**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**BÁO CÁO MÔN HỌC: PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: XÂY DỰNG WEBSITE BUÔN BÁN XE OTO CŨ**

Giảng viên hướng dẫn : ThS. Từ Thanh Trí

Sinh viên thực hiện : Nguyễn Việt Hoàng

Lớp: 06 ĐH CNTT 3

Khóa: 2017-2021

Nhóm: 05

TP. Hồ Chí Minh, tháng 5 năm 2020

**Lời nói đầu**

Trong thời buổi hiện nay, khi mà công nghệ phát triển, thương mại điện tử ngày càng phát triển trên thế giới cũng như Việt Nam và dần khẳng định được vị thế của mình trong nền kinh tế thị trường. Cùng với đó là sự ra đời của các ngôn ngữ lâp trình cho phép thiết kế và xây dựng các ứng dụng thương mại điện tử dưới nhiều hình thức khác nhau. Và một trong những ứng dụng của thương mại điện tử phổ biến ở nước ta là dịch vụ bán hàng qua mạng internet. Dịch vụ này cho phép người dùng tìm kiếm chọn lựa sản phẩm để mua, thực hiện giao dịch mà không cần phải trực tiếp đến cửa hàng, mà chỉ cần sử dụng một thiết bị máy tính có kết nối internet. Tuy nhiên hiện nay dịch vụ này vẫn còn khá mới với nhiều người tiêu dùng nước ta, đặc biệt là những người không có kiến thức về tin học nói chung và thương mại điện tử nói riêng. Nên hiện giờ các đối tượng chính của thương mại điện tử vẫn chủ yếu là tầng lớp tri thức và học sinh, sinh viên. Là những người sinh viên, cũng có chút kiến thức về tin học cũng như tiếp xúc với thương mại điện tử tuy nhiên đôi lúc cũng gặp một số khó khăn trong việc mua bán trên các hệ thống bán hàng qua mạng hiện tại. Như hình thức thanh toán, nhận hàng, chất lượng sản phẩm thực tế không được như trên website....

Hiểu được các vấn đề đó cũng như mong muốn đưa thương mại điện tử đến với nhiều người hơn chúng em thực hiện đề tài: **Xây dựng Website bán otô cũ.** Với mục đích xây dựng một hệthống bán hàng qua mạng uy tín, đơn giản, thân thiện, cũng như đem lại cho khách hàng những lựa chọn tối ưu để ngay cả những người không có kiến thức về tin học cũng có thể tham gia mua hàng qua mạng chỉ với một số thao tác đơn giản. Sản phẩm mà bọn em hướng tới là các sản phẩm xe hơi đã qua sử dụng.

Chúng em xin chân thành cảm ơn thầy Ths. Từ Thanh Trí đã giúp đỡ chúng em thực hiện đề tài này!

**CHƯƠNG I: Tổng quan về đề tài**

1. **Định nghĩa**

Hệ thống website buôn bán xe hơi cũ online là một website cung cấp cho khách có nhu cầu muốn mua xe hơi nhưng không bỏ ra chi phí quá cao, người quản lý website sẽ quản lý các chứ năng cần thiết để khi khách hàng có nhu cầu đặt mua có thể để lại thông tin của bản thân và thông tin sản phầm muốn mua. Về phía khách hàng trang web cho phép khách hàng có thể vào xem thông tin sản phẩm, và có thể đưa sản phẩm đó vào giỏ hàng. Sau khi chọn sản phẩm, khách hàng phải đăng nhập mới có thể tiến hành đặt hàng, sau khi đăng nhập khách hàng sẽ chuyển sang trang khách để điền đầy đủ thông tin như giới tính, họ tên, số điện thoại, thời gian nhận hàng, đặt hàng sau đó cửa hàng sẽ liên hệ trao đổi chi tiết về thời gian và những cách thức, giấy tờ và những thứ cần thiết khi đển nhận xe.

Đối với người tiếp nhận đơn hàng, sẽ có nhiệm vụ lấy thông tin của khách hàng và sản phẩm mà khách muốn đặt mua. Sau đó nhân viên đó sẽ liên hệ với khách hàng để trao đổi chi tiết hơn về các vấn đề về thủ tục, và lênh lịch hẹn chi tiết về thời gian khách hàng đến để nhận xe.

* Các nhóm chức năng của hệ thống

Hệ thống website có thể chia thành các nhóm chức năng chính như sau:

Về phía trang admin, người quản lý sẽ có thể thực hiện các chức năng như:

- Nhóm chức năng thêm mới sản phẩm, sửa thông tin của sản phẩm và xóa các sản phẩm mà cửa hàng không còn bán nữa.

- Nhóm chức năng đăng nhập của người quản lý.

- Nhóm chức năng xem thông tin, bao gồm thông tin giỏ hàng, xem sản phẩm trong đơn hàng, xem thông tin khách hàng đặt sản phẩm đó.

- Nhóm chức năng mua hàng, tiếp nhận và xử lý đơn hàng.

Về phía trang người dùng, người dùng có thể thực hiện các chức năng như:

Xem các sản phẩm của cửa hàng.

Tiến hành đặt sản phẩm mà mình muốn mua.

Đăng nhập để có thể mua sản phẩm.

* Xác định các tác nhân

- Khách hàng: là người giao dịch với hệ thống thông qua đơn đặt hàng, khách hàng có thể chọn loại sản phẩm, thời gian nhận hàng.

- Người quản lý: là người tiếp nhận và xử lý các đơn hàng, đồng thời là người điều hành, quản lý và theo dõi mọi hoạt động của hệ thống.

**CHƯƠNG 2: Cơ sở lý thuyết**

1. **Tổng quan về React**
2. **Giới thiệu**

ReactJS được hiểu nôm na là một thư viện trong đó có chứa nhiều JavaScript mã nguồn mở và cha đẻ của ReactJS đó chính là một ông lớn với cái tên ai cũng biết đó chính là Facebook. Mục đích của việc tạo ra ReactJS là để tạo ra những ứng dụng website hấp dẫn với tốc độ nhanh và hiệu quả cao với những mã hóa tối thiểu. Và mục đích chủ chốt của ReactJS đó chính là mỗi website khi đã sử dụng ReactJS thì phải chạy thật mượt thật nhanh và có khả năng mở rộng cao và đơn giản thực hiện.

Nhìn chung tất cả những tính năng hay sức mạnh của ReactJS thường xuất phát từ việc tập trung vào các phần riêng lẻ chính vì điểm này nên khi làm việc trên web thay vì nó sẽ làm việc trên toàn bộ ứng dụng của website thì ReactJS cho phép developer có chức năng phá vỡ giao diện của người dùng từ một cách phức tạp và biến nó trở thành các phần đơn giản hơn nhiều lần có nghĩa là render dữ liệu không chỉ được thực hiện ở vị trí sever mà còn có thể thực hiện ở vị trí Client khi sử dụng ReactJS.

**2. Ưu điểm**

Việc tạo ra dom như vậy giúp cải thiện hiệu suất làm việc rất nhiều, khi có tính toán cần thay đổi hoặc cần cập nhật những gì lên Dom thì ReactJS đều tính toán trước và việc còn lại chỉ là thực hiện chúng lên Dom, làm như vậy sẽ giúp cho ReactJS tránh được những thao tác cần có trên một Dom mà không cần tốn thêm bất cứ chi phí nào.

Ta có thể viết các đoạn code JS sẽ trởi nên dễ dàng hơn vì nó sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX nghĩa là cú pháp này cho phép ta trộn được giữa code HTML và Javascript. Ngoài ra ta còn có thể đem đoạn code thêm vào trong hàm render mà không cần phải thực hiện việc nối chuỗi và đây được đánh giá là một trong những đặc tính thú vị của ReactJS và việc chuyển đổi các đoạn HTML thành các hàm khởi động đều được thực hiện từ bộ biến đổi chính đó chính là JSX.

1. **Component trong React**

Component là thuật ngữ được hiểu như là bộ phận, thành phần hoặc yếu tố cấu thành của một hệ thống hoặc phần mềm trong lĩnh vực IT. Trong lĩnh vực phần mềm, component có chức năng khá mạnh mẽ, nó được tái sử dụng từ các chương trình khác, được liên kết và sử dụng như một bộ phận của chương trình.

Component trong ReactJS

Trong React hay Vue, Component là một thuật ngữ nên không thể định nghĩa rõ. Chúng ta có thể hiểu rằng mỗi đoạn code sẽ được phân chia thành những Component không lệ thuộc lẫn nhau và có thể tái sử dụng khi cần thiết.

Khác với các framework khác, React không sử dụng template mà được xây dựng xung quanh các component. Trong React, để xây dựng trang web chúng ta sử dụng những component, có thể tái sử dụng một component ở nhiều nơi với các trạng thái hoặc thuộc tính khác nhau và một component có thể chứa thành phần khác.

Mỗi component có một trạng thái riêng, có thể thay đổi và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Để khởi tạo Component chúng ta sẽ có nhiều cách như sau:

Dùng let, const trong JavaScript ES6 hoặc var ở các phiên bản JavaScript thấp hơn.

Dùng Function.

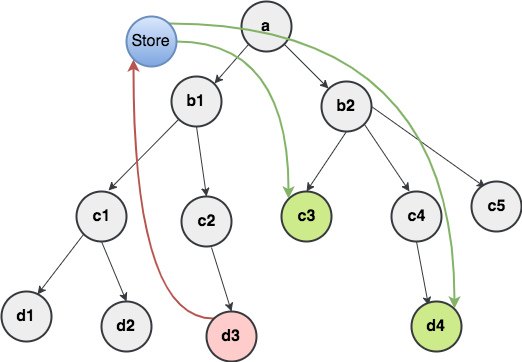
Dùng Class trong JavaScript ES6

**II. Tổng quan về Redux**

1. **Giới thiệu**

Redux là một công cụ quản lý dự đoán cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. Redux ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ Elm và kiến trúc Flux của Facebook. Do vậy Redux thường dùng kết hợp với React.

1. **Cách Redux làm việc**



Hình 1: Cách Vận hành của Redux

Cái cách mà Redux hoạt động là khá đơn giản. Nó có 1 store lưu trữ toàn bộ state của app. Mỗi component có thể access trực tiếp đến state được lưu trữ thay vì phải send drop down props từ component này đến component khác.

1. **Các thành phần của Redux**

Redux gồm 3 thành phần: Action, Store, Reducer.

* 1. **Actions**

Actions đơn giản là các events. Chúng là cách mà chúng ta send data từ app đến Redux store. Những data này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.

Actions được gửi bằng cách sử dụng store.dispatch() method, chúng phải có một type property để biểu lộ loại action để thực hiện. Chúng cũng phải có một payload chứa thông tin.

* 1. **Reducer**

Reducers là các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như những objects và chúng định rõ cách state của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store.

* 1. **Store**

Store lưu trạng thái ứng dụng và nó là duy nhất trong bất kỳ một ứng dụng Redux nào. Bạn có thể access các state được lưu, update state, và đăng ký or hủy đăng ký các listeners thông qua helper methods.

**III. Tổng quan về Redux-saga**

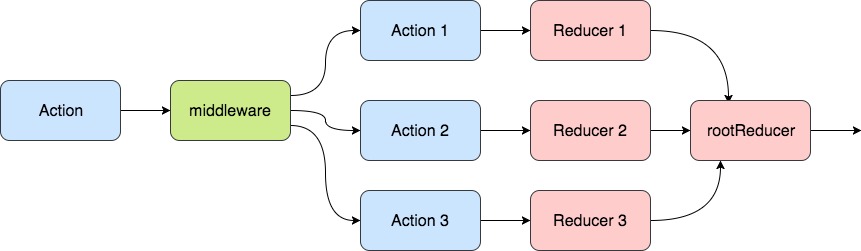
1. **Middleware**

Một ứng dụng thực tế thường đòi hỏi có những thao tác xử lý cần thời gian để phản hồi ( các thao tác bất đồng bộ lấy dữ liệu từ api hay các thao tác đọc ghi file hay đọc cookie từ trình duyệt, …). Các thao tác như vậy trong lập trình hàm gọi là side effects. Nhưng redux lại có một số ràng buộc:

Thứ nhất: Các xử lý trong Reducers phải là các hàm đồng bộ và pure, trả về state mới

Thứ hai: Reducers sẽ ko được sử dụng các hàm async vì không được thay đổi global state.

Do đó để có thể giải quyết được các side effect này ta cần thực hiện nó ở middeware.

1. **Middleware trong Redux**

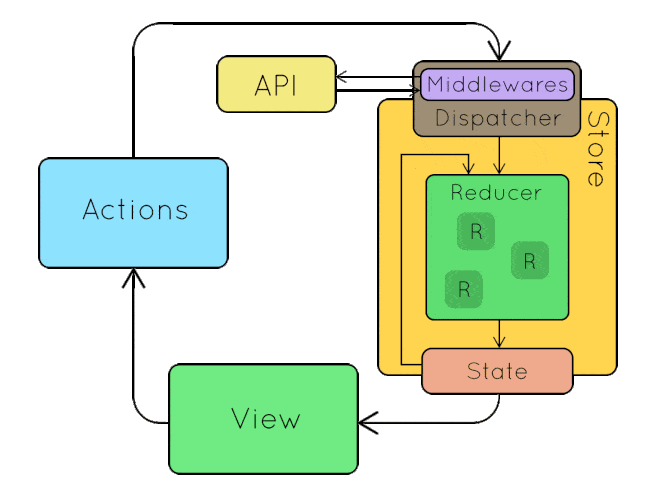
Hình 2: Middleware trong Redux

Trong Redux, middleware là một lớp nằm giữa Reducers và Dispatch Action, nó sẽ modify và được gọi trước khi action được dispatch. Hiện nay có khá nhiều thư viện middleware cho redux nhưng nổi tiếng và được sử dụng nhiều nhất đó là redux-saga, redux-thunk và redux-observable.

Cả 3 thư viện trên đều có thể giải quyết được hầu hết các trường hợp. Mặc dù redux thunk khá dễ hiểu và dễ code nhưng nó chưa thực sự mạnh trong một số tình huống, điều này được khắc phục với các thư viện mạnh mẽ hơn như redux saga và redux observale.

1. **Redux- Saga:**
   1. **Khái niệm**

Redux-Saga là một thư viện redux middleware, giúp quản lý những side effect (side effect là những hành động event có thể làm thay đổi DOM trong react components) trong ứng dụng redux trở nên đơn giản hơn. Bằng việc sử dụng tối đa tính năng Generators (function\*) của ES6, nó cho phép ta viết async code một cách đơn giản hơn.

* 1. **Vòng đời của Redux-Saga**

Hình 3: Vòng đời của Redux Saga

Khác với Redux-thunk, Readux-Saga cung cấp cho ta một số hàm helper effect, các hàm này sẽ trả về một effect object chứa đựng thông tin đặc biệt chỉ dẫn middeware của Redux có thể thực hiện tiếp các hành động khác. Các hàm helper effect sẽ được thực thi trong các generator function.

Một số helper trong redux saga:

takeEvery() : thực thi và trả lại kết quả của mọi actions được gọi.

takeLastest() : có nghĩa là nếu chúng ta thực hiện một loạt các actions, nó sẽ chỉ thực thi và trả lại kết quả của của actions cuối cùng.

take() : tạm dừng cho đến khi nhận được action

put() : dispatch một action.

call(): gọi function. Nếu nó return về một promise, tạm dừng saga cho đến khi promise được giải quyết.

race() : chạy nhiều effect đồng thời, sau đó hủy tất cả nếu một trong số đó kết thúc.

1. **Json-server**

Một công việc khá phổ biến đối với front-end developer là phải giả lập một backend REST service để nhận JSON data cung cấp cho ứng dụng front-end, và đảm bảo nó hoạt động như mong đợi trong khi đang chờ phía backend hoàn thiện service.

Bạn vẫn có thể cài đặt backend server đầy đủ, bằng cách sử dụng Node.js, Express và MongoDB, tuy nhiên việc này tốn khá nhiều thời gian và phức tạp. Trong khi đó JSON Server lại là một giải pháp khá hoàn thiện cho yêu cầu nhanh và đơn giản với đầy đủ các CRUD operation.

1. **React Router**
2. Khái niệm

React Router là một thư viện điều hướng tiêu chuẩn trong React. Nó giúp cho UI được đồng bộ với URL. Nó có API đơn giản nhưng mạnh mẽ, có thể giúp giải quyết được rất nhiều vấn đề.

1. **Axios**

Axios là một thư viện HTTP Client dựa trên Promise. Cơ bản thì nó cung cấp một API cho việc xử lý XHR (XMLHttpRequests). Trong bài viết này, chúng ta sẽ tìm hiểu làm thế nào để thực hiện một HTTP request bằng cách sử dụng axios nhé.

**CHƯƠNG 3: Phân tích hệ thống**

I. **Xác định các Use Case**

1.1. Tác nhân Khách hàng có các UC sau:

+ Xem thông tin sản phẩm

+ Xem thông tin giỏ hàng

+ Chọn sản phẩm cần mua

+ Đăng nhập để đặt hàng

+ Thêm, bớt sản phẩm trong giỏ hàng

+ Thực hiện đặt hàng

* 1. Tác nhân người quản lý có các UC sau:

+ Đăng nhập

+ Quản lý danh mục sản phẩm

+ Quản lý đơn hàng

+ Tiếp nhận đơn hàng

+ Thực hiện việc tư vấn cho khách hàng

1. **Đặc tả Use-Case**

**1. Tác nhân khách hàng**

1.1. **Xem thông tin sản phẩm**

a. Tác nhân: người quản lý, khách hàng

b. Mô tả: cho phép người quản lý, khách hàng xem thông tin về các sản phẩm có trong cửa hàng

c. Luồng sự kiện chính:

* + - Khách hàng xem danh sách sản phẩm hiển thị trên trang Web.
    - Người quản trị xem danh sách sản phẩm.

d. UC kết thúc.

* 1. **Thêm vào giỏ hàng**

1. Tác nhân: khách hàng
2. Mô tả: UC cho phép khách hàng đưa sản phẩm đã chọn vào giỏ hàng
3. Tiền điều kiện: sản phẩm đã được chọn, đã đăng nhập
4. Luồng sự kiện chính:

Chọn chức năng thêm vào giỏ hàng để đưa sản phẩm đã chọn vào lưu trữ trong giỏ hàng. Nếu chưa chọn sản phẩm mà nhấn nút thêm vào giỏ hàng thì qua f.

Hệ thống hiển thị thông tin sản phẩm đưa vào giỏ hàng

1. UC kết thúc
2. Chưa chọn sản phẩm cần mua

Hệ thống thông báo sản phẩm chưa được chọn

* 1. **Xác nhận đặt hàng**

1. Tác nhân: khách hàng
2. Mô tả: UC cho phép khách hàng tạo đơn hàng để tiến hành đặt hàng giỏ hàng sản phẩm
3. Tiền điều kiện: khách hàng đã chọn sản phẩm vào giỏ hàng
4. Luồng sự kiện chính:

Khách hàng nhập các thông tin : họ tên, số điện thoại, địa chỉ, thời gian xem hàng, để tiến hành đặt hàng.

Khách hàng nhấn đặt hàng tại giỏ hàng để tạo đơn hàng.

Đơn hàng được gửi cho Người quản trị.

1. UC kết thúc
   1. **Đăng nhập hệ thống**
   2. Tác nhân: khách hàng
   3. Mô tả: UC cho phép khách đăng nhập vào hệ thống
   4. Tiền điều kiện: khách hàng chưa đăng nhập vào hệ thống
   5. Luồng sự kiện chính:

Khách hàng truy cập vào Website.

Form đăng nhập hiển thị.

Nhập tên, mật khẩu vào form đăng nhập.

Hệ thống kiểm tra tên, mật khẩu của khách hàng.

Nếu việc đăng nhập thành công thì thông báo đăng nhập thành công. Nếu khách hàng nhập sai tên hoặc mật khẩu thì chuyển sang luồng rẽ.

* 1. UC kết thúc

1. Luống sự kiện rẽ nhánh:

Thành viên đăng nhập không thành công

Hệ thống thông báo quá trình đăng nhập không thành công

Chọn: đăng ký hay đăng nhập lại.

Hệ thống yêu cầu thành viên nhập lại tên và mật khẩu

Nếu khách xem đồng ý thì quay về bước d của luồng sự kiện chính, nếu không đồng ý thì UC kết thúc.

1. **Tác nhân Người quản trị**
   1. **Kiểm tra đơn hàng**
2. Tác nhân: Bộ phận kinh doanh
3. Mô tả: cho phép người quản trị kiểm tra và xác nhận thông tin đơn hàng
4. Luồng sự kiện:

Người quản trị kiểm tra đơn hàng

Hệ thống hiển thị thông tin đơn hàng

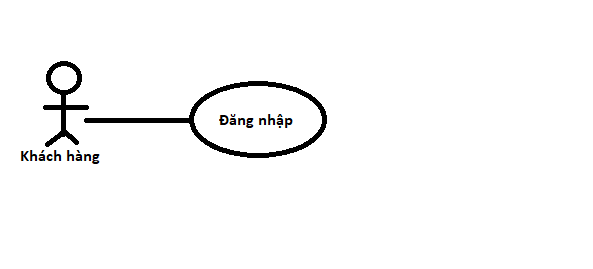
Người quản trị xác nhận thông tin trên đơn hàng

1. Kết thúc UC
   1. **Quản lí sản phẩm**
2. Tác nhân Người quản trị
3. Mô tả: cho phép người quản trị thêm, sửa, xóa sản phẩm hiên thị trên Website
4. Luồng sự kiện:

Người quản trị kiểm tra sản phẩm

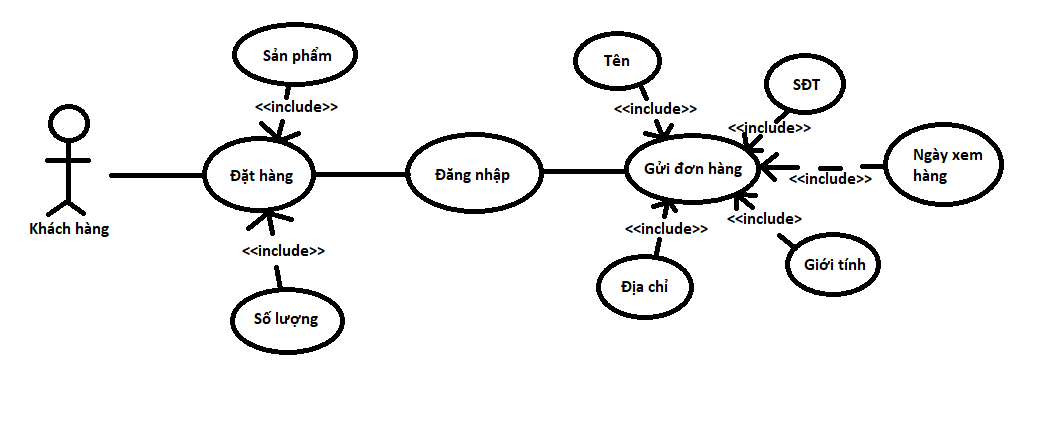
Thêm, sửa, xóa sản phẩm nếu cần.

Hệ thống thông báo thêm sửa xóa thành công.

1. UC kết thúc
2. **Biểu đồ UseCase**
3. **Tác nhân khách hàng**
   1. **Biểu đồ UseCase đăng nhập**

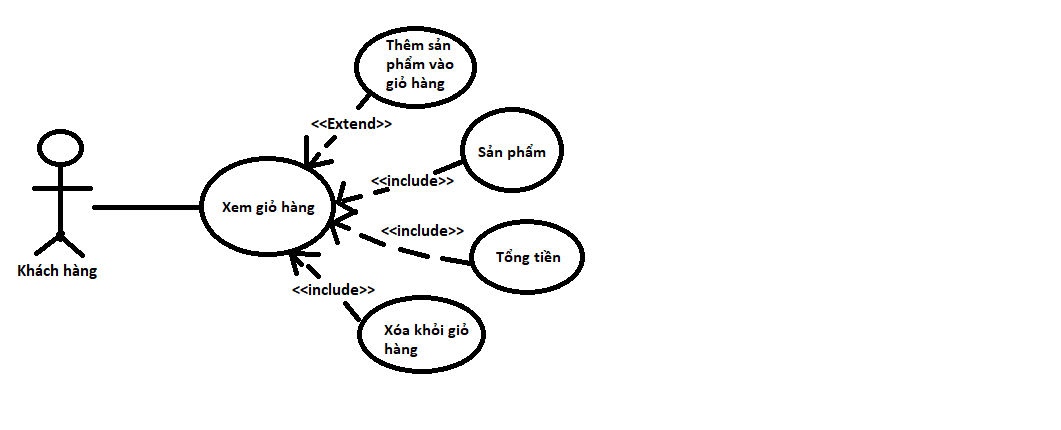
Hình 4: UseCase đăng nhập

* 1. **Biểu đồ UseCase đặt hàng**

****

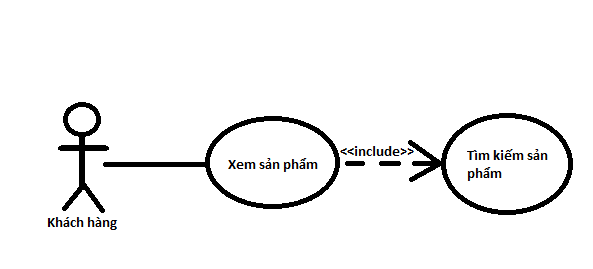
Hình 5: UseCase đặt hàng

* 1. **Biểu đồ UseCase Xem giỏ hàng**



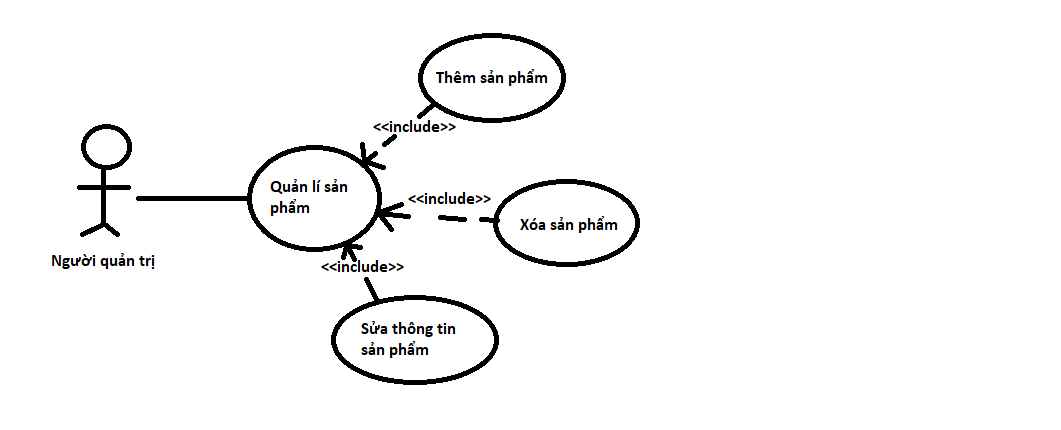
Hình 6: UseCase xem giỏ hàng

* 1. **Biểu đồ UseCase xem sản phẩm**

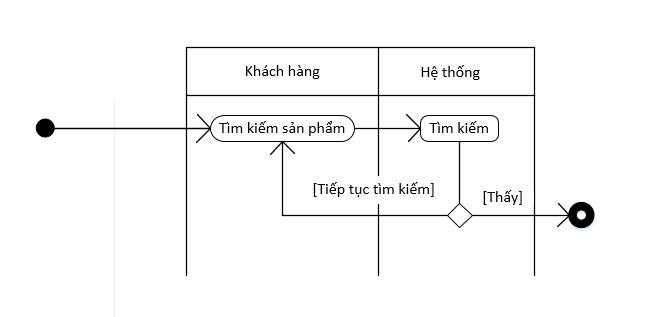


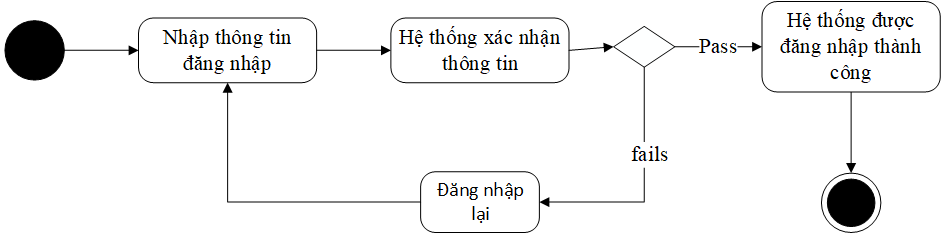
Hình 7: UseCase xem sản phẩm

1. **Tác nhận người quản trị**
   1. **Biểu đồ UseCase Quản lí sản phẩm**

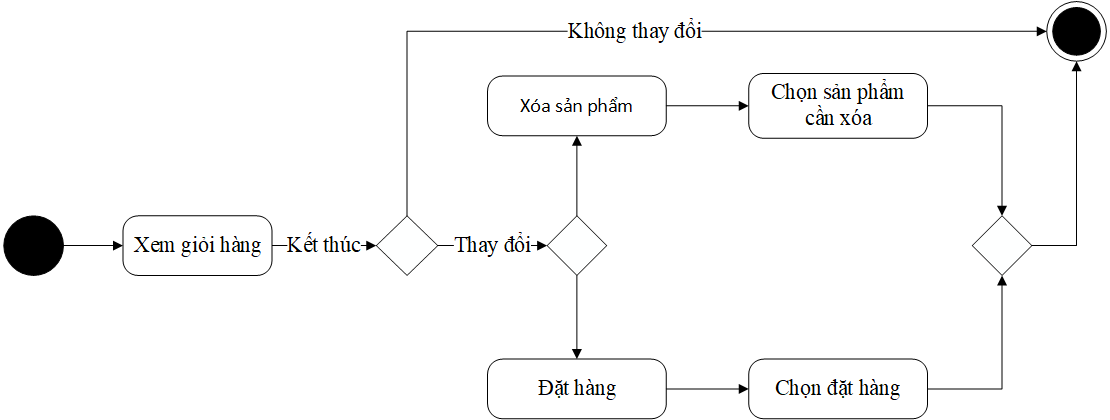


Hình 8: UseCase Quản lí sản phẩm

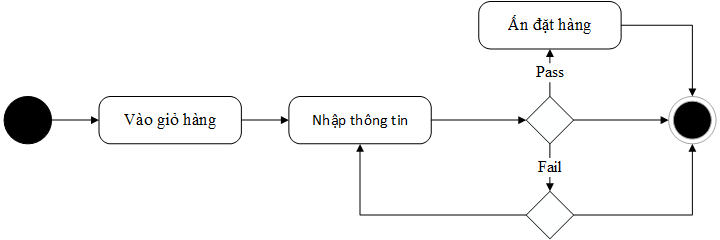
1. **Biểu đồ hoạt động (Activity)**
2. Tìm kiếm sản phẩm
3. Quản trị viên đăng nhập



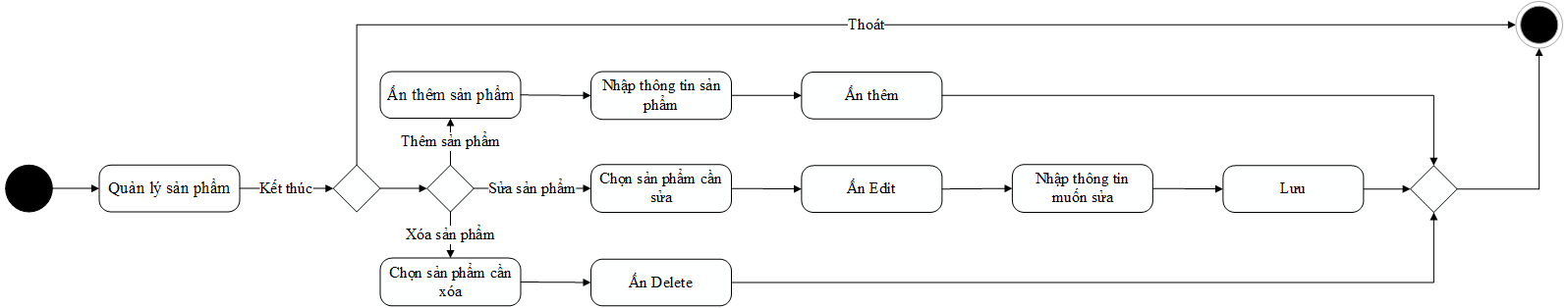
1. Xem giỏ hàng



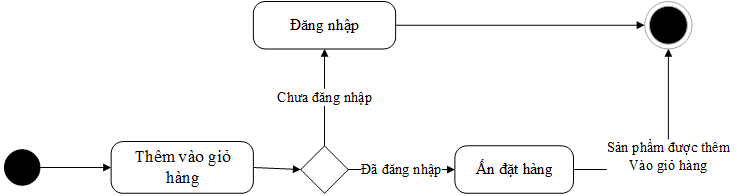
1. Đặt hàng



1. Quản lí sản phẩm



1. Thêm vào giỏ hàng



1. **Biểu đồ lớp**

|  |
| --- |
| Sản phẩm |
| Mã sản phẩm : int  Tên sản phẩm : char  Giá : int |
| PK: Mã sản phẩm |

|  |
| --- |
| Sản phẩm |
| Mã sản phẩm : int  Tên sản phẩm : char  Giá : int |
| PK: Mã sản phẩm |

|  |
| --- |
| Đơn hàng |
| Mã đơn hàng : int  Thông tin sản phẩm : char  Thông tin khách hàng : char |
| PK: Mã đơn hàng |
|  |

**CHƯƠNG 4: THIẾT KẾ VÀ CÀI ĐẶT**

1. **Thiết kế giao diện**
2. **Giao diên trang chủ**

**Kết luận**

* Qua quá trình tìm hiểu và xây dựng hệ thống website bán háng qua mạng, cộng thêm sự giảng dạy trên lớp của thầy Từ Thanh trí, chúng em đã phần nào hiểu được quy trình, cách thức xây dưng một hệ thống. Chúng em đã hiểu hơn về các biểu đồ trong ngôn ngữ UML như use case, biểu đồ hoạt động, biểu đồ trình tự, biểu đồ lớp, … và thấy được tầm quan trọng của các biểu đồ đó trong các dự án.
* Mặc dù đã hệ thống đã hoàn thành, nhưng nó vẫn còn rất nhiều mặt hạn chế và vấn đề tồn tại. Như là, mô tả các nghiệp vụ chưa đầy đủ, thiếu chính xác, cơ sở dữ liệu sơ xài và chưa được chuẩn hóa, giao diện web vẫn chưa thực sự hoàn thiện, ….
* Và trong thời gian tới thì chúng em sẽ cố gắng khác phục các hạn chế, những vân đề chưa được tốt, và bổ sung thêm những chức năng mới cho hệ thống.

**Tài liệu tham khảo**

* Website: http://www.uml-diagrams.org/
* Một số bài tiểu luận và các nguồn khác trên Internet