Trường ĐH Công Nghệ Sài Gòn

**KHOA CÔNG NGHỆ THÔNG TIN**

LUẬN VĂN TỐT NGHIỆP

*Tên đề tài:*

ỨNG DỤNG REACTJS VÀ SPRING BOOT ĐỂ XÂY DỰNG WEBSITE BÁN HOA QUẢ

TPHCM – Năm 2020

Trường ĐH Công Nghệ Sài Gòn

**KHOA CÔNG NGHỆ THÔNG TIN**

LUẬN VĂN TỐT NGHIỆP

*Tên đề tài:*

ỨNG DỤNG REACTJS VÀ JAVA SPRING BOOT ĐỂ XÂY DỰNG WEBSITE BÁN HOA QUẢ

Người hướng dẫn:

Sinh viên thực hiện:

1. Đặng Thanh Hiếu - DH51500088

2. Phan Lê Trọng Tiến - DH51500081

TPHCM – Năm 2020

LỜI CẢM ƠN

Để thực hiện và hoàn thành đề tài luận văn này, chúng em đã được sự giúp đỡ từ những người xung quanh. Đặc biệt hơn nửa là sự hợp tác của cán bộ giáo viên trường.

Trước hết chúng em xin gửi lời cảm ơn sâu sắc đến thầy Bằng là người trực tiếp hướng dẫn đã luôn dành nhiều thời gian, công sức chỉ dạy chúng em trong suốt quá trình thực hiện nghiên cứu và hoàn thành đề tài luận văn tốt nghiệp.

Chúng em xin chân thành cám ơn đến Ban giám hiệu cùng toàn thể các thầy cô giáo công tác trong trường Đại học Công Nghệ Sài Gòn đã tận tình truyền đạt những kiến thức quý báu, giúp đỡ chúng em trong suốt quá trình học tập và nghiên cứu.

Cảm ơn thầy cô Khoa Công Nghệ Thông Tin đã truyền đạt tri thức, kinh nghiệm thông qua các môn học bài giảng và quan trọng hơn là niềm đam mê trong công việc dể hoàn thành đề tài luận văn một cách tốt đẹp.

Tuy có nhiều cố gắng, nhưng không tránh khỏi còn nhiều thiếu sót. Chúng em kính mong Quí thầy côm nhà trường, những người quan tâm đến đề tài, gia đình và bạn bè tiếp tục đóng góp ý kiến của mình, giúp đỡ để đề tài hoàn thiện hơn nữa.

Chúng em xin chúc quý thầy cô luôn tràn đầy sức khỏe tiếp tục giáo dục tri thức và truyền đạt động lực nhiệt huyết cho nhiều thế hệ học sinh, sinh viên

Chúng em xin chân thành cảm ơn!

Mục lục

[Chương 1. Giới thiệu 7](#_Toc43965623)

[1.1 Đặt vấn đề 7](#_Toc43965624)

[1.2 Mục tiêu của đề tài 9](#_Toc43965625)

[Chương 2. Tổng quan REACTJS 10](#_Toc43965626)

[2.1 Giới thiệu 10](#_Toc43965627)

[2.2 Lịch sử phát triển 11](#_Toc43965628)

[2.3 Tình hình ứng dụng ReactJS trên thế giới 11](#_Toc43965629)

[2.4 Các chức năng chính của REACTJS 12](#_Toc43965630)

[2.5 Virtual DOM 13](#_Toc43965631)

[2.6 Component 14](#_Toc43965632)

[2.7 Render 16](#_Toc43965633)

[2.8 JSX 16](#_Toc43965634)

[2.9 Lifecycle 17](#_Toc43965635)

[2.10 Prop and State 18](#_Toc43965636)

[2.11 React Router 20](#_Toc43965637)

[2.12 Global state 25](#_Toc43965638)

[2.13 Công cụ cho Dev 26](#_Toc43965639)

[Chương 3. Tổng quan về Spring Boot 28](#_Toc43965640)

[3.1 Giới thiệu 28](#_Toc43965641)

[3.2 Lịch sử phát triển 28](#_Toc43965642)

[3.3 Tình hình ứng dụng Spring Boot trên thế giới 29](#_Toc43965643)

[3.4 Các chức năng chính của Spring Boot 32](#_Toc43965644)

[3.5 System Requirements 45](#_Toc43965645)

[Chương 4. Ứng Dụng Website Bán Hoa Quả 47](#_Toc43965646)

[4.1 Giới thiệu 47](#_Toc43965647)

[4.2 Yêu cầu khách hàng 47](#_Toc43965648)

[4.3 Quy trình đặt hàng của khách hàng 47](#_Toc43965649)

[4.4 Quy trình bán hàng 48](#_Toc43965650)

[4.5 Yêu cầu quản trị 48](#_Toc43965651)

[4.6 Quy trình giao hàng 48](#_Toc43965652)

[4.7 Quy trình trả hàng 48](#_Toc43965653)

[Chương 5. Ứng dụng 50](#_Toc43965654)

[5.1 Sơ đồ chức năng 50](#_Toc43965655)

[5.2 Sơ đồ use case 51](#_Toc43965656)

[5.3 Sơ đồ tuần tự 65](#_Toc43965657)

[Chương 6. Thành phần dữ liệu 66](#_Toc43965658)

[6.1 Phân tích ở mức quan niệm 66](#_Toc43965659)

[6.2 Thiết kế cơ sở dữ liệu 71](#_Toc43965660)

[Chương 7. Thành phần giao diện 74](#_Toc43965661)

[7.1 Người dùng 74](#_Toc43965662)

[7.2 Quản lý 75](#_Toc43965663)

Danh sách các hình vẽ và bảng

[Hình 2‑1. Web frameworkđược yêu thích nhất vào năm 2019 [2] 12](#_Toc43965664)

# Giới thiệu

## Đặt vấn đề

Hiện nay chúng ta đang sống trong thời đại 4.0 việc sử dụng công nghệ trong kinh doanh dù là quy mô lớn hay nhỏ thì công nghệ vẫn mang đến những lợi ích giúp các nhà kinh doanh tạo ra lợi nhuận cũng như đáp ứng được các yêu cầu của khách hàng.Trong môi trường kinh doanh hiện nay việc giao tiếp với khách hàng một cách nhanh chóng rõ ràng mà chi tiết là rất cần thiết khi khách hàng sử dụng công nghệ để tương tác với nhà kinh doanh thì nhà kinh doanh sẽ ảnh hưởng rất lớn vì đây là cách hữu hiệu giúp doanh nghiệp tạo dựng hình ảnh trước công chúng.Công nghệ kiểm kê kho giúp các chủ kinh doanh biết cách làm thế nào để quản lý tốt nhất các chi phí lưu trữ sản phẩm. Ngoài ra với công nghệ hiện đại, các chủ kinh doanh có thể tiết kiệm thời gian và tiền bạc.Thông qua mạn Internet việc bán hàng có thể được diễn ra liên tục 24/7.Khách hàng có thể truy cập vào trang web từ văn phòng hay nhà riêng thông qua mạng internet.Khách hàng là yếu tố quyết định tới doanh số bán hàng cũng như sự sống còn của doanh nghiệp. Việc quản lý khách hàng trở nên dễ dàng hơn rất nhiều khi có sự hỗ trợ đắc lực của công nghệ. Ngoài việc quản lý danh sách và thông tin khách hàng, nó còn giúp xác định các khách hàng tiềm năng và tự động tiến hành dịch vụ điện tử về chăm sóc khách hàng một cách hiệu quả nhất sự hỗ trợ của công nghệ khiến mọi việc trở nên dễ dàng hơn rất nhiều, thay vì mất nhiều thời gian vào việc ghi chép sổ sách, giấy giờ và làm các thao tác trên máy tính trở nên đơn giản hơn thì chỉ việc sử dụng một phần mềm quản lý hiệu quả doanh số bán hàng, doanh thu, chi phí. Hiện nay có rất nhiều công nghệ lập trình web ra đời và phát triển ngày càng nhiều, có thể kể đến như PHP, Python, ASP.net, java, ... thêm vào đó là các công nghệ lập trình web thực hiện trên nền tảng các framework như: Laravel, Nodejs, Angular js, React js, Symfony, Java Spring. Một trang web hoạt động về thương mại điện tử thì cần có tính nhanh và bảo mật tốt cho dữ liệu của trang web là một điều rất quan trọng trong số đó sẻ chọn công nghệ ReactJS và Spring Boot, để xây dựng một website thương mại điện tử.

* Vì sao chọn ReactJS và JAVA Spring Boot để nghiên cứu

Hiện nay các trang thương mại điện tử lớn như chotot, shopee, tiki , lazada đều sử dụng công nghệ reactJS cho website.Trước 2009 ông lớn mạng xã hội Facebook xây dựng website của mình bằng angularJS việc load dữ liệu và tìm kiếm rất lâu nên các lập trình viên đã xây dựng nên framework react để khắc phục những nhược điểm đó và hiện nay trang facebook sử dụng hầu hết là reactjs tốc độ load dữ liệu và tìm kiếm được cải thiện rất nhiều với lượng dữ liệu lớn như hiện nay Google đã nói hết lần này đến lần khác rằng một trang web nhanh giúp bạn có thứ hạng tốt hơn. Thời gian gần đây, Google đã ra mắt thuật toán “Cập nhật tốc độ”, biến nó trở thành một tiêu chí xếp hạng cho các tìm kiếm trên thiết bị di động. Google nhấn mạnh rằng các trang web có tốc độ tải nhanh sẽ dễ dàng nhận được chiến thắng hơn trong cuộc đua SEO cho nên ReactJS thân thiện với SEO. Một trong những vấn đề lớn của Javascript Framework là nó không thân thiện với các cỗ máy tìm kiếm. React có thể chạy tại phía client và cũng có thể chạy tại phía Server vì vậy nó không gây ra khó khăn gì cho các Search Engine.

Hiện nay rất nhiều doanh nghiệp lớn sử dụng Java để viết ứng dụng website, android ….vì Java có những tính chất như đơn giản , bảo mật ,hướng đối tượng,hiệu suất cao,đa nền tảng, tính chịu tải tốt, có thể mở rộng Scale,hỗ trợ lập trình bằng nhiều phần mềm khác nhau và nó giúp làm giảm chi phí lập trình và tối đa quá trình phát triển trang web. Spring Boot  là một phần quan trọng của Spring Framework(có nền tảng là Java nên nó có bao gồm các chức năng của Java) , nó nhẹ đến bất ngờ nhưng vẫn không thể phủ nhận rằng nó vẫn còn nhiều hạn chế , đặc biệt là nhiều cấu hình được sử dụng cho phần mềm này, Spring Boot được tổ chức tốt và rất linh hoạt, để tích hợp nhiều Framework khác, tính module mạnh mẻ, có rất nhiều API lợi thế. Là một Framework Java mạnh mẽ và phổ biến nhất hiện nay dành cho doanh nghiệp. Nó giúp rút ngắn thời gian lập trình và test, giảm sự rườm rà trong code, giảm thiểu bottleneck để tập trung nâng cao business cho ứng dụng.

Nói về tình chịu tải của Java thì nếu ta lập trình bằng Php thì 1000 người cùng lúc truy cập vào nó sẽ chậm, có thể mất kết nối đến website, còn nếu sử dụng Java thì có xử lý 200 triệu user cùng lúc. Điển hình là các hệ thống dịch vụ tài chính trong ngân hàng, ứng dụng Android. Javacó rất nhiều tính năng bảo mật có sẵn cần thiết cho lập trình web như: Xác thực nâng cao, mã hóa và kiểm soát mức độ truy cập.

* Ứng dụng ReactJS và Java Spring Boot để xây dựng website bán hoa quả nhằm mục đích:

Việc sử dụng ReactJS cho website phía clients sẻ giúp cho tốc độ tải trang và xử lý dữ liệu sẽ tăng cao, và cho người dùng sử dụng cảm thấy thoải mái trong thao tác với trang web. Tốc độ trang web xử lý nhanh sẽ giúp cho SEO tăng lên.

Phía sever lưu trữ dữ liệu sử dụng Java Spring Boot giúp cho dữ liệu được bảo mật hơn và tốc độ hoàn thiện sẻ nhanh hơn.

## Mục tiêu của đề tài

Mục tiêu của đề tài nhằm xây dựng một trang web đáp ứng những nhu cầu của khách hàng về mua các mặt hàng về hoa quả một cách thuận lợi và nhanh chóng. Hệ thống giúp cho khách hàng có thể truy cập trang web trên nhiều thiết bị khác nhau mọi lúc mọi nơi khi đang có kết nối mạng.Thị trường kinh doanh hoa quả như một cuộc chạy đua âm thầm nhưng đầy cạnh tranh, bên cạnh đó các cửa hàng, đại lý hoa quả đã tồn tại khá lâu, nhiều cửa hàng cũ đã đầu tư và mở rộng mặt bằng kinh doanh, sửa sang lại quầy hàng, tăng cường nhiều loại hoa quả,.. để tăng tính cạnh tranh so với đối thủ. Nhu cầu tiêu thụ hoa quả sạch của khách hàng ngày càng rất lớn và nguồn gốc xuất xứ của sản phẩm. Do đó chúng ta cần phải thực hiện ngay một trang website có thể bán trái cây sạch online. Cùng sự phát triển của công nghệ internet, việc xây dựng một trang web để mua hoặc quản lý trái cây trực tuyến là vô cùng cần thiết và tiện lợi. Xây dựng được trang web đầy đủ các chức năng cần thiết phục vụ cho việc quản lý thực phẩm hoa quả.Chỉ cần có một chiếc máy tính hoặc điện thoại thông minh khách hàng có thể mua hàng hoặc quản lý  một cách nhanh chóng tiết kiệm thời gian cũng như chi phí đi lại đặc biệt là trong xã hội bận rộn thời tiết thất thường trong những ngày vừa qua.Thông qua đó, khách hàng có thể xem chi tiết các mặt hàng thông qua trang web của chúng ta. Từ đó họ sẽ cân nhắc chọn lựa các sản phẩm cần thiết cho mình. Ngoài ra trang website còn có các chức năng tìm kiếm và tra cứu để hỗ trợ cho bộ quản trị viên tìm kiếm thông tin một cách nhanh chóng.

Qua dịch bệnh covid-19 vừa rồi. Tất cả các hàng quán đề cấm tụ tập buôn bán và người dân còn hạn chế ra đường rất nhiều những nhân viên công sở đều được làm việc tại nhà, học sinh sinh viên đều được cho nghỉ, đa phần người dân đều ở nhà rất ít người đi ngoài khiến cho các hàng quán không thể kinh doanh được và đóng cửa. Nhưng trong số đó có một vài cửa hàng bán hàng online giúp cho việc kinh doanh buôn bán trong mùa dịch vẫn ổn định. Người dân không cần phải đi ra để mua thực phẩm chỉ cần ở nhà có một chiếc smartphone hoặc máy tính có kết nối internet có thể đặt đồ ăn tại nhà. Vì vậy đã cho ta thấy được việc có một website bán hàng online là một điều cần thiết trong lúc này.Để xây dựng một trang web bán hoa quả với các tiêu chuẩn như năng suất lập trình cao, đơn giản, bảo mật thì chúng em quyết định chọn công nghệ ReactJS và Java Spring Boot. ReactJS giúp cho tốc độ load trang và xử lý dữ liệu một cách nhanh chóng hơn làm cho người dùng cảm thấy thoải mái và thao tác một cách nhanh chóng không phải ngồi chờ đợi, việc quản lý dữ liệu chọn công nghệ MySql và phần Sever thì sử dụng Java Spring Boot, dữ liệu của khách hàng cần được bảo mật một cách an toàn cũng như các thông tin của cửa hàng, giúp cho khách hàng cảm thấy an toàn với thông tin của mình không bị đánh cắp khi truy cập vào trang web để mua hàng.

# Tổng quan REACTJS

## Giới thiệu

ReactJS ra là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. ReactJS nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác.

ReactJS có tốc độ rất ấn tượng và sự phổ biến được sử dụng rất nhiều vì tính linh hoạt và đơn giản của nó. Phù hợp với các dự án lớn và cần tính mở rộng cao

Hiện nay ReactJS có hai cách để khởi tạo một component. Có thể khởi tạo một component bằng phương thức class hoặc có thể khởi tạo bằng function hay còn gọi là React Hooks

React có thể kết hợp với các thư viện khác nhau bootstrap. qua NPM công cụ quản lý package của Nodejs, bao gồm một hệ thống quản lý module gọi là CommonJS cho phép cài đặt bất cứ công cụ command line nào được viết bằng Javascript. Hầu hết các thư viện và công cụ trong React đều được triển khai thành các module CommonJS và có thể cài đặt thông qua npm.

Cài đặt React cho máy bằng cách mở command line và chạy lệnh:

**npm install -g create-react-app**

 Tạo ứng dụng đầu tiên bằng cách chạy dòng lệnh:

**create-react-app < tên thư mục muốn tạo >**

ReactJS với các framework phổ biến khác:

- Hiệu suất giữa ReactJs và AngularJS. AngularJS sẻ bị chậm đáng kể khi có nhiều người truy cập cùng lúc bị giới hạn trọng phạm vi hoạt động nhất định. Bên cạnh đó digest cycle chạy nhiều lần trước khi ổn định nếu được kích hoạt một cập nhật nào đó.

- ReactJS và VueJS đều có thế mạnh chung về tốc độ. Tất cả mọi thứ trong ReactJS đều là Javascript. Mỗi cấu trúc HTML trong ReactJS sẻ được thể hiện qua JSX, và có thể đưa cả CSS vào trong JS. Tuy nhiên VueJS vẫn bám theo các công nghệ web truyền thống và phát triển dựa trên chúng.

## Lịch sử phát triển

Trước năm 2013: Facebook thiết kê việc xây dựng UI cho Facebook hoàn toàn dựa trên AngularJS, tuy nhiên lượng người dùng trở nên quá lớn và dữ liệu Facebook lưu trữ ngày càng nhiều, kéo theo việc xử lý ngoài UI trở nên quá khó khăn vì AngularJS chậm và nặng nề. Facebook không thể tìm kiếm được một framework có đủ khả năng thay thế cho AngularJS, vậy nên các lập trình viên của Facebook đã tự tạo ra một công nghệ mới với tên gọi React

Tháng 3 năm 2013: ReactJS ra đời gồm hai phần ReactJS (dung cho web app) và React Native (dùng cho mobile app)

Năm 2015: Facebook quyết đinh open source mã nguồn của React cho cộng đồng dev trên toàn thế giới với version 0.3.0.

Tháng 6 năm 2017: version 15.6.0

Tháng 9 năm 2018: React phát hành version 16.5.2

Ngày 6 tháng 2 năm 2019: version 16.8.0 React Hook phát hành ban đầu.

Ngày 26 tháng 2 năm 2020: version 16.13.0.

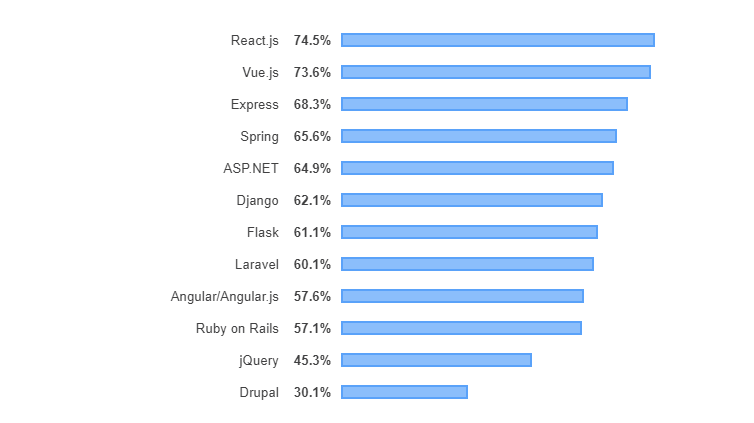
Ngày 19 tháng 3 năm 2020: version 16.13.1

## Tình hình ứng dụng ReactJS trên thế giới

Theo SamTech, React đang được sử dụng trên 205.00 tên miền duy nhất, tăng trưởng 2,46% mỗi tháng

Người ta ước tính có khoảng hơn 1.300 developer và hơn 94.000 website sử dụng ReactJS

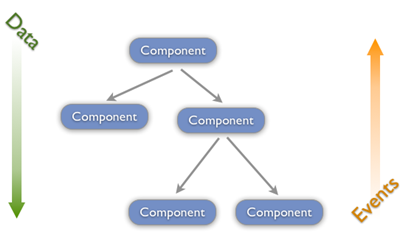
Những ông lớn thương mại điện tử đang sử dụng ReactJS cho trang web của minh như: chợ tốt, TIKI, LAZADA, UBER, Traveloka, Spotify, Paypal, Instagram, Dropbox, Udemy, Facebook, Walmart, Auth0, Discord … [3]



Hình 2‑. Web frameworkđược yêu thích nhất vào năm 2019 [2]

## Các chức năng chính của REACTJS

ReactJS là một Framework hiển thị view chú ý đến hiệu năng. Rất nhiều Framework khác mất một khoản thời gian để hiển thị những lượng data lớn. Nhưng React không còn là vấn đề, vì chỉ hiển thị những thay đổi ReactJS xây dựng lên các components có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Giúp cho việc quản lý dễ dàng mở rộng hệ thống. ReactJS tạo ra một virtual DOM, khi có bất kỳ sự thay đổi nào ReactJS chỉ cập nhật những components có liên quan điều này làm cho tốc độ của ReactJS cực nhanh. Ngoài ra ReactJS là một library cho phép nhúng code html trong code javascript nhờ vào JSX.ReactJS sử dụng cơ chế one-way binding (luồng dữ liệu một chiều) truyền dữ liệu giữa các component thông qua props

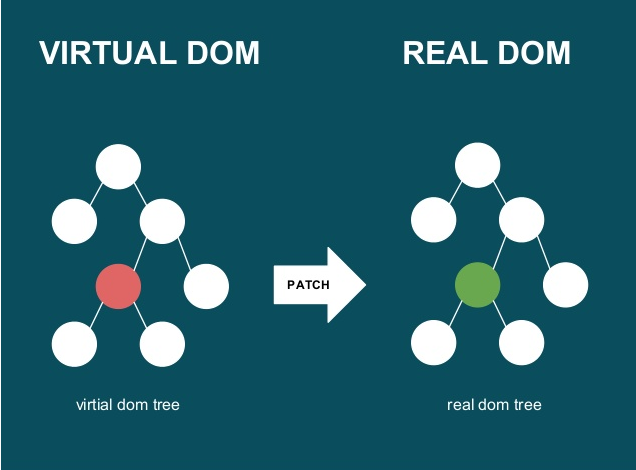


Hình 2‑2. Mô tả luồng dữ liệu một chiều trong ReactJS

## Virtual DOM

Virtual DOM một trong những điểm mạnh của ReactJS nằm ẩn trong mỗi view đây là lý do khiến cho React có hiệu năng tốt, Khi có một view yêu cầu thay đổi sẽ được đưa vào một bản sao ảo của DOM. Sau khi yêu cầu được hoàn thành, tiến hành một phép so sánh giữa DOM ảo và DOM thất và thực hiện những thay đổi chỉ ra trong phép so sánh trên.

Ví dụ: nếu chúng ta đang xem một danh sách có 500 data được hiển thị bởi React và thay đổi data thứ 100, thì chỉ data thứ 100 được hiển thị lại và 499 data còn lại được giữ nguyên (không cần hiển thị lại hoặc reload lại trang)



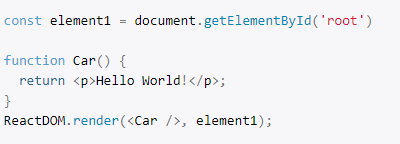
Hình 2‑3. Thể hiện sự thay đổi dữ liệu giữa dom ảo và dom thật

## Component

React xây quanh các component, chứ không dùng teamplace như các framework khác. Ta có thể tạo ra một component bằng cách extend component như sau.



Hình 2‑4. Sử dụng Class để tạo một component



Hình 2‑5. Sử dụng function để tạo một component

Trong component ta còn khái niệm gọi là Multiple Component. Cho phép ta có thể lòng nhiều component vào nhau, đặt các tên gọi component trong lệnh return của phương thức render

Ví dụ: ta có hai component Welcome và Date. Ta đặt tên component Date vào trong lệnh return của component Welcome. Khi gọi component Welcome để render ra view sẻ gọi hiện việc render luôn component Date



Hình 2‑6. Mô tả lòng các component với nhau

## Render

Phương thức quan trọng nhất là render. Hàm render chính là mô tả cụ thể của một view tại bất kì thời điểm nào. Vì thế nếu dữ liệu thay đổi React sẽ thực hiện việc update view với dữ liệu tương ứng. Có thể hiểu là khi thay đổi dữ liệu React sẻ tự động gọi hàm render để update lại view.

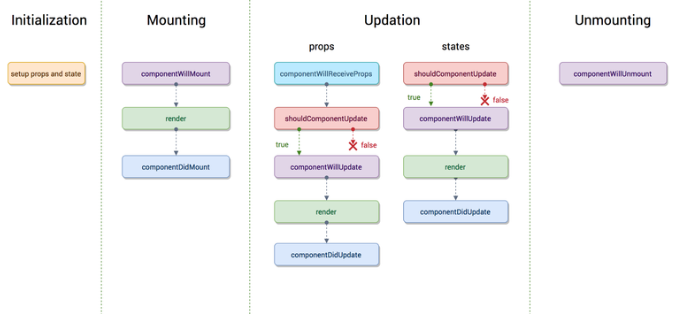
Như trong ví dụ ở hình 2.4 muốn render 1 thẻ p với lời chào là Hello Word

## JSX

JSX (Javascript Syntax Extension) có cú pháp gần giống với HTML. Cho phép chúng ta viết các mã html trong javascript. Các mã này trong thời gian thực sẻ nhanh hơn nhiều với một mã tương đương viết bằng javascript

## Lifecycle

Lifecycle thể hiện vòng đời của một component trong ReactJS.



Hình 2‑7. Vòng đời của một component

Qua hình 2.8 ta thấy được vòng đời của một component được thể hiện qua ba thành phần chính Mounting, Updation, Unmounting

Mounting khi truy cập hoặc refrech lại trang web thì sẽ được hay lần đầu tiên. Trong mounting có ba phương thức sẻ lần lượt chạy:

+ componentWillmount sẻ được gọi trước khi render

+ componentDidMount sẻ được gọi sau khi render component hoàn tất

Updation khi thay đổi state hoặc truyền một props gồm các phương thức

+ componentWillReceiveProps: chạy khi component con nhận props từ component cha. Sau khi component con nhận được dữ liệu từ props mới thì có thể set lại state.

+ shouldComponentUpdate: sẻ trả về kết quả true hoặc false, giá trị mặc định là true. Nếu không muốn component render lại sau khi update state hoặc props thì ta return giá trị về false.

+ componentWillUpdate: được gọi khi chúng ta update state của component trước khi render lại.

+ componentDidUpdate: sau khi componentWillUpdate được gọi thì sẽ đến componentDidUpdate được gọi.

Unmount: quá trình này xảy ra khi component bị xóa khỏi dom

+ componentWillUnmount: sau khi xóa một component ra dom thì quá trình này sẻ được gọi.

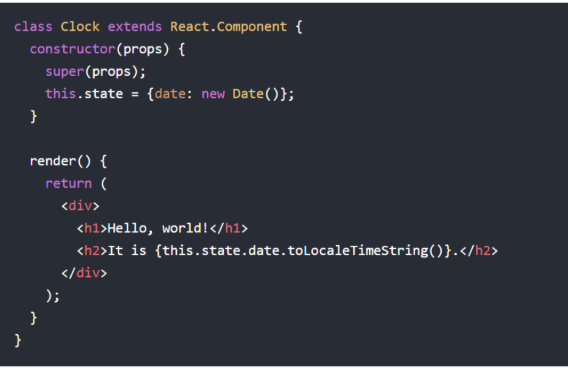
## Prop and State

Props chính là properties của một component, chúng ta có thể thay đổi props của component bằng cách truyền dữ liệu từ bên ngoài vào. Props có thể là một object, function, string, number …

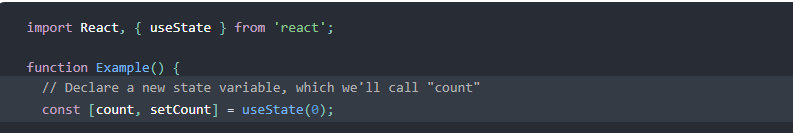


Hình 2‑8. Phương thức Props trong component

State: một component trong React có hai cách để lấy thông tin props và state. Không như props có thể truyền từ bên ngoài vào. Mỏi một component sẻ tự quyết định state của chính nó. State chỉ có giá trị trong component đó. Để tạo ra state cho một component, ta sẽ khai báo nó trong hàm constructor đối với component được khởi tạo bằng class còn với function component ta sẽ gọi hàm useState.



Hình 2‑9. Phương thức state trong component khởi tạo bằng class



Hình 2‑10. Phương thức state trong component khởi tạo bằng function

Để thay đổi giá trị state trong component đối với component khởi tạo bằng class trong constructor ta sẻ gọi hàm this.setState (< biến và giá trị cần thay đổi >). Còn đối với component được khởi tạo bằng function ta sẻ gọi set<tên biến> lúc ta khởi tạo state ví dụ như hình 2.10 chúng ta muốn thay đổi giá trị của count bằng cách ta sẻ gọi setCount (< truyền giá trị cần thay đổi >)

Props và state có sự giống nhau:

+ Props và state đều là plain JS objects.

+ Props và state đều render update khi thay đổi.

## React Router

ReactJS không có hỗ trợ Router vậy nên chúng ta cần đến một thư viện để thực hiện công việc này. React Router là một thư viện routing tiêu chuẩn trong React, sử dụng cấu trúc thành phần để gọi các thành phần hiển thị thông tin phù hợp. Phiên bản mới nhất hiện tại là V4.

Để cài đặt thư viện Reat Router, sử dụng yarn hoặc npm

**yarn add react-router-dom**

**npm install react-router-dom**

Sau đó có thể import thư viện để sử dụng trong các component

**import {BrowserRouter as Router, Route, Link} from ‘react-router-dom’**

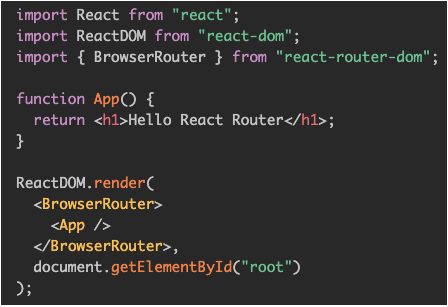
### BrowserRouter và HashRouter

React-Router cung cấp hai thành phần hay sử dụng đó là BrowserRouter và HashRouter. Hai thành phần này khác nhau ở kiểu URL mà chúng ta sẽ tạo ra và đồng bộ.

BrowserRouter: được sử dụng phổ biến hơn, nó sử dụng History API có trong HTML5 để theo dõi lịch sử Routing.

HashRouter: sử dụng hash của url để ghi nhớ mọi thứ.

Để sử dụng BrowserRouter chỉ cần đảm bảo rằng nó được hiển thị ở phần gốc của hệ thống. Thông thường sẻ nằm ở file index.js, sẻ bọc <App /> phần tử cao nhất của mình trong BrowserRouter như sau;



Hình 2‑9. Đặt thẻ BrowserRouter để sử dụng routing cho trang web

### Route Matchers

Có hai thành phần trong router matching: Switch và Route. Khi một Switch được hiển thị, nó tìm kiếm thông qua các children Router để tìm một phần tử path khớp với với URL hiện tại. Khi tìm thấy sẻ hiện ra Route và bỏ qua tất cả những cái khác. Nếu không có Route khớp phù hợp, Switch sẻ không xuất ra gì (null).



Hình 2‑10. Sử dụng Switch và Router để tạo URL cho trang web

Route là một ánh xạ giữa URL và một Component. Người dung truy cập theo một URL trên trình duyệt, một component tương ứng sẽ được render trên giao diện. Trong router exact giúp cho router chỉ hoạt động khi URL trên trình duyệt phù hợp tuyệt đối với giá trị của thuộc tính path.

### Link

Trong HTML có cặp thẻ chuyển hướng đó là thẻ <a></a> nhưng trong React để chuyển hướng chúng ta sử dụng cặp thẻ <Link></Link> được import từ React-router

**<Link to=”/Home”>Home</Link>**

to: giống thuộc tính href trong thẻ a

### NavLink

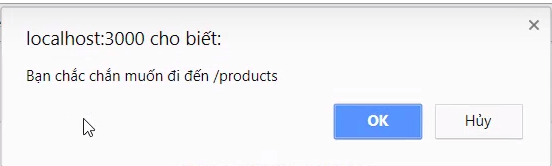
Về cách sử dụng thì NavLink giống với Link, nhưng NavLink có hỗ trợ thêm một số thuộc tính như là activeClassName và activeStyle hai thuộc tính này giúp cho khi trùng khớp thì sẽ được active lên và có thể style cho nó.



Hình 2‑11. Sử dụng NavLink thay đổi màu khi active.

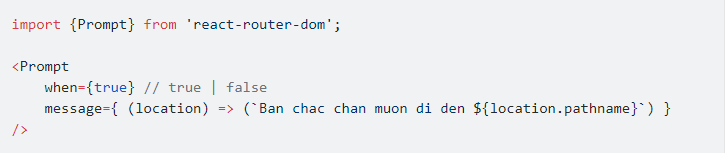
### **Prompt**

Xác nhận khi chuyển trang. Khi chúng ta đang nhập liệu ở form nào đó mà không may click nút back hay chuyển trang thì dữ liệu sẻ bị mất để khắc phục điều đó ta sử dụng đối tượng prompt sẻ giúp chúng ta back hay chuyển trang sẽ xác nhận xem là có chắc chắn muốn back hay chuyển trang không.



Hình 2‑12. Xác nhận trước khi chuyển trang

Để sử dụng đối tượng prompt thì import tử react-router.



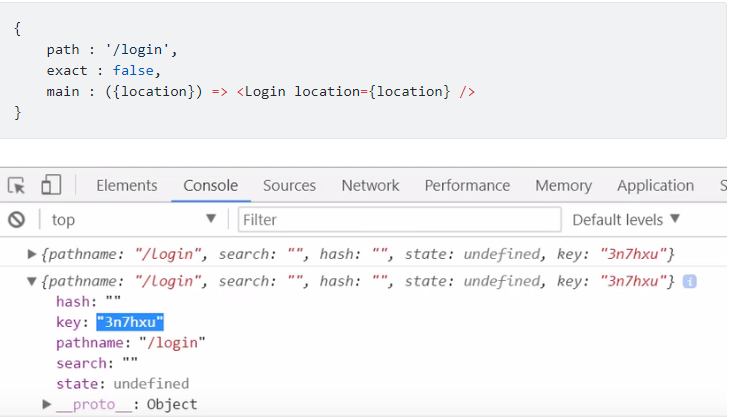
Hình 2‑13. Import prompt vào component

### **Redirect**

Dùng để chuyển trang trước đó. Có thể truy xuất thông tin trang trước đó thông qua đối tượng location, để sử dụng redirect ta chỉ cần import từ react-router

**Import { Redirect } from ‘react-router-dom’**

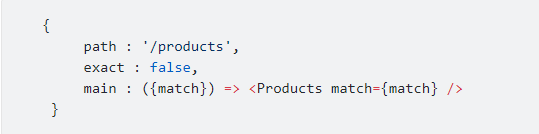
Khi muốn sử dụng location tại cấu hình router ta chỉ cần truyền đối tượng location vào component mà cần sử dụng đối location



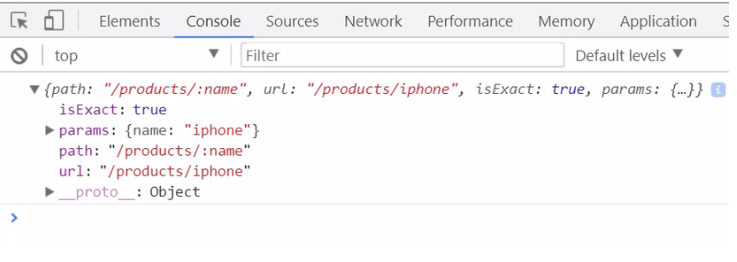
Hình 2‑14. Truyền đối tượng location vào router

### **Match**

Khi muốn lấy thông tin trên url có thể dung đối tượng match để lấy dữ liệu vể. Tại cấu hình router ta chỉ cần thêm đối tượng match vào component cần sử dụng.



Hình 2‑15. Truyền đối tượng match vào router

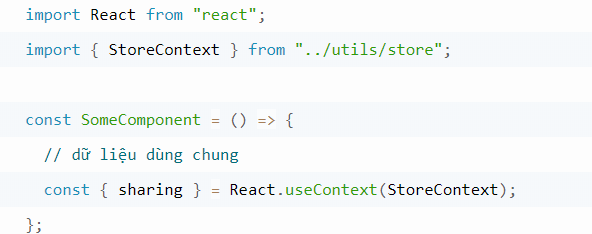


Hình 2‑16. Kết quả khi log đối tượng match

## Global state

Trong một dự án có nhiều component con long nhau việt truyền và cập nhật dữ liệu qua các component trở nên khó khăn hơn. Để khắc phục dung một kho chứa dữ liệu state như Redux, các component con bên trong có thể truy xuất và cập nhật dữ liệu một cách dễ dàng. Với version hiện tại React bổ sung các API mới là uesState, createContext, useContext và thư viện RecoilJS chúng ta có thêm lựa chọn làm store mà không cần dùng đến Redux.

useContext: với bất kỳ component dùng để lấy bất cứ dữ liệu nào trong store



Hình 2‑17: component sử dụng useContext để lấy dữ liệu trong store

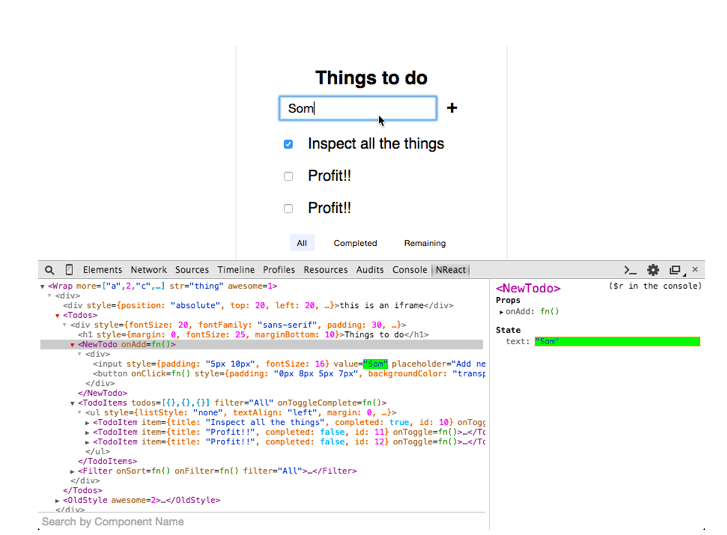
useReducer có hao tham số state, action và trả về new State sau khi thực hiện một action, useReducer function nhận vào reducer và initialState khởi tạo ban đầu.

Để đáp ứng việt quản lý state với nhiều tính năng hơn RecoilJS xuất hiện gần đây, RecoilJS có một số ưu điểm so với Redux như công cụ quản lý state đơn giản, cú pháp giống như state đơn thuần của react. Hỗ trợ quản lý state bất đồng bộ, việc sử dụng bất đồng bộ trên redux phải kết hợp với các thư viện khác như redux-thunk hoặc redux-saga

## Công cụ cho Dev

Ứng dụng mở rộng React Devtools trên chrome và firefox cho phép kiểm tra giá trị props và tây hiện tại của một cây react component.

Sau khi cài đặt React Devtools cho trình duyệt web, nhấp chuột phải vào bất kỳ phần tử nào trên web, chọn kiểm tra phần tử (Inspect) để mở công cụ cho nhà phát triển



Hình 2‑17. Kiểm tra giá trị một props và state của một component

# Tổng quan về Spring Boot

## Giới thiệu

* Spring là một mã nguồn mở,một giải pháp gọn nhẹ cho Java Enterprise, được phát triển, chia sẻ và có cộng đồng người dùng rất lớn,Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: Dependency Injection và Aspect Oriented Programming.Spring được coi là khung framework phổ biến nhất được sử dụng bởi các nhà phát triển Java

Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object).

Spring Boot là một phần mở rộng của Spring Framework của nền tảng Java được phát triển bởi Pivotal Team.

Spring Boot giúp chúng ta dễ dàng tạo các ứng dụng dựa trên Spring độc lập mà không cần cấu hình quá nhiều như Spring MVC.Có thể sử dụng Spring Boot để tạo các ứng dụng bằng cách sử dụng java –jar hoặc thực hiện thao tác thủ công.

Để phát triển 1 ứng dụng cơ bản của Spring Framework thì ta cần phải :

* Tạo project sử dụng Maven và các dependency cần thiết cho Spring MVC và Servlet API.
* Tạo 1 tập tin web.xml để khai báo DispatcherServlet của Spring MVC.
* Một tập tin cấu hình của Spring MVC.
* Một class Controller trả về một trang khi có request đến.
* Cuối cùng là phải có một web server dùng để triển khai ứng dụng lên chạy

Còn khi sử dụng Spring Boot thì ta có thể dùng Sublime Text để phát triển mà không nhất thiết phải sử dụng eclipse , netbean , cấu hình project tự động , thêm các dependency đơn giản , nhanh chóng , không yêu cầu file xml config , có sẵn nhiều plugin.

## Lịch sử phát triển

Vào tháng 10 năm 2012, Mike Youngstrom đã tạo một  [feature request in spring jira](https://jira.spring.io/browse/SPR-9888) yêu cầu hỗ trợ cho các kiến ​​trúc ứng dụng web không chứa trong Spring Framework. Yêu cầu đó được bắt đầu vào đầu năm 2013 và tháng 4 năm 2014 đội ngũ phát triển Spring đã cho ra đời Spring Boot 1.0.0 sau gần 1 năm bắt đầu dự án này .

Spring boot 1.1 : 6/2014

Spring Boot 1.2 : Tháng 3 năm 2015

Spring Boot 1.3 : Tháng 12 năm 2016

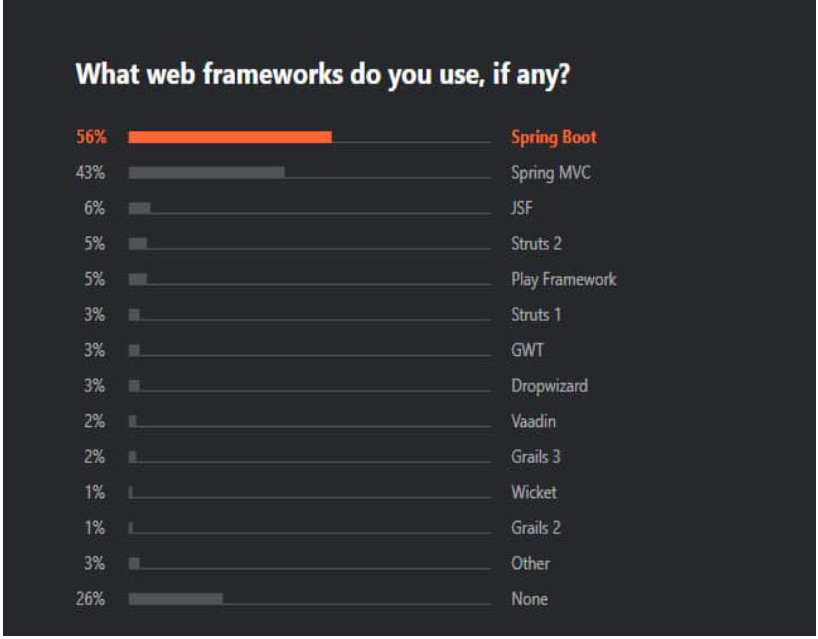
Spring Boot 1.4 : Tháng 1 năm 2017

Spring Boot 1.5 : Tháng 2 2017

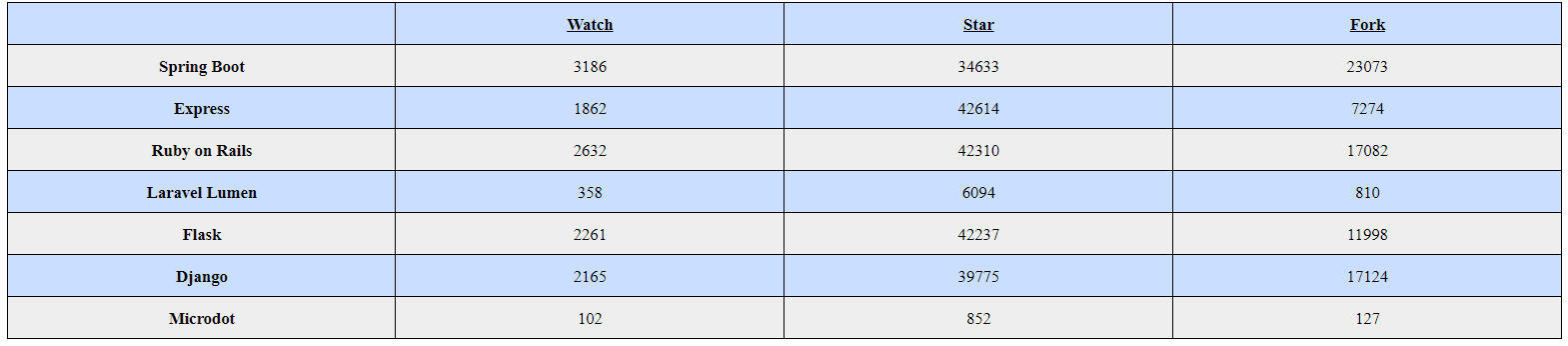
Vào tháng 3 năm 2017, nền tảng Spring io mới nhất (Brussels-SR1) đã được phát hành. Nó sử dụng bản phát hành Spring Boot mới nhất (1.5.2.RELEASE). Nền tảng Spring io tiếp theo là Cairo dự kiến ​​phát hành với spring boot 2.0 và spring framework 5.0

## Tình hình ứng dụng Spring Boot trên thế giới

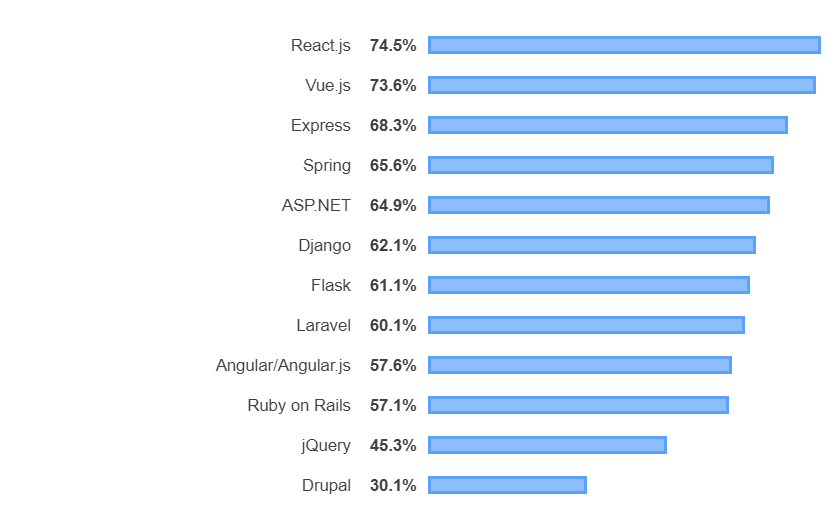
Hiện tại Spring Framework là top đầu trong các framework được ưa chuộng và sử dụng nhiều nhất trên thế giới (2020).[5,6,7]



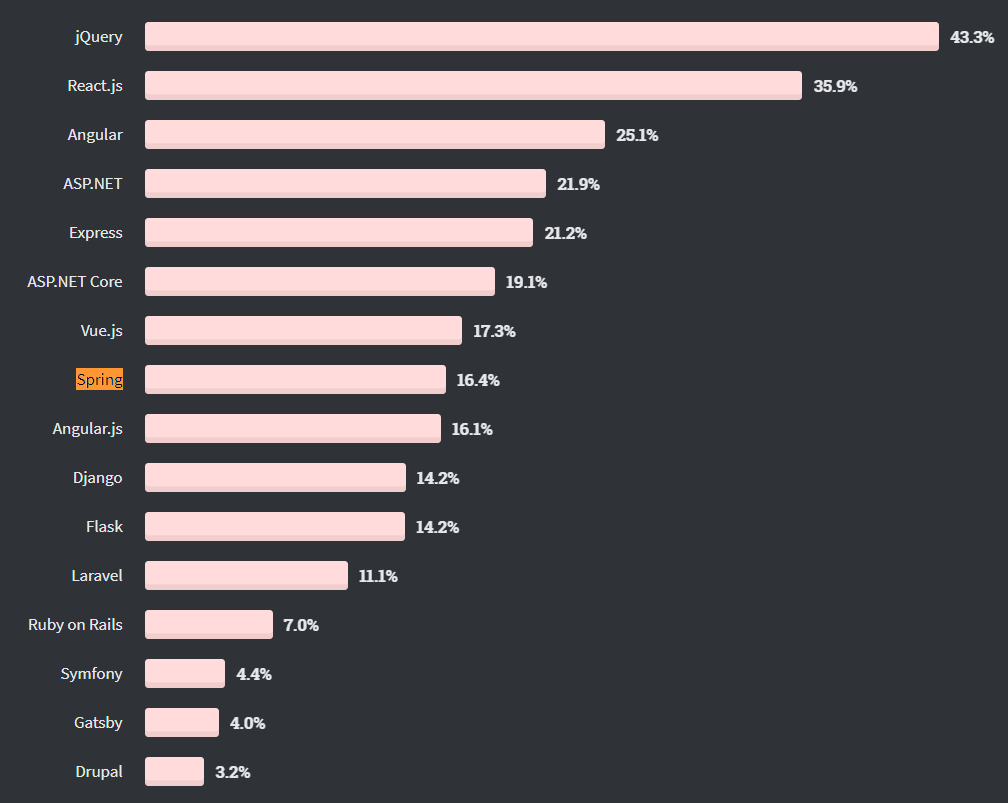
Hình 3.1 :Theo khảo sát đầu năm 2019 của JetBains về tỷ lệ sử dụng Spring Framework là rất cao(Cao hơn năm ngoái 14%)[8]



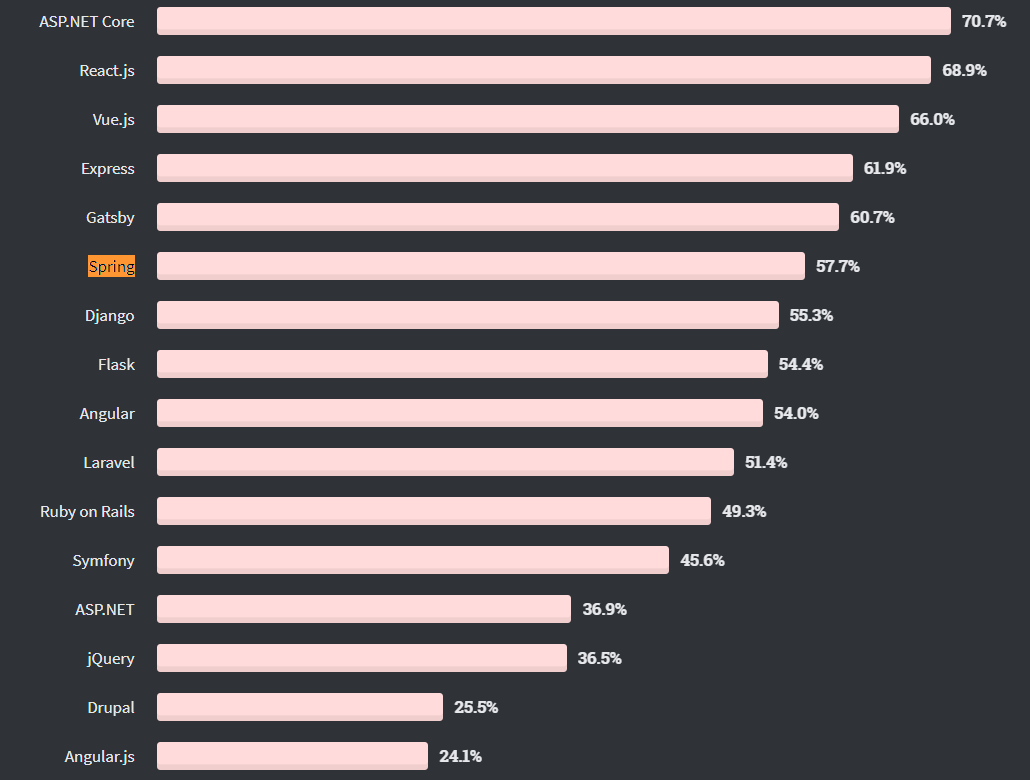
Hình 3.2: Spring Boot lọt top Trending Backend Framework năm 2019 theo Kellton Tech[9]



Hình 3.3 :Thống kê về những Web Framework được yêu thích nhất trong năm 2019 theo Stack OverFlow[10]



Hình 3.4 Top Web Framework được chọn theo Stack OverFlow 2020[11]



Hình 3.5: Các Web Framework được yêu thích theo Stack OverFlow 2020 [11]

## Các chức năng chính của Spring Boot

Dưới đây là 1 số chức năng nổi bật của Spring Boot:

* Đóng gói ứng dụng Spring dưới dạng một file JAR (stand-alone application). Có thể dễ dàng bắt đầu 1 ứng dụng Spring chỉ với câu lệnh quen thuộc java –jar(Write one Run Everywhere). Khi ta đóng file project thành file jar thì ta có thể đem đi tới bất cứ máy nào để sử dụng mà không cần phải gửi cả project.
* Tối ưu công đoạn cấu hình cho ứng dụng Spring, không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML(Spring Boot Starter , DI,Spring Boot Application).
* Cung cấp một loạt các tính năng phi chức năng phổ biến cho các dự án lớn như nhúng trực tiếp web server như Tomcat, Jetty, ... vào ứng dụng, bảo mật, health check...
* Cho phép dự án tích hợp thêm các module

Hiện tại, Spring Boot có 2 version chính là:

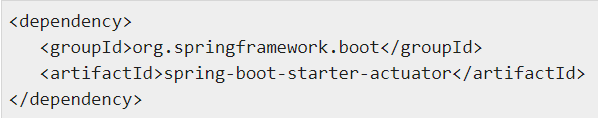
* v1.5.x
* v2.x

### Spring Boot Starter

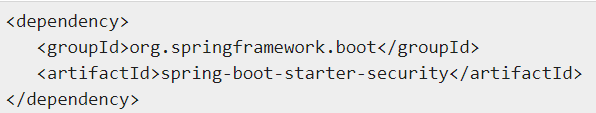
Việc khởi động một Project trong các framework khác thường yêu cầu cài đặt các thư viện , …. Nhưng Spring Boot Starter cung cấp cho ta chức năng để có thể lựa chọn dễ dàng . Ví dụ ta muốn khởi tạo project spring boot và sử dụng JPA để truy cập csdl , thì ta chỉ cần cài đặt dependence **spring-boot-starter-data-jpa** .

Ta có thêm vài ví dụ về spring-boot-start-\* dưới đây

-\* dấu \* trong chỉ ra cái loại ứng dụng mà ta sử dụng

**Ex 1: Spring Boot Starter Actuator**  sử dụng Actuator để giám sát và quản lí ứng dụng . 

**Ex 2: Spring Boot Security** trong dependency



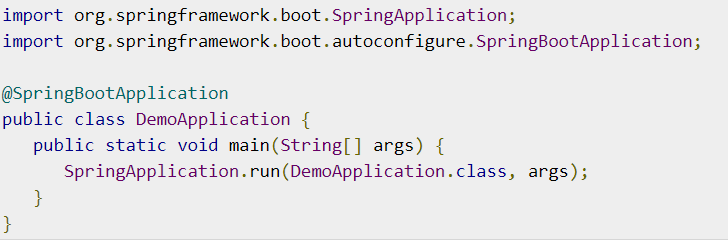
Ta có thể sử dụng các Module khác như Thymeleaf, Spring Web, Test bằng cách như trên

### **Spring Boot Application**

Khi bắt đầu project Spring Boot ta để Annotation **@SpringBootApplication** ở trên .  **@SpringBootApplication** bao gồm cấu hình tự động , quét project và khởi động.

Khi ta sử dụng nó thì ko cần phải thêm cái Annotation khác như **@EnableAutoConFiguration ,@ComponentScan** và **@SpringBootConFiguration.**

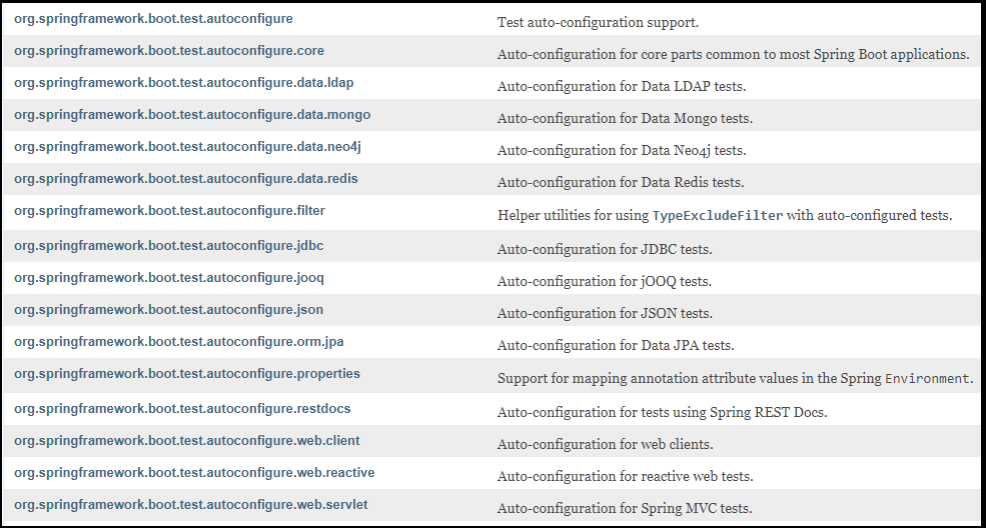
Xem qua hình mình họa để hiểu rõ hơn về **@SpringBootApplication:**



### **AutoConfiguaration**

Sử dụng Annotation @EnableAutoConfiguration để cấu hình tự động , tự động tìm thư viện đang thiếu để bổ sung

Sping Boot AutoConFiguation triển khai cấu hình theo spring-boot-auto-configuration.jar .Ta có thể xem list of package ở dưới đây :

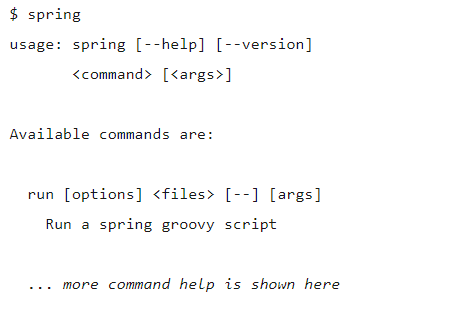


### **Spring Boot Command Line Interface**

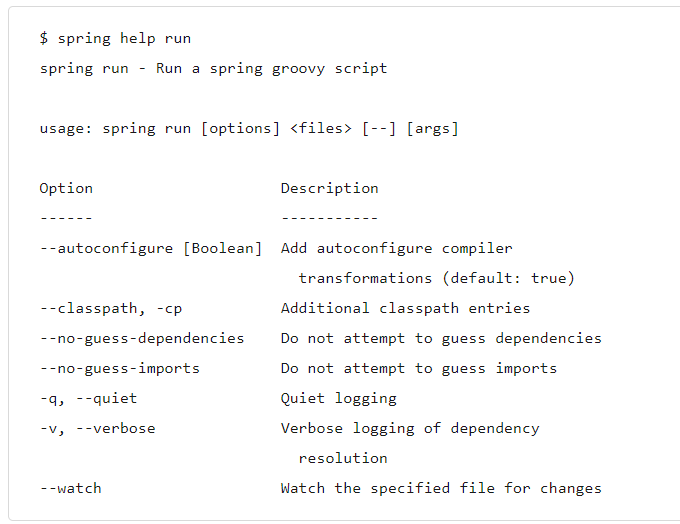
Spring Boot CLI hỗ trợ sử dụng cmd để viết ứng dụng SpringBoot .

Ta có thể cài đặt Spring Boot Cli thủ công bằng SDKMAN hoặc Homebrew,MacPorts nếu sử dụng hệ điều hành Macos. Truy cập [*get-started.html*](https://docs.spring.io/spring-boot/docs/current/reference/html/getting-started.html#getting-started-installing-the-cli) để xem hướng dẫn cài đặt.

Khi đã cài đặt Cli , ta có thể sử dụng cmd mà nhập spring và enter thì nó sẽ hiện thì màn hình trợ giúp đơn giản :



Nếu sử dụng spring help thì sẽ được thêm nhiều thông tin hỗ trợ hơn trừ Spring Boot CLI:



### Có thể liên kết với nhiều module khác

Khi sử dụng Spring Boot , đồng thời ta có thể sử dụng chung nhiều framework khác như Spring Security để gia tăng tính bảo mật của sản phẩm

Spring Security : là 1 framework  giúp ta tích hợp tính năng xác thực và phân quyền một cách dễ dàng vào ứng dụng, Chống lại CSRF attack Bảo vệ Session Fixation Mã hóa mật khẩu. Cache control X-XSS-Protection

Chế độ tự động kiểm tra ByteCode, và có các chức năng bảo mật như Servlet filter, URL endcoding, Secure Web Services

Không có con trỏ tường minh.

Chương trình chạy bên trong các hộp thiết bị ảo.

Thêm sự bảo vệ bằng việc phân biệt riêng rẽ các package cho các lớp của hệ thống local file mà từ đó chúng được import với các file từ nguồn mạng.

Khả năng tìm ra các đoạn code không hợp lệ mà có thể truy cập trái phép tới các đối tượng

Nó được bảo mật theo mặc định với xác thực cơ bản trên tất cả các endpoint HTTP. Một bộ endpoint phong phú có sẵn để phát triển ứng dụng Spring Boot an toàn.

### DI(Dependency Injection)

Là 1 trong những tính năng nổi bậc của Spring Boot cho phép chúng ta thay đổi version của Java bằng cách sử dụng **<java.version>**

<properties>

<java.version>1.8</java.version>

</properties>

Thêm Maven Plugin chỉ cần thêm vài dòng ở file Pom là có thể thêm vào

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

Nếu chúng ta không sử dụng **spring-boot starter-parent** dependency nhưng vẫn muốn sử dụng lợi thế của DI , ta có thể sử dụng tag <scope> ví dụ như :

<dependencyManagement>

<dependencies>

<dependency><!-- Import dependency management from Spring Boot -->

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-dependencies</artifactId>

<version>2.2.2.RELEASE</version>

<type>pom</type>

<scope>**import**</scope>

</dependency>

</dependencies>

</dependencyManagement>

Cách sử dụng DI vừa liệt kê trên không cho phép ghi đè . Để overriding thì ta cần sử dụng thẻ **<dependencyManagement> Ví dụ :**

**<dependencyManagement>**

<dependencies>

<!--Override Spring Data release train-->

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring-data-releasetrain</artifactId>

<version>Fowler-SR2</version>

<type>pom</type>

<scope>**import**</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-dependencies</artifactId>

<version>2.2.2.RELEASE</version>

<type>pom</type>

<scope>**import**</scope>

</dependency>

</dependencies>

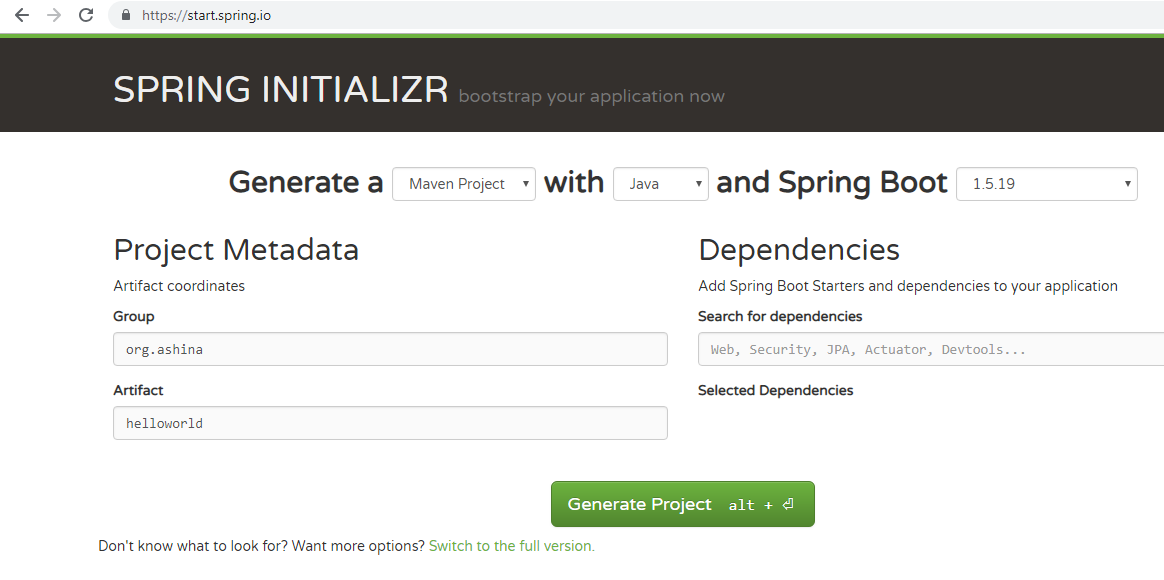
**</dependencyManagement>**

### Tạo một Ứng dụng bằng Spring Boot

Cách đơn giản nhất để khởi tạo một ứng dụng Spring Boot, đó là chúng ta sẽ truy cập vào trang web [Spring Initializr](https://start.spring.io/):

* Chọn build tool: Maven hay Gradle
* Chọn ngôn ngữ lập trình: Java, Kotlin hay Groovy
* Chọn version của Spring Boot
* Nhập tên của Group và Artifact
* Lựa chọn các dependency cần cho ứng dụng (ví dụ như Web, Security, JPA, Actuator, Devtools, ...)

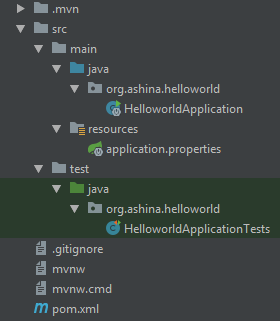
Click button **Generate Project** để download project xuống.

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/3v90pjljvq_201901270916.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/3v90pjljvq_201901270916.png)

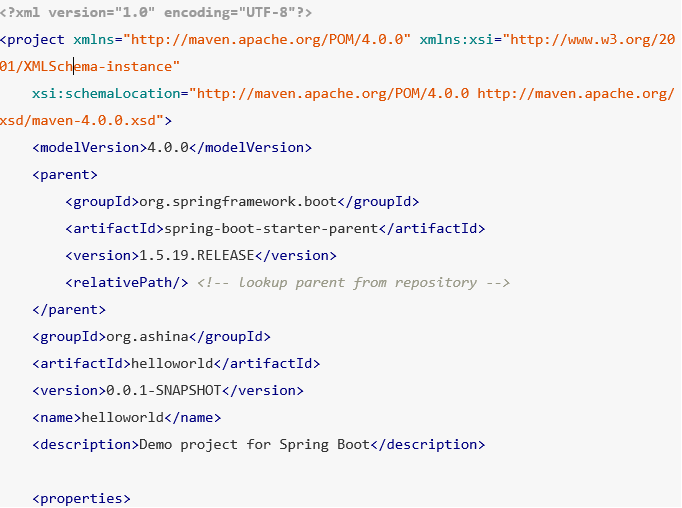
Ngoài ra, chúng ta cũng có thể dùng các [tool khác](https://spring.io/tools) để khởi tạo ứng dụng Spring Boot như Spring Tool Suite.

### Cấu trúc thư mục

Dưới đây là cây thư mục của ứng dụng

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/lbjw8pppiw_201901270924.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/lbjw8pppiw_201901270924.png)

Còn đây là file pom.xml, đơn giản hơn rất nhiều so với cách khởi tạo thông thường:





Lúc này toàn bộ cấu hình của dự án sẽ gói gọn trong 1 file HelloworldApplication.java:



Annotation @SpringBootApplication là composed annotation của 3 annotation @EnableAutoConfiguration, @ComponentScan, @Configuration:

* @Configuration: đánh dấu lớp HelloworldApplication là lớp chứa các định nghĩa bean của ApplicationContext.
* @EnableAutoConfiguration: nhắc Spring Boot bắt đầu thêm các bean dựa trên các classpath setting, các property setting và các bean khác. Đối với ứng dụng Spring MVC, chúng ta thường phải thêm @EnableWebMvc, tuy nhiên Spring Boot sẽ tự động thêm annotation này.
* @ComponentScan: nhắc Spring tìm kiếm các component và configuration bên trong package org.ashina.helloworld.

### Build ứng dụng

Trong thư mục của source code, chúng ta chạy lệnh sau để build ứng dụng:

$ mvn clean install

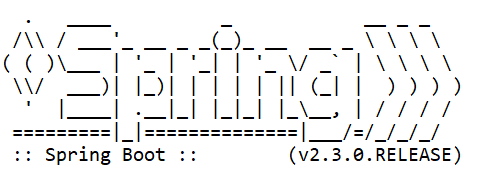
Sau khi build thành công, chúng ta thử vào thư mục target của source code, sẽ thấy file helloworld-0.0.1-SNAPSHOT.jar được sinh ra.

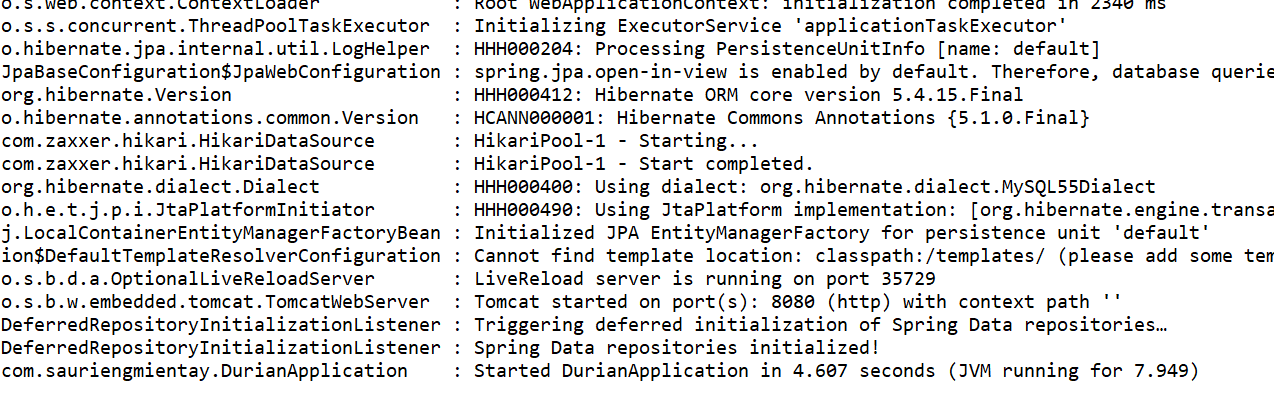
### Chạy ứng dụng

Trong thư mục target, chạy file helloworld-0.0.1-SNAPSHOT.jar bằng lệnh:



Khi chạy project thì sẽ hiện ra bảng Spring và hiện thị version của Spring Boot đang sử dụng.





Màn hình console sẽ hiện thị cho chúng ta biết các thông tin như Tomcat ở cổng 8080 , tình hình của template , JPA Data ,version của hibernate , LiveReload server ở cổng 35729 và nhiều thông tin khác.

## System Requirements

Spring Boot 2.3 yêu cầu java 8 và tương thích với java 14 . Spring Framework 5.2.6 hoặc cao hơn

Khi cài đặt ta phải đảm bảo cái Environment được cài đặt đúng chỗ.

Cài đặt JDK ví dụ :

PATH=%PATH%;C:\Program Files\Java\jdk1.7.0\_51\bin

JAVA\_HOME=C:\Program Files\Java\jdk1.7.0\_51

Hiện tại có nhiều phần mềm để viết java :

* **Eclipse**
* **IntelliJ**
* **NetBeans**
* **JDeveloper**
* **MyEclipse**
* **BlueJ**
* **Visual Studio Code**

|  |  |
| --- | --- |
| **Build Tool** | **Version** |
| Maven | 3.3+ |
| Gradle | 6 (6.3 or later). 5.6.x |

### Servlet Containers

|  |  |
| --- | --- |
| **Name** | **Servlet Version** |
| Tomcat 9.0 | 4.0 |
| Jetty 9.4 | 3.1 |
| Undertow 2.0 | 4.0 |

# Ứng Dụng Website Bán Hoa Quả

## Giới thiệu

Hệ thống dễ sử dụng, bảo mật và độ tin cậy. Kết nối an toàn với các hệ thống khác trong đó trách nhiệm của hệ thống thanh toán an toàn cũng cần thiết.

Hệ thống có giao diện đẹp, chạy nhanh và chạy được trên nhiều thiết bị như máy vi tính, điện thoại thông minh, máy tính bảng.

## Yêu cầu khách hàng

Khách hàng có thể nhấn vào trang chủ để xem chi tiết thông tin về trang web và các mặt hàng có ở đó. Tìm hiểu về thông tin các mặt hàng.

Liệt kê các loại vật phẩm của các khách hàng khi mua vào trong danh mục giỏ hàng.

Truy cập vào trang web để theo dõi bảng giá, có thể xem được giá cả chính thức các chương trình khuyến mãi ưu đãi.

Trang liên hệ sẽ hiển thị thông tin cách thức liên hệ với văn phòng phẩm qua số điện thoại cũng như địa chỉ cửa hàng. Khách hàng gồm 2 loại là thành viên và khách vãng lai.

-Khách vãng lai: Có thể xem tất cả các thông tin sản phẩm của website và đăng kí làm thành viên. Nhưng chưa thể mua hàng vì chưa đăng ký thành viên

-Thành viên: Có thể tất cả quyền của khách vãng lai, có thể đặt, mua hàng và quản lý thông tin cá nhân, xem lịch sử các đơn hàng.

## Quy trình đặt hàng của khách hàng

- Truy cập vào trang chủ

∙Khách vãng lai: phải đăng kí tài khoản. Sau đó đăng nhập vào tài khoản mới đăng kí và điền các thông tin của người nhận hàng vào,

cũng như địa chỉ nhận hàng vào form có sẵn (bao gồm số điện thoại, email, địa chỉ, tên khách hàng, nội dung nhắn nhủ, …).

∙Thành viên: giống như khách vãng lai. Nhưng họ đã có tài khoản rồi nên không cần đăng kí nữa.

-Nhấn nút thanh toán hệ thống sẽ nhận được các thông tin và tiến hành lưu đơn hàng của bạn và trả về kết quả.

Trường hợp khách hàng muốn hủy đơn hàng thì chỉ được hủy trước khi người quản trị xử lí đơn hàng.

Còn sau khi xử lý đơn hàng mà muốn hủy thì khách hàng cần liên hệ số điện thoại trực tiếp của văn phòng phẩm và đọc mã đơn hàng.

## Quy trình bán hàng

- Khi quản trị viên truy cập vào trang quản lý đơn hàng để xác nhận đơn hàng. Trang web sẽ hiện ra toàn bộ danh sách các đơn hàng đã và chưa xác nhận.

Quản trị viên sẽ xác nhận đơn hàng nếu trong kho đủ điều kiện để đáp ứng đơn hàng. Khi quản trị viên bấm xác nhận đơn hàng sẽ thông báo về email khách hàng với thông tin đơn hàng và trạng thái giao dịch.

Khi cửa hàng không đủ điều kiện để đáp ứng đơn hàng thì quản trị viên bấm hủy đơn hàng và gọi đến khách hàng để thông báo nguyên nhân hủy đơn hàng.

## Yêu cầu quản trị

- Quản trị viên sẻ đăng nhập vào trang quản trị với tư cách là admin

- Có thể quản lý các nội dung của trang web về thông tin khuyến mãi mặt hàng bán chạy, sản phẩm mới

- Quản lý loại sản phẩm và sản phẩm của cửa hàng

- Quản lý thông tin và xử lý đơn hàng của khách hàng

- Quản lý thông tin người dùng

- Quản lý đơn nhập hàng của cửa hàng

## Quy trình giao hàng

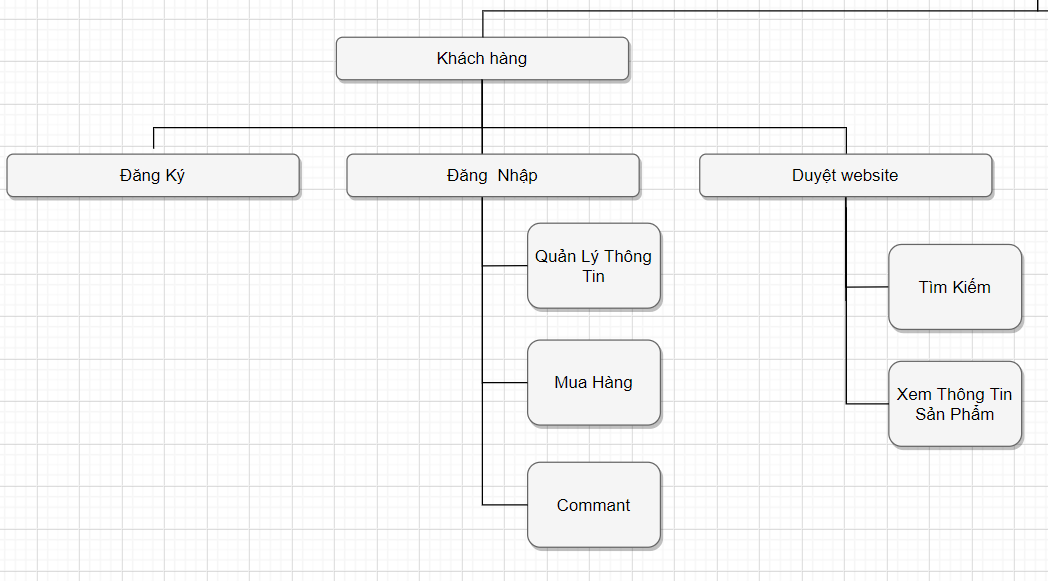
-Khi người dùng gửi đơn hàng, mặc định địa chỉ giao hàng sẽ là địa chỉ mà khi đăng kí người dùng sử dụng hoặc có thể tự đến cửa hàng để nhận hàng.

## Quy trình trả hàng

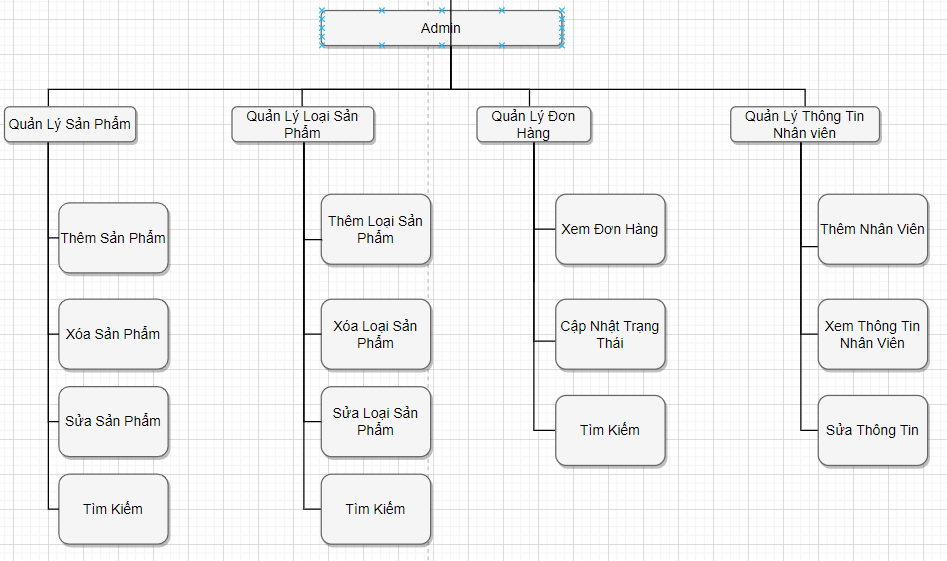
- Khi người dùng nhận được sản phẩm hoặc khi sử dụng nhưng có dấu hiệu hư hỏng , không tươi hoặc không đúng chất lượng , người dùng phải chụp ảnh và gọi điện cho bộ phận CSKH . Nếu Quản trị viên xác nhận hàng hóa của cửa hàng bị lỗi, hư hỏng, sẽ tiến hành đổi 1:1 và tiến hành quy trình giao hàng.

# Ứng dụng

## Sơ đồ chức năng



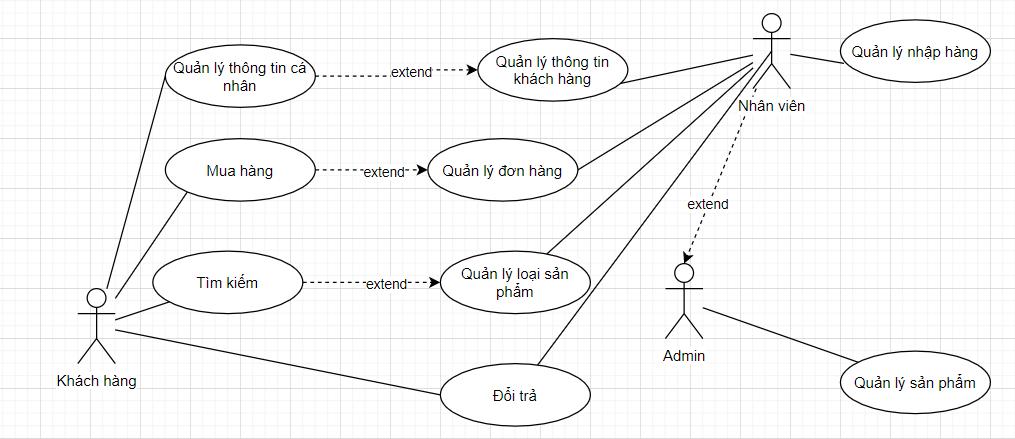
Hình 5-1: *Sơ đồ chức năng phía client*



Hình 5-2: *Sơ đồ chức năng phía Admin*

## Sơ đồ use case

### Sơ đồ Usecase tổng quát



Hình 5-2: *Sơ đồ Usecase tổng quát*

|  |  |  |  |
| --- | --- | --- | --- |
| Actor/Chức Năng | Admin | Nhân Viên | Khách hàng |
|  |  |  |  |
|  |  |  |  |

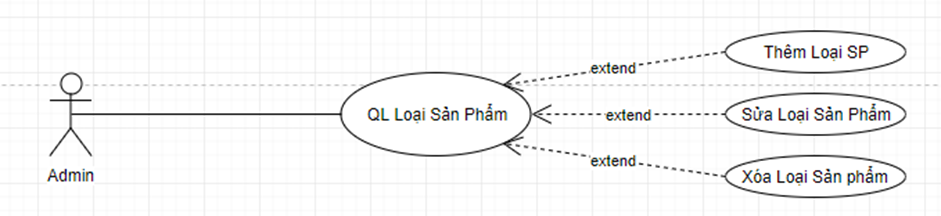
### Mô tả Usecase Quản lý Nhập hàng

Hình 5-3: *Sơ đồ Usecase quản lý nhập hàng*

Mô tả Usecase quản lý nhập hàng:

|  |  |
| --- | --- |
| Tên Use case | Quản lý Nhập Hàng |
| Actor | Admin Nhân viên |
| Mô tả | Admin nhân viên xem thông tin sản phẩm và cập nhật số lượng sản phẩm. |
| Điều kiện trước xử lý | Admin nhân viên đăng nhập vào trang quản trị |
| Điều kiện trước xử lý | Success: Hiển thị thông tin đã cập nhật  Fail: Báo lỗi cho quản trị viên |
| Luồng sự kiện chính | Admin,nhân viên chọn chức năng Quản lý Nhập Hàng.  Hệ thống hiển thị màn hình Quản lý Nhập Hàng.  Extend Use Case CapNhatSoLuong |
| Luồng sự kiện phụ | Admin nhấn nút Thoát  Hệ thống hủy màn hình Quản lý Nhập Hàng. |
| <Extend Use Case> | **CapNhatSoLuong**  1. Admin,nhân viên nhập thông tin đơn hàng.  2. Cập nhật số lượng của sản phẩm  3. Admin nhấn nút cập nhật  4. Cập nhật CSDL.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  7. Hủy màn hình cập nhật Quản lý Nhập hàng.  Rẽ nhánh 1:  2.1. Kiểm tra số lượng,mặt hàng có tồn tại không, có thì cập nhật csdl, không thì tạo mới  3.1. Lập lại bước 1 luồng sự kiện chính. |
| Tên Use case | Quản lý Nhập Hàng |
| Actor | Admin,Nhân viên |

### **Mô tả Usecase Quản lý Loại sản phẩm**



Hình 5-4: *Sơ đồ Usecase quản lý loại sản phẩm*

\*Mô tả Usecase quản lý nhập hàng

|  |  |
| --- | --- |
| Tên Use case | Quản lý Loại Sản Phẩm |
| Actor | Admin,nhân viên |
| Mô tả | Admin,nhân viên xem thông tin các loại sản phẩm và thực hiện các chức năng: Thêm, Xóa, Sửa loại sản phẩm. |
| Pre-conditions |  |
| Post-conditions | Success:  Fail: |
| Luồng sự kiện chính | Admin,nhân viên chọn chức năng Quản lý Loại Sản Phẩm.  Hệ thống hiển thị màn hình Quản lý Loại Sản Phẩm.   * Extend Use Case ThemLoaiSP * Extend Use Case SuaLoaiSP * Extend Use Case XoaLoaiSP |
| Luồng sự kiện phụ | Admin nhấn nút Thoát  Hệ thống hủy màn hình Quản lý Loại Sản Phẩm. |
| <Extend Use Case> | **ThemLoaiSP**  1. Admin nhập loại sản phẩm muốn thêm  2. Kiểm tra ràng buộc thông tin.  3. Admin nhấn nút thêm.  4. Cập nhật CSDL.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  7. Hủy màn hình cập nhật Quản lý Loại Sản Phẩm.  Rẽ nhánh 1:  2.1. Kiểm tra thông tin bị rỗng và không đúng yêu cầu sẽ hiện thị bảng thông báo ngay ô nhập |
| <Extend Use Case> | **SuaLoaiSP**  1. Admin chọn, nhập tên loại sản phẩm rồi sửa  2. Admin nhấn nút sửa  3. Cập nhật CSDL.  4. Hiển thị thông tin cập nhật.  5. Admin nhấn nút thoát.  6. Hủy màn hình cập nhật Quản lý Loại Sản Phẩm.  Rẽ nhánh 1:  2.1. Thông tin bị trùng, rỗng.  3.1. Lập lại bước 1 luồng sự kiện chính. |
| <Extend Use Case> | **XoaLoaiSP**  1. Admin chọn loại sản phẩm muốn xóa  2.Kiểm tra xem loại sản phẩm đó có sản phẩm nào chưa?  3. Admin nhấn nút xóa  4. He thong yeu xac nhan  5. Dong y  4. Cập nhật CSDL.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  7. Hủy màn hình cập nhật Quản lý Loại Sản Phẩm.  Rẽ nhánh 1:  2.1. Loại Sản Phẩm muốn xóa có sản phẩm  3.1. Lập lại bước 1 luồng sự kiện chính. |

### **Mô tả Usecase Quản lý Thông Tin Khách hàng**

Hình 5-5: *Sơ đồ Usecase quản lý thông tin khách hàng*

\*Mô tả Usecase quản lý thông tin khách hàng

|  |  |
| --- | --- |
| Tên Use case | Quản lý Thông Tin Khách Hàng |
| Actor | Admin,nhân viên |
| Mô tả | Admin,nhân viên xem thông tin các khách hàng và thực hiện các chức năng: Thêm, Xóa, Sửa và Tìm kiếm khách hàng. |
| Pre-conditions |  |
| Post-conditions | Success:  Fail: |
| Luồng sự kiện chính | Admin chọn chức năng Quản lý Thông Tin Khách Hàng.  Hệ thống hiển thị màn hình Quản lý Thông Tin Khách Hàng.   * Extend Use Case ThemKhachHang * Extend Use Case SuaKhachHang * Extend Use Case XoaKhachHang * Extend Use Case TimKiemKhachHang |
| Luồng sự kiện phụ | Admin nhấn nút Thoát  Hệ thống hủy màn hình Quản lý Thông Tin Khách Hàng. |
| <Extend Use Case> | **ThemKhachHang**  1. Admin nhập thông tin khách hàng  2. Kiểm tra ràng buộc thông tin.  3. Admin nhấn nút thêm.  4. Cập nhật CSDL.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  7. Hủy màn hình cập nhật Quản lý thông tin khách hàng.  Rẽ nhánh 1:  2.1. Kiểm tra thông tin bị rỗng.  3.1. Lập lại bước 1 luồng sự kiện chính.  Rẽ nhánh 2:  4.1. Thông tin bị trùng.  5.1. Lập lại bước 1 luồng sự kiện chính.  Rẽ nhánh 3:  4.2. Kiểm tra thông tin khách hàng hợp lệ.  4.3. Cập nhật CSDL. |
| <Extend Use Case> | **SuaKhachHang**  1. Admin chọn, nhập thông tin khách hàng  2. Admin nhấn nút sửa  3. Cập nhật CSDL.  4. Hiển thị thông tin cập nhật.  5. Admin nhấn nút thoát.  6. Hủy màn hình cập nhật Quản lý thông tin khách hàng.  Rẽ nhánh 1:  4.1. Thông tin bị trùng.  5.1. Lập lại bước 1 luồng sự kiện chính.  Rẽ nhánh 2:  4.2. Kiểm tra thông tin khách hàng hợp lệ. |
| <Extend Use Case> | **XoaKhachHang**  1. Admin nhập id khách hàng cần xóa  2.Kiểm tra xem có id khách hàng đó không  3. Admin nhấn nút xóa  4. Cập nhật CSDL.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  7. Hủy màn hình cập nhật Quản lý thông tin khách hàng.  Rẽ nhánh 1:  2.1. Không có Tên khách hàng này  2.1. Lập lại bước 1 luồng sự kiện chính. |
| <Extend Use Case> | **TimKiemKhachHang**  1. Admin nhập tên khách hàng cần tìm  2.Kiểm tra xem có tên khách hàng không  3. Admin nhấn nút tìm  4. Hiển thị CSDL.  5. Admin nhấn nút thoát.  6. Hủy màn hình cập nhật Quản lý thông tin khách hàng.  Rẽ nhánh 1:  2.1. Không có Tên khách hàng này  2.1. Lập lại bước 1 luồng sự kiện chính. |

### Mô tả Usecase Quản lý Đơn hàng

Hình 5-6: *Sơ đồ Usecase quản lý đơn đặt hàng*

Mô tả Usecase quản lý đơn đặt hàng

|  |  |
| --- | --- |
| Tên Use case | Quản lý ĐƠN HÀNG |
| Actor | Admin,nhân viên |
| Mô tả | Admin, nhân viên xem thông tin các đơn đặt hàng và thực hiện các chức năng: Cập nhật, xóa |
| Pre-conditions |  |
| Post-conditions | Success:  Fail: |
| Luồng sự kiện chính | Admin chọn chức năng Quản lý Đơn đặt hàng  Hệ thống hiển thị màn hình Quản lý Đơn đặt hàng   * Extend Use Case CapNhatDonHang * Extend Use Case HuyDonHang |
| Luồng sự kiện phụ | Admin nhấn nút Thoát  Hệ thống hủy màn hình Quản lý Đơn đặt hàng. |
| <Extend Use Case> | **CapNhatDonHang**  1. Admin nhập thông tin đơn hàng muốn cập nhật  2. Kiểm tra thông tin.  3. Admin nhấn nút cập nhật.  4. Cập nhật CSDL ở Server và bên phía người dùng.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  7. Hủy màn hình cập nhật Quản lý Đơn đặt hàng  Rẽ nhánh 1:  2.1. Kiểm tra thông tin hợp lệ, nếu không thì sẽ thông báo ở ô nhập và khóa ô cập nhật |
| <Extend Use Case> | **HuyDonHang**  1. Admin chọn đơn cần hủy  2. Admin nhấn nút hủy  3. Xác nhận đồng ý  4. Cập nhật CSDL.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  Rẽ nhánh 1 :  3.1 Bấm hủy thì hệ thống sẽ trở về trang quản lý đơn hàng |

### Mô tả Usecase Quản lý Sản phẩm

Hình 5-7: *Sơ đồ Usecase quản lý sản phẩm*

Mô tả Usecase quản lý đơn đặt hàng

|  |  |
| --- | --- |
| Tên Use case | Quản lý Sản Phẩm |
| Actor | Admin,nhân viên |
| Mô tả | Admin, nhân viên xem thông tin các Sản Phẩm và thực hiện các chức năng: Thêm, Cập nhật và Tìm kiếm Sản Phẩm. |
| Pre-conditions |  |
| Post-conditions | Success:  Fail: |
| Luồng sự kiện chính | Admin chọn chức năng Quản lý Sản Phẩm  Hệ thống hiển thị màn hình Quản lý Sản Phẩm   * Extend Use Case ThemSanPham * Extend Use Case CapNhatSanPham * Extend Use Case TimKiemSanPham |
| Luồng sự kiện phụ | Admin nhấn nút Thoát  Hệ thống hủy màn hình Quản lý Sản Phẩm. |
| <Extend Use Case> | **ThemSanPham**  1. Admin nhập thông tin Sản Phẩm  2. Kiểm tra thông tin không rỗng, đã tồn tại .  3. Admin nhấn nút thêm.  4. Cập nhật CSDL.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  7. Hủy màn hình cập nhật Quản lý Sản Phẩm.  Rẽ nhánh 1:  2.1. Kiểm tra thông tin bị rỗng, trùng.  2.2 Hiện thông báo nhập sai ở ngay ô nhập và không cho ấn nút Thêm |
| <Extend Use Case> | **CapNhatSanPham**  1. Admin nhập thông tin mới của Sản Phẩm  2. Hệ thống kiểm tra thông tin vừa nhập  3. Admin nhấn nút cập nhật  4. Cập nhật CSDL.  5. Hiển thị thông tin cập nhật.  6. Admin nhấn nút thoát.  7. Hủy màn hình cập nhật Quản lý Sản Phẩm.  Rẽ nhánh 1:  2.1. Kiểm tra thông tin Sản Phẩm bị rỗng, không đúng yêu cầu, hiện thị thông báo ngay ô nhập và không được phép ấn nút cập nhật  Rẽ nhánh 2:  6.1 Hiện thông báo đã cập nhật thành công, nhấn OK để quay trở về trang Quản Lý Sản Phẩm |
| <Extend Use Case> | **TimKiemSanPham**  1. Admin nhập tên Sản Phẩm cần tìm  2.Kiểm tra xem có tên Sản Phẩm không  3. Admin nhấn nút tìm  4. Hiển thị CSDL.  5. Admin nhấn nút thoát.  6. Quay về trang quản lý sản phẩm.  Rẽ nhánh 1:  2.1. Nếu tìm thấy sản phẩm , hệ thống xuất ra thông tin sản phẩm  2.2 Nếu không có sản phẩm , hệ thống hiện dòng chữ không tìm thấy sản phẩm. |

### **Mô tả Usecase Quản lý Đổi trả**

Hình 5-8 *Sơ đồ Usecase quản lý đổi trả*

Mô tả Usecase quản lý đổi trả

|  |  |
| --- | --- |
| Tên Use case | Quản lý Đổi Trả |
| Actor | Admin,nhân viên |
| Mô tả | Khi giao hàng hoặc đang sử dụng, khách hàng kiểm tra sản phẩm. Nếu sản phẩm bị lỗi hoăc hư hại và không giống như hình ảnh, thì khách hàng có thể đến cửa hàng đổi trả sản phẩm. |
| Pre-conditions |  |
| Post-conditions | Success:  Fail: |
| Luồng sự kiện chính |  |
| Luồng sự kiện phụ |  |
| <Extend Use Case> | **DoiTra**   1. KH kiểm tra hàng:    1. Hàng bị hư, không giống trong hình: đổi trả    2. Hàng ổn, khách hàng kí nhận đơn 2. Kết thúc |

### C:\Users\MyPC\AppData\Local\Microsoft\Windows\INetCache\Content.Word\huydonhang.png**Mô tả Usecase Quản lý Đơn hàng phía người dùng**

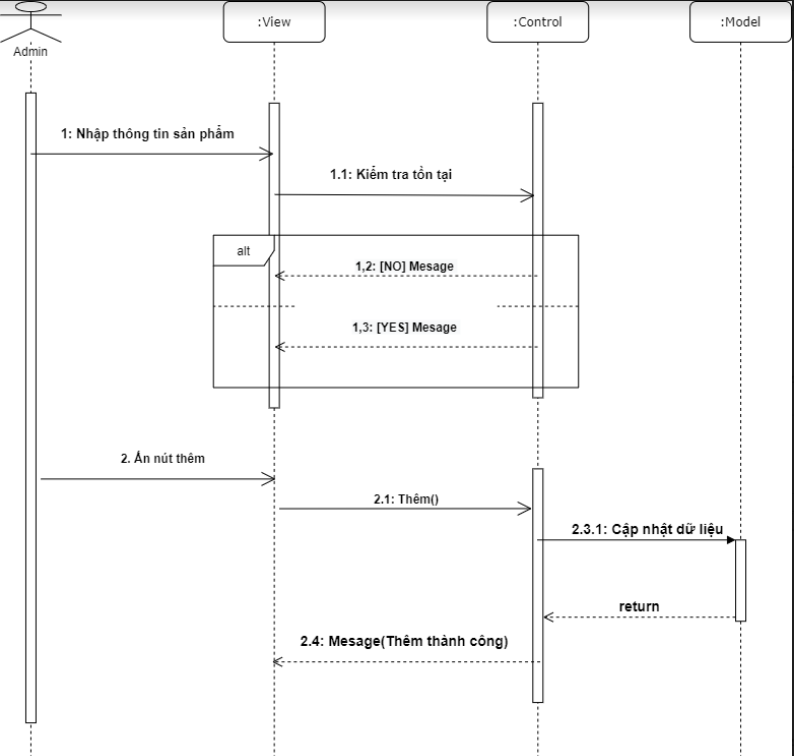
Hình 5-9 Sơ đồ Usecase quản lý đơn hàng từ phía người dùng

Mô tả Usecase quản lý đơn hàng phía khách hàng

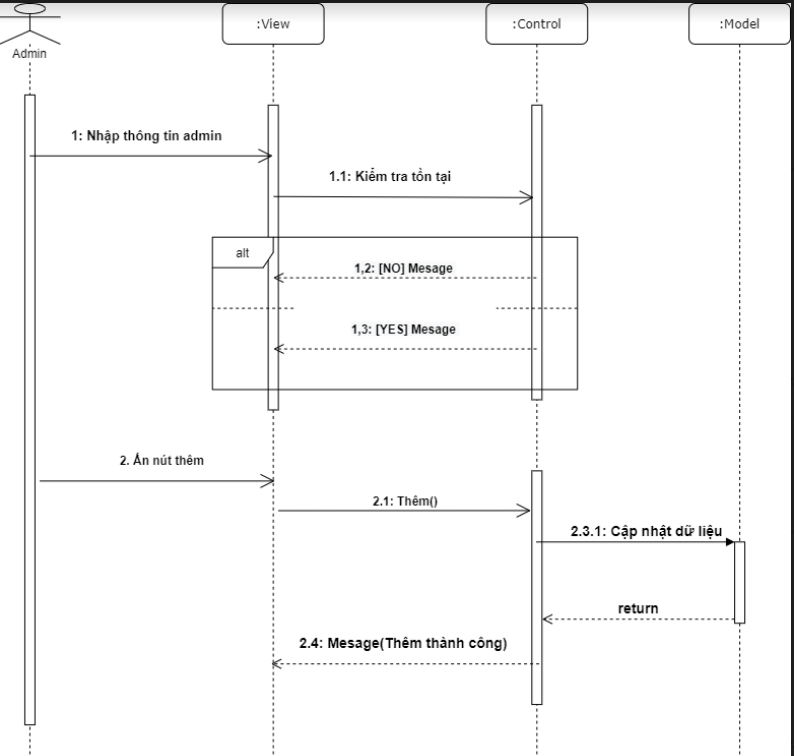
|  |  |
| --- | --- |
| Tên Use case | Quản lý Thông tin Khách hàng phía người dùng. |
| Actor | Khách hàng |
| Mô tả | Khách hàng khi muốn chỉnh sửa thông tin đơn hàng của mình, ấn vào nút đơn hàng |
| Pre-conditions |  |
| Post-conditions | Success:  Fail: |
| Luồng sự kiện chính | Khách hàng chọn chức năng xem đơn hàng.  Hệ thống hiển thị màn hình thông tin đơn hàng phía người dùng.   * Extend Use Case HuyDonhang |
| Luồng sự kiện phụ | Người dùng nhấn nút Thoát  Hệ thống hủy màn hình Quản lý Thông tin đơn hàng phía người dùng và hiện thị lại trang đơn hàng |
| <Extend Use Case> | **SuaThongTin**  1. Khách hàng vào Quản lý đơn hàng của khách hàng.  2. Khách hàng tùy chỉnh lại trạng thái đơn hàng  Rẽ nhánh   * 1. Hiện thị yêu cầu xác nhận lại trạng thái vừa chỉnh sửa   2. Nếu nhấn Ok hệ thống sẽ cập nhật CSDL và quay về màn hình Quản lý Thông tin đơn hàng phía người dùng   3. Nếu nhấn Hủy hệ thống sẽ trở về màn hình màn hình Quản lý Thông tin đơn hàng phía người dùng |

## Sơ đồ hoạt động

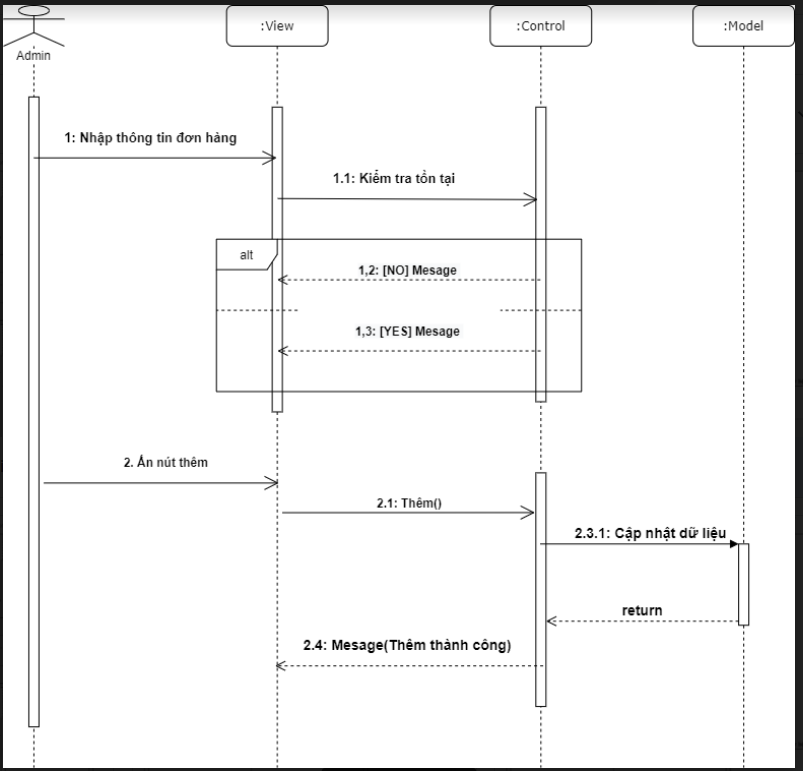
Thêm Sản Phẩm



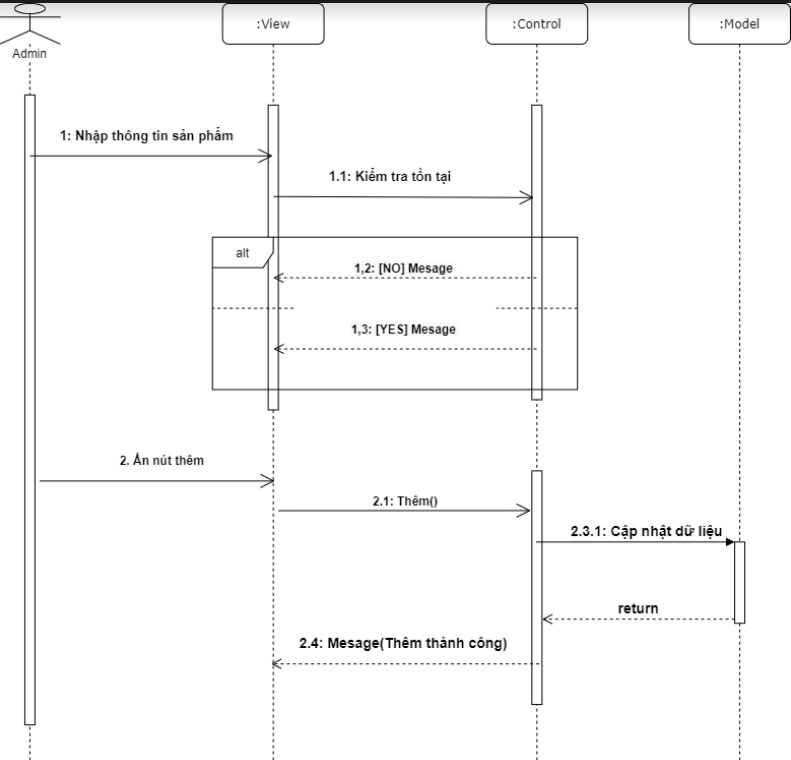
Thêm tài khoản Admin



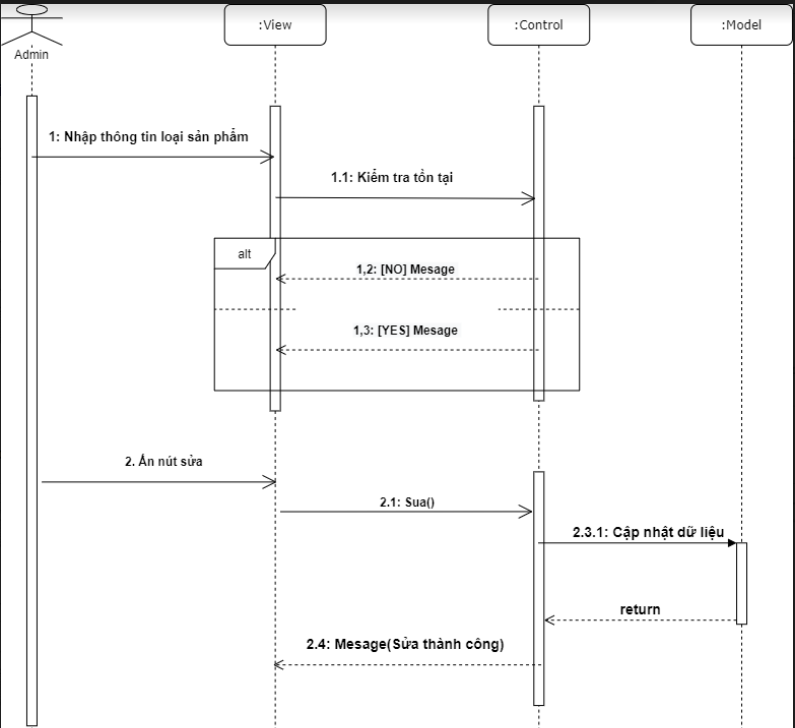
Thêm Đơn Hàng



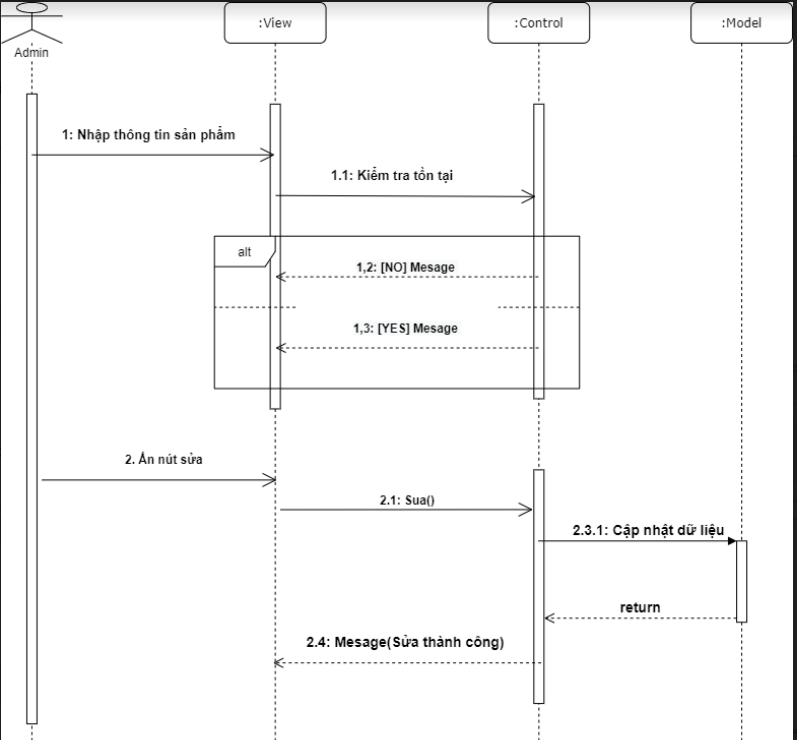
Thêm Sản Phẩm



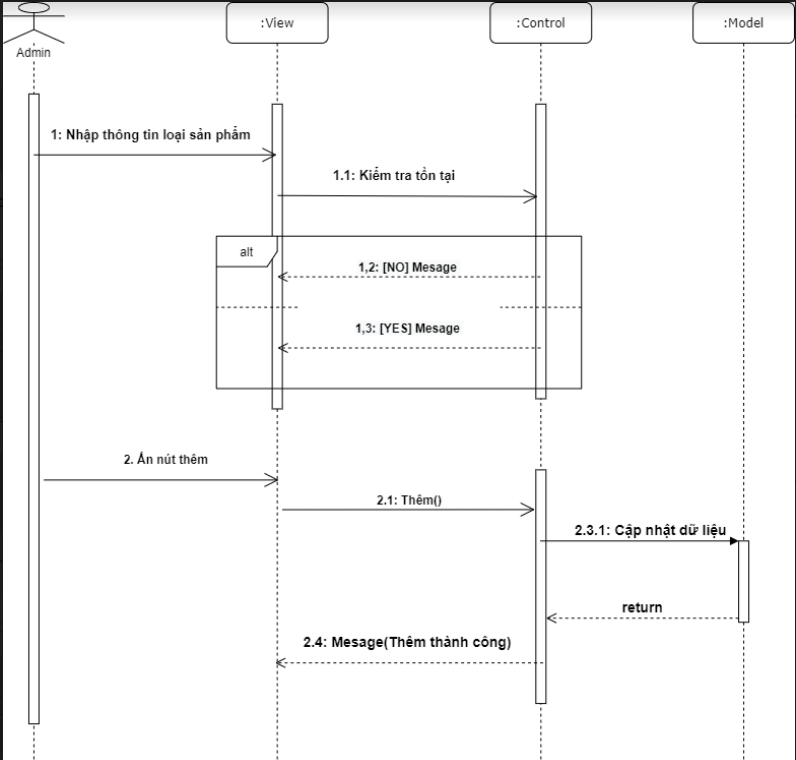
Sửa loại Sp



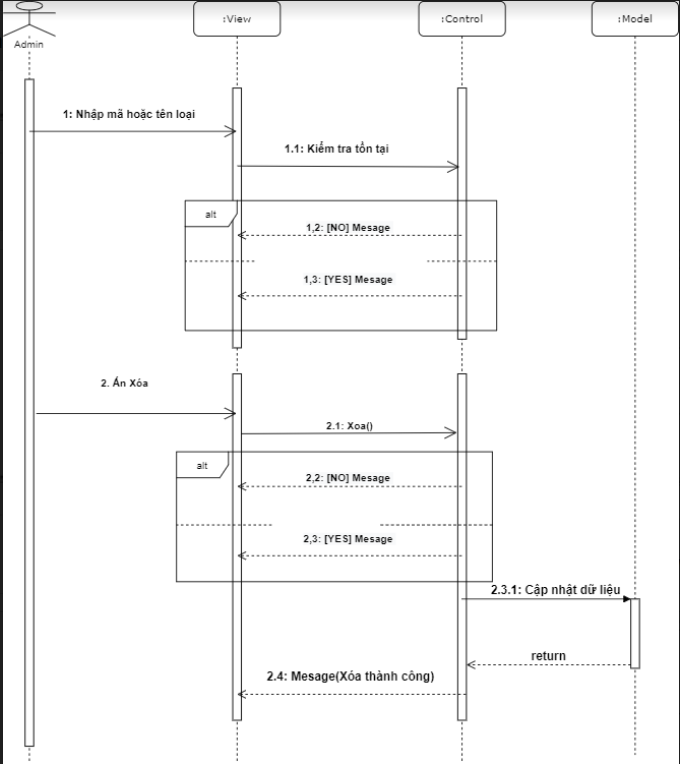
Sửa sản phẩm



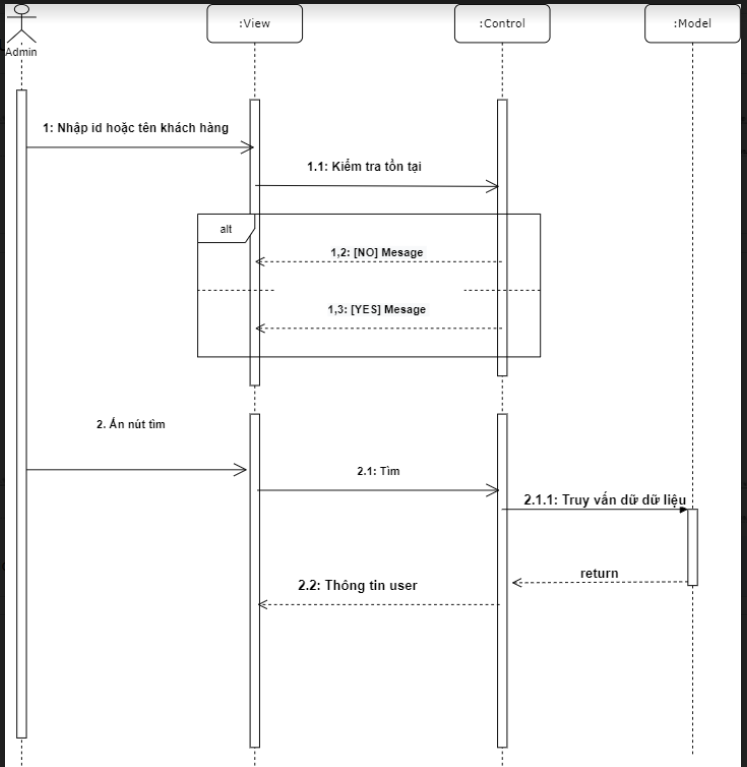
Thêm Sp



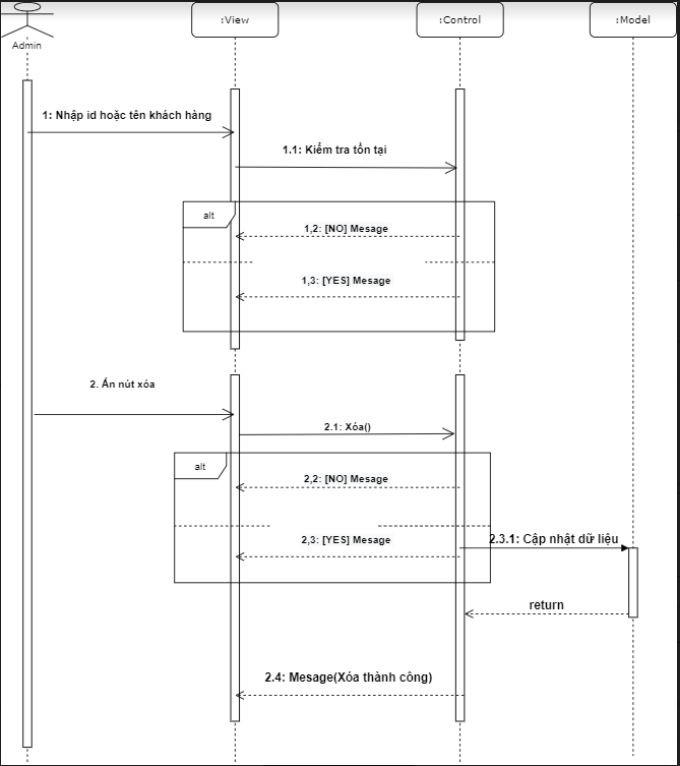
Sửa loại Sp



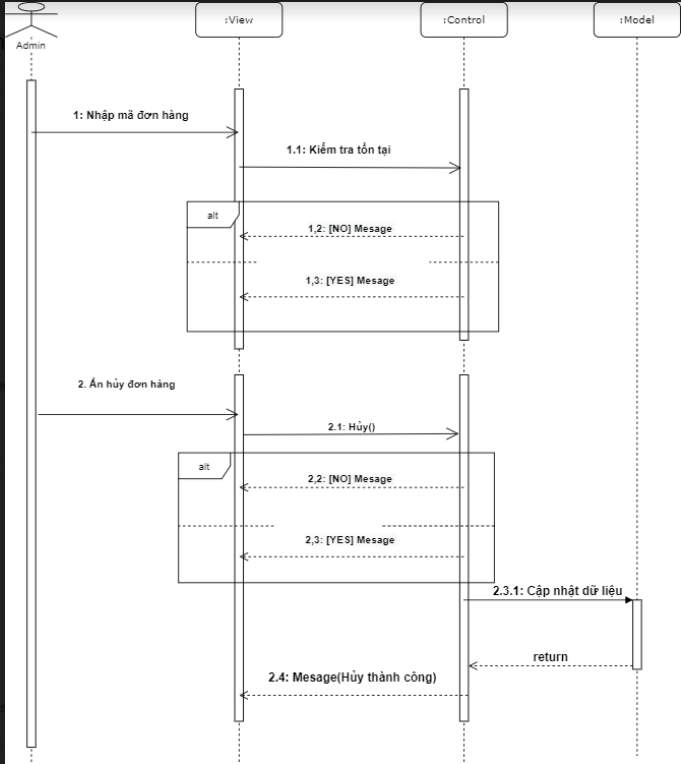
Admin tìm kiếm User



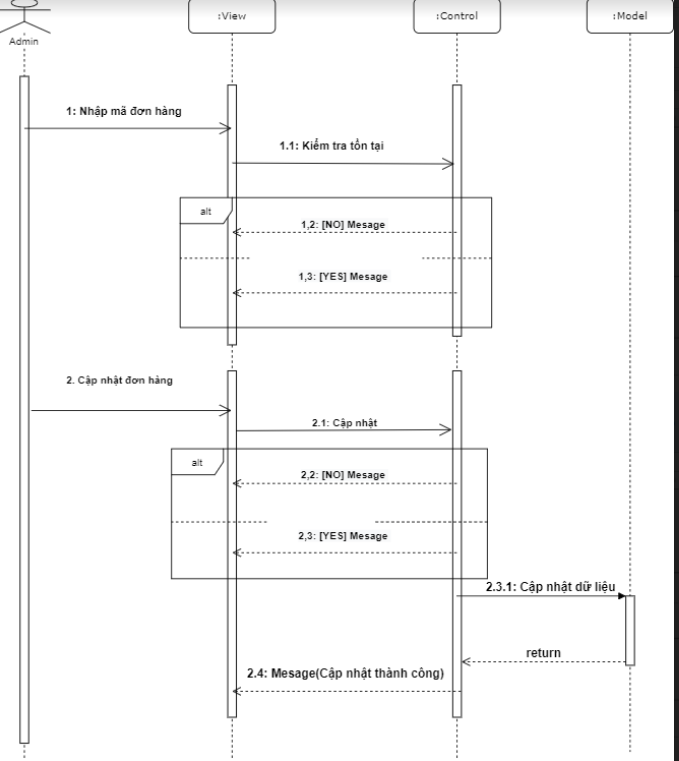
Admin Delete User



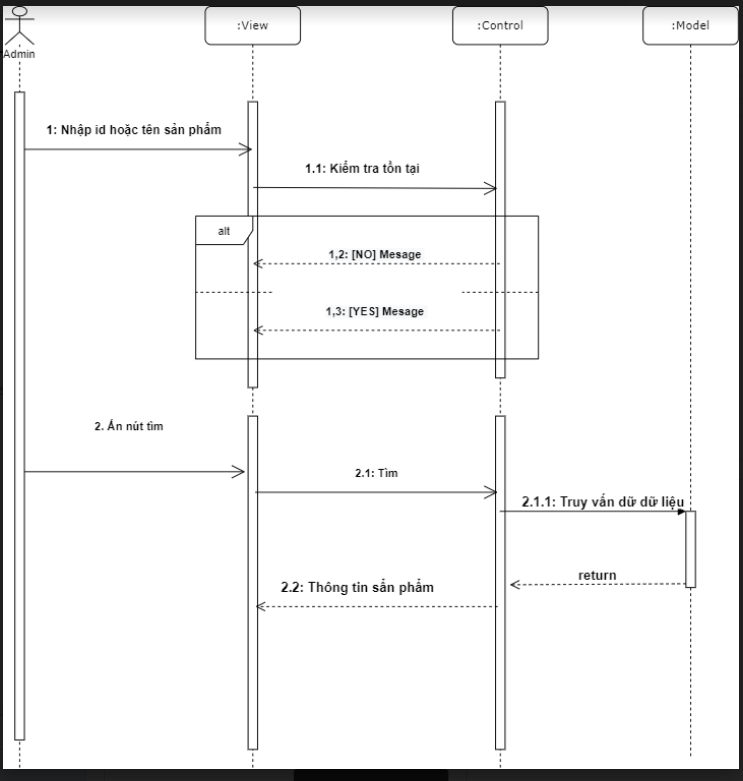
Hủy Đơn Hàng



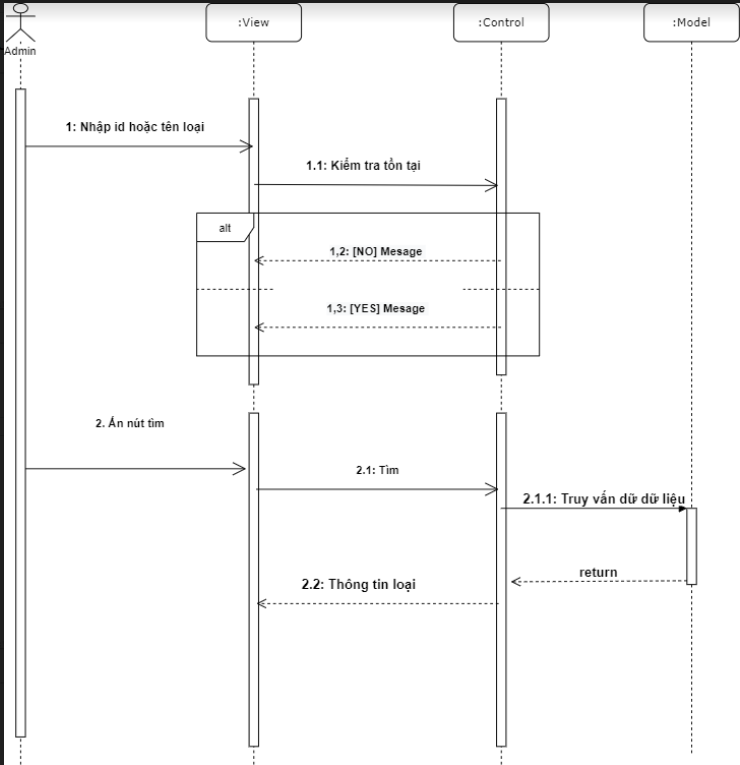
Updatẻ Đơn hàng



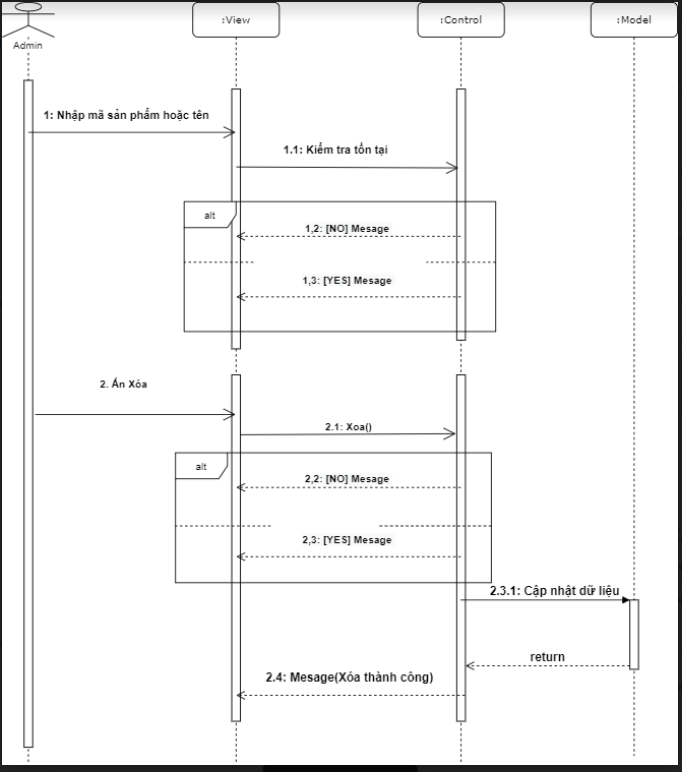
Tìm Sản Phẩm



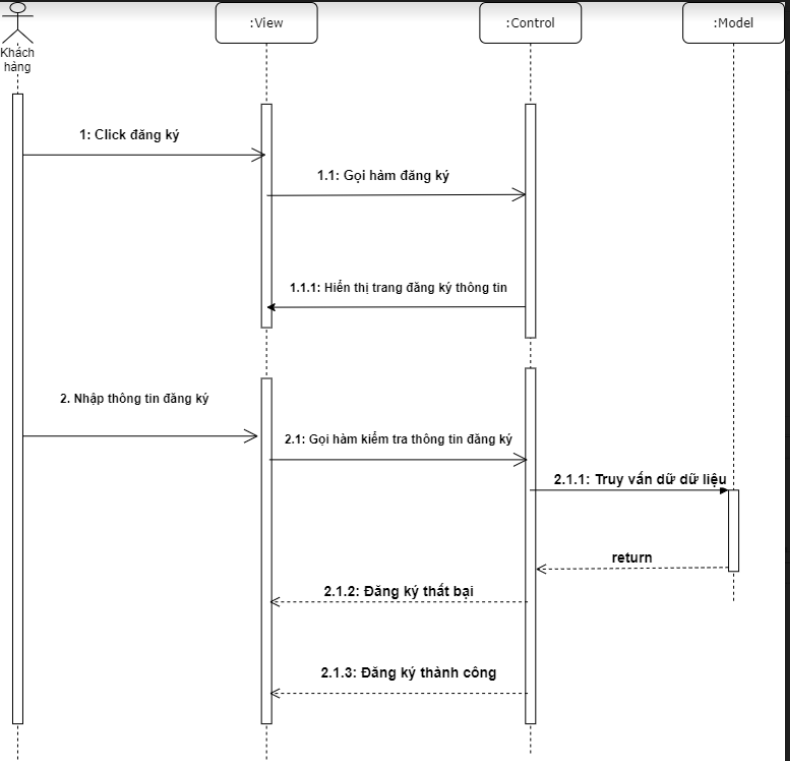
Tìm loại Sp



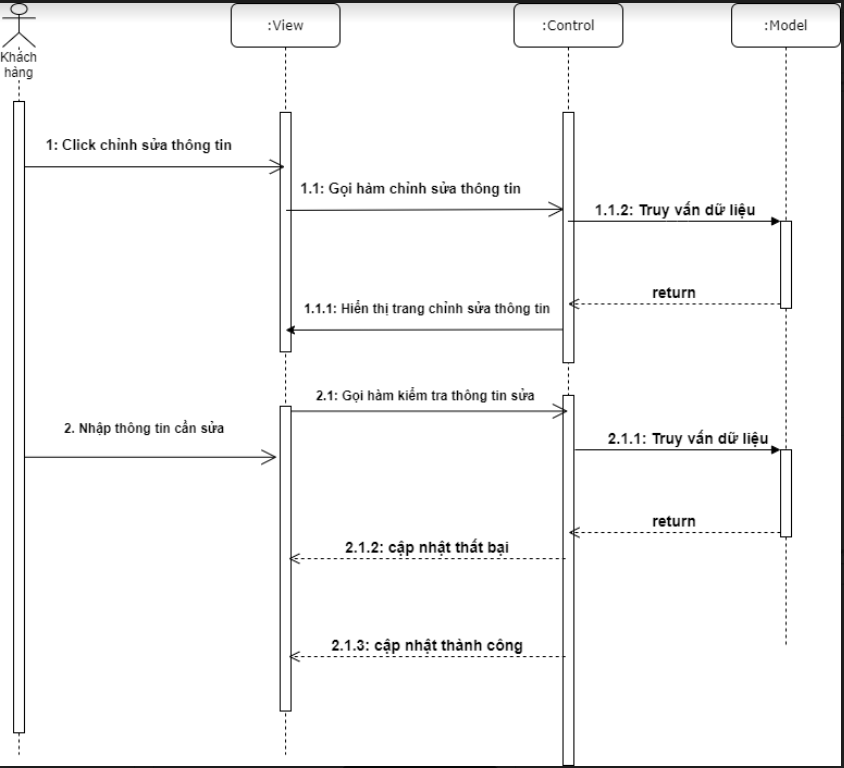
Tìm Loại Sp



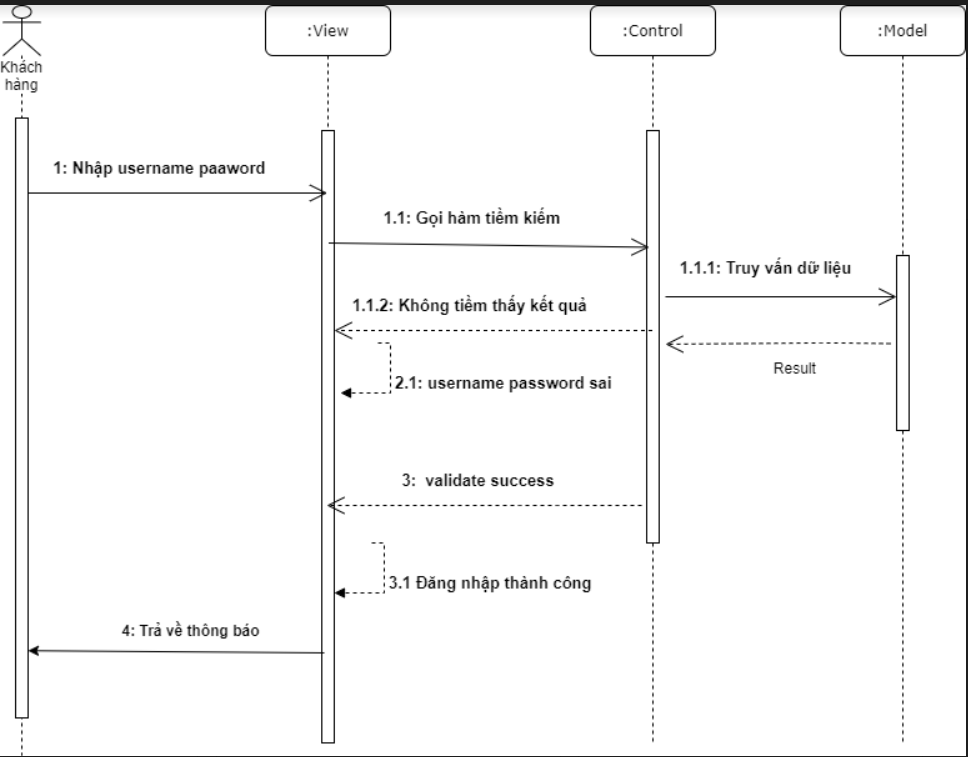
User Đăng kí



User Info



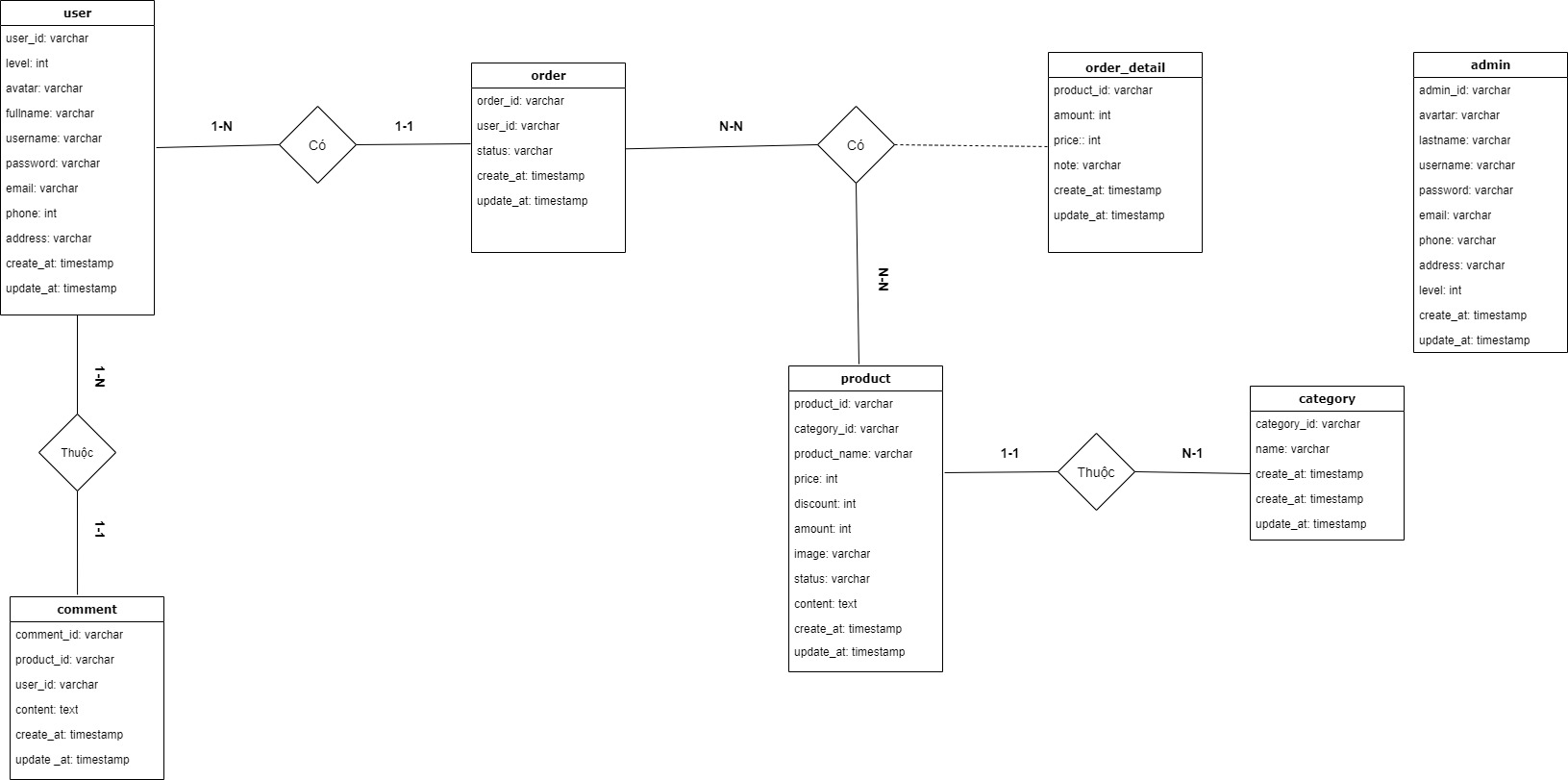
User Login



# Thành phần dữ liệu

## Phân tích ở mức quan niệm

### Sơ đồ ER / sơ đồ lớp



Hình 6-1. Sơ đồ ER

### Mô tả các loại thực thể

* Loại thực thể USER

| **Mô tả:** Loại thực thể USER gồm những thông tin khách hàng truy cập vào trang web… | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| user\_id | Varchar (30) | x | x | x | Mã khách hàng |
| level | Int (5) |  |  | x | Cấp độ khách hàng |
| avarta | Varchar (50) |  |  | x | Ảnh đại diện khách hàng |
| Lastname | Varchar (30) |  |  | x | Họ khách hàng |
| username | Varchar (10) |  |  | x | Tên khách hàng |
| password | Varchar (50) |  |  | x | Mật khẩu đăng nhập |
| email | Varchar (50) |  | X |  | Địa chỉ email |
| phone | Int (11) |  |  | x | Số điện thoại |
| address | Varchar (50) |  |  | x | Địa chỉ |
| create\_at | timestamp |  |  | x | Ngày thêm |
| update\_at | timestamp |  |  | x | Ngày cập nhật |

* Loại thực thể COMMENT

| **Mô tả:** Loại thực thể COMMENT gồm những tin nhắn của khách hàng bình luận về một sản phẩm… | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| Common\_id | Varchar (30) | x | x | x | Mã tin nhắn |
| Product\_id | Varchar (30) |  |  | x | Mã sản phẩm |
| User\_id | varchar (50) |  | x | x | Mã khách hàng |
| Content | Text |  |  | x | Nội dung bình luận |
| Create\_at | Timestamp |  |  | x | Ngày thêm |
| Update\_at | Timestamp |  |  | x | Ngày cập cật |

* Loại thực thể ORDER

| **Mô tả:** Loại thực thể ORDER gồm những thông tin đơn hàng khách hàng đã đặt… | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| Order\_id | Varchar (30) | x | x | x | Mã tin nhắn |
| Product\_id | Varchar (30) |  |  | x | Mã sản phẩm |
| User\_id | varchar (50) |  | x | x | Mã khách hàng |
| Content | Text |  |  | x | Nội dung bình luận |
| Create\_at | Timestamp |  |  | x | Ngày thêm |
| Update\_at | Timestamp |  |  | x | Ngày cập cật |

* Loại thực thể STATUS\_ORDER

| **Mô tả:** Loại thực thể STATUS\_ORDER gồm những trạng thái của đơn hàng | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| Status\_order\_id | Varchar (30) | x | x | x | Mã trạng thái |
| name | Varchar (30) |  |  | x | Tên trạng thái |

| **Mô tả:** Loại thực thể ORDER\_DETAIL gồm những thông tin chi tiết sản phẩm của đơn hàng.. | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| order\_detail\_id | Varchar (30) | x | X | x | Mã đơn hàng chi tiết |
| Product\_id | Varchar (30) |  | X | x | Mã sản phẩm |
| amount | Int (5) |  |  | x | Số lượng sản phẩm |
| price | Int (10) |  |  | x | Giá sản phẩm |
| note | Varchar (50) |  |  | x | Ghi chú đơn hàng |
| Create\_at | Timestamp |  |  | x | Ngày thêm |
| Update\_at | Timestamp |  |  | x | Ngày cập cật |

| **Mô tả:** Loại thực thể PRODUCT gồm những thông tin của sản phẩm… | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| product\_id | Varchar (30) | x | x | x | Mã sản phẩm |
| category\_id | Varchar (30) |  |  | x | Mã loại sản phẩm |
| product\_name | Varchar (30) |  |  | x | Tên sản phẩm |
| price | Int (10) |  |  | x | Giá sản phẩm |
| discount | Int (10) |  |  | x | Giá khuyến mãi |
| image | Varchar (30) |  |  | x | Hình ảnh sản phẩm |
| status\_product\_id | Varchar (30) |  |  | x | Trạng thái của sản phẩm |
| content | Text |  |  | x | Giới thiệu thông tin sản phẩm |
| create\_at | timestamp |  |  | x | Ngày thêm |
| update\_at | timestamp |  |  | x | Ngày cập nhật |

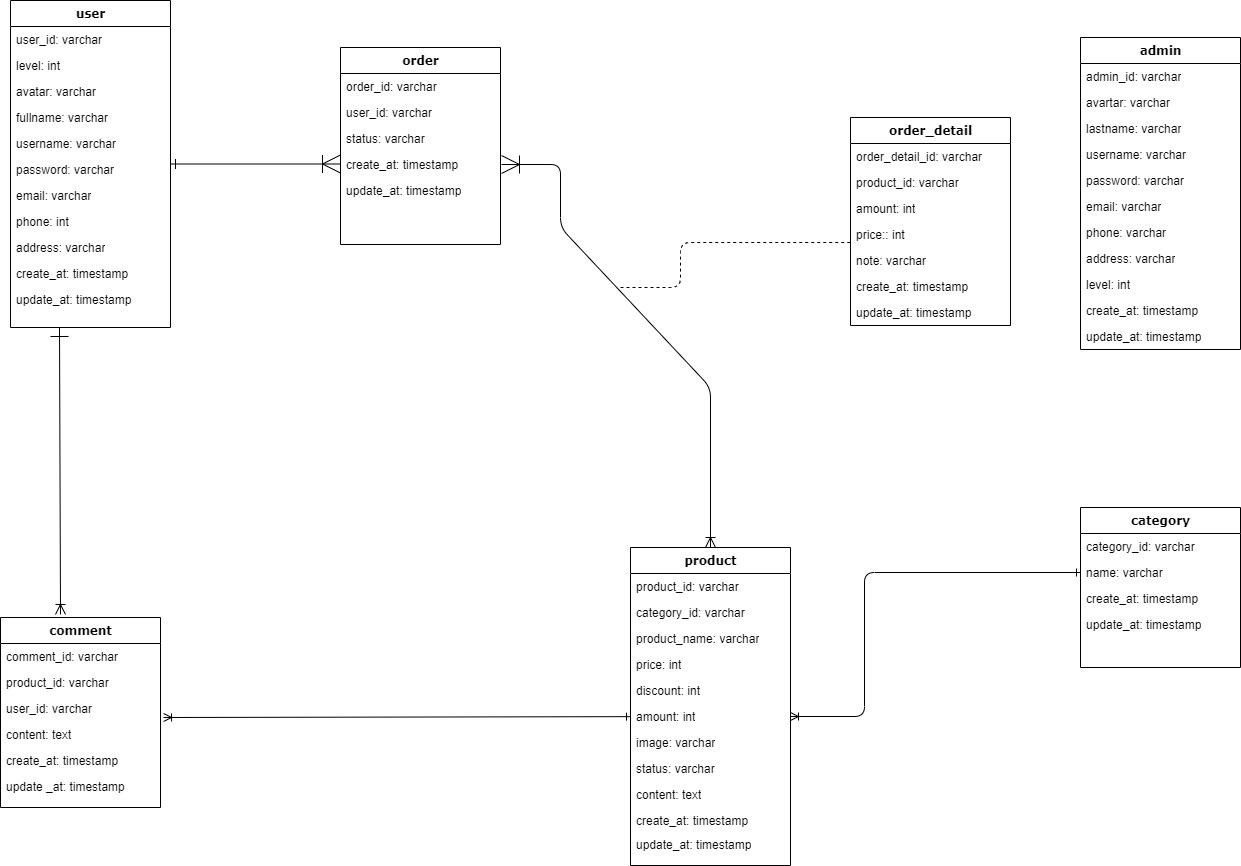
* Loại thực thể STATUS\_PRODUCT

| **Mô tả:** Loại thực thể STATUS\_ORDER gồm những trạng thái của đơn hàng | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| status\_product\_id | Varchar (30) | x | x | x | Mã trạng thái |
| name | Varchar (30) |  |  | x | Tên trạng thái |

| **Mô tả:** Loại thực thể CATEGORY gồm những thông tin loại sản phẩm.. | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| category\_id | Varchar (30) | x | X | x | Mã loại sản phẩm |
| name | Varchar (30) |  | x | x | Tên loại sản phẩm |
| create\_at | Timestamp |  |  | x | Ngày thêm |
| update\_at | Timestamp |  |  | x | Ngày cập cật |

| **Mô tả:** Loại thực thể ADMIN gồm những thông tin của người quản trị hệ thống.. | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu** | **K** | **U** | **M** | **Diễn giải** |
| admin\_id | Varchar (30) | x | x | x | Mã người quản trị |
| avarta | Varchar (30) |  |  | x | Ảnh |
| lastname | Varchar (30) |  |  | x | Họ |
| username | Varchar (30) |  |  | x | Tên |
| password | Varchar (15) |  |  | x | Mật khẩu |
| email | Varchar (30) |  |  | x | Địa chỉ email |
| phone | Int (11) |  |  | x | Số điện thoại |
| address | Varchar (30) |  |  | x | Địa chỉ |
| level | Int (3) |  |  | x | Cấp độ |
| create\_at | timestamp |  |  | x | Ngày thêm |
| update\_at | timestamp |  |  | x | Ngày cập nhật |

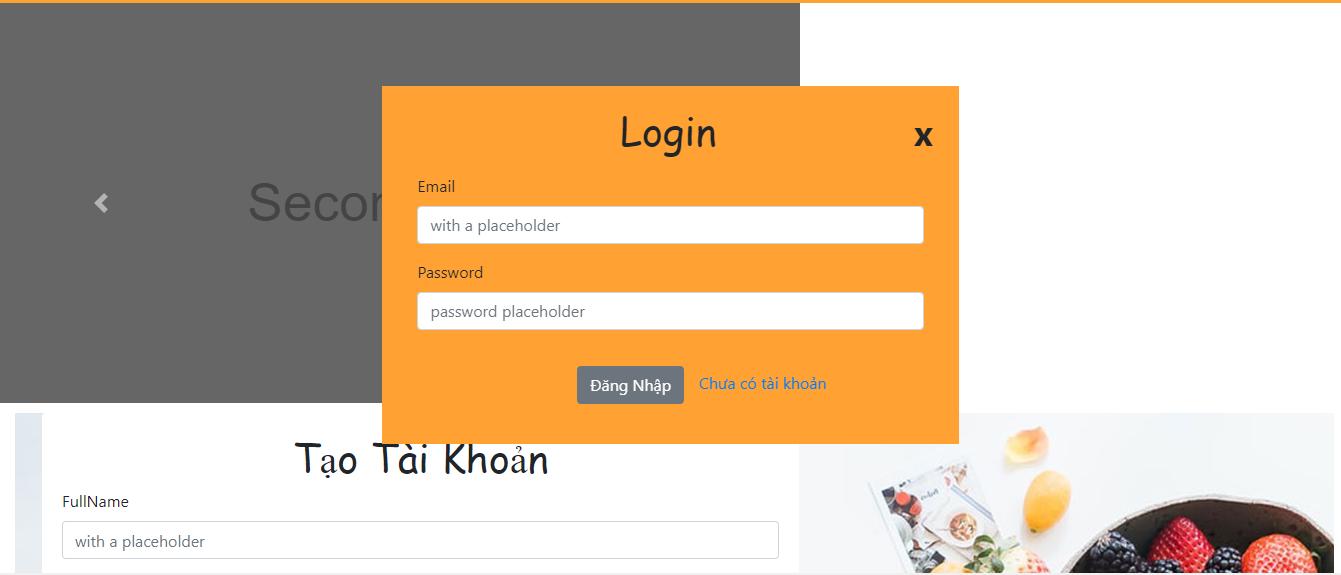
## Thiết kế cơ sở dữ liệu



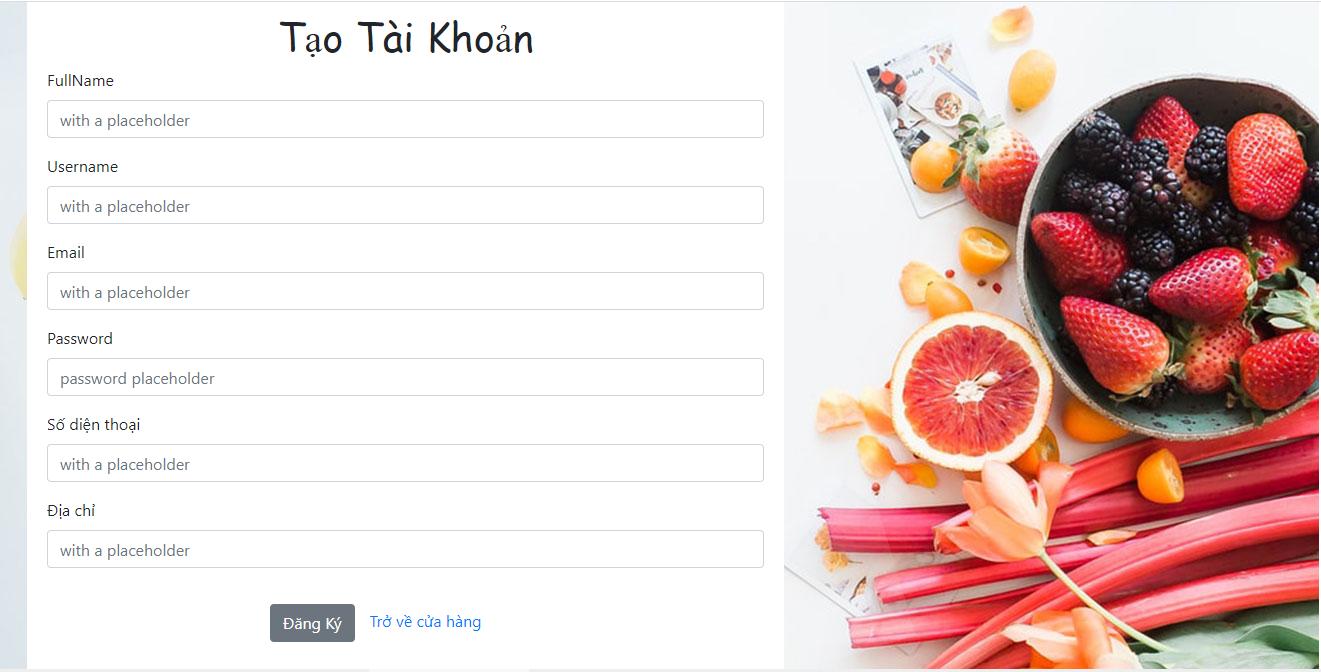
Hình 6-2. Sơ đồ logic dữ liệu

# Thành phần giao diện

## Người dùng

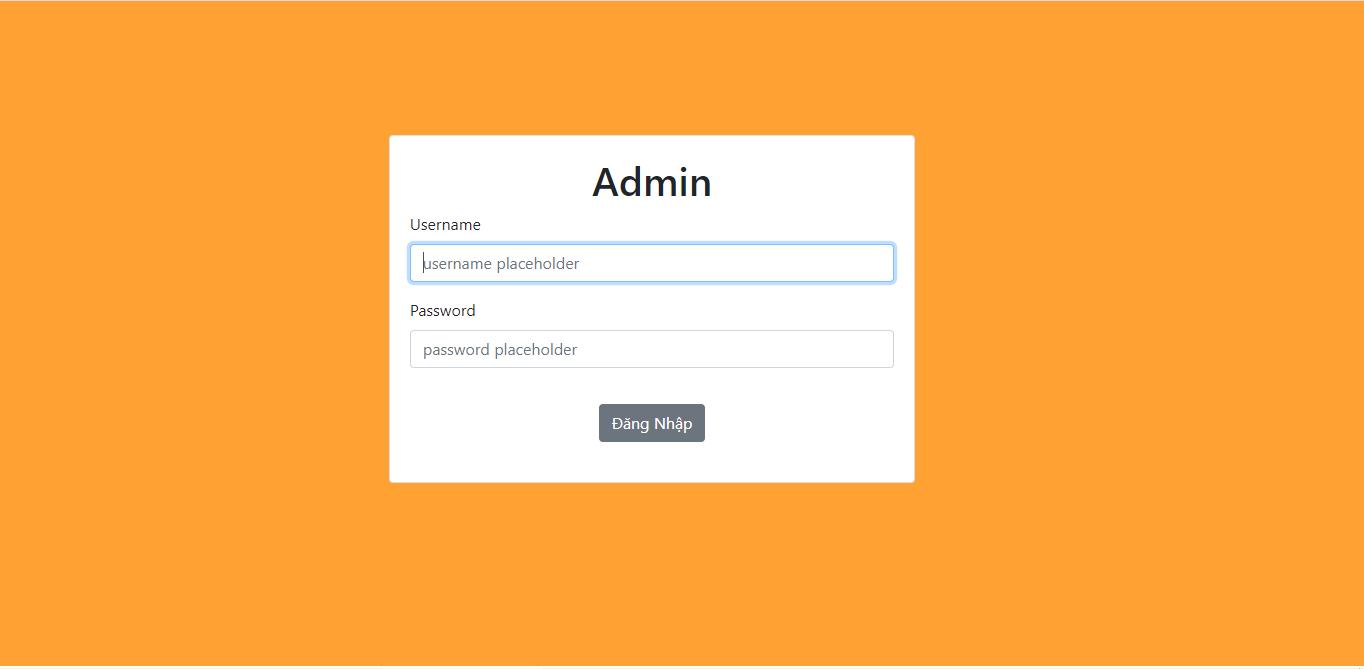


Hình 7.1: Trang đăng nhập



Hình 7.2: Trang đăng ký

## Quản lý



Hình 7.3: Trang đăng nhập admin

Chương 7 Kết luận

**Tài liệu tham khảo**

[1] ReactJS trên web: <https://reactjs.org/>

[2] Mức độ phổ biến của ReactJS: <https://insights.stackoverflow.com/survey/2019>

[3] Những trang web đang sử dụng ReactJS: <http://monsterdesign.vn/kien-thuc-website/ong-lon-thuong-mai-dien-tu-dang-su-dung-reactjs-nhu-the-nao/menu-id-492>

[4] React-router trên web: <https://reacttraining.com/react-router/web/guides/quick-start>

[5] Top 10 framework java phổ biến năm 2020 <https://stackify.com/10-of-the-most-popular-java-frameworks-of-2020/>

[6] Top 5 framework java phổ biến năm 2020<https://scand.com/company/blog/top-5-java-frameworks-of-2020/>

[7]Danh sách các framework java phổ biến cho các lập trình viên năm 2019 <https://willandway.vn/7-framework-java-pho-bien-bien-nhat-cho-cac-lap-trinh-vien-tai-cac-cong-ty-phat-trien-java-trong-nam-2019-2/>

Spring Boot trên web: spring <https://spring.io/projects/spring-boot>

[8] <https://www.jetbrains.com/lp/devecosystem-2019/java/>

[9]<https://www.kelltontech.com/kellton-tech-blog/top-7-backend-web-development-frameworks-2019>

[10] <https://insights.stackoverflow.com/survey/2019>

[11]<https://insights.stackoverflow.com/survey/2020#technology-web-frameworks-all-respondents2>