#### BỘ CÔNG THƯƠNG TRƯỜNG ĐẠI HỌC



CÔNG NGHIỆP THỰC PHẨM TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

## ĐỀ TÀI: QUẢN LÝ CỬA HÀNG BÁN GIÀY

### **Giảng viên hướng dẫn :Trần Văn Thọ**

Sinh viên thực hiện :

1. 2001198080-Lê Thanh Thái
2. 2001190509-Võ Thị Thu Hà
3. 2001190580-Ngô Thanh Huyền

### TP. HỒ CHÍ MINH – 2022

LỜI CẢM ƠN

Trước tiên nhóm đề tài xin chân thành cảm ơn Khoa Công Nghệ Thông Tin Trường Đại Học Công Nghiệp Thực phẩm đã tạo điều kiện tốt cho nhóm đề tài thực hiện tốt đề tài đồ án này. Nhóm đề tài xin chân thành cảm ơn thầy Trần Văn Thọ đã tận tình giảng dạy, trang bị cho nhóm những kiến thức quý báu trong những buổi học qua, giúp nhóm có một nền tảng kiến thức vững chắc để hoàn thành đề tài này. Mặc dù nhóm đã có nhiều cố gắng để thực hiện đề tài một cách hoàn chỉnh nhất.Song do thời gian có hạn, trình độ hiểu biết và nhận thức còn chưa cao cho nêntrong đồ án không thể tránh khỏi những thiếu sót, em rất mong nhận được sựđóng góp ý kiến của các thầy cô và bạn bè đã giúp đỡ để nhóm em có thể hoàn thiện đồ án nàytốt ơn.

Chúng em xin chân thành cảm ơn!

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc104414572)

[MỤC LỤC 3](#_Toc104414573)

[MỞ ĐẦU 4](#_Toc104414574)

[CHƯƠNG 1:TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM 6](#_Toc104414578)

[1.1. Các khái niệm cơ bản về kiểm thử phần mềm. 6](#_Toc104414579)

[1.1.1.1. Kiểm thử tĩnh-Static testing. 6](#_Toc104414580)

[1.1.1.2. Kiểm thử động-Dynamic testing. 6](#_Toc104414583)

[1.2. Nguyên tắc kiểm thử phần mềm. 13](#_Toc104414586)

[CHƯƠNG 2 :THIẾT KẾ TEST PLAN 14](#_Toc104414587)

[2.1. Khái niệm. 14](#_Toc104414588)

[2.2. Vai trò của thiết kế test-plan. 14](#_Toc104414589)

[2.3. Các định nghĩa và các từ viết tắt được dùng. 15](#_Toc104414590)

[2.4. Những người sử dụng tài liệu này. 15](#_Toc104414591)

[CHƯƠNG 3 THIẾT KẾ TEST CASE VÀ TEST DATA 19](#_Toc104414592)

[3.1.WBS 19](#_Toc104414593)

[3.2. Đặt tả 19](#_Toc104414594)

[3.3. Kiểm thử hộp đen 19](#_Toc104414602)

[3.4. Kiểm tra chức năng 19](#_Toc104414604)

[CHƯƠNG 4: ÁP DỤNG KIỂM THỬ THỰC TẾ 20](#_Toc104414606)

[4.1.Unit Test: 20](#_Toc104414607)

[4.2.UI automation Test: 20](#_Toc104414608)

[TÀI LIỆU THAM KHẢO 21](#_Toc104414612)

# MỞ ĐẦU

Trong sự phát triển không ngừng của xã hội thì nhu cầu của con người về cuộc sống ngày càng cao.Từ ý nghĩa thực tế đó mà đã nảy sinh ra nhiều loại hình dịch vụ để phần nào đáp ứng được những nhu cầu thiết yếu của con người. Các dịch vụ đó đa dạng về nhiều mặt với nhiều cách đáp ứng khác nhau. Và một loại hình dịch vụ khá phổ biến hiện nay đó là bán hàng qua mạng.

Nếu như trước đây máy tính trở nên xa lạ với mọi người thì ngày nay máy tính trở thành một người bạn đắc lực không thể thiếu. Người ta có thể thông qua máy tính làm được mọi chuyện tưởng như là không thể, không những thế mà hiệu quả công việc còn rất cao nó thể hiện như độ chính xác nhanh chóng và thuận tiện.

Những minh chứng cho sự tiện lợi của máy tính được thông qua dịch vụ mua bán trên mạng. Mặc dù dịch vụ mua bán này đối với các nước phát triển trên thế giới không phải là mới mẻ gì. Tuy nhiên ở Việt Nam ta nó là bước đi chập chững ban đầu.

Khi vào với dịch vụ mua bán qua mạng, người ta thấy được sự tiện lợi mà nhiều dịch vụ khác không có được, tạo cho con người có sự tin cậy, mà nhất là trong xã hội ngày nay khi con người lấy công việc làm nền tảng cho sự phát triển thì dịch vụ mua bán qua mạng ngày càng phát huy được thế mạnh của nó.

Mua bán được hoạt động với nhiều hình thức khác nhau nhưng để có một dẫn chứng cụ thể nói lên sự tiện lợi này trong khuân khổ của một đề tài luận văn tốt nghiệp tôi xin xây dựng và cài đặt một hệ thống mua bán sản phẩm giầy dép qua mạng. Nhiệm vụ chính của tôi là xây dựng còn sự tiện lợi của nó chỉ khi dùng đến nó thì mới có thể biết hết được.

Mục đích của tôi khi xây dựng nên một ứng dụng bán hàng qua mạng với phương thức thanh toán có thể là tiền mặt, thẻ tín dụng, chuyển khoản,…thông qua các hệ thống ngân hàng.

Ý tưởng là như vậy tuy nhiên không có môi trường thử nghiệm cho ứng dụng. Vậy cho nên tôi chỉ mô phỏng quá trình thanh toán mà thôi, trên thực tế để xây dựng tốt ứng dụng thì phức tạp hơn nhiều và cần nhiều sự hỗ trợ.

Hiện nay với sự phát triển vượt bậc của ngành CNTT, một công nghệ mới đang được đưa vào ứng dụng đó là dùng giao thức WAP để xây dựng mạng không dây. Như thế dù ở bất cứ đâu chúng ta chỉ cân một chiếc mobile là có thể truy cập vào mạng Internet một cách dễ dàng. Đây cũng là một điều thúc đẩy hay gợi ý lớn để tôi tiếp tục xây dựng nên website này ngày một hoàn thiện hơn về phương thức thanh toán cũng như phương thức giao dịch qua mạng.

## CHƯƠNG 1:TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

* 1. Các khái niệm cơ bản về kiểm thử phần mềm.

1.1.1.Kiểm thử phần mềm là gì?

Thử phần mềm là quá trình khảo sát một hệ thống hay thành phần dưới những điều kiện xác định, quan sát và ghi lại các kết quả, và đánh giá một khía cạnh nào đó của hệ thống hay thành phần đó.

Kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm lỗi. (Theo “The Art of Software Testing”-Nghệ thuật kiểm thử phần mềm).

Kiểm thử phần mềm là hoạt động khảo sát thực tiễn sản phẩm hay dịch vụ phần mềm trong đúng môi trường chúng dự định sẽ được triển khai nhằm cung cấp cho người có lợi ích liên quan những thông tin về chất lượng của sản phẩm hay dịch vụ phần mềm nhằm đảm bảo hiệu quả hoạt động tối tưu của phần mềm trong nhiều ngành khác nhau.

Kiểm thử phần mềm đảm bảo sản phẩm phần mềm đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng, yêu cầu của sản phẩm đề đã đặt ra.

Kiểm thử phần mềm cũng cung cấp mục tiêu, cái nhìn độc lập về phần mềm, điều này cho phép việc đánh giá và hiểu rõ các rủi ro khi thực thi phần mềm.

Kiểm thử phần mềm tạo điều kiện cho bạn tận dụng tối đa tư duy đánh giá và sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa nhìn thấy.

**Các phương pháp kiểm thử:**

Có 2 phương pháp kiểm thử chính là: Kiểm thử tĩnh và Kiểm thử động.

* + - 1. Kiểm thử tĩnh-Static testing.

Là phương pháp thử phần mềm đòi hỏi phải duyệt lại các yêu cầu và các đặc tả bằng tay, thông qua việc sử dụng giấy, bút để kiểm tra logic, lần từng chi tiết mà không cần chạy chương trình. Kiểu kiểm thử này thường được sử dụng bởi chuyên viên thiết kế người mà viết mã lệnh một mình.

Kiểm thử tĩnh cũng có thể được tự động hóa. Nó sẽ thực hiện kiểm tra toàn bộ bao gồm các chương trình được phân tích bởi một trình thông dịch hoặc biên dịch mà xác nhận tính hợp lệ về cú pháp của chương trình.

* + - 1. Kiểm thử động-Dynamic testing.

Là phương pháp thử phần mềm thông qua việc dùng máy chạy chương trình để kiểm tra trạng thái tác động của chương trình. Đó là kiểm thử dựa trên các ca kiểm thử xác định bằng sự thực hiện của đối tượng kiểm thử hay chạy các chương trình.

Kiểm thử động kiểm tra cách thức hoạt động của mã lệnh, tức là kiểm tra sự phản ứng vật lý từ hệ thống tới các biến luôn thay đổi theo thời gian. Trong kiểm thử động thực sự bao gồm làm việc với phần mềm, nhập các giá trị đầu vào và kiểm tra xem liệu đầu ra có như mong muốn hay không. Các phương pháp kiểm thử động gồm có kiểm thử Unit-Unit Tests, Kiểm thử tích hợp- Intergrantion Tests, Kiểm thử hệ thống-System Tests, và Kiểm thử chấp nhận sản phẩm-Acceptance Tests.

1.1.2.Các chiến lược kiểm thử.

Ba trong số những chiến lược kiểm thử thông dụng nhất bao gồm: Kiểm thử hộp đen, Kiểm thử hộp trắng và Kiểm thử hộp xám.

1.1.2.1. Kiểm thử hộp đen-Black box testing.

Một trong những chiến lược kiểm thử quan trọng là kiểm thử hộp đen, hướng dữ liệu, hay hướng vào/ra. Kiểm thử hộp đen xem chương tình như là một” hộp đen”. Mục đích của bạn là hoàn toàn không quan tâm về cách cư xử và cấu trúc bên trong của chương trình. Thay vào đó, tập trung vào tìm các trường hợp mà chương tình không thực hiện theo các đặc tả của nó.

Các phương pháp kiểm thử hộp đen:

- Phân tích tương đương-Equivalence partitioning.

- Phân tích giá trị biên -Boudary vâlue analysis.

- Kiểm thử moi cặp-All-pairs testing.

- Kiểm thử fuzz-Fuzz testing.

- Kiểm thử dựa trên mô hình -Model-based testing.

- Kiểm thử dấu vết- Traceability matrix.

- Kiểm thử thăm dò- Exploratory testing.

- Kiểm thử dựa trên đặc tả- Specification-base testing.

Kiểm thử dựa trên đặc tả tập trung vào kiểm tra tính thiết thực của phần mềm theo những yêu cầu thích hợp. Do đó, kiêm thử viên nhập dữ liệu vào, và chỉ thấy dữ liệu ra từ đối tượng kiểm thử. Mức kiểm thử này thường yêu cầu các ca kiểm thử triệt để được cung cấp cho kiểm thử viên mà khi đó có thể xác minh là đối với dữ liệu đầu vào đã cho, giá trị đầu ra(hay cách thức hoạt động) nó giống với giá trị mong muốn đã được xác định trong ca kiểm thử đó hay không. Kiểm thử dựa trên đặc tả là cần thiết, nhưng không đủ để ngăn chặn những rủi ro chắc chắn.

Ưu, nhược điểm:

Kiểm thử hộp đen không có mối liên quan nào tới mã lệnh, và kiểm thử viên chỉ rất đơn giản tâm niệm là: một mã lệnh phải có lỗi. Sử sụng nguyên tắc”Hãy đòi hỏi và bạn sẽ được nhận”, những kiểm thử viên hộp đen tìm ra lỗi mà những lập trình viên đã không tìm ra. Những mặc khác, người ta cũng nói kiểm thử hộp đen “gióng như là đi trong bóng tối mà không có đèn vậy”, bởi vì kiểm thử viên không biết các phâng mềm được kiểm tra thực sự được xây dựng như thế nào. Đó là lý do mà có nhiều trường hợp mà một kiểm thử viên hộp đen viết rất nhiều ca kiểm thử để kiểm tra một thứ gì đó mà đáng lẽ có thể chỉ cần kiểm tra bằng một ca kiểm thử duy nhất, và/hoặc một số sản phẩm của chương trình không được kiểm tra chút nào.

Do vậy, kiểm thử hộp đen dó ưu điểm của “ một sự đánh giá khách quan”, mặc khác nó lại có nhược điểm của “thăm dò”.

1.1.2.2. Kiểm thử hộp trắng-White box testing.

Là mộ chiến lược kiểm thử khác, trái ngược hoàn toàn với kiểm thử hộp đen, kiểm thử hộp trắng hay kiểm thử hướng logic cho phép bạn khảo sát cấu trúc bên trong của chương trình. Chiến lược này xuất phát từ dữ liệu kiểm thử bằng sự kiểm thử tính logic của chương trình. Kiểm thử viên sẽ truy cập vào cấu trúc dữ liệu và giải thuật bên trong chương trình.

Các phương pháp kiểm thử hộp trắng

- Kiểm thử giao diện lập trình ứng dụng- API testing( application programming interface): là phương pháp kiểm thử của ứng dụng sử dụng các API công khai và riêng tư.

- Bao phủ mã lệnh-Code coverage: tạo các kiểm tra để áp ứng một số tiêu chuẩn về bao phủ mã lệnh.

- Các phươn pháp gán lỗi-Fault injection.

- Các phương pháp kiểm thử hoán chuyển-Mutation testing methods.

- Kiểm thử tĩnh-Statac testing: kiểm thử hộp trắng bao gồm mọi kiểm thử tĩnh.

Phương pháp kiểm thử hộp trắng cũng có thể được sử dụng để đánh giá sự hoàn thành của một bộ kiểm thử mà được tạo cùng với các phương pháp kiểm thử hộp đen. Điều này chho phép các nhóm phần mềm khảo sát các phần của 1 hệ thống ít khi được kiểm tra và đảm bảo rằng những điểm chức năng quan trọng nhất đã được kiểm tra.

1.1.2.3. Kiểm thử hộp xám-Gray box testing.

Kiểm tra hộp xám đòi hỏi phải có sự truy cập tới cấu trúc dữ liệu và giải thuật bên trong cho những mục đích thiết kế các ca kiểm thử, nhưng là kiểm thử ở mức người sử dụng hay mức hộp đen. Việc thao tác tới dữ liệu đầu vào và định dạng dữ liệu đầu ra là không rõ ràng, giống như một chiếc “hộp xám”, bởi vì đầu vào và đầu ra rõ ràng là ở bên ngoài “hộp đen” mà chúng ta vẫn gọi về hệ thống được kiểm tra.

Sự khác biệt này đặc biệt quan trọng khi quản lý kiểm thử tích hợp-Intergartion testing giữa hai modun mã lệnh được viết bởi hai chuyên viên thiết kế khác nhau, trong đó chỉ giao diện là được đưa ra để kiểm thử. Kiểm thử hộp xám có thể cũng bao gồm cả thiết kế đối chiếu để quyết định.

1.1.3.Các cấp độ kiểm thử phần mềm.

#### Kiểm thử phần mềm gồm các cấp độ: Kiểm thử đơn vị, Kiểm thử tích hợp, Kiểm thử hệ thống và Kiểm thử chấp nhận sản phẩm.

#### Sơ đồ cấp độ kiểm thử:

#### 

1.1.3.1. Kiểm thử đơn vị-Unit Test.

Một đơn vị là một thành phần phần mềm nhỏ nhất mà ta có thể kiểm thử được. Ví dụ: các hàm(Function), thủ tục (Procedure), lớp (Class) hay phương thức (method) đều có thể được xem là Unit.

Vì Unit đuwocj chọn để kiểm tra thường có kích thước nhỏ và chức năng hoạt động đơn giản, chúng ta không khó khăn gì trong việc tổ chức kiểm thử, ghi nhận và phân tích kết quả kiểm thử. Nếu phát hiện lỗi, việc xác định nguyên nhân và khắc phục cũng tương đối dễ dàng vì chỉ khoanh vùng trong một đơn thể Unit đang kiểm tra. Một nguyên lý đúc kết từ thực tiễn: thời gian tốn cho Unit Test sẽ được đền bù bằng việc tiết kiệm rất nhiều thời gian và chi phí cho việc kiểm thử và sửa lỗi ở các mức kiểm thử sau đó.

Unit test thường do lập trình thực hiện. Công đoạn này cần được thực hiện càng sớm càng tốt trong giai đoạn viết code và xuyên suốt chi kỳ phát triển phần mềm.

Thông thường, Unit Test đòi hỏi kiểm thử viên có kiến thức về thiết kế và code của chương tình. Mục đích của Unit Test là bảo đảm thông tin được xử lý và xuất( khỏi Unit) là chính xác, trong mối tương quan với dữ liệu nhập và chức năng của Unit. Điều này thường đòi hỏi tất cả các nhánh bên trong Unit đều phải được kiểm tra để phát hiện nhánh phát sinh lỗi. Một nhánh thường là một chuỗi các lệnh được thực thi trong một Unit. Ví dụ:Chuối các lệnh sau điều kiện If và nằm giữa then...else là một nhánh. Thực tế việc chọn lựa các nhánh để đơn giản hóa việc kiểm thử và quét hết Unit đòi hỏi phải có ký thuật, đoi khi phải dùng thuật toán để chọn lựa.

Cùng với các mục kiểm thử khác, Unit Tes cũng đòi hỏi phải chuẩn bị trước các ca kiểm thử (Test case) hoặc kịch bản kiểm thử(Test script), trong đó chỉ định rõ dữ liệu đầu vào, các bước thực hiện và dữ liệu đầu ra mong muốn. Các Test case và Tesr script này nên được giữ lại để tái sử dụng.

1.1.3.2. Kiểm thử tích hợp-Intergration Test.

Integration test kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành. Trong khi Unit Test kiểm tra các thành phần và Unit riêng lẻ thì Intagration Test kết hợp chúng lại với nhau và kiểm tra sự giao tiếp giữa chúng.

Hai mục tiêu chính của Integration Test :

- Phát hiện lỗi giao tiếp xảy ra giữa các Unit.

- Ticha hợp các Unit đơn lẻ thành các hệ thống nhỏ (Subsystem) và cuối cùng là nguyên hệ thống hoàn chỉnh (System) chuẩn bị cho kiểm thử ở mức hệ thống (System Test).

Trong Unit Test, lập trình viên cố gắng phát hiện lỗi liên quan đến chức năng và cấu trúc nội tại của Unit. Có một số phép kiểm thử đơn giản trên giao tiếp giữa Unit với các thành phần liên quan khác, tuy nhiên mọi giao tiếp liên quan đến Unit chỉ thật sự được kiểm tra đầy đủ khi các Unit tích hợp với nhau trong khi thực hiện Integration Test.

Trừ một số ít ngoại lệ, Integration Test chỉ nên thực hiện trên những Unit đã được sửa chữa. Một số người hiểu sai rằng Unit một khi đã qua giai đoạn Unit Test với các giao tiếp giả lập thì không cần phải thực hiện Integration Test nữa. Thực tế việc tích hợp giữa các Unit dẫn đến những tình huống hoàn toàn khác. Một chiến lược cần quan tâm trong Integration Test là nên tích hợp dần từng Unit. Một Unit tại một thời điểm được tích hợp vào một nhóm các Unit khác đã tích hợp trước đó và đã hoàn tất các đợt Integration Test trước đó.

Lúc này, ta chỉ cần kiểm thử giao tiếp của Unit mới thêm vào với hệ thống các Unit đã tích hợp trước đó, điều này sẽ làm cho số lượng ca kiểm thử giảm đi rất nhiều, và sai sót sẽ giảm đáng kể.

Có 4 loại kiểm thử trong Integration Test:

- Kiểm thử cấu trúc (Structure Test): Tương tự White Box Test, kiểm thử cấu trúc nhằm bảo đảm các thành phần bên trong của một chương trình chạy đúng và chú trọng đến hoạt động của các thành phần cấu trúc nội tại của chương trình chẳng hạn các câu lệnh và nhánh bên trong.

- Kiểm thử chức năng (Functional Test): Tương tự Black Box Test, kiểm thử chức năng chỉ chú trọng đến chức năng của chương trình, mà không quan tâm đến cấu trúc bên trong, chỉ khảo sát chức năng của chương trình theo yêu cầu kỹ thuật.

- Kiểm thử hiệu năng (Performance Test): Kiểm thử việc vận hành của hệ thống.

- Kiểm thử khả năng chịu tải (Stress Test): Kiểm thử các giới hạn của hệ thống.

1.1.3.3. Kiểm thử hệ thống-System Test.

Mục đích System Test là kiểm thử thiết kế và toàn bộ hệ thống (sau khi tích hợp) có thỏa mãn yêu cầu đặt ra hay không. System Test bắt đầu khi tất cả các bộ phận của phần mềm đã được tích hợp thành công. Thông thường loại kiểm thử này tốn rất nhiều công sức và thời gian. Trong nhiều trường hợp, việc kiểm thử đòi hỏi một số thiết bị phụ trợ, phần mềm hoặc phần cứng đặc thù, đặc biệt là các ứng dụng thời gian thực, hệ thống phân bố, hoặc hệ thống nhúng. Ở mức độ hệ thống, người kiểm thử cũng tìm kiếm các lỗi, nhưng trọng tâm là đánh giá về hoạt động, thao tác, sự tin cậy và các yêu cầu khác liên quan đến chất lượng của toàn hệ thống.

Điểm khác nhau then chốt giữa Integration Test và System Test là System Test chú trọng các hành vi và lỗi trên toàn hệ thống, còn Integration Test chú trọng sự giao tiếp giữa các đơn thể hoặc đối tượng khi chúng làm việc cùng nhau.

Thông thường ta phải thực hiện Unit Test và Integration Test để bảo đảm mọi Unit và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện System Test. Sau khi hoàn thành Integration Test, một hệ thống phần mềm đã được hình thành cùng với các thành phần đã được kiểm tra đầy đủ. Tại thời điểm này, lập trình viên hoặc kiểm thử viên bắt đầu kiểm thử phần mềm như một hệ thống hoàn chỉnh. Việc lập kế hoạch cho System Test nên bắt đầu từ giai đoạn hình thành và phân tích các yêu cầu. System Test kiểm thử cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật. Mức kiểm thử này đặc biệt thích hợp cho việc phát hiện lỗi giao tiếp với phần mềm hoặc phần cứng bên ngoài, chẳng hạn các lỗi "tắc nghẽn" (deadlock) hoặc chiếm dụng bộ nhớ. Sau giai đoạn System Test, phần mềm thường đã sẵn sàng cho khách hàng hoặc người dùng cuối cùng kiểm thử chấp nhận sản phẩm (Acceptance Test) hoặc dùng thử (Alpha/Beta Test).

Đòi hỏi nhiều công sức, thời gian và tính chính xác, khách quan, System Test thường được thực hiện bởi một nhóm kiểm thử viên hoàn toàn độc lập với nhóm phát triển dự án.

Bản thân System Test lại gồm nhiều loại kiểm thử khác nhau, phổ biến nhất gồm:

- Kiểm thử chức năng (Functional Test): Bảo đảm các hành vi của hệ thống thỏa mãn đúng yêu cầu thiết kế.

-Kiểm thử hiệu năng (Performance Test): Bảo đảm tối ưu việc phân bổ tài nguyên hệ thống (ví dụ bộ nhớ) nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng câu truy vấn, ...

- Kiểm thử khả năng chịu tải (Stress Test hay Load Test): Bảo đảm hệ thống vận hành đúng dưới áp lực cao (ví dụ nhiều người truy xuất cùng lúc). Stress Test tập trung vào các trạng thái tới hạn, các "điểm chết", các tình huống bất thường như đang giao dịch thì ngắt kết nối (xuất hiện nhiều trong kiểm tra thiết bị như POS, ATM...)...

- Kiểm thử cấu hình (Configuration Test).

- Kiểm thử bảo mật (Security Test): Bảo đảm tính toàn vẹn, bảo mật của dữ liệu và của hệ thống.

- Kiểm thử khả năng phục hồi (Recovery Test): Bảo đảm hệ thống có khả năng khôi phục trạng thái ổn định trước đó trong tình huống mất tài nguyên hoặc dữ liệu; đặc biệt quan trọng đối với các hệ thống giao dịch như ngân hàng trực tuyến...

Nhìn từ quan điểm người dùng, các cấp độ kiểm thử trên rất quan trọng: Chúng bảo đảm hệ thống đủ khả năng làm việc trong môi trường thực.

Lưu ý là không nhất thiết phải thực hiện tất cả các loại kiểm thử nêu trên. Tùy yêu cầu và đặc trưng của từng hệ thống, tuỳ khả năng và thời gian cho phép của dự án, khi lập kế hoạch, người Quản lý dự án sẽ quyết định áp dụng những loại kiểm thử nào.

1.1.3.4. Kiểm thử chấp nhận sản phẩm-Acceptance Test.

Thông thường, sau giai đoạn System Test là Acceptance Test, được khách hàng thực hiện (hoặc ủy quyền cho một nhóm thứ ba thực hiện). Mục đích của Acceptance Test là để chứng minh phần mềm thỏa mãn tất cả yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm (và trả tiền thanh toán hợp đồng).

Acceptance Test có ý nghĩa hết sức quan trọng, mặc dù trong hầu hết mọi trường hợp, các phép kiểm thử của System Test và Acceptance Test gần như tương tự, nhưng bản chất và cách thức thực hiện lại rất khác biệt.

Đối với những sản phẩm dành bán rộng rãi trên thị trường cho nhiều người sử dụng, thông thường sẽ thông qua hai loại kiểm thử gọi là kiểm thử Alpha – Alpha Test và kiểm thử Beta – Beta Test. Với Alpha Test, người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm, lập trình viên sẽ ghi nhận các lỗi hoặc phản hồi, và lên kế hoạch sửa chữa. Với Beta Test, phần mềm sẽ được gửi tới cho người dùng để kiểm thử ngay trong môi trường thực, lỗi hoặc phản hồi cũng sẽ gửi ngược lại cho lập trình viên để sửa chữa.

Thực tế cho thấy, nếu khách hàng không quan tâm và không tham gia vào quá trình phát triển phần mềm thì kết quả Acceptance Test sẽ sai lệch rất lớn, mặc dù phần mềm đã trải qua tất cả các kiểm thử trước đó. Sự sai lệch này liên quan đến việc hiểu sai yêu cầu cũng như sự mong chờ của khách hàng. Ví dụ đôi khi một phần mềm xuất sắc vượt qua các phép kiểm thử về chức năng thực hiện bởi nhóm thực hiện dự án, nhưng khách hàng khi kiểm thử sau cùng vẫn thất vọng vì bố cục màn hình nghèo nàn, thao tác không tự nhiên, không theo tập quán sử dụng của khách hàng v.v...

Gắn liền với giai đoạn Acceptance Test thường là một nhóm những dịch vụ và tài liệu đi kèm, phổ biến như hướng dẫn cài đặt, sử dụng v.v... Tất cả tài liệu đi kèm phải được cập nhật và kiểm thử chặt chẽ.

1.1.3.5. Một số cấp độ kiểm thử khác

Ngoài các cấp độ trên, còn một số cấp độ kiểm thử khác như:

Kiểm thử hồi quy – Regression Testing:

Theo chuẩn IEEE610.12-90, kiểm thử hồi quy là “sự kiểm tra lại có lựa chọn của một hệ thống hay thành phần để xác minh là những sự thay đổi không gây ra những hậu quả không mong muốn…”. Trên thực tế, quan niệm này là chỉ ra rằng phần mềm mà đã qua được các kiểm tra trước đó vẫn có thể được kiểm tra lại. Beizer định nghĩa đó là sự lặp lại các kiểm tra để chỉ ra rằng cách hoạt động của phần mềm không bị thay đổi, ngoại trừ tới mức như yêu cầu. Hiển nhiên là sự thỏa hiệp phải được thực hiện giữa sự đảm bảo được đưa ra bởi kiểm thử hồi quy mỗi lần thực hiện một sự thay đổi và những tài nguyên được yêu cầu thực hiện điều đó.

Kiểm thử tính đúng – Correctness testing:

Tính đúng đắn là yêu cầu tối thiểu của phần mềm, là mục đích chủ yếu của kiểm thử. Kiểm thử tính đúng sẽ cần một kiểu người đáng tin nào đó, để chỉ ra cách hoạt động đúng đắn từ cách hoạt động sai lầm. Kiểm thử viên có thể biết hoặc không biết các chi tiết bên trong của các mođun phần mềm được kiểm thử, ví dụ luồng điều khiển, luồng dữ liệu, v.v …. Do đó, hoặc là quan điểm hộp trắng, hoặc là quan điểm hộp đen có thể được thực hiện trong kiểm thử phần mềm.

1.1.4.Các phương pháp kiểm thử con người.

Hai phương pháp kiểm thử con người chủ yếu là Code Inspections và Walkthroughs. Hai phương pháp này bao gồm một nhóm người đọc và kiểm tra theo mã lệnh của chương trình. Mục tiêu của chúng là để tìm ra lỗi mà không gỡ lỗi.

Một Inspection hay Walkthrough là 1 sự cải tiến của phương pháp kiểm tra mà lập trình viên đọc chương trình của họ trước khi kiểm thử nó. Inspections và Walkthroughs hiệu quả hơn là bởi vì những người khác sẽ kiểm thử chương trình tốt hơn chính tác giả của chương trình đó.

Inspections/Walkthroughs và kiểm thử bằng máy tính bổ sung cho nhau. Hiệu quả tìm lỗi sẽ kém đi nếu thiếu đi 1 trong 2 phương pháp. Và đối với việc sửa đổi chương trình cũng nên sử dụng các phương pháp kiểm thử này cũng như các kỹ thuật kiểm thử hồi quy.

* 1. Nguyên tắc kiểm thử phần mềm.

Để kiểm thử đạt hiệu quả thì khi tiến hành kiểm thử phần mềm cần phải tuân thủ một số quy tắc sau:

Quy tắc 1: Kiểm thử chứng minh sự hiện diện của lỗi.

Quy tắc 2: Kiểm thử toàn bộ là không thể.

Quy tắc 3: Kiểm thử càng sớm càng tốt.

Quy tắc 4: Lỗi thường được phân bố tập trung.

Quy tắc 5: Nghịch lý thuốc trừ sâu.

Quy tắc 6: Kiểm thử phù thuộc vào ngữ cảnh.

Quy tắc 7: Quan niệm sai lầm về việc “Hết lỗi”.

## CHƯƠNG 2 :THIẾT KẾ TEST PLAN

2.1. Khái niệm.

Test Plan là một tài liệu chi tiết mô tả chiến lược kiểm thử. Mục tiêu kiểm thử, tài nguyên (nhân lực, phần mềm, phần cứng) cần thiết để kiểm thử, lịch kiểm thử, ước lượng kiểm thử và bàn giao sản phẩm kiểm thử. Test Plan đóng vai trò là một kế hoạch chi tiết để tiến hành các hoạt động kiểm thử phần mềm như một quy trình xác định, được giám sát và kiểm soát từng bước bởi Test Manager.

2.2. Vai trò của thiết kế test-plan.

Test plan đóng vai trò như một kế hoạch chi tiết để tiến hành các hoạt động kiểm thử phần mềm, được giám sát và kiểm soát từng bước bởi người quản lý kiểm thử.

Lập Test Plan có nhiều lợi ích:

-Test Plan giúp chúng ta xác định efforts cần thiết để xác nhận chất lượng của ứng dụng đang kiểm thử.

-Giúp những người ngoài nhóm kiểm thử như developers, quản lý doanh nghiệp, khách hàng hiểu chi tiết về kiểm thử.

- Test Plan giống như một cuốn sách quy tắc, cần phải được tuân theo.

- Các khía cạnh quan trọng như ước lượng kiểm thử, phạm vi kiểm thử. Chiến lược kiểm thử được ghi lại trong Test Plan, do đó nó có thể được quản lý nhóm xem xét và sử dụng lại cho các dự án khác.

- Test plan hướng dẫn những quy tắc và điều luật cần tuân theo để ó những hiệu quả tốt nhất

- Nhóm quản lý có thể xem xét và sử dụng lại test plan cho các dự án khác nhau vì những khía cạnh quan trọng như ước tính kiểm thử, phạm vi kiểm thử, chiến lược kiểm thử được ghi lại trong test plan.

KẾ HOẠCH KIỂM THỬ (7 thành phần)

Bước 1 : Introduction (giới thiệu)

1.1.Tổng quan

Web bán giày là một trang web dành cho tất cả mọi người để xem, mua và sử dụng dịch vụ của trang web.

1.2.Phạm vi

Tài liệu kế hoạch kiểm thử này được áp dụng cho việc kiểm thử những chức năng của trang

web Bán Giày .

Các dịch vụ, chức năng cần kiểm tra:

- Tìm kiếm sản phẩm

- Tài khoản

- Chi tiết sản phẩm

- Giỏ hàng

-Thanh toán

2.3. Các định nghĩa và các từ viết tắt được dùng.

Các định nghĩa:

|  |  |
| --- | --- |
| Thuật ngữ | Định Nghĩa |
| cuasneaker | Tên một trang web mua bán giày |

2.4. Những người sử dụng tài liệu này.

Tài liệu này được sử dụng cho Test Manager, Test Designer và Tester.

Bước 2: Requirements for test (Các yêu cầu cần kiểm thử)

2.1. Yêu cầu chức năng.

1.Tìm kiếm

- Tìm kiếm sản phẩm theo tên và màu

2. Tài khoản

- Đăng nhập

- Đăng xuất

- Đăng ký

- Lấy thông tin khách hàng

3. Chi tiết sản phẩm

- Hiển thị thông tin chi tiết của sản phẩm

4. Giỏ hàng

- Hiển thị danh sách các sản phẩm mà khách hàng đang chọn

5. Thanh toán

- Gởi mail về cho khách hàng

2.2. Yêu cầu phi chức năng.

-Upload sản phẩm

-Thông tin về lượng sản phẩm đã bán

...

2.3. Công cụ kiểm thử.

|  |  |  |  |
| --- | --- | --- | --- |
| Hoạt động | Công cụ | Nhà cung cấp | Phiên bản |
| Quản lý Test Case | Microsoft Office Excel | Microsoft | 2016 |
| Quản lý Configuration | Microsoft Office Word | Microsoft | 2016 |
| Defect Tracking | Microsoft Office Excel | Microsoft | 2016 |
| Quản lý tiến độ kiểm thử | Microsoft Project Professional | Microsoft | 2016 |

#### Bước 3: Acceptance criteria (Các tiêu chí chấp nhận sản phẩm)

#### 3.1. Tiêu chí tạm dừng.

#### Nếu các thành viên trong nhóm báo cáo rằng có 40% trường hợp thử nghiệm thất bại, tạm dừng thử nghiệm cho đến khi nhóm phát triển sửa chữa tất cả trường hợp thất bại.

#### 3.2. Tiêu chí hoàn thành.

#### Chỉ định các tiêu chí biểu thị hoàn thành thành công giai đoạn thử nghiệm

#### Tốc độ chạy bắt buộc là 100% trừ khi có lý do rõ ràng

#### Tỷ lệ vượt qua là 80%, đạt được tỷ lệ vượt qua là bắt buộc.

#### Bước 4: Test strategy (Các chiến lược kiểm thử)

#### Kiểm thử chức năng:

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đẳm bảo các chức năng được kiểm tra hoạt động chính xác theo đặc tả yêu cầu |
| Kỹ thuật | Thực thi tất cả các trường hợp cơ thể có cho mỗi nhóm chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định:-Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng-Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng |
| Tiêu chuẩn dừng | Tất cả các testcase đã được thiết kế đều được thực thiTất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho developer khắc phục. |
| Chiệu trách nhiệm kiểm thử | Test Designer/Tester |
| Cách kiểm thử | Kiểm tra bằng tay thủ công, tuần tự theo các bước định nghĩa trong testcase. |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử |

#### Bước 5: Resources for testing (Nguồn lực cho kiểm thử)

#### 5.1. Nguồn nhân lực: Vai trò và trách nhiệm

|  |  |
| --- | --- |
| Thành viên | Vai trò |
| Lê Thanh Thái | Test Manager / Test Designer / Tester: Lập kế hoạch kiểm thử, quản lý tiến độ hoạt động kiểm thử, thiết kế các testcase bổ sung và thực thi các testcase bổ sung |
| Võ Thị Thu Hà | Test Designer / Tester: Thiết kế và viết các testcase, thực thi các testcase cho 2 chức năng Quản lý hồ sơ cá nhân và Mở bài hát / Video / Playlist, xem lại Test Plan |
| Ngô Thanh Huyền | Test Designer / Tester: Thiết kế và viết các test case, thực thi các testcase cho 2 chức năng Tìm kiếm và đăng nhập tài khoản, xem lại Test Plan |
| Lê Thanh Thái | Test Designer / Tester: Thiết kế và viết các testcase bổ sung, thực thi các testcase bổ sung, đọc các tài liệu tham khảo để cung cấp thông tin cần thiết cho nhóm, xem lại Test Plan |

#### 5.2. System: Hardware & Sofware (Nguồn lực hệ thống)

#### 5.2.1. Phần cứng:

|  |  |  |  |
| --- | --- | --- | --- |
| CPU | RAM | SSD | Architecture |
| Intel(R)) i7-2.30 GHz | 8GB | 225GB | 64bit |

#### 5.2.2.Phần mềm:

|  |  |  |
| --- | --- | --- |
| Tên phần mềm | Phiên bản | Loại |
| IE11 | 8.0.6001.18702IC | Trình duyệt Web |
| Mozilla Firefox | 70.0.1 | Trình duyệt Web |
| Google Chrome | 83.0.4103.97 | Trình duyệt Web |
| Microsoft Windows 10 Professional | 10 | Hệ điều hành |

#### Các chức năng của trang Web được kiểm tra trên các trình duyệt IE11, Mozilla Firefox 77.0.1 và Google Chrome 83.0.4103.97. Các trình duyệt này sử dụng phần mềm tích hợp Adobe Flash Player 12.822.109 để chạy các bài hát. Hệ điều hành được sử dụng là Microsoft Windows 10 Professional.

#### 5.3. Environment( môi trường test)

Test trên môi trường web,…

#### Bước 6: Test milestones (Các mốc kiểm thử)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Milestone | Deliverables | Duration | Start Date | End Date |
| Lập kế hoạch kiểm thử | Tài liệu Test Plan | 5 ngày | 01/04/2022 | 05/04/2022 |
| Xem lại các tài liệu | Tài liệu Tes Plan | 5 ngày | 06/04/2022 | 10/04/2022 |
| Thiết kế các testcase | Tài liệu Testcase | 3 ngày | 011/04/2022 | 13/04/2022 |
| Viết các testcase | Tài liệu Testcase | 5 ngày | 014/04/2022 | 18/04/2022 |
| Xem lại các testcase | Tài liệu Testcase | 3 ngày | 19/04/2022 | 21/04/2022 |
| Thực thi các testcase | Tài liệu Testcase | 3 ngày | 22/04/2022 | 24/04/2022 |
| Ghi nhận và đánh giá kết quả kiểm thử | Tài liệu Testcase | 5 ngày | 25/04/2022 | 30/04/2022 |

#### Bước 7: Deliverables of test: Test Plan, Test Case,Test Reports (sản phẩm cần bàn

#### giao giữa các bên có trách nhiệm liên quan)

CHƯƠNG 3 THIẾT KẾ TEST CASE VÀ TEST DATA

3.1.WBS

3.2. Đặt tả

## Các chức năng chính của phần mềm:

## - Quản lý tìm kiếm

## - Quản lý thông tin khách hàng

## - Quản lý giỏ hàng

## - Chức năng thanh toán

## Những thông tin cần để mua hàng trực tuyến phải thỏa mãn:

## -Thông tin khách hàng không được để trống họ tên, email, di động. Định dạng email phải hợp lệ có chứa đuôi @gmail. Số si động phải chưa đủ 10 ký tự số không cho phép nhập chữ và ký tự đặc biệt, đầu số phải bắt đầu bằng số 0.

3.3. Kiểm thử hộp đen

## (Test-Case.xlsx.)

3.4. Kiểm tra chức năng

(Test-Case.xlsx.)

## CHƯƠNG 4: ÁP DỤNG KIỂM THỬ THỰC TẾ

4.1.Unit Test:

4.2.UI automation Test:

## Nhập địa chỉ website vào thanh Url” <https://cuasneaker.cf/>

## ”.Khi mở Selenium, mặc định nút Record đang được bật.

TÀI LIỆU THAM KHẢO