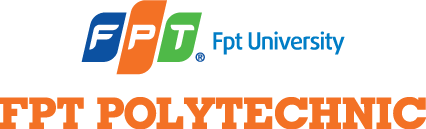
**TRƯỜNG CAO ĐẲNG THỰC HÀNH**

**FPT-POLYTECHNIC**

******

**KIỂM THỬ DỰ ÁN SHOPPING ONLINE**

**WED BÁN ÁO**

**Người thực hiện :**

Mai Kỷ Viễn – PS

Nguyễn Lê Tiến Bảo – PS11095

Nguyễn Minh Khôi – PS11136

Lê Duẩn Anh – PS11137

**Lớp :** UD15201

**Giáo viên hướng dẫn: Trần Phúc Hậu**

***Tp Hồ Chí Minh, ngày 25 tháng 5 năm 2019.***

**MENU**

[I. NỘI DUNG CHÍNH CỦA DỰ ÁN 2](#_Toc16812_WPSOffice_Level1)

[1. GIỚI THIỆU DỰ ÁN 3](#_Toc7951_WPSOffice_Level2)

[2. Cấu trúc dự án : 3](#_Toc21493_WPSOffice_Level2)

[3. Sơ đồ quan hệ thực thể 3](#_Toc16004_WPSOffice_Level2)

[a. Logon và Logout: 3](#_Toc7951_WPSOffice_Level3)

[b. Chức năng quản lý dành cho chủ shop. 4](#_Toc21493_WPSOffice_Level3)

[c. Chức năng dành cho khách hàng. 4](#_Toc16004_WPSOffice_Level3)

[d. Dịch vụ : 4](#_Toc4674_WPSOffice_Level3)

[e. Địa chỉ, thông tin liên hệ 4](#_Toc3899_WPSOffice_Level3)

[5. Các công cụ mã nguồn mở . 4](#_Toc4674_WPSOffice_Level2)

[5.1 Giới thiệu về JUnit 4](#_Toc22632_WPSOffice_Level3)

[II. TEST DESIGN 5](#_Toc7951_WPSOffice_Level1)

[1. Tạo tài khoản và đăng nhập 5](#_Toc3899_WPSOffice_Level2)

[2. Thêm sửa tìm kiếm : 5](#_Toc22632_WPSOffice_Level2)

[3. Testing Shopping Cart - Kiểm tra giỏ hàng 5](#_Toc647_WPSOffice_Level2)

[4. Post Purchase Test – Kiểm thử mua hàng 6](#_Toc24935_WPSOffice_Level2)

[III. TEST CASE 6](#_Toc21493_WPSOffice_Level1)

[1. Test khách hàng dao : 6](#_Toc8828_WPSOffice_Level2)

[1.1 Đăng nhập : 6](#_Toc647_WPSOffice_Level3)

[1.2 Lấy thông tin khách hàng 7](#_Toc24935_WPSOffice_Level3)

[1.3 Chức năng lấy list của khách hàng. 8](#_Toc8828_WPSOffice_Level3)

[1.4 Chức năng thêm khách hàng 9](#_Toc9680_WPSOffice_Level3)

[1.5 Chức năng xóa khách hàng 10](#_Toc26890_WPSOffice_Level3)

[2. Chức năng giỏ hàng 11](#_Toc9680_WPSOffice_Level2)

[3. Chức năng đặt đơn hàng : 14](#_Toc26890_WPSOffice_Level2)

[4. Kết quả 15](#_Toc5355_WPSOffice_Level2)

[4.1 Giỏ hàng dao test 16](#_Toc5355_WPSOffice_Level3)

[4.2 Hóa đơn test 16](#_Toc22793_WPSOffice_Level3)

[4.3 Khách hàng test 16](#_Toc31770_WPSOffice_Level3)

1. **NỘI DUNG CHÍNH CỦA DỰ ÁN**
2. **GIỚI THIỆU DỰ ÁN**

Cửa hàng áo Sur Dot chuyên bán các loại áo thun trơn basic, nhiều kiểu dáng và hiện tại không có 1 trang website bán hàng riêng của cửa hàng. Vì vậy cửa hàng yêu cầu xây dựng 1 trang website bán hàng online để bán hàng 1 cách hiệu quả hơn .

1. **Cấu trúc dự án :**

- Homepage – Trang chủ

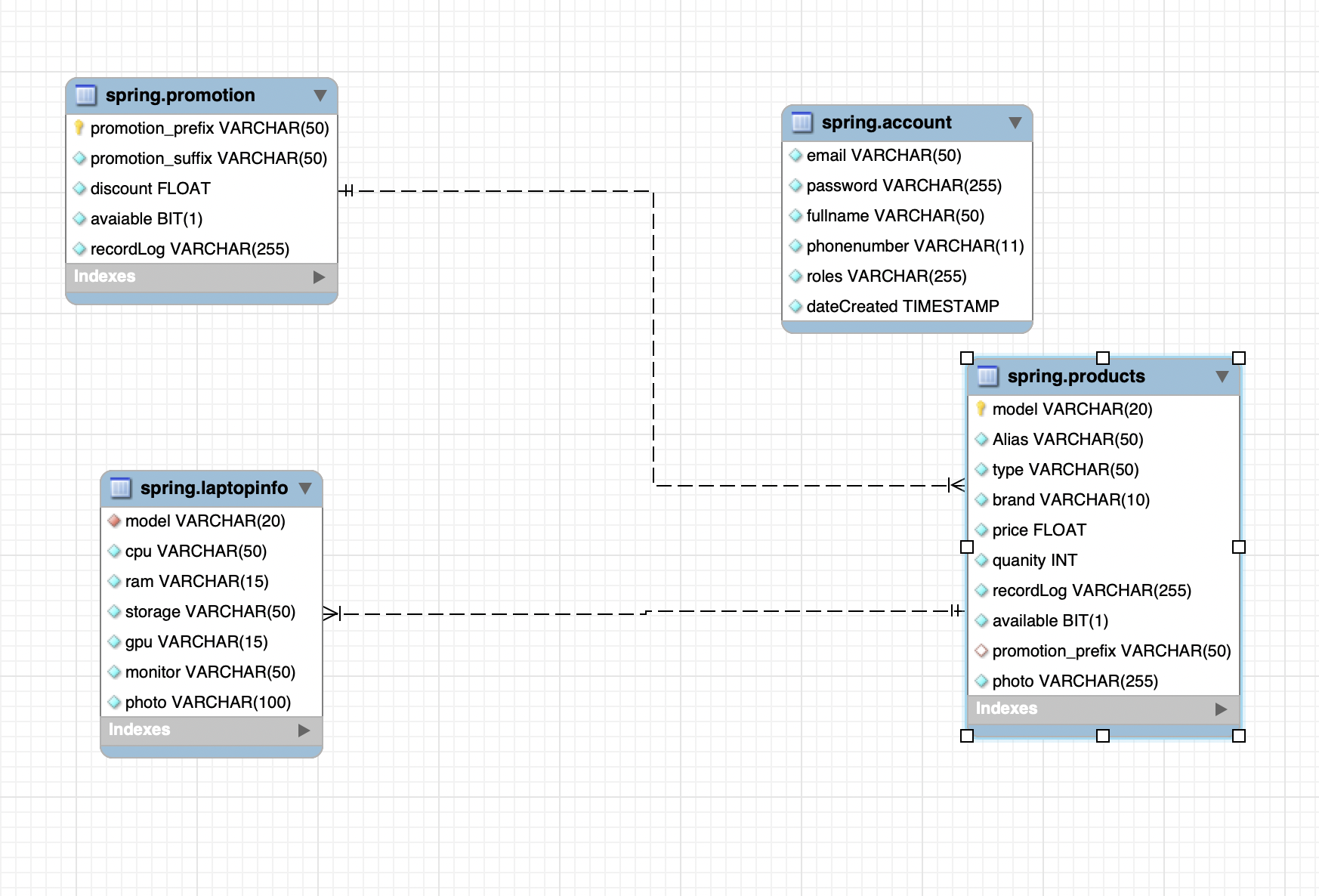
- Search Results Page – Trang kết quả tìm kiếm

- Product Details Page – Trang chi tiết sản phẩm

- Order Form Page – Trang đặt hàng

- Order Confirmation Page – Trang xác nhận đặt hàng

- Login Form Page and Accounts Pages – Trang đăng nhập và tài khoản

1. **Sơ đồ quan hệ thực thể **
2. **Các yêu cầu chức năng :**

### **Login và Logout:**

Người dùng nhập User name và Password để đăng nhập vào hệ thống

Nếu người dùng quên mật khẩu, họ có thể nhấp vào liên kết 'Quên mật khẩu' trên trang Đăng nhập. Màn hình sẽ hiển thị cho phép người dùng nhập Tên đăng nhập và Email để lấy lại mật khẩu. Nếu Tên đăng nhập và Email không khớp, thông báo lỗi sẽ hiển thị 'Tên người dùng và Email không khớp'. Nếu không, hệ thống sẽ gửi mật khẩu cho đầu vào email.

1. **Chức năng quản lý dành cho chủ shop.**

- Chức năng thêm sản phẩm mới cần bán .

- Chức năng thay đổi thông tin sản phẩm đã có .

- Chức năng xóa bỏ các sản phẩm mà cửa hàng không còn bán .

- Chức năng xem toàn bộ sản phẩm của shop phân loại theo hãng .

- Chức năng tìm kiếm sản phẩm

- Chức năng xem đơn đặt hàng và sản phẩm đã bán trong ngày.

1. **Chức năng dành cho khách hàng.**

- Xem các sản phẩm theo danh mục và xem chi tiết từng sản phẩm.

- Chức năng thêm sản phẩm vào giỏ hàng và xóa sản phẩm ra khỏi giỏ hàng

- Chức năng đặt đơn hàng và xóa đơn hàng.

- Chức năng xem thông tin đăng ký

1. **Dịch vụ :**

Giới thiệu về dịch vụ của cửa hàng : như bảo trì, giao hàng v.v

1. **Địa chỉ, thông tin liên hệ**

Giới thiệu địa chỉ shop để khách hàng cần đến mua bán trực tiếp hay trao đổi thông tin dịch vụ, giới thiệu thông tin liên hệ như số điện thoại, email, các mạng xã hội dành cho khách hàng có nhu cầu..

1. **Các công cụ mã nguồn mở .**
   1. **Giới thiệu về JUnit**
      1. **JUnit là gì ?**

Trong Java, để thực hiện viết code cho Unit Test chúng ta có thể sử dụng một trong hai Framework: [JUnit](https://junit.org/) và [TestNG](https://testng.org/).

**JUnit** là một framework mã nguồn mở, miễn phí, đơn giản dùng để unit test cho ngôn ngữ lập trình Java. Trong Java, chúng ta thường sẽ sử dụng method để làm unit test.

Chúng ta có thể sử dụng JUnit để viết code test cho cả unit testing và integration testing.

## 5.1.2 Các tính năng của JUnit

* JUnit là một framework mã nguồn mở, được sử dụng để viết và chạy kiểm thử.
* Cung cấp các annotation để định nghĩa các phương thức kiểm thử.
* Cung cấp các Assertion để kiểm tra kết quả mong đợi.
* Cung cấp các test runner để thực thi các test script.
* Test case JUnit có thể được chạy tự động.
* Test case JUnit có thể được tổ chức thành các test suite.
* JUnit cho thấy kết quả test một cách trực quan: pass (không có lỗi) là màu xanh và fail (có lỗi) là màu đỏ.
  1. **Giới thiệu về Selenium**

## 5.2.1 Selenium là gì?

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá là giống với HP Quick Test Pro (QTP bây giờ là UFT) chỉ khác là Selenium thì tập trung vào việc tự động hoá các ứng dụng dựa trên nền tảng web. Kiểm thử được thực hiện bằng cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium. Selenium không chỉ là 1 công cụ độc lập mà là 1 bộ công cụ của phần mềm, mỗi bộ đều đáp ứng được nhu cầu kiểm thử khác nhau của 1 tổ chức. Nó có 4 thành phần.

• Selenium Integrated Development Environment (IDE)

• Selenium Remote Control (RC)(s1)

• WebDriver

• Selenium Grid

Hiện tại, Selenium RC và WebDriver được hợp nhất thành một framework duy nhất để tạo ra Selenium 2. Còn Selenium 1 thì tham chiếu đến Selenium RC.

**5.2.2 Những điều cần chú ý khi sử dụng selenium**

Tests nên luôn có 1 starting point đã biết. Với Selenium, nó nghĩa là mở 1 page để bắt đầu 1 workflow.

Tests không nên phải phụ thuộc vào bất kỳ 1 test nào khác để chạy. Nếu 1 test dự định add 1 cái gì đó thì phải đảm bảo rằng không có 1 test khác sẽ delete cái định add đó. Điều này đảm bảo rằng nếu có gì đó bị sai trong 1 test, thì nó cũng không phải là unnecessary failures.

Tests chỉ nên test 1 thứ tại 1 thời điểm.

Tests nên làm sạch sau khi chạy xong.

Nếu không thỏa mãn các rule này khi chạy test tự động Selenium thì có thể sẽ gặp phải nhiều issues sau đó và nếu như có rất nhiều TH cần phải test thì những vấn đề nhỏ này sẽ dẫn tới có rất nhiều phần của 1 test suite sẽ bị ra kết quả thất bại.

1. **TEST DESIGN**
2. **Tạo tài khoản và đăng nhập**

Wedsite chỉ cho khách hàng mua hàng bằng việc tạo tài khoản khách hàng trước khi mua hàng . Chúng ta cần kiểm tra tất cả sự thay đổi trong qua trình mua hàng của khách hàng khi đăng nhập tài khoản.

- **Tạo tài khoản và đăng nhập trước khi mua hàng**  : kiểm tra rằng các mặt hàng bạn mua được thêm vào và kết nối với tài khoản chính xác. Ngoài ra, bạn không nên bị yêu cầu phải login lại nếu bạn đã login rồi, hoặc chưa login mà bạn muốn mua hàng thì sẽ được đưa về login .

**- Đăng nhập và Đăng xuất** : khi bạn đã đăng nhập và sau đó đăng xuất, hãy chắc chắn rằng bạn không thể truy cập vào trang tài khoản.

1. **Thêm sửa tìm kiếm :**

**Sản phẩm thích hợp** – Kiểm tra các sản phẩm được hiển thị tương ứng với điều kiện tìm kiếm..

**Thông tin sản phẩm**  : các sản phẩm hiển thị kèm theo hình ảnh, tên, giá sản phẩm.

**Thêm sản phẩm** : kiểm tra xem sản phẩm khi được thêm vào có xuất hiện trên trang views các sản phẩm của cửa hàng hay không. Và sản phẩm có được lưu trữ lại trong dữ liệu của hàng hay không.

**Update sản phẩm** : kiểm tra sự thay đổi thông tin sản phẩm có được hiện thị trên các chức năng tương ứng khác .

1. **Testing Shopping Cart - Kiểm tra giỏ hàng**

**Shopping cart là 1 trong những chức năng chính,** cho phép khách hàng lựa chọn và lưu trữ nhiều sản phẩm trong giỏ hàng và thực hiện giao dịch mua chúng trong 1 lần. Vì vậy chúng ta cần test :

**- Thêm sản phẩm vào giỏ hàng** – giỏ hàng được cập nhật sản phẩm với tên, hình ảnh, giá tương ứng chính xác.

**- Tăng số lượng sản phẩm trong giỏ hàng** – giá được cập nhật chính xác tương ứng với sản phẩm.

**- Thêm cùng một sản phẩm nhiều lần** – có phải là cùng một mặt hàng trong giỏ hàng, nhưng số lượng phải phản ánh chính xác số sản phẩm được thêm vào và tổng giá trị đơn hàng được cập nhật bằng tổng giá của từng sản phẩm trong giỏ hàng.

**- Thêm nhiều loại sản phẩm khác nhau** – Đối với mỗi sản phẩm được thêm vào chúng ta sẽ thấy tên sản phẩm, hình ảnh, giá tương ứng và tổng giá trị tất cả các sản phẩm.

**- Loại bỏ một số sản phẩm trong giỏ hàng** – giỏ hàng cập nhật các sản phẩm còn lại trong giỏ hàng và tổng giá trị tương ứng.

**- Loại bỏ tất cả sản phẩm trong giỏ hàng** – số dư của giỏ hàng trở về 0, và không còn sản phẩm nào được hiển thị trong giỏ hàng.

1. **Post Purchase Test – Kiểm thử mua hàng**

Khi thực hiện mua hàng, có rất nhiều hành động mà người dùng có thể làm liên quan đến việc mua hàng của họ như :

- Hủy đơn hàng hoặc thay đổi số lượng đơn hàng

- Xem lại đơn hàng và lịch sử các mặt hàng đã mua gần đây.

1. **Bảng Phân chia công việc**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Duration** | **Start** | **Finish** | **Predecessers** | **Resource**  **Name** |
| Thời gian phát triển dự án | 30days | Start 19/5/2019 | 19/6/2019 |  | Tuấn anh |
| Test login | 3days | Start 19/5/2019 | 22/5/2019 |  | Tuấn Anh |
| Test thêm người dùng | 3days | Start 22/5/2019 | 25/5/2019 |  | Tuấn Anh |
| Test tìm kiếm người dùng | 3days | Start 25/5/2019 | 28/5/2019 |  | Tuấn Anh |
| Test Thêm sản phẩm trong giỏ hàng | 3days | Start 28/5/2019 | 31/5/2019 |  | Tuấn Anh |
| Test xóa bỏ sản phẩm trong giỏ hàng | 3days | Start 31/5/2019 | 3/6/2019 |  | Tuấn Anh |
| Test hủy dơn hàng | 3days | Start 3/6/2019 | 6/6/201 |  | Tuấn Anh |

1. **TEST CASE**

Trong Test Packages tạo 1 Packages Dao để chưa các lớp thực thi test của lớp Dao. Đê test lớp GioHangDao, HoaDonDao,KhachHangDao ta tạo các lớp chứa các phương thức test GioHangDaoTest, HoaDonDaoTest, KhachHangDaoTest. Chi tiết như sau :

1. **Test khách hàng dao :**

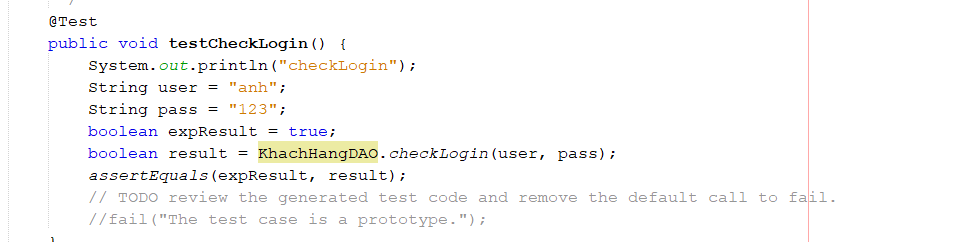
**1.1 Đăng nhập :**

Mục đích kiểm tra: kiểm tra việc đăng nhập vào hệ thống của user thành công thì sẽ hiện thị như thế nào và không thành công thì sẽ thông báo lỗi ra sao ( admin or user)

Chức năng kiểm tra: login()

Giả định : có 2 trường là username và password. Username : **anh**; password : **123**

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Thực hiện | Kết quả mong đợi | Kết quả thực hiện |
| 1 | user = "anh";  pass = "123";  Với 2 biến user và pass được truyền vào.  Sử dụng AssertEquals | expResult = passed | Result = passed |
| 2 | user = "";  pass = "123";  Với 2 biến user và pass được truyền vào | expResult = false | Result = false |
| 3 | user = "anh";  pass = "";  Với 2 biến user và pass được truyền vào | expResult = false | Result = false |

**Test login bằng selenium :**

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Thực hiện | Kết quả mong đợi | Kết quả thực hiện |
| 1 | user = "user";  pass = "123";  Với 2 biến user và pass được truyền vào. | expResult = passed | Result = passed |
| 1 | user = "user";  pass = "4214131";  Với 2 biến user và pass được truyền vào. | expResult = false  Xuất hiện trang báo Error | Result = false  Xuất hiện trang báo Error |



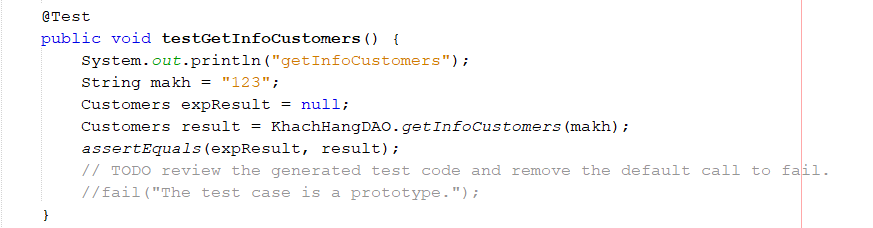
**1.2 Lấy thông tin khách hàng**

Mục đích kiểm tra: kiểm tra việc load thông tin của người dùng lên Views.

Chức năng kiểm tra: GetInfoCustomers()

Giả định: Admin đã có mã khách hàng . Makh= “anh”

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Thực hiện | Kết quả mong đợi | Kết quả thực hiện |
| 1 | Makh= “anh”  Với 1 biến makh được truyền vào  Sử dụng AssertEquals  Run Junit | expResult = false | Result = false |
| 1 | Makh= “”  Với 1 biến makh được truyền vào  Run Junit | expResult = passed | Result = passed |
| 3 | Makh= “123”  Với 1 biến makh được truyền vào  Run Junit | expResult = passed | Result = passed |

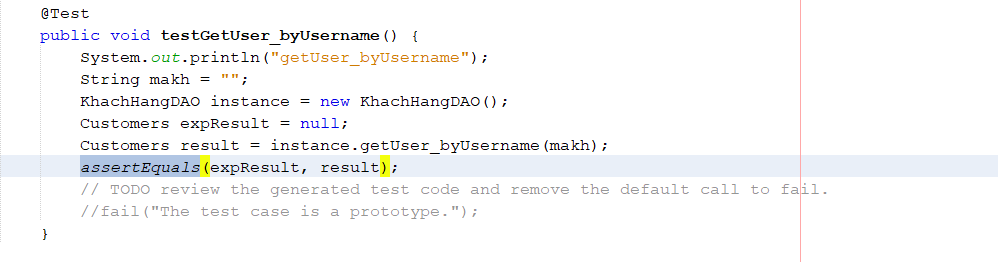


**1.3 Chức năng lấy khách hàng khi có mã khách hàng**

Mục đích kiểm tra : kiểm tra xem khi dùng 1 mã khách hàng đã cho trước có thể lấy ra user tương ứng không

Chức năng : testGetUser\_byUsername();  
Giả Định : data đầy đủ dữ liệu khách hàng , mã số khách hàng sử dụng : makh = “anh” hoặc makh = “ ”;

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | Truyền makh = “anh”  Run Junit  Sử dụng assertEquals | expResult = false | result= false |
| 2 | Truyền makh = “”  Run Junit  Sử dụng assertEquals | expResult = passed | result= passed |

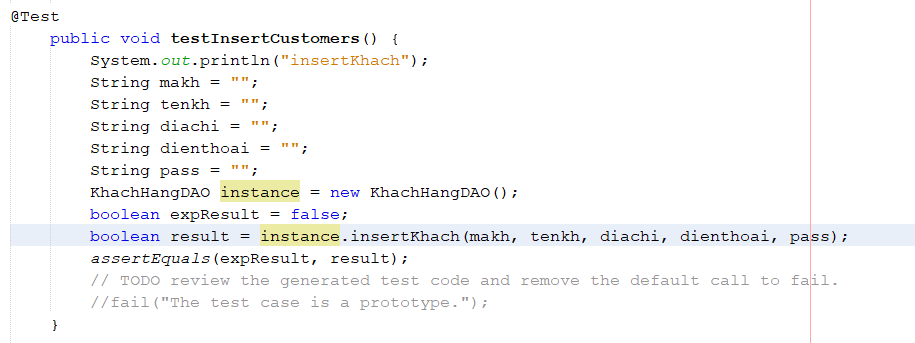


**1.4 Chức năng thêm khách hàng**

Mục đích kiểm tra : kiểm tra thêm 1 khách hàng vào data.

Chức năng : InsertKhach();  
Giả Định : thông tin khách hàng cần nhập đã đầy đủ .

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | makh = "";  tenkh = "";  diachi = "";  dienthoai = "";  pass = "";  Truyền vào 5 biến.  Run Junit  Sử dụng AssertEquals | expResult = passed | result= passed |
| 2 | makh = "123";  tenkh = "";  diachi = "";  dienthoai = "";  pass = "";  Run Junit  Sử dụng AssertEquals | expResult = false | result= false |

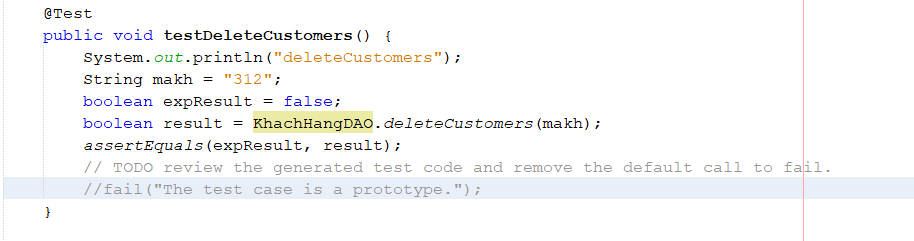


**1.5 Chức năng xóa khách hàng**

Mục đích kiểm tra : kiểm tra xóa 1 khách hàng vào data.

Chức năng : DeleteCustomers();  
Giả Định : có mã khách hàng.

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | makh = "321";  Run Junit  Sử dụng AssertEquals | expResult = passed | result= passed |
| 2 | makh = "anh";  Run Junit  Sử dụng AssertEquals | expResult = false | result= false |



1. **Chức năng giỏ hàng**

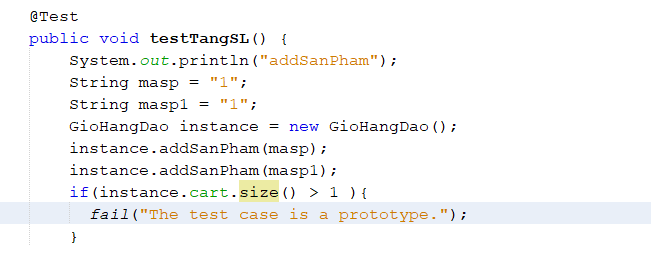
Mục đích kiểm tra : kiểm tra việc thay đổi thông tin trong giỏ hàng thì sẽ hiển thị như thế nào, có chính xác như mong muốn không.

Chức năng : thêm sản phẩm vào giỏ hàng, xóa sản phẩm ra khỏi giỏi hàng.(GetListItems() , RemoveSanPham(),RemoveAllSanPham(),testGetTongTien())

Giả định : user đang đăng nhâp vào hệ thống với tài khoản khách hàng và muốn chọn sản phẩm vào giỏ hàng.

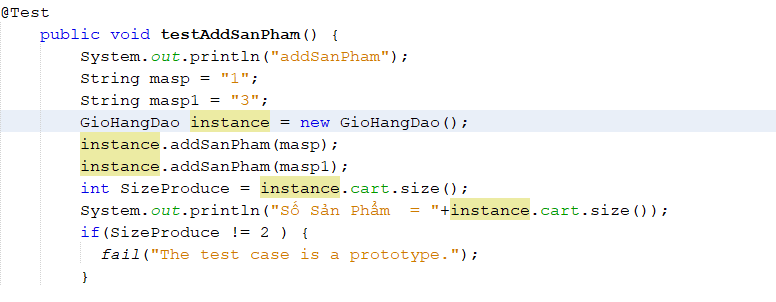
- **Trường hợp** 1 : test tăng số lượng hiện thi khi khách hàng chọn cùng 1 sản phẩm nhiều lần vào giỏ hàng

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | masp =”1”,  Masp2 = “1”  Run Junit  Sử dụng Fail | expResult = passed | Result = passed |



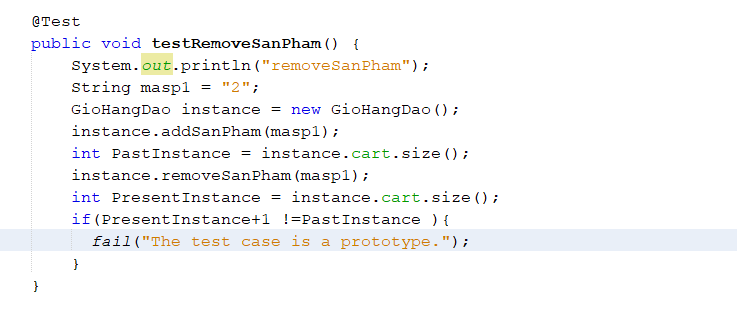
|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | masp =”1”,  Masp2 = “2”  Run Junit | expResult = passed | Result = passed |

- **Trường hợp** 2 :kiểm tra khi không chọn cùng 1 sản phẩm nhiều lần thì sản phẩm có được vào trong giỏ hàng không.



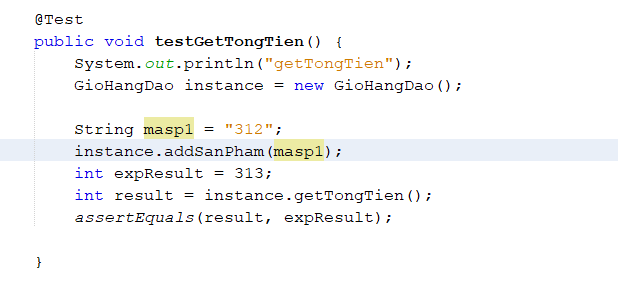
**Trường hợp 3 :** Test xóa 1 sản phẩm ra khỏi giỏ hàng

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | masp =”1”,  Run Junit  Sử dụng Fail | expResult = passed | Result = passed |



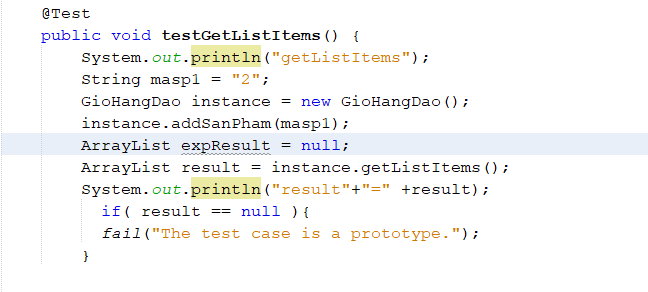
**Trường hợp 4 :** Test tổng giá tại 1 sản phẩm trong giỏ hàng với masp = “312”

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | masp =”312”,  Run Junit  Sử dụng AssertEquals | expResult = passed | Result = passed |



**Trường hợp 5** : Test chức năng lấy list item trong giỏ hàng với Masp1 =”2”

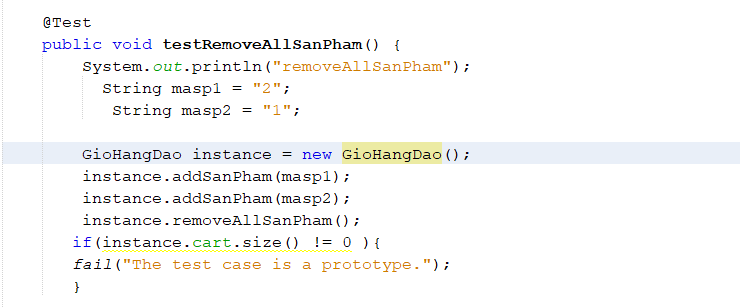
|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | Masp1 =”2”,  Run Junit | expResult = passed | Result = passed |



**Trường hợp 6** : Test xóa nhiều sản phẩm ra khỏi giỏ hàng masp1 = “2”,

Masp2=””1

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | masp1 = “2”,  Masp2=””1  Run Junit | expResult = passed | Result = passed |



1. **Chức năng đặt đơn hàng :**

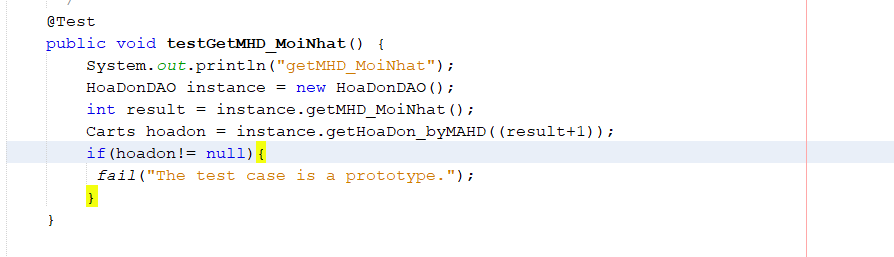
Mục đích kiểm tra : kiểm tra việc thay đổi thông tin đơn hàng và các thay đổi liên quan đến việc mua hàng của họ.

Chức năng : GetMHD\_MoiNhat(),GetHoaDon\_byMAHD() ,GetInfoCustomers()

Giả định : user đang đăng nhâp vào hệ thống với tài khoản khách hàng và đã có các sản phẩm trong giỏ hàng và muốn đặt đơn hàng này.

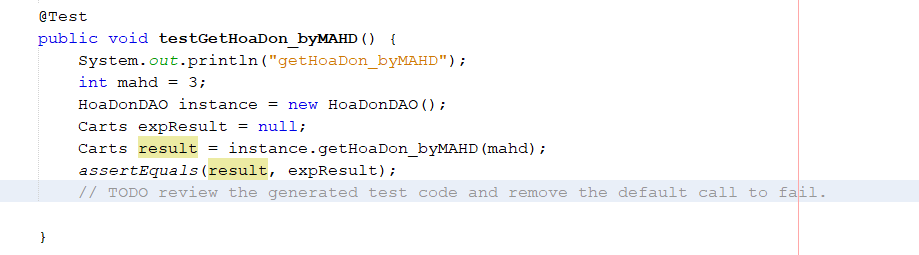
**Trường hợp 1** : kiểm tra độ chính xác của hàm lấy mã hóa đơn mới nhất.

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | Run Junit test | expResult = passed | Result = passed |



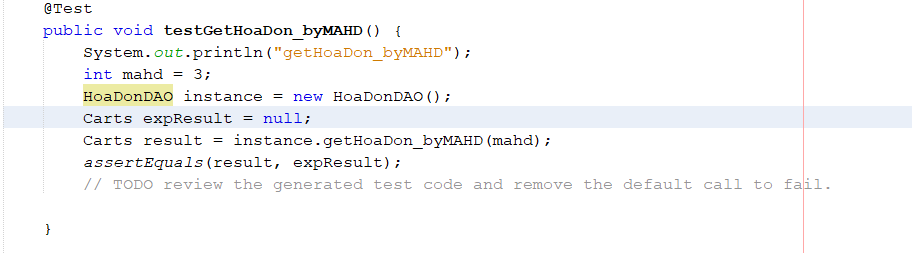
**Trường hợp 2** : Kiểm tra lấy mã hóa đơn bằng mahd = 3 ;

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | Run Junit test  mahd = 3 ; | expResult = passed | Result = passed |
| 2 | Run Junit test;  mahd = 12 | expResult = false | Result = false |



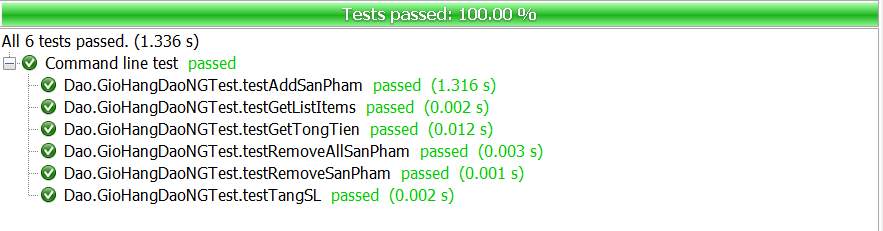
**Trường hợp 2** : Kiểm tra lấy mã hóa đơn bằng mahd = 3 ;

|  |  |  |  |
| --- | --- | --- | --- |
| step | Thực hiện | Kết quả mong đơi | Kết quả thực hiện |
| 1 | Run Junit test  mahd = 3 ; | expResult = passed | expResult = passed |
| 2 | Run Junit test;  mahd = 12 | expResult = false | expResult = false |

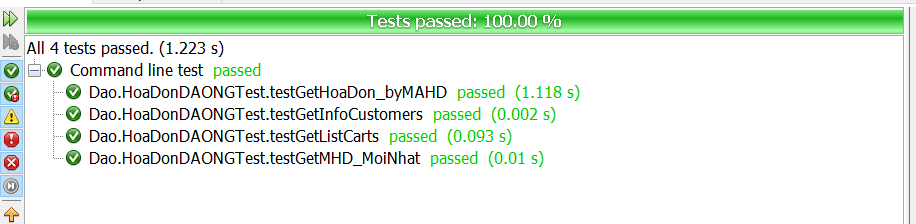


1. **Kết quả**

**4.1 Giỏ hàng dao test**



**4.2 Hóa đơn test**



**4.3 Khách hàng test**

