Tài liệu kiểm thử phần mềm

Software Document Testing

PHẦN MỀM QUẢN LÍ CỬA HÀNG THỜI TRANG

**Phiên bản 4.0**

**Phiên bản mới nhất (28/11/2022)**

**Nhóm 6**

**Hướng dẫn bởi: TS.Võ Hoàng Quân**

**Hồ Chí Minh, ngày 14, tháng 12, năm 2022**

**Table of Contents**

[**1. Static test**](#_heading=h.1fob9te) **1**

[**1.1 Kiểm thử phần mềm mà không thực hiện phần mềm**](#_heading=h.3znysh7) **1**

[**1.2 Kiểm tra tính đúng đắn của code, thuật toán,..**](#_heading=h.fzu06lrb5pck) **1**

[**1.3 Quy trình đánh giá**](#_heading=h.tw8uidautjov) **1**

[**2. Dynamic Test**](#_heading=h.xazh2o1ke74r) **1**

[**2.1 Unit Test (Test đơn vị nhỏ nhất)**](#_heading=h.2s8eyo1) **1**

[**2.2 Integration Test (Kiểm thử tích hợp)**](#_heading=h.fpujz37njo55) **2**

[**2.3 System Test (Kiểm thử toàn bộ hệ thống)**](#_heading=h.gfbg2tt647xi) **2**

[**2.4 Acceptance Test (Kiểm thử chấp thuận)**](#_heading=h.wtfdyngkql56) **2**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Trương Thông Thế Thái |  |  |  |

# Static test

## Kiểm thử phần mềm mà không thực hiện phần mềm

* Kiểm tra các requirements, functional requirements, security requirements xem code có thực hiện theo đúng yêu cầu đã đưa ra trong SRS hay không nhắm phát hiện sớm các các thiếu sót trước khi chúng ta bước vào Dynamic Test, nhằm giảm chi phí phát sinh của dự án
* Sau khi kiểm tra thì bạn code làm đã đúng với những yêu cầu của tài liệu SRS đưa ra

## Kiểm tra tính đúng đắn của code, thuật toán,..

* Kiểm tra cách đặt tên biến, tên hàm, cách đặt tên thư mục có đúng quy chuẩn quy cách hay chưa, code có được tối ưu hay không?
* Sau quy trình kiểm tra đã đánh giá được: Code gọn gàng sạch sẽ, tên biến tên hàm viết bằng tiếng anh đúng với quy chuẩn

## Quy trình đánh giá

* Lập kế hoạch: Tiêu chí, nhân sự
* Khởi động: Phân phối tài liệu, giải thích mục tiêu, vai trò
* Chuẩn bị cá nhân: Xem xét tài liệu, ghi nhận lỗi tiềm năng, câu hỏi, bình luận
* Kiểm tra, đánh giá, ghi kết quả
* Làm việc lại: Khi đa tìm thấy lỗi
* Thực hiện tiếp theo: Kiểm tra lỗi đã sửa

# Dynamic Test

## Unit Test (Test đơn vị nhỏ nhất)

* **Vòng đời chính của Unit Test gồm có 3 phần chính** 
  + Fail(Trạng thái lỗi)
  + Ignore(Tạm ngừng phiên thực hiện)
  + Pass(Trạng thái làm việc)
* **Test Point**
  + Tiến hành kiểm tra những dơn vị nhỏ nhất trong soure
* **Test Suite**
  + Tiến hành test một tập hợp các test case định nghĩa cho từng module hoặc hệ thống con
* **Regression Testing (hoặc Automated Testing)**
  + Ở đây chúng ta có thể lựa chọn các giải pháp test tự động với Aumation testing nhằm ngăn chặn lỗi cú pháp (Syntax) có thể văng ra
* **Production Code**
  + Phần mã chính của ứng dụng sẽ được chuyển giao cho khách hàng
* **Unit Testing Code**
  + Phần mã phụ để kiểm tra ứng dụng chính không được chuyển giao cho khách hàng
* **Mã các tính năng thử nghiệm (Test Design Specification Identifier)**
  + Tính năng đăng kí (DK01, DK02, DK03, DK04)
  + Tính năng đăng nhập (DN01)
  + Tính năng mua hàng (MH01)
  + Tính năng tạo hóa đơn (HĐ01)
  + Tính năng thêm sản phẩm mới vào kho (QLK1)
  + Tính năng trả hàng (TH01)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | **Test Case Description** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/ Fail** |
| DK01 | Kiểm tra để trống thông tin khi đăng kí | 1. Vào trang đăng kí 2. Không nhập bất kì thông tin nào 3. Bấm nút đăng kí |  | Đăng kí không thành công và yêu cầu nhập thông tin | Đăng kí không thành công và yêu cầu nhập thông tin | Pass |
| DK02 | Kiểm tra mật khẩu và nhập lại mật khẩu không khớp nhau | 1. Vào trang đăng kí 2. Nhập 2 mật khẩu khác nhau 3. Bấm nút đăng kí | Mật khẩu: 123456  Nhập lại mật khẩu: 1234567 | Đăng kí không thành công và yêu cầu nhập đúng mật khẩu | Đăng kí không thành công và yêu cầu nhập đúng mật khẩu | Pass |
| DK03 | Kiểm tra validate email khách hàng | 1. Vào trang đăng kí 2. Nhập email sai định dạng 3. Bấm nút đăng kí | Email: rhinogmail.com | Đăng kí không thành công và yêu cầu nhập lại email | Đăng kí thành công. | Fail |
| DK04 | Kiểm tra validate số điện thoại khách hàng | 1. Vào trang đăng kí 2. Nhập chữ trong ô số điện thoại 3. Bấm nút đăng kí | SĐT: asd123 | Đăng kí không thành công và yêu cầu nhập đúng số điện thoại | Đăng kí thành công | Fail |
| DN01 | Kiểm tra để trống tài khoản và mật khẩu xem có đăng nhập đc không | 1. Vào trang đăng nhập 2. Bấm nút đăng nhập |  | Đăng nhập không thành công, yêu cầu nhập đầy đủ thông tin | Đăng nhập không thành công, yêu cầu nhập đầy đủ thông tin | Pass |
| MH01 | Kiểm tra số lượng hàng trước khi đặt và sau khi đặt | 1. Chọn sản phẩm 2. Nhập số lượng 3. Đặt hàng |  | Số lượng hàng giảm đúng với so với số lượng khách đặt | Số lượng hàng giảm đúng với so với số lượng khách đặt | Pass |
| HĐ01 | Kiểm tra nhập sai số điện thoại khi lập hóa đơn | 1. Chọn sản phẩm 2. Thêm vào hóa đơn 3. Nhập số điện thoại sai 4. Tạo hóa đơn |  | Tạo hóa đơn không thành công và báo nhập lại số điện thoại | Tạo hóa đơn thành công | Fail |
| QLK1 | Thêm sản phẩm mới vào kho và sản phẩm được thêm vào thành công | 1. Chọn ảnh 2. Điền đầy đủ thông tin sản phẩm 3. Thêm sản phẩm |  | Sản phẩm được thêm vào kho | Sản phẩm được thêm vào kho | Pass |
| TH01 | Kiểm tra trả hàng và cập nhật lại số lượng hàng có trong kho | 1. Chọn hóa đơn cần trả hàng 2. Điền giá tiền hoàn trả 3. Trả hàng |  | Sản phẩm được lưu vào trả hàng và cập nhật số lượng hàng | Sản phẩm được lưu vào trả hàng nhưng không cập nhật lại số lượng hàng | Fail |

## Integration Test (Kiểm thử tích hợp)

- **Tại sạo lại phải thực hiện kiểm thử tích hợp**

- **Mặc dù mỗi module đều được kiểm thử đơn vị (Unit test) nhưng các lỗi vẫn còn tồn tại với các nguyên nhân sau:**

* Một Module nói chung được thiết kế bởi một lập trình viên có hiểu biết và logic lập trình có thể khác với các lập trình viên khác. Kiểm thử tích hợp là cần thiết để đảm bảo tính hợp nhất của phần mềm.
* Tại thời điểm phát triển module vẫn có thể có thay đổi trong spec của khách hàng, những thay đổi này có thể không được kiểm tra ở giai đoạn unit test trước đó.
* Giao diện và cơ sở dữ liệu của các module có thể chưa hoàn chỉnh khi được ghép lại.
* Khi tích hợp hệ thống các module có thể không tương thích với cấu hình chung của hệ thống.
* Thiếu các xử lý ngoại lệ có thể xảy ra.

- Tiến hành kiểm thử tích hợp Top-Down

* Việc kiểm tra diễn ra từ trên xuống dưới, theo dòng điều khiển hoặc cấu trúc kiến ​​trúc

## System Test (Kiểm thử toàn bộ hệ thống)

- Sau khi tiến hành Unit Testing và Integration Test chúng ta tiến hành System testing,

Tiến hành test qua các loại test như:

* Functionality Testing (Test chức năng): Đảm bảo chức năng của sản phẩm hoạt động đúng như yêu cầu, trong khả năng của hệ thống
* Recoverability Testing (Test khả năng phục hồi): Đảm bảo hệ thống có khả năng phục hồi tốt từ các lỗi đầu vào khác nhau và các tình huống lỗi khác.
* Interoperability Testing (Test khả năng tương tác): Đảm bảo hệ thống có thể vận hành tốt với các sản phẩm thứ ba hay không.
* Performance Testing (Kiểm tra hiệu năng): Đảm bảo hiệu năng của hệ thống dưới các điều kiện khác nhau.
* Scalability Testing (Test khả năng mở rộng): Đảm bảo khả năng mở rộng của hệ thống trong những điều kiện khác nhau như: mở rộng user, địa lý hay tài nguyên.
* Reliability Testing (Test tính tin cậy): Đảm bảo hệ thống có thể vận hành trong thời gian dài mà không có lỗi phát sinh.
* Regression Testing (Test hồi quy): Đảm bảo sự ổn định của hệ thống khi nó được tích hợp với các hệ thống con khác nhau và thực hiện các task bảo trì khác nhau.
* Documentation Testing (Kiểm tra tài liệu): Đảm bảo tài liệu hướng dẫn sử dụng và các tài liệu khác đúng và có thể sử dụng
* \*\*Security Testing \*\*(Test bảo mật): Đảm bảo hệ thống không cho phép sự truy cập trái phép vào dữ liệu hoặc tài nguyên.
* Usability Testing (Test khả năng sử dụng): Đảm bảo hệ thống dễ sử dụng, dễ học và dễ vận hành.

## Acceptance Test (Kiểm thử chấp thuận)

* Cuối cùng chúng ta sẽ Acceptance Test xem hệ thống của chúng ta có đạt yêu cầu kỹ thuật hay không
* Điều kiện cần :
* Cần phải đảm bảo những yêu cầu nghiệp vụ quan trọng của ứng dụng hoạt động;
* Phần mềm đã hoàn thiện tốt nhất có thể;
* Các khâu kiểm thử như Unit Testing, Integration Testing và System Testing đều đã hoàn thành;
* Không tồn tại lỗi quan trọng trong hệ thống;
* Lỗi về thẩm mỹ đã được chấp nhận trước kiểm thử chấp nhận;
* Regression Testing phải được hoàn thành, không có lỗi lớn;
* Mọi lỗi đã phát hiện đều phải được sửa, kiểm tra kỹ trước kiểm thử chấp nhận;
* Môi trường Acceptance Test đã được chuẩn bị sẵn sàng;
* Nhà phát triển cần phải chắc chắn rằng hệ thống đã sẵn sàng thực hiện kiểm thử chấp nhận

- Quy trình thực hiện

* Phân tích những yêu cầu nghiệp vụ của phần mềm;
* Tạo kế hoạch Acceptance Test;
* Xác định mọi kịch bản kiểm thử;
* Tạo ra các trường hợp kiểm thử chấp nhận;
* Chuẩn bị data test cần đảm bảo giống với data thật nhất có thể;
* Thực hiện kiểm thử chấp nhận;
* Ghi nhận kết quả cuối cùng;
* Xác nhận các chức năng sản phẩm.