 14 Giao diện nhập kho

* Danh sách sản phẩm nhập

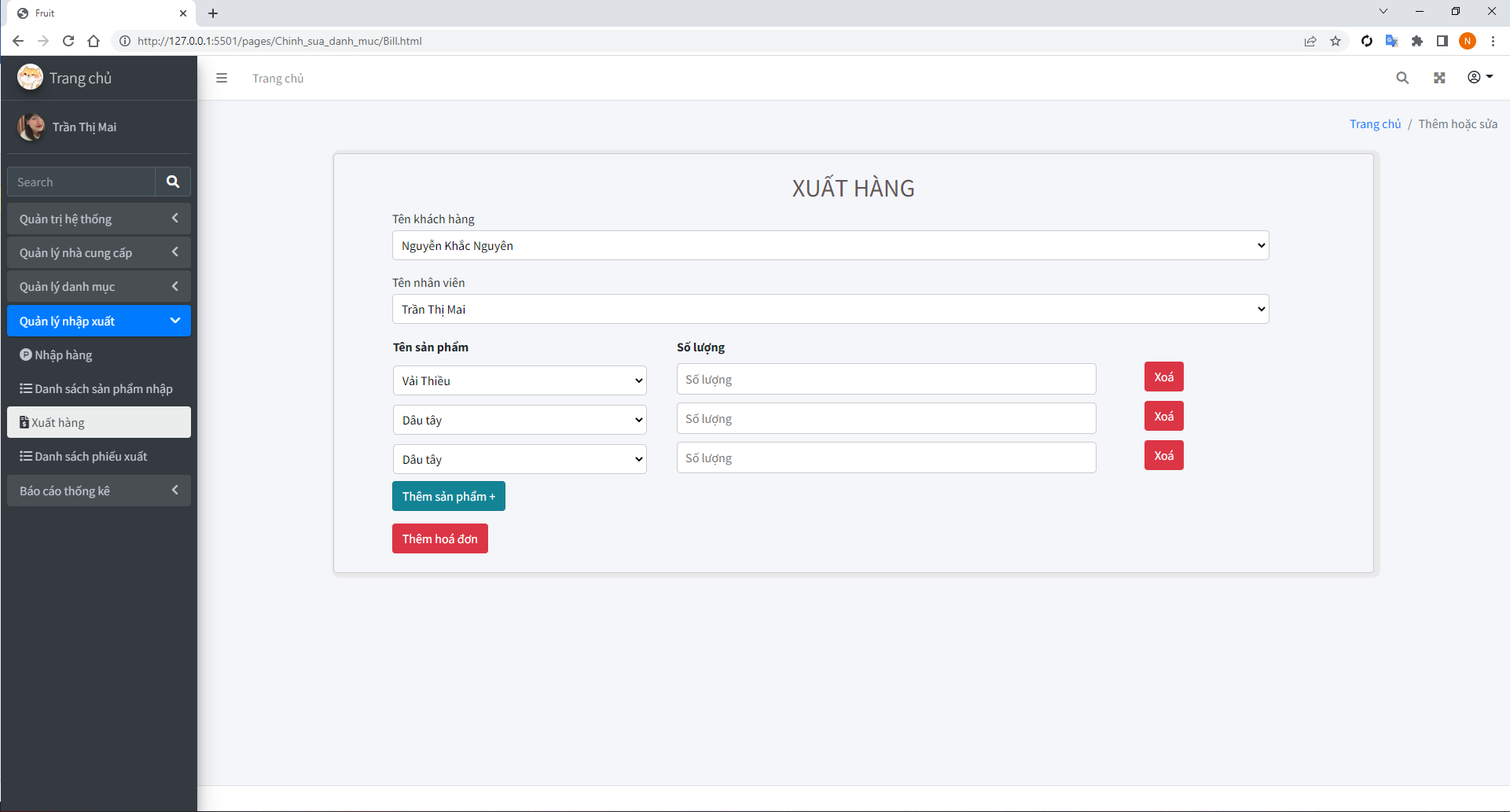
Danh sách các sản phẩm đã được nhập với các số lượng ở trong kho.



Hình 4. 15 Giao diện danh sách các sản phẩm nhập kho

* Xuất hàng

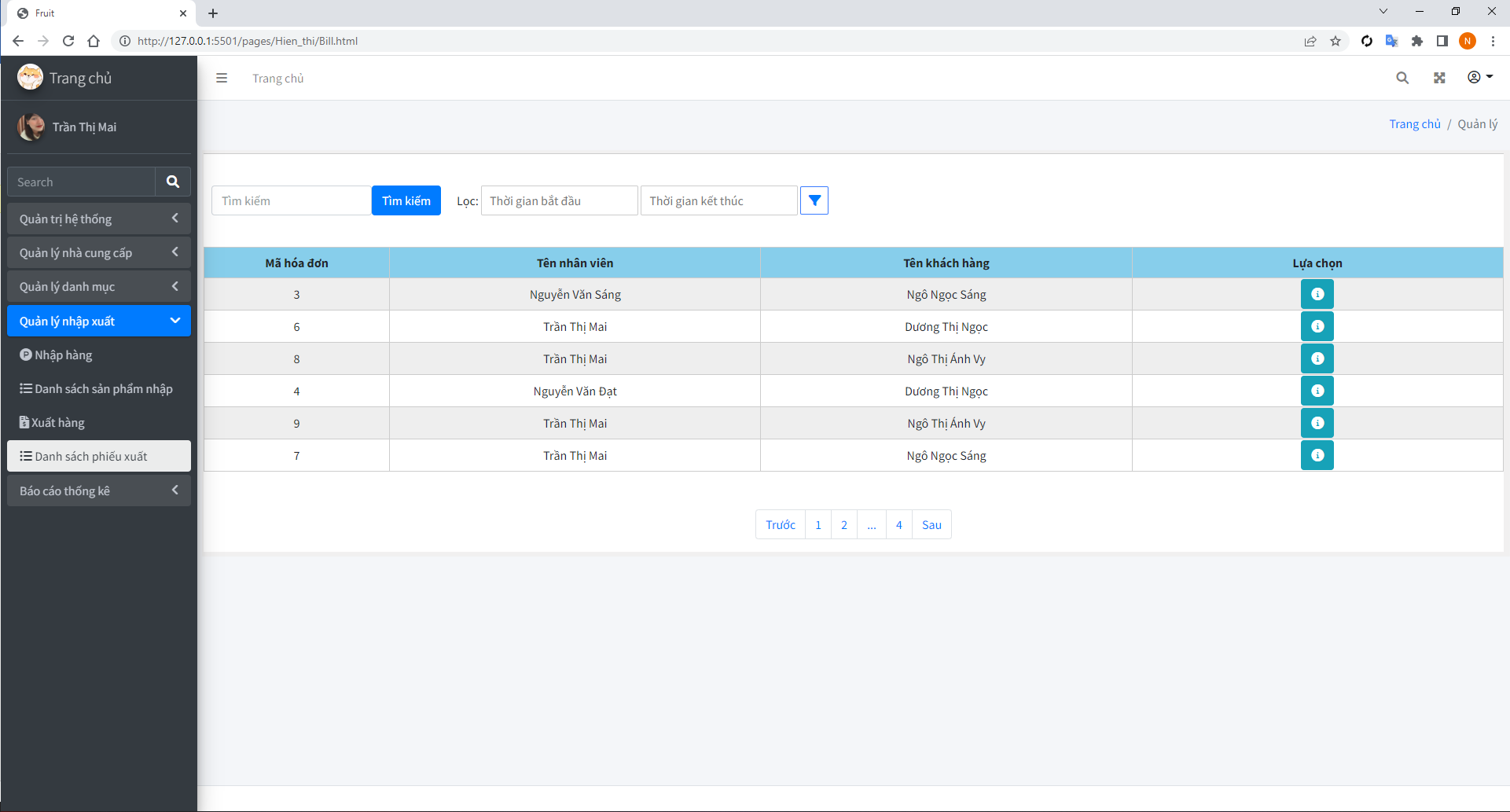
Đây là giao diện xuất hàng hoá ở trong kho, Khi xuất hàng phải đảm bảo là hàng tồn tại ở trong kho với điều kiện số lượng xuất kho phải nhỏ hơn hoặc bằng số lượng có ở trong kho



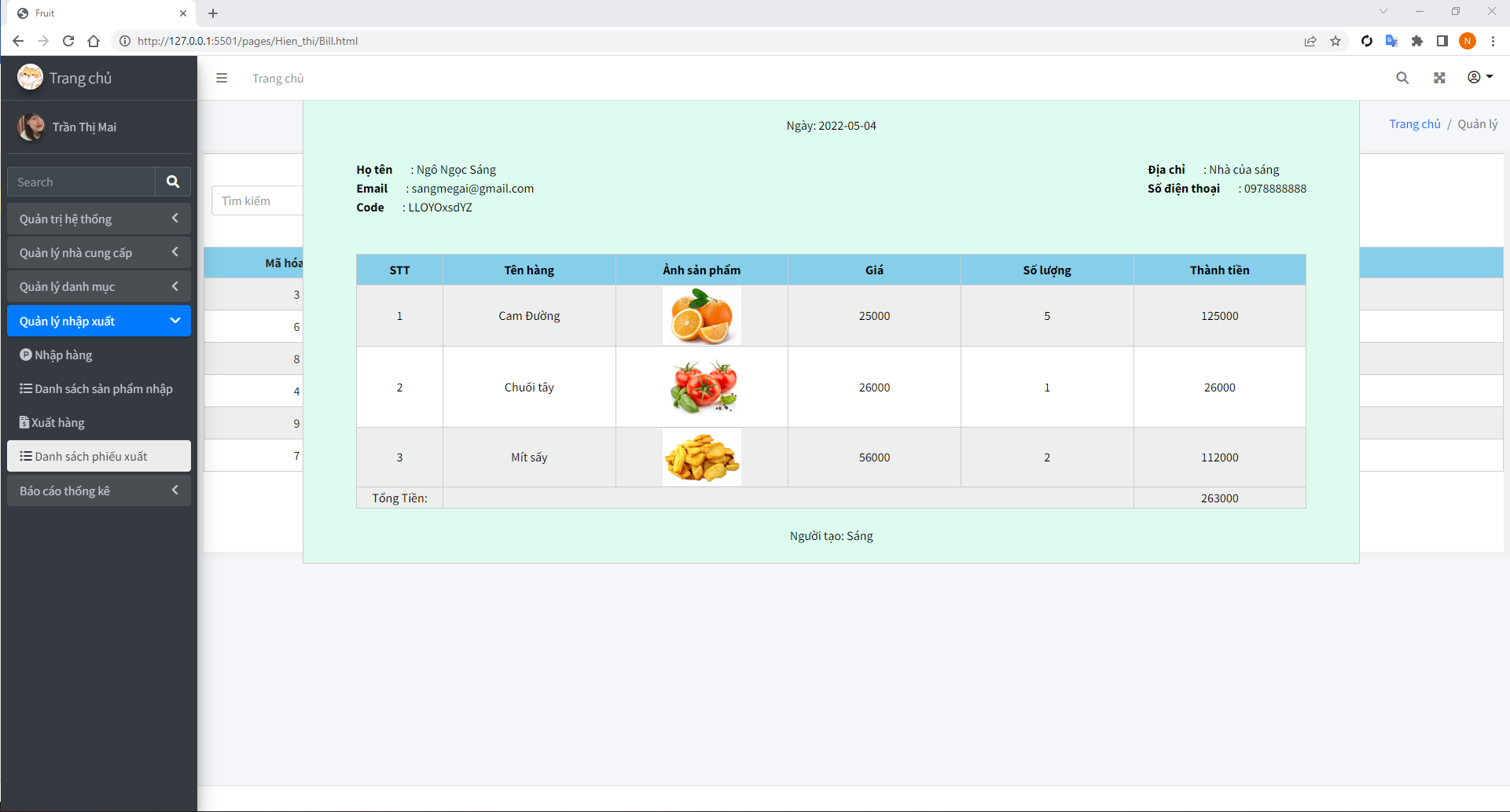
Hình 4. 16 Giao diện xuất kho

* Danh sách phiếu xuất

Danh sách thông tin lịch sử các phiếu xuất kho với đầy đủ các thông tin như ngày tháng, số lượng, tên loại hoa quả, nhân viên tạo phiếu xuất kho và đối tác nhận hàng



Hình 4. 17 Giao diện hiển thị danh sách phiếu xuất



Hình 4. 18 Giao diện chi tiết phiếu xuất

# KẾT LUẬN

Sau một thời gian học tập và nghiên cứu, được sự giúp đỡ tận tình của thầy giáo giảng viên hướng dẫn TS. Hà Mạnh Đào, nhóm em đã nắm bắt được quy trình làm một phần mềm ứng dụng trên web, nắm rõ hơn các kiến thức về Java, cũng như về công nghệ Java Spring. Tuy bản thân nhóm em cũng đã cố gắng tự học hỏi và tiếp thu ý kiến hướng dẫn từ thầy hướng dẫn nhưng cũng không tránh khỏi những thiếu sót. Vậy sau đây em xin nêu lên những kết quả đạt được, những mặt còn hạn chế và hướng phát triển của đề tài “***Phát triển ứng dụng xây dựng website quản lý kho hàng với RESTful Spring Boot***”.

Kết quả đạt được của đề tài:

* Nắm bắt được cách xây dựng lên một project Java với việc cấu hình và sử dụng công nghệ Java Spring
* Khảo sát, phân tích và thiết kê phần mềm quản lý.
* Xây dựng được phần mềm quản lý với các chức năng cơ bản: Quản lý nhân viên, quản lý danh mục, quản lý nhà cung cấp, quản lý sản phẩm, quản lý hoá đơn.

Hạn chế của đề tài:

* Chưa xây dựng được chức năng Admin nhắn tin với nhân viên realtime
* Chưa xây dựng dược tốt giao diện.

Hướng phát triển:

* Xây dựng chức năng Admin nhắn tin với nhân viên realtime
* Thiết kế giao diện nổi bật hơn nữa
* Nhanh chóng hoàn thiện và đưa vào sử dụng ở công ty

# TÀI LIỆU THAM KHẢO

[1] <https://cuuduongthancong.com/s/phan-mem-ma-nguon-mo-va-linux>

[2] <https://tailieu.vn/>