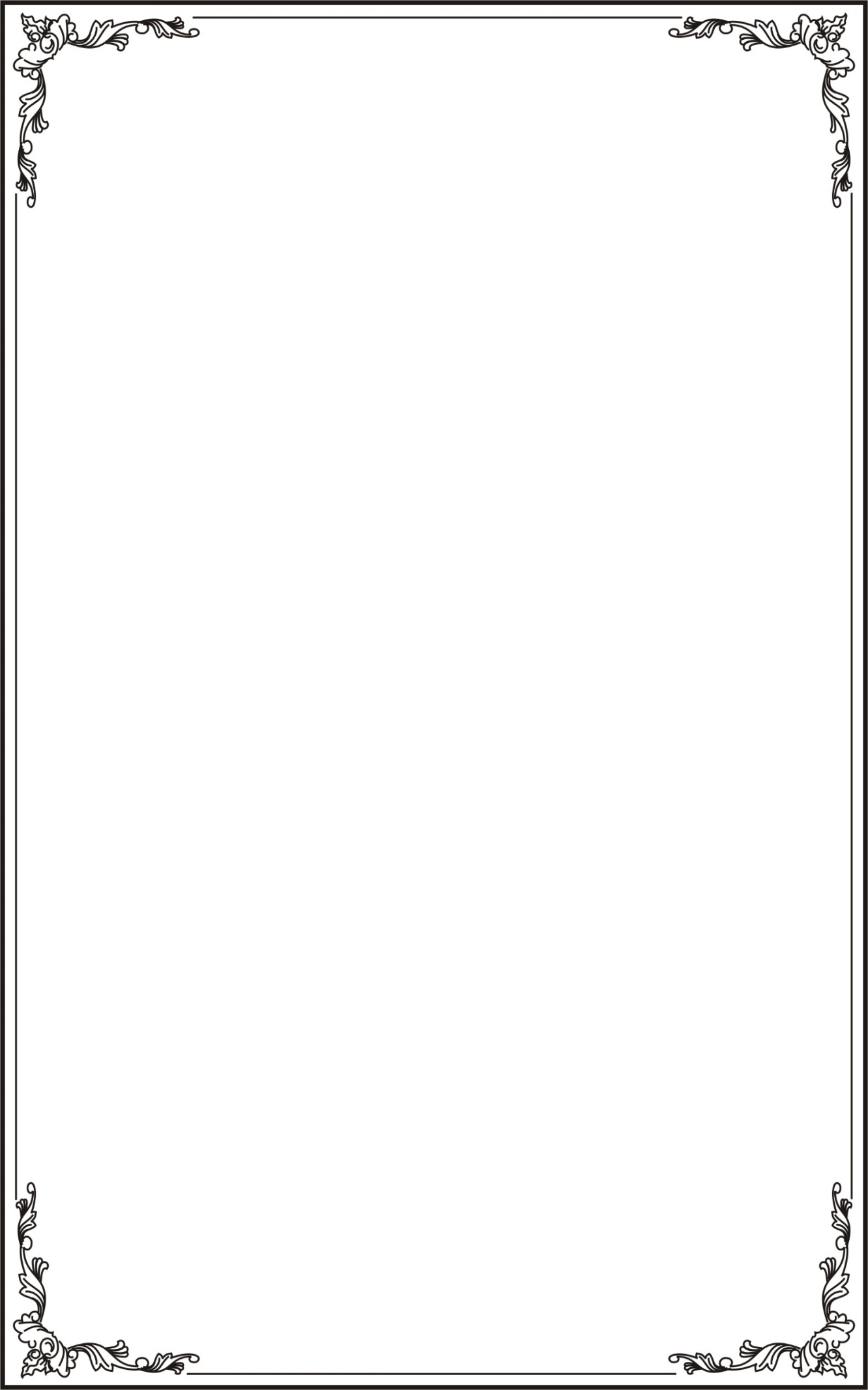
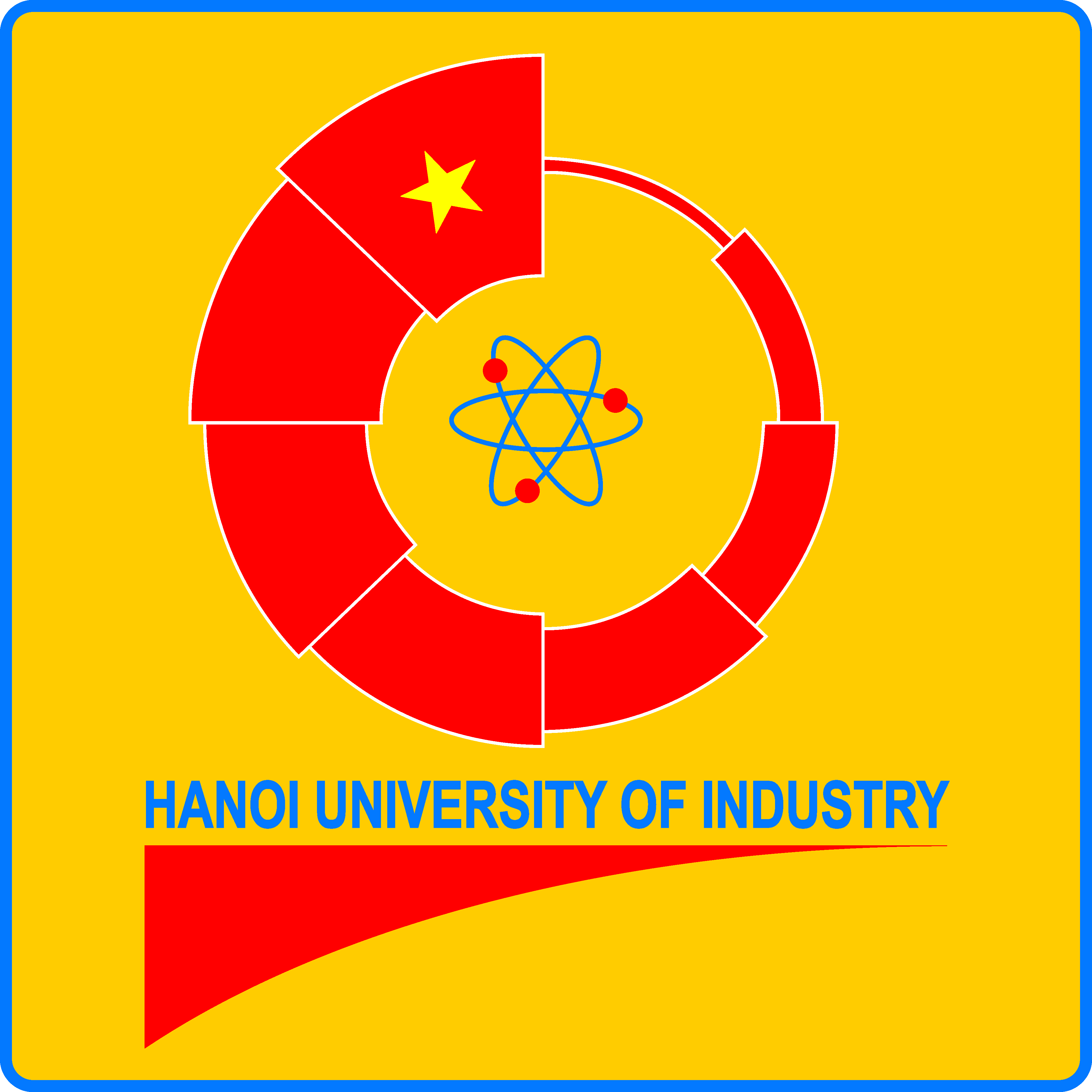
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH: KỸ THUẬT PHẦN MỀM**

***Đề tài:***

**NGHIÊN CỨU NGÔN NGỮ NODEJS THƯ VIỆN REACTJS VÀ XÂY DỰNG WEBSITE BÁN ĐỒ ĂN ĐÊM**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | : ThS. TRỊNH BÁ QUÝ |
| Sinh viên thực hiện | : TẠ THỊ MỸ HẠNH |
| Mã sinh viên | : 1041360146 |
| Lớp | : ĐH KTPM 2 - K10 |

***Hà Nội, 05/2019***

# LỜI NÓI ĐẦU

Ngày nay, công nghệ thông tin đã và đang thay đổi thế giới một cách nhanh chóng. Trong đó Website đã đóng một vai trò quan trọng đối với con người từ giải trí cho tới quảng cáo, thương mại, quản lý,… mang lại nhiều thuận tiện như nhanh hơn, tiện dụng, hiệu quả hơn và không bị giới hạn không gian và thời gian. Trong hoạt động sản xuất, kinh doanh, giờ đây thương mại điện tử đã khẳng định được vai trò xúc tiến và thúc đẩy sự phát triển của doanh nghiệp.

Nhu cầu ăn uống là một nhu cầu thiết yếu không thể thiếu với bất kì cá nhân nào và cũng là một bài toán cho các nhà kinh doanh để đáp ứng tốt nhất cho con người cải thiện đời sống ngày càng cao. Việc đặt đồ ăn online trên website hay các trang mạng xã hội giờ đây đã không còn xa lạ với chúng ta, giúp tiết kiệm thời gian đi lại cho khách hàng và tránh những bất tiện không mong muốn tại quán ăn. Tuy nhiên với những người thức đêm học tập làm việc nhu cầu ăn đêm khá khó khăn trong việc đi lại mua đồ ăn, cũng như không biết mua ở đâu. Bắt nguồn với ý tưởng này, cùng với những gợi ý của thầy giáo Trịnh Bá Quý, em đã chọn đề tài “Nghiên cứu ngôn ngữ NodeJS thư viện ReactJS và xây dựng website bán đồ ăn đêm”.

Để hoàn thành đề tài đồ án tốt nghiệp này, em xin gửi lời cảm ơn chân thành đến các thầy cô giáo Trường Đại Học Công Nghiệp Hà Nội, các thầy cô giáo Khoa Công Nghệ Thông Tin đã tạo điều kiện học tập, truyền đạt những kiến thức quý báu và dìu dắt em trong suốt quá trình học tập tại trường. Bên cạnh đó, em xin cảm ơn những người bạn trong lớp Kỹ thuật phần mềm 2 – K10 đã giúp đỡ em trong suốt quãng đời sinh viên.

Em xin được gửi lời cảm ơn sâu sắc tới thầy Trịnh Bá Quý đã hướng dẫn tận tình cho em trong suốt quá trình học tập để em có thể hoàn tốt đề tài đồ án tốt nghiệp này.

Em xin chân thành cảm ơn.!

# TÓM TẮT ĐỀ TÀI

Với đề tài “Nghiên cứu ngôn ngữ NodeJS thư viện ReactJS và xây dựng website bán đồ ăn đêm”, nhằm xây dựng website thân thiện bán đồ ăn tới người tiêu dùng. Qua đó tìm hiểu nghiệp vụ bán đồ ăn, áp dụng kiến thức đã học phân tích, thiết kế hệ thống, tìm hiểu các kiến thức và công nghệ để thực hiện đề tài như ngôn ngữ NodeJS để lập trình back end, thư viện ReactJS, thư viện MobX để xây dựng front end.

Nội dung chính của đồ án bao gồm:

*Chương 1. Tổng quan*

Nguyên cứu tổng quan nội dung bao quanh đề tài.

*Chương 2. Cơ sở lý thuyết*

Nghiên cứu, tìm hiểu các công nghệ áp dụng trong đề tài.

*Chương 3. Phân tích và thiết kế hệ thống*

Tìm hiểu, phân tích chi tiết về nghiệp vụ của website.

*Chương 4. Kết quả đạt được*

Giao diện website đã đạt được.

*Kết luận*

Những kết quả đã đạt được và những hướng phát triển trong tương lai đối với website.

**MỤC LỤC**

[LỜI NÓI ĐẦU i](#_Toc8595045)

[TÓM TẮT ĐỀ TÀI ii](#_Toc8595046)

[DANH SÁCH HÌNH VẼ vi](#_Toc8595047)

[DANH SÁCH BẢNG BIỂU viii](#_Toc8595048)

[DANH SÁCH CÁC TỪ VIẾT TẮT ix](#_Toc8595049)

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc8595050)

[1.1. Đặt vấn đề 1](#_Toc8595051)

[1.2. Mục đích thực hiện đề tài 1](#_Toc8595052)

[1.3. Ý nghĩa thực tiễn của đề tài 2](#_Toc8595053)

[1.3.1. Ý nghĩa nghiệp vụ 2](#_Toc8595054)

[1.3.2. Ý nghĩa kinh tế 2](#_Toc8595055)

[1.4. Yêu cầu thực hiện đề tài 2](#_Toc8595056)

[1.5. Phương pháp nghiên cứu và hướng giải quyết 2](#_Toc8595057)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 3](#_Toc8595058)

[2.1. NodeJS 3](#_Toc8595059)

[2.1.1. Giới thiệu về NodeJS 3](#_Toc8595060)

[2.1.2. Ưu điểm NodeJS 4](#_Toc8595061)

[2.1.3. Nhược điểm NodeJS 5](#_Toc8595062)

[2.1.4. Ứng dụng của NodeJS 5](#_Toc8595063)

[2.2. Giới thiệu về NoSQL - MongoDB 5](#_Toc8595064)

[2.2.1. NoSQL 5](#_Toc8595065)

[2.2.1.1. Tổng quan NoSQL 5](#_Toc8595066)

[2.2.1.2. So sánh SQL và NoSQL 6](#_Toc8595067)

[2.2.1.3. Phân loại cơ bản các CSDL NoSQL 8](#_Toc8595068)

[2.2.2. MongoDB 9](#_Toc8595069)

[2.2.2.1. Giới thiệu MongoDB 9](#_Toc8595070)

[2.2.2.2. Ưu điểm của MongoDB 11](#_Toc8595071)

[2.2.2.3. Nhược điểm của MongoDB 11](#_Toc8595072)

[2.3. ReactJS 12](#_Toc8595073)

[2.3.1. Giới thiệu về ReactJS 12](#_Toc8595074)

[2.3.2. Đặc điểm ReactJS 13](#_Toc8595075)

[2.4. NextJS Framework 14](#_Toc8595076)

[2.5. Thư viện MobX 15](#_Toc8595077)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 16](#_Toc8595078)

[3.1. Mô tả bài toán 16](#_Toc8595079)

[3.2. Xác định yêu cầu chức năng của hệ thống 17](#_Toc8595080)

[3.3. Phân tích UseCase 18](#_Toc8595081)

[3.3.1. Tác nhân hệ thống 18](#_Toc8595082)

[3.3.2. Các UseCase hệ thống 18](#_Toc8595083)

[3.3.3. Biểu đồ UseCase tổng quát 19](#_Toc8595084)

[3.4. Đặc tả UseCase hệ thống 19](#_Toc8595085)

[3.4.1. UseCase đăng nhập 19](#_Toc8595086)

[3.4.2. UseCase quản lý sản phẩm 20](#_Toc8595087)

[3.4.3. UseCase quản lý người dùng 22](#_Toc8595088)

[3.4.4. UseCase quản lý đơn hàng 24](#_Toc8595089)

[3.4.5. UseCase quản lý giỏ hàng 25](#_Toc8595090)

[3.4.6. UseCase xem chi tiết sản phẩm 27](#_Toc8595091)

[3.4.7. UseCase đặt hàng 28](#_Toc8595092)

[3.4.8. UseCase cập nhật tài khoản 28](#_Toc8595093)

[3.4.9. UseCase tìm kiếm 29](#_Toc8595094)

[3.4.10. UseCase thống kê 30](#_Toc8595095)

[3.5. Biểu đồ lớp 31](#_Toc8595096)

[3.6. Biểu đồ lớp tham gia ca sử dụng 33](#_Toc8595097)

[3.6.1. Quản lý sản phẩm 33](#_Toc8595098)

[3.6.2. Quản lý người dùng 34](#_Toc8595099)

[3.6.3. Quản lý đơn hàng 34](#_Toc8595100)

[3.6.4. Quản lý giỏ hàng 35](#_Toc8595101)

[3.6.5. Xem chi tiết sản phẩm 35](#_Toc8595102)

[3.6.6. Đặt hàng 36](#_Toc8595103)

[3.6.7. Cập nhật tài khoản 37](#_Toc8595104)

[3.6.8. Tìm kiếm 37](#_Toc8595105)

[3.7. Biểu đồ tuần tự 38](#_Toc8595106)

[3.7.1. Quản lý sản phẩm 38](#_Toc8595107)

[3.7.1.1. Thêm sản phẩm 38](#_Toc8595108)

[3.7.1.2. Cập nhật sản phẩm 39](#_Toc8595109)

[3.7.1.3. Xóa sản phẩm 40](#_Toc8595110)

[3.7.2. Quản lý người dùng 41](#_Toc8595111)

[3.7.2.1. Thêm người dùng 41](#_Toc8595112)

[3.7.2.2. Sửa người dùng 42](#_Toc8595113)

[3.7.2.3. Xóa người dùng 43](#_Toc8595114)

[3.7.3. Xem chi tiết sản phẩm 43](#_Toc8595115)

[3.7.4. Quản lý đơn hàng 44](#_Toc8595116)

[3.7.5. Quản lý giỏ hàng 45](#_Toc8595117)

[3.7.5.1. Thêm sản phẩm giỏ hàng 45](#_Toc8595118)

[3.7.5.2. Sửa sản phẩm giỏ hàng 45](#_Toc8595119)

[3.7.5.3. Xóa sản phẩm giỏ hàng 46](#_Toc8595120)

[3.7.6. Cập nhật tài khoản 46](#_Toc8595121)

[3.7.7. Đặt hàng 47](#_Toc8595122)

[3.7.8. Tìm kiếm 47](#_Toc8595123)

[3.8. Biểu đồ hoạt động 48](#_Toc8595124)

[3.8.1. Quản lý sản phẩm 48](#_Toc8595125)

[3.8.2. Quản lý người dùng 48](#_Toc8595126)

[3.8.3. Quản lý đơn hàng 49](#_Toc8595127)

[3.8.4. Quản lý giỏ hàng 49](#_Toc8595128)

[3.8.5. Cập nhật tài khoản 50](#_Toc8595129)

[3.8.6. Đặt hàng 50](#_Toc8595130)

[3.8.7. Tìm kiếm 51](#_Toc8595131)

[3.8.8. Xem chi tiết sản phẩm 51](#_Toc8595132)

[3.9. Biểu đồ lớp chi tiết 52](#_Toc8595133)

[3.10. Thiết kế cơ sở dữ liệu 52](#_Toc8595134)

[CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC 55](#_Toc8595135)

[4.1. Giao diện người dùng 55](#_Toc8595136)

[4.2. Giao diện quản trị 58](#_Toc8595137)

[KẾT LUẬN 61](#_Toc8595138)

[TÀI LIỆU THAM KHẢO 62](#_Toc8595139)

# DANH SÁCH HÌNH VẼ

[Hình 3.1. Biểu đồ UseCase tổng quát 19](#_Toc8594833)

[Hình 3.2. Biểu đồ Usecase đăng nhập 19](#_Toc8594834)

[Hình 3.3. Biểu đồ Usecase quản lý sản phẩm 20](#_Toc8594835)

[Hình 3.4. Biểu đồ Usecase quản lý người dùng 22](#_Toc8594836)

[Hình 3.5. Biểu đồ UseCase quản lý đơn hàng 24](#_Toc8594837)

[Hình 3.6. Biểu đồ UseCase quản lý giỏ hàng 25](#_Toc8594838)

[Hình 3.7. Biểu đồ Usecase xem chi tiết sản phẩm 27](#_Toc8594839)

[Hình 3.8. Biểu đồ lớp tham gia ca sử dụng quản lý sản phẩm 33](#_Toc8594840)

[Hình 3.9. Biểu đồ lớp tham gia ca sử dụng quản lý người dùng 34](#_Toc8594841)

[Hình 3.10. Biểu đồ lớp tham gia ca sử dụng quản lý đơn hàng 34](#_Toc8594842)

[Hình 3.11. Biểu đồ lớp tham gia ca sử dụng quản lý giỏ hàng 35](#_Toc8594843)

[Hình 3.12. Biểu đồ lớp tham gia ca sử dụng xem chi tiết sản phẩm 36](#_Toc8594844)

[Hình 3.13. Biểu đồ lớp tham gia ca sử dụng đặt hàng 36](#_Toc8594845)

[Hình 3.14. Biểu đồ lớp tham gia ca sử dụng cập nhật tài khoản 37](#_Toc8594846)

[Hình 3.15. Biểu đồ lớp tham gia ca sử dụng tìm kiếm sản phẩm 37](#_Toc8594847)

[Hình 3.16. Biểu đồ tuần tự thêm sản phẩm 38](#_Toc8594848)

[Hình 3.17. Biểu đồ tuần tự sửa cập nhật sản phẩm 39](#_Toc8594849)

[Hình 3.18. Biểu đồ tuần tự xóa sản phẩm 40](#_Toc8594850)

[Hình 3.19. Biểu đồ tuần tự thêm người dùng 41](#_Toc8594851)

[Hình 3.20. Biểu đồ tuần tự sửa người dùng 42](#_Toc8594852)

[Hình 3.21. Biểu đồ tuần tự xóa người dùng 43](#_Toc8594853)

[Hình 3.22. Biểu đồ tuần tự xem chi tiết sản phẩm 43](#_Toc8594854)

[Hình 3.23. Biểu đồ tuần tự cập nhật đơn hàng 44](#_Toc8594855)

[Hình 3.24. Biểu đồ tuần tự thêm sản phẩm giỏ hàng 45](#_Toc8594856)

[Hình 3.25. Biểu đồ tuần tự sửa thông tin sản phẩm giỏ hàng 45](#_Toc8594857)

[Hình 3.26. Biểu đồ tuần tự xóa sản phẩm giỏ hàng 46](#_Toc8594858)

[Hình 3.27. Biều đồ tuần tự cập nhật tài khoản 46](#_Toc8594859)

[Hình 3.28. Biều đồ tuần tự đặt hàng 47](#_Toc8594860)

[Hình 3.29. Biểu đồ tuần tự tìm kiếm đồ ăn 47](#_Toc8594861)

[Hình 3.30. Biểu đồ hoạt động quản lý sản phẩm 48](#_Toc8594862)

[Hình 3.31. Biểu đồ hoạt động quản lý người dùng 48](#_Toc8594863)

[Hình 3.32. Biểu đồ hoạt động cập nhật đơn hàng 49](#_Toc8594864)

[Hình 3.33. Biểu đồ hoạt động quản lý giỏ hàng 49](#_Toc8594865)

[Hình 3.34. Biểu đồ hoạt động cập nhật tài khoản 50](#_Toc8594866)

[Hình 3.35. Biểu đồ hoạt động đặt hàng 50](#_Toc8594867)

[Hình 3.36. Biểu đồ hoạt động tìm kiếm sản phẩm 51](#_Toc8594868)

[Hình 3.37. Biểu đồ hoạt động xem chi tiết sản phẩm 51](#_Toc8594869)

[Hình 3.38. Biểu đồ lớp chi tiết 52](#_Toc8594870)

[Hình 3.39. Thiết kế cơ sở dữ liệu 52](#_Toc8594871)

[Hình 4.1. Giao diện trang chủ 55](#_Toc7088933)

[Hình 4.2. Giao diện chi tiết sản phẩm 55](#_Toc7088934)

[Hình 4.3.Giao diện đặt mua hàng 56](#_Toc7088935)

[Hình 4.4. Giao diện đăng nhập 56](#_Toc7088936)

[Hình 4.5. Giao diện tài khoản người dùng 57](#_Toc7088937)

[Hình 4.6. Giao diện lịch sử đơn hàng 57](#_Toc7088938)

[Hình 4.7. Giao diện trang báo cáo thống kê 58](#_Toc7088939)

[Hình 4.8. Giao diện trang quản lý sản phẩm 58](#_Toc7088940)

[Hình 4.9. Giao diện cập nhật chi tiết sản phẩm 59](#_Toc7088941)

[Hình 4.10. Giao diện cập nhật trạng thái đơn hàng 59](#_Toc7088942)

[Hình 4.11. Giao diện quản lý người dùng 60](#_Toc7088943)

# DANH SÁCH BẢNG BIỂU

[Bảng 2.1. So sánh SQL và NoSQL 6](#_Toc8594608)

[Bảng 3.1. Bảng lớp danh mục 31](#_Toc8594609)

[Bảng 3.2. Bảng lớp sản phẩm 31](#_Toc8594610)

[Bảng 3.3. Bảng lớp người dùng 32](#_Toc8594611)

[Bảng 3.4. Bảng lớp đơn hàng 32](#_Toc8594612)

[Bảng 3.5. Bảng danh mục 53](#_Toc8594613)

[Bảng 3.6. Bảng sản phẩm 53](#_Toc8594614)

[Bảng 3.7. Bảng người dùng 54](#_Toc8594615)

[Bảng 3.8. Bảng đơn hàng 54](#_Toc8594616)

# DANH SÁCH CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Đầy đủ** |
| CSDL | Cơ sở dữ liệu |
| UI | User interface |
| DOM | Document Object Model |
| XML | eXtensible Markup Language |
| HTML | HyperText Markup Language |
| API | Application Programming Interface |
| RDBMS | Relational Database Management System |
| DBMS | Database Management System |

# CHƯƠNG 1. TỔNG QUAN

* 1. **Đặt vấn đề**

Theo Decision Lab – công ty cung cấp dịch vụ nghiên cứu thị trường - nhu cầu ăn vặt của giới trẻ khá nhiều, mỗi người tối thiểu một lần trên tuần. Nam giới đến các quán ăn vặt với lý do ‘địa điểm thích hợp để đi ăn cùng bạn bè, đồng nghiệp’ khoảng 54.5%. Còn với nữ giới thì ‘yêu thích các món ăn’ là lý do chính 68.0% . Tuy nhiên nhiều người ngại khi ra ngoài vào tối muộn, không muốn phải chờ đợi gọi đồ, hay có những món muốn ăn mà các quán ở gần không có…. Từ thực tế em có ý tưởng xây dụng website bán đồ ăn đêm nhằm đáp ứng nhu cầu thực tế trên. Với sự bùng nổ của Internet hiện nay, Internet đã trở thành nhu cầu không thể thiếu, có thể thỏa mãn được rất nhiều nhu cầu của con người thì việc đặt mua đồ ăn qua website không còn xa lạ. Website bán đồ ăn đêm nhằm xây dựng thương hiệu, giới thiệu món ăn, thu hút khách hàng biết tới quán ăn và đặt mua đồ ăn trực tuyến. Chủ đạo là những món ăn nhanh bao gồm cả đồ ăn và nước uống mà giới trẻ yêu thích, đặc biệt chú trọng đến đối tượng sinh viên bởi nhu cầu ăn đêm ăn vặt khá lớn. Website phải đem lại trải nghiệm tối ưu tới người dùng từ việc chọn món tới đặt hàng phải nhanh chóng chính xác.

Đồng thời thực hiện việc tin học hóa việc quản lý bán đồ ăn đêm thông qua website tăng tính chuyên nghiệp cũng như thuận lợi, hiệu quả trong quá trình quản lý sản phẩm, đơn hàng, khách hàng và xem thống kê doanh thu.

* 1. **Mục đích thực hiện đề tài**

Khi thực hiện đề tài này mục đích mà em hướng tới là có thể nắm bắt được quá trình xây dựng một sản phẩm hoàn thiện phải trả qua những giai đoạn nào. Tìm hiểu rõ hơn về một ngôn ngữ và các công nghệ liên quan để xây dựng nên một website cụ thể. Các giai đoạn đều có những khó khăn riêng nhưng qua đó em cũng đã tích lũy được những kinh nghiệm kiến thức cho bản thân để có thể áp dụng được vào thực tế.

Website này cũng là sản phẩm thực tế đầu tiên nên để có thể hoàn thiện hơn nữa cũng nhờ sự giúp đỡ rất nhiều từ thầy cô bạn bè, qua đó giúp ích cho em trong việc tiếp thu kiến thức, học hỏi và sắp xếp thời gian phân công công việc cần phải làm một cách khoa học hơn.

Website bán đồ ăn đêm giúp tăng lợi nhuận kinh doanh của quán ăn, tăng tính chuyện nghiệp, mang lại sự trải nghiệm tối ưu nhất tới khách hàng, giúp khách hàng tiết kiệm thời gian đi lại mua hàng, dễ dàng chọn lựa đồ ăn và tránh những bất tiện tại quán ăn.

* 1. **Ý nghĩa thực tiễn của đề tài**
     1. **Ý nghĩa nghiệp vụ**
* Tăng khả năng xử lý: thông tin được xử lý một cách tự động, có thể xử lý đồng thời và cho kết quả nhanh chóng, chính xác.
* Người dùng có thể nhanh chóng đặt hàng và tìm kiếm thông tin mà mình muốn một cách nhanh chóng chính xác.
* Nhân viên có thể nhận thông tin đặt mua đồ ăn của khách hàng nhanh chóng, tiện lợi thông qua hệ thống website, cập nhật tình trạng của các đơn hàng và dễ dàng tổng hợp số liệu đưa ra báo cáo.
* Người quản lý có thể dễ dàng cập nhập các thông tin về các món ăn, khuyến mãi, ưu đãi của món ăn cho khách hàng.
  + 1. **Ý nghĩa kinh tế**
* Tăng đối tượng khách hàng nhờ có hệ thống website đặt đồ ăn trực tuyến, thông tin được xử lý tự động, không phải mất công nhận, nhập giấy tờ.
* Đơn giản hóa quá trình mua hàng, tiết kiệm thời gian cho khách hàng và nhân viên tại quán.
* Giúp người quản lý theo dõi được tình trạng mặt hàng, đơn hàng của quán.
  1. **Yêu cầu thực hiện đề tài**
* Tìm hiểu về ngôn ngữ NodeJS để lập trình backend.
* Tìm hiểu thư viện ReactJS, thư viện MobX để xây dựng frontend.
* Khảo sát, tìm hiểu nghiệp vụ bán đồ ăn đêm và nhu cầu sử dụng website.
* Phân tích, thiết kế hệ thống
* Thiết kế, triển khai xây dựng website bán đồ ăn đêm.
  1. **Phương pháp nghiên cứu và hướng giải quyết**
* Tiến hành thu thập, nghiên cứu các tài liệu có liên quan đến đề tài.
* Khảo sát thực tế.
* Lên ý tưởng, sử dụng kiến thức đã học để phân tích thiết kế hệ thống cho trang ứng dụng.
* Tìm hiểu các công nghệ như NodeJS, ReactJS, Mongodb, …
* Áp dụng công nghệ xây dựng website.
* Tổng hợp lại tất cả kiến thức đã học và hoàn thành báo cáo.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

* 1. **NodeJS**
     1. **Giới thiệu về NodeJS**

NodeJS là một môi trường thực thi JavaScript đa nền tảng và là mã nguồn mở (nguồn nodejs.dev). NodeJS là một **nền tảng (platform).**

JavaScript được sử dụng chủ yếu cho kịch bản phía máy khách, trong đó các tập lệnh được viết bằng JavaScript được nhúng trong HTML của trang web và chạy phía máy khách bởi một công cụ JavaScript trong trình duyệt web của người dùng. NodeJS cho phép các nhà phát triển sử dụng JavaScript viết các công cụ dòng lệnh và kịch bản để tạo nội dung trang web động trước khi trang được gửi đến trình duyệt web của người dùng.

NodeJS được tạo bởi Ryan Dahl từ năm 2009, và phát triển dưới sự bảo trợ của Joyent. Mục tiêu ban đầu của Dahl là làm cho trang web có khả năng push như trong một số ứng dụng web như Gmail. Sau khi thử với vài ngôn ngữ Dahl chọn Javascript vì một API Nhập/Xuất không đầy đủ. Điều này cho phép anh có thể định nghĩa một quy ước Nhập/Xuất điểu khiển theo sự kiện, non-blocking.

NodeJS cho phép tạo các máy chủ Web và các công cụ mạng sử dụng Javascript và một bộ modules xử lý các chức năng cốt lõi khác nhau. Các modules của NodeJS sử dụng API được thiết kế để giảm độ phức tạp của việc viết các ứng dụng máy chủ.

Vài môi trường tương tự được viết trong các ngôn ngữ khác bao gồm Twisted cho Python, Perl, ObjectEnvironment cho Perl, libevent cho C và EventMachine cho Ruby. Khác với hầu hết các chương trình Javascript, NodeJS không chạy trên một trình duyệt mà chạy trên Server. NodeJS sử dụng nhiều chi tiết kỹ thuật của CommonJS. Nó cung cấp một môi trường REPL cho kiểm thử tương tác.

NodeJS được InfoWorld bình chọn là “Công nghệ của năm” năm 2012.

NodeJS sử dụng một kiến trúc module để đơn giản hóa việc tạo ra các ứng dụng phức tạp.

Module giống như các thư viện trong C, hoặc các đơn vị trong Pascal. Mỗi module có chứa một tập hợp các chức năng liên quan đến “đối tượng”, của các module. Ví dụ, các module http chứa các chức năng cụ thể cho HTTP. NodeJS cung cấp một vài module cơ bản để giúp truy cập các tập tin trên hệ thống tập tin, tạo ra trình điều khiển server HTTP và TCP / UDP và thực hiện các chức năng hữu ích khác.

Module được đóng gói từng phần mã. Đoạn mã nằm trong một module chủ yếu là private - có nghĩa là các chức năng và biến được định nghĩa nằm trong đó chỉ có thể truy cập từ bên trong của các module. Tuy nhiên, có thể tiếp xúc với chức năng và / hoặc các biến được sử dụng từ bên ngoài của module. Để làm như vậy, phải sử dụng các đối tượng export với các thuộc tính và phương thức của nó với từng phần mã mà muốn gọi từ bên ngoài.

Khi Node.js cần thực hiện thao tác I/O, như đọc từ mạng, truy cập cơ sở dữ liệu hoặc hệ thống tệp, thay vì chặn luồng và lãng phí chu kỳ CPU chờ, NodeJS sẽ tiếp tục hoạt động khi phản hồi trở lại. Điều này cho phép NodeJS xử lý hàng ngàn kết nối đồng thời với một máy chủ mà không gây ra gánh nặng quản lý đồng thời luồng.. NodeJS có một lợi thế duy nhất bởi vì hàng triệu nhà phát triển frontend có thể viết JavaScript cho trình duyệt hiện có có thể viết code phía máy chủ bên cạnh code phía máy khách mà không cần phải học một ngôn ngữ hoàn toàn khác.

Node là một môi trường chạy JavaScript với google V8 engine do đó hỗ trợ chạy được ở server-side. Do đó nên tuân thủ các kinh nghiệm có trong lập trình với các ứng dụng client-side. Ví dụ khi tạo các biến global trong Node không phải lúc nào cũng có thể tạo. Nhưng có thể tạo dễ dàng các biến hoặc hàm global với cách bỏ từ khóa var trước các biến.

Cộng đồng phát triển NodeJS chủ yếu tập trung ở hai nhóm google : NodeJS và NodeJS-dev, một kênh IRC là # NodeJS trên mạng freenode.

Hiện nay NodeJS được sử dụng bởi nhiều công ty trong đó có Linkedin, Microsoft, Yahoo! và Walmart.

* + 1. **Ưu điểm NodeJS**

Đầu tiên là ưu điểm về tốc độ thực thi và khả năng mở rộng. NodeJS có tốc độ rất nhanh. Đó là một yêu cầu khá quan trọng khi tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web phát triển lên.

Nếu viết một ứng dụng thể hiện trên một trang thì NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request đồng thời thời gian phản hồi nhanh. Các ứng dụng khi không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn rất tuyệt.

Cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn cho các dịch vụ Webs làm bằng JSON.

NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra một luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.

Các web thông thường gửi HTTP request và nhận phản hồi lại (luồng dữ liệu). Giả sử sẽ cần xử lý một luồng dữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.

* + 1. **Nhược điểm NodeJS**

Nếu cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption... hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS. Bởi vì NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm một trình biên dịch của NodeJS sẽ lâu hơn. Trường hợp này viết một Addon C++ để tích hợp với NodeJS sẽ tăng hiệu suất tối đa.

Giống như hầu hết các công nghệ mới, việc triển khai NodeJS trên host không phải là điều dễ dàng. Nếu có một web hosting xài chung, việc tải lên một ứng dụng NodeJS không đơn giản và có thể hoạt động tốt ngay. VPS và dedicated server là một sự lựa chọn tốt hơn - có thể cài đặt NodeJS. Thậm chí dễ hơn là sử dụng một dịch vụ có khả năng mở rộng như là Heroku, và để phát triển trang web trên đó - chỉ cần trả tiền khi cần thêm nhiều tài nguyên hơn.

Một nhược điểm lớn khác của NodeJS đó là NodeJS vẫn đang trong giai đoạn phát triển ban đầu, điều này có nghĩa là một số đặc trưng sẽ thay đổi trong quá trình phát triển tiếp theo. Trong thực tế, theo các tài liệu đi kèm, thì NodeJS bao gồm một chỉ số ổn định (stability index), chỉ số này cho thấy mức độ rủi ro khi sử dụng các đặc trưng hiện có.

* + 1. **Ứng dụng của NodeJS**
* **Websocket server:** Các máy chủ web socket như là Online Chat, Game Server…
* Hệ thống Notification .
* **Fast File Upload Client:** là các chương trình upload file trên client.
* **Ad Server:** Các máy chủ quảng cáo.
* **Cloud Services:** Các dịch vụ đám mây.
* **Any Real-time Data Application** : Các ứng dụng dữ liệu thời gian thực khác.
* **RESTful API:** đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.
  1. **Giới thiệu về NoSQL - MongoDB**
     1. **NoSQL**
        1. ***Tổng quan NoSQL***

Với hầu hết các thời kỳ web, Hệ quản trị cơ sở dữ liệu quan hệ hầu hết dựa trên SQL. Tuy nhiên, thời gian gần đây, một cách tiếp cận mới đã bắt đầu biết đến là NoSQL, tạo ra sự thay thế cho các hệ quản trị cơ sở dữ liệu quan hệ truyền thống.

Cơ sở dữ liệu NoSQL tên gốc là "Non SQL" (phi SQL) hoặc "non relational" (phi quan hệ) cung cấp một cơ chế để lưu trữ và truy xuất dữ liệu được mô hình hóa khác với các quan hệ bảng được sử dụng trong các cơ sở dữ liệu kiểu quan hệ - nguồn Wikipedia .

Các hệ thống NoSQL cũng đôi khi được gọi là "Not only SQL" (không chỉ là SQL) để nhấn mạnh rằng chúng có thể hỗ trợ các ngôn ngữ truy vấn dạng như SQL. NoSQL ám chỉ đến những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu trong lĩnh vực phần mềm. Nguyên nhân cho hướng tiếp cận này bao gồm: tính đơn giản trong thiết kế, mở rộng theo “chiều ngang” cho các điện toán cụm đơn giản hơn (là bài toán cho các cơ sở dữ liệu kiểu quan hệ), và kiểm soát tính khả dụng tốt hơn. Cấu trúc dữ liệu được thiết kế cho các cơ sở dữ liệu NoSQL (ví dụ: key-value, wide column, biểu đồ hoặc tài liệu) khác với cấu trúc dữ liệu được sử dụng mặc định trong các cơ sở dữ liệu quan hệ, khiến cho nó thao tác nhanh hơn trong NoSQL. Cơ sở dữ liệu NoSQL thích hợp với từng trường hợp cụ thể cho vấn đề mà nó phải giải quyết. Đôi khi cấu trúc dữ liệu thiết kế dưới dạng NoSQL được xem là "linh hoạt" hơn các bảng cơ sở dữ liệu kiểu quan hệ.

Thuật ngữ NoSQL được sử dụng bởi Carlo Strozzi vào năm 1998 sử dụng làm tên gọi chung cho các cơ sở dữ liệu quan hệ nguồn mở nhỏ nhưng không sử dụng SQL cho truy vấn. Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL khi Johan Oskarsson của Last.fm muốn tổ chức một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ CSDL mới: một thế hệ CSDL không ràng buộc, phân tán, nguồn mở, khả năng mở rộng theo chiều ngang, có thể lưu trữ, xử lý từ một lượng rất nhỏ cho tới hàng petabytes dữ liệu trong hệ thống có độ chịu tải, chịu lỗi cao với những đòi hỏi về tài nguyên phần cứng thấp.

Dựa trên doanh thu năm 2014, các hãng dẫn đầu thị trường NoSQL là MarkLogic, MongoDB, và Datastax. Dựa trên các bảng xếp hạng phổ biến năm 2015, Các cơ sở dữ liệu NoSQL phổ biến nhất là MongoDB, Apache Cassandra, và Redis.

Một số đặc điểm nhận dạng cho thế hệ CSDL mới này bao gồm: schema- free, hỗ trợ mở rộng dễ dàng, API đơn giản, nhất quán cuối (eventual consistency), không giới hạn không gian dữ liệu,...

* + - 1. ***So sánh SQL và NoSQL***

Bảng 2.1. So sánh SQL và NoSQL

|  | **SQL** | **NoSQL** |
| --- | --- | --- |
| Thiết kế cho | RDBMS truyền thống sử dụng cú pháp và truy vấn SQL để phân tích và lấy dữ liệu để có thêm thông tin chi tiết. Chúng được sử dụng cho các hệ thống xử lý phân tích trực tuyến. | Hệ thống cơ sở dữ liệu NoSQL bao gồm nhiều loại công nghệ cơ sở dữ liệu khác nhau. Các cơ sở dữ liệu này được phát triển để đáp ứng nhu cầu trình bày cho sự phát triển của ứng dụng hiện đại. |
| Type | SQL databases là cơ sở dữ liệu dựa trên bảng | NoSQL databases có thể dựa trên document, key-value, cơ sở dữ liệu biểu đồ |
| Schema | SQL databases có lược đồ được xác định trước | NoSQL databases sử dụng lược đồ động cho dữ liệu phi cấu trúc. |
| Khả năng mở rộng | SQL databases có thể mở rộng theo chiều dọc | NoSQL databases có thể mở rộng theo chiều ngang |
| Phù hợp cho | Đây là lựa chọn lý tưởng cho môi trường truy vấn phức tạp | Không phù hợp với truy vấn phức tạp |
| Lưu trữ dữ liệu phân cấp | SQL databases không thích hợp cho việc lưu trữ dữ liệu phân cấp. | Phù hợp hơn cho kho lưu trữ dữ liệu phân cấp vì có hỗ trợ phương thức key-value. |
| Biến thể | Một loại có biến thể nhỏ | Nhiều loại khác nhau bao gồm các kho key-value, cơ sở dữ liệu tài liệu và cơ sở dữ liệu đồ thị. |
| Open-source | Một sự kết hợp của mã nguồn mở như Postgres & MySQL, và thương mại như Oracle Database. | Open-source |
| Lựa chọn tốt nhất | Khi bạn cần hỗ trợ truy vấn động | Sử dụng khi bạn cần mở rộng quy mô dựa trên yêu cầu thay đổi |
| Tính năng tốt nhất | Hỗ trợ đa nền tảng, bảo mật và miễn phí | Dễ sử dụng, hiệu suất cao và công cụ linh hoạt. |
| Hiệu suất | SQL hoạt động tốt và nhanh thì việc thiết kế tốt là cực kì quan trọng và ngược lại. | Nhanh hơn SQL, NoSQL không chuẩn hóa cho phép lấy được tất cả thông tin về một data cụ thể với các mã hóa mà không cần JOIN liên quan hoặc truy vấn SQL phức tạp. |
| Kết luận | Dự án đã có yêu cầu dữ liệu rõ ràng xác định quan hệ logic có thể được xác định trước. | Phù hợp với những dự án yêu cầu dữ liệu không liên quan, khó xác định, đơn giản mềm dẻo khi đang phát triển |

* + - 1. ***Phân loại cơ bản các CSDL NoSQL***

Dựa trên mô hình dữ liệu:

* *Key-Value Store*

Key-Value Store sử dụng mảng kết hợp (còn được gọi là bản đồ hoặc từ điển) như là mô hình dữ liệu cơ bản của chúng. Trong mô hình này, dữ liệu được biểu diễn như một bộ sưu tập các cặp key-value, như vậy mỗi key có thể xuất hiện chỉ một lần trong bộ sưu tập.

Mô hình key-value là một trong những mô hình dữ liệu đơn giản nhất. Mô hình key-value có thể được mở rộng tới một mô hình ra lệnh rời rạc, duy trì các key trong lệnh từ điển. Phần mở rộng này có khả năng tính toán mạnh mẽ, trong đó nó có thể truy hồi hiệu quả các dãy key chọn lọc.

Một số cơ sở dữ liệu hỗ trợ đặt lệnh của các key. Có nhiều triển khai phần cứng khác nhau, và một số người dùng duy trì dữ liệu trong bộ nhớ (RAM), trong khi những người khác sử dụng ổ SSD hoặc đĩa cứng.

* *Document Store*

Trong khi mỗi document-oriented database thực hiện khác nhau về chi tiết của định nghĩa này, nói chung, tất cả chúng đều giả định rằng các document đóng gói và mã hóa dữ liệu (hoặc thông tin) trong một số định dạng hoặc mã hóa tiêu chuẩn. Mã hóa được sử dụng bao gồm XML, YAML và JSON cũng như các dạng nhị phân như BSON. Các document được định địa chỉ trong cơ sở dữ liệu thông qua một *key* duy nhất đại diện cho document đó. Một trong những đặc điểm định nghĩa khác của một document-oriented database là ngoài việc tra cứu từ khóa được thực hiện bởi một key-value store, cơ sở dữ liệu đó còn cung cấp một API hoặc ngôn ngữ truy vấn để lấy document dựa trên nội dung.

Những triển khai khác nhau cung cấp nhiều cách khác nhau để tổ chức và hoặc nhóm các document:

* Collections - bộ sưu tập
* Tags
* Non-visible metadata
* Directory hierarchies - phân cấp thư mục

So với cơ sở dữ liệu quan hệ, ví dụ, collections có thể được coi là tương tự như các bảng biểu và documents tương tự như các hồ sơ/bản ghi. Nhưng chúng là khác nhau: mỗi bản ghi trong một bảng có cùng một trình tự của các miền, trong khi documents trong collections có thể có các miền hoàn toàn khác nhau.

* *Graph Database*

Loại cơ sở dữ liệu này được thiết kế cho dữ liệu có quan hệ cũng được biểu diễn như một đồ thị bao gồm các yếu tố kết nối qua lại với một số hữu hạn các quan hệ giữa chúng. Loại dữ liệu này có thể là các mối quan hệ xã hội, liên kết giao thông công cộng, bản đồ đường bộ hoặc các topo mạng.

Tuy cùng mang những đặc điểm chung của NoSQL nhưng mỗi CSDL NoSQL cũng có những đặc điểm riêng, và vì thế thường được dùng cho những dự án khác nhau. Ví dụ:

* MongoDB và Redis là những lựa chọn tốt cho việc lưu trữ các dữ liệu thống kê ít được đọc mà lại được viết thường xuyên.
* Hadoop, một CSDL dạng tự do, phân tán làm tốt công việc lưu trữ các dữ liệu lớn như các con số thống kê thời tiết hoặc công việc phân tích nghiệp vụ.
* Memcachedb, một CSDL nhất thời chóng tàn, tuyệt vời trong lưu trữ các phiên làm việc web, các khóa, và các con số thống kê ngắn hạn.
* Cassandra và Riak (các lưu trữ dư thừa, tự động tạo bó cluster) làm tốt trong các môi trường với các ứng dụng có tính sẵn sàng cao, khi thời gian sống tối đa là sống còn.
  + 1. **MongoDB**
       1. ***Giới thiệu MongoDB***

MongoDB là một CSDL có khả năng mở rộng, hiệu suất cao, mã nguồn mở và document-oriented. MongoDB được biết đến là một CSDL đáng tin cậy và dễ dùng. Mongo viết bằng C++. Nó thích hợp cho các ứng dụng tầm trung trở lên. Nếu tỉ lệ lượng dữ liệu ghi vào CSDL của ứng dụng lớn hơn lượng đọc thì đây càng là lựa chọn hợp lý.

* *Một số khái niệm cơ bản của MongoDB:*

1. Document là đơn vị cơ bản của dữ liệu trong MongoDB, nó tương đương với một dòng trong CSDL quan hệ.

Document là một khái niệm quan trọng trong MongoDB. Document bao gồm tập hợp các key với các value tương ứng và có thể chứa nhiều key/value.

Ví dụ: {“greeting” : “Hello, world!”, “foo” : 3}

Document trong MongoDB có cấu trúc tương tự như kiểu dữ liệu JSON, nghĩa là sẽ có các cặp (key => value) nên nó có tính năng động rất lớn. Document có thể hiểu nó giống như các record dữ liệu trong MySQL, tuy nhiên nó có sự khác biệt là các cặp (key => value) có thể không giống nhau ở mỗi document.

*Một số lưu ý:*

* + Các cặp key/ value trong document được sắp xếp. Document trên sẽ khác với document sau

{“foo” : 3, “greeting” : “Hello, world!”}

* + Khóa trong document là một chuỗi
  + MongoDB phân biệt chữ hoa chữ thường
  + Document trong MongoDB không được chứa những khóa giống nhau. Ví dụ document sau là không hợp lệ

{“greeting” : “Hello, world!”, “greeting” : “Hello, MongoDB!”}

1. Collection có thể được coi như tương đương với một bảng.

Collection là một nhóm các document. Nếu document tương đương với dòng trong CSDL quan hệ thì collection tương đương với bảng.

Collection là một Schema-Free, nghĩa là các document có hình dạng khác nhau có thể cùng được lưu trữ trong một collection. Ví dụ các docment sau có thể cùng được lưu trong một collection:

{“greeting” : “Hello, world!”}

{“foo” : 5}

Collection được xác định bởi tên của nó là một chuỗi UTF-8

*Các đặc trưng của MongoDB:*

* Lưu trữ hướng document: Document theo phong cách JSON với những lược đồ động đơn giản.
* Hỗ trợ chỉ mục đầy đủ: chỉ mục trên bất kỳ các thuộc tính
* Tính sao lặp và tính sẵn sàng cao: mở rộng
* Auto-sharding: mở rộng theo chiều ngang mà không ảnh hưởng đến chức năng
* Truy vấn: đa dạng, truy vấn dựa trên văn bản
* Cập nhật nhanh
* Map/Reduce
* GridFS: lưu trữ file với bất kỳ kích cỡ nào mà không làm phức tạp ngăn xếp
* Hỗ trợ thương mại: hỗ trợ doanh nghiệp, đào tào, tư vấn

Với MongoDB, ít phải “chuẩn hóa” hơn so với khi làm việc với lược đồ quan hệ vì trong MongoDB không có khái niệm liên kết (join). Với mỗi đối tượng (object) mức cao nhất, ta sẽ có một collection dữ liệu.

Một collection không phải cho tất cả các lớp (class), thay vào đó, các đối tượng sẽ được nhúng vào đó.

1. MongoDB có thể lưu trữ nhiều CSDL độc lập, mỗi CSDL này có các collection và điều khoản riêng của mình.
2. MongoDB đi kèm với một trình tiện ích JavaScript đơn giản nhưng mạnh mẽ, nó hữu ích trong quản trị và thao tác dữ liệu.
3. Mỗi document có một khóa đặc biệt, đó là “\_id”, nó là duy nhất trong collection của document.
   * + 1. ***Ưu điểm của MongoDB***

Open source: các sản phẩm nguồn mở đưa ra cho những người phát triển với nhiều lợi ích to lớn, đặc biết là việc sử dụng miễn phí. Những lợi ích khác là phần mềm nguồn mở có xu hướng sẽ là tin cậy hơn, an ninh hơn và nhanh hơn để triển khai so với các lựa chọn thay thế sở hữu độc quyền.

Khả năng mở rộng linh hoạt: NoSQL thay thế câu lệnh của các nhà quản trị CSDL về “mở rộng phạm vi” với một thứ mới “mở rộng ra ngoài”. Thay vì bổ sung thêm các máy chủ lớn hơn để điều khiển nhiều tải dữ liệu hơn, thì CSDL NoSQL cho phép một công ty phân tán tải qua nhiều máy chủ khi mà tải gia tăng.

Các CSDL NoSQL khác nhau cho những dự án khác nhau: MongoDB và Redis là những lựa chọn tốt cho việc lưu trữ các dữ liệu thống kê ít được đọc mà lại được viết thường xuyên, như một số đếm truy cập web. Hadoop, một CSDL dạng tự do, phân tán làm tốt công việc lưu trữ các dữ liệu lớn như các con số thống kê thời tiết hoặc công việc phân tích nghiệp vụ.

NoSQL được các hãng lớn sử dụng: Các công ty như Amazon, BBC, Facebook và Google dựa vào các CSDL NoSQL.

NoSQL phù hợp với công nghệ đám mây: NoSQL và đám mây là một sự trùng khớp tự nhiên. Các máy chủ ngày nay là không đắt và có thể dễ dàng mở rộng phạm vi được theo yêu cầu có sử dụng một dịch vụ .

Các CSDL NoSQL hầu hết sử dụng bộ nhớ qua đĩa như là vị trí ghi đầu tiên. Vì thế ngăn ngừa được sự thực thi không ổn định của I/O. Và vì NoSQL lưu trữ dữ liệu thường thúc đẩy được tính mở rộng phạm vi theo chiều ngang thông qua việc ngăn chia, chúng có khả năng tận dụng được việc cung cấp mềm dẻo của đám mây.

* + - 1. ***Nhược điểm của MongoDB***

Hỗ trợ không đồng đều cho các doanh nghiệp: trong khi các nhà cung cấp chủ chốt của các hệ thống quản lý cơ sở dữ liệu quan hệ như SQL Server, Oracle, IBM,… thường đưa ra sự hỗ trợ tốt cho khách hàng thì các nhà cung cấp nguồn mở mới thành lập không thể được mong đợi sẽ cung cấp hỗ tốt hơn.

Chưa đủ hoàn hảo cho các doanh nghiệp: dù chúng đã được triển khai tại một số công ty lớn thì các CSDL NoSQL vẫn đối mặt với một vấn đề về sự tin cậy chính với nhiều doanh nghiệp. Vấn đề lớn của NoSQL là các vấn đề về tính không ổn định, trong khi đó hỗ trợ đầy đủ chức năng và tính ổn định của các hệ thống quản lý cơ sở dữ liệu quan hệ được thiết lập đã từ lâu.

Những hạn chế về tri thức nghiệp vụ: các CSDL NoSQL không có nhiều sự liên kết tới các công cụ BI (Business Intelligence) thường được sử dụng, trong khi những yêu cầu và phân tích hiện đại đơn giản nhất thì cũng liên quan khá nhiều tới sự tinh thông về lập trình.

Thiếu sự tinh thông: tính mới mẻ của NoSQL có nghĩa là không có nhiều lập trình viên và người quản trị biết công nghệ này. Như vậy sẽ rất khó khăn cho các công ty tìm người phù hợp.

Những vấn đề về tính tương thích: không giống như các CSDL quan hệ, các CSDL NoSQL chia sẻ ít theo cách thức của các tiêu chuẩn. Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng API riêng của mình, các giao diện truy vấn riêng,… Sự thiếu hụt các tiêu chuẩn sẽ gây ra rất nhiều khó khăn khi chuyển từ một nhà cung cấp này sang một nhà cung cấp khác nếu có nhu cầu.

* 1. **ReactJS** 
     1. **Giới thiệu về ReactJS**

ReactJS là thư viện Javascript để xây dựng giao diện người dùng. Nó được duy trì bởi Facebook và cộng đồng các nhà phát triển cá nhân và công ty.

 React có thể được sử dụng làm cơ sở trong việc phát triển các single-page application (web application hoặc website) hoặc ứng dụng di động, sự tối ưu cho mục đích sử dụng để lấy dữ liệu thay đổi nhanh chóng. Tuy nhiên, tìm nạp dữ liệu chỉ là khởi đầu của những gì xảy ra trên trang web, đó là lý do tại sao các ứng dụng React phức tạp thường yêu cầu sử dụng các thư viện bổ sung để quản lý trạng thái, định tuyến và tương tác với API.

React được tạo ra bởi Jordan Walke, một kỹ sư phần mềm tại Facebook. Ông chịu ảnh hưởng của XHP, HTML component framework cho PHP. Nó được triển khai lần đầu tiên trên newsfeed của Facebook vào năm 2011 và sau đó trên Instagram.com vào năm 2012. Nó là mở nguồn tại JSConf US vào tháng 5 năm 2013.

Vào ngày 18 tháng 4 năm 2017, Facebook đã công bố React Fiber, một thuật toán cốt lõi mới của thư viện khung React để xây dựng giao diện người dùng. React Fiber đã trở thành nền tảng của bất kỳ cải tiến nào trong tương lai và phát triển tính năng của React framework.

Vào ngày 19 tháng 4 năm 2017, React 360 V1.0.0 đã được phát hành ra công chúng. Điều này cho phép các nhà phát triển có kinh nghiệm sử dụng React để tham gia vào phát triển thực tại ảo.

Một trong những đặc trưng của ReactJS là việc render dữ liệu không những có thể thực hiện ở tầng server mà còn ở tầng client. Render tầng server giúp tối ưu SEO và thời gian tải trang do ReactJS là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về tuy nhiên cần config phức tạp mới có thể áp dụng server-side rendering.

Tích hợp ReactJS vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

ReactJS chỉ là View library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần View. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác.

* + 1. **Đặc điểm ReactJS**
* ReactJS sử dụng cơ chế one-way data binding. Dữ liệu được truyền từ parent component đến child component thông qua props. Luồng dữ liệu đơn giản giúp dễ dàng kiểm soát cũng như sửa lỗi.
* JSX (JavaScript XML) là Javascript mở rộng cho phép viết các mã HTML trong Javascript. Có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
* Virtual DOM – Virtual Document Object Model : React tạo bộ đệm cấu trúc dữ liệu trong bộ nhớ, tính toán sự khác biệt kết quả và sau đó cập nhật DOM hiển thị của trình duyệt một cách hiệu quả. Điều này cho phép lập trình viên viết code như thể toàn bộ trang được hiển thị trên mỗi thay đổi, trong khi các thư viện React chỉ hiển thị các subcomponents thực sự thay đổi. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. ReactJS cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp ReactJS tránh những thao tác cần trên DOM mà nhiều chi phí, tối ưu việc update UI khi có sự thay đổi về dữ liệu.
* Props và State

Dữ liệu trong props có thể được truy cập vào component, nguyên tắc của React là các props không được thay đổi và truyền từ trên xuống dưới. Điều này có nghĩa là một component cha có thể truyền bất cứ dữ liệu nào nó muốn cho con của nó dưới dạng props, nhưng component con không thể sửa đổi props của chính nó.

Mặt khác, State là một đối tượng thuộc sở hữu của component nơi state đó được khai báo. Phạm vi của nó được giới hạn trong component hiện tại. Một component có thể khởi tạo state của nó và cập nhật nó bất cứ khi nào cần thiết. State của component cha thường kết thúc là props của các components con.

* Phương thức vòng đời
* **Constructor() :** được gọi khi component được khởi tạo. Có thể thiết lập các trạng thái ban đầu bên trong component (initial component state) và ràng buộc các class methods.
* **ComponentWillMount() :** được gọi trước phương thức render() , sử dụng để thiết lập các trạng thái ban đầu bên trong component,  không được kích hoạt trong lần hiển thị (render) thứ hai.
* Render() là phương thức vòng đời quan trọng nhất và là phương thức duy nhất cần thiết trong bất kỳ thành phần nào,  trả về các thành phần như một outputs của component.
* ComponentDidMount()đưponentDnhư mentDidMounth phần như một outputs của component cần thiết giúp th mentDidMounth render(), là th)DidMounth phần như một orver lh)DidMounth phần n
* Stateful components: Các states giữ các giá trị trong toàn bộ component và có thể được truyền cho các component con thông qua props.
  1. **NextJS Framework**

NextJSlà một framework cho phép xây dựng ứng dụng ReactJS theo hướng server-side rendering

Server-side rendering là khả năng ứng dụng JavaScript để server sẽ trả về cho browser file HTML của page đã được rendered. Chạy được trên phần lớn mọi trình duyệt, kể cả disable JavaScript vẫn chạy tốt. Giúp tải trang ban đầu nhanh hơn vì toàn bộ dữ liệu đã được xử lý ở server, client chỉ việc hiển thị, hỗ trợ công cụ tìm kiếm có thể thu thập dữ liệu một cách dễ dàng để SEO page tốt hơn.

*Tính năng chính của NextJS:*

* Hot code reloading: NextJSsẽ tự động tải trang khi phát hiện bất cứ thay đổi nào được lưu lại.
* Automatic Routing: Client side routing rất đơn giản, các file, URL được đưa vào hệ thống trong thư mục pages sẽ được tự động map với project mà không cần thêm bất cứ config nào.
* Server Rendering: có thể render React component bằng server side, trước khi gửi HTML cho phía client.
* Ecosystem Compatibility: NextJShoạt động tốt với hệ sinh thái của JavaScript, Node và React.
* Có thể cài đặt với Express hay bất cứ Node.js HTTP server nào.
* Automatic Code Splitting: Trang web sẽ được tạo ra chỉ với những thư viện và code JavaScript cần thiết, không cần phải import những thư viện không cần thiết.
* Dynamic Components: Có thể imoprt các modules và React Component mộc cách tự linh hoạt.
* Static Exports: Sử dụng next export command, NextJScho phép xuất một trong web tĩnh từ ứng dụng.
  1. **Thư viện MobX**

Trong React thì state đó được xử lý bằng this.state và this.setState(). Với những app lớn, vấn đề quản lý state có thể trở nên khó khăn:

* Một component cần phải chia sẻ state với một component khác.
* Một component cần phải thay đổi state của một component khác.
* Đôi khi setState là không đồng bộ.

Vấn đề này càng ngày càng trở nên khó giải quyết khi có quá nhiều các state object cần quản lý và các thay đổi diễn ra chồng chéo nhau khắp các component và rất khó để có thể quản lý tất cả một cách hiệu quả.

Thư viện MobX cung cấp cho công cụ để lưu giữ state, thay đổi state và nhận những thay đổi của state. Chúng tuân theo nguyên tắc single source of truth (chỉ một nơi để lưu trữ toàn bộ state). Nó làm cho việc quản lý state trở nên dễ dàng hơn vì nó được tách rời khỏi các component.

Đoạn code không chỉ ngắn gọn hơn mà MobX còn giải quyết tất cả vấn đề liên quan đế setState. Những thay đổi của state ngay lập tức phản ánh các local component state. Điều này làm cho logic đơn giản và dễ sử dụng.

MobX bởi Michel Weststrate lại chịu ảnh hưởng của lập trình hướng đối tượng, nhưng cũng một phần bởi lập trình phản ứng (Reactive programming). Nó đóng gói state thành những observable. Vì thế dữ liệu có thể chỉ cần có setter và getter, observable có khả năng nhận update khi dữ liệu thay đổi. Và MobX dùng nhiều Store để giữ mọi state.

Không bị rằng buộc khi sử dụng thư viện MobX, có thể chuyển qua sử dụng một thư viện khác bất kỳ lúc nào.

# CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

* 1. **Mô tả bài toán**

Website bán đồ ăn nhanh được xây dựng với mục đích bán đồ ăn đêm trực tuyến.

* Hoạt động chung của website đặt hàng qua mạng Internet
* Khi khách hàng truy cập vào website họ có khả năng xem và tìm kiếm những món ăn mà họ cần theo tên món ăn, theo danh mục món. Khi khách hàng xem một sản phẩm cụ thể, website sẽ đưa ra thông tin chi tiết về sản phẩm, bao gồm mô tả sản phẩm, hình ảnh và giá của sản phẩm. Khách hàng có thể thêm sản phẩm vào giỏ hàng và tiếp tục xem các sản phẩm khác trong website. Nếu khách hàng muốn đặt hàng, khách hàng chọn mua hàng tại giỏ hàng, sau đó khách hàng kiểm tra lại sản phẩm đã chọn tại giỏ hàng và điền thông tin cá nhân hoặc đăng nhập tài khoản đã đăng ký tại hệ thống rồi gửi về hệ thống nhận đơn hàng của quán ăn.
* Sau khi gửi đơn đặt hàng về hệ thống của quán ăn, người quản trị hay nhân viên phụ trách nhận đơn hàng sẽ xác nhận đơn hàng, cập nhật lại trạng thái đơn hàng và giao hàng đến đúng địa chỉ khách hàng đã cung cấp. Khi hàng đã đến nơi khách hàng sẽ nhận hàng rồi thanh toán.
* Đối tượng tương tác với hệ thống

*Khách hàng*: Khách hàng là đối tượng có nhu cầu tìm hiểu và đặt mua các món ăn của quán ăn trực tiếp trên website. Vì vậy trang web cần đáp ứng được những yêu cầu như:

* Hiển thị rõ ràng các món ăn của quán ăn với hình ảnh, giá cả, chi tiết để khách hàng có thể xem và chọn lựa.
* Cung cấp chức năng tìm kiếm nhanh cho khách hàng. Mục đích lớn nhất của khách hàng khi ghé thăm một trang web là tìm kiếm những thứ mình cần do vậy chức năng tìm kiếm là rất quan trọng.
* Sau khi khách hàng chọn lựa xong món ăn mà mình thích thì hệ thống phải cung cấp quá trình làm đơn đặt hàng và gửi thông tin về hệ thống quán ăn.
* Việc đặt hàng phải nhanh chóng, chính xác, hạn chế qua nhiều giai đoạn để có thể đem lại sự trải nghiệm tối ưu nhất cho khách hàng.

*Nhân viên*: Nhân viên sẽ được cấp một username và password. Khi nhân viên đăng nhập thành công vào hệ thống nhân viên có thể làm những công việc như :

* Có thể thực hiện chức năng quản lý sản phẩm (thêm, sửa, xóa) đối với các món ăn thức uống trên trang web.
* Tiếp nhận và xử lý các đơn đặt hàng của khách hàng

*Người quản trị:* Người quản trị quản lý toàn bộ hệ thống trang web, người quản trị sẽ được cấp một username và password. Khi người quản trị đăng nhập thành công vào hệ thống quản trị có quyền thực thi làm những công việc của nhân viên, bên cạnh đó còn có thể quản lý người dùng bao gồm tài khoản khách hàng và tài khoản nhân viên tại hệ thống.

* 1. **Xác định yêu cầu chức năng của hệ thống**
  + *Xem chi tiết đồ ăn*

Sản phẩm phải được hiển thị chi tiết thông tin: tên sản phẩm, giá thành , tình trạng sản phẩm, mô tả nguồn gốc xuất xứ thành phần, …. Đồng thời khách hàng có thể chọn phân loại hàng (với sản phẩm có phân loại hàng ) và số lượng sản phẩm nếu khách hàng có nhu cầu mua sản phẩm đó. Và khách hàng có thể tiến hành mua ngay để đi tới đặt hàng.

* + *Tìm kiếm đồ ăn*

Khi khách hàng tìm kiếm hệ thống phải cung cấp cho khách hàng một danh sách các món ăn theo sự tìm kiếm của khách hàng để họ có thể lựa chọn và đặt món ăn. Thông tin phản hồi phải nhanh chóng chính xác.

* + *Quản lý giỏ hàng*

Trong quá trình xem và mua hàng, khách hàng có thể thêm vào giỏ hàng những sản phẩm dự định sẽ mua. Tại trang giỏ hàng khách hàng có thể thấy danh sách những sản phẩm khách hàng đã lựa chọn. Khách hàng có thể thay đổi số lượng sản phẩm trước khi tiến hành đặt mua hàng, hay xóa những sản phẩm không cần thiết nữa. Nếu như khách hàng không tiến hành đặt hàng thì sản phẩm sẽ được lưu lại tại giỏ hàng.

* + *Đặt hàng*

Khi khách hàng muốn đặt đồ ăn thì cung cấp cho họ form điền thông tin cá nhân để khách hàng có thể nhận đồ ăn đã đặt. Khách hàng có thể kiểm tra trước các sản phẩm trước khi đặt hàng, thay đổi số lượng, loại hàng hay xóa sản phẩm trong giỏ hàng. Việc tiến hành mua hàng phải dễ dàng, đơn giản, hiệu quả, tạo điều kiện tối đa cho khách hàng. Thông tin phản hồi phải nhanh chóng chính xác.

* + *Cập nhật tài khoản*

Đối với khách hàng đã có tài khoản sẽ xem được thông tin tài khoản của mình, cập nhật thông tin, để thuận tiện trong quá trình đặt hàng và có nhiều ưu đãi hơn.

Đối với nhân viên có quyền thay đổi thông tin tài khoản cá nhân trên hệ thống.

Đối với người quản trị có quyền thay đổi thông tin các tài khoản trên hệ thống.

* + *Xem đơn mua*

Khách hàng có tài khoản tại hệ thống có thể theo dõi được các đơn mua hàng đang chờ cửa hàng xử lý, đã nhận, đã hủy hay đang được giao.

* + *Quản lý sản phẩm*

Khi nhân viên hay người quản trị muốn thêm sản phẩm mới hay cập nhập thông tin về sản phẩm thì hệ thống cung cấp cho họ form để có đầy đủ thông tin về sản phẩm để họ có thể cập nhập thông tin chi tiết chính xác nhất.

* + *Quản lý đơn hàng*

Khi nhận được đơn hàng nhân viên thực hiện kiểm tra đơn hàng và xác nhận trạng thái thực hiện đơn hàng trên hệ thống và phải gửi lại phản hồi cho khách hàng một cách nhanh chóng chính xác.

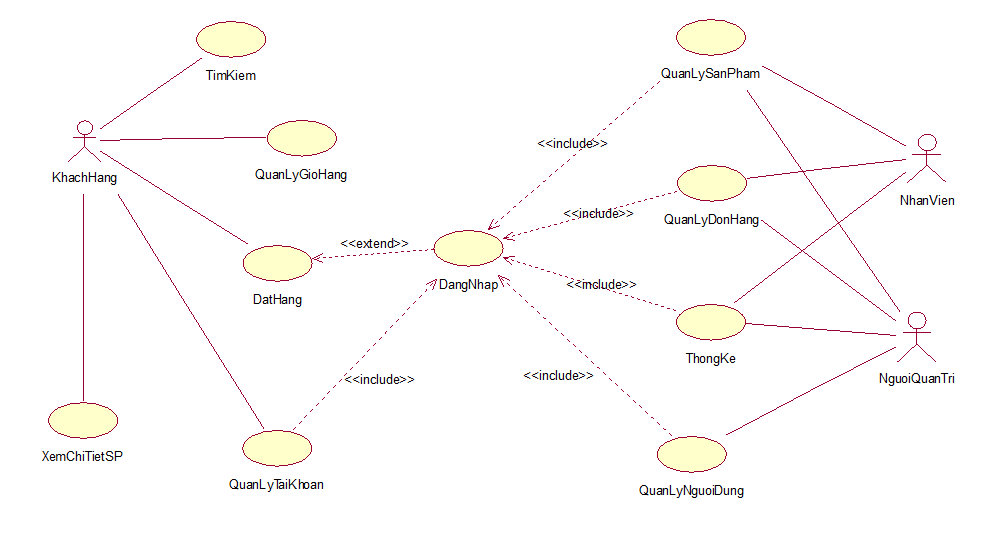
* + *Quản lý người dùng*

Thông tin tài khoản của khách hàng, nhân viên, quản trị sẽ được lưu tại hệ thống. Chỉ người quản trị có quyền thêm, cập nhật, xóa thông tin người dùng hay thay đổi quyền truy cập đối với tài khoản đó.

* + *Thống kê*

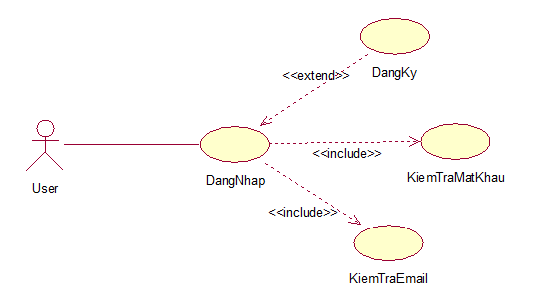
Người quản trị có thể xem thống kê doanh thu, tình trạng đơn hàng, người dùng tại hệ thống. Thông tin phải được cập nhật kịp thời chính xác.

* 1. **Phân tích UseCase**
     1. **Tác nhân hệ thống**
* Khách hàng là tác nhân chính giao tiếp với hệ thống, khách hàng có thể xem, tìm kiếm đồ ăn nước uống và thực hiện mua hàng trực tuyến.
* Nhân viên quản lý các sản phẩm tại hệ thống và tiếp nhận xử lý đơn hàng của khách hàng.
* Người quản trị quản lý toàn bộ hệ thống, bao gồm các quyền thực thi đối với quản lý sản phẩm, quản lý đơn hàng và quản lý tài khoản người dùng.
  + 1. **Các UseCase hệ thống**
* Đăng nhập
* Tìm kiếm
* Xem chi tiết sản phầm
* Quản lý giỏ hàng
* Đặt hàng
* Cập nhật tài khoản
* Quản lý sản phẩm
* Quản lý đơn hàng
* Quản lý người dùng
* Thống kê
  + 1. **Biểu đồ UseCase tổng quát**



Hình 3.1. Biểu đồ UseCase tổng quát

* 1. **Đặc tả UseCase hệ thống**
     1. **UseCase đăng nhập**

****

Hình 3.2. Biểu đồ Usecase đăng nhập

* ***Đặc tả UseCase***

1. Mô tả tóm tắt

Tên ca sử dụng: Đăng nhập

Mục đích: Ca sử dụng này cho phép người dùng đăng nhập hệ thống để sử dùng chức năng được phép của hệ thống với tài khoản đã có trong hệ thống.

Tác nhân: Người dùng (Khách hàng/ Nhân viên/ Quản trị viên)

1. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi người dùng đăng nhập hệ thống.
* Hệ thống hiển thị màn hình đăng nhập hệ thống
* Người dùng nhập thông tin tài khoản email và mật khẩu.
* Hệ thống xác nhận thông tin nhập đúng chưa? Nếu đúng người dùng đăng nhập hệ thống thành công. Chuyển đến màn hình làm việc.
* Ca sử dụng kết thúc
* Luồng sự kiện khác

Luồng sự kiện A1: Nếu thông tin tài khoản đăng nhập mật khẩu sai hay thiếu thông tin, hệ thống yêu cầu người dùng nhập lại thông tin.

Luồng sự kiện A2: Nếu thông tin tài khoản đăng nhập email chưa tồn tại, hệ thống yêu cầu người dùng tiến hành đăng ký để đăng nhập vào hệ thống.

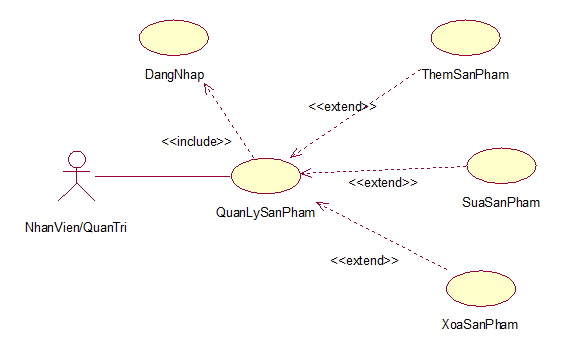
1. Tiền điều kiện

Không

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, dùng đăng nhập thành công vào hệ thống và thực hiện các chức năng tài khoản được phép. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa thay đổi.

* + 1. **UseCase quản lý sản phẩm**



Hình 3.3. Biểu đồ Usecase quản lý sản phẩm

* ***Đặc tả UseCase***

1. Mô tả tóm tắt

Tên ca sử dụng: Quản lý sản phẩm

Mục đích: Ca sử dụng này cho phép người quản trị hay nhân viên quản lý thông tin sản phẩm trong hệ thống thông qua màn hình chính. Người quản trị hay nhân viên có thể thực hiện việc Thêm, Sửa, Xóa thông tin sản phẩm từ hệ thống.

Tác nhân: Người quản trị, Nhân viên

1. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi người quản trị hay nhân viên thực hiện thao tác thêm, sửa, xóa thông tin sản phẩm trong hệ thống.
* Hệ thống hiển thị các lựa chọn: Thêm, Sửa, Xóa
* Hệ thống yêu cầu người quản trị/ nhân viên chọn công việc mà họ muốn thực hiện.
* Nếu người quản trị/ nhân viên chọn “Thêm một sản phẩm”, luồng sự kiện con Thêm một sản phẩm sẽ được thực hiện.
* Nếu người quản trị/ nhân viên chọn “Sửa thông tin sản phẩm”, luồng sự kiện con Sửa thông tin sản phẩm sẽ được thực hiện.
* Nếu người quản trị/ nhân viên chọn “Xoá một sản phẩm”, luồng sự kiện con Xoá một sản phẩm sẽ được thực hiện.
* Kịch bản con: Thêm một sản phẩm
* Hệ thống hiển thị danh sách các sản phẩm và người quản trị/ nhân viên chọn mục thêm sản phẩm.
* Hệ thống hiển thị ra màn hình thao tác với thông tin sản phẩm.
* Người quản trị/ nhân viên nhập thông tin sản phẩm, bao gồm:
* Tên sản phẩm
* Danh mục
* Hình ảnh
* Giá sản phẩm
* Phân loại hàng ( nếu có )
* Trạng thái sản phẩm
* Mô tả sản phẩm

Hệ thống tạo ra mã số tự động (duy nhất) và gán cho mỗi sản phẩm.

* Hệ thống kiểm tra thông tin đã đầy đủ và đúng chưa? Nếu đúng sản phẩm mới được thêm vào hệ thống. Nếu sai hệ thống thông báo yêu cầu nhập lại.
* Ca sử dụng kết thúc.
* Kịch bản con: Sửa đổi thông tin sản phẩm
* Hệ thống hiển thị danh sách sản phẩm.
* Người quản trị/ nhân viên chọn sản phẩm cần sửa.
* Hệ thống hiển thị thông tin sản phẩm vừa chọn.
* Người quản trị/ nhân viên sửa những thông tin cần thiết về sản phẩm và xác nhận việc sửa đổi.
* Hệ thống cập nhật những thay đổi vừa được thực hiện.
* Ca sử dụng kết thúc.
* Kịch bản con: Xoá một sản phẩm
* Hệ thống hiển thị danh sách sản phẩm.
* Người quản trị/ nhân viên chọn sản phẩm cần xóa
* Hệ thống nhắc người quản trị/ nhân viên xác nhận việc xoá sản phẩm.
* Người quản trị/ nhân viên xác nhận việc xoá.
* Hệ thống sẽ tự động giảm số lượng sản phẩm đi một.
* Thông tin sản phẩm đó sẽ bị xóa khỏi hệ thống.
* Ca sử dụng kết thúc.
* Các luồng rẽ nhánh

Không

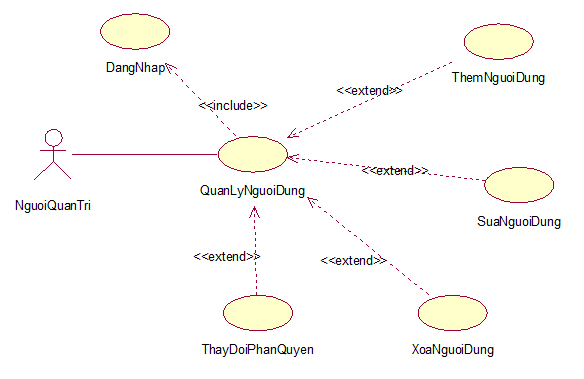
1. Tiền điều kiện

Người quản trị/ nhân viên phải đăng nhập thành công vào hệ thống

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin về sản phẩm sẽ được thêm vào, sửa đổi, hay xoá khỏi hệ thống. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa thay đổi.

* + 1. **UseCase quản lý người dùng**



Hình 3.4. Biểu đồ Usecase quản lý người dùng

* ***Đặc tả UseCase***
  1. Mô tả tóm tắt

Tên ca sử dụng: Quản lý người dùng

Mục đích: Ca sử dụng này cho phép người quản trị quản lý tài khoản người dùng trong hệ thống thông qua màn hình chính. Người quản trị có thể thực hiện việc thêm, sửa, xóa, thay đổi phân quyền tài khoản người dùng từ hệ thống.

Tác nhân: Người quản trị

* 1. Các luồng sự kiện

Luồng sự kiện chính: Ca sử dụng này bắt đầu khi người quản thực hiện thao tác thêm, sửa, xóa, thay đổi phân quyền tài khoản người dùng từ hệ thống.

* Hệ thống hiển thị các lựa chọn: Thêm, Sửa, Xóa, Thay đổi phân quyền
* Hệ thống yêu cầu người quản trị chọn công việc mà họ muốn thực hiện.
* Nếu người quản trị chọn “Thêm một người dùng”, luồng sự kiện con Thêm một người dùng sẽ được thực hiện.
* Nếu người quản trị chọn “Sửa thông tin người dùng”, luồng sự kiện con Sửa thông tin người dùng sẽ được thực hiện.
* Nếu người quản trị/ nhân viên chọn “Xoá một người dùng”, luồng sự kiện con Xoá một người dùng sẽ được thực hiện.
* Kịch bản con: Thêm một người dùng
* Hệ thống hiển thị danh sách người dùng và người quản trị chọn mục thêm người dùng.
* Hệ thống hiển thị ra màn hình thêm người dùng.
* Người quản trị nhập thông tin người dùng, bao gồm:
* Tên người dùng
* Số điện thoại
* Địa chỉ
* Ảnh đại diện
* Email
* Phân quyền
* Mật khẩu

Hệ thống tạo ra mã số tự động (duy nhất) và gán cho mỗi tài khoản người dùng.

* Hệ thống kiểm tra thông tin đã đầy đủ và đúng chưa? Nếu đúng người dùng mới được thêm vào hệ thống. Nếu sai hệ thống thông báo yêu cầu nhập lại.
* Ca sử dụng kết thúc.
* Kịch bản con: Sửa đổi thông tin người dùng
* Hệ thống hiển thị danh sách người dùng.
* Người quản trị chọn tài khoản cần sửa.
* Hệ thống hiển thị thông tin người dùng vừa chọn.
* Người quản trị sửa những thông tin cần thiết về người dùng và xác nhận việc sửa đổi.
* Hệ thống cập nhật những thay đổi vừa được thực hiện.
* Ca sử dụng kết thúc.
* Kịch bản con: Xoá một người dùng
* Hệ thống hiển thị danh sách người dùng.
* Người quản trị chọn tài khoản cần xóa
* Hệ thống nhắc người quản trị xác nhận việc xoá tài khoản người dùng.
* Người quản trị xác nhận việc xoá.
* Hệ thống sẽ tự động giảm số lượng người dùng đi một.
* Thông tin người dùng đó sẽ bị xóa khỏi hệ thống.
* Ca sử dụng kết thúc.
* Kịch bản con: Thay đổi phân quyền một người dùng
* Hệ thống hiển thị danh sách người dùng.
* Người quản trị chọn quyền tài khoản cần thay đổi
* Hệ thống hiển thị các quyền trong hệ thống.
* Người quản trị chọn quyền cần sửa đổi và xác nhận.
* Hệ thống xác nhận thay đổi và lưu lại thông tin thay đổi.
* Ca sử dụng kết thúc.
* Các luồng rẽ nhánh

Luồng A1: Không tiếp tục nhập thông tin người dùng hay xác nhận thay đổi quyền

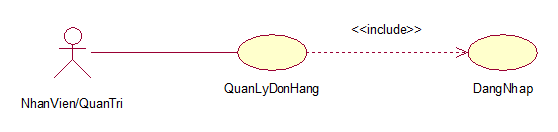
* Ca sử dụng kết thúc
  1. Tiền điều kiện

Người quản trị phải đăng nhập thành công vào hệ thống

* 1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin về người dùng sẽ được thêm vào, sửa đổi, hay xoá khỏi hệ thống. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa thay đổi.

* + 1. **UseCase quản lý đơn hàng**



Hình 3.5. Biểu đồ UseCase quản lý đơn hàng

* ***Đặc tả UseCase***

1. Mô tả tóm tắt

Tên ca sử dụng: Quản lý đơn hàng

Mục đích: Ca sử dụng này cho phép người quản trị hay nhân viên quản lý đơn hàng trong hệ thống thông qua màn hình chính. Từ thông tin đặt hàng của khách hàng người quản trị hay nhân viên có thể thực hiện việc xác nhận đơn hàng, cập nhật thông tin trạng thái đơn hàng trên hệ thống.

Tác nhân: Người quản trị, Nhân viên

1. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi người quản trị hay nhân viên thực hiện thao tác cập nhật thông tin đơn hàng trong hệ thống.
* Hệ thống hiển thị danh sách các đơn hàng và người quản trị/ nhân viên chọn đơn hàng cần xử lý.
* Hệ thống hiển thị ra màn hình thao tác với thông tin đơn hàng đó.
* Người quản trị/ nhân viên cập nhật trạng thái đơn hàng: Xác nhận/ hủy đơn với những đơn hàng đang trong trạng thái chờ xử lý. Hoặc cập trạng thái giao hàng của khách hàng: đang giao, đã nhận, hủy đơn.
* Hệ thống lưu thông tin đơn hàng
* Ca sử dụng kết thúc

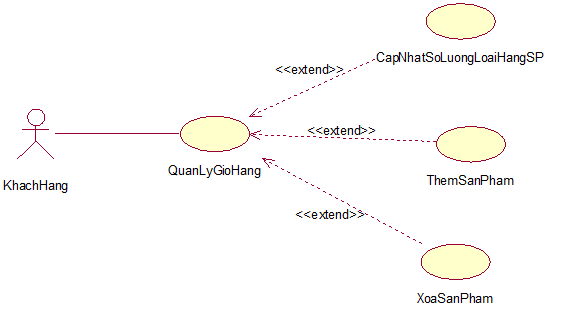
1. Tiền điều kiện

Người quản trị/ nhân viên phải đăng nhập thành công vào hệ thống

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin về đơn hàng sẽ được cập nhật trong hệ thống. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa thay đổi.

* + 1. **UseCase quản lý giỏ hàng**



Hình 3.6. Biểu đồ UseCase quản lý giỏ hàng

* ***Đặc tả UseCase***

1. Mô tả tóm tắt

Tên ca sử dụng: Quản lý giỏ hàng

Mục đích: Ca sử dụng này cho phép khách hàng thêm sản phẩm vào giỏ hàng thông qua website trong quá trình mua hàng. Và sửa hoặc xóa sản phẩm ra khỏi giỏ hàng. Khi người dùng chưa tiến hành mua hàng thì các sản phẩm thêm vào giỏ sẽ được lưu tại giỏ hàng.

Tác nhân: Khách hàng

1. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi khách hàng thực hiện thao tác thêm sản phẩm vào giỏ hàng, thay đổi số lượng, phân loại sản phẩm hoặc xóa sản phẩm trong giỏ hàng.
* Nếu khách hàng chọn thêm sản phẩm vào giỏ hàng thì luồng sự kiện con “Thêm một sản phẩm” giỏ hàng sẽ được thực hiện
* Nếu khách hàng chọn thay đổi số lượng, phân loại sản phẩm thì luồng sự kiện con “Cập nhật sản phẩm giỏ hàng” sẽ được thực hiện
* Nếu khách hàng xóa sản phẩm trong giỏ hàng thì luồng sự kiện con xóa một sản phẩm sẽ được thực hiện
* Kịch bản con: Thêm một sản phẩm giỏ hàng
* Hệ thống hiển thị danh sách các sản phẩm của quán ăn đã được nhập thông tin trên hệ thống quản lý.
* Khách hàng xem chi tiết sản phẩm.
* Hệ thống hiển thị chi tiết về sản phẩm.
* Khách hàng chọn số lượng mua, phân loại hàng ( nếu có ). Và “Thêm vào giỏ hàng”.
* Ca sử dụng kết thúc.
* Kịch bản con: Cập nhật sản phẩm giỏ hàng
* Hệ thống hiển thị danh sách sản phẩm.
* Khách hàng thay đổi số lượng hay phân loại hàng sản phẩm cần cập nhật.
* Hệ thống cập nhật những thay đổi vừa được thực hiện.
* Ca sử dụng kết thúc.
* Kịch bản con: Xóa một sản phẩm giỏ hàng
* Hệ thống hiển thị danh sách sản phẩm.
* Khách hàng chọn sản phẩm cần xóa
* Hệ thống sẽ tự động giảm số lượng sản phẩm đi một.
* Thông tin sản phẩm đó sẽ bị xóa khỏi giỏ hàng
* Ca sử dụng kết thúc.
* Luồng sự kiện khác

Không

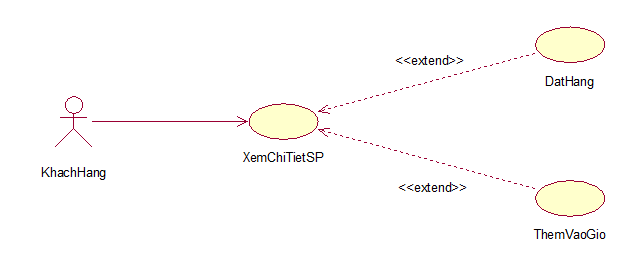
1. Tiền điều kiện

Không

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin về sản phẩm tại giỏ hàng sẽ được thêm vào, sửa đổi, hay xoá khỏi giỏ hàng. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa thay đổi.

* + 1. **UseCase xem chi tiết sản phẩm**



Hình 3.7. Biểu đồ Usecase xem chi tiết sản phẩm

* ***Đặc tả UseCase***

1. Mô tả tóm tắt

Tên ca sử dụng: Xem chi tiết sản phẩm

Mục đích: Ca sử dụng này cho phép khách hàng xem chi tiết từng sản phẩm trên website. Từ đó khách hàng có thể biết thêm được nhiều thông tin hơn của sản phẩm và tiến hành mua hàng.

Tác nhân: Khách hàng

1. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi khách hàng thực hiện thao tác xem thông tin sản phẩm.
* Hệ thống hiển thị danh sách các sản phẩm tại trang chủ.
* Khách hàng chọn vào sản phẩm muốn xem chi tiết thông tin hoặc sản phẩm mà khách hàng muốn đặt hàng.
* Hệ thống hiển thị thông tin sản phẩm và các thông số như tình trạng hàng hiện tại, phân loại hàng ( nếu có ), ô nhập số lượng mua hàng để khách hàng có thể thêm sản phẩm đó vào giỏ hàng hoặc tiến hành mua mặt hàng đó luôn.
* Ca sử dụng kết thúc.
* Luồng sự kiện khác

Không có

1. Tiền điều kiện

Không

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin về sản phẩm mà khách hàng muốn xem sẽ được hiển thị chi tiết các thông tin về mặt hàng để khách hàng có thể lựa chọn mua sản phẩm đó.

* + 1. **UseCase đặt hàng**
* ***Đặc tả UseCase***
  1. Mô tả tóm tắt

Tên ca sử dụng: Đặt hàng

Mục đích: Ca sử dụng này cho phép khách hàng đặt mua hàng thông qua website. Thông tin mua hàng sẽ được gửi tới hệ thống quán ăn.

Tác nhân: Khách hàng

* 1. Các luồng sự kiện
* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi khách hàng thực hiện thao tác thanh toán sản phẩm trong giỏ hàng.
* Hệ thống hiển thị danh sách các sản phẩm khách hàng đã chọn thêm vào giỏ hàng.
* Khách hàng thay đổi các sản phẩm nếu cần, sau đó khách hàng đăng nhập tài khoản hoặc nhập thông tin tên, địa chỉ, số điện thoại để tiến hành đặt hàng. Khách hàng nhập thông tin mua hàng rồi chọn “Mua hàng”.
* Hệ thống xác nhận thông tin nhập đúng chưa? Nếu sai yêu cầu khách hàng nhập lại. Nếu đúng thông báo trạng thái gửi đơn hàng tới khách hàng. Thông tin được gửi tới hệ thống của quán ăn.
* Ca sử dụng kết thúc.
* Luồng sự kiện khác

Luồng A1: Nếu khách hàng chưa tiến hành mua hàng. Hệ thống sẽ lưu lại thông tin sản phẩm đã chọn cho khách hàng.

* 1. Tiền điều kiện

Không

* 1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin về đơn hàng sẽ được gửi tới hệ thống của quán ăn. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa thay đổi.

* + 1. **UseCase cập nhật tài khoản**
* ***Đặc tả UseCase***

1. Mô tả tóm tắt

Tên ca sử dụng: Cập nhật tài khoản

Mục đích: Ca sử dụng này cho phép người dùng cập nhật thông tin tài khoản của mình trên hệ thống.

Tác nhân: Người dùng ( Người quản trị, Nhân viên, Khách hàng)

1. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi người dùng thực hiện thao tác cập nhật tài khoản trong hệ thống.
* Người dùng chọn tài khoản/ hồ sơ.
* Hệ thống hiển thị thông tin tài khoản của người dùng.
* Người dùng cập nhật thông tin tài khoản.
* Hệ thống kiểm tra thông tin cập nhật. Nếu đầy đủ và chính xác hệ thống cập nhật thay đổi
* Ca sử dụng kết thúc.
* Luồng sự kiện khác

Hệ thống kiểm tra thông tin cập nhật. Nếu thiếu thông tin hoặc sai thông tin hệ thống thông báo cập nhật không thành công, yêu cầu người dùng nhập lại.

1. Tiền điều kiện

Người dùng phải đăng nhập thành công vào hệ thống

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin tài khoản được cập nhật thành công. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa thay đổi.

* + 1. **UseCase tìm kiếm**
* ***Đặc tả UseCase***

1. Mô tả tóm tắt

Tên ca sử dụng: Tìm kiếm

Mục đích: Ca sử dụng này cho phép khách hàng tìm kiếm đồ ăn trên website nhanh chóng, dễ dàng nhất.

Tác nhân: Khách hàng

1. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi khách hàng thực hiện thao tác tìm kiếm sản phẩm
* Hệ thống hiển thị danh sách các sản phẩm của quán ăn đã được nhập thông tin trên hệ thống quản lý.
* Khách hàng nhập thông tin cần tìm kiếm và chọn “Tìm kiếm”.
* Nếu từ khoá tìm kiếm hợp lệ hệ thống hiển thị danh sách các sản phẩm có liên quan đến từ khóa tìm kiếm của khách hàng.
* Ca sử dụng kết thúc.
* Luồng sự kiện khác

Nếu khách hàng không đưa ra từ khóa hợp lệ. Khi đó khách hàng có thể điền lại từ khóa, hoặc trở lại trang chủ, khi đó ca sử dụng kết thúc.

1. Tiền điều kiện

Không

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công. Khách hàng có thể biết thêm những sản phẩm ưu thích của khách hàng tại quán ăn.

* + 1. **UseCase thống kê**
* ***Đặc tả UseCase***

1. Mô tả tóm tắt

Tên ca sử dụng: Thống kê

Mục đích: Ca sử dụng này cho phép nhân viên/ người quản trị theo dõi sản phẩm, đơn hàng, người dùng tại hệ thống, doanh thu thống kê theo tháng.

Tác nhân: Nhân viên/ Người quản trị

1. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi nhân viên/ người quản trị thực hiện xem thống kê.
* Hệ thống hiển thị thống kê số lượng sản phẩm, người dùng trên hệ thống, số lượng đơn hàng chưa được xử lý, đang được giao và doanh thu theo theo hiện tại.
* Người dùng chọn tháng muốn xem doanh thu.
* Hệ thống trả về thông số doanh thu theo tháng đã chọn.
* Ca sử dụng kết thúc
* Luồng sự kiện khác

Không

1. Tiền điều kiện

Đã đăng nhập vào hệ thống quản trị.

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công. Nhân viên có thể xem được số lượng đơn hàng chưa được xử lý để kịp thời xử lý và người quản trị có thể xem doanh thu theo từng tháng.

* 1. **Biểu đồ lớp**

1. Lớp danh mục

Bảng 3.1. Bảng lớp danh mục

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| MaDanhMuc | Mã danh mục |
| TenDanhMuc | Tên danh mục |
| Level | Cấp độ danh mục |

1. Lớp sản phẩm

Bảng 3.2. Bảng lớp sản phẩm

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| MaSanPham | Mã sản phẩm |
| TenSanPham | Tên sản phẩm |
| HinhAnh | Ảnh sản phẩm |
| TrangThaiSanPham | Trạng thái sản phẩm |
| PhanLoaiSanPham | Phân loại sản phẩm |
| Gia | Giá sản phẩm |
| MoTa | Mô tả sản phẩm |
| DanhMuc | Danh mục của sản phẩm |
| NgayTao | Ngày tạo sản phẩm |

1. Lớp người dùng

Bảng 3.3. Bảng lớp người dùng

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| MaNguoiDung | Mã người dùng |
| TenNguoiDung | Tên người dùng |
| AnhDaiDien | Ảnh đại diện |
| SoDienThoai | Số điện thoại |
| Email | Email người dùng |
| MatKhau | Mật khẩu tài khoản |
| DiaChi | Địa chỉ người dùng |
| PhanQuyen | Phân quyền tài khoản |
| SoLanMuaHang | Số lần mua hàng |

1. Lớp đơn hàng

Bảng 3.4. Bảng lớp đơn hàng

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| MaDonHang | Mã đơn hàng |
| SanPham | Danh sách sản phẩm trong đơn hàng |
| TrangThaiDonHang | Trạng thái đơn hàng |
| TrangThaiGiaoHang | Trạng thái giao hàng |
| GhiChuDonHang | Ghi chú đơn hàng của khách hàng |
| MaNguoiDung | Mã khách hang nếu có tài khoản trong hệ thống |
| ThongTinNguoiNhan | Thông tin người nhận hàng |
| NgayTao | Ngày tạo đơn hàng |

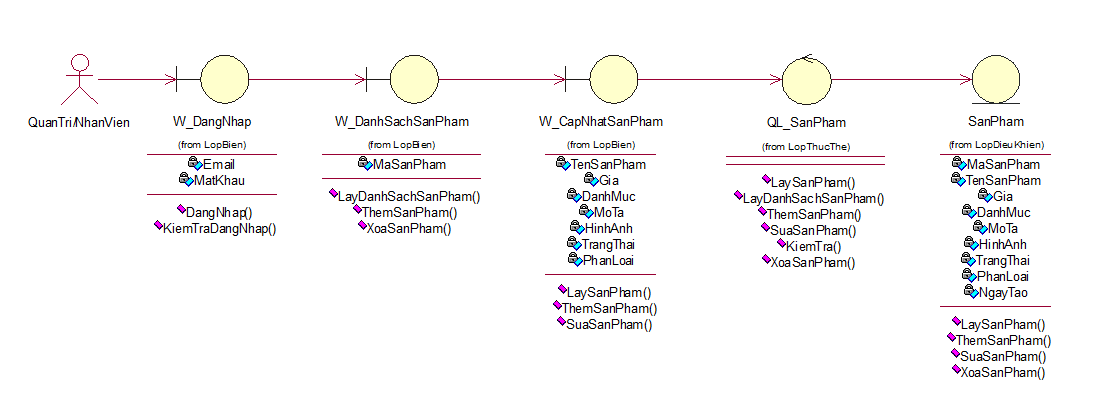
* 1. **Biểu đồ lớp tham gia ca sử dụng**
     1. **Quản lý sản phẩm**

*Lớp tham gia Usecase*

Các lớp biên gồm: W\_DangNhap, W\_DanhSachSanPham, W\_CapNhatSanPham.

Lớp điều khiển: QL\_SanPham .

Lớp thực thể: SanPham.



Hình 3.8. Biểu đồ lớp tham gia ca sử dụng quản lý sản phẩm

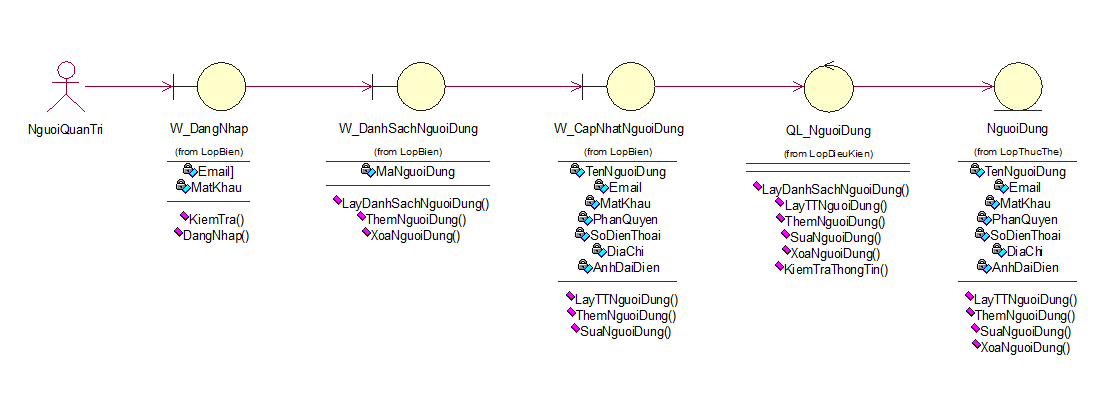
* + 1. Quản lý người dùng

*Lớp tham gia Usecase*

Các lớp biên gồm: W\_DangNhap, W\_DanhSachNguoiDung, W\_CapNhatNguoiDung.

Lớp điều khiển: QL\_NguoiDung.

Lớp thực thể: NguoiDung.



Hình 3.9. Biểu đồ lớp tham gia ca sử dụng quản lý người dùng

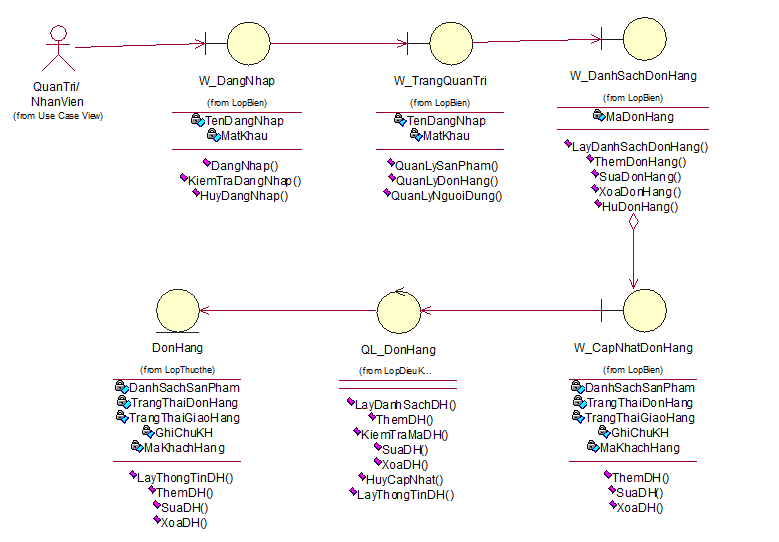
* + 1. **Quản lý đơn hàng**

*Lớp tham gia Usecase*

Các lớp biên gồm: W\_DangNhap, W\_TrangQuanTri, W\_DanhSachDonHang, W\_CapNhatDonHang.

Lớp điều khiển: QL\_DonHang .

Lớp thực thể: DonHang.

******

Hình 3.10. Biểu đồ lớp tham gia ca sử dụng quản lý đơn hàng

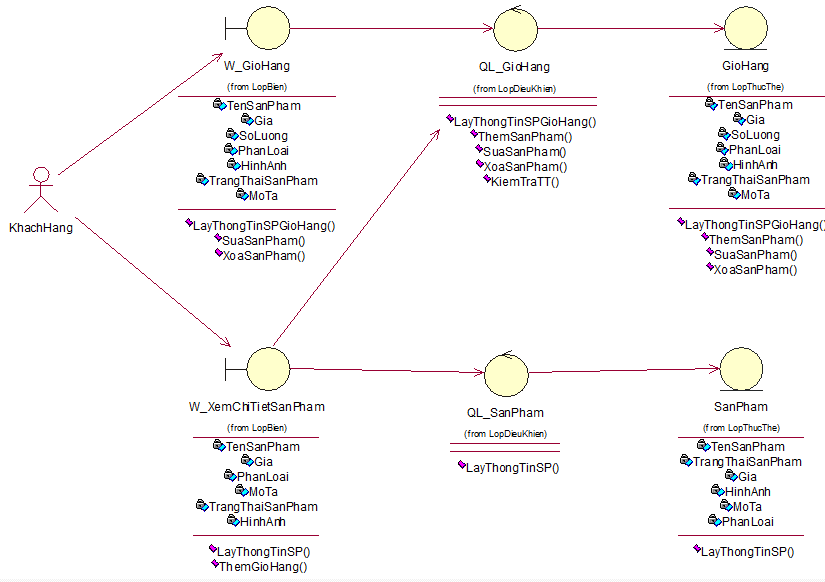
* + 1. **Quản lý giỏ hàng**

*Lớp tham gia Usecase*

Các lớp biên gồm: W\_XemChiTietSanPham, W\_GioHang

Lớp điều khiển: QL\_SanPham, QL\_GioHang.

Lớp thực thể: SanPham, GioHang.



Hình 3.11. Biểu đồ lớp tham gia ca sử dụng quản lý giỏ hàng

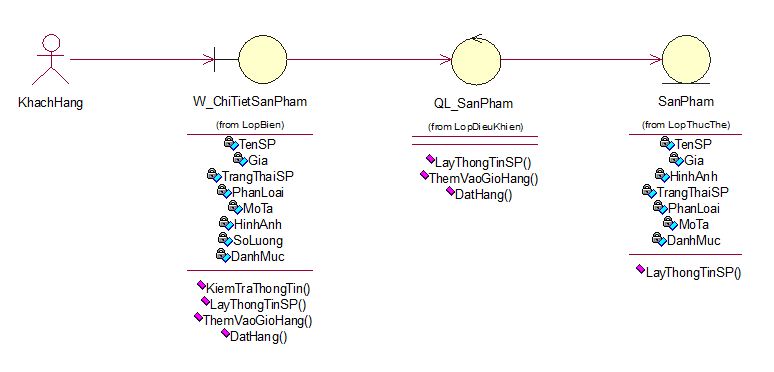
* + 1. Xem chi tiết sản phẩm

*Lớp tham gia Usecase*

Các lớp biên gồm: W\_ChiTietSanPham.

Lớp điều khiển: QL\_SanPham .

Lớp thực thể: SanPham.



Hình 3.12. Biểu đồ lớp tham gia ca sử dụng xem chi tiết sản phẩm

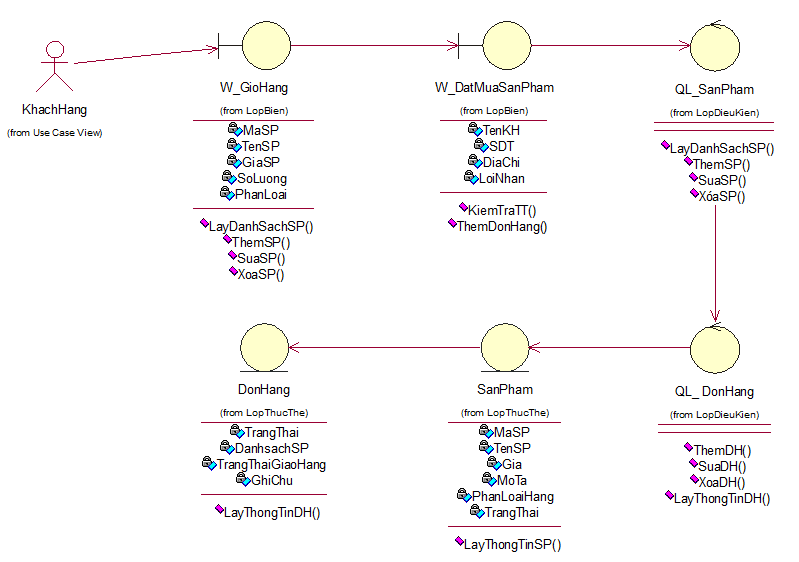
* + 1. Đặt hàng

*Lớp tham gia Usecase*

Các lớp biên gồm: W\_GioHang, W\_DatMuaSanPham.

Lớp điều khiển: QL\_ DonHang, QL\_SanPham .

Lớp thực thể: SanPham, DonHang.



Hình 3.13. Biểu đồ lớp tham gia ca sử dụng đặt hàng

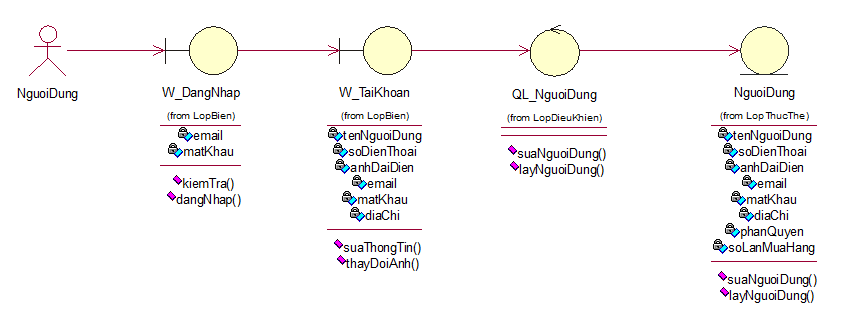
* + 1. Cập nhật tài khoản

*Lớp tham gia Usecase*

Các lớp biên gồm: W\_DangNhap, W\_TaiKhoan.

Lớp điều khiển: QL\_NguoiDung.

Lớp thực thể: NguoiDung.



Hình 3.14. Biểu đồ lớp tham gia ca sử dụng cập nhật tài khoản

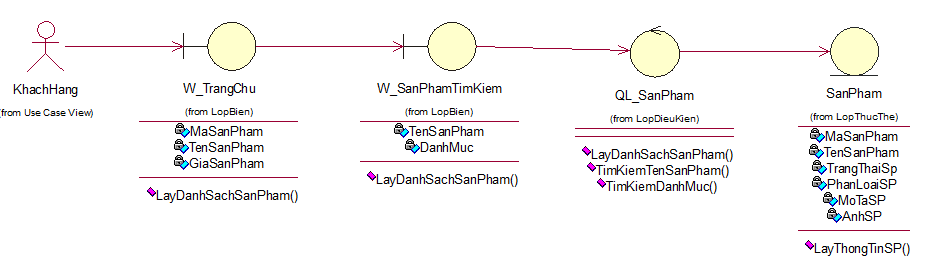
* + 1. Tìm kiếm

*Lớp tham gia Usecase*

Các lớp biên gồm: W\_TrangChu, W\_SanPhamTimKiem.

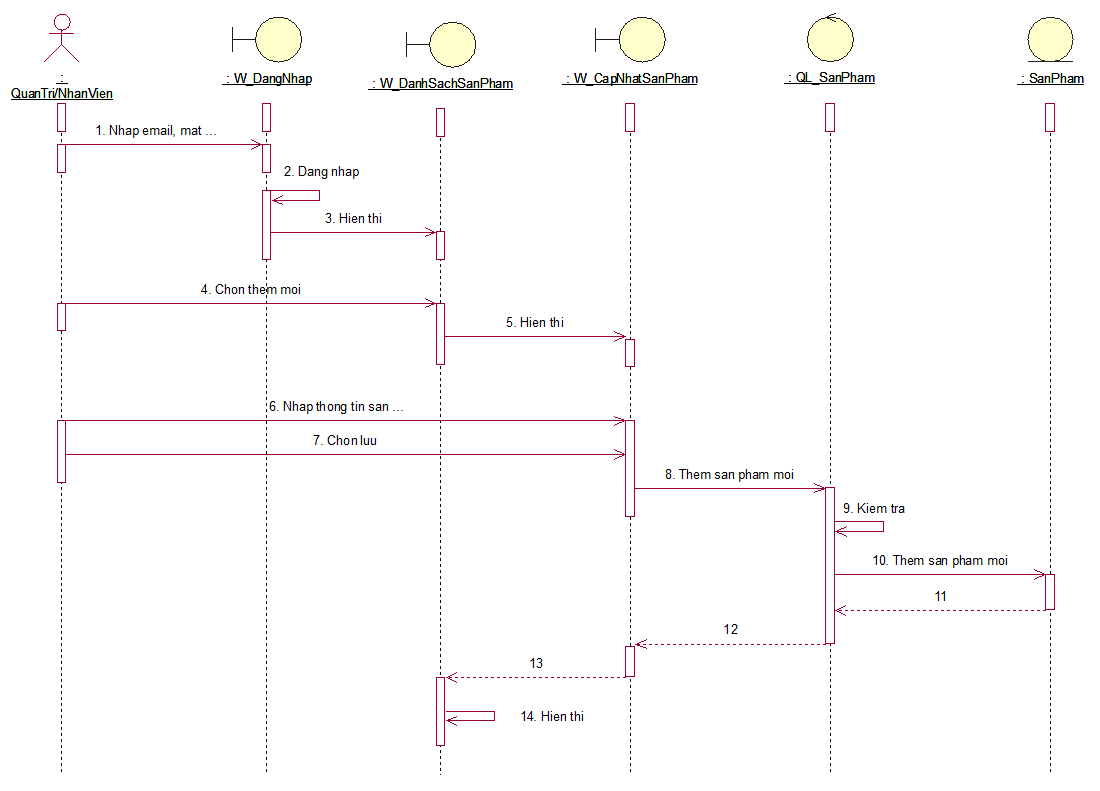
Lớp điều khiển: QL\_SanPham .

Lớp thực thể: SanPham.



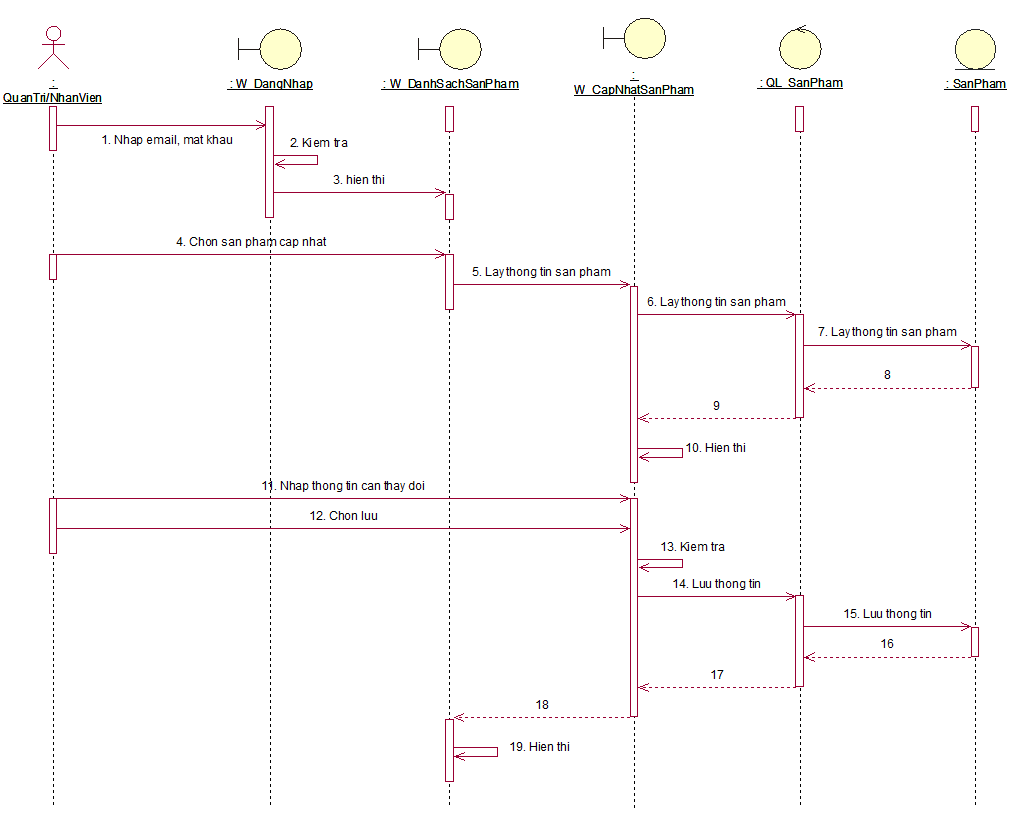
Hình 3.15. Biểu đồ lớp tham gia ca sử dụng tìm kiếm sản phẩm

* 1. **Biểu đồ tuần tự**
     1. **Quản lý sản phẩm**
        1. ***Thêm sản phẩm***



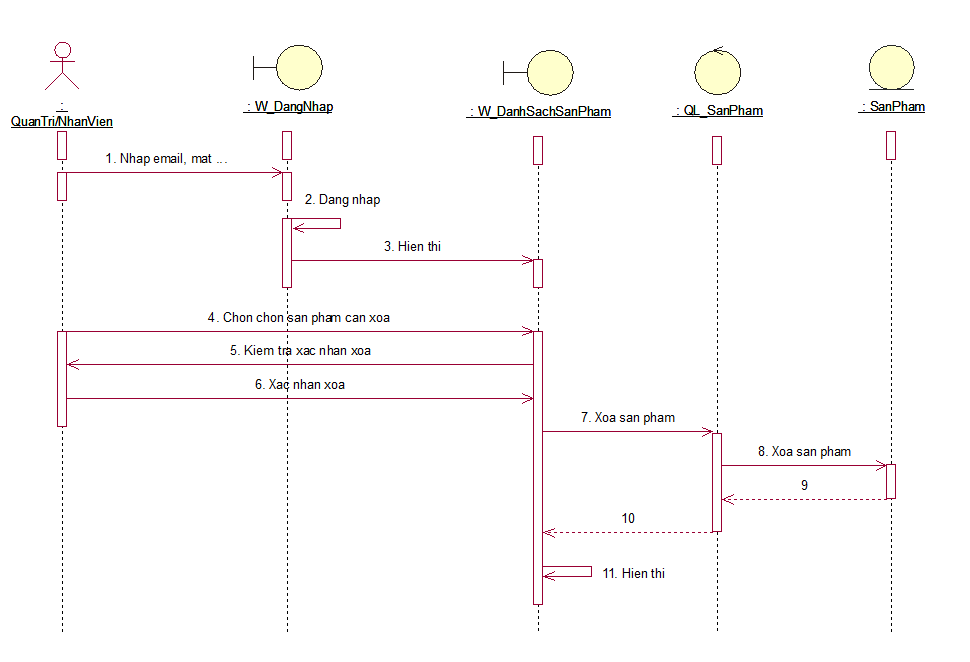
Hình 3.16. Biểu đồ tuần tự thêm sản phẩm

* + - 1. ***Cập nhật sản phẩm***



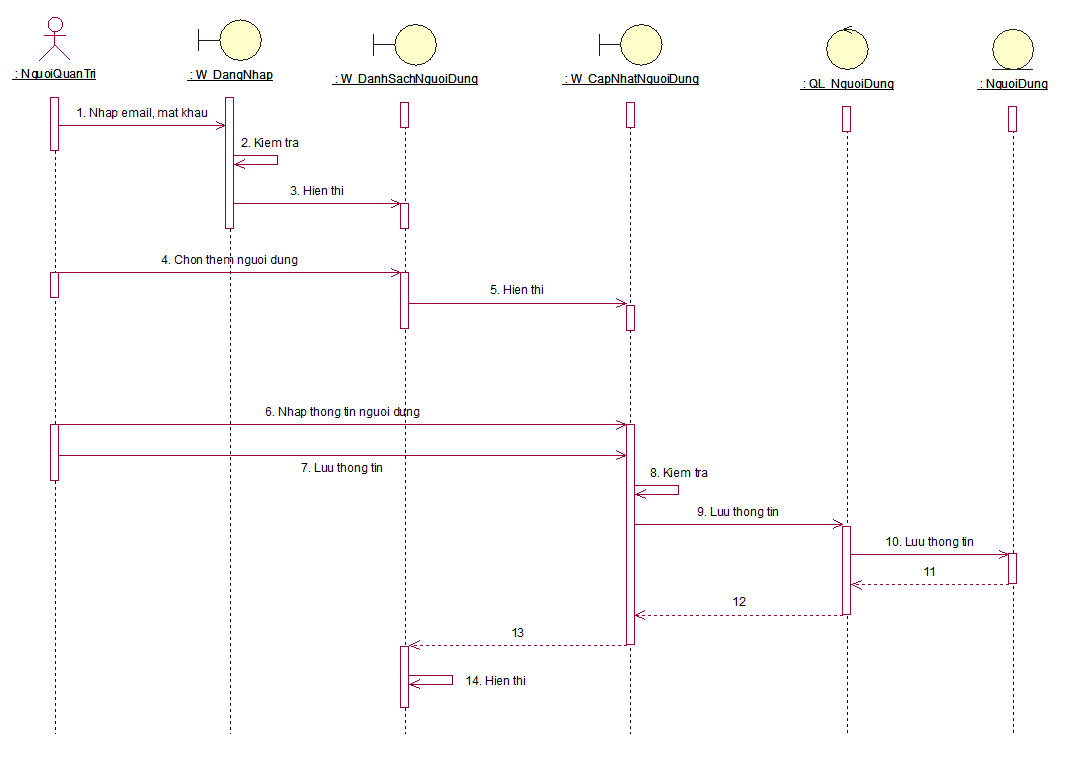
Hình 3.17. Biểu đồ tuần tự sửa cập nhật sản phẩm

* + - 1. *Xóa sản phẩm*



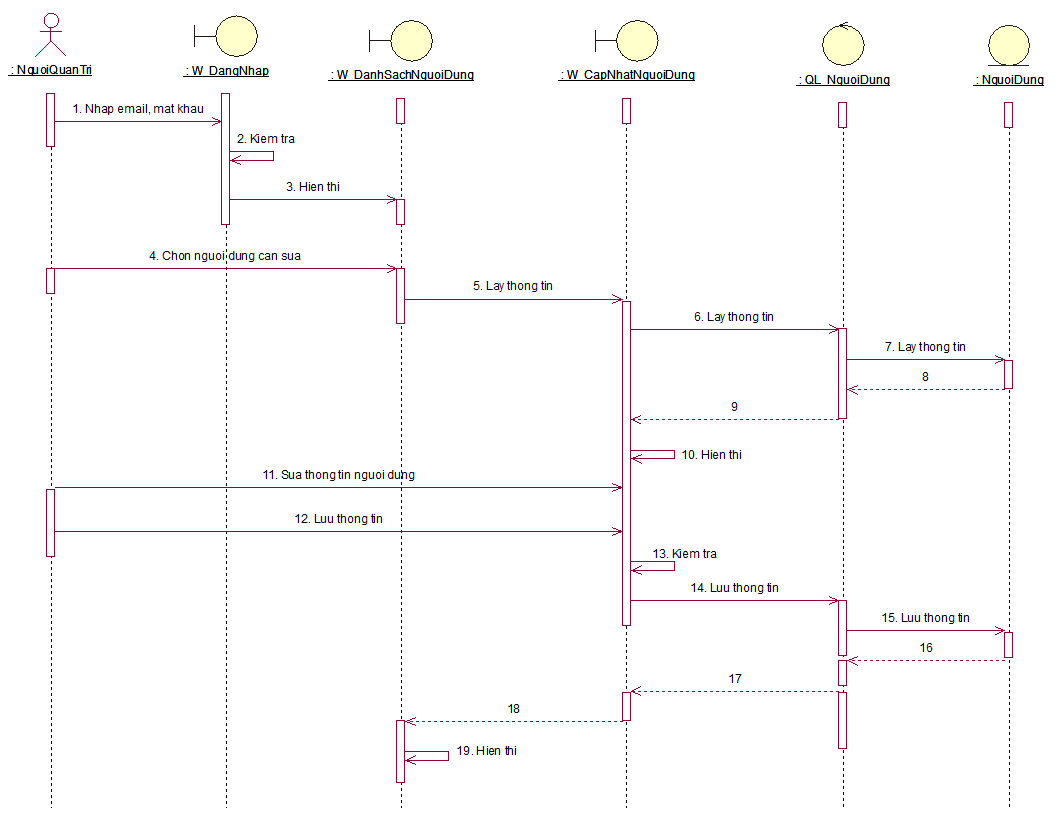
Hình 3.18. Biểu đồ tuần tự xóa sản phẩm

* + 1. **Quản lý người dùng**
       1. ***Thêm người dùng***



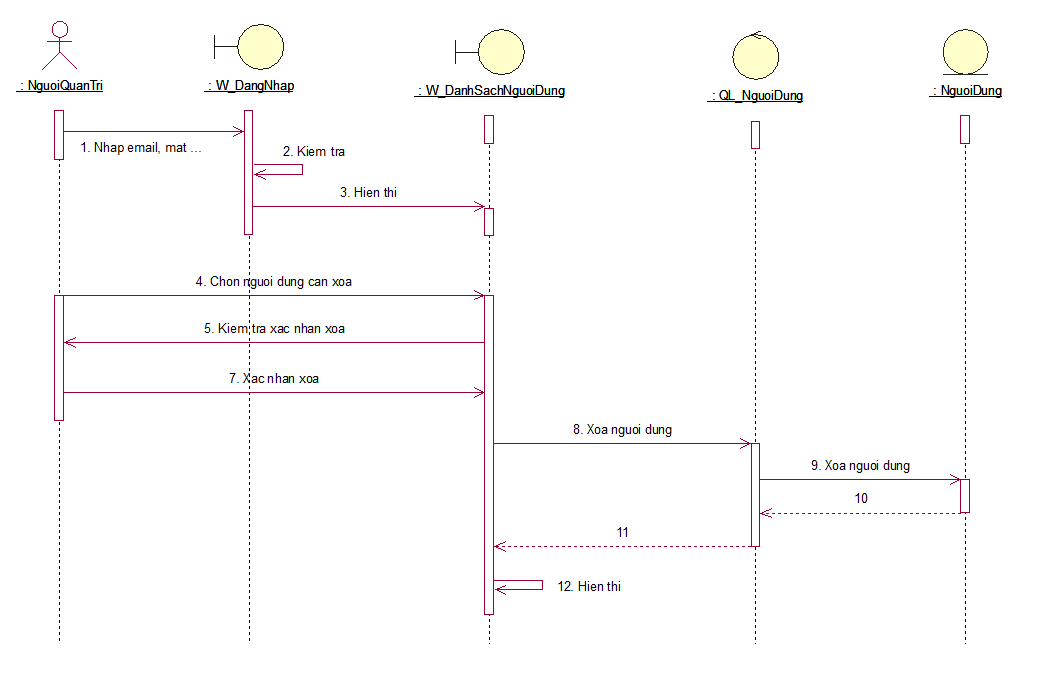
Hình 3.19. Biểu đồ tuần tự thêm người dùng

* + - 1. ***Sửa người dùng***



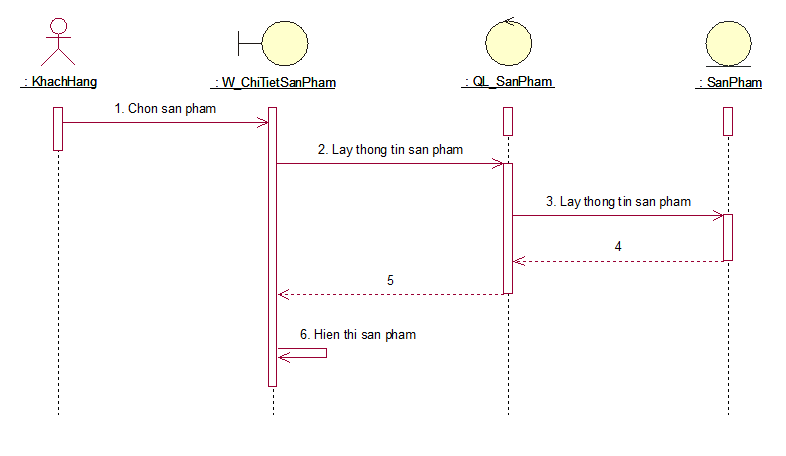
Hình 3.20. Biểu đồ tuần tự sửa người dùng

* + - 1. ***Xóa người dùng***



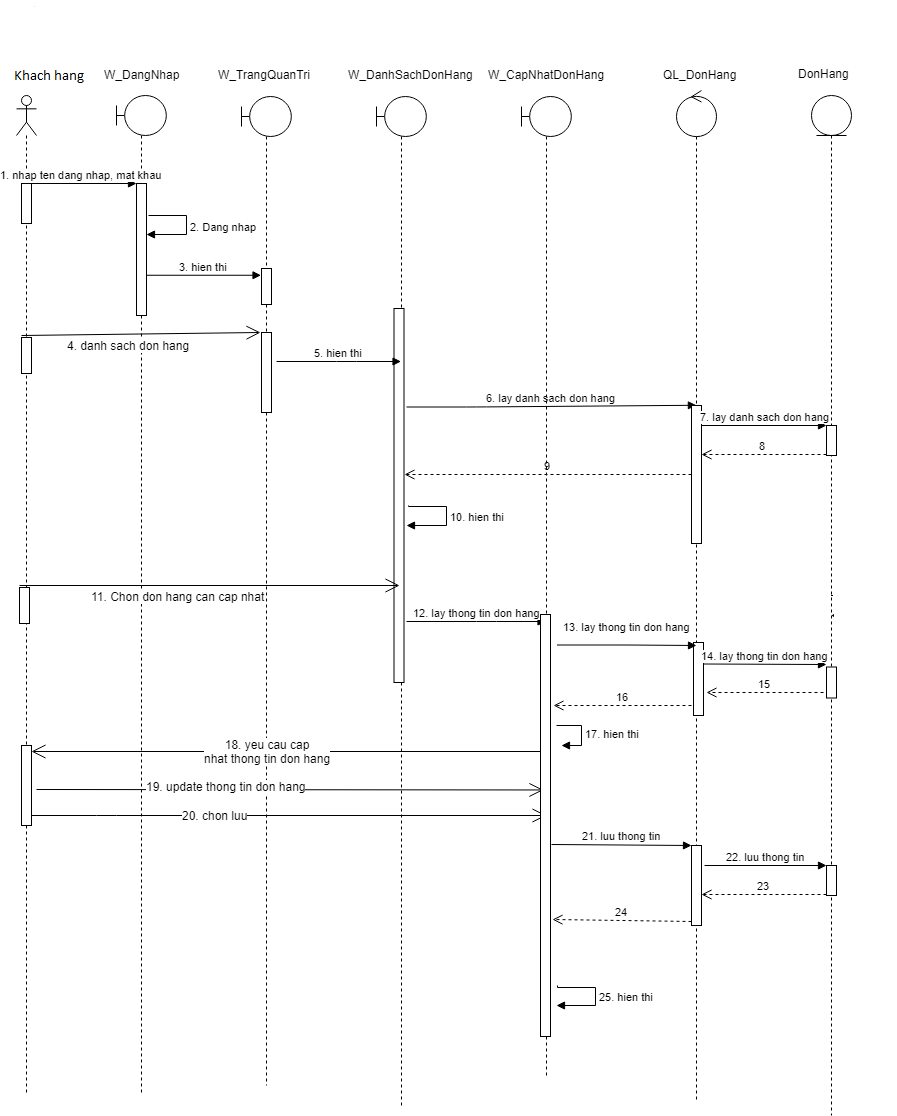
Hình 3.21. Biểu đồ tuần tự xóa người dùng

* + 1. Xem chi tiết sản phẩm



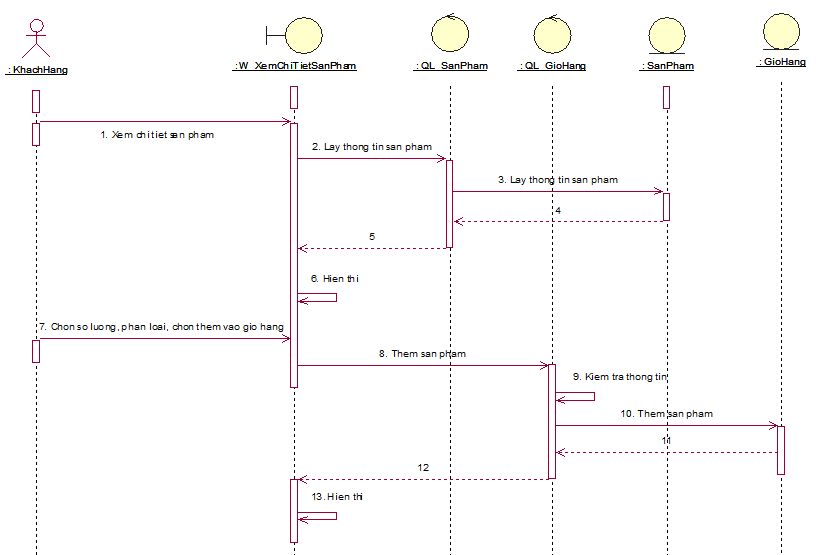
Hình 3.22. Biểu đồ tuần tự xem chi tiết sản phẩm

* + 1. **Quản lý đơn hàng**



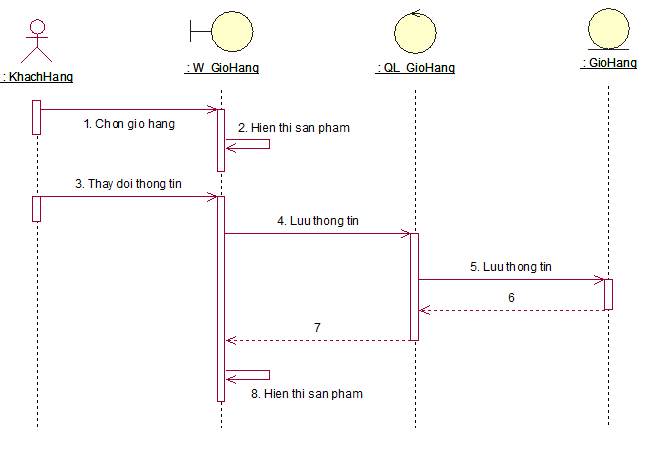
Hình 3.23. Biểu đồ tuần tự cập nhật đơn hàng

* + 1. **Quản lý giỏ hàng**
       1. ***Thêm sản phẩm giỏ hàng***



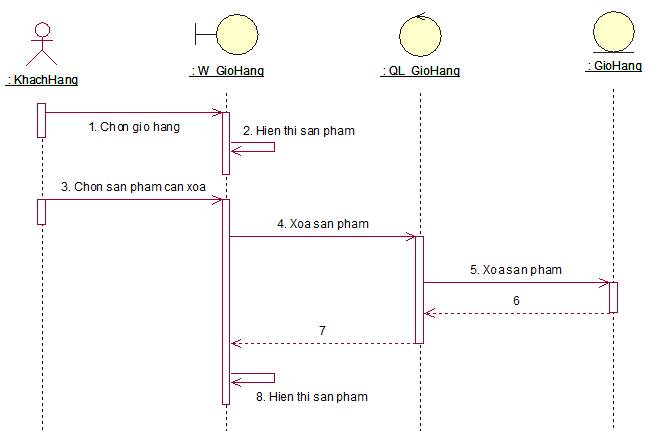
Hình 3.24. Biểu đồ tuần tự thêm sản phẩm giỏ hàng

* + - 1. *Sửa sản phẩm giỏ hàng*



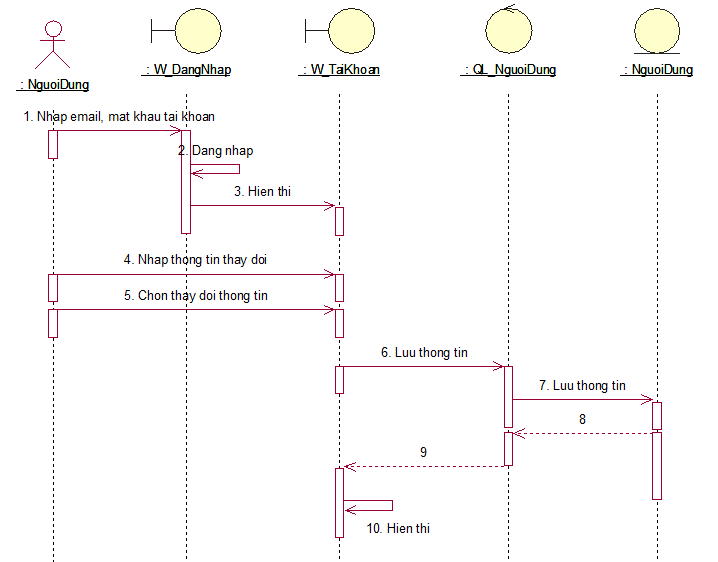
Hình 3.25. Biểu đồ tuần tự sửa thông tin sản phẩm giỏ hàng

* + - 1. *Xóa sản phẩm giỏ hàng*



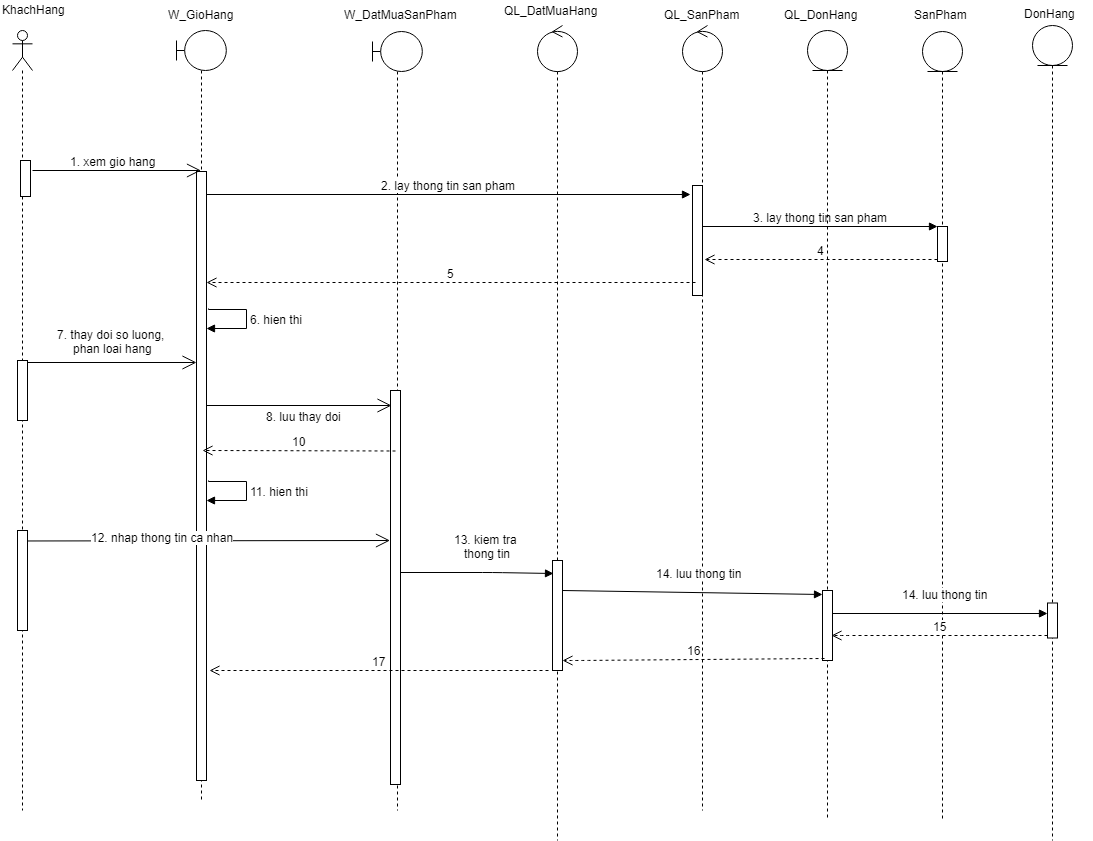
Hình 3.26. Biểu đồ tuần tự xóa sản phẩm giỏ hàng

* + 1. **Cập nhật tài khoản**



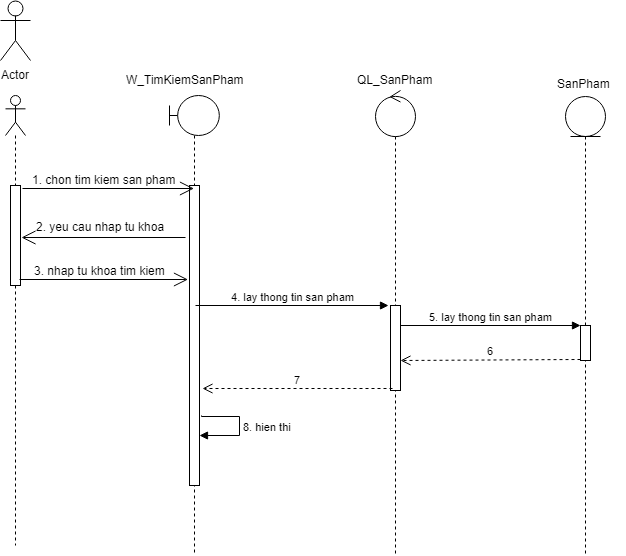
Hình 3.27. Biều đồ tuần tự cập nhật tài khoản

* + 1. **Đặt hàng**



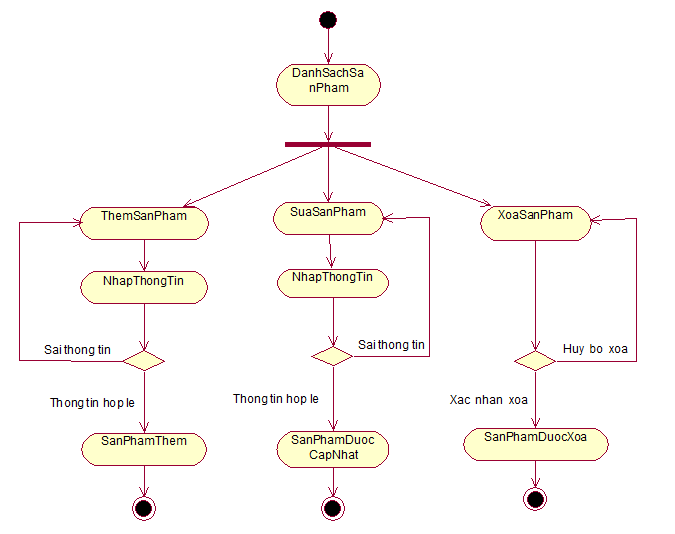
Hình 3.28. Biều đồ tuần tự đặt hàng

* + 1. **Tìm kiếm**



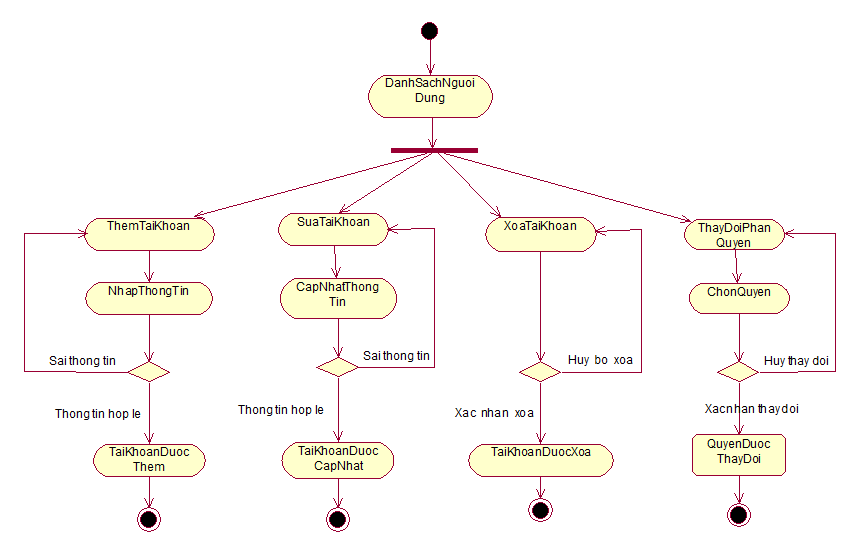
Hình 3.29. Biểu đồ tuần tự tìm kiếm đồ ăn

* 1. **Biểu đồ hoạt động**
     1. **Quản lý sản phẩm**



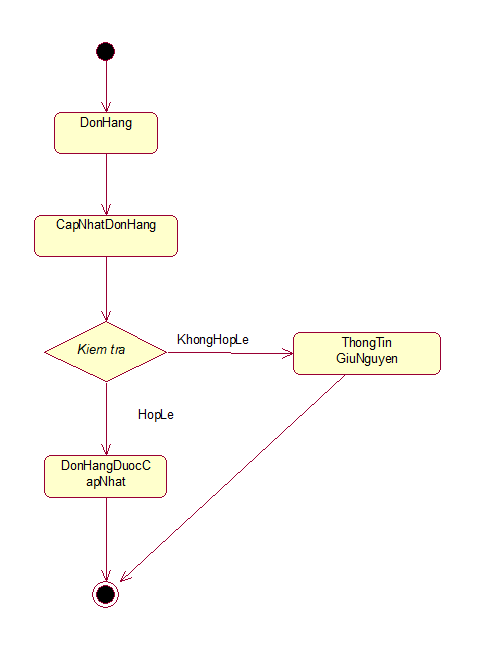
Hình 3.30. Biểu đồ hoạt động quản lý sản phẩm

* + 1. **Quản lý người dùng**



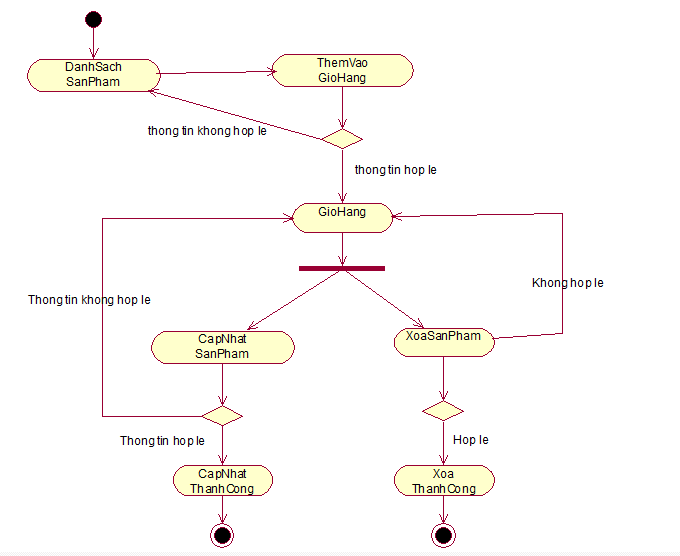
Hình 3.31. Biểu đồ hoạt động quản lý người dùng

* + 1. **Quản lý đơn hàng**



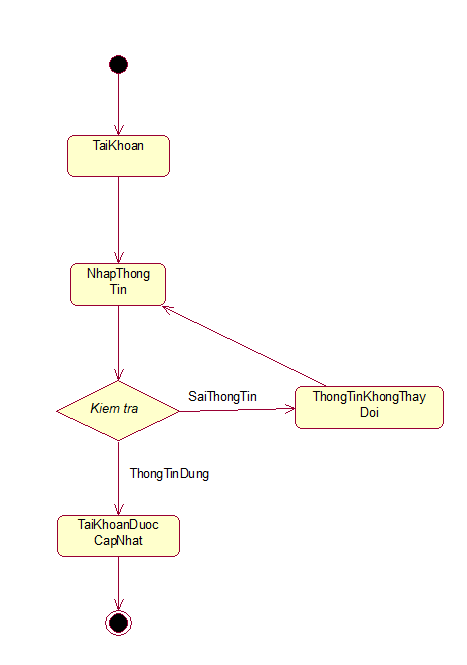
Hình 3.32. Biểu đồ hoạt động cập nhật đơn hàng

* + 1. **Quản lý giỏ hàng**



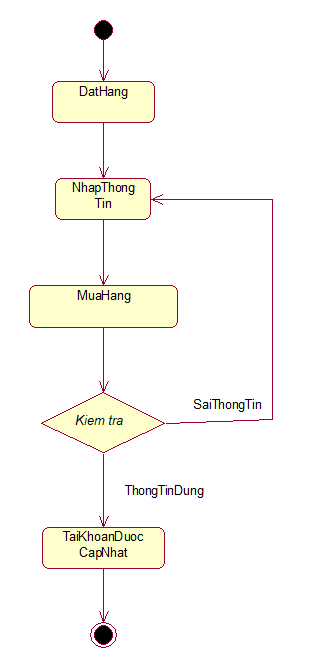
Hình 3.33. Biểu đồ hoạt động quản lý giỏ hàng

* + 1. **Cập nhật tài khoản**



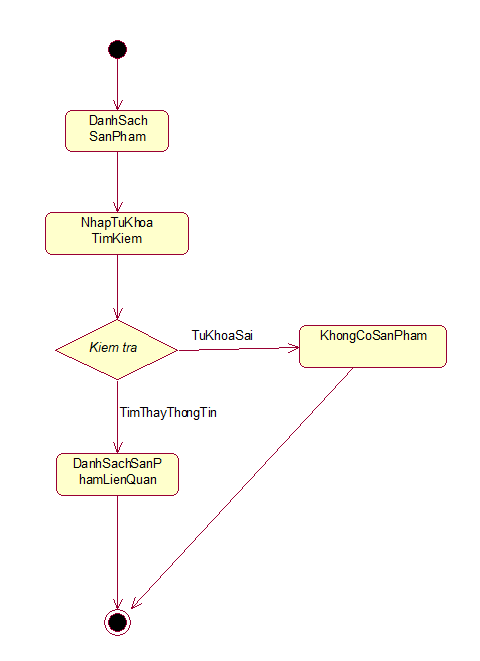
Hình 3.34. Biểu đồ hoạt động cập nhật tài khoản

* + 1. **Đặt hàng**



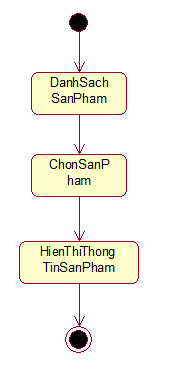
Hình 3.35. Biểu đồ hoạt động đặt hàng

* + 1. **Tìm kiếm**



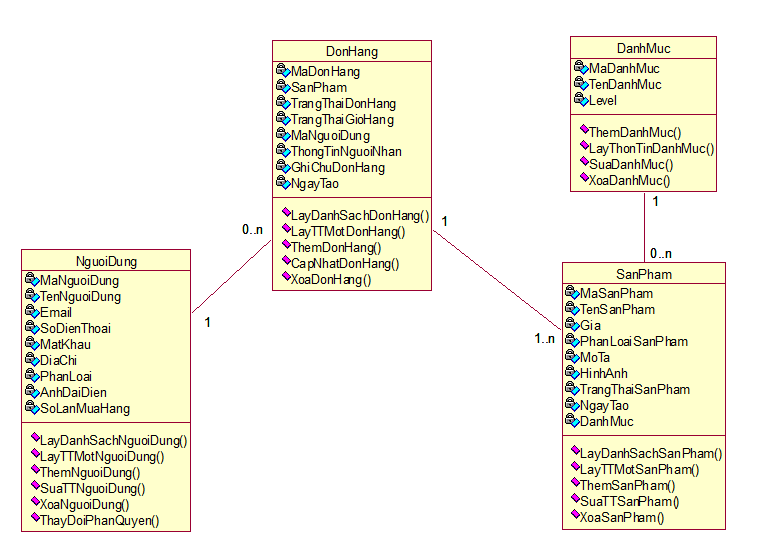
Hình 3.36. Biểu đồ hoạt động tìm kiếm sản phẩm

* + 1. Xem chi tiết sản phẩm



Hình 3.37. Biểu đồ hoạt động xem chi tiết sản phẩm

* 1. Biểu đồ lớp chi tiết



Hình 3.38. Biểu đồ lớp chi tiết

* 1. Thiết kế cơ sở dữ liệu



Hình 3.. Thiết kế cơ sở dữ liệu

Mô tả chi tiết thiết kế các bảng cơ sở dữ liệu

Bảng 3.5. Bảng danh mục

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| id | ObjectId | null | Id của danh mục |
| name | String | null | Tên danh mục |
| key | String | null | Từ khóa danh mục |
| level | Int32 | 1 | Cấp độ của danh mục |

Bảng 3.6. Bảng sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| id | ObjectId | null | Id của danh mục |
| name | String | null | Tên danh mục |
| highlight | Int32 | 1 | Sản phẩm nổi bật |
| topBuy | Int32 | 0 | Sản phẩm được mua nhiều |
| image | Array | null | Hình ảnh |
| status | Int32 | 0 | Trạng thái sản phẩm |
| type | Array | null | Phân loại sản phẩm |
| price | String | null | Giá sản phẩm |
| description | String | null | Mô tả sản phẩm |
| category | String | null | Danh mục của sản phẩm |
| createdAt | Date | null | Ngày tạo |
| updatedAt | Date | null | Ngày cập nhật |

Bảng 3.7. Bảng người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| id | ObjectId | null | Id khách hàng |
| name | String | null | Tên khách hàng |
| phone | String | null | Số điện thoại |
| email | String | null | Email |
| address | String | null | Địa chỉ |
| role | String | customer | Phân quyền tài khoản |
| type | String | null | Số lần mua hàng |
| password | String | null | Mật khẩu |
| createdAt | Date | null | Ngày tạo |
| updatedAt | Date | null | Ngày cập nhật |

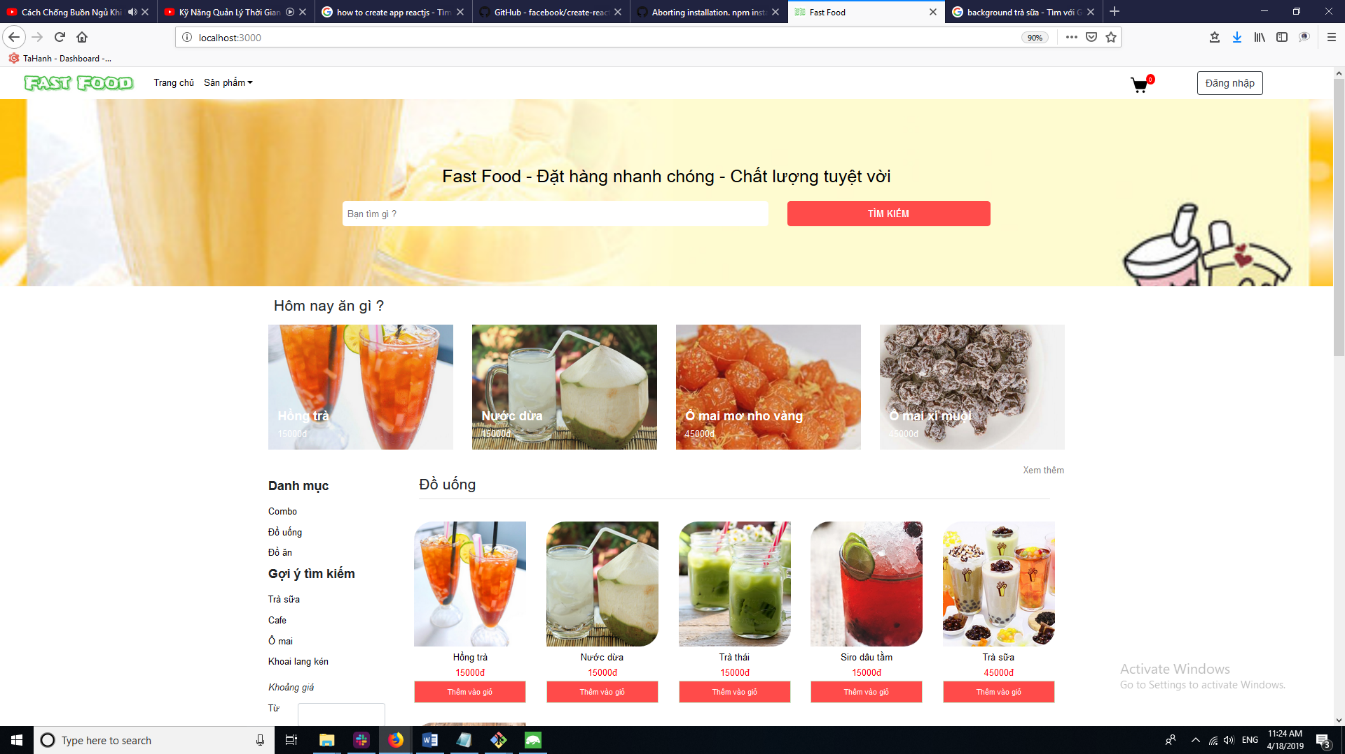
Bảng 3.8. Bảng đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| id | ObjectId | null | Id đơn hàng |
| products | Array | null | Danh sách sản phẩm |
| statusOrder | Array | null | Trạng thái đơn hàng |
| statusShip | Array | null | Trạng thái giao hàng |
| message | String | null | Ghi chú đơn hàng của khách hàng |
| idUser | String | null | Id khách hàng |
| user | Object | Null | Thông tin người nhận hàng |
| createdAt | Date | null | Ngày tạo |
| updatedAt | Date | null | Ngày cập nhật |

# CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

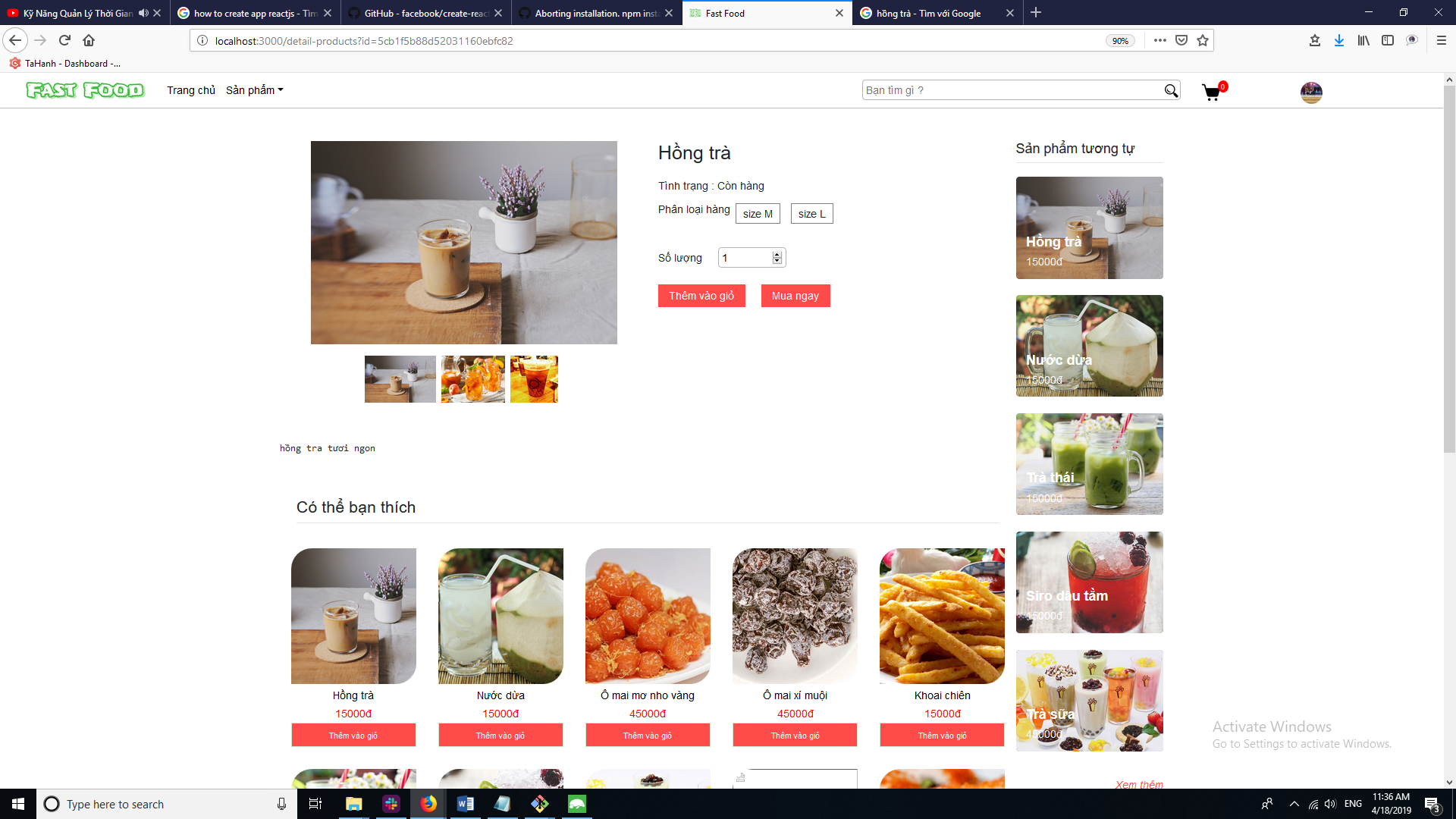
1. **Giao diện người dùng**

Trang chủ hiển thị các sản phẩm theo danh mục đồ ăn, đồ uống, combo, sản phẩm được yêu thích và thanh tìm kiếm sản phẩm theo từ khóa nhập của khách hàng.

****

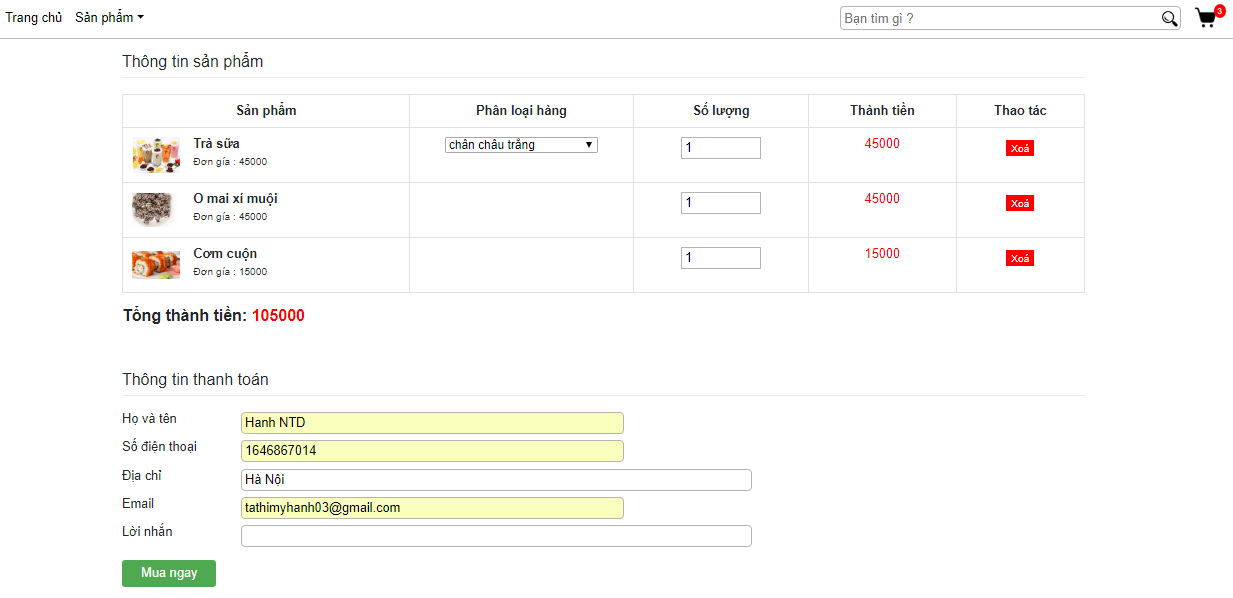
Hình 4.1. Giao diện trang chủ

Giao diện chi tiết sản phẩm đưa ra thông tin chi tiết sản phẩm như trạng thái sản phẩm hiện tại còn hay hết hàng, phân loại hàng ( nếu có ), số lượng sản phẩm mua. Sau khi lựa chọn có thể thêm sản phẩm vào giỏ hàng rồi xem tiếp các sản phẩm khác hoặc mua ngay để đi tới đặt hàng. Đồng thời gợi ý những món ăn cũng danh mục khách hang đang xem và những món ăn đang nổi bật tại quán.

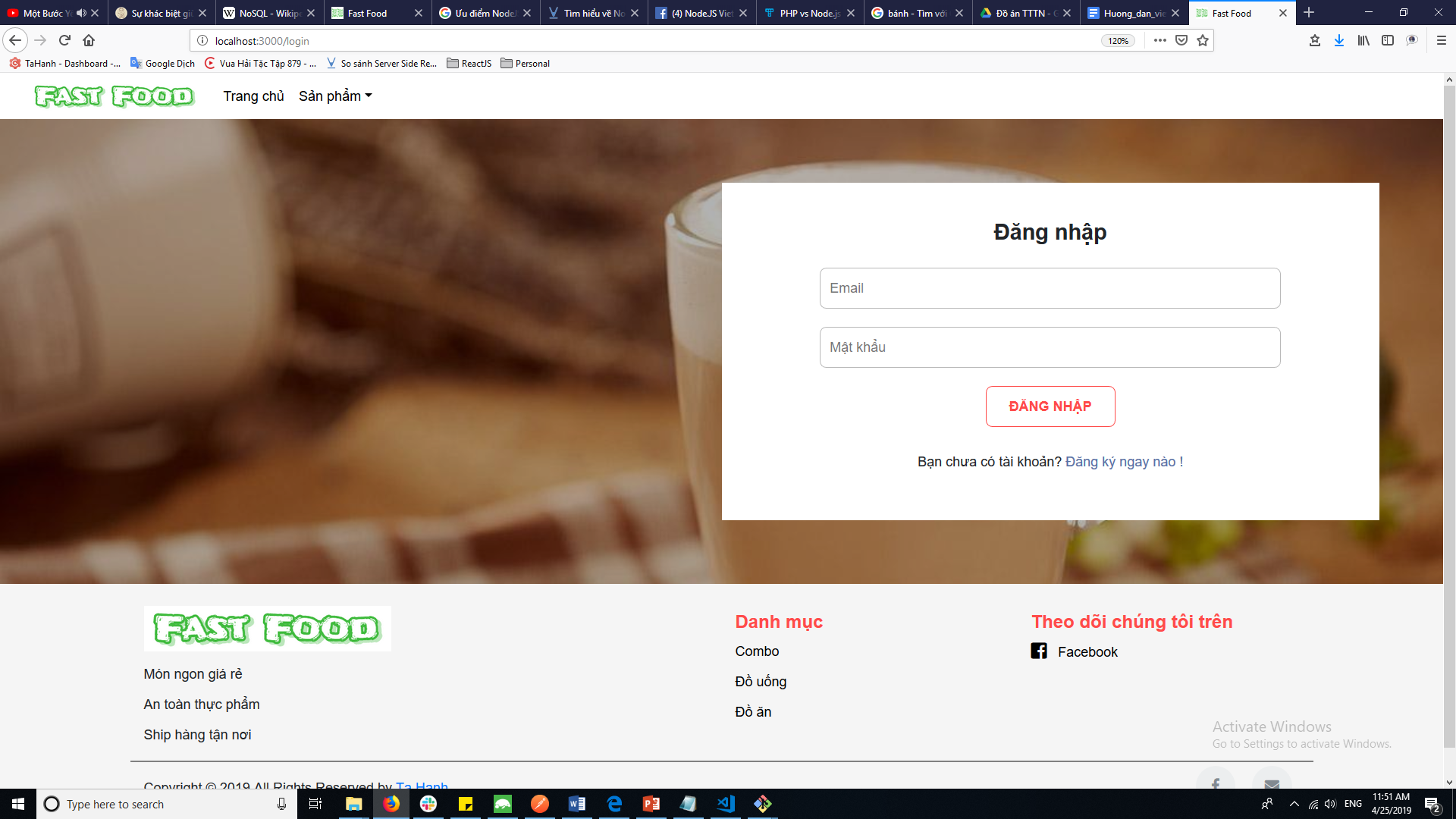


Hình 4.2. Giao diện chi tiết sản phẩm

Giao diện đặt mua hàng hiển thị danh sách các mặt hàng đã được thêm vào giỏ hàng. Tại đây khách hàng có thể thay đổi phân loại hàng, số lượng để đi tới đặt đơn hàng. Có thể xóa sản phẩm không mua nữa. Sau đó khách hàng nhập thông tin cá nhân chính xác nhất để quán ăn có thể liên hệ và giao hàng chính xác cho khách hàng.

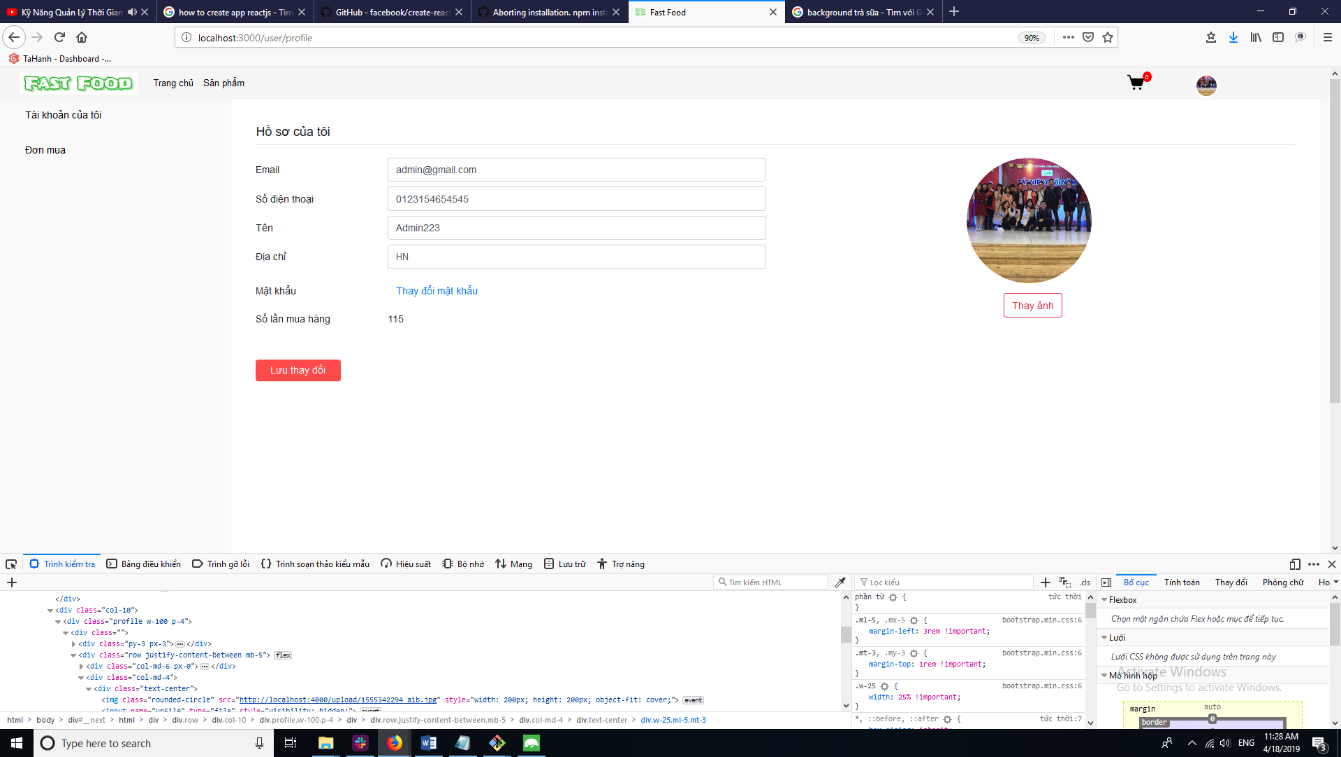


Hình 4.3. Giao diện đặt mua hàng

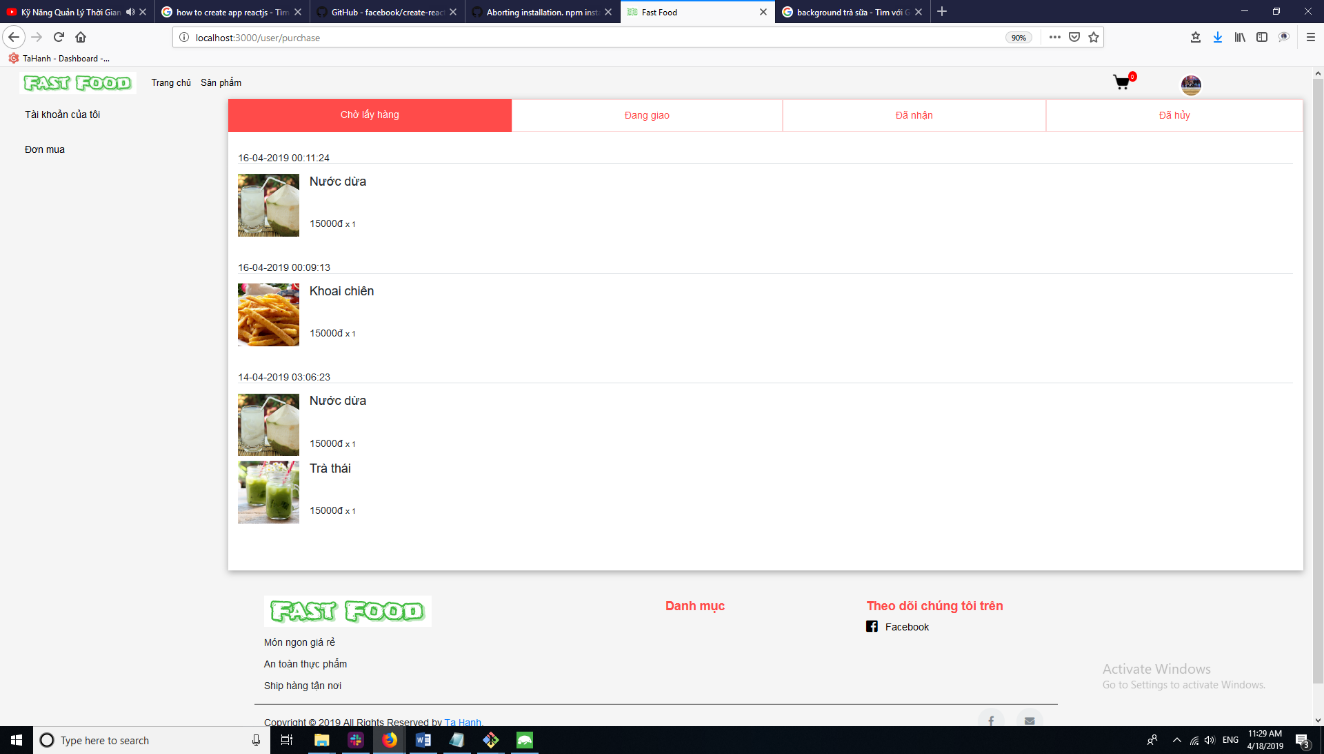


Hình 4.4. Giao diện đăng nhập

Với những khách hàng đã có tài khoản tại hệ thống sẽ có thể xem chỉnh sửa được thông tin tài khoản của mình. Thuận lợi khi có tài khoản khách hàng có thể theo dõi đơn hàng của mình: đã được xác nhận chưa hay đang giao hàng…



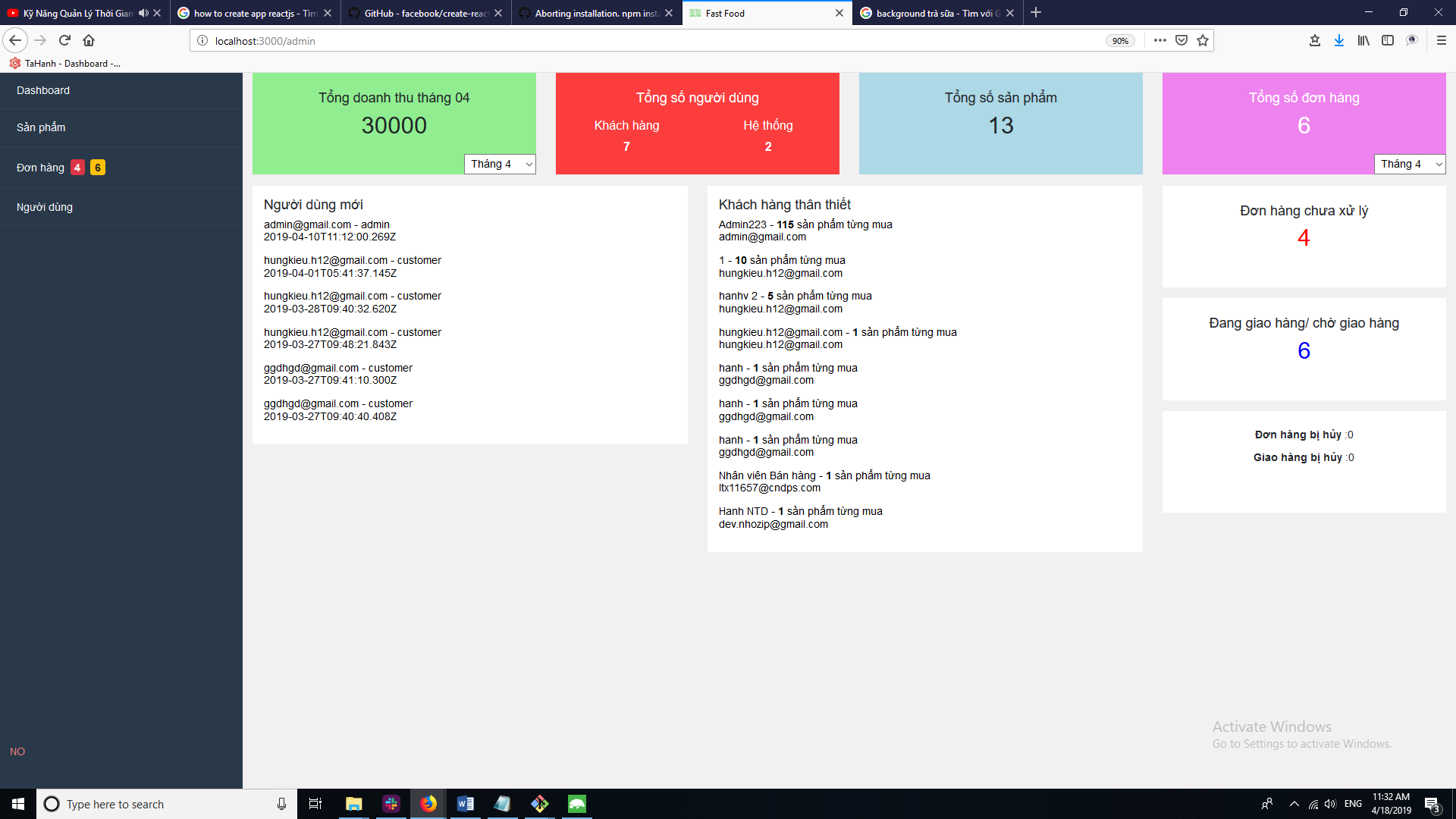
Hình 4.5. Giao diện tài khoản người dùng



Hình 4.6. Giao diện lịch sử đơn hàng

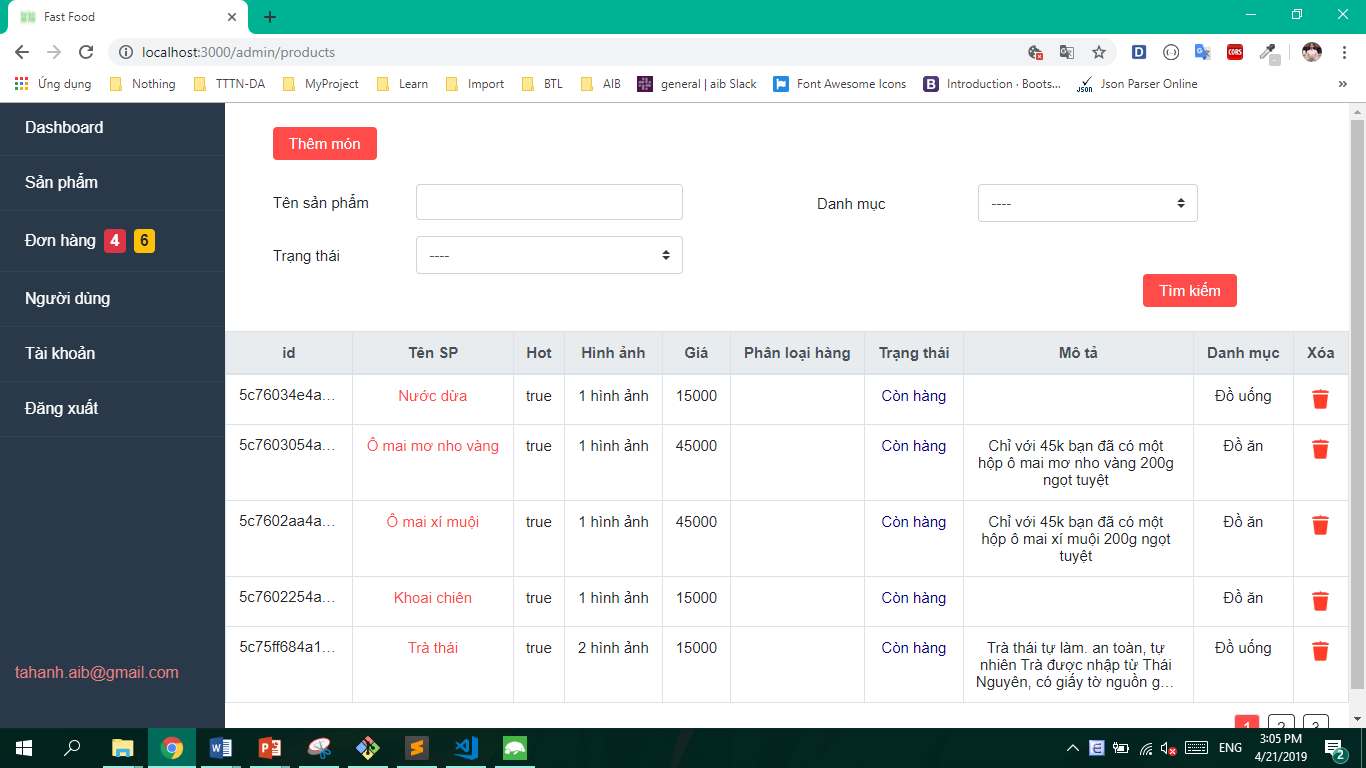
1. **Giao diện quản trị**

Tại trang theo dõi hoạt động của hệ thống. người quản lý sẽ biết được thông tin tài khoản mới cập nhật, doanh thu hàng tháng, số lượng đơn hàng chưa được xử lý hay những khách hàng có số lượng mua hàng cao hiện nay để có những chính sách ưu đãi cho khách hàng thân thiết.



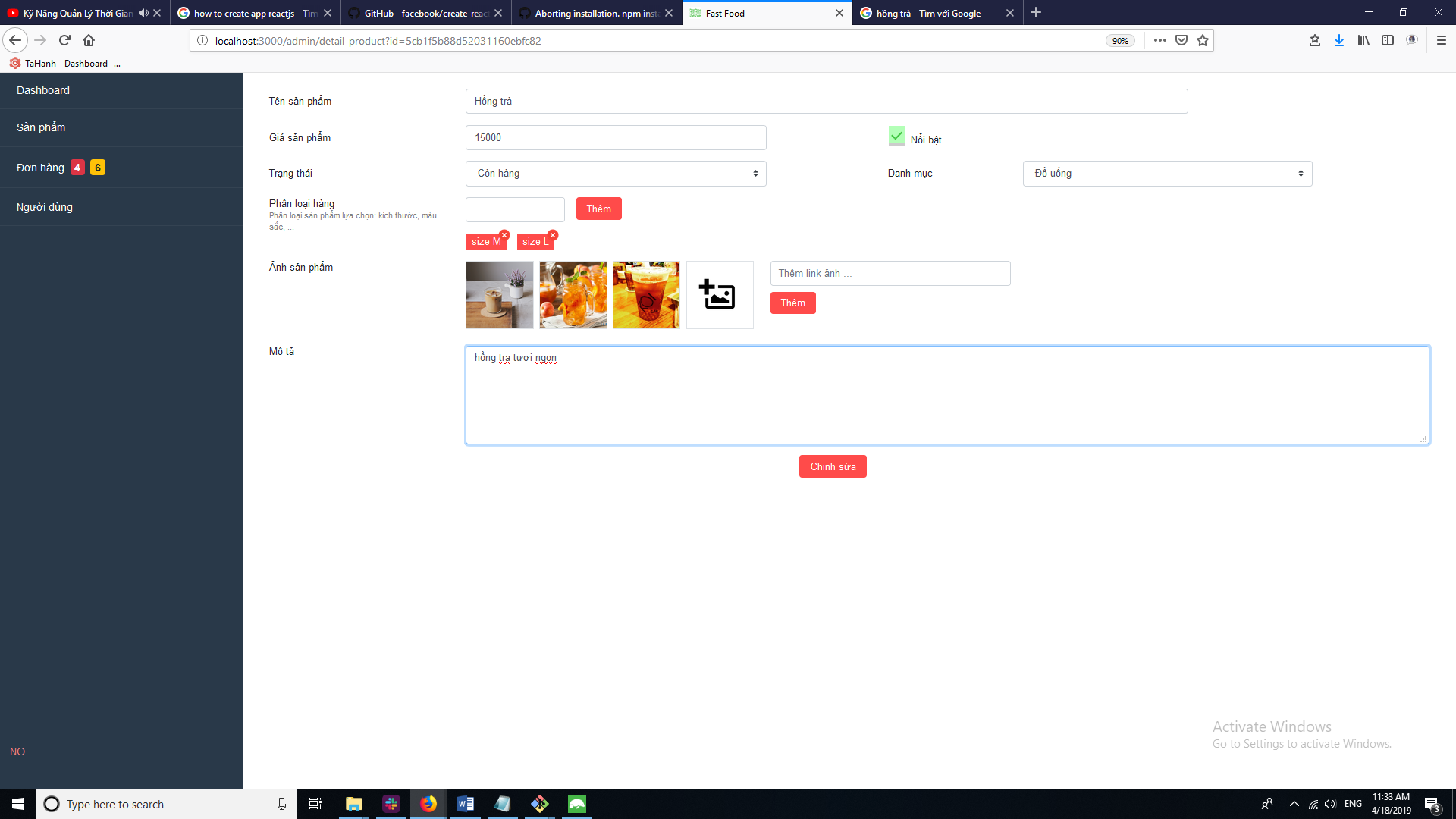
Hình 4.7. Giao diện trang báo cáo thống kê

Màn hình quản lý danh sách sản phẩm. Các thông tin cơ bản của sản phẩm sẽ được hiển thị trên màn hình, có thể thực hiện “Xóa” sản phẩm tại màn hình này, đồng thời giao diện có form tìm kiếm theo một vài thông số của sản phẩm như tên sản phẩm, danh mục của sản phẩm hay trạng thái sản phẩm. Và nút “Thêm” khi muốn thêm sản phẩm mới.



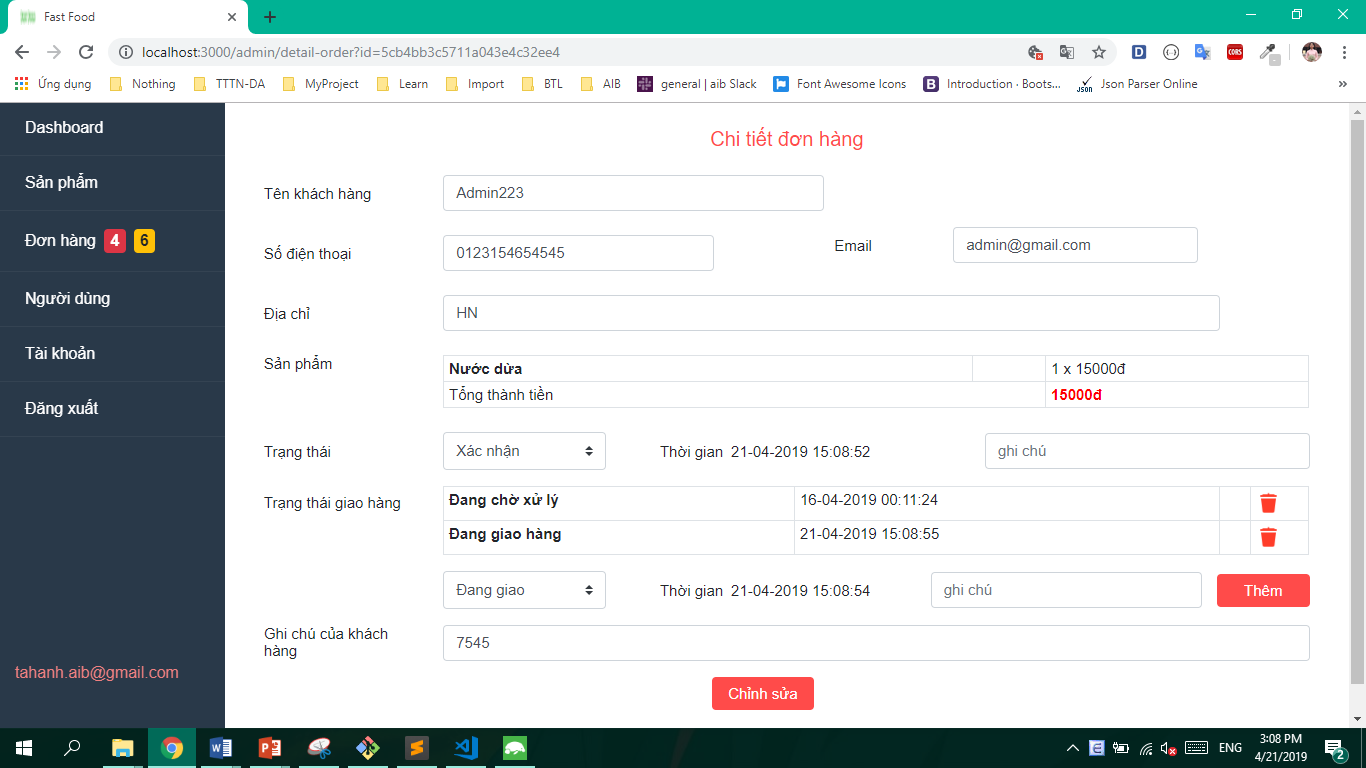
Hình 4.8. Giao diện trang quản lý sản phẩm

Màn hình chi tiết sản phẩm hiển thị khi thêm mới sản phẩm hay xem một sản phẩm đã tồn tại bất kì bao gồm các thông tin về sản phẩm như: tên sản phẩm, giá, sản phẩm nổi bật hay không, trạng thái hiện thái của sản phẩm, phân loại sản phẩm ( nếu có), ảnh sản phẩm, mô tả chi tiết.

****

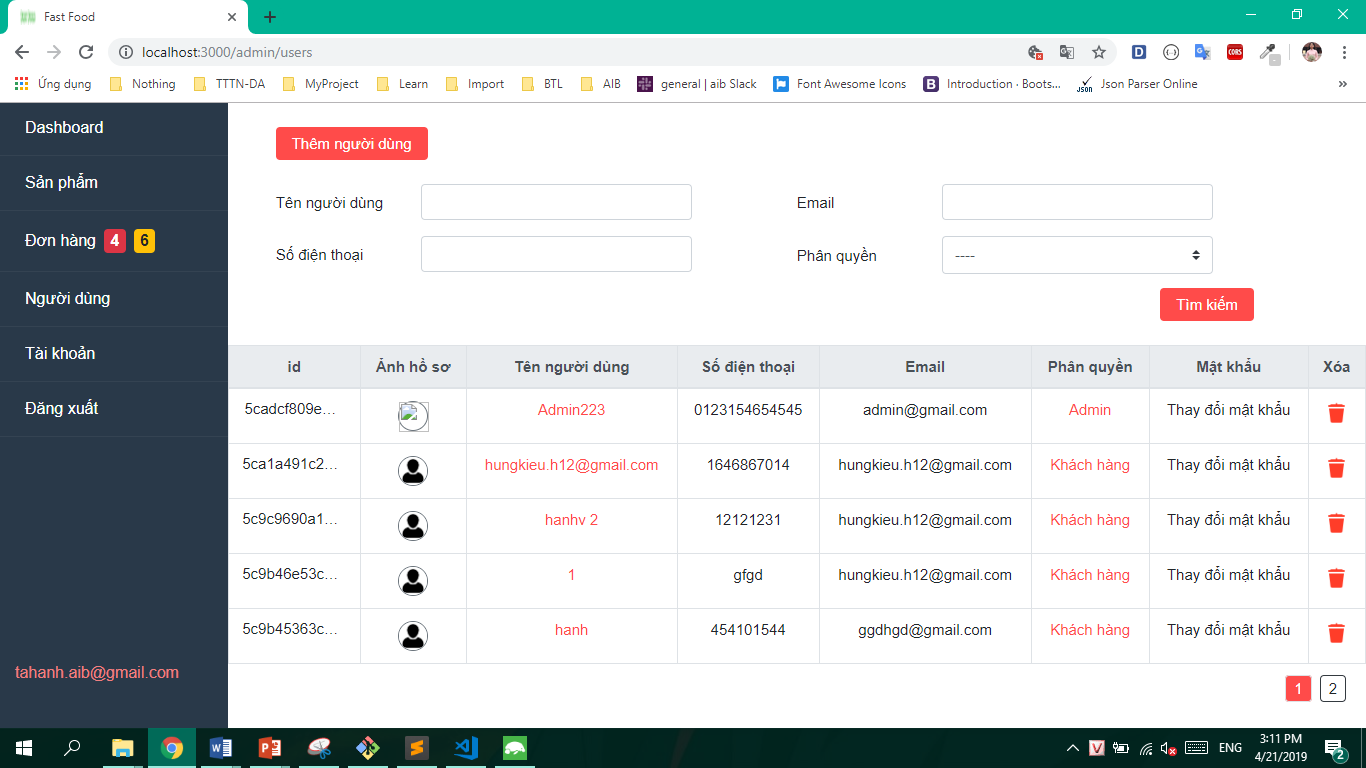
Hình 4.9. Giao diện cập nhật chi tiết sản phẩm

Màn hình chi tiết đơn hàng. Tại đây sẽ thực hiện cập nhật trạng thái đơn hàng từ lúc đơn hàng từ khách hàng gửi tới hệ thống. Nhân viên sẽ cập nhật trạng thái đơn hàng có được xác nhận hay hủy đơn, và cũng cập nhật trạng thái chuyển giao: đang chuyển, đã nhận hay đã bị hủy đơn để quán ăn có thể thống kê được tình hình kinh doanh của quán.



Hình 4.10. Giao diện cập nhật trạng thái đơn hàng

Màn hình quản lý người dùng. Các thông tin cơ bản của người dùng sẽ được hiển thị trên màn hình, có thể thực hiện “Xóa” người dùng tại màn hình này. Và nút “Thêm” khi muốn thêm tài khoản người dùng mới.



Hình 4.11. Giao diện quản lý người dùng

# KẾT LUẬN

Qua thời gian tìm hiểu và nghiên cứu đề tài “**Nghiên cứu ngôn ngữ NodeJS thư viện ReactJS và xây dựng website bán đồ ăn đêm**“ em đã học hỏi được cơ bản một số công nghệ: NodeJS, MongoBD, ReactJS, NextJS, MobX và áp dụng chúng vào chương trình của mình. Phân tích, thiết kế và hoàn thành cơ bản theo đúng những yêu cầu về nội dung và thời gian đã định.

Trong quá trình nghiên cứu và thực hiện đồ án do hạn chế về kinh nghiệm và kiến thức nên đề tài của em không thể tránh khỏi những thiếu xót. Em rất mong nhận được ý kiến đóng góp từ quý thầy cô và các bạn để chương trình được hoàn thiện hơn.

**Kết quả đạt được**

* Hoàn thiện khảo sát, phân tích, thiết kế website bán đồ ăn đêm.
* Website đã có một vài chức năng cơ bản như: đăng nhập, xem chi tiết sản phẩm, giỏ hàng, đặt hàng, quản lý sản phẩm, quản lý người dùng, quản lý đơn hàng, tìm kiếm.
* Website đã có cơ sở dữ liệu đúng với yêu cầu phân tích, thiết kế.
* Website đã được cập nhật giao diện đúng như mẫu thiết kế.

**Hạn chế của đề tài**

Do thời gian làm có hạn lên em chưa thể hoàn thiện đầy đủ chức năng của hệ thống, cũng như xử lý tất cả các ngoại lệ xảy ra trong quá trình sử dụng:

* Nội dung website vẫn còn sơ sài, vẫn còn hạn chế chức năng.
* Chưa kiểm soát được tất cả các lỗi có thể xảy ra trong quá trình sử dụng.
* Chưa hiểu rõ và áp dụng tốt được các công nghệ vào đề tài.

**Hướng phát triển tương lai**

* Xây dựng thêm nhiều tiêu chí tìm kiếm trên trang chủ và trang quản trị.
* Chức năng thống kê doanh thu theo từng thời điểm để người quản lý có thể so sánh doanh thu của cửa hàng qua các tháng.
* Thông báo khi có đơn hàng mới.
* Quản lý danh mục và giao diện.
* Tìm kiếm phương pháp kỹ thuật gợi ý món ăn theo sở thích người dùng.
* Xử lý đơn hàng nghiệp vụ nhanh chóng chính xác hơn và có thể xuất hóa đơn.

Cuối cùng xin gửi lời cảm ơn đến thầy Trịnh Bá Quý đã hướng dẫn em hoàn thành đề tài đồ án tốt nghiệp này!

# TÀI LIỆU THAM KHẢO

[1]. Giáo trình Phân tích thiết kế hệ thống hướng đối tượng, trường Đại học Công nghiệp Hà Nội

[2]. Đoàn Văn Ban, Phân tích thiết kế hướng đối tượng bằng UML, NXB Đại học Quốc gia Hà Nội, 2005

[3]. <https://techblog.vn/redux-hay-mobx-ly-giai-su-nham-lan> , truy cập cuối cùng ngày 03/03/2019

[4]. <https://reactjs.org/> , truy cập cuối cùng ngày 20/04/2019

[5]. MongoDB - <https://www.mongodb.com/> , truy cập cuối cùng ngày 20/04/2019

[6]. [https://www.npmjs.com/package/generator-rest , truy cập cuối cùng ngày 05/04/2019](https://www.npmjs.com/package/generator-rest%2005/04/2019)

[7]. <https://www.youtube.com/channel/UCKYDx_z7cXSUevpqnHZc8VQ> , truy cập cuối cùng ngày 02/04/2019

[8]. <https://en.wikipedia.org/wiki/React_(JavaScript_library)#One-way_data_binding_with_props> , truy cập cuối cùng ngày 16/03/2019

[9]. <https://vi.wikipedia.org/wiki/NoSQL> , truy cập cuối cùng ngày 15/03/2019

[10]. <https://docs.mongodb.com/manual/> , truy cập cuối cùng ngày 20/04/2019