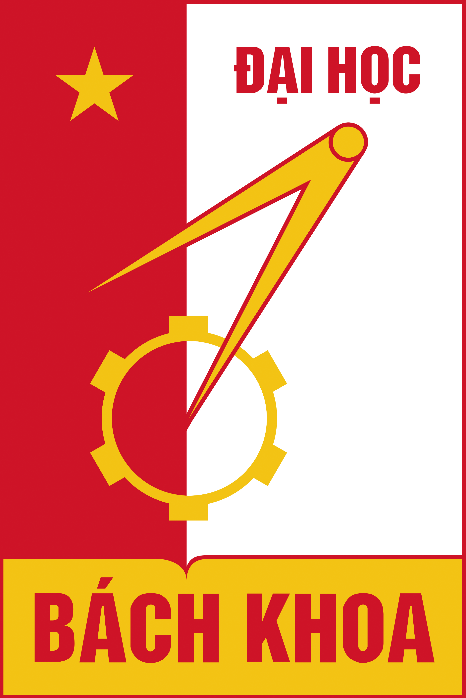
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

🙥🙦✯✯✯🙤🙧

****

**BÁO CÁO BÀI TẬP LỚN**

**Môn: LẬP TRÌNH CẤU TRÚC**

**Giảng viên hướng dẫn:** *TS. Vũ Thị Hương Giang*

**Nhóm:** *SP\_01*

**Sinh viên thực hiện:** *Trần Văn Thông - 20167386*

*Nguyễn Mạnh Tiến - 20164069*

*Bùi Ngọc Tú - 20164453*

*Trần Việt Hoàng - 20161721*

**Lớp:** *Application Specialist - K61*

**Mã lớp:** *110597*

*Hà Nội, ngày 23 tháng 12 năm 2019*

MỤC LỤC

[LỜI MỞ ĐẦU 3](#_Toc28124345)

[PHÂN CÔNG CÔNG VIỆC 4](#_Toc28124346)

[CÁC TÀI LIỆU LIÊN QUAN 4](#_Toc28124347)

[CHƯƠNG 1. MÔ TẢ YÊU CẦU 5](#_Toc28124348)

[1. Mô tả yêu cầu bài toán 5](#_Toc28124349)

[2. Các chức năng chính của trang web 5](#_Toc28124350)

[CHƯƠNG 2. ĐẶC TẢ YÊU CẦU 6](#_Toc28124351)

[1. Phân rã yêu cầu của trang web 6](#_Toc28124352)

[1.1. Theo hướng chức năng 6](#_Toc28124353)

[1.2. Theo hướng thuật toán 6](#_Toc28124354)

[1.3. Theo hướng đối tượng 7](#_Toc28124355)

[2. Module Giỏ hàng và Thanh toán 8](#_Toc28124356)

[CHƯƠNG 3. THIẾT KẾ CHI TIẾT - RAPID SOFTWARE DEVELOPMENT 9](#_Toc28124357)

[1. Các vòng phát triển: 9](#_Toc28124358)

[2. Mô tả các user story, task card, test card 9](#_Toc28124359)

[2.1. Vòng 1 9](#_Toc28124360)

[2.2. Vòng 2 11](#_Toc28124361)

[2.3. Vòng 3 13](#_Toc28124362)

[3. API Cung cấp 14](#_Toc28124363)

[CHƯƠNG 4. CẤU TRÚC CHƯƠNG TRÌNH 15](#_Toc28124364)

[1. Công nghệ sử dụng - Technology 15](#_Toc28124365)

[2. Mẫu thiết kế phần mềm - Software design pattern 15](#_Toc28124366)

[3. Áp dụng các nguyên tắc SOLID 17](#_Toc28124367)

[3.1. Single responsibility principle 17](#_Toc28124368)

[3.2. Open-Closed principle 17](#_Toc28124369)

[CHƯƠNG 5. TỔNG HỢP BÁO CÁO TUẦN 18](#_Toc28124370)

[1. Group assignment 1 18](#_Toc28124371)

[2. Group assignment 2 22](#_Toc28124372)

[3. Group assignment 3 26](#_Toc28124373)

[4. Group assignment 4 30](#_Toc28124374)

[5. Group assignment 5 33](#_Toc28124375)

[6. Group assignment 6 34](#_Toc28124376)

[7. Group assignment 7 36](#_Toc28124377)

# LỜI MỞ ĐẦU

Hiện nay, vấn đề áp dụng Công nghệ thông tin để xử lý thông tin trong công tác quản lý đã trở thành nhu cầu bức thiết, nhất là trong thời đại công nghệ 4.0 - thời đại Đất nước ta vươn mình đổi mới, đón đầu công nghệ.

Cùng với đó, xã hội ngày càng trở nên phát triển và hiện đại hơn với sự ứng dụng mạnh mẽ của Công nghệ thông tin. Sau cùng, nhu cầu của con người không ngừng tăng cao, mong muốn sự đa dạng, mới mẻ, mong muốn “công nghệ”. Vì thế, việc mua sắm vốn dĩ là nhu cầu thiết yếu của con người, nay mà được thực hiện online thì càng trở nên phong phú, phát triển, dễ dàng đáp ứng được nhu cầu của con người; trở thành bước tiến đáng ghi nhận.

Từ khi con người có thể dễ dàng sử dụng internet thông qua máy vi tính, di động, các thiết bị điện tử thì việc mua bán online (thương mại điện tử) ngày càng trở nên phổ biến và dần dần thay thế các cách bán hàng truyền thống. Vì vậy việc tham gia vào thị trường thương mại điện tử là hướng đi tương lai của nhiều cửa hàng đã và đang kinh doanh bằng các cách truyền thống.

Như vậy chúng ta có thể thấy được tầm quan trọng của thương mại điện tử, hay chính là những website bán hàng online. Nếu như cửa hàng đã có hệ thống chính quản lý sản phẩm, logistic, vận chuyển, sổ sách kinh doanh,… vậy làm sao để có thể xây dựng hệ thống bán hàng online tích hợp cho phù hợp với các thành phần đã có sẵn của cửa hàng. Đây cũng chính là bài toán mà nhóm **SP\_01** đề cập và giải quyết trong bản báo cáo này, đặc biệt với **module Giỏ hàng và Thanh toán**.

Bản báo cáo được xây dựng trên những kiến thức chúng em được dạy cũng như tự tìm hiểu được. Vì thế trong quá trình làm, nghiên cứu chắc chắn không tránh khỏi một số thiếu sót, mong cô góp ý thêm nhằm giúp cho báo cáo được hoàn thiện hơn nữa. Mọi ý kiến đóng góp xin gửi về hòm thư: [tien.nm164069@sis.hust.edu.vn](mailto:tien.nm164069@sis.hust.edu.vn).

Qua đây, nhóm chúng em cũng xin được gửi lời cảm ơn chân thành tới cô *Vũ Thị Hương Giang* (giảng viên bộ môn) đã giảng dạy, giải đáp các thắc mắc trong giờ học, cung cấp tài liệu tham khảo để chúng em có thể hoàn thiện bài tập và bản báo cáo này.

***Hà Nội, tháng 12 năm 2019***

# PHÂN CÔNG CÔNG VIỆC

Module: Quản lý giỏ hàng, thanh toán

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ tên** | **Công việc** | **Mức độ hoàn thành** | **Đóng góp tổng thể** |
| Trần Văn Thông | * Thiết kế vòng 1&2: test card cho chức năng giỏ hàng và thanh toán * Làm báo cáo tuần | 100% | 25% |
| Nguyễn Mạnh Tiến | * Lập trình module * Làm báo cáo tuần * Hiệu đính báo cáo tổng kết * Nộp tổng kết | 100% | 25% |
| Bùi Ngọc Tú | * Thiết kế vòng 1: user story, task card chức năng giỏ hàng * Làm báo cáo tuần | 100% | 25% |
| Trần Việt Hoàng | * Thiết kế vòng 2: user story, task card chức năng thanh toán * Làm báo cáo tuần | 100% | 25% |

# CÁC TÀI LIỆU LIÊN QUAN

* + - * Link github 3 vòng phát triển: <https://github.com/tientt-glose/wearShopSP01>
      * Video demo: <https://youtu.be/iAfmx7B9MmM>
      * Website demo:
        + Vòng 1&2: <http://shopsp01.herokuapp.com/>
        + Vòng 3: <http://shopsp-01.herokuapp.com/>
      * Slide báo cáo: <https://bit.ly/2t1s3XT>
      * Tổng hợp thông tin các module của toàn hệ thống: <https://bit.ly/371SHyK>
      * Hướng dẫn sử dụng: <https://bit.ly/34TGbQc>
      * Kế hoạch kiểm thử: <https://bit.ly/34Ve6YO>

# CHƯƠNG 1. MÔ TẢ YÊU CẦU

## Mô tả yêu cầu bài toán

* Xây dựng một trang web bán hàng online sử dụng cơ sở dữ liệu từ một hệ thống có sẵn.
* Trang web có các chức năng cơ bản của một web bán hàng online: Tìm kiếm, hiển thị sản phẩm, đặt hàng, theo dõi đơn hàng,…
* Trang web xây dựng theo phát triển phần mềm nhanh (Rapid software development) trong 3 vòng đời.

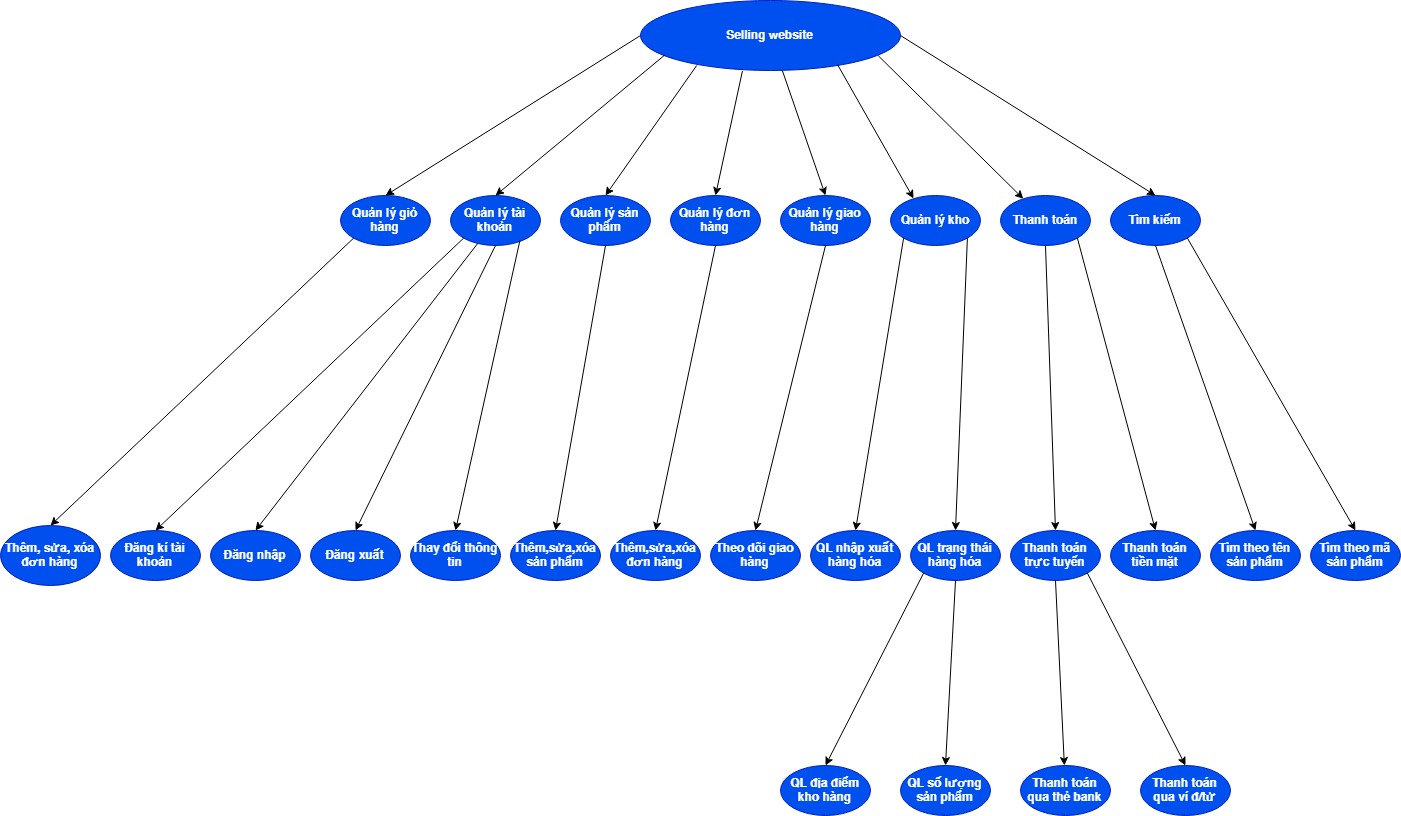
## Các chức năng chính của trang web

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Chức năng** | **Người thực hiện** | **Mô tả thêm** |
| 1 | Đăng nhập, Đăng xuất, Đăng ký | Khách hàng | Khách hàng có thể đăng nhập vào trang web bằng tài khoản họ đăng ký hoặc đăng nhập thông qua tài khoản mạng xã hội. |
| 2 | Tìm kiếm sản phẩm | Khách hàng | Khách hàng có thể tìm kiếm sản phẩm qua tên hoặc đặc tính |
| 3 | Lọc, sắp xếp, phân loại sản phẩm | Khách hàng | Khách hàng có thể xem sản phẩm được sắp theo các tiêu chí dựa trên đặc điểm, công dụng, đặc tính |
| 4 | Hiển thị sản phẩm | Hệ thống | Hiển thị thông tin chi tiết của sản phẩm |
| 5 | Quản lý giỏ hàng | Khách hàng | Khách hàng thêm mặt hàng cần mua vào giỏ hàng, chỉnh sửa số lượng mặt hàng trong giỏ, xóa mặt hàng từ giỏ hàng |
| 6 | Quản lý đơn hàng | Quản lý  Khách hàng | Tùy từng phân hệ người sử dụng mà sẽ có thẩm quyền khác nhau với đơn hàng |
| 7 | Quản lý sản phẩm | Quản lý | Các hoạt động CRUD sản phẩm |
| 8 | Quản lý khách hàng | Quản lý | Quản lý và xử lý tập các khách hàng; Lưu giữ phương thức thanh toán của khách hàng |
| 9 | Quản lý quảng cáo - khuyến mại | Quản lý | Chạy các banner, mã khuyến mại và quản lý trình trạng của chúng |
| 10 | Quản lý giao hàng | Quản lý | Quản lý, theo dõi việc vận chuyển từ lúc đặt hàng đến khi được ship đi; Đảm bảo sự cập nhật của người giao hàng cũng như thông báo tình trạng tới khách hàng. |
| 11 | Quản trị cấu hình | Quản lý | Quản lý bố trí, bố cục của toàn trang web cũng như các hoạt động CRUD tổng thể. |

# CHƯƠNG 2. ĐẶC TẢ YÊU CẦU

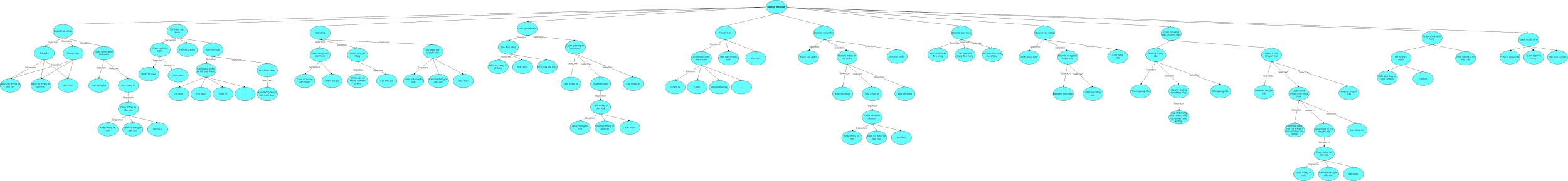
## Phân rã yêu cầu của trang web

### Theo hướng chức năng

****

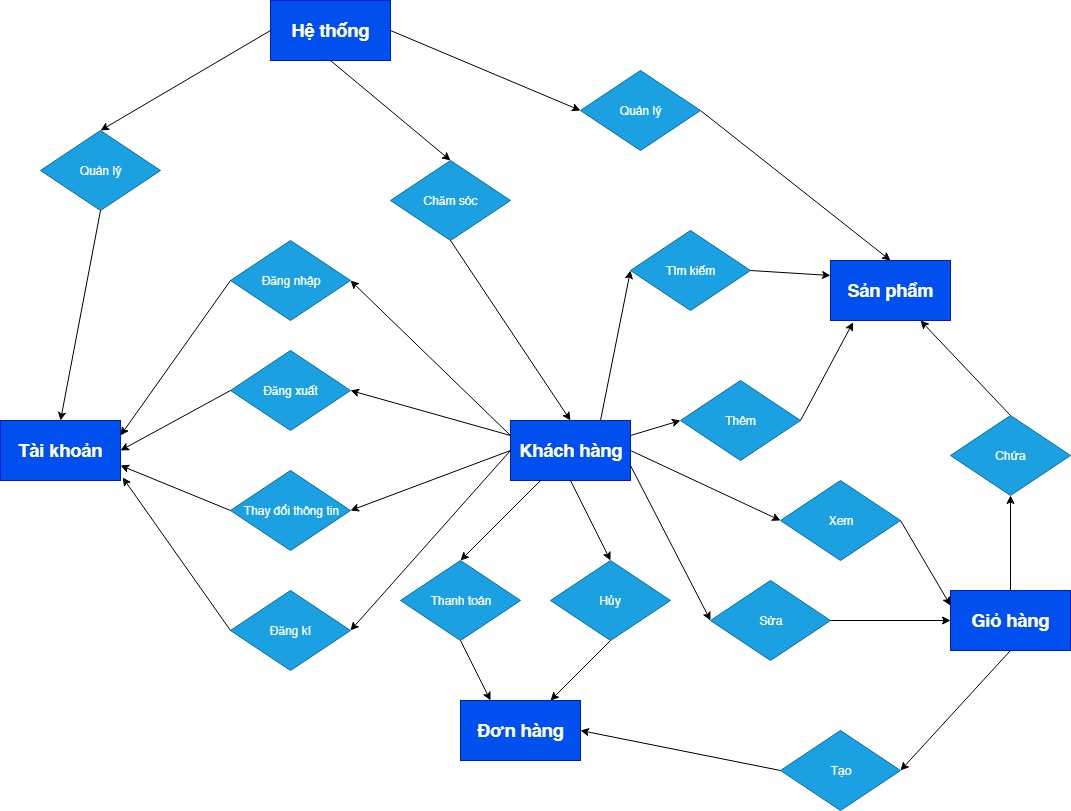
Link bản vẽ: <https://bit.ly/2VI1Kk9>

### Theo hướng thuật toán

****

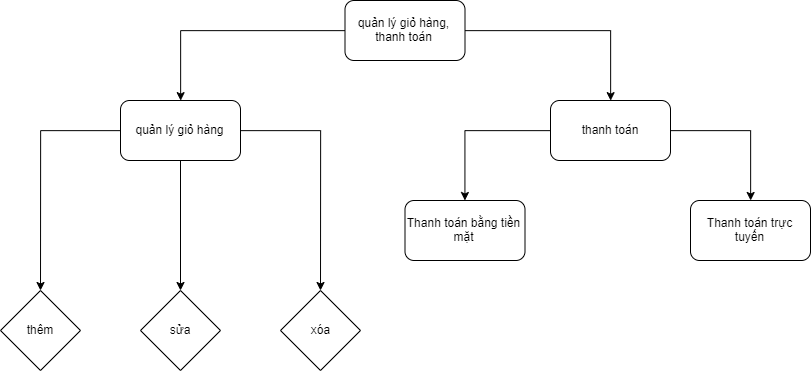
Link bản vẽ: <https://bit.ly/2VzeWY8>

### Theo hướng đối tượng

****

Link bản vẽ: <https://bit.ly/2Mw7x8b>

## Module Giỏ hàng và Thanh toán



Quy ước:

* Hình chữ nhật: Module
* Hình thoi: tính năng

Kỹ thuật module hóa theo chức năng của phần mềm, chia thành những module như thế này sẽ làm giảm độ phức tạp của phần mềm; giúp dễ dàng phân công các module cho các nhóm phát triển.

Các module được tạo ra trong phần mềm thực hiện một nhóm các tính năng và có thể được thay thế, chỉnh sửa dễ dàng mà không làm ảnh hưởng đến các phần còn lại của phần mềm, do đó thích hợp để đối phó với sự thay đổi. Mỗi module nhận đầu vào từ module khác và đầu ra đều là các dữ liệu API giúp các module dễ dàng giao tiếp với nhau.

Việc phân chia module như trên giúp các thay đổi đặc thù như: thêm chức năng áp dụng mã khuyến mại, chức năng tùy biến đơn vị vận chuyển,... chỉ làm thay đổi trên module đó và không làm thay đổi cấu trúc chương trình và không phải viết lại toàn bộ chương trình, dễ dàng cho việc phát triển và sửa đổi.

# CHƯƠNG 3. THIẾT KẾ CHI TIẾT - RAPID SOFTWARE DEVELOPMENT

## Các vòng phát triển:

Module Giỏ hàng và Thanh toán được thực hiện trong 3 vòng phát triển:

* Vòng 1: Phát triển các chức năng xem, sửa số lượng, xóa mặt hàng trong giỏ hàng;
* Vòng 2: Hiển thị kết quả thông tin giỏ hàng và yêu cầu các thông tin thanh toán, xử lý thanh toán; Xử lý mã khuyến mại
* Vòng 3: Hoàn thiện UI, refractor, kiểm thử, tích hợp API cùng các module của nhóm khác. Cụ thể là:

|  |  |  |
| --- | --- | --- |
| **TT** | **Task** | **Nội dung** |
| 1 | Tích hợp với nhóm 2 | Chức năng đăng nhập;  Lấy thông tin phương thức thanh toán |
| 2 | Tích hợp với nhóm 3 | Lấy thông tin sản phẩm |
| 3 | Tích hợp với nhóm 4 | Tạo đơn hàng |
| 4 | Tích hợp với nhóm 5 | Lấy mã khuyến mại |
| 5 | Tích hợp với nhóm 9 | Button "Thêm vào giỏ hàng" ở các ô sản phẩm |
| 6 | Tích hợp với nhóm 10 | Lấy thông tin đơn vị vận chuyển |

## Mô tả các user story, task card, test card

### Vòng 1

#### User story: Sử dụng giỏ hàng

Trong giai đoạn lựa hàng, khách hàng đã xem các sản phẩm, thêm các sản phẩm muốn mua vào giỏ hàng sau khi xem thông tin chi tiết mặt hàng đó. Người dùng truy cập phần giỏ hàng. Tại đây người dùng có thể thay đổi các thông tin về số lượng các sản phẩm có trong giỏ hàng trước khi tiếp tục thanh toán.

Các hành động của người dùng:

* Tại đây người dùng thực hiện các chức năng thêm mặt hàng mới vào giỏ hàng, thay đổi số lượng mặt hàng hoặc xóa mặt hàng khỏi giỏ hàng.
* Sau khi thay đổi thông tin giỏ hàng, người dùng có thể lựa chọn tiếp tục quay lại mua sắm hoặc tiến tới nhập thông tin cá nhân.

#### Task card

|  |  |  |
| --- | --- | --- |
| **TT** | **Task card** | **Nội dung** |
| 1 | Thêm sản phẩm vào giỏ hàng (thêm ngoài giỏ hàng) | Hệ thống thêm một sản phẩm vào giỏ hàng khi người dùng chọn thêm sản phẩm ở mục tìm kiếm |
| 2 | Thêm bớt sản phẩm ở giỏ hàng | Hệ thống sửa số lượng sản phẩm ứng với thao tác thêm bớt của người dùng |
| 3 | Hiển thị thông tin giỏ hàng | Hệ thống hiển thị thông tin giỏ hàng sau khi người dùng chọn thao tác giỏ hàng |
| 4 | Xóa sản phẩm khỏi giỏ hàng | Hệ thống loại bỏ sản phẩm khỏi giỏ hàng khi người dùng chọn xóa sản phẩm |

#### Test card

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Test Card** | **Mục tiêu** | **Bước thực hiện** | **Kết quả** | **Lỗi** |
| 1 | Thêm sản phẩm vào giỏ hàng | Kiểm tra xem đã thêm mặt hàng hay chưa sau khi người dùng chọn "thêm vào giỏ hàng" | Người dùng bấm nút "thêm vào giỏ hàng" phía dưới của mặt hàng | Số lượng sản phẩm của mặt hàng vừa chọn trong giỏ hàng tăng lên hoặc số lượng mặt hàng tăng lên | Không thêm được sản phẩm |
| 2 | Thêm sản phẩm vào giỏ hàng | Kiểm tra đã đúng Mặt hàng được thêm vào giỏ hàng hay chưa | Người dùng bấm nút "thêm vào giỏ hàng" phía dưới của mặt hàng | Trong danh sách các sản phẩm có ID của mặt hàng vừa được thêm | Thêm sai sai mặt hàng |
| 3 | Hiển thị chi tiết giỏ hàng | Hệ thống di chuyển đến trang thông tin giỏ hàng. | Người dùng chọn nút "Giỏ hàng" | Giỏ hàng hiển thị các mặt hàng mà người dùng đã chọn, bao gồm mặt hàng, số lượng, đơn giá và thành tiền. | Không hiển thị được giỏ hàng |
| 4 | Thêm, chỉnh sửa số lượng của sản phẩm trong giỏ hàng | Thay đổi số lượng của sản phẩm trong giỏ hàng | Người dùng chọn "sửa sổ lượng" hoặc "thêm số lượng" của sản phẩm trong giao diện giỏ hàng | Số lượng của sản phẩm thay đổi đúng theo người dùng đã chọn | Không chỉnh sửa được số lượng sản phẩm |
| 5 | Xóa mặt hàng khỏi giỏ hàng | Xóa mặt hàng khỏi giỏ hàng | Người dùng chọn "xóa mặt hàng" trong giao diện giỏ hàng | Xóa ID mặt hàng, số lượng,... của mặt hàng vừa chọn khỏi danh sách sản phẩm | Không xóa được mặt hàng |

### Vòng 2

#### User story: Sử dụng mã khuyến mại và Thanh toán

Các hành động của người dùng:

* Người dùng xác nhận lại thông tin chi tiết giỏ hàng (mặt hàng, đơn giá, thành tiền,...)
* Áp dụng mã khuyến mại (nếu có); Khách hàng có thể xóa mã khuyến mại đang áp dụng.
* Sau đó người dùng xác nhận lần cuối thông tin về đơn hàng. Lựa chọn hình thức thanh toán, thông tin, hình thức giao hàng.
* Tiến hành thanh toán.
* Nhận được phản hồi của hệ thống
* Kết hợp từ vòng 1, khách hàng đã có thể thực hiện đầy đủ quá trình mua hàng với việc thêm hàng vào giỏ, tương tác với giỏ và chuyển sang bước thanh toán với việc lựa chọn hình thức thanh toán, áp dụng mã khuyến mại.

#### Task card

|  |  |  |
| --- | --- | --- |
| **TT** | **Task card** | **Nội dung** |
| 1 | Chuyển sang trang thanh toán | Hệ thống hiển thị trang thanh toán khi người dùng thao tác "thanh toán" |
| 2 | Hiển thị kết quả áp dụng mã khuyến mại | Hệ thống hiển thị cập nhật về giá tiền sau khi người dùng áp dụng mã khuyến mại |
| 3 | Xóa mã khuyến mại | Tiến hành xóa mã khuyến mại khi được khách hàng yêu cầu |
| 4 | Hiển thị phương thức thanh toán | Hệ thống hiển thị các phương thức thanh toán khả dụng |
| 5 | Xác nhận phương thức thanh toán | Hệ thống nhận phương thức thanh toán từ thao tác người dùng |
| 6 | Lấy thông tin người dùng | Hệ thống nhận thông tin liên quan từ phía người dùng |
| 7 | Xác nhận thông tin người dùng | Hệ thống hiển thị thông tin 1 lần nữa để yêu cầu xác nhận từ người dùng, kiểm tra tính chính xác của thông tin |
| 8 | Tạo đơn hàng | Hệ thống đóng gói các sản phẩm trong giỏ hàng và thông tin liên quan thành đơn hàng |
| 9 | Reset giỏ hàng | Hệ thống xóa tất cả sản phẩm trong giỏ hàng |

#### Test card

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Test Card** | **Mục tiêu** | **Bước thực hiện** | **Kết quả** | **Lỗi** |
| 1 | Chuyển trang thanh toán | Chuyển sang giao diện thanh toán để người dùng thực hiện thao tác thanh toán | Người dùng chọn "thanh toán" | Thanh toán hiện lên các thông tin thanh toán và vận chuyển | Không chuyển được trang |
| 2 | Nhận mã khuyến mại | Nhận mã và giảm giá cho người dùng | Người dùng nhập mã khuyến mại | Kiểm tra và xác nhận mã khuyến mại. Nếu đúng thì áp dụng giảm giá hoặc ưu đãi cho khách hàng | Không nhập được mã khuyến mại |
| 3 | Nhận thông tin thanh toán và giao hàng | Lưu trữ và áp dụng thông tin thanh toán, giao hàng của khách hàng | Người dùng nhập và xác nhận thông tin | Xác nhận, lưu trữ và áp dụng thông tin của khách hàng | Không xác nhận được thông tin của khách hàng |
| 4 | Tạo đơn hàng | Đóng gói giỏ hàng và tạo đơn hàng | Người dùng chọn "Xác nhận đơn hàng" | Đóng gói thông tin của mặt hàng trong giỏ hàng, lưu trữ thông tin thanh toán và vận chuyển của khách hàng và đóng gói. | Không xuất được đơn hàng |
| 5 | Reset giỏ hàng | Sau khi đóng gói giỏ hàng cũ thành đơn hàng, tiến hành reset giỏ hàng | Sau khi xuất đơn hàng thành công | Xóa tất cả thông tin về các mặt hàng trong giỏ hàng như ID, số lượng,.. khỏi giỏ hàng | Không reset được giỏ hàng |

### Vòng 3

#### User story

Kết hợp từ vòng 1, vòng 2, các API của module nhóm khác, khách hàng đã có thể thực hiện đầy đủ quá trình mua hàng, thanh toán và việc bổ sung thêm việc lựa chọn đơn vị vận chuyển.

Quy trình tổng quát:

* Khách hàng xem thông tin chi tiết mặt hàng (bản thể hiện chi tiết của nhóm 9), bấm “Thêm vào giỏ”.
* Khách hàng bấm “Giỏ hàng” để xem thông tin trong giỏ hàng; Thực hiện các thao tác sửa số lượng, xóa, mua hàng tiếp hoặc thanh toán.
* Khách hàng chuyển sang bước thanh toán sẽ nhập các thông tin của người nhận.
* Lựa chọn đơn vị vận chuyển (dữ liệu đơn vị được lấy từ nhóm 10)
* Lựa chọn phương thức thanh toán (thẻ lưu sẵn từ nhóm 2; COD; khai thông tin thẻ tạm thời)
* Áp dụng mã khuyến mại (nếu có)
* Xác nhận đơn hàng

#### Task card

|  |  |  |
| --- | --- | --- |
| **TT** | **Task card** | **Nội dung** |
| 1 | Tích hợp với nhóm 2 | Chức năng đăng nhập;  Lấy thông tin phương thức thanh toán |
| 2 | Tích hợp với nhóm 3 | Lấy thông tin sản phẩm |
| 3 | Tích hợp với nhóm 4 | Tạo đơn hàng |
| 4 | Tích hợp với nhóm 5 | Lấy mã khuyến mại |
| 5 | Tích hợp với nhóm 9 | Button "Thêm vào giỏ hàng" ở các ô sản phẩm |
| 6 | Tích hợp với nhóm 10 | Lấy thông tin đơn vị vận chuyển |

#### Test card

Sử dụng các test card ở vòng 1 và vòng 2 với nguồn dữ liệu mới từ việc tích hợp với API của module nhóm khác.

## API Cung cấp

Link test: <http://shopsp-01.herokuapp.com/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **URI Pattern** | **Nội dung** | **Cung cấp cho** | **Thời điểm cung cấp** | **Mẫu JSON** |
| GET | /api/cart/{user\_id} | Xem giỏ hàng của người dùng cụ thể |  | Vòng 3 |  |
| PATCH | /api/cart/{user\_id} | Cập nhật số lượng giỏ hàng |  | Vòng 3 | {  "user\_id" : 2, "session\_id" : 123, "product\_id" : 17  "quantity" : 5 } |
| POST | /api/carts | Thêm vật phẩm mới vào giỏ hàng | Nhóm 9 | Vòng 3 | {  "user\_id" : 2, "session\_id" : 123, "product\_id" : 17 } |
| DELETE | /api/cart/{user\_id} | Xóa mặt hàng trong giỏ |  | Vòng 3 | {  "user\_id" : 2, "session\_id" : 123, "product\_id" : 17 } |

# CHƯƠNG 4. CẤU TRÚC CHƯƠNG TRÌNH

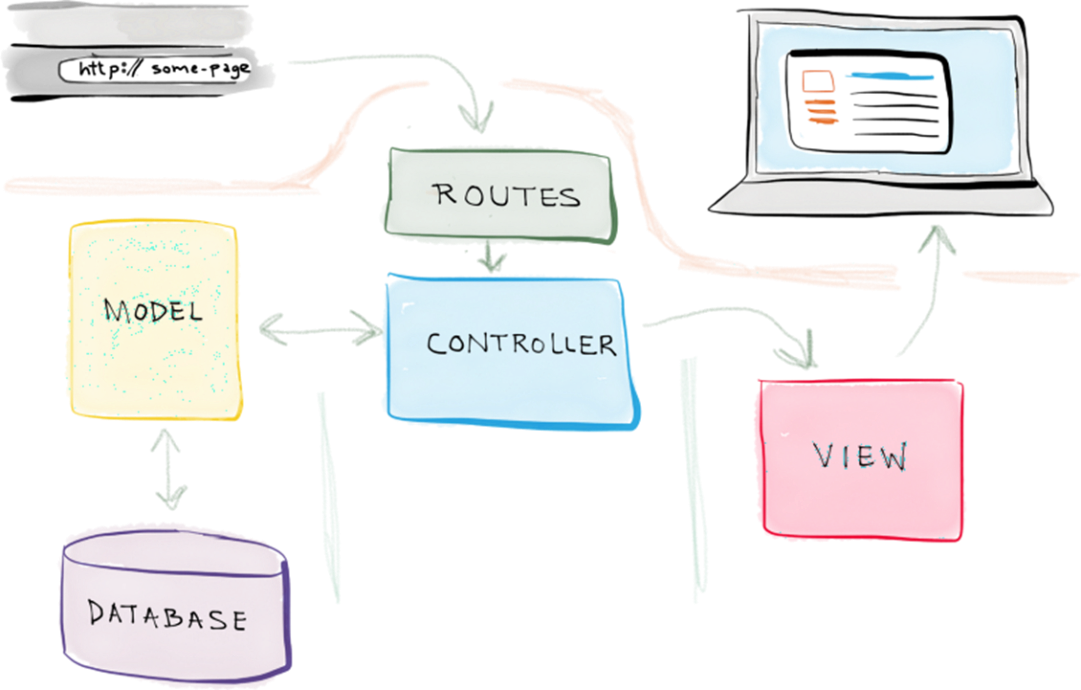
## Công nghệ sử dụng - Technology

Dựa vào yêu cầu từ bài toán nhóm quyết định xây dựng module dựa trên các nền tảng công nghệ sau:

* Ngôn ngữ: PHP, HTML, CSS, Javascript, Bootstrap
* Back-end Framework: Laravel 6
* Front-end: HTML, CSS, Javascript, Bootstrap
* Web server: Heroku
* Hệ quản trị cơ sở dữ liệu: MySQL

## Mẫu thiết kế phần mềm - Software design pattern

Hệ thống phát triển dựa trên mẫu thiết kế MVC



Cách triển khai:

* Tách riêng phần biểu diễn và phần tương tác ra khỏi dữ liệu hệ thống.
* Cấu trúc hóa hệ thống thành ba thành phần logic tương tác với nhau:
  + View component: định nghĩa và quản lý cách dữ liệu được biểu diễn tới người dùng như thế nào, quản lý dữ liệu nào được hiển thị cho người dùng. Mỗi khi controller xử lý ra được kết quả, sẽ kết xuất (render) ra view gửi về máy client, hiển thị tới người dùng.
  + Controller component: Quản lý tương tác người dùng (ví dụ như ấn phím, nhấp chuột, sử dụng các chức năng). Controller xử lý các yêu cầu của người dùng, bằng việc giao tiếp tương tác với model để lấy được dữ liệu từ database để xử lý, sau đó kết xuất kết quả thành view. Ngoài ra, yêu cầu của người dùng sẽ được tiền xử lý thông qua Routes rồi gửi tới Controller.
  + Model component: lưu trữ các thao tác tương tác với cơ sở dữ liệu (nhập, sửa, xóa, truy xuất thông tin), được quản lý bởi Controller.
* Ưu điểm : Cho phép dữ liệu thay đổi độc lập với việc biểu diễn của nó và ngược lại. Hỗ trợ biểu diễn theo nhiều cách khác nhau trên cùng một dữ liệu. Tránh để việc từ view tương tác trực tiếp tới cơ sở dữ liệu, việc tương tác thông qua controller đảm bảo việc vận hành ổn định và tránh các sai sót dữ liệu trực tiếp đến từ người dùng.
* Nhược điểm: MVC thường được sử dụng vào những dự án lớn. Do đó, với các dự án nhỏ, mô hình MVC có thể gây cồng kềnh, tốn thời gian trong quá trình phát triển cũng như thời gian trung chuyển dữ liệu.

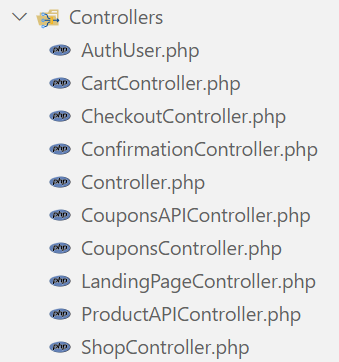
Kết quả thực tế:



## Áp dụng các nguyên tắc SOLID

### Single responsibility principle

Mỗi Controller chỉ đảm nhiệm một công việc liên quan duy nhất:



Các Controller phát triển chính, tiêu biểu của module:

* **CartController.php**: chịu trách nhiệm xử lý logic phần giỏ hàng
* **CheckoutController.php**: chịu trách nhiệm xử lý logic phần thanh toán

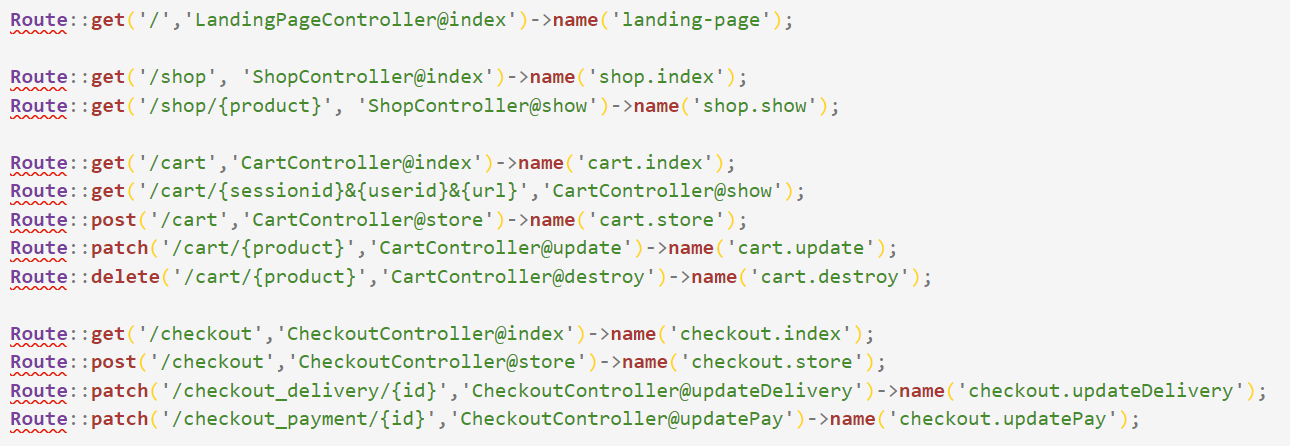
### Open-Closed principle

Sau này nếu phát triển thêm mới những xử lý logic mới, tính năng mới thì sẽ không thay đổi code sẵn có mà chỉ mở rộng và nạp chồng.

Ví dụ:

class NewCartController < CartController

Khi có nhu cầu thêm tính năng chỉ cần tạo controller, model, view mới và chỉnh sửa trong routes. Khi có nhu cầu thay thế controller này bằng controller khác, ta cũng chỉ cần chỉnh sửa trong routes.



# CHƯƠNG 5. TỔNG HỢP BÁO CÁO TUẦN

## Group assignment 1

* 1. **Identify the must-have functionalities of the selling website.**
* Dành cho khách hàng:
  + Tìm kiếm sản phẩm
  + Đặt hàng và quản lý đơn hàng (theo dõi tình trạng đơn hàng) của mình
* Dành cho admin:
  + Quản lý các thông tin về khách hàng, mà đăng ký tài khoản trên website.
  + Quản lý đơn hàng
  1. **How to deal with complexity in the following cases:**

1. Developing new selling website.

Xử lý được các phức tạp đến từ:

* + Độ phức tạp nhu cầu từ độ đa dạng của người dùng:
    - Đa dạng về mục đích sử dụng:
      * Khách hàng: mua hàng, tìm kiếm các mặt hàng và khảo giá, ...
      * Người quản lý (admin): có thể phân hóa rõ hơn theo loại hình quản lý (quản lý đơn hàng, quản lý khách hàng,...)
      * Kế toán: thực hiện quản lý tiền, tổng kết doanh thu qua các giai đoạn thời gian 🡪 làm các nghiệp vụ liên quan đến quản lý tài chính (doanh thu, lời lãi) khi trang web được vận hành
      * Thu ngân: thực hiện mua bán, kiểm tra, xác nhận và đảm bảo các giao dịch đã được hoàn tất
      * Quản kho: kiểm tra, đảm bảo cùng với các công tác quản lý khác về vấn đề lượng hàng thông báo trên website tới người dùng sao cho ánh xạ thực tế với lượng hàng trong kho là chính xác (điều này cực kỳ quan trọng nếu như website sẽ phát triển tính năng giao nhanh giống như TikiNow)
    - Đa dạng về độ tuổi: đa dạng về độ tuổi trong trường hợp này chủ yếu về đa dạng từ phía khách mua hàng (làm sao khi truy cập vào trang web, thì người dùng thấy được dải sản phẩm phù hợp với độ tuổi của mình)
    - Đa dạng về sở thích: đa dạng chủ yếu từ đa dạng sở thích của khách mua hàng
    - Đa dạng về trình độ: nhân viên mới và khách hàng mới có thể khó sử dụng hay phải làm quen dần dần. Vậy thì xử lý phức tạp này sao cho, người dùng hay nhân viên có thể làm quen một cách nhanh nhất mà trang web vẫn phải đầy đủ các tính năng hiện đại, cập nhật. (ví dụ như: ở Nhật Bản, có một công ty IT thiết kế riêng một trang thương mại điện tử dành riêng cho người lớn tuổi)
    - Đa dạng về mức độ phức tạp nhu cầu: có những người chỉ đơn giản là mua hàng và đợi ship COD sau 2-3 ngày và họ thanh toán đơn hàng; nhưng cũng có những người đòi hỏi phải giao nhanh và họ có thể thanh toán bằng các dịch vụ thanh toán điện tử từ bên thứ ba, hay thêm các ghi chú về ngày giờ nhận hàng phù hợp.
  + Mức độ phức tạp ngay trong các chức năng:
    - Một số chức năng như gợi ý sản phẩm cho khách hàng, một đơn hàng có nhiều sản phẩm được ship theo nhiều hình thức khác nhau, có thể rất phức tạp và khó thực hiện.
* Độ phức tạp xuất phát từ việc chuyển hóa từ công việc kia vào phần mềm
  + Độ phức tạp khi lĩnh hội, nắm rõ được nhu cầu công việc:
    - Khó tiếp cận được đối tượng: khách hàng có thể khảo sát ở các trang web khác nhưng để tiến hành khảo sát cũng tốn chi phí
    - Đối tượng quá đa dạng: mặt hàng kinh doanh càng đa dạng, đối tượng khách hàng càng đông thì đối tượng càng đa dạng
  + Độ phức tạp cài đặt của các nhu cầu: một số nghiệp vụ cần sự linh hoạt như kế toán thì rất khó nắm bắt hết các kịch bản sử dụng vì thế phải thiết kế các công cụ sao cho vừa linh hoạt vừa có mức logic sát nhất với nghiệp vụ của họ
  + Mức độ phổ biến của nhu cầu:
    - Một website bán hàng là khá phổ biến và có rất nhiều công cụ cũng như các website bán hàng đã có sẵn nên phần công việc đã được hỗ trợ đủ cả.
* Vì thế cần thiết kế phần mềm theo cách tiếp cận là thiết kế có tính cấu trúc cao
* Decomposition: Từ một chức năng lớn tổng thể (must-have functionalities) như đặt/mua hàng ta phân thành các chức năng nhỏ hơn và có thể lập trình một cách độc lập (module) và chia làm sao cho để đảm bảo khi gộp lại ta tốn ít công sức nhất.
* Abstraction: các chức năng có các tính chất một phần nào đó giống nhau thì tạo một lớp cha đại diện, các chức năng (lớp con) sẽ kế thừa và có thể sử dụng những phần chung đó 🡪 điều này tận dụng hiệu quả việc tái sử dụng code và hỗ trợ sửa chữa.
* Module&Hierarchy: Từ chiến lược trên, ta tiến hành một cách thực tế là các chức năng được viết theo từng module (giao cho được từng người trong team) và phân cấp các chức năng (các mức kế thừa (chức năng C, D kết thừa từ chức năng B và B kế thừa từ chức năng A)
* Pattern: Sử dụng mô hình MVC(Model - view - controller) phân tách thành 3 phần riêng biệt dễ quản lý là database, repositories và controller giao tiếp vs client.

Ví dụ: Trong chức năng mua hàng sẽ có chuỗi các hành động xảy ra và lấy đơn cử là ship hàng khi khách hàng đặt hàng thành công là một chức năng được chia ra (Decomposition) từ việc mua hàng. Vậy ship hàng nói chung sẽ là một abstract, và trong đó sẽ có các hình thức ship như là vận chuyển nhanh (trong 2 giờ), vận chuyển COD, vận chuyển thông thường (sau 2-3 ngày). Mọi thứ đều được tiến hành lập trình theo từng module và phân cấp chúng.

1. Activating the website in the whole system configuration.

* Độ phức tạp xuất phát chủ yếu từ việc tích hợp website vào với hệ thống

1. Reusing a search engine and a member administration module of the main system.

* Độ phức tạp từ việc search engine và member administration của hệ thống chính được tổ chức như thế nào (áp dụng các chiến lược như Decomposition, Abstraction hay chỉ là viết dành riêng cho hệ thống)
* Trong hai trường hợp b và c này, đều có sự liên quan đến hệ thống. Nếu như chúng ta sử dụng một chiến dịch thiết kế cấu trúc chuẩn giống như việc giải quyết phức tạp của một website thì các chức năng giống của website cũng có thể kế thừa (reuse) lại phần mã nguồn của hệ thống chính. Từ đó, công đoạn tích hợp vào toàn hệ thống cũng giảm đi đáng kể. Ngoài ra, việc có một cơ sở dữ liệu hợp lý (chuẩn hóa, tránh dư thừa, đảm bảo tốc độ truy xuất) thì việc tương tác, truyền thông tin giữa website và hệ thống chính (thực tế ở kho) sẽ trở nên hiệu quả hơn.
  1. **How to deal with changes in the following cases:**

a. Developing new selling website.

b. Activating the website in the whole system configuration.

c. Reusing a search engine and a member administration module of the main system.

Cần xác định rõ thay đổi là thay đổi cái nào?

* **Các thay đổi về quy mô, điều kiện của project:** Ta xác định xem chúng ta mong muốn phát triển thành một thương mại điện tử lớn (a complex sale) với tất cả sản phẩm trên thế giới hay chỉ đơn giản là phân phối sản phẩm của chính công ty mình. 🡪 Ta chọn ra một vòng đời của phần mềm phù hợp (vẫn đủ các bước nhưng sẽ tập trung thời gian vào các bước thiết yếu tùy theo điều kiện của project).
* **Các thay đổi về yêu cầu và công nghệ:** Ta chắc chắn điều này sẽ luôn xảy ra với website này và từ đó ta nên chọn một quy trình phát triển phần mềm không tuyến tính như Agile (Scrum),...
* **Các thay đổi từ các bên thứ ba (stakeholder):** Ta phải xác định các bên tác động tới việc làm project này (hệ thống chính, lãnh đạo công ty). Ví dụ như tiền đầu tư vào dự án bị thay đổi (tăng hoặc giảm) ta phải có những sự thay đổi liên quan như quy mô project, yêu cầu và công nghệ để thích ứng với sự tăng giảm đó.
  1. **Identify the quality indicators of the website in the following cases:**

1. Developing new selling website.
2. Activating the website in the whole system configuration.
3. Reusing a search engine and a member administration module of the main system.

Phần mềm đang xét là một website bán hàng, bởi vậy mục tiêu đề ra bao gồm 3 mục lớn:

* Tăng trưởng số lượng bán ra.
* Tăng tỉ lệ chuyển đổi từ khách vãng lai, khách chỉ khảo giá thành khách hàng thân thiết (tăng số lượng khách hàng trung thành sử dụng website để mua hàng)
* Tăng khả năng hỗ trợ bán hàng (hỗ trợ khách hàng trước, trong và sau dịch vụ (giao dịch), hỗ trợ tìm đối tượng khách hàng).

Một sản phẩm phần mềm có thể được đánh giá từ 3 góc nhìn:

* Khách hàng
* Người quản lý (tham gia hoặc không tham gia vào quá trình phát triển)
* Đội ngũ phát triển

Vì một phần mềm coi như một “hộp đen” với người dùng cuối (giao diện website là bề nổi giao tiếp trực tiếp với người dùng cuối), các tiêu chí đánh giá chất lượng nhìn chung của các góc nhìn này là như nhau cho cả 3 trường hợp: Phát triển một phần mềm mới (Developing new selling website), Thiết lập cho một nền tảng có sẵn (Activating the website in the whole system configuration) hay Tái sử dụng lại những gì đã có (Reusing a search engine and a member administration module of the main system).

* Góc nhìn khách hàng:
* Các tiêu chí nhất định phải có đối với một website:
  + Nội dung phù hợp, có ích.
  + Khả năng tìm kiếm nhanh chóng.
  + Điều hướng tiện lợi.
  + Thông tin đầy đủ, rõ ràng.
  + Nếu yêu cầu cung cấp thông tin thì cần có tính bảo mật cao.
  + Các chức năng dễ sử dụng.
* Các tiêu chí nhằm nâng cao chất lượng một trang web:
* Giao diện thân thiện, bắt mắt.
* Tốc độ tải trang nhanh.
* Thời gian vận hành thường xuyên (luôn khả dụng một cách kịp thời)
* Có hệ thống tư vấn, thanh toán (đối với website bán hàng).
* Không có quảng cáo.
* Các tiêu chí trên đối với khách hàng đều sẽ được kiểm chứng bằng mắt và trải nghiệm khi mua hàng trên website.
* Góc nhìn quản lý

Website cần đáp ứng tiêu chí duy nhất của doanh nghiệp là thúc đẩy lợi nhuận, dù bằng hình thức trực tiếp hay gián tiếp.

* Các tiêu chí mang lại lợi nhuận trực tiếp:
* Tăng số lượng bán ra.
* Giảm thiểu chi phí quản lý.
* Thời gian, chi phí phát triển và bảo trì.
* Phương thức đánh giá đơn giản thông qua số lượng thống kê, KPI, báo cáo tình hình kinh doanh của công ty.
* Các tiêu chí mang lại lợi nhuận gián tiếp:
  + Tối ưu về SEO - Search Engine Optimization (hình ảnh doanh nghiệp dễ tìm đến với khách hàng).
  + Tỉ lệ khách hàng trung thành tăng (dễ dàng trong việc định hướng kinh doanh)
* Sử dụng các tool như Google Analytics để có thống kê số lượng khách hàng.
* Góc nhìn phát triển:
  + Tốc độ phát triển
  + Số lượng dòng code
  + % code được test
  + Technical Debt
  + Số lượng bugs
  + Closing Tickets
  + Số lượng chức năng
  + Khả năng mở rộng
  + Độ tin cậy
  + Tính chính xác
  + Hiệu suất
  + Dễ bảo trì
  + Dễ thích ứng
  + Dễ tương tác
  1. **Describe a measurement plan for the whole system that are suitable for the following cases:**

1. Developing new selling website.
2. Activating the website in the whole system configuration.
3. Reusing a search engine and a member administration module of the main system.

MEASURE SOFTWARE QUALITY INDICATORS = METRICS + METHODS

Chất lượng phần mềm được xác định qua thông số và phương thức là để xác định, tận dụng những thông số đó nhằm nâng cao chất lượng. Đây là nguyên tắc quan trọng nhất trong mọi phương thức đánh giá.

Các bước để đánh giá chất lượng một sản phẩm phần mềm:

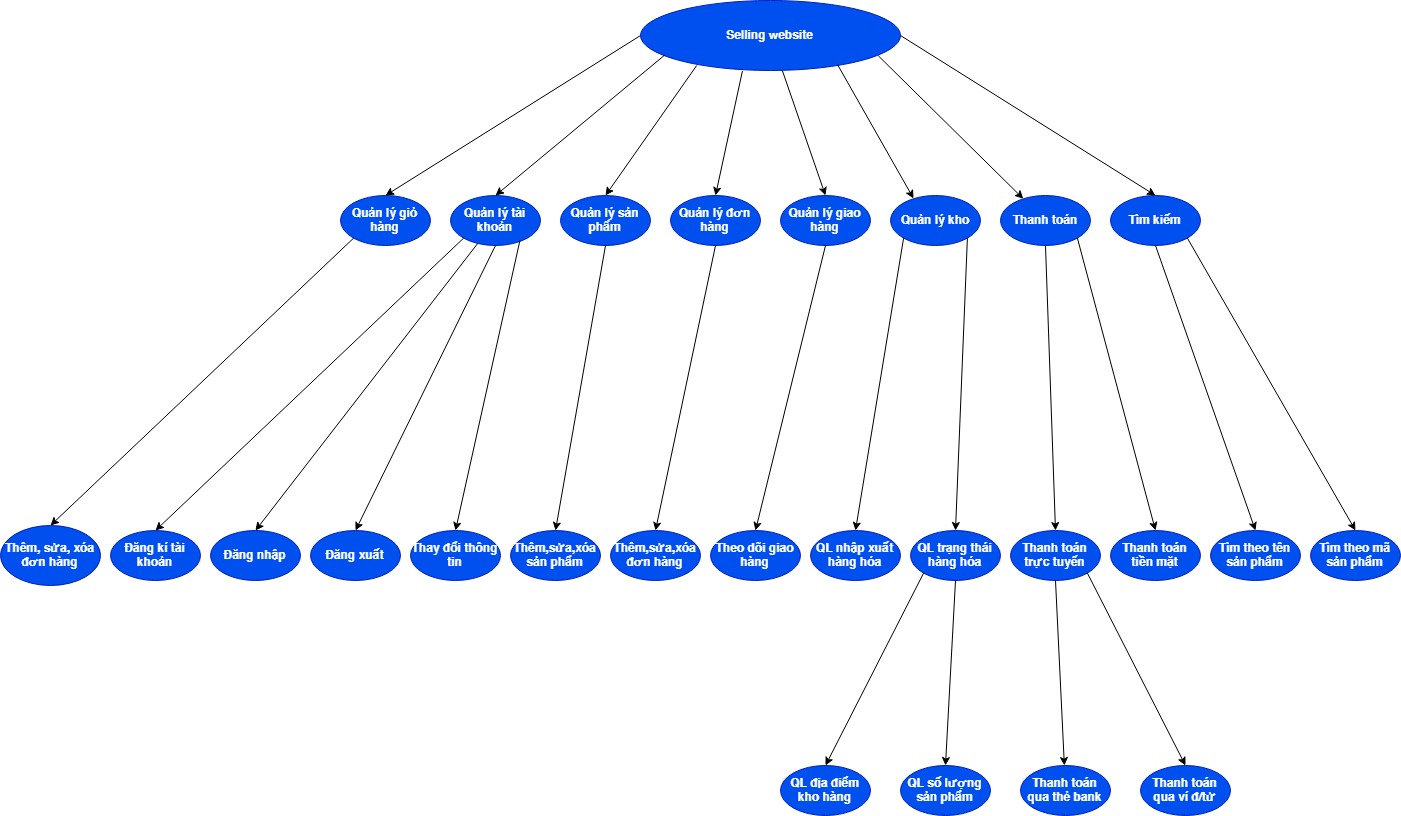
* Bước 1: Xác định mục tiêu mà phần mềm hướng tới. (Sử dụng S.M.A.R.T - Specific, Measurable, Attainable, Relevant, Timely để đánh giá). Chất lượng của một sản phẩm sẽ phụ thuộc vào tỉ lệ % đạt được mục tiêu ban đầu, cũng là tiêu chí để đánh giá những thiếu sót.
* Bước 2: Xác định những tiêu chí cần thiết, thu thập thông số, chỉ số của những tiêu chí đấy.
* Bước 3: Đánh giá chất lượng. Tùy thuộc vào góc nhìn mà chất lượng có thể kiểm định bằng mắt, bằng cảm nhận hoặc qua những phần mềm thống kê.

## Group assignment 2

* 1. **Please help users in clarification of the website’s core functionalities**
* Khách hàng:
  + Tìm kiếm sản phẩm: Khách có thể tìm kiếm sản phẩm theo tên, loại sản phẩm, màu sắc sản phẩm (xanh, đỏ, tím vàng, …), kích cỡ (S, M, L, XL, …), phong cách (công sở, đi dạo, party, thu đông, …), loại hàng (mới nhập, like new, cũ,…)
  + Đặt hàng và quản lý đơn hàng (theo dõi tình trạng đơn hàng) của mình: Người dùng có quyền biết được điều gì đang được thực hiện với đơn hàng của mình như vị trí (kho, shipper đã nhận, đang được ship,…), các thông tin đã xác nhận của đơn hàng (cách chứng thực với khách hàng là đơn hàng không có sai sót thông tin).
* Admin:
  + Quản lý thành viên website: người mua hàng sau khi đăng ký tài khoản, họ có thể đăng nhập vào tài khoản, trở thành thành viên của website, những người này có thể gửi các thắc mắc, các góp ý về website cũng như sản phẩm, việc của người quản lý nhân viên website là trả lời các thắc mắc của khách hàng, bên cạnh đó cũng tổng hợp lại các góp ý, đánh giá về website và sản phẩm để làm báo cáo lên công ty về nhiều mặt như lượng khách hàng thành viên, lượng đơn từ họ,….
  + Quản lý dữ liệu hàng hóa và quản lý đơn hàng: dựa vào tổng kết đánh giá của khách hàng về sản phẩm hiện có của website cũng như qua các bản điều tra thị trường thì công ty sẽ quyết định nên thay đổi cơ sở dữ liệu của họ như thế nào, người quản lý và chịu trách nhiệm về vấn đề này là người quản lý dữ liệu của hàng hóa trên website với những cập nhật mới nhất về việc đơn hàng này đã tiêu thụ bao nhiêu mặt hàng về số lượng và cả chủng loại
  1. **Create the image of software quality by nominating the assessment criteria of website (common qualities, process qualities, application – specific qualities)**
* Common qualities:
  + Tính chính xác (correctness): Các chức năng phải hoạt động một cách chính xác, nhất là các tính năng về tìm kiếm, đặt hàng, thanh toán.
    - Ví dụ: Khi người dùng nhập vào thanh tìm kiếm sản phẩm mà họ muốn tìm với đa dạng cách tìm kiếm khác nhau đã liệt kê ở trên, đứng trên phương diện người lập trình, kết quả trả về phải là những sản phẩm đúng theo nội dung yêu cầu tìm kiếm. Tuy nhiên, nhiều lúc khách hàng cũng chưa miêu tả đúng được sản phẩm họ muốn tìm kiếm nên có thể họ chưa hài lòng với kết quả này. Để đảm bảo tính chính xác của website thì người lập trình sẽ tạo thêm các kết quả gợi ý cho yêu cầu chính, ví dụ như đưa ra các sản phẩm có nét tương đồng với sản phẩm được yêu cầu tìm kiếm, hoặc gợi ý về những sản phẩm đang được khách hàng yêu thích của website.
  + Tin cậy (reliability): Độ tin cậy của website dựa trên nhiều yếu tố. Đầu tiên là độ hoạt động của website - hệ thống đảm bảo hoạt động 24/7/365, nếu có sự cố thì hệ thống sẽ được refresh và hoạt động lại trong vòng 30p, hạn chế trong vòng 3 tháng chỉ được lỗi hệ thống 1 lần (hạn chế tối đa lỗi sập hệ thống xảy ra). Tiếp đến là khi khách hàng yêu cầu tìm kiếm một sản phẩm, hệ thống có đưa ra các sản phẩm gợi ý, nhưng không quá nhiều, không làm phân tâm khách hàng, để khách hàng tập trung vào những sản phẩm mà họ muốn tìm kiếm (tránh gây cho khách hàng cảm giác “đẽo cày giữa đường” khi có quá nhiều mặt hàng lan man như mời chào, gây khó chịu và thiếu đáng tin). Khi khách hàng thanh toán chi phí sản phẩm với công ty cần đảm bảo tính bảo mật và có lợi nhất đối với khách hàng (ví dụ: không thu thêm phụ phí ngân hàng khi chuyển khoản, các hình thức khuyến mại đối với các loại/hạng thẻ riêng biệt, đa dạng hình thức thanh toán…)
  + Robustness: hệ thống dù hoàn thiện đến đâu nhưng vẫn có thể có lỗi xảy ra, hệ thống sẽ không vì những lỗi này mà ngưng hoạt động toàn bộ, chỉ có một hoặc một vài chức năng liên quan của hệ thống bị hỏng và cần bảo trì, thời gian bảo trì kéo dài lâu nhất 30p kể từ khi phát hiện lỗi, trong quá trình tạm khóa chức năng đó để bảo trì thì các chức năng khác của hệ thống vẫn hoạt động bình thường. Đồng thời, các chức năng đang bảo trì phải được thông báo kịp thời tới người dùng.
  + Hiệu năng (performance): Đáp ứng được việc nhiều người dùng một lúc, nhiều giao dịch song song tồn tại và được xác nhận. Thời gian phản hồi một yêu cầu của hệ thống phải nhanh, kịp thời (lý tưởng là không quá 2 giây), hệ thống đảm bảo làm việc tốt trên nguồn tài nguyên được cung cấp, dễ mở rộng quy mô trang web.
  + Khả năng sử dụng (usability): Hệ thống rất đề cao tiện ích của người dùng (khách hàng dễ dàng chọn 🡪 mua được sản phẩm, theo dõi tình trạng đơn hàng,…)
    - Ví dụ: Khi khách hàng tìm kiếm sản phẩm thì hệ thống sẽ đưa ra thêm những gợi ý liên quan đến sản phẩm khách hàng muốn tìm (có thể dựa trên sự đánh giá các sản phẩm được đánh giá cao từ người dùng liên quan đến sản phẩm đó), như thế, nếu trong những kết quả tìm kiếm chính không sản phẩm khách hàng mong muốn thì họ có thể dựa vào gợi ý của hệ thống để tìm được một sản phẩm khác ưng ý, như vậy người dùng đã giảm được đáng kể công việc nhập yêu cầu của họ và cảm thấy thú vị hơn, thử nghĩ mà xem, khi bạn đi mua hàng, bạn không thể miêu tả đúng sản phẩm mà bạn cần, hoặc là bạn không có chủ ý mua gì đó, chỉ đơn thuần là đi xem hàng, thì hệ thống lại đưa ra cho bạn, dẫn bạn đến với sản phẩm mà bạn mong muốn, thật tiện lợi, mà được đánh giá cao của người dùng khác.
  + Khả năng tái sử dụng (reusability): Trang web có thể tái sử dụng một số phần cho các dự án khác.
  + Khả năng bảo trì: hệ thống phải được thường xuyên bảo trì dù có phát sinh lỗi hay không
  + Tính đa nền tảng và khả năng chạy tức thì: hệ thống sẽ chạy tốt trên các hệ điều hành khác nhau như window hoặc Linux, MacOs, android (khách hàng có thể mua hàng trên cả điện thoại, miễn là điện thoại của họ có thể kết nối internet). Hệ thống cũng chạy tốt trên nhiều môi trường phần mềm, ví dụ như các trình duyệt web phổ biến hiện nay đảm bảo tiện dụng nhất đối với người sử dụng.
  + Tính dễ hiểu (understandability): hệ thống rất dễ hiểu, mỗi chức năng trong hệ thống sẽ có hình minh họa một cách phù hợp, hạn chế dùng các lời nói phức tạp để miêu tả chức năng của hệ thống,
    - Ví dụ: Trang homepage thì sẽ có biểu tượng hình ngôi nhà, khi muốn mua đồ thì sẽ có biểu tượng hình giỏ đồ, khi tìm kiếm sẽ có biểu tượng kính lúp cạnh ô tìm kiếm. Hệ thống đảm bảo người lần đầu sử dụng hệ thống cũng không gặp vấn đề khó khăn gì, và dễ dàng làm quen với hệ thống sau thời gian ngắn
* Process qualities:
  + Hiệu suất (productivity): hệ thống hoạt động liên tục nhưng phải có khả năng tương tác mạnh mẽ với thời gian phản hồi nhanh chóng. Từ đó, sau một khoảng thời gian được khách hàng sử dụng sẽ đem lại sự hài lòng cho họ (mua nhanh – mua mọi lúc, mọi nơi – mua chuẩn)
  + Kịp thời (timeliness): phát hành sản phẩm đúng thời gian người dùng yêu cầu. Thường thì một sản phẩm phần mềm sẽ phát triển chậm hơn so với yêu cầu của người sử dụng, vì thế sẽ xuất hiện sự không phù hợp giữa yêu cầu của người sử dụng đối với trạng thái hiện tại của sản phẩm phần mềm, trong trường hợp này, cách giải quyết là bảo trì để nâng cấp hệ thống, cứ 3 tháng sẽ bảo trì nâng cấp 1 lần, nếu yêu cầu của người dùng thay đổi nhiều sẽ giảm thời gian bảo trì xuống còn 2 đến 1 tháng để đảm bảo rút ngắn khoảng cách giữa yêu cầu của người dùng với trạng thái của hệ thống.
  + Trực quan (visibility): Mọi chức năng, bước phát triển, trạng thái, dự định thay đổi của trang web đều được ghi rõ ràng trong tài liệu và phải có sự đồng ý của cả 2 bên: bên phát triển website và bên công ty yêu cầu.
* Application – specific qualities
  + Bảo mật (security): bảo mật thông tin khách hàng, đơn hàng và khâu thanh toán…
    - Ví dụ:
      * Hệ thống sẽ liên kết với một số ngân hàng lớn để đảm bảo việc thanh toán bằng thẻ tín dụng của khách hàng được thuận lợi và an toàn.
      * Hệ thống sau mỗi lần bảo trì nâng cấp sẽ xóa bỏ những tài khoản đã lâu không hoạt động, những tài khoản đã có hành vi lừa đảo thì sẽ bị liệt vào danh sách đen của công ty.
  + Chất lượng thời gian phản hồi (time real): Trang web trả lại kết quả sau khi người dùng thực hiện 1 thao tác trong một thời gian nhất định. Đồng thời, hệ thống sẽ ghi lại thời gian khách hàng chọn, nhận, thanh toán một sản phẩm, đảm bảo thông tin chi tiết về các mốc thời gian quan trọng này một cách chính xác nhất
  1. **Write down the user’s requirements on the following aspects of website: functional, Quality of Service, policy, system and organizational**
* Chức năng: Trang web phải cung cấp đầy đủ các chức năng mà người dùng yêu cầu, đặc biệt là các chức năng chính cơ bản (must have) của một web bán hàng online (như đã được đề cập ở Assignment 1 và làm rõ hơn ở phần 1 của tài liệu này). Tuy nhiên, theo thời gian, yêu cầu về chức năng của người dùng sẽ trở nên cao hơn.
  + Ví dụ:
    - Từ việc tìm kiếm đơn thuần, người dùng sẽ muốn xem thêm về thông tin chi tiết của sản phẩm có phù hợp với cách phối đồ, dáng người phù hợp,…
    - Người dùng cũng muốn biết website hiện tại đang có những sự kiện kỷ niệm, khuyến mại gì, khi mua hàng ở website nhiều lần có đặc quyền đặc lợi gì hay không, trong thời gian tới hệ thống có những sản phẩm mới nào. 🡪 Những nhu cầu này buộc hệ thống chúng ta phải đáp ứng và tính toán, xử lý nhiều hơn.
    - Hệ thống nên có thêm chức năng đánh giá xếp hạng cho từng sản phẩm chứ không chỉ là đánh giá cho website.
    - Người dùng có thể hay không xem lại mặt hàng đã xem của mình tại website….
* QoS: Dịch vụ được cung cấp bởi trang web được cập nhật liên tục, tốc độ cao và an toàn cho người sử dụng.
* Chính sách:
  + Thông tin về người dùng phải được giữ bí mật an toàn, không sử dụng trái phép trên các hệ thống khác. Thông tin của khách hàng cũng cần phải đáng tin cậy và xác thực cao để tránh việc mất tài khoản hay tài khoản ảo, đơn hàng ảo.
  + Chất lượng sản phẩm được thể hiện trong trang web phải tương đương với chất lượng của sản phẩm thật.
* Hệ thống và tổ chức: Trang web phải có cấu trúc mạch lạc, có tính thống nhất giữa trang web mới và hệ thống bán hàng trước đó.
  + Ví dụ: Sử dụng một ngôn ngữ thống nhất, ưu tiên tiếng Việt Nam, ngay cả những chỗ nhỏ nhất như tìm kiếm mặc dù người Việt Nam vẫn thường nót Search (sớt). Việc sắp xếp menu chức năng của trang web nên sắp xếp từ trái qua phải, từ chức năng chung đến chức năng phân hóa hơn. Ví dụ: “trang chủ, quảng cáo, sự kiện ưu đãi, thông tin cá nhân,…” là thứ tự sắp xếp khá phù hợp và được ưa chuộng hiện nay.
  1. **Separate the concerns accordingly to the following criteria: quality, time, view and size**
* Quality:
  + Tính chính xác
  + Khả năng sử dụng
  + Tin cậy
  + Bảo mật
* Time:
  + Pha tiền xử lý: thông tin nhận vào để xử lý có khả năng xử lý được hay không, thao tác đấy có thực hiện được không, có gây ra lỗi hệ thống hay không?
  + Pha xử lý: thời gian xử lý mất bao lâu?
  + Pha hậu xử lý: kết quả trả về có làm hài lòng người dùng hay không?
* Size: Nhóm các chức năng tương tự lại với nhau:
  + Nhóm 1: chức năng tìm kiếm, chức năng xem thông tin chi tiết của sản phẩm
  + Nhóm 2: chức năng mua hàng, chức năng thanh toán, chức năng sự kiện website
  + Nhóm 3: chức năng xem thông tin cá nhân, chức năng xem lịch sử mua hàng
  + Nhóm 4: chức năng quản lý người dùng, chức năng quản lý thông tin sản phẩm lọc, sắp xếp sản phẩm; hiển thị sản phẩm; giỏ hàng thành chức năng sản phẩm để giảm độ phức tạp của hệ thống.
* View:
  + Điều khiển: người dùng sử dụng các chức năng trong hệ thống
  + Xử lý: hệ thống kiểm tra xem chức năng có hoạt động bình thường không bằng cách chạy các chức năng và show kết quả ra màn hình.

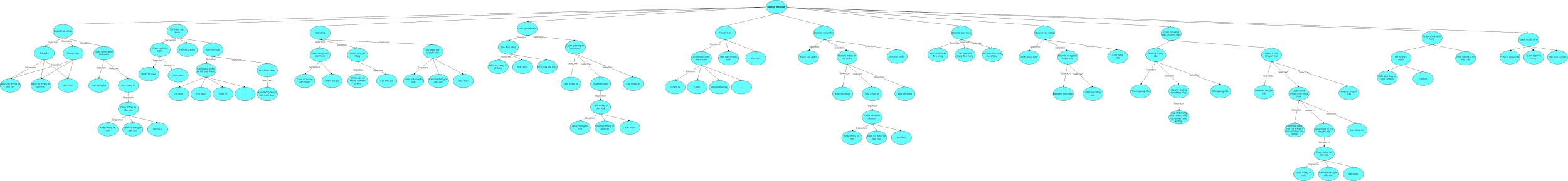
## Group assignment 3

* 1. **Decompose the selling website (i.e. online selling software) using three approaches: functional, algorithmic and object-oriented.**
     1. **Functional**

****

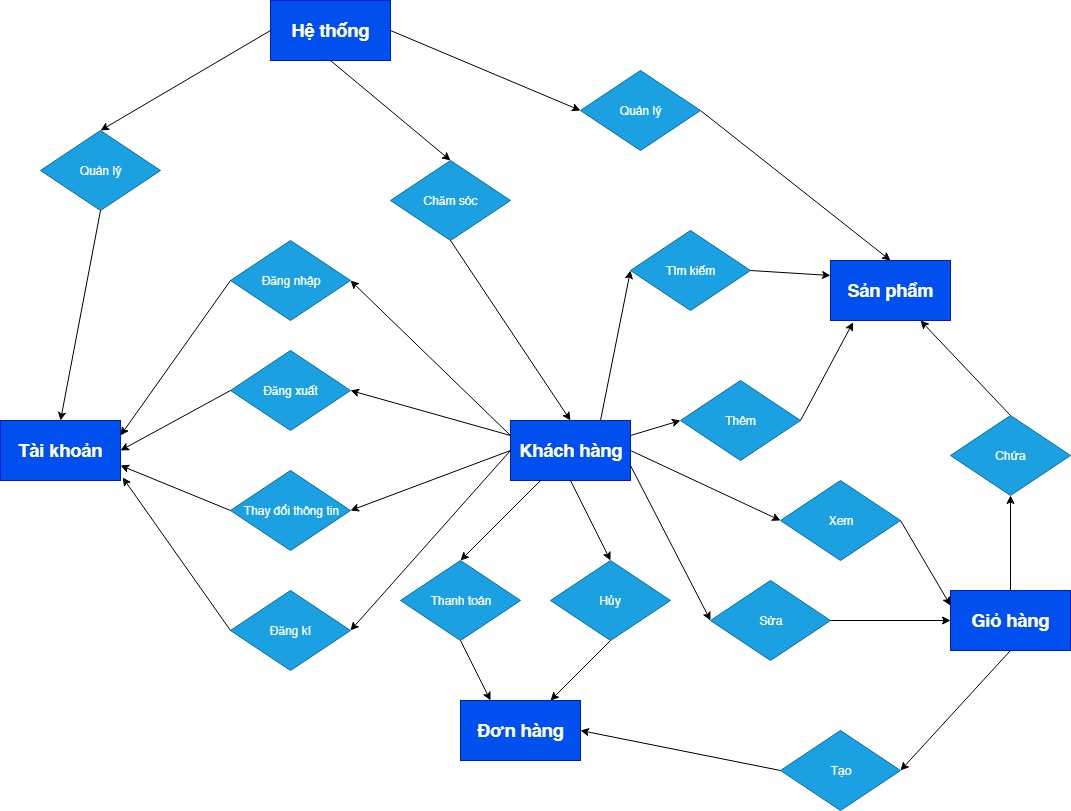
**Link bản vẽ:** <https://bit.ly/2VI1Kk9>

* + 1. **Algorithmic**

****

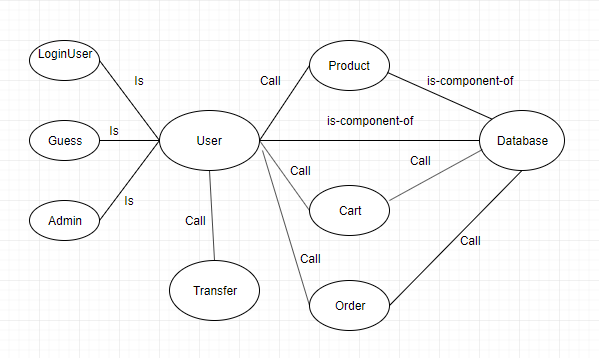
**Link bản vẽ:** <https://bit.ly/2VzeWY8>

* + 1. **Object-oriented**

****

**Link bản vẽ:** <https://bit.ly/2Mw7x8b>

* 1. **Apply the modularization techniques for designing the online selling software. Point out the modularization techniques that are suitable to cope with the changes identified in the 1st in-class assignment. Explain.**
* Trong hệ thống website bán hàng, có các module tương ứng với các module interface như sau :
* User : là một interface cung cấp các chức năng cơ bản cho các nhóm người dùng : Guess, Admin và LoginUser
* Product : là một interface cung cấp các thông tin về sản phẩm cho người dùng, cho phép thay đổi, chỉnh sửa thông tin sản phẩm đối với admin.
* Cart : là một interface cung cấp các chức năng quản lý giỏ hàng cho User.
* Order : là một interface cung cấp chức năng tra cứu và tìm kiếm cho người dùng, chỉnh sửa hoặc xóa thông tin đối với admin
* Transfer: interface cung cấp các chức năng chuyển hàng cho người dùng, chỉnh sửa thông tin đối với admin.
* Payment: interface cung cấp các chức năng thanh toán đối với người dùng.
* Database: interface cung cấp các chức năng quản lý dữ liệu website/kho hàng đối với admin, tra cứu các thông tin cơ bản đối với người dùng.
* Mối quan hệ giữa các interface được biểu diễn qua biểu đồ sau:



* Point out the modularization techniques that are suitable to cope with the changes identified in the 1st in-class assignment. Explain.

Để đối phó với sự thay đổi trong:

* Phát triển trang web bán hàng mới: Sử dụng software generation ( interface prototyping tools) : Trước tiên xây dựng một mẫu thử dựa trên những module cơ bản và quan trọng. Sau đó tiến hành phát triển dựa trên khuôn mẫu sẵn có.

Ưu điểm: Phát triển nhanh, dựa trên mẫu thử đã có nên giảm chi phí phát triển.

Nhược điểm: Trong trường hợp mẫu thử được xây dựng không đáp ứng nhu cầu cần xây dựng lại -> tốn chi phí và thời gian

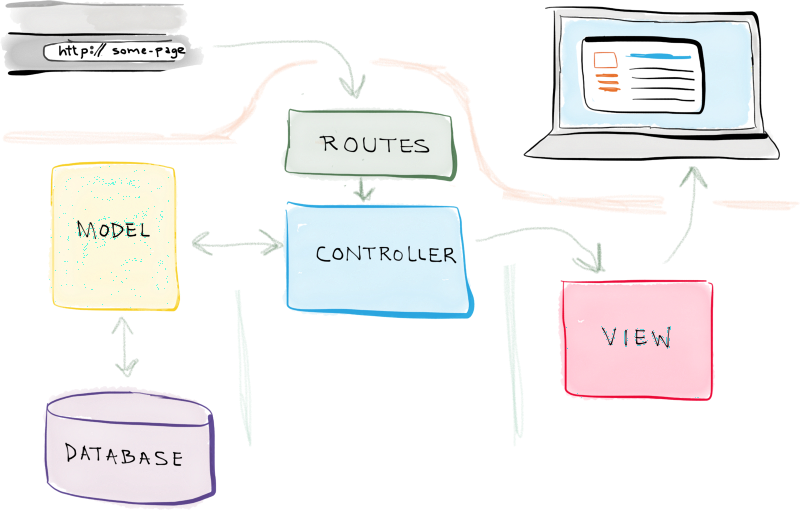
* Kích hoạt trang web trong toàn bộ cấu hình hệ thống: Sử dụng conditional compilation. Tùy thuộc vào cấu hình hệ thống khác nhau, phần mềm sẽ được xây dựng các phiên bản tương ứng và được lựa chọn khi cài đặt.

Ưu điểm: tăng trải nghiệm người dùng, tương thích với nhiều phiên bản/cấu hình khác nhau.

Nhược điểm: tốn tài nguyên khi phải xây dựng trước những phiên bản khác nhau. Trong trường hợp xuất hiện những yêu cầu mới cần phải xây dựng thêm những phiên bản khác.

* Sử dụng lại một công cụ tìm kiếm và một module của hệ thống chính: Sử dụng constant configuration. Định nghĩa trước những thuộc tính cần thiết để sẵn sàng cho những thay đổi khi cần để áp dụng cho từng mục đích khác nhau ( reusability)
  1. **Point out a software architecture that is convenient to build the online selling software. Explain.**

Kiến trúc phần mềm thuận lợi để xây dựng phần mềm/website là MVC(Model-View-Controller)



* Cách triển khai:
  + Tách riêng phần biểu diễn và phần tương tác ra khỏi dữ liệu hệ thống.
  + Cấu trúc hóa hệ thống thành ba thành phần logic tương tác với nhau:
    - View component: định nghĩa và quản lý cách dữ liệu được biểu diễn tới người dùng như thế nào, quản lý dữ liệu nào được hiển thị cho người dùng. Mỗi khi controller xử lý ra được kết quả, sẽ kết xuất (render) ra view gửi về máy client, hiển thị tới người dùng
    - Controller component: Quản lý tương tác người dùng (ví dụ như ấn phím, nhấp chuột, sử dụng các chức năng). Controller xử lý các yêu cầu của người dùng, bằng việc giao tiếp tương tác với model để lấy được dữ liệu từ database để xử lý, sau đó kết xuất kết quả thành view
    - Model component: lưu trữ các thao tác tương tác với cơ sở dữ liệu (nhập, sửa, xóa, truy xuất thông tin), được quản lý bởi Controller.
* Ưu điểm : Cho phép dữ liệu thay đổi độc lập với việc biểu diễn của nó và ngược lại. Hỗ trợ biểu diễn theo nhiều cách khác nhau trên cùng một dữ liệu. Tránh để việc từ view tương tác trực tiếp tới cơ sở dữ liệu, việc tương tác thông qua controller đảm bảo việc vận hành ổn định và tránh các sai sót dữ liệu trực tiếp đến từ người dùng.

## Group assignment 4

* 1. **Identify and choose the user stories to be developed as the 1st version of website:**
* *Trong giai đoạn lựa hàng, khách hàng đã xem các sản phẩm, thêm các sản phẩm muốn mua vào giỏ hàng sau khi xem thông tin chi tiết mặt hàng đó. Người dùng truy cập phần giỏ hàng. Tại đây người dùng có thể thay đổi các thông tin về số lượng các sản phẩm có trong giỏ hàng trước khi tiếp tục thanh toán.*

⇒  Giai đoạn quản lý giỏ hàng:

* Tại đây người dùng thực hiện các chức năng thêm mặt hàng mới vào giỏ hàng, thay đổi số lượng mặt hàng hoặc xóa mặt hàng khỏi giỏ hàng.
* Sau khi thay đổi thông tin giỏ hàng, người dùng có thể lựa chọn tiếp tục quay lại mua sắm hoặc tiến tới bước đặt hàng và thanh toán.

⇒  Giai đoạn thanh toán giỏ hàng:

* Người dùng xác nhận lại thông tin chi tiết giỏ hàng (mặt hàng, đơn giá, thành tiền,...) và tùy chọn áp dụng mã khuyến mại ( nếu có)
* Sau khi xác nhận người dùng thông báo cho hệ thống việc đặt hàng của mình với thông tin tài khoản đang đăng nhập.
* Sau đó người dùng xác nhận lần cuối thông tin về đơn hàng.Lựa chọn hình thức thanh toán, thông tin, hình thức giao hàng
* Tiến hành thanh toán.
* Nhận được phản hồi của hệ thống
  1. **Specify completely theirs related task cards/ test cards**
     1. **Task card**

|  |  |  |
| --- | --- | --- |
| **STT** | **Task card** | **Description** |
| 1 | Thêm sản phẩm vào giỏ hàng (thêm ngoài giỏ hàng) | Hệ thống thêm một sản phẩm vào giỏ hàng khi người dùng chọn thêm sản phẩm ở mục tìm kiếm |
| 2 | Thêm bớt sản phẩm ở giỏ hàng | Hệ thống sửa số lượng sản phẩm ứng với thao tác thêm bớt của người dùng |
| 3 | Hiển thị thông tin giỏ hàng | Hệ thống hiển thị thông tin giỏ hàng sau khi người dùng chọn thao tác giỏ hàng |
| 4 | Xóa sản phẩm khỏi giỏ hàng | Hệ thống loại bỏ sản phẩm khỏi giỏ hàng khi người dùng bớt số lượng sản phẩm về 0 hoặc chọn xóa sản phẩm |
| 5 | Chuyển sang trang thanh toán | Hệ thống hiển thị trang thanh toán khi người dùng thao tác "thanh toán" |
| 6 | Hiển thị phương thức thanh toán | Hệ thống hiển thị các phương thức thanh toán khả dụng |
| 7 | Xác nhận phương thức thanh toán | Hệ thống nhận phương thức thanh toán từ thao tác người dùng |
| 8 | Lấy thông tin người dùng | Hệ thống nhận thông tin liên quan từ phía người dùng |
| 9 | Xác nhận thông tin người dùng | Hệ thống hiển thị thông tin 1 lần nữa để yêu cầu xác nhận từ người dùng, kiểm tra tính chính xác của thông tin |
| 10 | Tạo đơn hàng | Hệ thống đóng gói các sản phẩm trong giỏ hàng và thông tin liên quan thành đơn hàng |
| 11 | Reset giỏ hàng | Hệ thống xóa tất cả sản phẩm trong giỏ hàng |

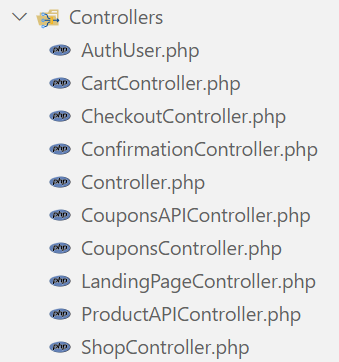
* + 1. **Test card**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Test Card | Mục tiêu | Bước thực hiện | Kết quả | Lỗi |
| 1 | Thêm sản phẩm vào giỏ hàng | Kiểm tra xem đã thêm mặt hàng hay chưa sau khi người dùng chọn "thêm vào giỏ hàng" | Người dùng bấm nút "thêm vào giỏ hàng" phía dưới của mặt hàng | Số lượng sản phẩm của mặt hàng vừa chọn trong giỏ hàng tăng lên hoặc số lượng mặt hàng tăng lên | Không thêm được sản phẩm |
| 2 | Thêm sản phẩm vào giỏ hàng | Kiểm tra đã đúng Mặt hàng được thêm vào giỏ hàng hay chưa | Người dùng bấm nút "thêm vào giỏ hàng" phía dưới của mặt hàng | Trong danh sách các sản phẩm có ID của mặt hàng vừa được thêm | Thêm sai sai mặt hàng |
| 3 | Hiển thị chi tiết giỏ hàng | Hệ thống di chuyển đến trang thông tin giỏ hàng. | Người dùng chọn nút "Giỏ hàng" | Giỏ hàng hiển thị các mặt hàng mà người dùng đã chọn, bao gồm mặt hàng, số lượng, đơn giá và thành tiền. | không hiển thị được giỏ hàng |
| 4 | Thêm, chỉnh sửa số lượng của sản phẩm trong giỏ hàng | thay đổi số lượng của sản phẩm trong giỏ hàng | Người dùng chọn "sửa sổ lượng" hoặc "thêm số lượng" của sản phẩm trong giao diện giỏ hàng | số lượng của sản phẩm thay đổi đúng theo người dùng đã chọn | Không chỉnh sửa được số lượng sản phẩm |
| 5 | xóa mặt hàng khỏi giỏ hàng | xóa mặt hàng khỏi giỏ hàng | Người dùng chọn "xóa mặt hàng" trong giao diện giỏ hàng | Xóa ID mặt hàng, số lượng,... của mặt hàng vừa chọn khỏi danh sách sản phẩm | Không xóa được mặt hàng |
| 6 | Chuyển trang thanh toán | Chuyển sang giao diện thanh toán để người dùng thực hiện thao tác thanh toán | Người dùng chọn "thanh toán" | Thanh toán hiện lên các thông tin thanh toán và vận chuyển | không chuyển được trang |
| 7 | Nhận mã khuyến mại | nhận mã và giảm giá cho người dùng | Người dùng nhập mã khuyến mại | Gọi API của nhóm 05, kiểm tra và xác nhận mã khuyến mại. Nếu đúng thì áp dụng giảm giá hoặc ưu đãi cho khách hàng | Không nhập được mã khuyến mại |
| 8 | Nhận thông tin thanh toán và giao hàng | Lưu trữ và áp dụng thông tin thanh toán, giao hàng của khách hàng | người dùng nhập và xác nhận thông tin | Xác nhận, lưu trữ và áp dụng thông tin của khách hàng | không xác nhận được thông tin của khách hàng |
| 9 | Tạo đơn hàng | Đóng gói giỏ hàng và tạo đơn hàng | Người dùng chọn "Xác nhận đơn hàng" | đóng gói thông tin của mặt hàng trong giỏ hàng, lưu trữ thông tin thanh toán và vận chuyển của khách hàng và đóng gói, chuyển cho API của nhóm 04 | Không xuất được đơn hàng |
| 10 | Reset giỏ hàng | sau khi đóng gói giỏ hàng cũ thành đơn hàng, tiến hành reset giỏ hàng | Sau khi xuất đơn hàng thành công | Xóa tất cả thông tin về các mặt hàng trong giỏ hàng như ID, số lượng,.. khỏi giỏ hàng | Không reset được giỏ hàng |

## Group assignment 5

* 1. **Justify (by means of code and comment) the application of SOLID principles to your design.**
     1. **Single responsibility principle**

Mỗi Controller chỉ đảm nhiệm một công việc liên quan duy nhất:



Các Controller phát triển chính, tiêu biểu của module:

* **CartController.php**: chịu trách nhiệm xử lý logic phần giỏ hàng
* **CheckoutController.php**: chịu trách nhiệm xử lý logic phần thanh toán
  + 1. **Open-Closed principle**

Sau này nếu phát triển thêm mới những xử lý logic mới, tính năng mới thì sẽ không thay đổi code sẵn có mà chỉ mở rộng và nạp chồng.

Ví dụ:

class NewCartController < CartController

Khi có nhu cầu thêm tính năng chỉ cần tạo controller, model, view mới và chỉnh sửa trong routes. Khi có nhu cầu thay thế controller này bằng controller khác, ta cũng chỉ cần chỉnh sửa trong routes.

## Group assignment 6

* 1. **Identify website’s functionalities that can be developed as a web service**

Những chức năng có thể có thể phát triển thành một web service là :

* Chức năng “Quản lý giỏ hàng”: Thêm, sửa, xóa các mặt hàng trong giỏ hàng
  1. **Develop such a web service**

Link test: <http://shopsp-01.herokuapp.com/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **API CUNG CẤP** | | | **Cung cấp cho** | **Thời điểm cung cấp** | **Mẫu JSON** |
| **Method** | **URI Pattern** | **Nội dung** |
| GET | /api/cart/{user\_id} | Xem giỏ hàng của người dùng cụ thể |  | Vòng 3 |  |
| PATCH | /api/cart/{user\_id} | Cập nhật số lượng giỏ hàng |  | Vòng 3 | {  "user\_id" : 2, "session\_id" : 123, "product\_id" : 17  "quantity" : 5 } |
| POST | /api/carts | Thêm vật phẩm mới vào giỏ hàng | Nhóm 9 | Vòng 3 | {  "user\_id" : 2, "session\_id" : 123, "product\_id" : 17 } |
| DELETE | /api/cart/{user\_id} | Xóa mặt hàng trong giỏ hàng |  | Vòng 3 | {  "user\_id" : 2, "session\_id" : 123, "product\_id" : 17 } |

* 1. **Use agile methods to integrate this web service to the existing version of website**

Sự phát triển phần mềm Agile nhấn mạnh vào 4 giá trị cốt lõi sau:

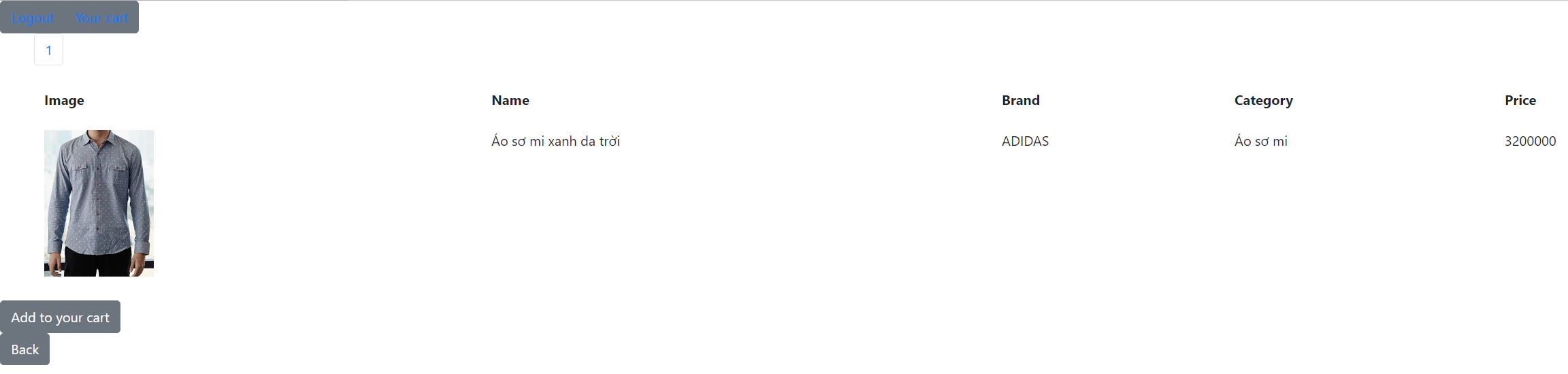
* Sự tương tác của cá nhân và nhóm thông qua các quy trình và công cụ.
* Phần mềm làm việc thông qua các tài liệu đầy đủ
* Sự hợp tác của khách hàng thông qua việc thương thuyết hợp đồng
* Đáp ứng để thay đổi nhằm theo sát các kế hoạch

Sử dụng agile method để tích hợp web service vào phiên bản hiện tại của website theo quy trình sau:

* Lập kế hoạch
* Phân tích
* Thiết kế và lập trình
* Kiểm thử sản phẩm
* Bàn giao sản phẩm

Sử dụng agile trong quản lý thời gian dự án: Do 1 vòng thực hiện công việc theo kiểu agile là không dài, đòi hỏi người phát triển phải phân tích và tập trung vào phần công việc mang lại giá trị nhiều nhất chứ không phải là làm nhiều việc có thể nhất. Vì vậy, việc đánh giá thời gian và độ ưu tiên công việc là cần thiết. Có 2 cách thường được sử dụng: Ma trận ưu tiên Eisenhower và Đánh độ ưu tiên dựa vào ROI (giá trị thu được / chi phí bỏ ra) và mức độ khẩn cấp.

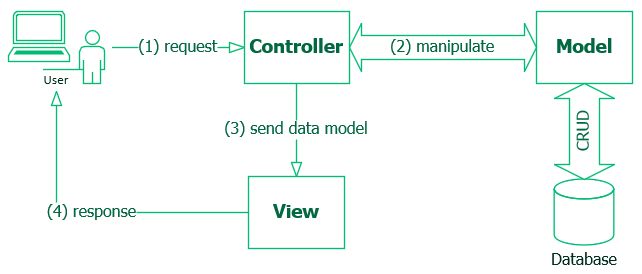
Kết quả thực tế: Cung cấp kịp thời cho nhóm 9 API thêm vào giỏ để thiết đặt button “Add to your cart” trong website của họ.

****

## Group assignment 7

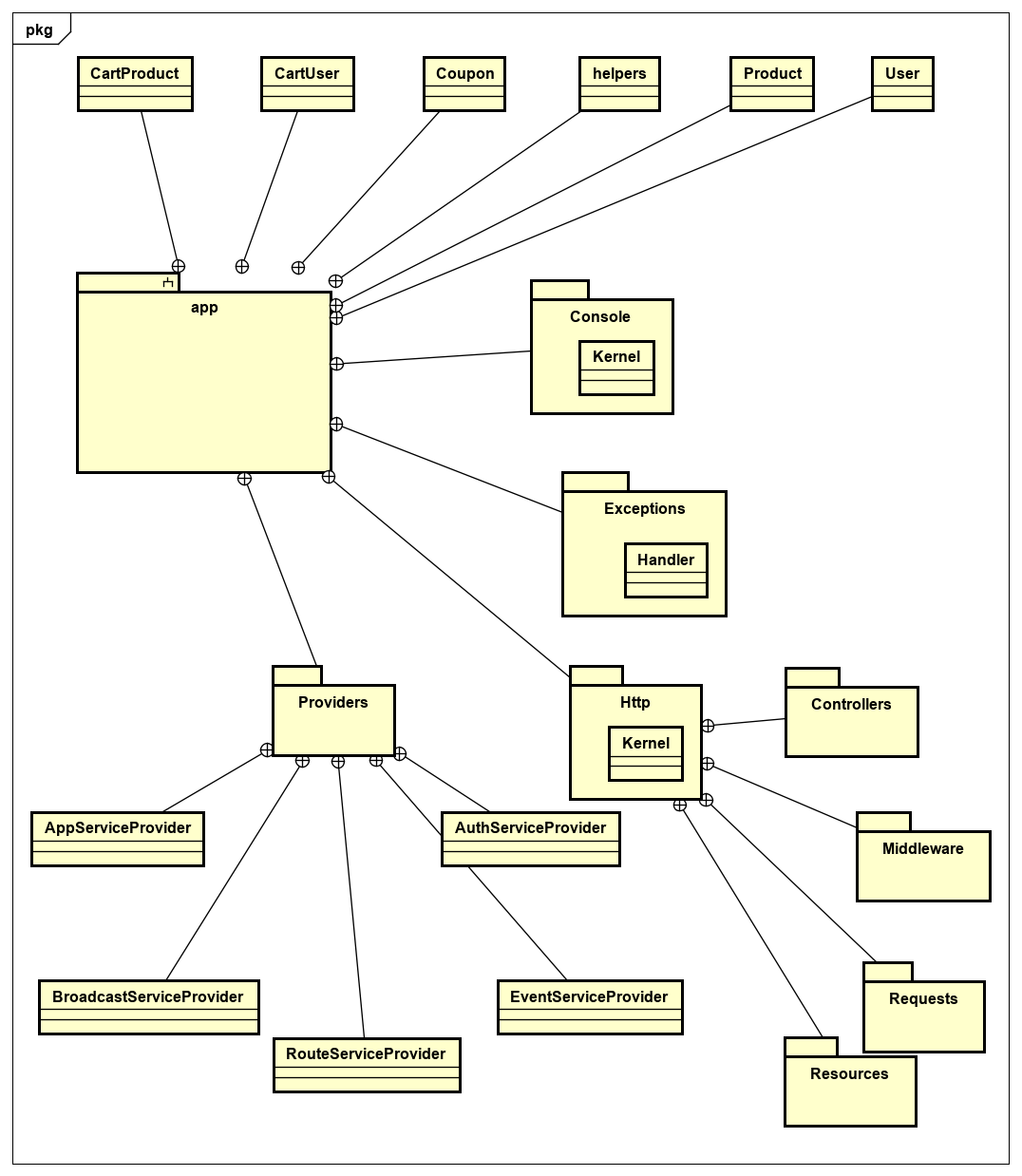
* 1. **Which patterns should be applied to which functionalities ?**

Hệ thống phát triển dựa trên mẫu thiết kế MVC

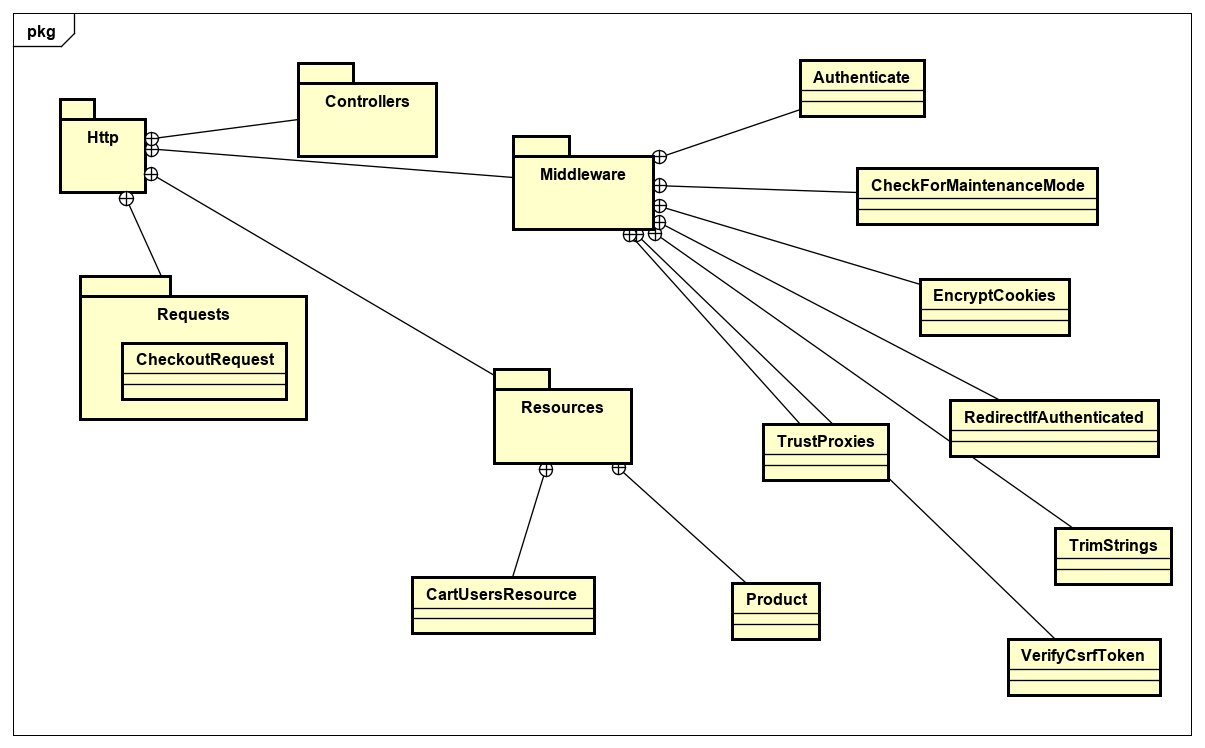


Cách triển khai:

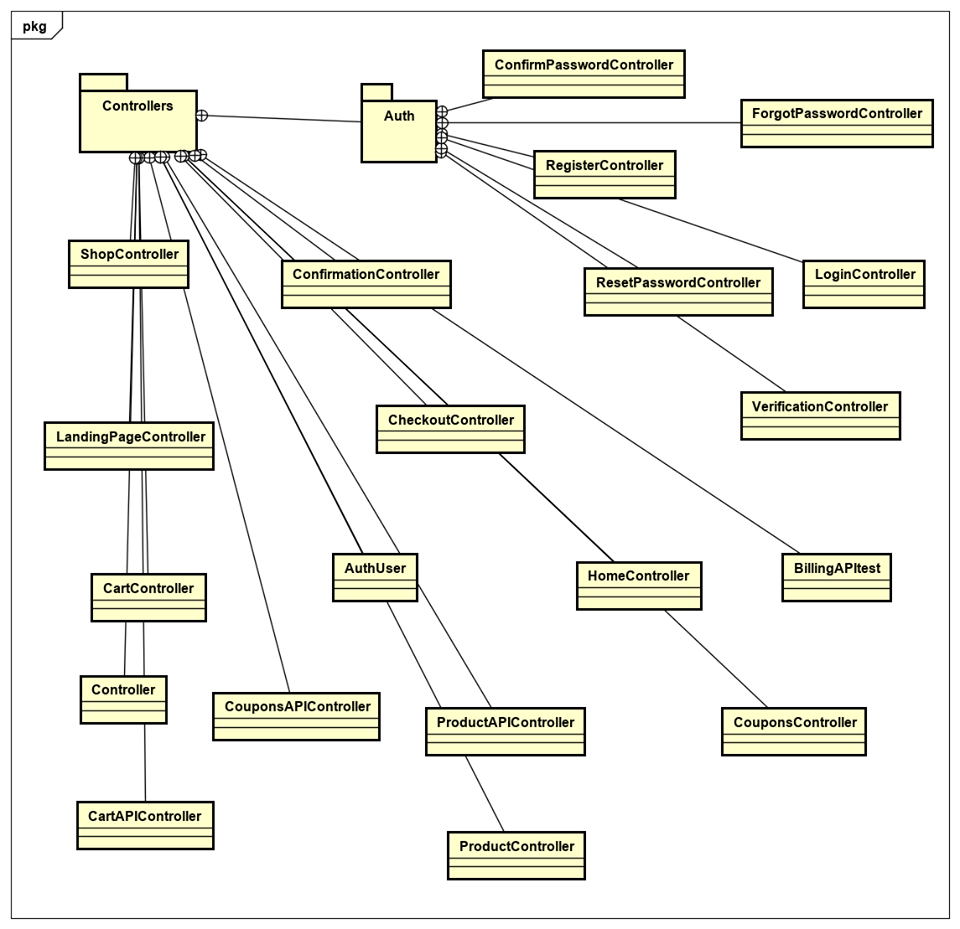
* Tách riêng phần biểu diễn và phần tương tác ra khỏi dữ liệu hệ thống.
* Cấu trúc hóa hệ thống thành ba thành phần logic tương tác với nhau:
  + View component: định nghĩa và quản lý cách dữ liệu được biểu diễn tới người dùng như thế nào, quản lý dữ liệu nào được hiển thị cho người dùng.
  + Controller component: Quản lý tương tác người dùng (ví dụ như ấn phím, nhấp chuột, sử dụng các chức năng). Controller xử lý các yêu cầu của người dùng, bằng việc giao tiếp tương tác với model để lấy được dữ liệu từ database để xử lý, sau đó kết xuất kết quả thành view.
  + Model component: lưu trữ các thao tác tương tác với cơ sở dữ liệu (nhập, sửa, xóa, truy xuất thông tin), được quản lý bởi Controller.
  1. **Illustrate by means of UML diagrams.**
* Class của chương trình:

****

* Class của file http:



* Class của http/Controllers:

****

* 1. **Justify by code**

