NỘI DUNG ĐỀ CƯƠNG CHI TIẾT

1. Ý nghĩa của đề tài

Hiện nay cùng với sự phát triển của khoa học và công nghệ nên vấn đề tin học cũng không quá xa lạ với tất cả mọi người và máy tính điện tử đã trở thành một công cụ đắc lực không chỉ giam nhẹ lao động mà còn giúp cho chúng ta những năm lực mới mà trước đây chúng ta không thể làm được.

Ngày nay dưới sự phát triển mạnh mẽ của hệ thống mạng internet thì các trang web thương mại, bán hàng online càng được phổ biến và ngày càng mở rộng.Việc cung cấp tới khách hàng các mặt hàng mà cửa hàng của mình có thông qua chính website bán hàng của công ty, doanh nghiệp hay cá nhân cửa hàng , tạo sự chọn lựa một cách nhanh chóng, thuận tiện đối với những người bận rộn, hay vì khoảng cách đia lý mà không thể tới trực tiếp cửa hàng để có thể chọn lựa, mua được món hàng mà mình ưa thích.Điều đó đã tạo nên sự phát triển manh mẽ của các website thương mại. Phần mềm quản lý website bán quần áo online là một phần mềm hỗ trợ việc quản lý, xây dựng, và phát triển website bán quần áo online thong qua hệ thống mạng internet.

Và để giúp cho hoạt động mua bán hàng thong qua website được thuận lợi, khách hàng lựa chọn trong danh mục hàng, loại hàng có giá trị phù hợp với nhu cầu muốn mua và cho vào giỏ hàng của mình.Khách hàng tiến hành thanh toán các mặt hàng trong giỏ thì hệ thống sẽ gửi 1 đường dẫn tới hệ thống thanh toán.Hệ thống thanh toán xác thực thông tin và gửi thông tin lại cho hệ thống.

Nhân viên của hệ thống có nhiệm vụ hỗ trợ khách hàng, xửa lý các giao dịch với khách hàng và chuyển hàng hóa cho hệ thống vận chuyển.

Quản lý hệ thống(admin) có quyền thêm , bớt, sửa, xóa các mặt hàng có trên website, điều chỉnh giá hàng cho phù hợp với thị trường, và quản lý thông tin người dùng.

1. **Mục tiêu đề tài .**

* Xây dựng website bán máy tính.

1. **Đối Tượng và phạm vi nghiên cứu**
2. **Đối tượng nghiên cứu.**
   * Phầm mềm Visual studio code, Sqlite3, Ruby, Ruby on rails, Mô hình MVC, Angularjs, phân tích thiết kế UML.
3. Phạm vi nghiên cứu.
   * Lý thuyết về quy trình quản lý bán hàng.
   * Lý thuyết về mô hình MVC.
   * Xấy dựng website máy tính tập trung vào hàng được bán ra thị trường.
4. **Phương pháp nghiên cứu**
   * Nghiên cứu tư liệu (tài liệu Ruby, Ruby on rails, Angularjs, phân tích thiết kế hệ thống theo hướng đối tượng).
   * Phương pháp thực nghiệm (Lập trình, triển khai ứng dụng)
5. **Kết quả dự kiến đạt được.**

Xây Dựng một website bán phụ kiện máy tính chính hãng cho khách hàng.

**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.**

* 1. **Tìm hiểu Ruby.**

****

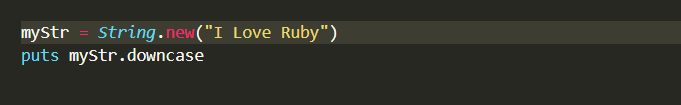
**1.1.1 Lịch sử ra đời của ruby.**

Ruby được tạo ra bởi Yukihiro "Matz" Matsumoto từ 24 tháng 2, 1993 và đưa ra bản chính thức vào năm 1995, hiện tại nó vẫn đang được tiếp tục phát triển.

Ruby là một ngôn ngữ hướng đối tượng: mỗi bit dữ liệu đều là một đối tượng (object), bao gồm các kiểu dữ liệu mà đối với các ngôn ngữ khác, chúng là kiểu cơ bản (primitive) như integer. Mỗi hàm (function) là một phương thức (method). Tên biến (variables) chính là tham chiếu (references) đến các đối tượng, bản thân nó không phải là đối tượng. Ruby hỗ trợ kế thừa (inheritance) với dynamic dispatch, mixin và singleton method (thuộc về, và để định nghĩa cho, một instance đơn hơn là định nghĩa dành cho lớp). Mặc dù Ruby không hỗ trợ đa kế thừa, các lớp vẫn có thể được đưa vào các module dưới dạng các mixins. Cú pháp dạng thủ tục (procedural syntax) vẫn còn được hỗ trợ, có vẻ như là ngoài tầm vực của mọi đối tượng, nhưng thực sự là thuộc một thể hiện của class Object tên là 'main'. Vì class này là cha của mọi class khác, nó trở trên ẩn đối với mọi lớp và đối tượng.

Ruby là ngôn ngữ lập trình kịch bản bởi mã lệnh của Ruby có thể chạy trực tiếp bởi máy tính mà không cần phải biên soạn thành một tệp thực thi tệp tin .exe trên Windows hay tệp tin binary trên Linux.

Ruby là ngôn ngữ lập trình hướng đối tượng (object oriented) và mọi thứ trong Ruby đều là một đối tượng ngay cả một chuỗi đơn giản:



**1.1.2. Điểm mạnh và điểm yếu của ruby**

**Điểm mạnh của Ruby:**

* Ruby được viết thuần hướng đối tượng, vì thế, mọi thứ trong Ruby đều là đối tượng. Tất cả những mặt mạnh của ngôn ngữ hướng đối tượng, Ruby đều kế thừa hoàn toàn, bao gồm cả sự linh động của chính ngôn ngữ. Ta có thể viết thêm những function khác vào Ruby mà không hề ảnh hưởng tới những function đã có.
* Ngôn ngữ Ruby được gọi là ngôn ngữ mà “The code comments itself”. Có nghĩa là bản thân các dòng lệnh trong Ruby đã nói lên ý nghĩa và công việc của nó.
* Được viết với sự focus chủ yếu vào tính dễ phát triển, nên Ruby cần ít code để thực hiện 1 công việc hơn so với các ngôn ngữ khác. Chính vì điều này, khi mới tiếp xúc với Ruby, chúng ta sẽ cảm thấy nó rất ấn tượng.

**Điểm yếu của Ruby**

* Ruby chậm hơn PHP khoảng 1.75~2 lần và chậm hơn JAVA tới 10 lần
* Vì Ruby là một ngôn ngữ mới, thực sự phát triển cũng chưa lâu nên có những hạn chế nhất định về mặt môi trường:
  + Chưa có nhiều người lập trình viên sử dụng Ruby
  + Có ít IDE hỗ trợ cho công việc
  + Có ít server hỗ trợ việc deploy bằng Ruby

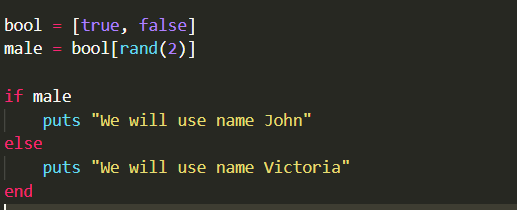
**1.1.3. Kiểu dữ liệu của ruby.**

Tất cả các chương trình máy tính trên thế giới đều sử dụng dữ liệu, từ trình soạn thảo văn bản, máy tính, game… dữ liệu cũng được chia làm nhiều loại khác nhau chẳng hạn như số, kí tự, hình ảnh, âm thanh… Kiểu dữ liệu là một tập các giá trị và các thao tác có thể có trên các giá trị đó.

* Boolean – kiểu luận lý
* Symbol
* Number – số
* String – chuỗi kí tự
* Array – mảng
* Hashe – bảng băm

**1.1.3.1. Kiểu dữ liệu Boolean.**

Kiểu dữ liệu Boolean là đối tượng thuộc kiểu này có một trong 2 trạng thái là True (đúng) hoặc False (sai). Đây là một kiểu dữ liệu quan trọng mà hầu hết ngôn ngữ lập trình nào cũng có.

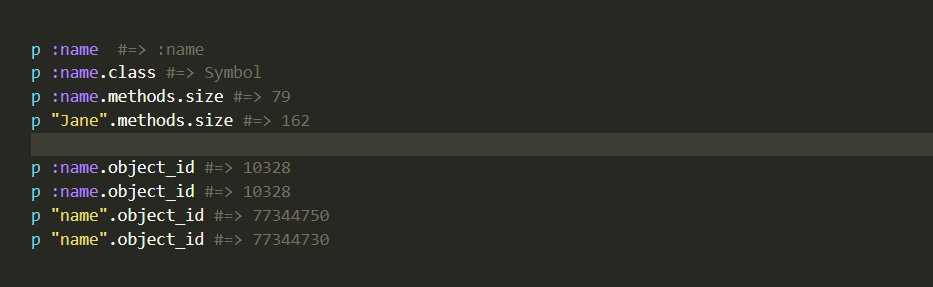


Khi phương thức rand() trả về 0 hoặc 1, chúng ta dùng kết quả đó để lấy giá trị tương ứng trong mảng bool, tức là nếu 0 thì biến male có giá tri là true, nếu 1 thì male là false.

**1.1.3.2. Kiểu dữ liệu Symbol**

Symbol được dùng để biểu diễn các đối tượng khác. Dùng symbol sẽ tiết kiệm được nhiều tài nguyên hơn so với dùng biến thông thường. Tất cả các biến symbol đều là một đối tượng từ lớp Symbol. Để định nghĩa một symbol thì chúng ta thêm dấu hai chấm ":" vào trước giá trị, ví dụ :name. Hầu hết các đối tượng trong Ruby đều có phương thức to\_sym để chuyển một đối tượng thành một symbol.

Symbol không thể thay đổi được giá trị. Thường thì symbol được dùng để làm khóa trong bảng băm (Hash).

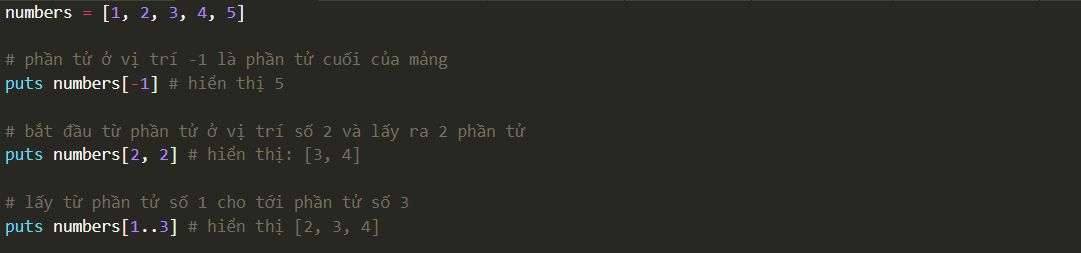


**1.1.3.3 Kiểu dữ liệu Array - Mảng.**

Kiểu dữ liệu mảng giống như một bộ sưu tập bao gồm các giá trị và từng giá trị được đánh số thứ tự bắt đầu từ 0.



Mảng có thể chứa các giá trị thuộc nhiều kiểu dữ liệu khác nhau. Trong Ruby, chúng ta có thể truy cập mảng theo các cách sau:

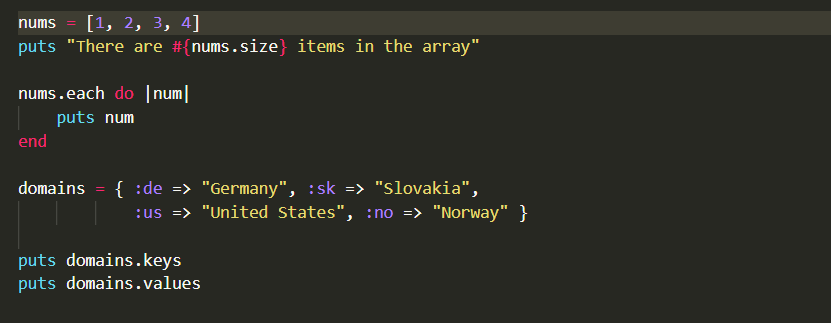


**1.1.3.4. Kiểu dữ liệu Hash - Mảng băm.**

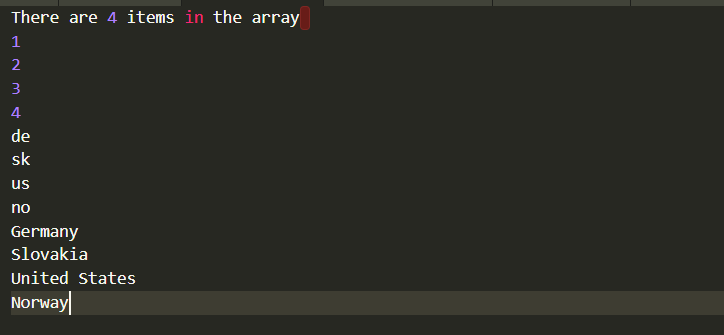
Trong Ruby kiểu dữ liệu hash tương tự như mảng với giá trị khoá có thể là 1 chuỗi, 1 số hay một đối tượng object thay vì số.

Mảng và bảng băm lưu dữ liệu theo dạng tập hợp, tức là chúng không lưu từng giá trị cụ thể mà lưu một nhóm các phần tử có giá trị khác nhau.

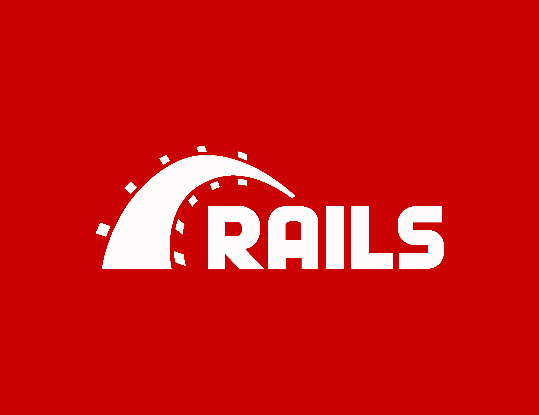
Mảng lưu các phần tử theo một thứ tự trong khi bảng băm lưu phần tử theo các cặp khóa-giá trị. Chúng ta sẽ tìm hiểu thêm về 2 kiểu dữ liệu này trong các đoạn mã sau.



Kết quả của đoạn mã:



**1.2. Tìm hiểu về thư viện Ruby on Rails.**

****

**1.2.1. Ruby on rails là gì?**

Ruby on Rails là một khung làm việc (web framework) được viết trên ngôn ngữ lập trình Ruby và được sử dụng trong việc xây dựng và phát triển các ứng dụng web. Phiên bản đầu tiên của Ruby on Rails được giới thiệu ra cộng đồng vào năm 2015. Ruby on Rails là phần mềm mã nguồn mở miễn phí được phát triển bởi David Heinemeier Hansson.

**1.2.2. Lợi ích của Ruby on rails**

Ruby on Rails được xây dựng trên các tập quán hiện đại nhất trong xây dựng và phát triển website do đó khi bạn nắm vững Ruby on Rails bạn sẽ dễ dàng chuyển sang một web framework khác.

Ruby on Rails có cấu trúc trong sáng và dễ hiểu rất dễ tiếp cận cho những lập trình viên mới vào nghề. Ngoài ra, Rails cung cấp cho chúng ta công cụ cửa sổ dòng lệnh CLI giúp tự động hoá nhiều công việc thường gặp khi lập trình qua đó tăng tốc độ và hiệu suất làm việc.

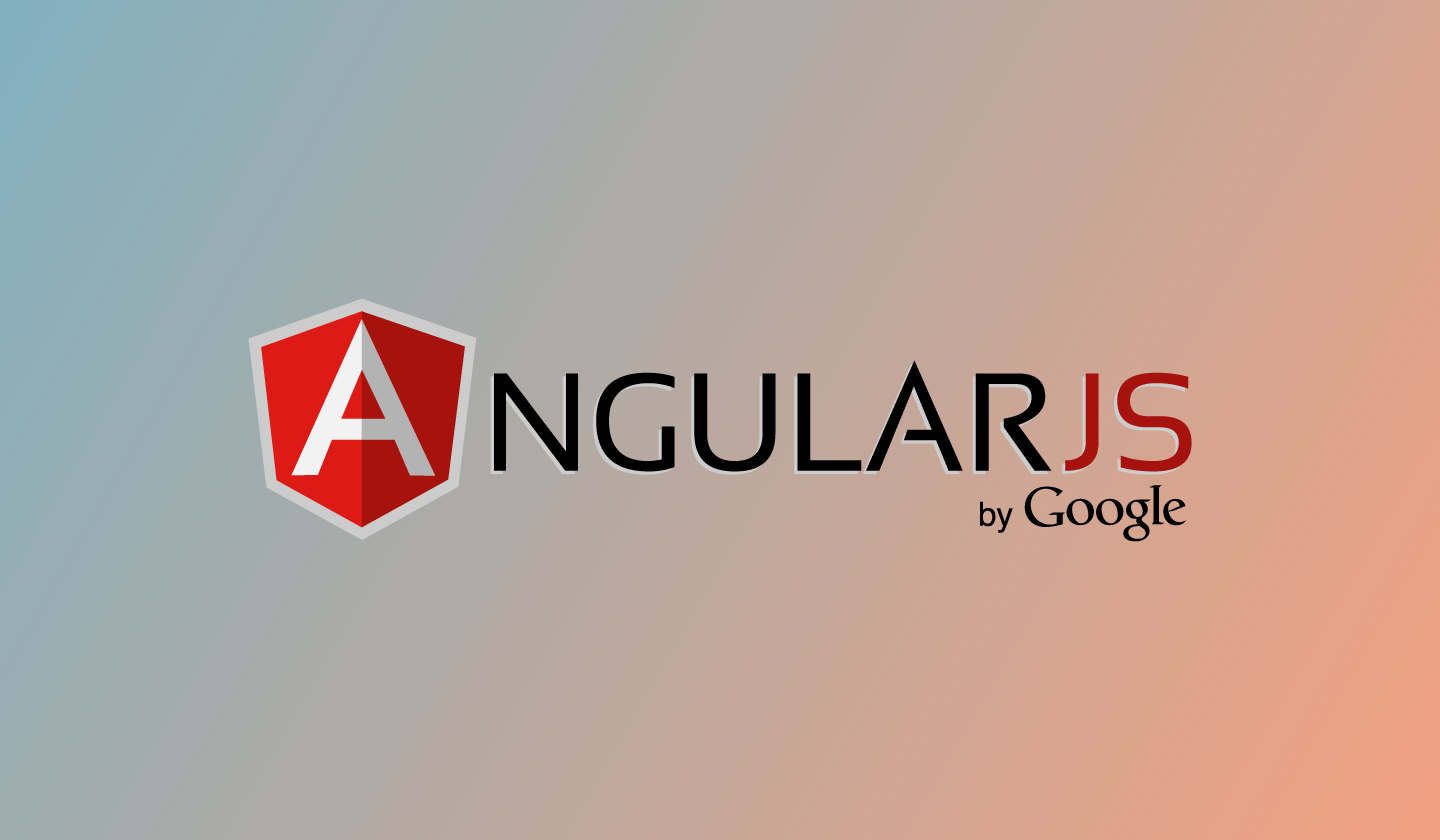
Ngoài ra, Ruby on Rails cũng là một trong những web framework được sử dụng phổ biến nhất hiện nay và được sử dụng bởi nhiều website lớn trên thế giới trong đó có mạng xã hội Twitter, trang mạng xã hội cho lập trình viên Github, công cụ quản lý dự án hàng đầu Redmine...

**1.2.3. Nhược điểm của Ruby on rails**

Thời gian chạy: Phần lớn những cuộc tranh luận về Rail đều có những ý kiến răng Rails chậm. Điều này hoàn toàn đúng, so với NodeJS hay GoLang, Rails có tốc độ chạy khá chậm. Mặc dù trong thực thế, hiệu suất của ứng dụng Rails rất khó làm chậm lại việc kinh doanh, Vấn đề thường nằm ở việc thiết kế server, thậm chí là cả cơ sở dữ liệu.

Tốc độ boot: Vấn đề chính gây khó chịu khi làm việc với Rails là tốc độ boot của Rails framework. Phụ thuộc vào số lượng gem và file, chúng ta có thể mất một khoảng thời gian kha khá để khởi động, điều này có thể ảnh hưởng trực tiếp tới hiệu suất làm việc. Với việc Spring ra đời, tốc độ boot đã được cải thiện nhưng có vẻ như nó vẫn còn có thể nhanh hơn nữa.

**1.3. Giới thiệu thư viện Angularjs**

****

**1.3.1. Angularjs**

AngularJS là một framework của JavaScript. Nó là 1 thư viện được viết trong JavaScript. Một thư viện dành xây đựng ứng dụng web động (dynamic web app).

Angular là một bộ thư viện javascript rất mạnh và thường được sử dụng để xây dựng project Single Page Application (SPA). Nó hoạt động dựa trên các thuộc tính mở rộng HTML (các atributes theo quy tắc của Angular). Đây là một bộ mã nguồn mở hoàn toàn miễn phí và được hàng ngàn các lập trình viên trên thế giới ưa chuộng và sử dụng.

**1.3.2. Ưu điểm của thư viện Angularjs**

Cấu trúc của nó cực tốt (rất tiện lợi khi làm single page với service routing)

Code ngắn gọn, phát triển ứng dụng single page rất nhanh(Nếu ko cần server thì người code cứng tay có thể phất triển ứng dụng trong vòng 10 ngày cả html + css + angularjs).

**1.3.3. Nhược điểm của thư viện Angularjs**

Vấn đề về hiệu suất sẽ là 1 vấn đề lớn đối với những ứng dụng có khối lượng data cực lớn. Nếu như app của bạn có rất nhiều dữ liệu ( ví dụ nếu bạn có rất nhiều model cần giải quyết) thì angular trở nên rất chậm chạp( thời gian có thể tính bằng giây). Cách giải quyết nó là bạn có thể tích hợp nó với ReactJS.

**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**2.1. Khảo sát yêu cầu**

Hiện nay có rất nhiều hình thức bán hàng khác nhau trên trực tuyến của một hệ thống hay nhiều hệ thống khác nhau. Với sự phát triển nhanh chóng của Internet và xu hướng công nghệ phát triển, thì việc bán hàng trực tuyến trở nên dễ dàng. Khách hàng có thể giảm bớt thời gian cho việc mua sắm khi bồn bề công việc công sở và có thể chọn mua sản phẩm trên tại trực tuyến mà không cần đi đến tận nơi để mua. Bên cạnh đó, việc tạo ra một trang web bán hàng phổ biến hiện nay rất dễ dàng, nhưng tạo ra một trang web vừa có tính bảo mật, đẹp, thuận tiện cho người dùng và đặc biệt là tạo ra sản phẩm trong thời gian ngắn để đáp ứng cho khách hàng. Đây là một lợi ích sau này của những ngôn ngữ mới, được các nhà phát triển tạo ra và phát trển bằng những framework (thư viện chức năng) . Và những điều trên sẽ giúp thuận tiện cho khách hàng của cửa hàng thương mại, chủ cửa hàng có một sản phẩm nhanh chóng để đưa vào hệ thống cửa hàng.

Website bán hàng máy tính và phụ kiện máy tính thường có những yêu cầu sau:

- Đăng nhập vào hệ thống và tự động chuyển tới giao diện của từng người dùng. Gồm có : Giao diện cho khách hàng và giao diện cho người quản trị và nhân viên cửa hàng. (quản trị hệ thống có thể vào các giao diện khác) tùy thuộc vào quyền người dùng trong hệ thống.

- Đăng xuất khỏi hệ thống và quay lại trạng thái chờ đăng nhập.

**Khách hàng:**

- Cho phép khách hàng đăng kí để sử dụng chức năng mua hàng của cửa hàng.

- Có thể xem thông tin cá nhân của khách hàng.

- Cập nhật thông tin của khách hàng.

- Có thể mua sản phẩm của cửa hàng được người quản trị và nhân viên cập nhật trên hệ thống.

- Có thể thấy thông tin, hình ảnh, giá tiền, số lần mua hàng của những sản phẩm được mua.

- Sử dụng chức năng bình luận để đánh giá nhiều sản phẩm.

- Thấy được những sản phẩm đã chọn mua trong giỏ hàng.

- Có thể chọn nhiều sản phẩm trong giỏ hàng.

- Có thể xóa sản phẩm trong giỏ hàng trước khi kích mua hàng.

- Có thể yêu cầu cửa hàng mua hàng.

**Người quản trị:**

- Có thể xem được thống kê doanh thu của của hàng theo hiển thị biểu đồ.

- Quản lý tất cả tài khoản người dùng và nhân viên được đăng kí hệ thống.

- Quản lý tất cả danh mục của sản phẩm.

- Quản lý tất cả sản phẩm.

- Quản lý yêu cầu của người dùng.

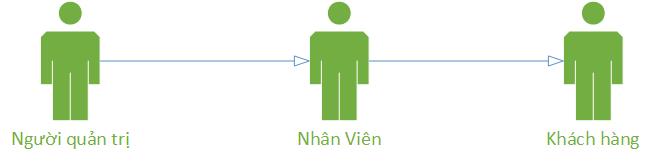
**2.2. Nhận diện các tác nhân**

Trong hệ thống website bán hàng bao gồm có các tác nhân sau:

**-** Người quản trị hệ thống(Admin): tác nhân này có chức năng quản trị toàn bộ hoạt động của hệ thông. Admin có quyền truy cập đến tất cả các chức năng của hệ thống, có mọi quyền của các tác nhân khác. Ngoài ra admin có thêm chức năng thêm, xoá người dùng và phân quyền cho người dùng.

- Nhân Viên: tác nhân này có chức năng là quản lý tài khoản khách hàng. Có thể thêm, sửa sản phẩm vào hệ thống**.**

**-** Khách hàng: tác nhân này có chức năng xem sản phẩm và mua sản phẩm vào giao diện trang chủ của hệ thống. Ngoài ra có thêm chức năng bình luận và đánh giá nhiều sản phẩm.



*Hình 2.1 Các Tác nhân của hệ thống.*

**2.3. Sơ đồ ca sử dụng (Use Case Diagram)**

**2.3.1. Xác định các Use case hệ thống**

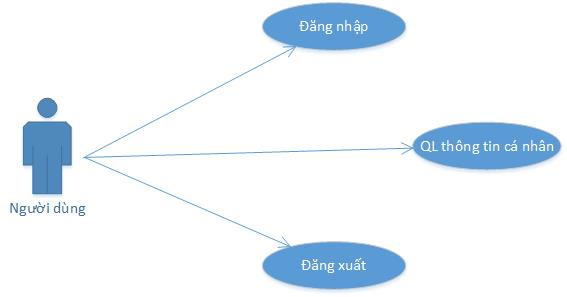
Hệ thống bao gồm các Use Case chính tương ứng với các tác nhân như sau :

*Bảng 2.1. Danh sách các tác nhân và Use Case*

|  |  |
| --- | --- |
| Danh sách các tác nhân và Use case | |
| **Actor (tác nhân)** | **Use case (ca sử dụng)** |
| Người dùng | 1. Đăng nhập  2. Đăng xuất  3. Quản lý thông tin cá nhân |
| Khách hàng | 1. Đăng kí  2. Mua sản phẩm  3. Xem thông tin của sản phẩm  4. Bình luận sản phẩm  5. Xem giỏ hàng  6. Xóa sản phẩm trong giỏ hàng  7. Thêm nhiều sản phẩm vào giỏ hàng  8. Yêu cầu sản phẩm |
| Nhân viên | 1. Quản lý sản phẩm hệ thống.  2. Xem danh sách danh mục sản phẩm.  3. Xem danh sách khách hàng.  4. Xem danh sách mua hàng. |
| Người quản trị | 1. Quản lý sản phẩm hệ thống  2. Quản lý danh mục sản phẩm  3. Quản lý người dùng (Nhân viên, khách hàng)  4. Quản lý mua hàng  5. Quản lý yêu cầu |

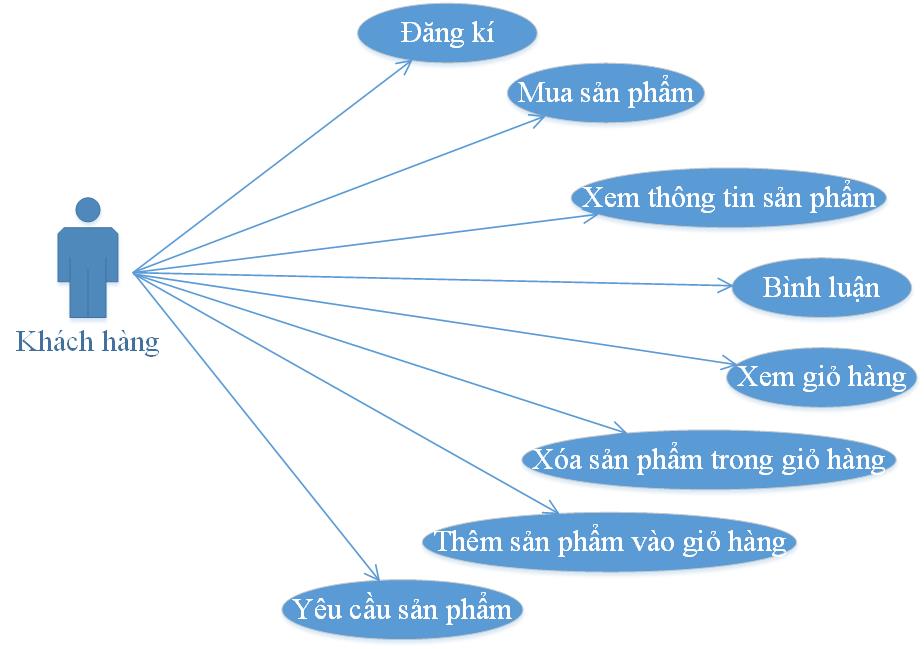
**2.3.2. Tác nhân người dùng**

Bao gồm những tác nhân khác: khách hàng, nhân viên, người quản trị



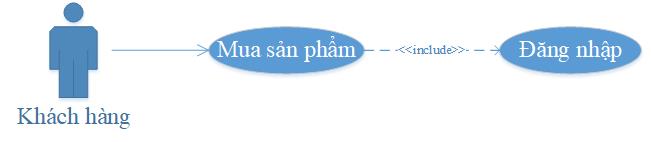
*Hình 2.2 Sơ đồ use case tác nhân người dùng*

**2.3.3. Tác nhân khách hàng.**



*Hình 2.3 Sơ đồ use case tác nhân khách hàng*

**2.3.3.1. Use case chức năng mua sản phẩm**

**

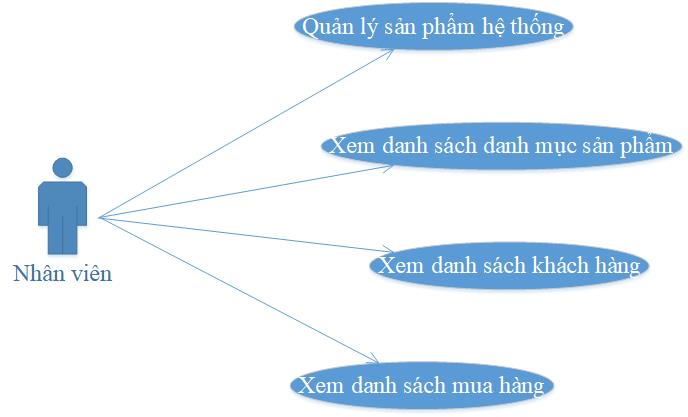
*Hình 2.4 Sơ đồ use case mua sản phẩm*

**2.3.3.2. Use case chức năng bình luận**



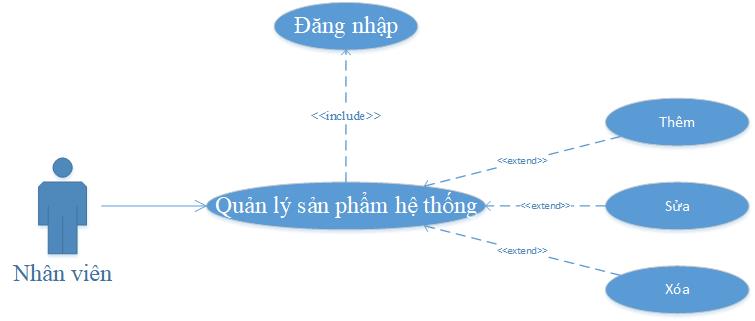
*Hình 2.5 Sơ đồ use case bình luận*

**2.3.4. Tác nhân nhân viên**

****

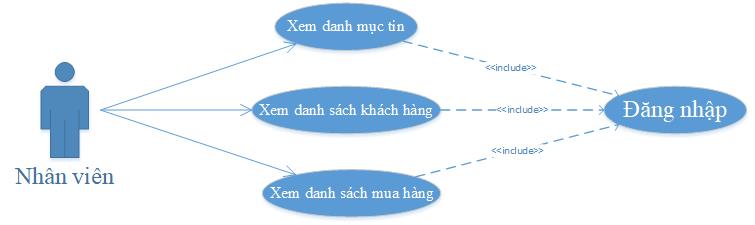
*Hình 2.6 Sơ đồ use case tác nhân nhân viên*

**2.3.4.1. Use case quản lý sản phẩm hệ thống**



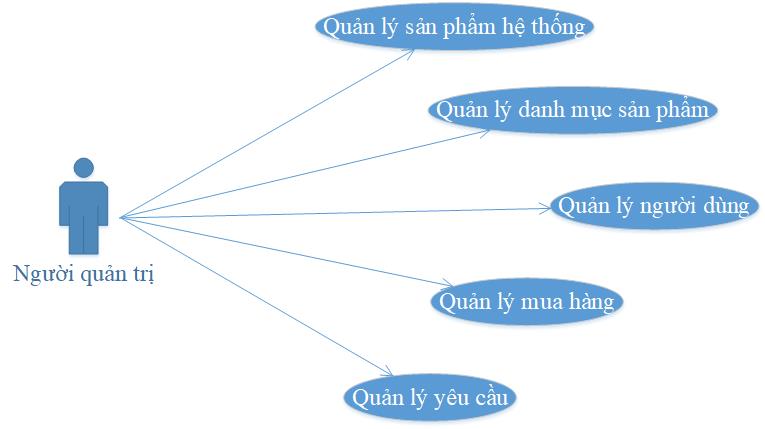
*Hình 2.7 Sơ đồ use case quản lý sản phẩm hệ thống*

**2.3.4.2. Use case xem danh mục, danh sách khách hàng, danh sách mua hàng**



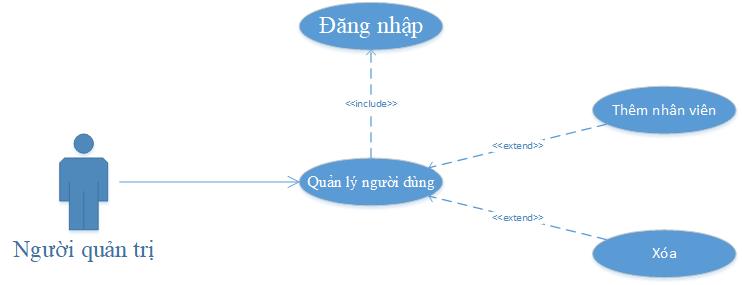
*Hình 2.7 Sơ đồ use case* xem danh mục, danh sách khách hàng, danh sách mua hàng

**2.3.5. Tác nhân người quản trị**

****

*Hình 2.8 Sơ đồ use case* tác nhân người quản trị

**2.3.5.1 Use case người dùng**



*Hình 2.9 Sơ đồ use case* người dùng

**2.3.5.2. Use case quản lý yêu cầu**



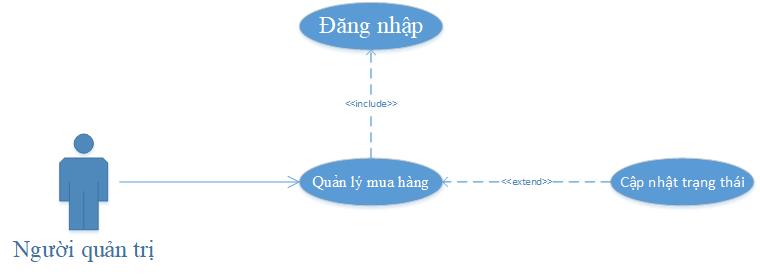
*Hình 2.10 Sơ đồ use case* quản lý yêu cầu

**2.3.5.3. Use case quản lý sản phẩm hệ thống**

****

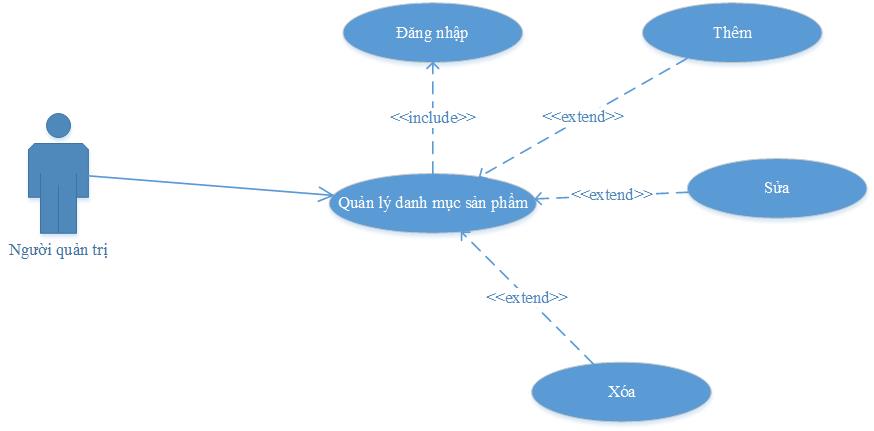
*Hình 2.11 Sơ đồ use case* quản lý sản phẩm hệ thống

**2.3.5.4. Use case quản lý mua hàng**

****

*Hình 2.11 Sơ đồ use case* quản lý mua hàng

**2.3.5.5. Use case quản lý danh mục**

****

*Hình 2.11 Sơ đồ use case* quản lý danh mục sản phẩm

**2.4. Mô tả các ca sử dụng (Use case)**

**2.4.1 Use case đăng nhập**

*Bảng 2.2 Use case đăng nhập*

|  |  |
| --- | --- |
| **Tên Use Case** | Đăng nhập |
| Người đặc tả | Văn Long |
| Tác nhân | Người dùng (Người quản trị, nhân viên, khách hàng) |
| Tổng quan : | Cho phép người dùng đã có tài khoản trong hệ thống đăng nhập vào hệ thống. |
| Tiền điều kiện : | Người dùng có tài khoản hệ thống. |
| Hậu điều kiện : | Đăng nhập thành công |
| Luồng làm việc : | (1) Người dùng vào hệ thống  (2) Người dùng chọn chức năng đăng nhập  (3) Hệ thống sẽ hiện thị trang đăng nhập  (4) Người dùng nhập email và mật khẩu đăng nhập  (5) Nếu tên đăng nhập và mật khẩu không tồn tại trong hệ thống thì hệ thống báo lỗi.  (6) Nếu tên đăng nhập và mật khẩu tồn tại trong hệ thống thì hệ thống sẽ chuyển trang:  - Nếu tài khoản là tài khoản Quản trị và nhân viên thì hệ thống sẽ hiển thị giao diện quản lý hệ thống.  - Nếu tải khoản là tải khoản khách hàng thì hệ thống sẽ hiển thị giao diện trang chủ danh cho người dùng.  (7) |
| Luồng làm việc rẽ nhánh |  |

*2.4.2 Use case đăng xuất.*

|  |  |
| --- | --- |
| **Tên Use Case** | Đăng xuất |
| Người đặc tả | Văn Long |
| Tác nhân | Người dùng (Người quản trị, nhân viên, khách hàng) |
| Tổng quan : | Cho phép người dùng đang sử dụng trong hệ thống đăng xuất khỏi hệ thống. |
| Tiền điều kiện : | Đã đăng nhập thành công và người dùng đang sử dụng hệ thống |
| Hậu điều kiện : | Thoát khỏi hệ thống |
| Luồng làm việc : | (1) Người dùng đang sử dụng hệ thống  (2) Người dùng chọn chức năng đăng xuất hệ thống  (3) Hệ thống sẽ chuyển về trang đăng nhập  (4) Hệ thống hiển thị thông báo đăng xuất thành công |
| Luồng làm việc rẽ nhánh |  |

*2.4.3 Use case quản lý thông tin*

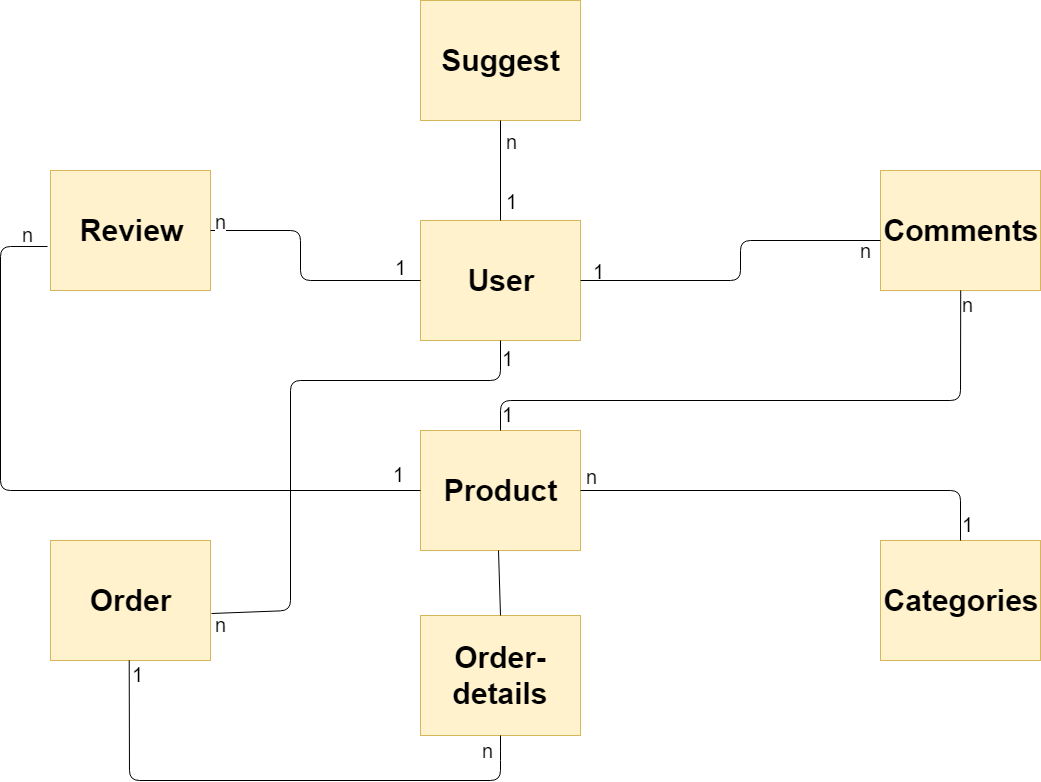
|  |  |
| --- | --- |
| **Tên Use Case** | Quản lý thông tin cá nhân |
| Người đặc tả | Văn Long |
| Tác nhân | Người dùng (Người quản trị, nhân viên, khách hàng) |
| Tổng quan : | Dùng để quản lý thông tin của người dùng trong hệ thống |
| Tiền điều kiện : | Người dùng có tài khoản và đăng nhập thành công vào hệ thống |
| Hậu điều kiện : | Thông tin người dùng sẽ được cập nhật lại hệ thống |
| Luồng làm việc : | (1) Người dùng sẽ đăng nhập vào hệ thống  (2) Sau khi đăng nhập thành công người dùng sẽ chọn chức năng quản lý thông tin cá nhân.  (3) Hệ thống sẽ hiện thị trang quản lý thông tin cá nhân.  (4) Hệ thống cho phép cập nhật thông tin cá nhân  (5) Điền thông tin cần thiết để cập nhật lại hệ thống  (6) Xác nhận  (7) Hiện thị thông báo thao tác thành công hoặc thất bạ |
| Luồng làm việc rẽ nhánh |  |

*2.4.3 Use case đăng kí tài khoản*

|  |  |
| --- | --- |
| **Tên Use Case** | Đăng kí tài khoản |
| Người đặc tả | Văn Long |
| Tác nhân | Khách hàng |
| Tổng quan : | Cho phép tạo tài khoản để người chưa có tài khoản trong hệ thống |
| Tiền điều kiện : | Cần có tài khoản email |
| Hậu điều kiện : | Có tài khoản để đăng kí vào hệ thống |
| Luồng làm việc : | (1) Khách vãn lai truy cập trang chủ hệ thống  (2) Khách vãn lai chọn chức năng đăng kí tài khoản ở trang chủ  (3) Hệ thống sẽ chuyển về trang đăng kí tài khoản  (5) Điền email, mật khẩu, tên người dùng và một số thông tin  (6) Hệ thống kiểm tra:  - Trường email có đúng định dạng một chuỗi email  - Trường mật khẩu kiểm tra chuỗi mật khẩu thấp nhất 6 kí tự và nhiều nhất 24 kí tự.  (7) Xác nhận  (8) Hệ thống hiển thị thông báo đăng xuất thành công. |
| Luồng làm việc rẽ nhánh |  |

*2.4.4 Use case mua sản phẩm.*

|  |  |
| --- | --- |
| **Tên Use Case** | Mua sản phẩm |
| Người đặc tả | Văn Long |
| Tác nhân | Khách hàng |
| Tổng quan : | Cho phép tạo tài khoản để người chưa có tài khoản trong hệ thống |
| Tiền điều kiện : | Cần có tài khoản email |
| Hậu điều kiện : | Có tài khoản để đăng kí vào hệ thống |
| Luồng làm việc : | (1) Khách vãn lai truy cập trang chủ hệ thống  (2) Khách vãn lai chọn chức năng đăng kí tài khoản ở trang chủ  (3) Hệ thống sẽ chuyển về trang đăng kí tài khoản  (5) Điền email, mật khẩu, tên người dùng và một số thông tin  (6) Hệ thống kiểm tra:  - Trường email có đúng định dạng một chuỗi email  - Trường mật khẩu kiểm tra chuỗi mật khẩu thấp nhất 6 kí tự và nhiều nhất 24 kí tự.  (7) Xác nhận  (8) Hệ thống hiển thị thông báo đăng xuất thành công. |
| Luồng làm việc rẽ nhánh |  |



**Cơ sở dữ liệu**

