|  |  |
| --- | --- |
| **NGUYỄN HẢI LUYẾN CÔNG NGHỆ THÔNG TIN** |  |
| TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI  **KHOA CÔNG NGHỆ THÔNG TIN**  **--------------------------** |
|  |
| **ĐỒ ÁN**  **TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH: CÔNG NGHỆ THÔNG TIN** |
| **NGHIÊN CỨU CÔNG CỤ KIỂM THỬ TỰ ĐỘNG KATALON STUDIO** |
|  |
| **GVHD:**ThS. Trần Phương Nhung |
| **Sinh viên:** Nguyễn Hải Luyến |
| **Mã số sinh viên:** 2019606738 |
|  |
| **Hà Nội, 03/2023** |

|  |
| --- |
|  |
| TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI  **KHOA CÔNG NGHỆ THÔNG TIN**  **--------------------------** |
|  |
| **ĐỒ ÁN**  **TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH: CÔNG NGHỆ THÔNG TIN** |
| **NGHIÊN CỨU CÔNG CỤ KIỂM THỬ TỰ ĐỘNG KATALON STUDIO** |
|  |
| **GVHD:**ThS. Trần Phương Nhung |
| **Sinh viên:** Nguyễn Hải Luyến |
| **Mã số sinh viên:** 2019606738 |
|  |
| **Hà Nội, 03/2023** |

**LỜI CẢM ƠN**

*Trong quá trình nghiên cứu đề tài “Nghiên cứu công cụ kiểm thử tự động Katalon studio”, em đã nhận được rất nhiều sự giúp đỡ để hoàn tất. Từ đó, bản thân em đã có thêm thật nhiều kỹ năng và kiến thức giúp ích cho công việc sau này của mình.*

*Trước tiên, em xin gửi lời cảm ơn chân thành cô Trần Phương Nhung đã tận tình hướng dẫn, truyền đạt kiến thức, kinh nghiệm cho em trong suốt quá trình thực hiện đồ án tốt nghiệp này.*

*Xin gửi lời cảm ơn đến quý thầy cô khoa Công nghệ Thông tin, Trường Đại học Công Nghiệp, những người đã truyền đạt kiến thức quý báu cho em suốt trong thời gian học tập vừa qua.*

*Cuối cùng, em rất mong nhận được những lời nhận xét và góp ý quý báu từ thầy cô cũng như các bạn học để bài luận của em có thể hoàn thiện hơn nữa.*

*Em xin chân thành cảm ơn!*

MỤC LỤC

[DANH MỤC CÁC KÍ HIỆU, CÁC CHỮ VIẾT TẮT 1](#_Toc134476014)

[DANH MỤC CÁC HÌNH ẢNH 2](#_Toc134476015)

[LỜI MỞ ĐẦU 3](#_Toc134476016)

[Chương 1. PHẦN MỀM VÀ KIỂM THỬ PHẦN MỀM 5](#_Toc134476017)

[1.1. Phần mềm và khái niệm liên quan 5](#_Toc134476018)

[1.1.1. Phần mềm 5](#_Toc134476019)

[1.1.2. Lỗi phần mềm 5](#_Toc134476020)

[1.1.3. Yêu cầu của khách hàng 5](#_Toc134476021)

[1.1.4. Đặc tả yêu cầu phần mềm 6](#_Toc134476022)

[1.1.5. Chất lượng và độ tin cậy của phần mềm 8](#_Toc134476023)

[1.2. Kiểm thử phần mềm 8](#_Toc134476024)

[1.2.1. Khái niệm 8](#_Toc134476025)

[1.2.2. Vai trò của kiểm thử phần mềm 9](#_Toc134476026)

[1.2.3. Các cấp độ trong kiểm thử phần mềm 10](#_Toc134476027)

[1.2.4. Quy trình kiểm thử phần mềm 17](#_Toc134476028)

[1.2.5. Phân loại kiểm thử phần mềm 20](#_Toc134476029)

[1.2.6. Các mức độ nghiêm trọng của lỗi 23](#_Toc134476030)

[1.2.7. Ca kiểm thử 24](#_Toc134476031)

[1.2.8. Kiểm thử tự động 26](#_Toc134476032)

[1.2.9. Nguyên tắc quan trọng trong kiểm thử phần mềm 28](#_Toc134476033)

[1.3. Các kĩ thuật xác định ca kiểm thử 29](#_Toc134476034)

[1.3.1. Phân vùng tương đương 29](#_Toc134476035)

[1.3.2. Phân tích giá trị biên 29](#_Toc134476036)

[1.3.3. Bảng quyết định 30](#_Toc134476037)

[1.3.4. Đoán lỗi 30](#_Toc134476038)

[Chương 2. KIỂM THỬ ỨNG DỤNG TRÊN NỀN WEB 32](#_Toc134476039)

[2.1. Khái quát về kiểm thử ứng dụng trên nền web 32](#_Toc134476040)

[2.1.1. Khái quát 32](#_Toc134476041)

[2.1.2. Các loại ứng dụng Web 33](#_Toc134476042)

[2.1.3. Đặc tả về chất lượng của một ứng dụng trên nền Web 34](#_Toc134476043)

[2.2. Công việc chính khi kiểm thử ứng dụng Web 34](#_Toc134476044)

[2.2.1. Kiểm thử chức năng 34](#_Toc134476045)

[2.2.2. Kiểm thử khả năng sử dụng 36](#_Toc134476046)

[2.2.3. Kiểm thử sự tương thích 37](#_Toc134476047)

[2.2.4. Kiểm thử hiệu suất 38](#_Toc134476048)

[2.2.5. Kiểm thử bảo mật 38](#_Toc134476049)

[2.3. Một số công cụ hỗ trợ kiểm thử ứng dụng trên nền Web 39](#_Toc134476050)

[2.3.1. Công cụ kiểm thử hiệu năng 39](#_Toc134476051)

[2.3.2. Công cụ kiểm thử bảo mật 41](#_Toc134476052)

[2.3.3. Công cụ kiểm thử chức năng 42](#_Toc134476053)

[Chương 3.KIỂM THỬ ỨNG DỤNG TRÊN NỀN WEB BẰNG CÔNG CỤ KATALON STUDIO 44](#_Toc134476054)

[3.1. Công cụ kiểm thử tự động Katalon Studio 44](#_Toc134476055)

[3.1.1. Giới thiệu chung 44](#_Toc134476056)

[3.1.2. Những ứng dụng liên quan 46](#_Toc134476057)

[3.2. Bài toán thực tế 46](#_Toc134476058)

[3.3. Kiểm thử các chức năng trên trang web với Katalon Studio 47](#_Toc134476059)

[3.3.1. Chức năng “Đăng ký” 47](#_Toc134476060)

[3.3.2. Chức năng “Đăng nhập” 50](#_Toc134476061)

[3.3.3. Chức năng “Đăng xuất” 53](#_Toc134476062)

[3.3.4. Chức năng “Tìm kiếm sản phẩm” 55](#_Toc134476063)

[3.3.5. Chức năng “Xem chi tiết sản phẩm” 57](#_Toc134476064)

[3.3.6. Chức năng “Địa chỉ” 58](#_Toc134476065)

[3.3.7. Chức năng “Giỏ hàng” 62](#_Toc134476066)

[3.3.8. Chức năng “Đặt hàng” 64](#_Toc134476067)

[3.4. Tổng kết 67](#_Toc134476068)

[KẾT LUẬN 68](#_Toc134476069)

[TÀI LIỆU THAM KHẢO 69](#_Toc134476070)

[TÀI LIỆU HƯỚNG DẪN SỬ DỤNG 70](#_Toc134476071)

# DANH MỤC CÁC KÍ HIỆU, CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Viết tắt | Định nghĩa |
| 1 | TC | Test case |
| 2 | CSS | Cascading Style Sheet |
| 3 | HTML | Hypertext Markup Language |
| 4 | UI/UX | User Interface/ User Experience |

# DANH MỤC CÁC HÌNH ẢNH

[Hình 1. 1 Các giai đoạn kiểm thử phần mềm 11](#_Toc134476072)

[Hình 1. 2 Kiểm thử Alpha và Beta 14](#_Toc134476073)

[Hình 1. 3 Quy trình kiểm thử phần mềm 17](#_Toc134476074)

[Hình 1. 4 Ví dụ minh họa về kiểm thử đơn giản 25](#_Toc134476075)

[Hình 3. 1 Giao diện công cụ katalon 44](#_Toc134476146)

[Hình 3. 2 Data cho ca kiểm thử Đăng ký 47](#_Toc134476147)

[Hình 3. 3 Kịch bản kiểm thử cho test case Đăng ký 49](#_Toc134476148)

[Hình 3. 4 Giao diện thực thi test case Đăng ký 49](#_Toc134476149)

[Hình 3. 5 Kết quả thực hiện của test case Đăng ký 50](#_Toc134476150)

[Hình 3. 6 Data cho ca kiểm thử Đăng nhập 51](#_Toc134476151)

[Hình 3. 7 Kịch bản kiểm thử cho test case Đăng nhập 52](#_Toc134476152)

[Hình 3. 8 Giao diện thực thi test case Đăng nhập 53](#_Toc134476153)

[Hình 3. 9 Kết quả thực hiện của test case Đăng nhập 53](#_Toc134476154)

[Hình 3. 10 Kịch bản kiểm thử cho test case Đăng xuất 54](#_Toc134476155)

[Hình 3. 11 Giao diện thực thi test case Đăng xuất 54](#_Toc134476156)

[Hình 3. 12 Kết quả thực hiện của test case Đăng xuất 55](#_Toc134476157)

[Hình 3. 13 Data cho ca kiểm thử Tìm kiếm 55](#_Toc134476158)

[Hình 3. 14 Kịch bản kiểm thử cho test case Tìm kiếm 56](#_Toc134476159)

[Hình 3. 15 Giao diện thực thi test case Tìm kiếm sản phẩm 56](#_Toc134476160)

[Hình 3. 16 Kết quả thực hiện của test case Tìm kiếm 57](#_Toc134476161)

[Hình 3. 17 Kịch bản kiểm thử cho test case Xem chi tiết sản phẩm 57](#_Toc134476162)

[Hình 3. 18 Giao diện thực thi test case Xem chi tiết sản phẩm 58](#_Toc134476163)

[Hình 3. 19 Kết quả thực hiện của test case Xem chi tiết sản phẩm 58](#_Toc134476164)

[Hình 3. 20 Kịch bản kiểm thử cho test case Địa chỉ 60](#_Toc134476165)

[Hình 3. 21 Giao diện thực thi test case Địa chỉ 61](#_Toc134476166)

[Hình 3. 22 Kết quả thực hiện của test case Địa chỉ 61](#_Toc134476167)

[Hình 3. 23 Kịch bản kiểm thử cho test case Giỏ hàng 62](#_Toc134476168)

[Hình 3. 24 Giao diện thực thi test case Giỏ hàng 63](#_Toc134476169)

[Hình 3. 25 Kết quả thực hiện của test case Giỏ hàng 63](#_Toc134476170)

[Hình 3. 26 Kịch bản kiểm thử cho test case Đặt hàng 65](#_Toc134476171)

[Hình 3. 27 Giao diện thực thi test case Đặt hàng 66](#_Toc134476172)

[Hình 3. 28 Kết quả thực hiện của test case Đặt hàng 66](#_Toc134476173)

[Hình 3. 29 Trạng thái các trường hợp thử nghiệm 67](#_Toc134476174)

# LỜI MỞ ĐẦU

**1. Lý do chọn đề tài**

* Trong thế giới công nghệ ngày càng phát triển một cách nhanh chóng hiện nay, thì kéo theo đó là mạng và các phần mềm cũng gia tăng về cả số lượng theo quy mô mở rộng lẫn chất lượng. Nhưng bên cạnh đó cũng nảy sinh ra rất nhiều vấn đề như sản phẩm lỗi, hỏng gây ra ảnh hưởng đến nền kinh tế cho nước nhà, các lỗi xảy ra cũng là do một phần không được kiểm duyệt kĩ càng trước khi đưa ra thị trường để phục vụ mọi khách hàng. Để làm được điều đó, trước một phần mềm hoàn hảo thì không thể thiếu một doanh nghiệp, một bộ phận riêng cần có công tác kiểm thử phần mềm thật kỹ lưỡng nhằm ngăn chặn các lỗi hay hỏng hóc trong phần mềm mà ta chưa kịp nhận ra để cho một sản phẩm trước khi được đưa đến người sử dụng có một sản phẩm hoàn hảo nhất. Tuy nhiên vì phần mềm ngày càng lớn, các module càng nhiều vì vậy để kiểm thử một phần mềm lớn như vậy việc làm thủ công sẽ dẫn đến việc mất thời gian, công sức cũng như chi phí, tiền bạc và chất lượng. Với một công nghệ hiện đại, tự động đã lên ngôi cần thiết cho việc tìm ra lỗi một cách chuyên nghiệp nhất mà đôi khi ta sử dụng bằng tay khó tránh khỏi những sai sót không đáng có. Chính vì vậy em đã chọn đề tài: “Nghiên cứu công cụ kiểm thử tự động Katalon Studio” để ta càng hiểu rõ sâu hơn về nó.

**2. Mục tiêu nghiên cứu**

* Đề tài tập trung nghiên cứu tìm hiểu về kiểm thử phần mềm là gì, nắm rõ được các quy trình, test case, vai trò… Từ đó có nền tảng để tìm hiểu sâu về công cụ hỗ trợ cho chúng ta kiểm thử một cách nhanh chóng và chính xác: công cụ katalon.
* Giúp chúng ta nắm rõ được khái niệm, các quy trình làm việc hay cách để tạo ra một test case trên công cụ từ đó ứng dụng vào để kiểm thử các trang web hay mobile.

**3. Phương pháp nghiên cứu**

* Đề tài tập chung nghiên cứu, tìm hiểu các quy trình chiến lược …
* Sử dụng công cụ katalon để mô tả, thiết kế các testcase
* Sử dụng ngôn ngữ lập trình: Groovy
* Đối tượng: Công cụ Katalon studio

# PHẦN MỀM VÀ KIỂM THỬ PHẦN MỀM

## Phần mềm và khái niệm liên quan

### ****Phần mềm****

* Phần mềm là tập hợp các câu lệnh được viết bằng ngôn ngữ lập trình, các tài liệu có liên quan và các dữ liệu để giải quyết bài toán.
* Phần mềm được Mô tả với ba bộ phận cấu thành:
  + Tập các lệnh (chương trình máy tính) trên máy tính khi được thực hiện sẽ tạo ra các dịch vụ và đem lại những kết quả mong muốn cho người dùng.
  + Cấu trúc dữ liệu (lưu giữ trên các bộ nhớ) làm cho chương trình thao tác hiệu quả với các thông tin thích hợp và nội dung thông tin được số hóa.
  + Các tài liệu để Mô tả thao tác, cách sử dụng và bảo trì phần mềm (hướng dẫn sử dụng, tài liệu kỹ thuật, tài liệu phân tích, thiết kế, kiểm thử,…).

### ****Lỗi phần mềm****

* Một lỗi phần mềm là một lỗi, lỗ hổng, thất bại, hoặc có lỗi trong một chương trình máy tính hoặc hệ thống đó là nguyên nhân nó tạo ra kết quả không chính xác hoặc không mong muốn, hoặc vận hành theo cách không được định hướng trước.

### ****Yêu cầu của khách hàng****

* Định nghĩa yêu cầu phần mềm là tất cả những nhu cầu tính năng sản phẩm mà khách hàng muốn, bao gồm chức năng, hiệu năng, giao diện,… Các yêu cầu thường xoay quanh 4 nhóm sau: yêu cầu về phần cứng; yêu cầu về phần mềm, yêu cầu về data (dữ liệu) và cuối cùng là những yêu cầu về con người. Khi nhận các yêu cầu từ khách hàng, đội phát triển phần mềm phải tiến hành tìm hiểu, phân tích yêu cầu phần mềm để cuối cùng lập ra một bản đặc tả chuẩn chỉnh nhất.

### ****Đặc tả yêu cầu phần mềm****

* Đặc tả yêu cầu là một Mô tả của hệ thống phần mềm được phát triển, đưa ra các yêu cầu chức năng và phi chức năng, và có thể bao gồm một tập hợp các ca sử dụng (use cases) để mô tả tương tác giữa người dùng với phần mềm.
* Đặc tả yêu cầu tạo cơ sở cho một thỏa thuận giữa khách hàng và nhà cung cấp về những gì phần mềm đã làm được tốt cũng như những gì chưa được như mong đợi. Nó cũng cung cấp một cơ sở thực tế để ước tính giá thành sản phẩm, rủi ro và lịch trình.
* Đối với các hệ thống phức tạp có 3 loại tài liệu được tạo ra là: định nghĩa hệ thống, yêu cầu hệ thống và các yêu cầu phần mềm. Đối với sản phẩm phần mềm đơn giản chỉ cần 1 trong 3 tài liệu.

#### Tài liệu đặc tả hệ thống.

* Còn được biết như là tài liệu yêu cầu người dùng hay là tài liệu vận hành ghi lại những yêu cầu hệ thống. Nó xác định yêu cầu hệ thống ở mức cao với cách nhìn từ domain. Độc giả của tài liệu bao gồm hệ thống người dùng hoặc khách hàng. Vì vậy nội dung của nó phải được diễn đạt bằng những từ ngữ của những lĩnh vực riêng. Tài liệu sẽ liệt kê các yêu cầu hệ thống cùng với các thông tin cơ bản về đối tượng hệ thống, môi trường mục tiêu của nó, giả định và các yêu cầu phi chức năng.
* Nó có thể bao gồm mô hình khái niệm được thiết kế để minh họa cho ngữ cảnh hệ thống, sử dụng kịch bản, và các miền thực thể chính, cũng như luồng công việc.

#### Đặc tả yêu cầu hệ thống

* Người phát triển những dự án phần mềm có những thành phần thuần túy là software và những phần non-software - ví dụ như máy bay hiện đại thường tách biệt yêu cầu hệ thống với yêu cầu phần mềm. Theo quan điểm này, yêu cầu hệ thống được quy định, các yêu cầu phần mềm có nguồn gốc từ các yêu cầu hệ thống, và sau đó các yêu cầu đối với các thành phần phần mềm được xác định.

#### Đặc tả yêu cầu phần mềm

* Đặc tả yêu cầu phần mềm tạo cơ sở cho việc thỏa thuận giữa khách hàng và nhà thầu hoặc các nhà cung cấp về những gì sản phẩm phần mềm có làm việc đúng như mong muốn không. Nó cho phép một đánh giá nghiêm ngặt các yêu cầu trước khi có thể bắt đầu vào việc thiết kế và làm giảm việc thiết kế lại. Nó cũng cần cung cấp một cơ sở thực tế để ước tính giá thành sản phẩm, rủi ro, và lịch trình.
* Các tổ chức cũng có thể sử dụng một tài đặc tả yêu cầu phần mềm làm cơ sở để phát triển kế hoạch kiểm tra và xác minh. Đặc tả yêu cầu phần mềm cung cấp một cơ sở thông báo cho chuyển một sản phẩm phần mềm cho người dùng mới hoặc các nền tảng phần mềm. Cuối cùng, nó có thể cung cấp một cơ sở để nâng cao phần mềm. Yêu cầu phần mềm thường được viết bằng ngôn ngữ tự nhiên, nhưng đặc tả yêu cầu phần mềm có thể được bổ sung bằng các Mô tả chính thức hoặc gần chính thức. Lựa chọn các ký hiệu thích hợp và các khía cạnh của kiến trúc phần mềm cụ thể được Mô tả chính xác hơn so với ngôn ngữ tự nhiên.
* Các nguyên tắc chung là ký hiệu nên được sử dụng cho phép các yêu cầu để được mô tả là chính xác càng tốt. Điều này đặc biệt quan trọng đối với các phần mềm an toàn cao và một số loại phần mềm đáng tin cậy khác. Tuy nhiên, sự lựa chọn của các kí hiệu thường được hạn chế bởi việc đào tạo, kỹ năng, và sở thích của các tác giả và độc giả.
* Một số chỉ tiêu chất lượng đã được phát triển có thể được sử dụng liên quan đến chất lượng của đặc tả yêu cầu phần mềm như chi phí, hài lòng, hiệu quả, đúng tiến độ, và tái sản xuất. Chỉ tiêu chất lượng cho đặc tả yêu cầu của cá nhân bao gồm mệnh lệnh, chỉ thị, các pha yếu, tùy chọn, và sự duy trì. Các chỉ số cho các tài liệu đặc tả yêu cầu phần mềm bao gồm kích thước, dễ đọc, đặc tả kỹ thuật, chiều sâu và cấu trúc văn bản.
* Tài liệu đặc tả yêu cầu phần mềm bao gồm kích thước, dễ đọc, đặc tả kỹ thuật, chiều sâu và cấu trúc văn bản.

### ****Chất lượng và độ tin cậy của phần mềm****

* Chất lượng phần mềm là mức độ mà nó đáp ứng được yêu cầu chức năng hoặc đặc tả yêu cầu (hoặc là mức độ mà một hệ thống, thành phần, hay tiến trình đáp ứng được nhu cầu, mong muốn của khách hàng, người dùng.)
* Đảm bảo chất lượng phần mềm là 1 mô hình có kế hoạch và hệ thống của tất cả các hành động cần thiết để đem lại sự tin cậy rằng 1 sản phẩm phần mềm tuân thủ đầy đủ các yêu cầu về kĩ thuật đã đề ra.

## Kiểm thử phần mềm

### Khái niệm

* Kiểm thử phần mềm hay còn được gọi là test đó là một quá trình đánh giá một hệ thống hay là các thành phần của nó với mục đích là xác định xem nó có thỏa mãn những yêu cầu được đưa ra hay không. Hiểu một cách đơn giản, kiểm thử phần mềm là chạy một chương trình để xác nhận bất kì lỗ hổng, lỗi sai hay những yêu cầu bị bỏ quên, những yêu cầu không đúng so với yêu cầu thực tế đề ra
* Mục tiêu kiểm thử
* Tìm các bug phát sinh do dev tạo ra khi code.
* Đạt được sự tự tin và cung cấp thông tin về mức độ chất lượng.
* Để ngăn ngừa lỗi.
* Đảm bảo rằng kết quả cuối cùng đáp ứng các yêu cầu kinh doanh và người sử dụng.
* Để đạt được sự tín nhiệm của khách hàng bằng cách cung cấp cho họ một sản phẩm chất lượng.

### Vai trò của kiểm thử phần mềm

#### Vai trò của kiểm thử phần mềm là gì?

* Có một số lý do rõ ràng cho chúng ta biết lý do [tại sao kiểm thử phần mềm](https://www.devpro.edu.vn/tai-sao-phai-kiem-thu-phan-mem) là quan trọng và những điều chúng ta cần lưu ý nên xem xét khi thử nghiệm bất kỳ sản phẩm hoặc ứng dụng nào.
* Vai trò của kiểm thử phần mềm như sau:
* Kiểm thử phần mềm có nhiệm vụ chỉ ra các lỗi và sai sót hình thành trong các giai đoạn phát triển sản phẩm.
* Kiểm thử là điều cần thiết vì nó đảm bảo độ tin cậy của khách hàng và sự hài lòng của họ về ứng dụng.
* Nó rất quan trọng trong việc đảm bảo chất lượng của sản phẩm. Sản phẩm chất lượng được giao cho khách hàng giúp họ sử dụng hiệu quả hơn.
* Thử nghiệm phần mềm là cần thiết để cung cấp các phần mềm chất lượng cao cho khách hàng hoặc ứng dụng phần mềm đòi hỏi chi phí bảo trì thấp hơn và do đó dẫn đến kết quả chính xác, nhất quán và đáng tin cậy hơn.
* Thử nghiệm là cần thiết cho một hiệu suất hiệu quả của ứng dụng phần mềm hoặc sản phẩm.
* Điều quan trọng là đảm bảo rằng ứng dụng không dẫn đến bất kỳ lỗi nào vì nó có thể rất tốn kém trong tương lai hoặc trong các giai đoạn sau của quá trình phát triển.
* Nó là khâu quan trọng để đảm bảo hoạt động kinh doanh phần mềm phát triển.
* Đảm bảo uy tín của chủ sở hữu sản phẩm đối với khách hàng.

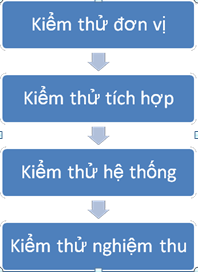
#### Vai trò của chuyên viên kiểm thử

* Trong các giai đoạn lập kế hoạch và chuẩn bị thử nghiệm, người kiểm thử nên xem xét và đóng góp vào các kế hoạch thử nghiệm, cũng như phân tích, xem xét và đánh giá các yêu cầu và thông số kỹ thuật thiết kế. Họ có thể tham gia hoặc thậm chí là những người chính xác định các điều kiện thử nghiệm và tạo ra các thiết kế thử nghiệm, các trường hợp kiểm tra, các đặc tả thủ tục kiểm tra và dữ liệu thử nghiệm, và có thể tự động hóa hoặc giúp tự động hóa các thử nghiệm.
* Họ thường thiết lập các môi trường thử nghiệm hoặc hỗ trợ quản trị hệ thống và nhân viên quản lý mạng khi làm như vậy.
* Khi thực thi thử nghiệm bắt đầu, số lượng người thử nghiệm thường tăng lên, bắt đầu với công việc cần thiết để thực hiện các thử nghiệm trong môi trường thử nghiệm.
* Testers thực hiện và đăng nhập các bài kiểm tra, đánh giá kết quả và các vấn đề tài liệu được tìm thấy.
* Họ theo dõi thử nghiệm và môi trường thử nghiệm, thường sử dụng các công cụ cho tác vụ này và thường thu thập số liệu hiệu suất.
* Trong suốt vòng đời kiểm thử phần mềm, họ xem xét công việc, hỗ trợ lẫn nhau, bao gồm các thông số kiểm tra, báo cáo lỗi và kết quả kiểm tra.

### Các cấp độ trong kiểm thử phần mềm

Kiểm thử phần mềm gồm 4 giai đoạn chính:

* Kiểm thử đơn vị
* Kiểm thử tích hợp
* Kiểm thử hệ thống
* Kiểm thử chấp nhận



Hình 1. 1 Các giai đoạn kiểm thử phần mềm

#### Kiểm thử đơn vị

* Kiểm thử đơn vị nhằm vào kiểm tra đơn vị thiết kế nhỏ nhất – một module phần mềm. Một module hoạt động thường có trao đổi thông tin với module thứ cấp (mức dưới) hay module thượng cấp (mức trên) của nó. Do đó phạm vi phát hiện lỗi liên quan chặt chẽ đến các module này.
* Người tiến hành kiểm thử đơn vị thường là người lập trình module đó hoặc lập trình viên cùng nhóm. Kỹ thuật kiểm thử đơn vị chủ yếu là kiểm thử hộp trắng, trong trường hợp này cần thiết có thể sử dụng thêm kiểm thử hộp đen.
* Mục đích: Đảm bảo thông tin được xử lý đúng và có đầu ra chính xác trong mối tương quan giữa dữ liệu nhập và chức năng của đơn vị.
* Người thực hiện: Do việc kiểm thử đơn vị đòi hỏi phải kiểm tra từng nhánh lệnh, nên đòi hỏi người kiểm thử có kiến thức về lập trình cũng như về thiết kế của hệ thống nên người thực hiện thường là lập trình viên.

#### Kiểm thử tích hợp

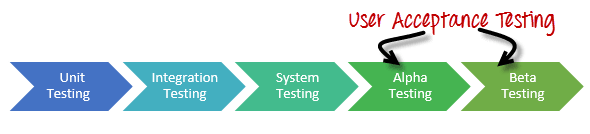
* Sau khi từng module được kiểm thử riêng rẽ, công việc quan trọng để tạo nên phần chức năng của hệ thống phần mềm hoàn chỉnh là tích hợp các module và kiểm thử như một thể thống nhất. Các đơn vị phần mềm được tích hợp dần thành các module chức năng, các hệ con và cuối cùng là hệ thống hoàn chỉnh. Kiểm thử này thường do các nhóm kiểm thử độc lập thực hiện. Phương pháp chính là hộp đen. Phương pháp hộp trắng được sử dụng bổ sung khi cần kiểm thử lại module.
* Mục đích:
* Phát hiện lỗi giao tiếp xảy ra giữa các đơn vị cũng như lỗi của bản thân từng đơn vị (nếu có).
* Tích hợp các đơn vị đơn lẻ thành các hệ thống nhỏ (subsystem) và cuối cùng là tích hợp các hệ thống nhỏ thành một hệ thống hoàn chỉnh (system) để chuẩn bị cho kiểm thử hệ thống
* Lưu ý:
* Kiểm thử tích hợp chỉ nên thực hiện trên từng đơn vị đã được kiểm tra cẩn thận trước đó bằng kiểm thử đơn vị, và tất cả các lỗi mức đơn vị đã được sửa chữa.
* Nên tích hợp dần từng đơn vị: Một đơn vị nên được tích hợp vào một nhóm các đơn vị khác đã được tích hợp và hoàn thành kiểm thử tích hợp trước đó vì khi đó chỉ cần kiểm tra giao tiếp giữa đơn vị mới được thêm vào với nhóm các đơn vị đã được tích hợp trước đó.
* Những lỗi cần được phát hiện khi kiểm thử tích hợp là lỗi dữ liệu qua giao diện, hiệu ứng của một module gây ra lỗi cho module khác, lỗi xung đột về bộ nhớ hay các thiết bị ngoại vi… Dưới đây là một số lỗi giao diện điển hình:
* Sử dụng sai giao diện: Tham số được truyền giữa các module có thể không tương thích về kiểu hay về thứ tự. Đây là một lỗi rất phổ biến.
* Hiểu lầm về giao diện: Module gọi có thể hiểu nhầm về chức năng của module được gọi, do đó dẫn đến việc các kết quả sai.
* Xung đột: Có nhiều kiểu xung đột có thể xảy ra, điển hình là trong các hệ thống thời gian thực hay các hệ thống cơ sở dữ liệu khi các module tương tác với thiết bị nhưng không đồng bộ với nhau.

#### Kiểm thử hệ thống

* Kiểm thử hệ thống bắt đầu khi tất cả các đơn vị của hệ thống được tích hợp thành công. Đây là công đoạn kiểm thử tốn nhiều công sức với thời gian hơn cả. Và đặc biệt, công đoạn này thường đòi hỏi được thực hiện bởi một nhóm nhân viên tách biệt với nhóm phát triển, có chuyên môn và kinh nghiệm được thử.
* Kiểm thử hệ thống gồm nhiều loại kiểm thử khác nhau, dưới đây là một số cách kiểm thử hệ thống :
* Kiểm thử chức năng.
* Kiểm thử hiệu năng.
* Kiểm thử an toàn thông tin.
* Mục đích: Kiểm tra xem hệ thống được làm ra có thỏa mãn yêu cầu hay không về nhiều khía cạnh: hoạt động, độ tin cậy, hiệu năng của hệ thống.
* Người thực hiện: Nhóm nhân viên kiểm thử.
* Lưu ý: Việc lập kế hoạch cho kiểm thử hệ thống nên bắt đầu từ giai đoạn bắt đầu dự án.

#### Kiểm thử chấp nhận

* Đây là một kiểm thử liên quan đến nhu cầu của người sử dụng, yêu cầu và quy trình kinh doanh được tiến hành để xác định một hệ thống có đáp ứng các tiêu chí chấp nhận và đáp ứng yêu cầu của khách hàng hay không.
* Kiểm thử chấp nhận kiểm thử các chức năng để kiểm tra hành vi của hệ thống bằng cách sử dụng dữ liệu thực tế. Nó cũng được gọi là thử nghiệm người dùng doanh nghiệp.
* Kiểm thử chấp nhận được thực hiện bởi người dùng cuối để kiểm tra hệ thống được xây dựng để phù hợp với yêu cầu kinh doanh của tổ chức.
* Trong kiểm thử này, tất cả các giao diện đã được kết hợp và hệ thống đã hoàn thành và đã được kiểm tra. Người dùng cuối cùng thực hiện các kiểm thử để kiểm tra khả năng sử dụng của hệ thống.
* Nhiều kỹ thuật kiểm thử chức năng sử dụng cho loại này là phân tích giá trị biên giới, phân vùng tương đương, bảng quyết định. Đây là loại kiểm tra tập trung chủ yếu vào các kiểm thử hợp lệ của hệ thống.
* Đối với những sản phẩm bán rộng rãi trên thị trường cho nhiều người sử dụng, quá trình kiểm thử trải qua hai bước gọi là *kiểm thử Alpha* và *kiểm thử Beta*.



Hình 1. 2 Kiểm thử Alpha và Beta

a) Kiểm thử alpha

* Người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm dưới sự hỗ trợ của nhân viên kiểm thử, nhân viên kiểm thử sẽ ghi nhận các lỗi hoặc phản hồi của khách hàng và báo lại với đơn vị phát triển phần mềm để lên kế hoạch sửa chữa.
* Tiêu chuẩn đầu vào
* Tài liệu đặc tả yêu cầu phần mềm hoặc đặc tả yêu cầu Business.
* Ca kiểm thử cho tất cả các yêu cầu.
* Đội ngũ kiểm thử có kiến thức tốt về các ứng dụng phần mềm.
* Thiết lập môi trường kiểm thử Lab.
* Chất lượng phần mềm đã sẵn sàng để thực hiện kiểm thử.
* Công cụ quản lý kiểm thử để tải ca kiểm thử và thực hiện kiểm thử (Kiểm thử tự động).
* Tiêu chuẩn đầu ra
* Tất cả các ca kiểm thử đã được thực hiện và đạt yêu cầu.
* Tất cả các lỗi nghiêm trọng phải được sửa và đóng.
* Bàn giao báo cáo về quá trình kiểm thử.
* Chắc chắn không có một chức năng mới nào được thêm vào.
* Kết thúc kiểm thử Alpha.
* Ưu điểm
* Cung cấp cách nhìn nhận tốt hơn về độ tin cậy của phần mềm trong giai đoạn đầu.
* Giúp mô phỏng các hành vi thực của người dùng và mô phỏng môi trường.
* Phát hiện nhiều lỗi nghiêm trọng.
* Có thể phát giện sớm các lỗi về thiết kế và chức năng.
* Nhược điểm
* Các chức năng không thể được kiểm tra vì phần mềm vẫn đang trong giai đoạn phát triển. Đôi khi lập trình viên và kiểm thử không hài lòng với kết quả của kiểm thử Alpha.

b) Kiểm thử beta

* Phần mềm sẽ được gửi tới cho người dùng để kiểm thử trong môi trường thực, lỗi hoặc phản hồi cũng sẽ gửi lại cho đơn vị phát triển phần mềm để lên kế hoạch sửa chữa.
* Tiêu chuẩn đầu vào
* Kết thúc phần tài liệu của quá trình kiểm thử Alpha.
* Sẵn sàng cho quá trình kiểm thử Beta.
* Tạo dựng môi trường, chuẩn bị cho sự ra mắt của sản phẩm.
* Sử dụng công cụ để nắm bắt được lỗi xảy ra trong quá trình thực hiện.
* Tiêu chuẩn đầu ra
* Tất cả các vấn đề, lỗi lớn hay nhỏ cũng phải được chắc chắn là đã được giải quyết.
* Chuẩn bị báo cáo từ những ý kiến phản hồi của người dùng.
* Bàn giao báo cáo của quá trình kiểm thử Beta.
* Ưu điểm
* Giảm rủi ro cho sản phẩm thông qua việc xác nhận của người dùng.
* Kiểm thử Beta cho phép kiểm tra trên hệ thống cơ sở hạ tầng trước khi được giao cho khách hàng.
* Cải thiện chất lượng sản phẩm thông qua phản hồi của người dùng.
* Chi phí tương đương với phương pháp thu thập dữ liệu.
* Tạo ra thiện cảm với người dùng và tăng sự thỏa mãn của người dùng.
* Nhược điểm
* Quản lý việc kiểm thử là một vấn đề. So với các phương pháp kiểm thử thường dùng trong môi trường có thể kiểm soát được khác, kiểm thử Beta được thực hiện trong thế giới thực nơi mà ta rất khó có thể kiểm soát được.
* Tìm ra đúng người dùng và duy trì sự tham gia của họ là một thách thức.

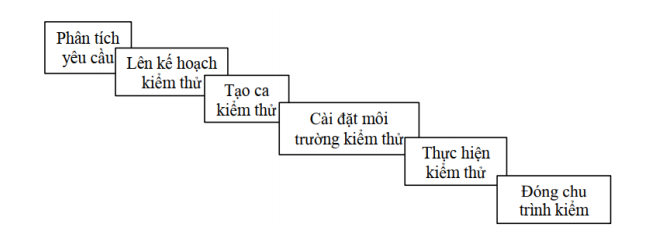
#### Kiểm thử hồi quy

* Kiểm thử hồi quy là hoạt động cần thiết để chỉ ra rằng việc thay đổi mã nguồn, nghiệp vụ, yêu cầu của khách hàng không gây ra những ảnh hưởng bất lợi đến hệ thống nói chung.
* Mục đích kiểm thử: dùng để kiểm tra các phần được sửa chữa và các phần liên quan đến các phần sửa chữa trong phần mềm, để đảm bảo rằng những sự thay đổi đó không gây ra lỗi trong những phần khác
* Cách thực hiện
* Tái sử dụng các kịch bản kiểm thử từ những phần kiểm thử trước để kiểm thử các module đã được sửa chữa.
* Sử dụng công cụ kiểm thử tự động: Tạo một số test script về chức năng.
* Xây dựng một chương trình phân tích cơ sở hạ tầng. Chúng ta dựng một cơ sở hạ tầng có thể mở rộng được để thực hiện và đánh giá chương trình phân tích. Dựa vào kết quả phân tích chúng ta xác định phạm vi cần kiểm thử hồi quy.
* Điều kiện hoàn thành: Toàn bộ các trường hợp kiểm thử đã chọn được thực hiện và đạt yêu cầu.

### Quy trình kiểm thử phần mềm

#### Quy trình kiểm thử phần mềm là gì?

* Quy trình kiểm thử phần mềm là chuỗi các hoạt động được tiến hành để thực hiện việc kiểm thử. Nó bao gồm hàng loạt các hoạt động được tiến hành theo phương pháp luận để giúp xác nhận sản phẩm phần mềm của bạn.



Hình 1. 3 Quy trình kiểm thử phần mềm

#### Các bước về kiểm thử

**a. Phân tích yêu cầu**

* Nhóm kiểm thử sẽ tương tác với các bên liên quan để hiểu rõ những yêu cầu cụ thể cần cho việc kiểm thử. Các yêu cầu có thể là chức năng (xác định phần mềm cần phải làm những gì) hoặc phi chức năng (hiệu năng, tính bảo mật hệ thống, màu sắc, ...)
* Hoạt động cụ thể:
* Xác định loại kiểm thử sẽ thực hiện
* Tổng hợp chi tiết về và mức độ tập trung thứ tự ưu tiên
* Chuẩn bị RTM (Requirement Traceability Matrix – một tài liệu dưới dạng bảng sử dụng để theo dõi các yêu cầu của khách hàng và kiểm tra xem các yêu cầu này đã được đáp ứng đầy đủ hay chưa)
* Xác định môi trường kiểm thử
* Phân tích khả năng sử dụng kiểm thử tự động
* Tài liệu sử dụng:
* RTM
* Báo cáo về khả năng sử dụng kiểm thử tự động (nếu cần)

**b. Lên kế hoạch kiểm thử**

* Còn được gọi bằng tên khác là lên chiến lược thử nghiệm. Ở giai đoạn này, trưởng nhóm kiểm thử sẽ dự toán chi phí cho dự án cũng như chuẩn bị kế hoạch kiểm thử.
* Hoạt động cụ thể:
* Lựa chọn công cụ kiểm thử (test tool).
* Lên kế hoạch về nhân sự và ấn định vai trò trách nhiệm cho từng người trong nhóm.
* Phổ biến cho mọi người trong nhóm kiểm thử về yêu cầu dự án.
* Tài liệu sử dụng:
* Bản kế hoạch kiểm thử.

**c. Tạo ca kiểm thử**

* Giai đoạn này cần phải tạo, xác minh, kiểm tra lại các ca kiểm thử. Dữ liệu kiểm thử cũng được tạo và xác định trong giai đoạn này.
* Hoạt động cụ thể:
* Tạo ca kiểm thử.
* Xác minh, kiểm tra lại các ca kiểm thử.
* Tạo dữ liệu kiểm thử.
* Tài liệu sử dụng:
* Ca kiểm thử.
* Dữ liệu kiểm thử.

**c. Cài đặt môi trường kiểm thử**

* Môi trường kiểm thử quyết định bởi các điều kiện phần cứng và phần mềm trong từng dự án. Thiết lập môi trường kiểm thử có thể thực hiện song song với giai đoạn sinh ca kiểm thử và là một tiêu chí quan trọng trong quá trình kiểm thử. Tuy nhiên, nhóm kiểm thử có thể không cần tham gia vào giai đoạn này nếu đã có các bên liên quan khác hỗ trợ, nhiệm vụ của nhóm kiểm thử chỉ là yêu cầu môi trường kiểm thử cần thiết.
* Hoạt động cụ thể:
* Hiểu được kiến trúc yêu cầu, thiết lập môi trường và chuẩn bị danh sách yêu cầu về phần cứng và phần mềm cho môi trường thử nghiệm.
* Thiết lập môi trường kiểm thử.

**d. Thực hiện kiểm thử**

* Nhóm kiểm thử thực hiện kiểm thử theo kế hoạch và danh sách ca kiểm thử đã chuẩn bị từ giai đoạn trước. Các lỗi phát hiện ở giai đoạn này sẽ được thông báo lại cho nhóm phát triển phần mềm để chỉnh sửa và thực hiện kiểm thử lại.
* Hoạt động cụ thể:
* Thực hiện kiểm thử theo kế hoạch.
* Làm tài liệu về kết quả kiểm thử, cập nhật lại các lỗi trong ca kiểm thử.
* Kiểm thử lại các lỗi đã được chỉnh sửa.
* Kiểm tra để đóng lỗi.
* Tài liệu sử dụng:
* Ca kiểm thử (cập nhật kết quả).
* Báo cáo lỗi.

**e. Đóng chu trình kiểm thử**

* Nhóm kiểm thử sẽ họp, thảo luận và phân tích những bài học rút ra sau quá trình kiểm thử, đưa ra chiến lược cho những lần kiểm thử kế tiếp hoặc chia sẻ kinh nghiệm cho những dự án tương tự.
* Hoạt động cụ thể:
* Đánh giá việc hoàn thành quy trình kiểm thử dựa vào thời gian, mức độ bao phủ, chi phí và chất lượng.
* Chuẩn bị dữ liệu dựa trên các tiêu chí trên.
* Chuẩn bị báo cáo kết thúc kiểm thử.
* Báo cáo chất lượng sản phẩm cho khách hàng.
* Phân tích kết quả kiểm thử để tìm ra sự phân bố lỗi theo loại và mức độ nghiêm trọng.
* Tài liệu sử dụng:
* Báo cáo kết thúc kiểm thử**.**

### Phân loại kiểm thử phần mềm

#### Kiểm thử chức năng

* Định nghĩa
* Kiểm thử chức năng là 1 loại kiểm thử phần mềm nơi mà hệ thống được kiểm thử so với yêu cầu chức năng, chi tiết kỹ thuật
* Đặc điểm
* Chức năng (tính năng) được kiểm thử bằng việc cho nó đầu vào và thu được đầu ra
* Kiểm thử chức năng đảm bảo thỏa mãn yêu cầu của ứng dụng.
* Loại kiểm thử này không quan tâm là từ đầu vào đó được xử lý như thế nào mà chỉ quan tâm đến kết quả đầu ra mà thôi.
* Kiểm thử hộp đen là kỹ thuật được sử dụng cho loại kiểm thử này bởi vì người kiểm thử sẽ không biết được logic hệ thống như thế nào.
* Loại kiểm thử này được sử dụng ở mức kiểm thử hệ thống và kiểm thử chấp nhận.
* Các loại kiểm thử chức năng: gồm 5 loại:
* Kiểm thử chức năng.
* Kiểm thử giao diện người dung.
* Kiểm thử tích hợp giữa dữ liệu và cơ sở dữ liệu.
* Kiểm thử vòng lặp công việc.
* Kiểm thử kiểm soát hệ thống.

#### Kiểm thử phi chức năng

* Định nghĩa
* Kiểm thử phi chức năng được định nghĩa là một loại kiểm thử Phần mềm để kiểm tra các khía cạnh phi chức năng (hiệu suất, khả năng sử dụng, độ tin cậy, v.v.) của ứng dụng phần mềm.
* Đặc điểm
* Tính sử dụng: khả năng học sử dụng ứng dụng nhanh hay không và tính lôi cuốn.
* Tính bảo trì: khả năng thay đổi, khả năng ổn định, khả năng kiểm tra, khả năng phân tích.
* Tính bền vững.
* Tính linh hoạt và tính hiệu quả.
* Các loại kiểm thử phi chức năng: có rất nhiều loại kiểm thử phi chức năng, nhưng bốn loại mà chúng ta thường dùng bao gồm:
* Kiểm thử hiệu năng.
* Kiểm thử tải.
* Kiểm thử tập trung.
* Kiểm thử với lượng dữ liệu lớn.

#### Kiểm thử cấu trúc (Structural testing)

* Kiểm thử cấu trúc thường được gọi là “hộp trắng” hoặc “hộp thủy tinh” bởi vì phương pháp này quan tâm đến việc tìm kiếm những gì đang xảy ra bên trong, kiểm tra dựa trên phân tích cấu trúc bên trong của thành phần hoặc hệ thống. Nó thường được sử dụng như một cách đo lường của kiểm thử, thông qua độ bao phủ của một tập hợp các yếu tố cấu trúc. Kiểm thử cấu trúc chủ yếu được áp dụng ở kiểm thử thành phần, kiểm thử tích hợp.
* Các mục tiêu chính của kiểm thử cấu trúc bao gồm:
* Nhận ra những điểm bất cập
* Test chức năng bổ sung
* Xác định những phần bị thiếu trong bộ kiểm thử
* Ưu điểm của kiểm thử cấu trúc:
  + Loại bỏ code chết
  + Có khả năng tìm ra lỗi ở giai đoạn đầu
  + Đảm bảo kiểm tra phần mềm kỹ lưỡng hơn
  + Tiết kiệm thời gian
* Bên cạnh đó, nhược điểm của kiểm thử cấu trúc:
* Kiểm tra kết cấu khá tốn kém
* Yêu cầu kiến thức về code
* Đòi hỏi kiến thức vững chắc về công cụ được sử dụng để test

#### Kiểm thử liên quan đến các thay đổi (Change related testing)

* Kiểm thử xác nhận (Confirmation testing)
* Khi kiểm thử gặp lỗi, Tester phải xác định nguyên nhân lỗi là do lỗi phần mềm. Sau khi Tester phát hiện lỗi và báo cho Developer để sửa thì phần mềm sau đó sẽ cập nhật phiên bản vá lỗi. Cuối cùng, Tester cần thực hiện kiểm tra thêm một lần nữa để xác định rằng lỗi thực sự đã được giải quyết.
* Khi thực hiện kiểm tra xác nhận, điều quan trọng nhất là phải đảm bảo rằng các trường hợp kiểm thử phải được thực hiện chính xác giống như lần đầu tiên, sử dụng cùng một đầu vào, dữ liệu và môi trường kiểm thử để đảm bảo rằng các lỗi đã được sửa. Tester cần phải biết rằng trong lần kiểm thử sau khi vá lỗi khả năng sinh ra lỗi khác trong phần mềm là điều hoàn toàn có thể xảy ra. Vì vậy kiểm thử chính xác ở phiên bản hiện tại của phần mềm là chưa đủ. Cách phát hiện các điểm ngoài ý muốn của việc kiểm lỗi là thực hiện kiểm thử hồi quy.
* Kiểm thử hồi quy (Regression testing)
* Tương tự như kiểm thử xác nhận thì kiểm thử hồi quy liên quan đến việc lặp lại các trường hợp kiểm thử đã được thực hiện trước đó. Kiểm thử hồi quy được thực hiện khi phần mềm thay đổi do sửa lỗi, chức năng mới.
* Mục đích của kiểm thử hồi quy để xác minh rằng các sửa đổi trong phần mềm hoặc môi trường không gây ra bất lợi ngoài ý muốn, ảnh hưởng hoặc làm hư các chức năng và hệ thống vẫn đáp ứng các yêu cầu của phần mềm. Tất cả các trường hợp trong quá trình kiểm thử hồi quy sẽ được thực hiện mỗi khi một phiên bản vá lỗi của phần mềm được phát hành, và điều này khiến chúng trở nên lý tưởng cho tự động hóa.

### Các mức độ nghiêm trọng của lỗi

* Mức độ nghiêm trọng (Severity) được định nghĩa là mức độ ảnh hưởng của lỗi đối với sự phát triển hoặc hoạt động của thành phần ứng dụng đang được thử nghiệm.
* Hiệu quả cao hơn đối với chức năng hệ thống sẽ dẫn đến việc gán mức độ nghiêm trọng cao hơn cho lỗi. Quality Assurance (QA) (or Tester) thường là người xác định mức độ nghiêm trọng của lỗi.
* Trong Kiểm thử phần mềm, mức độ nghiêm trọng của lỗi có thể được phân thành 4 cấp độ :
* Critical (Nghiêm trọng) : Lỗi ở cấp độ này có nghĩa là xử lý của hệ thống bị ngừng hoàn toàn, không có gì có thể tiến hành thêm bất kỳ xử lý gì khác.
* Major/High (Quan trọng) : Lỗi ở cấp độ này là rất nghiêm trọng và làm sập hệ thống. Tuy nhiên, một số phần của hệ thống vẫn hoạt động.
* Medium (Trung bình) : Lỗi ở cấp độ này gây ra một số hành vi không mong muốn, nhưng hệ thống vẫn hoạt động
* Low (Thấp) : Lỗi ở cấp độ này sẽ không gây ra bất kỳ sự cố lớn nào của hệ thống
* Các tip để xác định mức độ nghiêm trọng của lỗi
* Decide the frequency of occurrence (Quyết định dựa trên tần suất xuất hiện) : Trong một số trường hợp, nếu sự xuất hiện của một lỗi nhỏ thường xảy ra trong code, nó có thể trở nên nghiêm trọng hơn. Vì vậy, từ quan điểm của người dùng, nó là nghiêm trọng hơn mặc dù đó là một lỗi nhỏ.
* Isolate the defect (Cô lập khuyết điểm) **:** Cô lập khuyết điểm có thể giúp tìm ra mức độ nghiêm trọng của tác động.

### Ca kiểm thử

#### Ca kiểm thử là gì?

Ca kiểm thử là một khái niệm không thể thiếu trong kiểm thử phần mềm. Theo ISTQB “ca kiểm thử là một tập hợp các giá trị đầu vào, tiền điều kiện, các kết quả mong đợi và điều kiện kết thúc, được xây dựng cho mục đích hoặc điều kiện kiểm thử riêng biệt để kiểm tra tính đúng đắn của chương trình với yêu cầu của bản đặc tả yêu cầu phần mềm”. Hay nói cách khác, ca kiểm thử Mô tả dữ liệu bao gồm: đầu vào, hành động hoặc sự kiện và kết quả đầu ra mong đợi (expected results) để xác định liệu 1 ứng dụng, hệ thống phần mềm hoặc một trong các tính năng của nó có hoạt động đúng như mong muốn hay không.

#### Cấu trúc của một ca kiểm thử

* Cấu trúc của một ca kiểm thử thông thường bao gồm:
* Test case ID: Xác định số lượng trường hợp cần kiểm thử.
* Function (Chức năng): Các function có thể được chia nhỏ dựa theo chức năng của hệ thống nhằm giúp ca kiểm thử trở nên rõ ràng hơn.
* Pre-condition: Điều kiện đầu vào của ca kiểm thử, ví dụ như khi thực hiện kiểm thử form đăng nhập, pre-condition sẽ là form đăng nhập phải được hiển thị ra.
* Test Data: Dữ liệu đầu vào cần chuẩn bị trước khi kiểm thử.
* Test Steps: Mô tả chi tiết các bước thực hiện kiểm thử.
* Expected Results: Kết quả mong đợi sau khi thực hiện các bước kiểm thử.
* Actual result: Mô tả kết quả thực tế khi thực hiện kiểm thử trên môi trường của hệ thống. Actual result thường bao gồm ba giá trị: pass, fail và pending.
* Comments: Có thể chứa screen shot hoặc thông tin liên quan khi thực hiện ca kiểm thử.
* Ngoài ra có thể có thêm một số cột như: Designed by (người thực hiện kiểm thử), Execute Date (ngày thực hiện kiểm thử), v.v. Mức độ chi tiết của ca kiểm thử sẽ phụ thuộc vào từng dự án và quy mô của công ty sản xuất phần mềm.



Hình 1. 4 Ví dụ minh họa về kiểm thử đơn giản

* Một ca kiểm thử được cho là hiệu quả khi:
* Dựa vào ca kiểm thử có thể tìm thấy lỗi.
* Tìm được nhiều lỗi khó phát hiện.
* Chỉ ra được những điểm ban đầu mà khi thực hiện kiểm thử không tìm ra vấn đề.
* Ca kiểm thử cần có những bước thực hiện kiểm thử (Test steps) đơn giản, minh bạch, dễ hiểu.
* Các trường hợp thử nghiệm nên có giá trị, tóm tắt và ngắn.
* Các ca kiểm thử nên có sự liên kết: Mỗi ca kiểm thử cần được đánh số thứ tự (Test case ID) để đảm bảo ca kiểm thử đã bao phủ 100% bản đặc tả yêu cầu phần mềm
* Ca kiểm thử có thể bảo trì: Nên viết ca kiểm thử sao cho khi có thay đổi, chỉnh sửa thì các bên liên quan có thể dễ dàng nhận thấy được sự thay đổi đó.
* Ca kiểm thử có tính ứng dụng cao.
  + Tóm lại, ca kiểm thử được viết ra để kiểm tra hoạt động của các chức năng có đúng như mong muốn trong bản đặc tả yêu cầu phần mềm hay không. Khi viết ca kiểm thử nên cố gắng viết đơn giản, dễ hiểu nhưng phải đầy đủ các dữ liệu chuẩn cần có của một ca kiểm thử.

### Kiểm thử tự động

* Kiểm thử tự động là quá trình thực hiện một cách tự động các bước trong một ca kiểm thử. Nó sử dụng một công cụ kiểm thử tự động nào đó để rút ngắn thời gian kiểm thử. Kiểm thử tự động hỗ trợ các kiểm thử viên rất nhiều tùy vào công cụ và các nội dung kiểm thử có thể thực hiện bằng tay hay không. Đối với những nhiệm vụ kiểm tra khó mà thực hiện bằng tay hoặc yêu cầu chi phí về nhân công là quá lớn thì sử dụng công cụ hỗ trợ là điều hết sức cần thiết.
* Ưu và nhược điểm
* Ưu điểm
* Chính xác hơn: Nhờ độ ổn định cao, kiểm thử tự động có thể thực thi các test cases với độ chính xác cao hơn.
* Độ bao phủ cao: Khi sử dụng kiểm thử tự động, người kiểm thử có thể thực thi số lượng lớn các test cases trong một thời gian ngắn. Điều này giúp người kiểm thử tăng độ bao phủ trong giai đoạn regression test.
* Khả năng lặp: Công cụ kiểm thử tự động ra đời để giúp cho các tester không phải lặp đi lặp lại các thao tác một cách nhàm chán với độ tin cậy và ổn định cao.
* Khả năng tái sử dụng: Với một bộ kiểm thử tự động người ta có thể sử dụng nhiều phiên bản ứng dụng khác nhau.
* Tốc độ cao: Do là công cụ kiểm thử tự động bằng máy nên việc kiểm thử tự động nhanh hơn nhiều so với kiểm thử thủ công.
* Nhược điểm
* Chi phí khởi đầu cho kiểm thử tự động rất cao: Bất kỳ dự án lớn nào cũng cần một đội ngũ kiểm thử, việc thuê số lượng người thử nghiệm thủ công thực sự hiệu quả về chi phí. Chi phí tự động hóa là nhiều hơn cho thiết lập ban đầu như: chi phí mua công cụ tự động hóa, đào tạo và bảo trì các kịch bản kiểm thử chi phí rất cao.
* Kiểm thử tự động không phải là 100%: Kiểm thử tự động không thể là 100%. Kiểm thử tự động có thể được áp dụng cho việc kiểm thử hồi quy, hiệu suất (performance), tải (load/stress). Các lĩnh vực như giao diện người dùng, tài liệu, cài đặt, khả năng tương thích và phục hồi thì phải được thực hiện thủ công.
* Không kiểm thử tự động với trường hợp UI không cố định: Nếu giao diện người dùng luôn thay đổi, chi phí liên quan đến bảo trì tập lệnh sẽ rất cao.
* Người kiểm thử nên có kiến thức lập trình tốt: Người kiểm thử có kiến thức lập trình tốt có thể sử dụng công cụ tự động hóa tốt hơn và giải quyết vấn đề ứng dụng tốt hơn.

### Nguyên tắc quan trọng trong kiểm thử phần mềm

Có bảy nguyên tắc cơ bản cần chú ý khi kiểm thử phần mềm:

* Kiểm thử chỉ ra sự hiện diện của lỗi trong phần mềm: Kiểm thử có thể cho thấy sự có mặt của lỗi nhưng không thể chứng minh rằng phần mềm không có lỗi. Việc kiểm thử giảm xác suất lỗi chưa được tìm thấy vẫn còn trong phần mềm nhưng ngay cả khi không tìm thấy lỗi thì cũng không thể chứng minh được sản phẩm phần mềm được phát triển hoàn toàn chính xác.
* Kiểm thử tất cả các trường hợp là điều không thể: Nguyên tắc này nói rằng kiểm tra mọi thứ trong phần mềm một cách trọn vẹn là không thể. Việc kiểm thử không thể thực hiện được cho tất mọi trường hợp kiểm thử. Do vậy thay vì kiểm thử mọi khía cạnh, chúng ta phải tập trung vào kiểm thử những yếu tố quan trọng và nhiều rủi do.
* Kiểm thử càng sớm càng tốt: Các hoạt động kiểm thử nên bắt đầu trong giai đoạn đầu của vòng đời phát triển phần mềm, và nên tập trung và những mục tiêu kiểm thử nhất định. Các hoạt động kiểm thử từ giai đoạn đầu sẽ giúp phát hiện lỗi sớm hơn.
* Sự phân cụm của các lỗi: Thông thường, lỗi tập trung vào các module, thành phần chính của hệ thống. Nếu xác định được điều này chúng ta sẽ tập trung vào tìm kiếm lỗi quanh khu vực được xác định. Nó được coi là một trong những cách hiệu quả nhất để thực hiện kiểm tra hiệu quả.
* Nghịch lý thuốc trừ sâu: Nếu một phương pháp kiểm thử được lặp đi lặp lại nhiều lần, các trường hợp kiểm thử giống nhau sẽ không phát hiện được triệt để lỗi mới. Để khắc phục điều này, các trường hợp kiểm thử cần phải được xem xét và duyệt lại một cách đều đặn, việc kiểm thử mới cần phải được viết lại để thực thi những phần khác của phần mềm hay hệ thống để tìm ra những lỗi tiềm ẩn.
* Kiểm thử theo các ngữ cảnh: Việc kiểm thử phụ thuộc vào ngữ cảnh và chúng ta phải tiếp cận kiểm thử theo nhiều ngữ cảnh khác nhau.
* Sai lầm về việc không có lỗi: Việc không tìm thấy lỗi trên sản phẩm không đồng nghĩa với việc sản phẩm đã sẵn sàng để tung ra thị trường. Việc không tìm thấy lỗi cũng có thể là do bộ trường hợp kiểm thử được tạo ra chỉ nhằm kiểm tra những tính năng được làm đúng theo yêu cầu thay vì nhằm tìm kiếm lỗi mới.

## Các kĩ thuật xác định ca kiểm thử

### Phân vùng tương đương

* Phương pháp kiểm thử phần mềm này chia miền đầu vào của chương trình thành các lớp dữ liệu mà từ đó các trường hợp kiểm thử nên được thiết kế.
* Phân vùng lớp tương đương cho phép phân chia tập hợp các điều kiện kiểm tra thành một phân vùng nên gần như là giống nhau.
* Các giá trị đầu vào chia thành các vùng tương đương sau:
* Vùng tương đương hợp lệ: Tập hợp các giá trị kiểm thử thỏa mãn điều kiện của hệ thống.
* Vùng tương đương không hợp lệ: Tập hợp các giá trị kiểm thử Mô tả các trạng thái khác nhau của hệ thống: Sai, thiếu, không đúng,…
* Mục đích của việc này là để mỗi lớp tương đương chỉ cần test trên các phân tử đại diện. Từ đó nhằm giảm số lượng test case cần phải thiết kế.
* Để thiết kế test case bằng phân lớp tương đương cần tiến hành theo 2 bước:
* Xác định các lớp tương đương
* Xác định các ca kiểm thử
* Với nguyên tắc là một lớp các giá trị lớn hơn, một lớp các giá trị nhỏ hơn hoặc n lớp các giá trị hợp lệ.

### Phân tích giá trị biên

* Phân tích giá trị biên dựa trên việc kiểm thử tại các ranh giới giữa các phân vùng, Chúng ta sẽ tập trung vào các giá trị biên chứ không test toàn bộ dữ liệu. Thay vì chọn nhiều giá trị trong lớp đương tương để làm đại diện, phân tích giá trị biên yêu cầu chọn một hoặc vài giá trị là các cạnh của lớp tương đương để làm điều kiện test.
* Chúng ta thường thấy rằng một số lượng lớn lỗi xảy ra tại các ranh giới của các giá trị đầu vào được xác định thay vì các giá trị giữa, còn được gọi là các giá trị biên. Từ đó đưa ra lựa chọn các test cases thực hiện giá trị đầu vào các giá trị biên.
* Kỹ thuật thiết kế test cases này bổ sung cho phân vùng tương đương. Kỹ thuật kiểm thử phần mềm này dựa trên nguyên tắc: Nếu một hệ thống hoạt động tốt với các giá trị biên thì nó sẽ hoạt động tốt cho tất cả các giá trị nằm giữa hai giá trị biên.

### Bảng quyết định

* Bảng quyết định còn được gọi là bảng Nguyên nhân – Kết quả (Cause-Effect).
* Kỹ thuật kiểm thử phần mềm này được sử dụng cho các chức năng cần sự kết hợp của các yếu tố đầu vào các biến.
* Đầu tiên phải xác định đầu ra của các chức năng có phụ thuộc sự kết hợp của các đầu vào. Nếu có tập hợp kết hợp đầu vào lớn thì chia thành các tập hợp nhỏ hơn. Điều này sẽ hữu ích cho việc quản lý bảng quyết định.
* Cần tạo một bảng và liệt kê tất cả các loại kết hợp đầu vào và đầu ra tương ứng đối với mọi chức năng. Điều này giúp xác định các điều kiện bị tester bỏ qua.
* Các bước để tạo bảng quyết định:
* Nhập đầu vào theo hàng
* Nhập tất cả các quy tắc trong cột
* Điền vào bảng với sự kết hợp của đầu vào
* Trong hàng cuối, ghi chú đầu ra để so với kết hợp đầu vào.

### Đoán lỗi

* Đoán lỗi là một kỹ thuật kiểm thử phần mềm dựa trên việc đoán lỗi có thể chiếm ưu thế trong code. Đây là một kỹ thuật dựa trên kinh nghiệm, trong đó nhà phân tích kiểm thử sử dụng kinh nghiệm của mình để đoán phần có vấn đề hoặc có lỗi của ứng dụng kiểm thử.
* Kỹ thuật xác định danh sách các lỗi có thể xảy ra hoặc các tình huống dễ xảy ra lỗi. Sau đó, người kiểm thử viết test cases để tìm kiếm những lỗi đó. Để thiết kế các test cases dựa trên kỹ thuật kiểm thử phần mềm này, nhà phân tích có thể sử dụng các kinh nghiệm trong quá khứ để xác định các điều kiện.
* Cách đoán lỗi:
* Tester nên sử dụng kinh nghiệm trước đây để kiểm thử các ứng dụng tương tự
* Hiểu biết về hệ thống đang kiểm thử
* Kiến thức về các lỗi thực hiện điển hình
* Nhớ những chức năng phức tạp trước đây
* Đánh giá lịch sử dữ liệu và kết quả kiểm thử

# 

# KIỂM THỬ ỨNG DỤNG TRÊN NỀN WEB

## Khái quát về kiểm thử ứng dụng trên nền web

### Khái quát

* Các ứng dụng Web đã được phát triển và trở thành một nền tảng kết nối thông tin thiết yếu trong nhiều doanh nghiệp. Các ứng dụng Web đóng vai trò quyết định của thương mại điện tử, trao đổi thông tin.
* Để có thể đạt được điều này các ứng dụng Web cần phải có hiệu năng cao, đáng tin cậy,… Việc đưa ra một ứng dụng web hoàn hảo cho những người đang và sẽ sử dụng ứng dụng đã trở thành một thử thách chính trong đảm bảo chất lượng.
* Kiểm thử là một trong những công việc quan trọng để đánh giá chất lượng của một sản phẩm và đương nhiên là các ứng dụng Web cũng không phải là ngoại lệ. Các phương pháp kiểm thử thông thường là các kỹ thuật tập trung vào đánh giá các chức năng yêu cầu của ứng dụng.
* Tuy nhiên, không thể tập trung được vào hết tất cả các chức năng yêu cầu của ứng dụng. Không thể nào tập trung được vào hết tất cả các chức năng yêu cầu. Bởi có rất nhiều chức năng quan trọng cho người sử dụng ứng dụng như: tính hiệu năng, tính dễ sử dụng, độ tin cậy và tính bảo mật cần được xem xét. Những yêu cầu và mong đợi của người sử dụng, những vấn đề về nền tảng và cấu hình, mô hình nghiệp vụ, sự phát triển và chi phí cho việc kiểm thử là những vấn đề thường hay gặp phải và thay đổi liên tục đổi xuyên suốt chu trình của một ứng dụng Web. Vì thế, cần thiết phải phát triển một chiến lược hiệu quả cho việc kiểm thử mà có thể bao quát được giới hạn tổng thể và rộng lớn của những yêu cầu, chức năng cho một ứng dụng Web qua đó có thể giúp cho việc cài đặt, hoàn thành ứng dụng cũng như tránh được các rủi ro có thể gặp.

### Các loại ứng dụng Web

* Ứng dụng Web tĩnh: Là loại ứng dụng Web hiển thị ít nội dung và không có tính linh hoạt. Ứng dụng Web tĩnh thường chỉ được xây dựng từ HTML, CSS và Javascript. Do không có cơ sở dữ liệu và công cụ điều khiển nội dung gián tiếp nên người quản trị không thể tuỳ ý thay đổi nội dung mà cần có kiến thức về HTML, CSS cơ bản để chỉnh sửa. Điểm cộng của loại website này là nội dung đơn giản, không mất nhiều thời gian, công sức để xây dựng do không phải xử lý những câu lệnh phức tạp. Tuy nhiên, do không có hệ thống hỗ trợ thay đổi nội dung nên việc cập nhật thông tin cho website gặp rất nhiều khó khăn, thậm chí phải bỏ ra chi phí lớn khi thay đổi nhiều lần. Một số ví dụ cho Web tĩnh là những trang giới thiệu công ty, tổ chức, sản phẩm, khoá học ít có nhu cầu cập nhật thông tin, v.v.
* Ứng dụng Web động: So với web tĩnh thì Web động phức tạp hơn về mặt kỹ thuật khi xây dựng. Web động sử dụng cơ sở dữ liệu để hiển thị nội dung cũng như cho phép người dùng tương tác được với nội dung đó. Web động được chia làm 2 phần là back-end (dành cho người quản trị Web thay đổi, cập nhật nội dung) và front-end (dành cho người dùng truy cập). Hiện nay có rất nhiều ngôn ngữ lập trình được sử dụng để xây dựng Web động như Java, PHP, ASP.NET, VB.NET, Ruby, v.v. Đối với Web động, việc cập nhật nội dung là rất đơn giản và dễ dàng. Không những thế, một số hệ thống lớn hiện nay còn cho phép người quản trị có thể thay đổi giao diện Web trên trang quản trị mà không cần phải can thiệp trực tiếp vào mã nguồn. Đó là những lý do khiến cho Web động được sử dụng phổ biến hơn Web tĩnh.

### Đặc tả về chất lượng của một ứng dụng trên nền Web

* Yêu cầu về chức năng: Sự hiện diện của các chức năng đáp ứng những yêu cầu được xác định. Các yêu cầu cần có nữa là tính phù hợp, chính xác, khả năng tương tác, tuân thủ và bảo mật.
* Yêu cầu về độ tin cậy: Khả năng của một ứng dụng để duy trì sự hiệu quả của nó trong một điều kiện cụ thể và trong một khoảng thời gian xác định.
* Yêu cầu về khả năng sử dụng: Tính dễ sử dụng và hiệu quả của một ứng dụng. Vấn đề này có thể được thẩm định bởi một nhóm người dùng giả định.
* Yêu cầu về hiệu quả: Tỷ lệ giữa mức độ hiệu quả của một ứng dụng và các tài nguyên mà nó sử dụng trong các điều kiện cụ thể.Các yêu cầu về chất lượng đóng một vai trò thiết yếu khi thử nghiệm các ứng dụng Web. Mặc dù nhìn chung thì chúng tương tự như những yêu cầu về chất lượng cho các hệ thống phần mềm truyền thống. Tuy nhiên, chúng có thể có mức độ đòi hỏi cao hơn về chiều sâu.

## Công việc chính khi kiểm thử ứng dụng Web

### Kiểm thử chức năng

* Kiểm tra các trang web cho chức năng chính xác, định dạng, tập tin cookie, và xác nhận dữ liệu. Chức năng thử nghiệm là lý tưởng để thực hiện kiểm tra khói, kiểm tra hồi quy, và thử nghiệm hội nhập.
* Kiểm tra các liên kết:
* Liên kết bên trong một cấu trúc siêu văn bản mà điểm đến không tồn tại một nút (các trang web, hình ảnh...) gọi là liên kết hỏng thường xuyên xảy ra sai sót trong các ứng dụng web. Để kiểm tra tính chính xác của các trang liên kết (link kiểm tra), tất cả các liên kết được hệ thống theo sau bắt đầu trên một trang bắt đầu, và sau đó được nhóm trong một đồ thị liên kết (bản đồ trang web).
* Ngoài ra, khi vượt qua các liên kết, người ta thường có thể tìm thấy dữ liệu bổ sung để cung cấp chỉ dẫn tiềm năng lỗi. Ví dụ, độ sâu và bề rộng của các cơ cấu chuyển hướng, khoảng cách giữa hai trang liên quan, được đo bằng số lượng các liên kết hoặc lần tải của các trang.
* Kiểm tra các hình thức web trên trang:
* Kiểm tra các lĩnh vực logic xác nhận cho từng lĩnh vực
* Kiểm tra các giá trị mặc định cho từng lĩnh vực
* Kiểm tra xem các lĩnh vực mật khẩu không hiển thị nội dung mật khẩu
* Kiểm tra giá trị đầu vào không hợp lệ cho từng lĩnh vực
* Xác nhận đáp ứng với một hình thức gửi
* Thử nghiệm quản lý phiên làm việc và cookie
* Kiểm tra các ứng dụng đăng nhập trong phiên bằng cách cho phép và vô hiệu hóa các tập tin cookie
* Cookie thử nghiệm tiêu cực bằng cách sử dụng một tên miền không phối hợp
* Kiểm tra xem cookie phiên thiết lập lại giữa các phiên trình duyệt
* Kiểm tra bảo mật ứng dụng bằng cách xóa các tập tin cookie có chọn lọc trong khi kiểm tra hoạt động
* Xác nhận CSS tags:
* Xác định các tag CSS 404 trả lại hoặc lỗi tải khác CSS.
* Xác định trên HTML id, class, và các thuộc tính tên không phù hợp với bất kỳ thẻ CSS nào.
* Xác nhận thẻ JavaScript:
* Xác định các tag Script trả lại lỗi 404 hoặc lỗi tải khác.
* Xác định id, tên, trên ... thuộc tính không phù hợp với bất kỳ thẻ Script.
* Kiểm tra nội dung động (kiểm tra cơ sở dữ liệu):
* Kiểm tra dữ liệu thống nhất trong các hình thức web cơ sở dữ liệu theo định hướng
* Kiểm tra chức năng tạo, chỉnh sửa, xóa, cập nhật công việc
* Kiểm tra dữ liệu cung cấp dữ liệu chính xác
* Xác định kết nối cơ sở dữ liệu và các lỗi truy vấn

### Kiểm thử khả năng sử dụng

* Việc thiết kế và trình bày của một ứng dụng có ảnh hưởng lớn đến thành công người dùng của bạn sẽ có trong việc sử dụng các ứng dụng Web.
* Kiểm tra điều hướng của Website
* Tất cả các tùy chọn như UI/UX, menu, liên kết hoặc các button trên website phải hiển thị và có thể truy cập.
* Điều hướng trang web dễ dàng sử dụng .
* Nội dung hướng dẫn phải rõ ràng và phải đáp ứng được mục đích.
* Tất cả tùy chọn trên header, footer và các điều hướng trái / phải phải nhất quán trên mỗi trang.
* Kiểm tra nội dung:
* Không có lỗi chính tả hoặc ngữ pháp lỗi trong nội dung.
* Tích hợp Alt trong hình ảnh .
* Không có ảnh hỏng.
* Xác nhận tính hợp lệ tất cả giao diện người dùng.
* Thực hiện theo một số tiêu chuẩn về xây dựng nội dung trên trang web.
* Tất cả nội dung phải rõ ràng và dễ hiểu.
* Màu tối gây bất bình cho người sử dụng, vì vậy tránh sử dụng theme màu tối.
* Kích thước hình ảnh kích nên phù hợp.
* Anchor text phải hoạt động bình thường

### Kiểm thử sự tương thích

* Sự khác biệt trong các trình duyệt Web, môi trường hoạt động, và các thiết bị phần cứng ảnh hưởng đến các hoạt động chính xác của ứng dụng Web của bạn.
* Trình duyệt tương thích:
* Thử nghiệm ứng dụng web của bạn cho chức năng chính xác trên một số trình duyệt như Firefox, IE, Chrome, Opera, và Safari. Lý tưởng nhất là ứng dụng web của bạn xử lý sự khác biệt trình duyệt thanh lịch.
* Kiểm tra chức năng ứng dụng với một loạt các cài đặt cấu hình bảo mật trình duyệt.
* Kiểm tra chức năng ứng dụng với các tính năng trình duyệt bật -tắt (JavaScript, cookies).
* Kiểm tra dựng hình trình duyệt của giao diện người dùng ứng dụng của bạn.
* Kiểm tra các thiết lập bảo mật của trình duyệt cho tên miền chéo truy cập và hack.
* Kiểm tra chức năng ứng dụng nhất quán trên nhiều phiên bản của một trình duyệt.
* Môi trường hoạt động tương thích:
* Kiểm tra ứng dụng giao diện người dùng vẽ trên hệ thống cửa sổ hệ điều hành.
* Kiểm tra chức năng tích hợp máy tính để bàn, bao gồm kéo và thả tập tin và lựa chọn.
* Thử nghiệm ứng dụng web của bạn trên các hệ điều hành khác nhau, bao gồm cả Windows, Unix, Mac, Linux, và Solaris .
* Thiết bị di động tương thích:
* Bạn cũng nên kiểm tra khả năng tương thích website trên điện thoại di động, đôi khi có thể xảy ra sự cố tương thích trên các trình duyệt điện thoại di động.

### Kiểm thử hiệu suất

* Load Test:
* Tuyến tính khả năng mở rộng - nơi hiệu suất của một ứng dụng không thay đổi khi số lượng người dùng tăng lên. Kiểm tra tải trọng xác định một chỉ số: Khả năng mở rộng cho hiệu suất ứng dụng web của bạn
* Kiểm tra phản ứng máy chủ để dưới dạng trình duyệt gửi yêu cầu
* Xác định thay đổi hoạt động trong một khoảng thời gian
* thử nghiệm cho các chức năng mà ngừng làm việc ở các cấp độ cao hơn của người sử dụng tải
* Xác định các vấn đề về độ trễ mạng về chức năng ứng dụng Web
* Stress Test:
* Xác định cách thức ứng dụng đáp ứng theo mức độ tải
* Xác định các phần của ứng dụng web mà không theo mức độ tải
* Xác định các chức năng ứng dụng sau khi một vụ tai nạn hệ thống hoặc thành phần thất bại
* Xác định các hình thức và các liên kết hoạt động khác nhau theo mức độ tải

### Kiểm thử bảo mật

* Được thực hiện để đảm bảo rằng có bất kỳ rò rỉ thông tin nào về mã hoá dữ liệu hay không. Trong website thương mại điện tử, kiểm thử bảo mật đóng một vai trò rất quan trọng, nếu thông tin an toàn thì kiểm tra xem làm thế nào để lưu trữ các thông tin nhạy cảm như thẻ tín dụng, thanh toán hóa đơn,… Các hoạt động kiểm tra sẽ bao gồm:
* Kiểm tra truy cập trái phép vào các trang an toàn, nếu người dùng thay đổi từ “https” sang “http” thì thông báo thích hợp sẽ được hiển thị và ngược lại
* Kiểm tra việc truy cập các trang internal, nếu đăng nhập được yêu cầu thì người dùng nên được chuyển hướng đến trang đăng nhập hoặc thông báo thích hợp sẽ được hiển thị
* Các thông tin liên quan đến giao dịch, thông báo lỗi, cố gắng đăng nhập nên được ghi vào file log
* Kiểm tra các tệp tin có bị hạn chế tải xuống hay không
* Kiểm tra các thư mục web hoặc tập tin web có thể truy cập được trừ khi không được cấu hình để tải xuống
* Kiểm tra CAPTCHA đã được thêm vào và hoạt động bình thường cho đăng nhập để tự động ngăn chặn các đăng nhập hay chưa
* Kiểm tra việc cố truy cập thông tin bằng cách thay đổi tham số trong chuỗi truy vấn. Ví dụ: nếu bạn đang chỉnh sửa thông tin và trên URL bạn thấy UserID = 123, hãy thử thay đổi các giá trị tham số này và kiểm xem ứng dụng có cung cấp thông tin người dùng khác không, nên từ chối hiển thị cho trường hợp này để ngăn chặn việc xem thông tin người dùng khác
* Kiểm tra session hết hạn sau thời gian được xác định nếu người dùng không thao tác trên website
* Kiểm tra user/password không hợp lệ

## Một số công cụ hỗ trợ kiểm thử ứng dụng trên nền Web

### Công cụ kiểm thử hiệu năng

#### Apache Jmeter

* Khái niệm: Công cụ mã nguồn mở này được thiết kế để kiểm tra hiệu suất tự động, chủ yếu dành cho các ứng dụng web. Nó có thể kích thích một lượng lớn người dùng truy cập các dịch vụ web và phân tích hiệu suất của AUT. Jmeter cũng có thể được sử dụng để kiểm tra API chức năng.
* Tính năng nổi bật:
* Giao diện thân thiện và dễ sử dụng
* Thử nghiệm tạo kế hoạch với tính năng ghi âm
* Kiểm tra thực thi ở cả hai chế độ GUI và CLI
* Hỗ trợ nhiều máy chủ, ứng dụng và loại giao thức khác nhau
* Tích hợp với các công cụ CI / CD phổ biến
* Yêu cầu hệ thống: hoạt động theo Unix và Windows OS

#### Webload

* Công cụ để test performance và stress test cho các ứng dụng web. WebLOAD là công cụ được lựa chọn để test tải cho các trang web của các doanh nghiệp có số lượng người dùng lớn và yêu cầu kiểm tra phức tạp. Nó cho phép bạn thực hiện load test và stress test trên bất kỳ ứng dụng internet nào bằng cách generate từ cloud hoặc máy local. Điểm mạnh của WebLOAD là tính linh hoạt và dễ sử dụng - cho phép bạn nhanh chóng xác định các thử nghiệm bạn cần với các tính năng như ghi, phát lại dựa trên DOM, tương quan tự động và ngôn ngữ kịch bản JavaScript.
* Công cụ này cung cấp một bản phân tích rõ ràng về hiệu suất ứng dụng web của bạn, xác định chính xác các vấn đề và những tắc nghẽn có thể làm cản trở việc load trang web. WebLOAD hỗ trợ hàng trăm công nghệ - từ giao thức web đến các ứng dụng doanh nghiệp và có thể tích hợp với Jenkins, Selenium và nhiều công cụ khác để cho phép thử tải liên tục cho DevOps.
* Yêu cầu hệ thống : Windows, Linux

### Công cụ kiểm thử bảo mật

#### WireShark

* WireShark là một công cụ phân tích mạng trước đây được gọi là Ethereal. Nó chụp gói tin trong thời gian thực và hiển thị chúng ở định dạng có thể đọc được. Về cơ bản, nó là một bộ phân tích gói mạng – cung cấp các chi tiết nhỏ nhất về các giao thức mạng, giải mã, thông tin gói,… Nó là một mã nguồn mở và có thể được sử dụng trên Linux, Windows, OS X, Solaris, NetBSD, FreeBSD và nhiều các hệ thống khác.

#### Owasp

* OWASP là một tổ chức phi lợi nhuận trên toàn thế giới tập trung vào việc cải thiện tính bảo mật của phần mềm. Dự án có nhiều công cụ để kiểm tra các môi trường và giao thức phần mềm khác nhau. Các công cụ hàng đầu của dự án bao gồm:
* Zed Attack Proxy (ZAP – một công cụ tự động tìm các lỗi bảo mật trong khi bạn đang phát triển và thử nghiệm ứng dụng)
* Kiểm tra phụ thuộc OWASP (quét các phụ thuộc của dự án và kiểm tra các lỗ hổng đã biết)
* Dự án môi trường thử nghiệm web OWASP (bộ sưu tập các công cụ và tài liệu bảo mật)

#### W3af

* W3af là một framework giúp kiểm tra và xác định các lỗ hổng trong các ứng dụng web. Công cụ này đi kèm với một số plugin hữu ích để quét một trang web với hơn 200 lỗ hổng công khai. Các plugin hiện có sẵn bao gồm kiểm tra, xác thực, bruteforce, thu thập thông tin, trốn, grep và cơ sở hạ tầng. Mỗi plugin có một bộ mục tiêu quét khác nhau.

### Công cụ kiểm thử chức năng

#### Selenium

* Khái niệm: Selenium là một công cụ kiểm tra phần mềm được sử dụng để kiểm tra hồi quy( Regression Testing). Đây là một công cụ kiểm tra mã nguồn mở cung cấp chức năng phát lại và thu âm để kiểm tra hồi quy. Các Selenium IDE chỉ hỗ trợ trình duyệt web Mozilla Firefox.
* Đặc điểm của Selenium:
* Cung cấp các điều khoản để xuất khẩu ghi lại kịch bản trong các ngôn ngữ khác như Java, Ruby, RSpec, Python, C #, JUnit và TestNG
* Nó có thể thực hiện nhiều bộ kiểm thử cùng một lúc
* Xác định phần tử sử dụng id, tên, đường dẫn X, v.v.
* Lưu trữ các bộ kiểm thử như Ruby Script, HTML và bất kỳ định dạng nào khác Hỗ trợ tệp tin người dùng selenium-extensions.js
* Cho phép để chèn ý kiến ở giữa của kịch bản để hiểu rõ hơn nội dung và mục đích của kịch bản kiểm thử

#### TestComplete

* Khái niệm: TestComplete có thể tự động hóa kiểm tra chức năng UI cho các ứng dụng máy tính để bàn, thiết bị di động và web. Với hỗ trợ tích hợp cho hơn 500 điều khiển và khuôn khổ của bên thứ ba, TestComplete có thể xử lý và xác định các phần tử giao diện người dùng động trong hầu hết các công nghệ hiện có.
* Đặc điểm:
* Ngôn ngữ lập trình được hỗ trợ: JavaScript, Python, VBScript, JScript, Delphi, C ++, C #
* Các phương pháp linh hoạt để thiết kế thử nghiệm: Chế độ ghi và phát lại, thủ công và kịch bản với các từ khóa cài sẵn.
* Nhận dạng đối tượng với nhận dạng hình ảnh dựa trên thuộc tính và AI
* Thử nghiệm song song, trình duyệt chéo và thiết bị chéo
* Tích hợp với các khuôn khổ thử nghiệm khác, công cụ CI / CD và hệ sinh thái SmartBear

#### Protractor

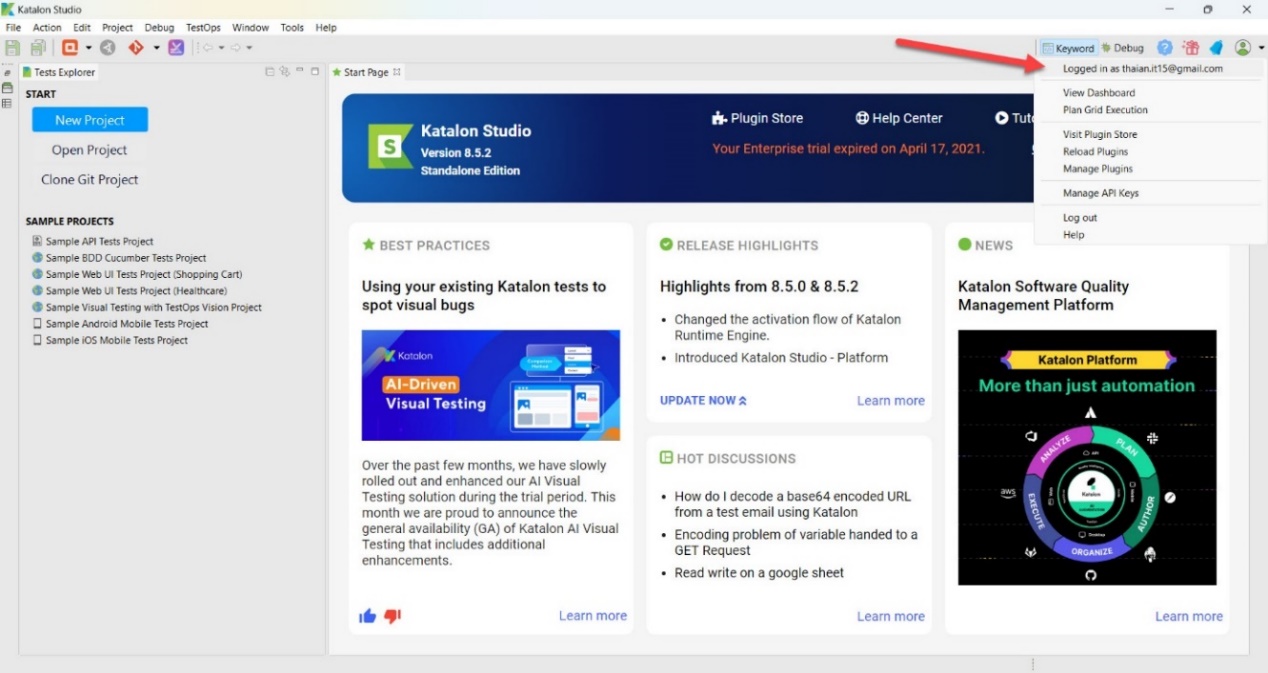
* Khái niệm: Khung mã nguồn mở này tự động kiểm tra đầu cuối chủ yếu cho các ứng dụng Angular và AngularJS. Nó hoạt động như một bộ tích hợp của Selenium, WebDriver, Jasmine, NodeJS và các công nghệ khác. Điều đó nói rằng, Thước đo góc cũng có thể hoạt động tốt để kiểm tra hồi quy với các ứng dụng không phải Angular.
* Đặc điểm:
* Kiểm tra thực thi trong một trình duyệt thực bằng cách sử dụng các sự kiện gốc
* Nhận dạng phần tử trong các ứng dụng Angular với các thuộc tính HTML mở rộng
* Hỗ trợ cho các khung phát triển theo hướng hành vi như Jasmine và Mocha
* Tự động chờ
* Thử nghiệm song song và nhiều trình duyệt với Selenium Grid

# KIỂM THỬ ỨNG DỤNG TRÊN NỀN WEB BẰNG CÔNG CỤ KATALON STUDIO

## Công cụ kiểm thử tự động Katalon Studio

### Giới thiệu chung

* Katalon Studio là giải pháp kiểm thử tự động được phát hành vào tháng 9 năm 2016 bởi Katalon LLC. Được xây dựng dựa trên các khung tự động hóa nguồn mở Selenium, Appium với giao diện IDE chuyên dụng để kiểm thử ứng dụng web, API, Mobile và PC. Katalon tool được nhiều khách hàng tin dùng trong lĩnh vực kiểm thử phần mềm của Gartner Peer Insights.
* Cung cấp cho người sử dụng có một giao diện có thể hoán đổi giúp tạo nên các trường hợp kiểm thử khác nhau. Với các chế độ xem thủ công dành cho các đối tượng không chuyên kỹ thuật và chế độ xem các tập lệnh hướng đến kiểm thử có kinh nghiệm.
* Tạo ra các kiểm thử tự động để đánh dấu các cú pháp và hoàn thành các mã thông minh. Katalon Studio theo mô hình đối tượng trang với các thành phần GUI trên ứng dụng mobile, web, PC để ghi lại các tiện ích và lưu trữ trong kho đối tượng.



Hình 3. 1 Giao diện công cụ katalon

* Ưu điểm
* Hỗ trợ viết test case nhanh chóng và đơn giản, có thể bật được chế độ manual test và scrip test.
* Test các ứng dụng trên Mobile, Desktop, Web, Api.
* Sử dụng được trên các nền tảng: Window, Linux, MacOS.
* Hỗ trợ Codeless: Record hay spy giúp tạo các test case mà không cần phải viết code.
* Sẵn các built-in keywords cho Api, PC, mobile, Web. Hỗ trợ sử dụng các dạng external file: CSV, database source, Excel.
* Các tính năng
* Triển khai đơn giản (Simple deployment): gói triển khai duy nhất, gắn kết chứa mọi thứ cần để triển khai một công cụ kiểm thử tự động mạnh mẽ.
* Cài đặt nhanh chóng và dễ dàng (Quick & easy set-up): không chỉ cung cấp cài đặt đơn giản, Katalon Studio còn giúp tester dễ dàng thiết lập môi trường. Người kiểm thử có thể chạy kịch bản kiểm thử đầu tiên của họ khá nhanh chóng bằng cách sử dụng các mẫu và tập lệnh kiểm thử dựng sẵn của nó, chẳng hạn như kho đối tượng và thư viện từ khóa.
* Kết quả nhanh hơn và tốt hơn (Faster & Better results): các mẫu dựng sẵn với hướng dẫn rõ ràng giúp người kiểm tra nhanh chóng xây dựng và chạy các kịch bản kiểm thử tự động. Có thể thực hiện từng bước một tốc độ và hiệu quả, từ thiết lập dự án, tạo kiểm thử, thực hiện, tạo báo cáo và bảo trì.
* Chế độ linh hoạt (Flexible modes): có thể sử dụng bản ghi và từ khóa để xây dựng bài kiểm thử tự động, trong khi có IDE đầy đủ để xây dựng tập lệnh nâng cao.
* Dễ sử dụng (Ease of use): ngay cả thủ công với kinh nghiệm lập trình tối thiểu cũng có thể khai thác lợi ích của nó một cách dễ dàng.
* Ứng dụng đa trình duyệt (Cross-browser application): Katalon Studio hỗ trợ nhiều nền tảng: Windows 32 và 64 (7, 8, và 10) và OS X 10.5+.

### Những ứng dụng liên quan

* Katalon Studio: là một giải pháp tự động hóa mạnh mẽ và miễn phí dành cho kiểm thử API, Web và ứng dụng di động. Nó tích hợp tất cả các thành phần cần thiết với các từ khóa tích hợp và các mẫu dự án vào một framework hoàn chỉnh. Katalon Studio rất dễ sử dụng cho người mới bắt đầu nhưng vẫn cung cấp khả năng nâng cao cho người dùng có kinh nghiệm. Giải pháp này được tin cậy bởi một cộng với hơn 150 nghìn người sử dụng từ hơn 150 quốc gia trên thế giới.
* Katalon Recorder: thay thế cho Selenium IDE trên Chrome và Firefox. Katalon Recorder có thể ghi lại, chạy, gỡ lỗi, quản lý test case, và xuất sang các định dạng C#, Java, Ruby, Python, Groovy hay Robot Framework.
* Katalon Analytics: Ứng dụng dựa trên đám mây cung cấp khả năng phân tích sâu về các báo cáo sau khi chạy test thông qua các loại biểu đồ, đồ thị và chỉ số.
* Katalium là một khung cung cấp thiết kế cho các dự án tự động kiểm thử dựa trên Selenium và TestNG. Khung được xây dựng để giúp người dùng vẫn cần làm việc với TestNG và Selenium nhanh chóng thiết lập các trường hợp kiểm thử.

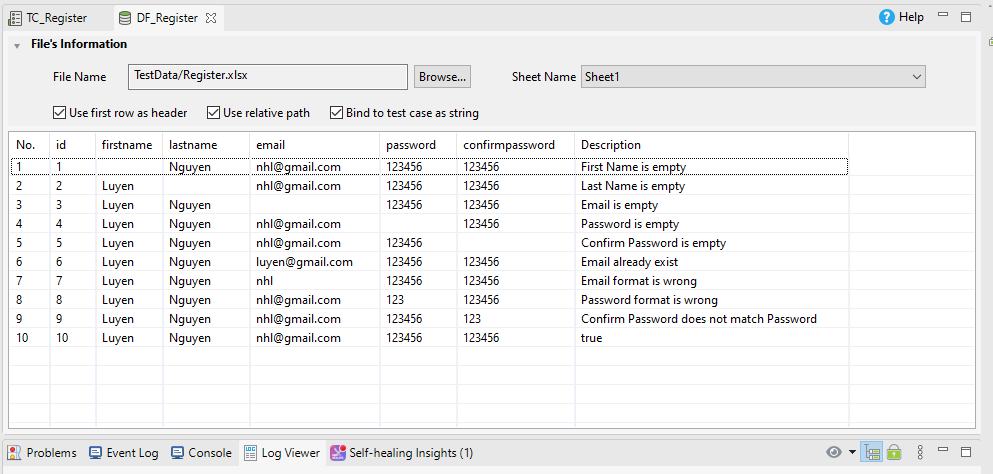
## Bài toán thực tế

* Trang web “Nopcommerce” được xây dựng nhằm cung cấp một hệ thống bán hàng trực tuyến.
* Các chức năng chính của hệ thống:
* Đăng nhập: khách hàng có thể đăng nhập vào hệ thống để thực hiện các chức năng như đặt hàng. Sau khi đặt hàng thành công, khách hàng có thể xem thông tin đặt hàng.
* Đăng kí: Người dùng khi chưa có tài khoản đăng nhập hệ thống, có thể thực hiện đăng ký tài khoản mới.
* Xem chi tiết sản phẩm
* Tìm kiếm sản phẩm
* Thêm, sửa, xóa địa chỉ
* Thêm, sửa, xóa sản phẩm trong giỏ hàng
* Đặt hàng

## Kiểm thử các chức năng trên trang web với Katalon Studio

### Chức năng “Đăng ký”

* Khi đăng ký tài khoản, người dùng được yêu cầu nhập vào: First Name, Last Name, Email, Password, Confirm Password. Khi người dùng nhập thiếu một trong các thông tin trên, nhập email đã được đăng kí hay nhập Confirm Password không khớp, hệ thống sẽ đưa ra thông báo cụ thể.
* Lấy dữ liệu từ data file (file excel)



Hình 3. 2 Data cho ca kiểm thử Đăng ký

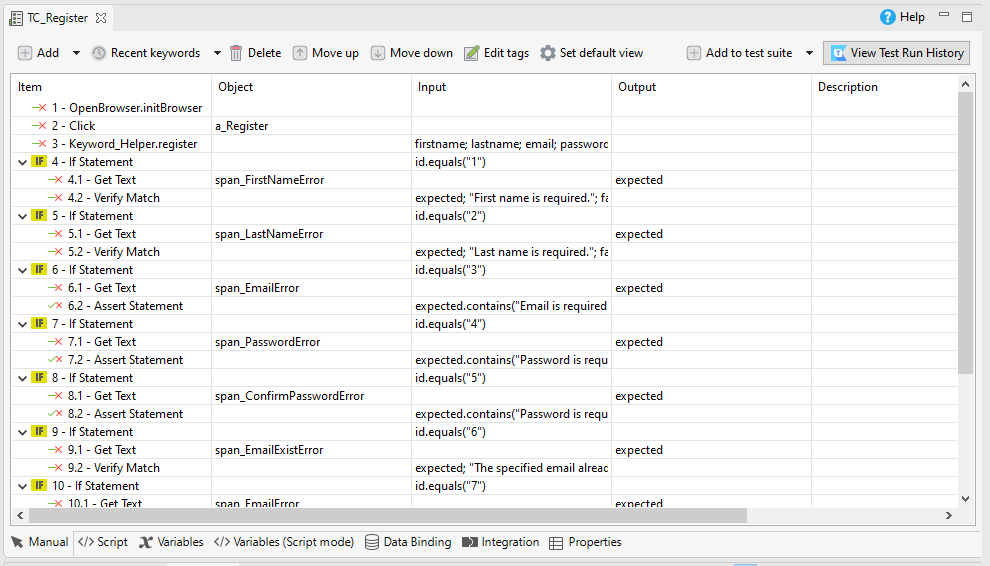
* Kịch bản kiểm thử





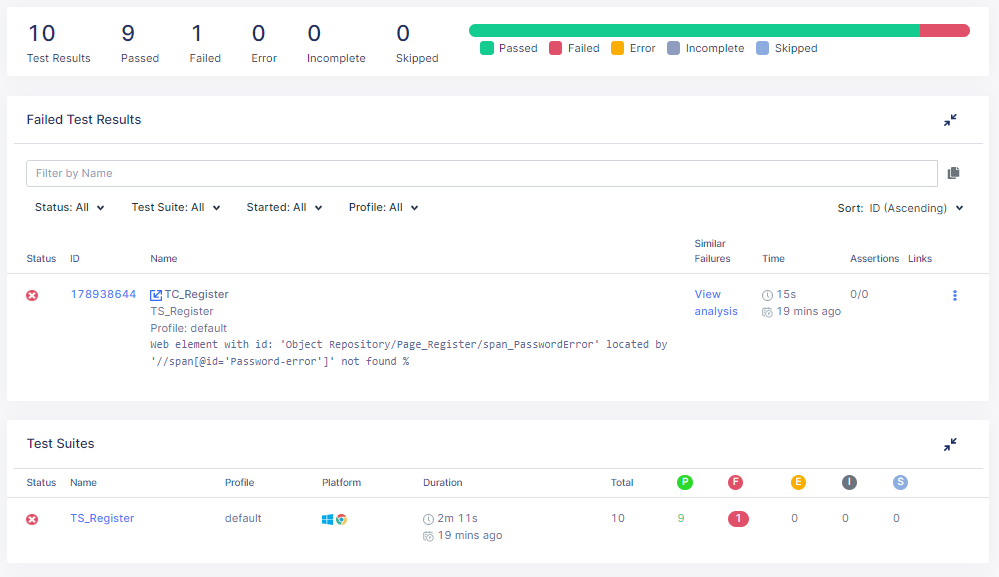
Hình 3. 3 Kịch bản kiểm thử cho test case Đăng ký

* Thực hiện test chức năng đăng ký



Hình 3. 4 Giao diện thực thi test case Đăng ký

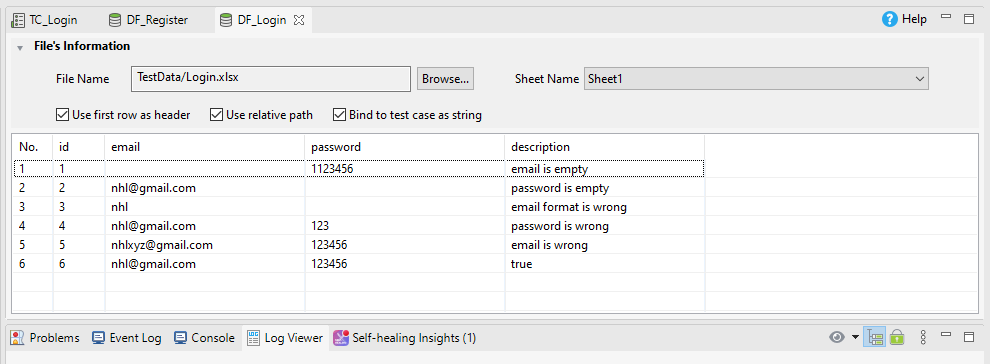
* Kết quả khi chạy các test case



Hình 3. 5 Kết quả thực hiện của test case Đăng ký

### Chức năng “Đăng nhập”

* Khi đăng nhập vào hệ thống, người dùng được yêu cầu nhập vào: Email, Password. Khi người dùng nhập thiếu một trong các thông tin trên hay nhập email, nhập Password không khớp, hệ thống sẽ đưa ra thông báo cụ thể.
* Lấy dữ liệu từ data file (file excel)



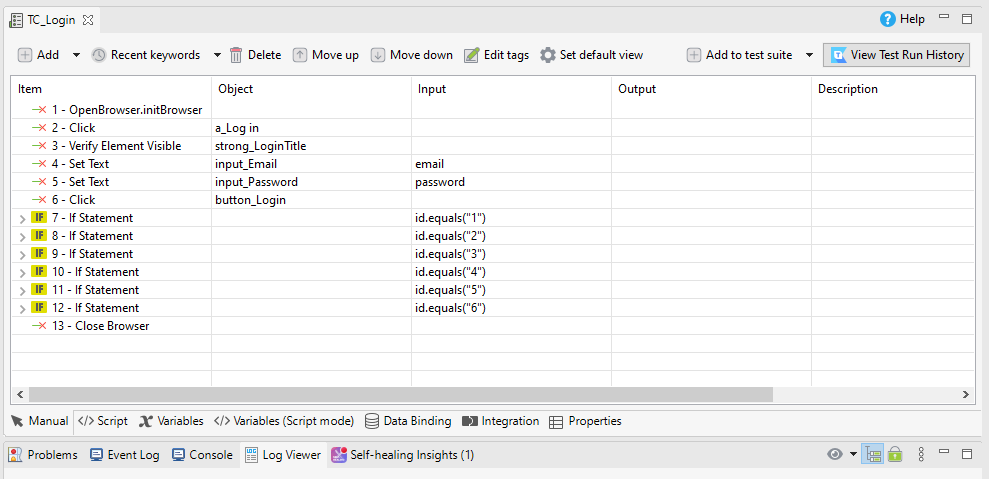
Hình 3. 6 Data cho ca kiểm thử Đăng nhập

* Kịch bản kiểm thử



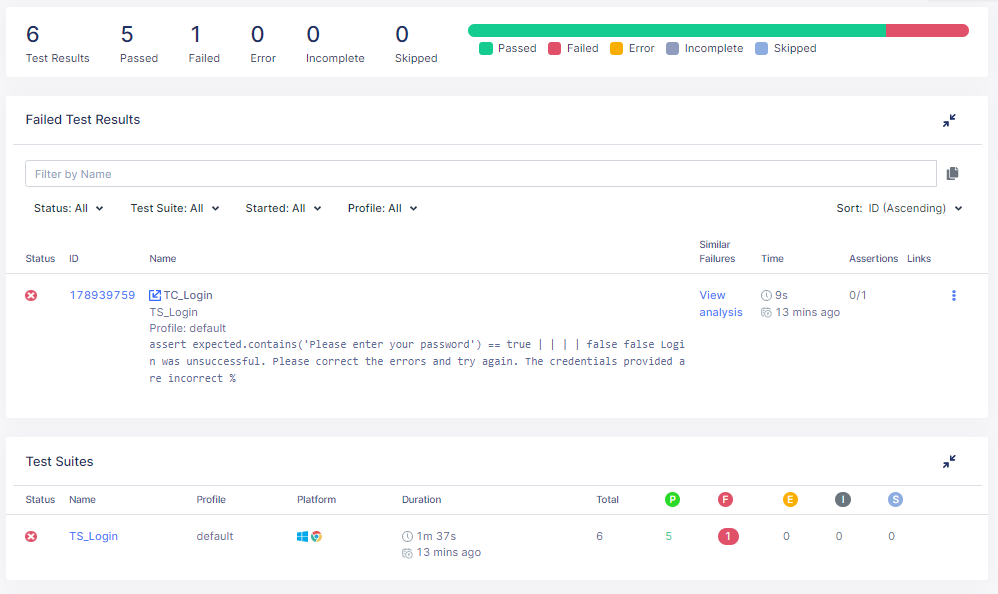
Hình 3. 7 Kịch bản kiểm thử cho test case Đăng nhập

* Thực hiện test chức năng



Hình 3. 8 Giao diện thực thi test case Đăng nhập

* Kết quả khi chạy các test case



Hình 3. 9 Kết quả thực hiện của test case Đăng nhập

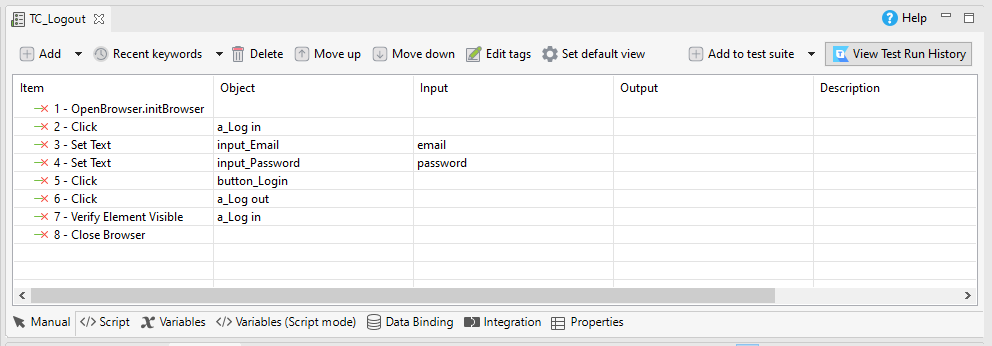
### Chức năng “Đăng xuất”

* Kịch bản kiểm thử



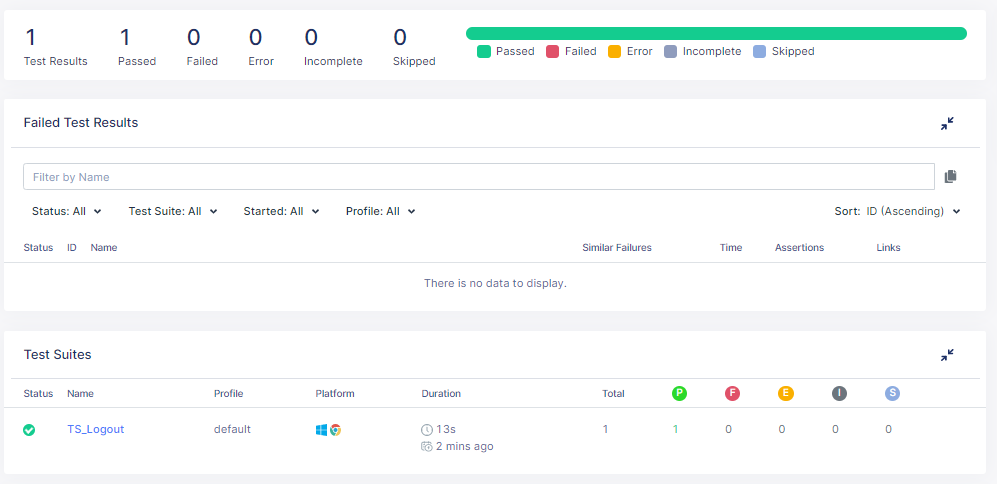
Hình 3. 10 Kịch bản kiểm thử cho test case Đăng xuất

* Thực hiện test chức năng



Hình 3. 11 Giao diện thực thi test case Đăng xuất

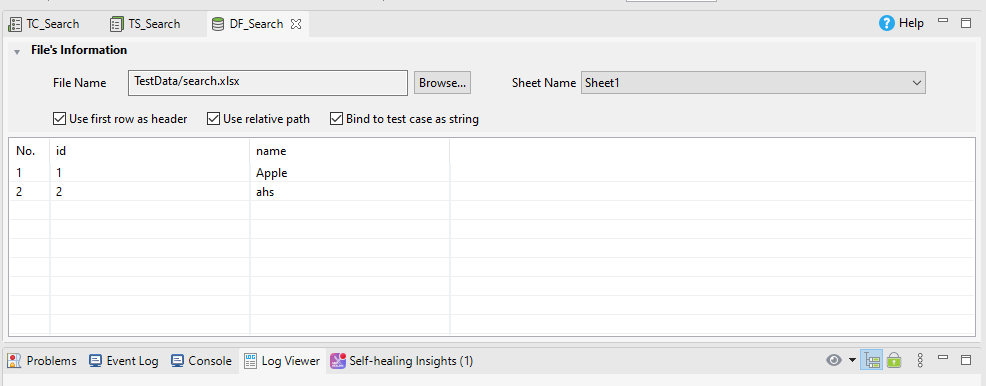
* Kết quả khi chạy các test case



Hình 3. 12 Kết quả thực hiện của test case Đăng xuất

### Chức năng “Tìm kiếm sản phẩm”

* Lấy dữ liệu từ data file (file excel)



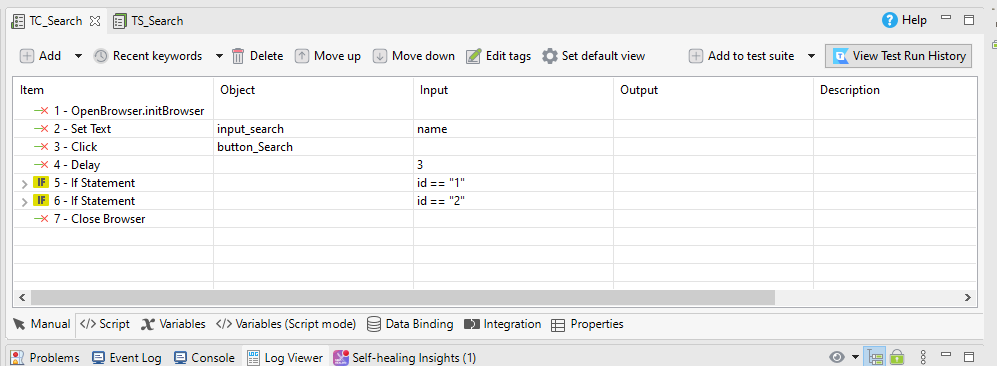
Hình 3. 13 Data cho ca kiểm thử Tìm kiếm

* Kịch bản kiểm thử



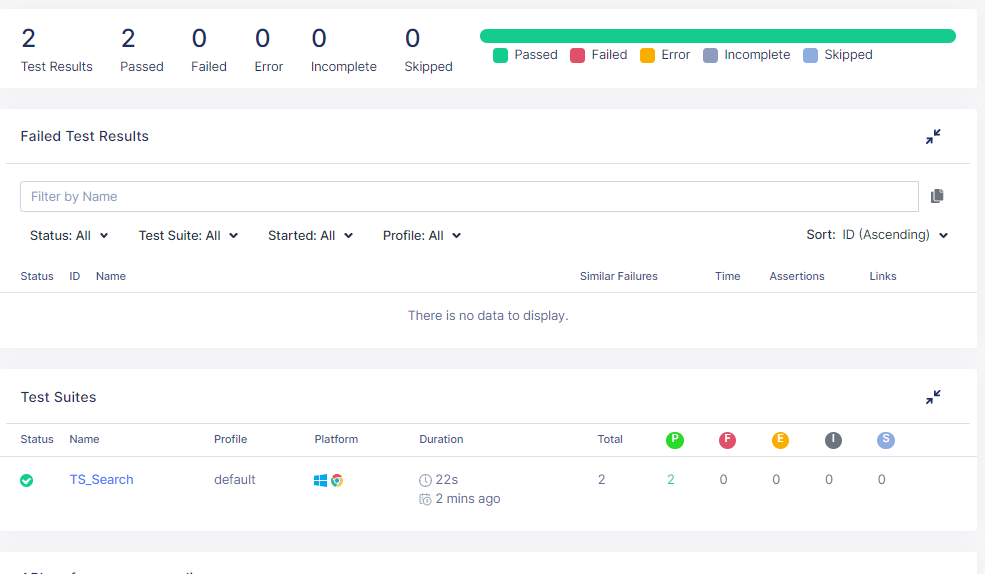
Hình 3. 14 Kịch bản kiểm thử cho test case Tìm kiếm

* Thực hiện test chức năng



Hình 3. 15 Giao diện thực thi test case Tìm kiếm sản phẩm

* Kết quả khi chạy các test case



Hình 3. 16 Kết quả thực hiện của test case Tìm kiếm

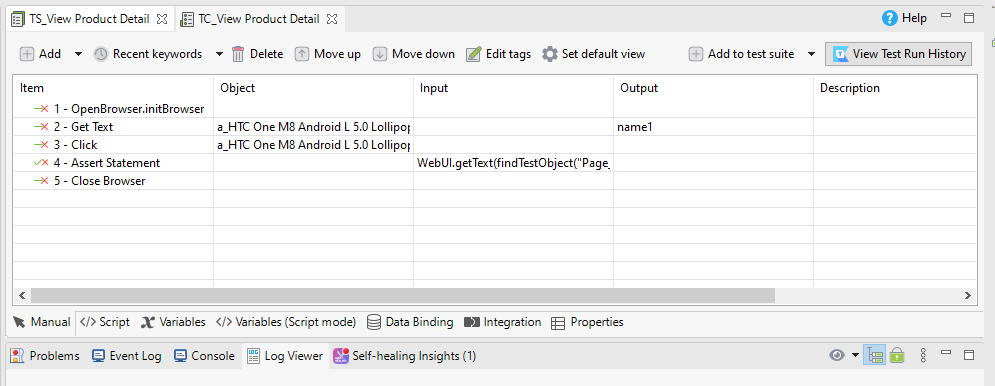
### Chức năng “Xem chi tiết sản phẩm”

* Kịch bản kiểm thử



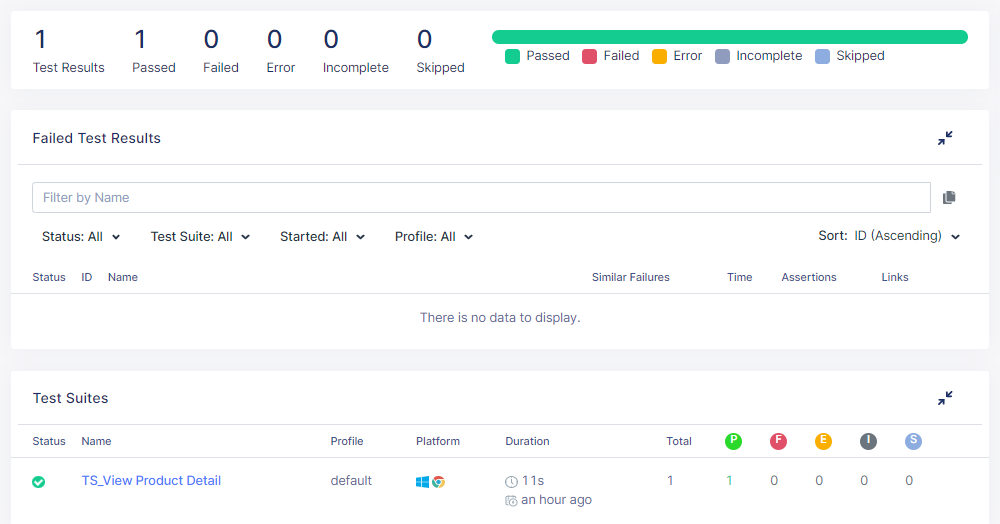
Hình 3. 17 Kịch bản kiểm thử cho test case Xem chi tiết sản phẩm

* Thực hiện test chức năng đăng ký



Hình 3. 18 Giao diện thực thi test case Xem chi tiết sản phẩm

* Kết quả khi chạy các test case



Hình 3. 19 Kết quả thực hiện của test case Xem chi tiết sản phẩm

### Chức năng “Địa chỉ”

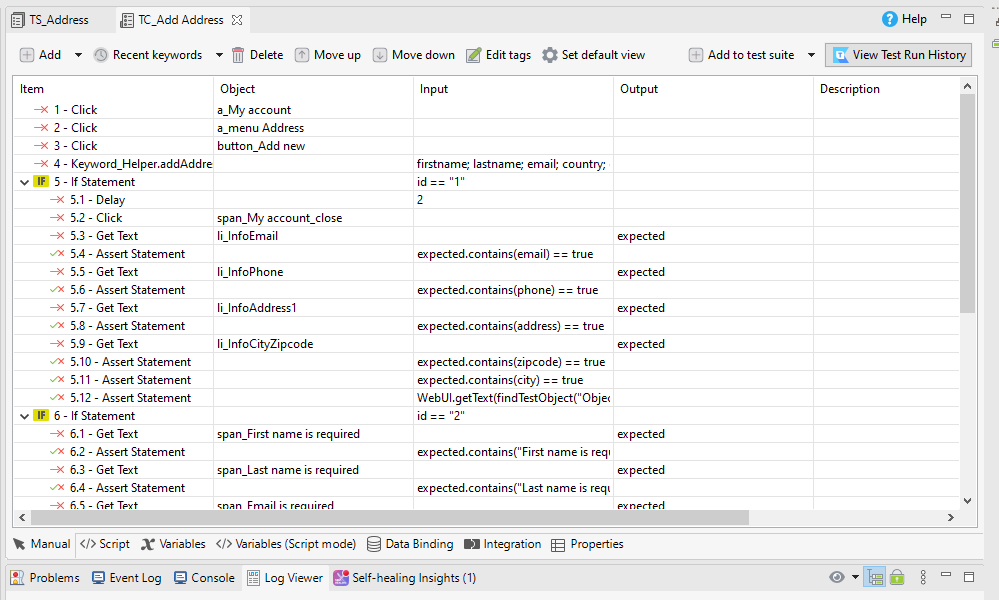
Khi thêm địa chỉ vào tài khoản, người dùng được yêu cầu nhập vào: firstname, lastname, email, country, city, address, zipcode, phone, description. Khi người dùng nhập thiếu hay sai một trong các thông tin trên, hệ thống sẽ đưa ra thông báo cụ thể.

* Kịch bản kiểm thử



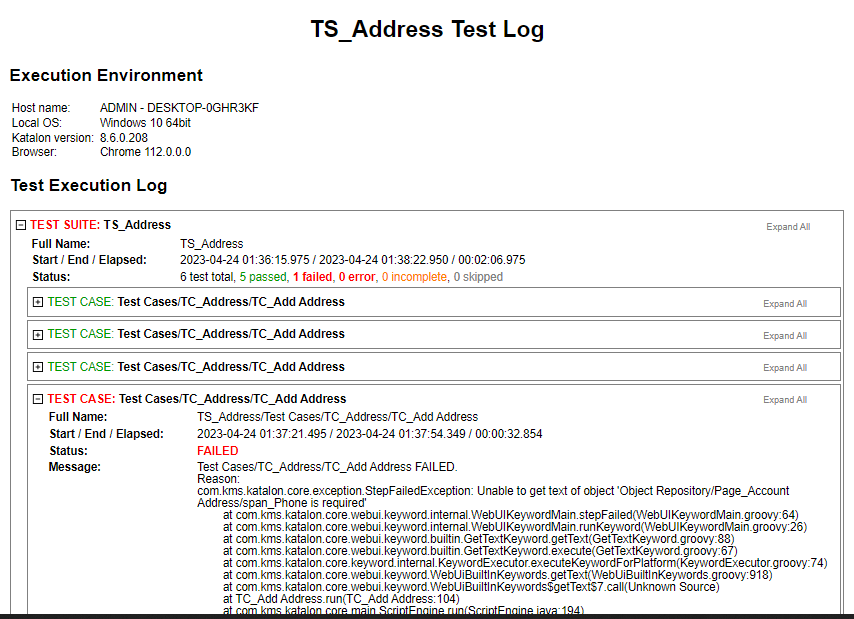
Hình 3. 20 Kịch bản kiểm thử cho test case Địa chỉ

* Thực hiện test chức năng



Hình 3. 21 Giao diện thực thi test case Địa chỉ

* Kết quả khi chạy các test case



Hình 3. 22 Kết quả thực hiện của test case Địa chỉ

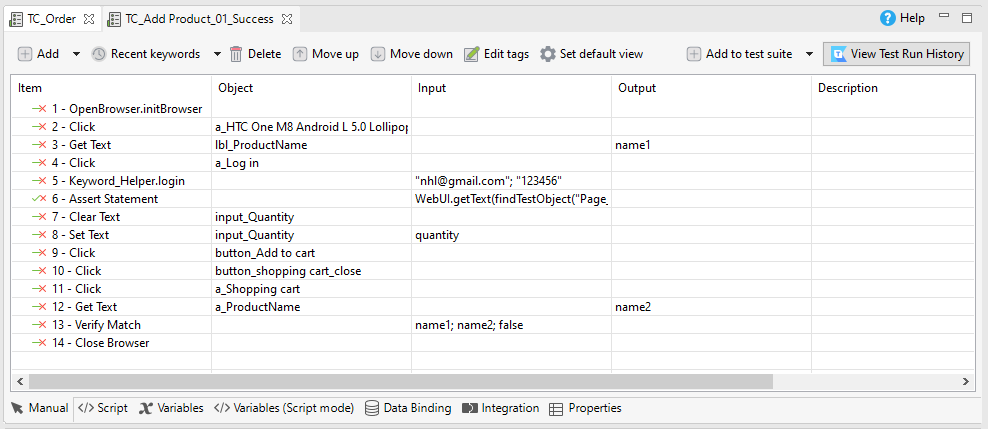
### Chức năng “Giỏ hàng”

* Kịch bản kiểm thử



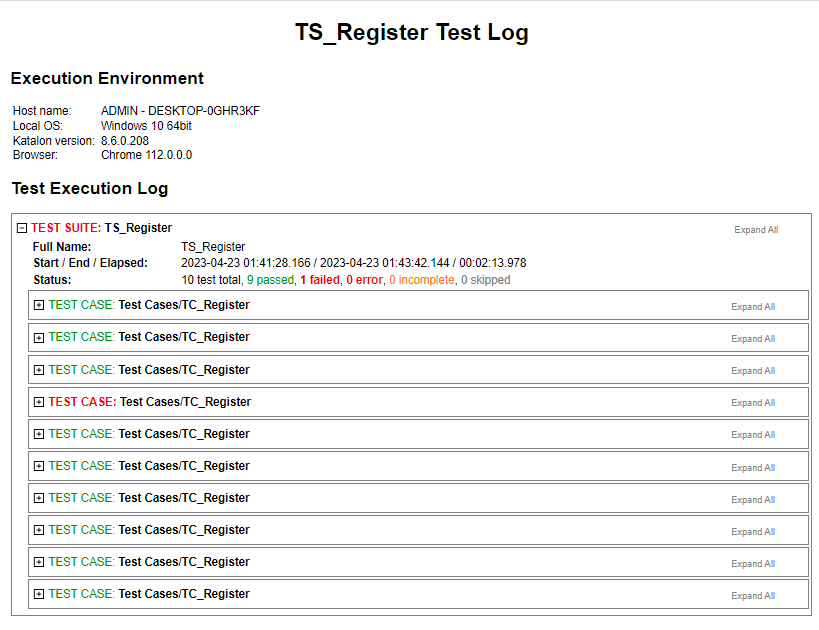
Hình 3. 23 Kịch bản kiểm thử cho test case Giỏ hàng

* Thực hiện test chức năng



Hình 3. 24 Giao diện thực thi test case Giỏ hàng

* Kết quả khi chạy các test case



Hình 3. 25 Kết quả thực hiện của test case Giỏ hàng

### Chức năng “Đặt hàng”

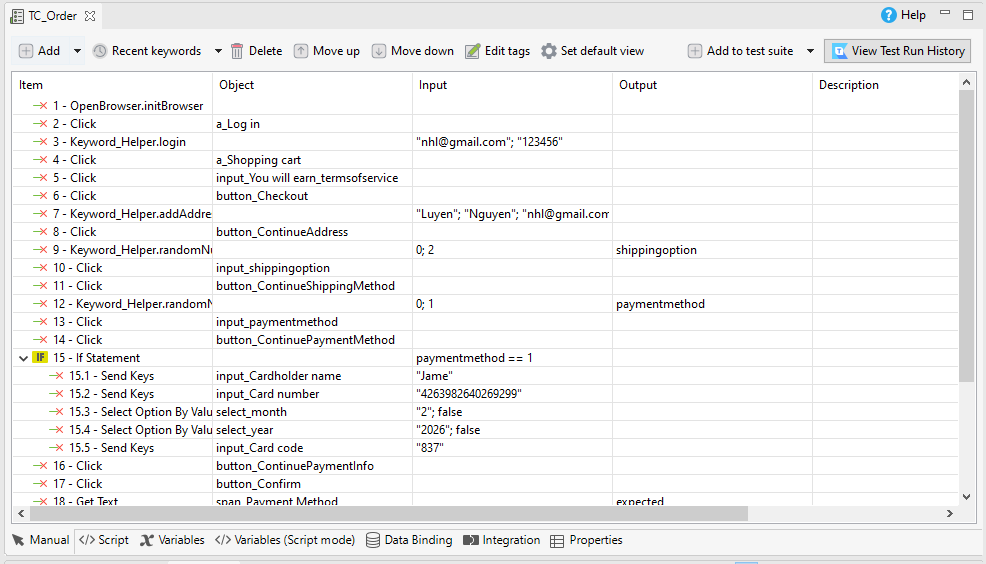
* Kịch bản kiểm thử





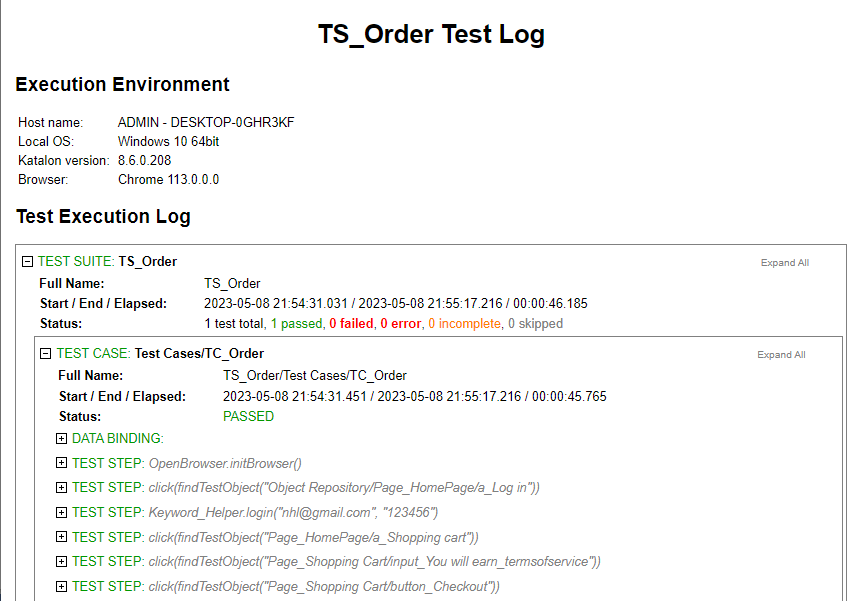
Hình 3. 26 Kịch bản kiểm thử cho test case Đặt hàng

* Thực hiện test chức năng



Hình 3. 27 Giao diện thực thi test case Đặt hàng

* Kết quả khi chạy các test case



Hình 3. 28 Kết quả thực hiện của test case Đặt hàng

## Tổng kết

Trạng thái các trường hợp thử nghiệm



Hình 3. 29 Trạng thái các trường hợp thử nghiệm

# KẾT LUẬN

Kiểm thử phần mềm là hoạt động quan trọng nhằm đảm bảo chất lượng phần mềm cũng như đáp ứng sự tin tưởng của khách hàng. Việc nghiên cứu lựa chọn các kĩ thuật và chiến lược kiểm thử phần mềm phù hợp giúp cho việc kiểm thử hiệu quả, giảm chi phí và thời gian. Trong thời gian qua, với sự giúp đỡ của cô Trần Phương Nhung, em đã đạt được những kết quả sau:

* Đã tìm hiểu, nghiên cứu và trình bày được tổng quan về lĩnh vực kiểm thử phần mềm nói chung và lĩnh vực web nói riêng
* Trình bày đầy đủ tổng quan về phần mềm, lỗi phần mềm, đặc tả yêu cầu phần mềm, chất lượng và độ tin cậy của phần mềm cũng như các vấn đề liên quan tới kiểm thử phần mềm.
* Chỉ ra được sự khác biệt của công việc kiểm thử ứng dụng trên nền Web với công việc kiểm thử trên các môi trường khác
* Đồ án có thể xem như tài liệu hướng dẫn sử dụng Katalon Studio một cách cơ bản nhất bằng tiếng Việt để tham khảo.

Do trình độ, khả năng và thời gian còn hạn chế nên báo cáo của bản thân còn tồn tại một số điểm hạn chế

* Trong khuôn khổ báo cáo chưa thể trình bày đầy đủ các tính năng, và nghiên cứu các kỹ thuật nâng cao được sử dụng trong Katalon Studio như test API
* Cần nghiên cứu sâu hơn về Katalon Studio trong lĩnh vực an toàn thông tin
* Kết hợp công cụ kiểm thử tự động Katalon Studio với công cụ kiểm thử bảo mật Burp Suite để có thể cải tiến được quy trình kiểm thử. Đảm bảo ứng dụng web có thể được kiểm tra về chức năng cũng như về bảo mật

Em xin trân trọng cảm ơn sự quan tâm giúp đỡ của các thầy, cô giáo giảng viên đã giúp đỡ tôi trong quá trình nghiên cứu và thực hiện khóa luận này.

*Em xin chân thành cảm ơn!*

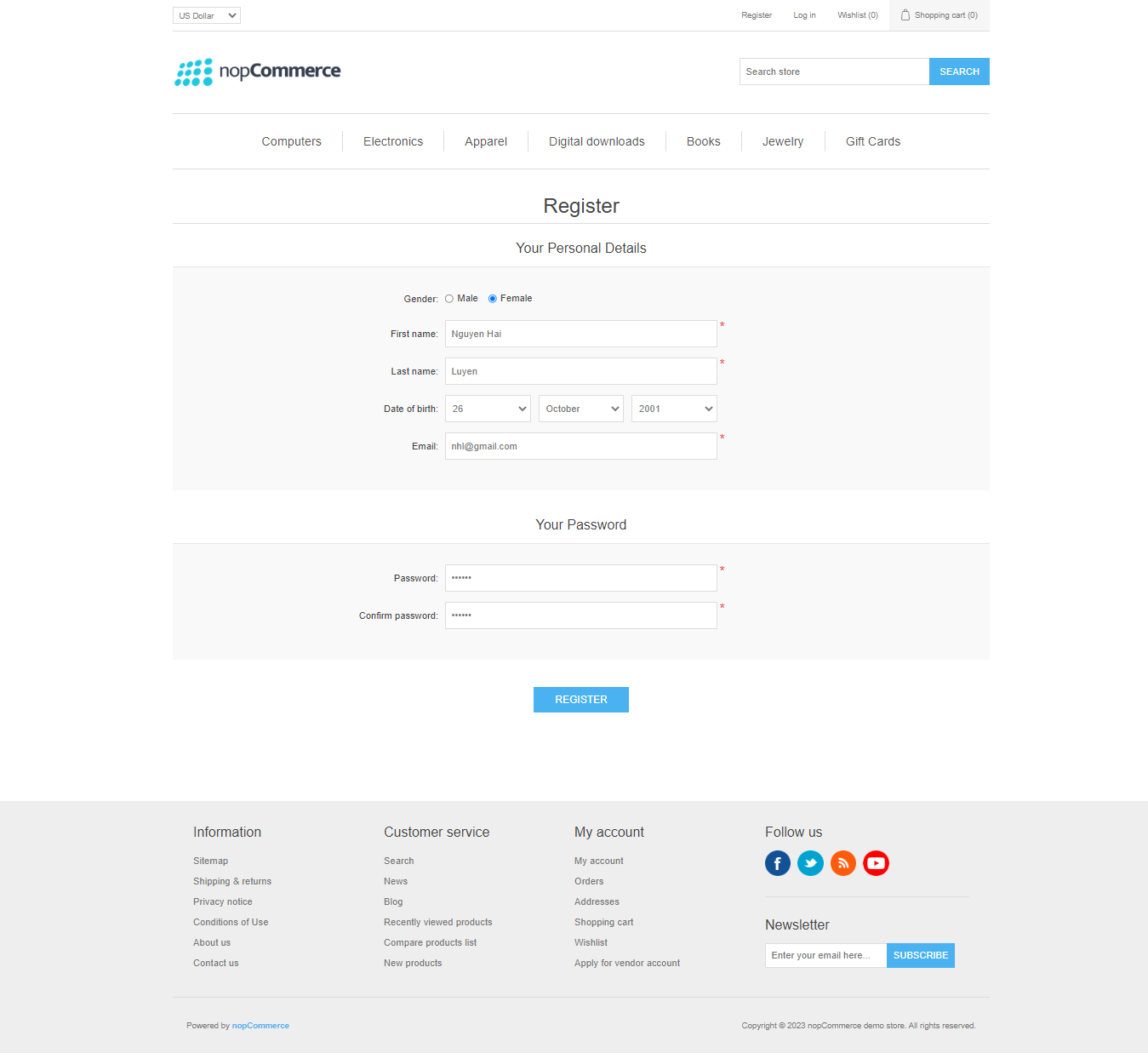
# TÀI LIỆU THAM KHẢO

1. https://www.katalon.com/
2. <https://www.youtube.com/playlist?list=PLhW3qG5bs-L_D4ZePNNjvmIULuu6mBHbu>
3. https://www.guru99.com/software-testing.html

# TÀI LIỆU HƯỚNG DẪN SỬ DỤNG

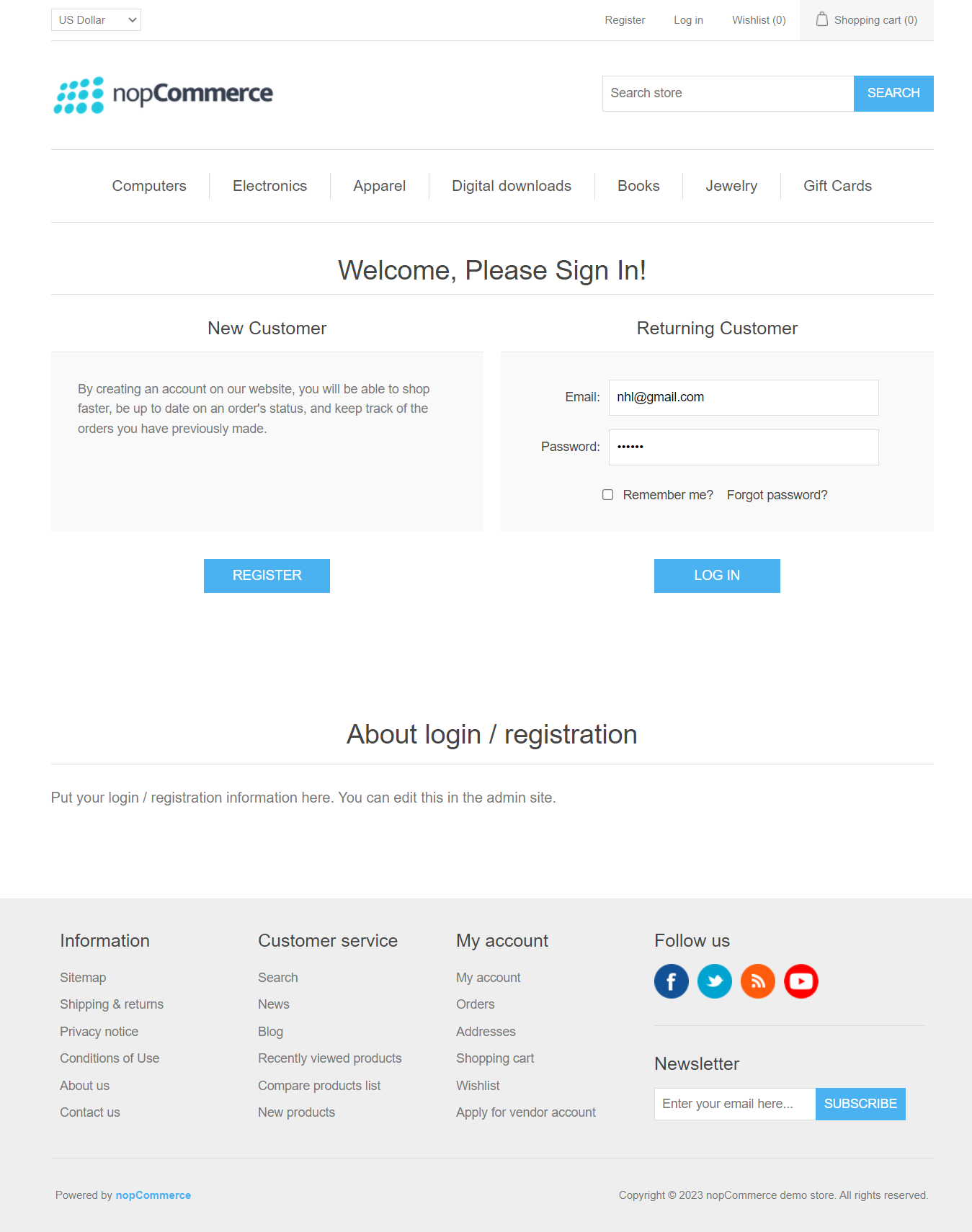
1. Đăng ký

* Trên thanh điều hướng, click vào nút “Register”
* Điền đầy đủ các thông tin: First Name, Last Name, Email, Password, Confirm Password và click vào button 



1. Đăng nhập

* Trên thanh điều hướng, click vào nút “Log in”
* Điền đầy đủ các thông tin: Email, Password và click vào button 

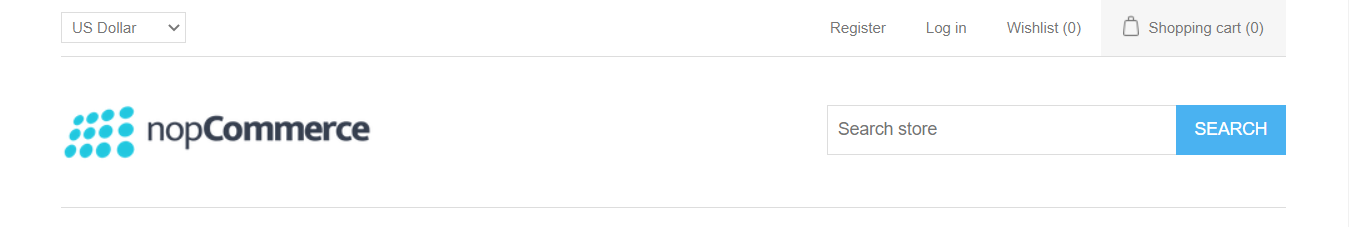


1. Đăng xuất

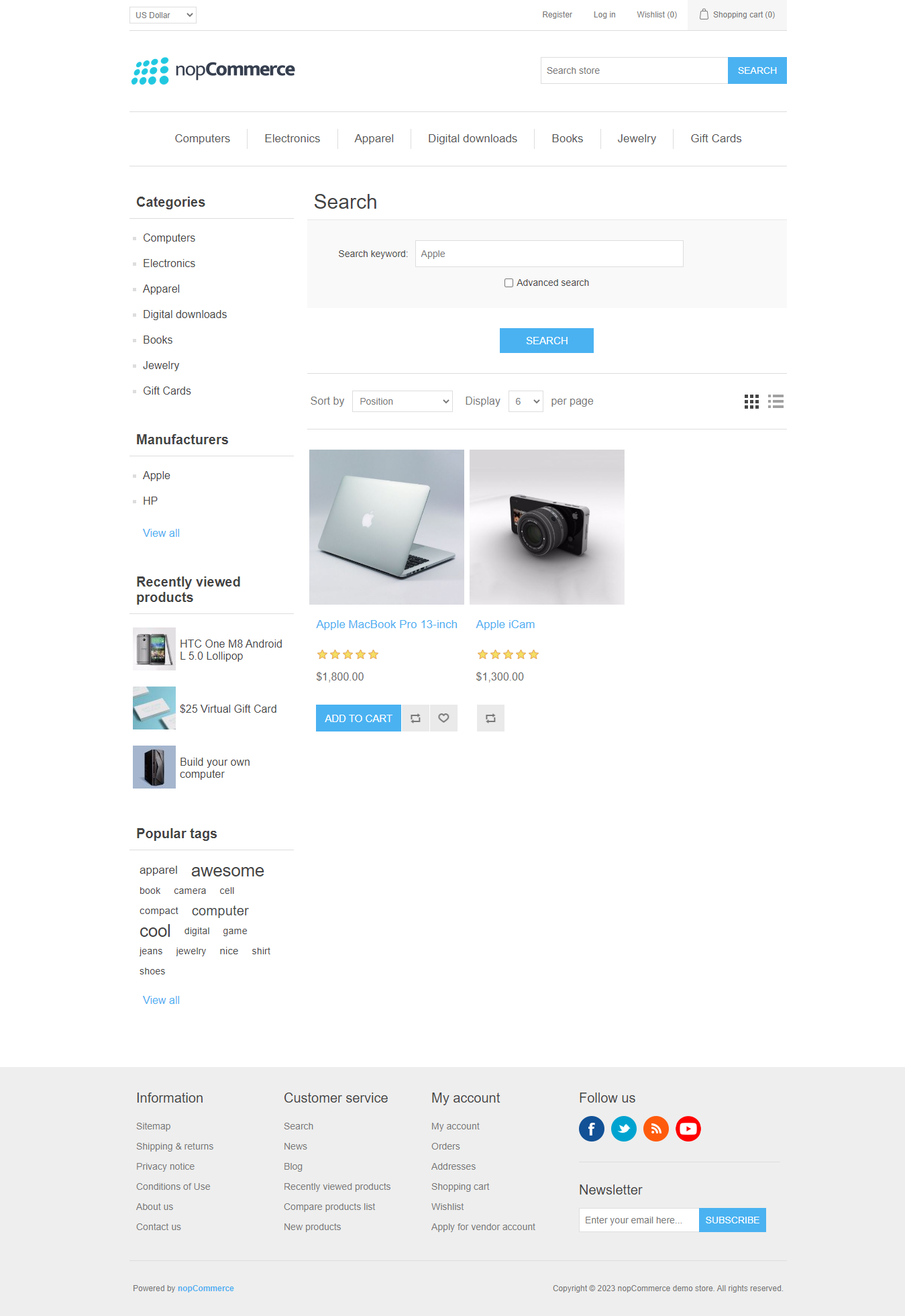
* Trên thanh điều hướng, click vào nút “Log out”

1. Tìm kiếm sản phẩm

* Gõ từ khóa cần tìm vào hộp tìm kiếm

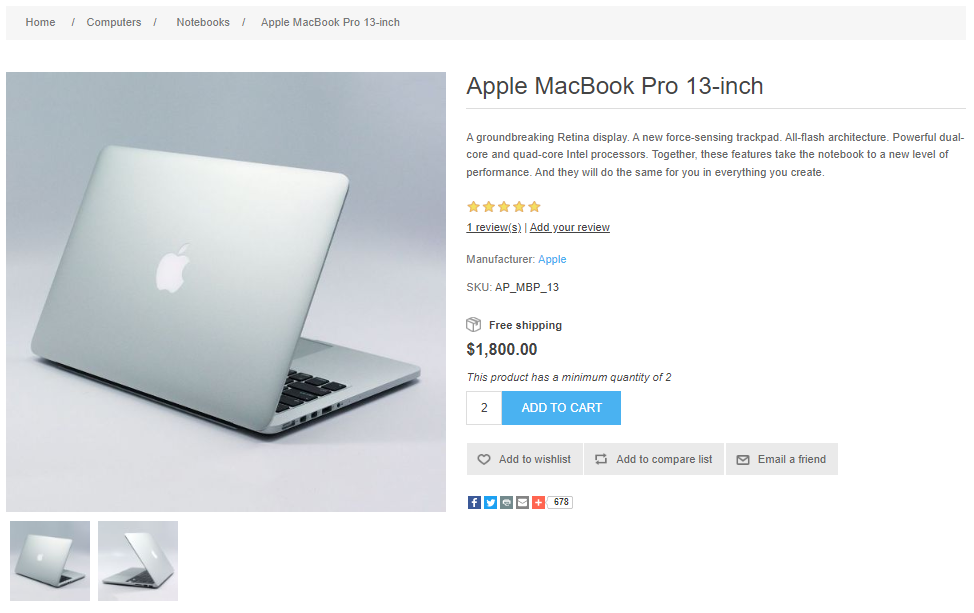


* Kết quả tìm kiếm hiện ra các sản phẩm có chứa từ khóa tìm kiếm



1. Xem chi tiết sản phẩm

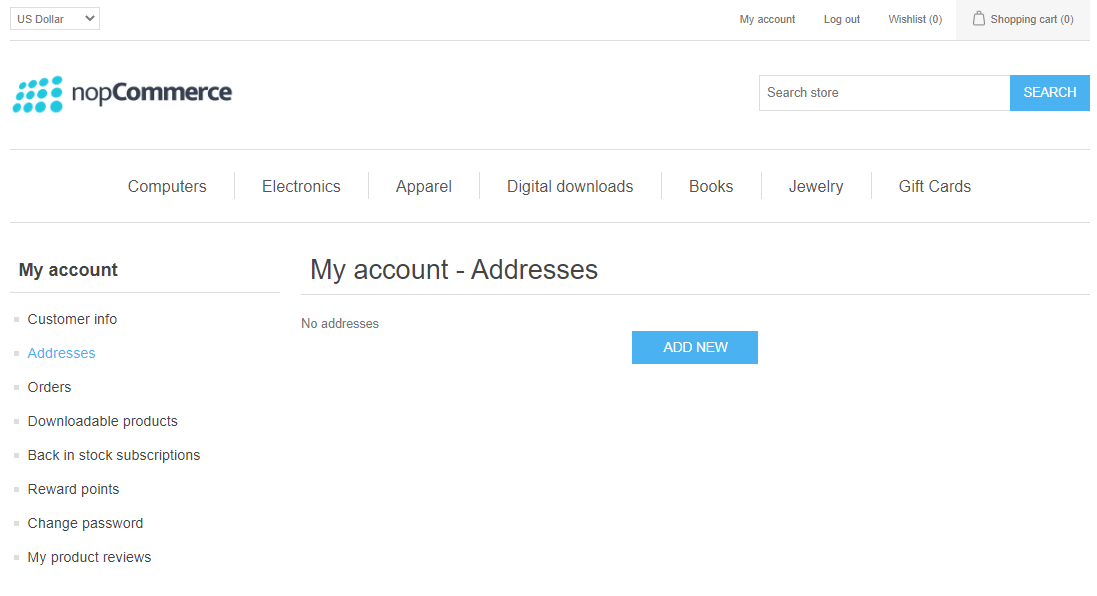
* Khách truy cập có thể chọn xem chi tiết của một sản phẩm, gồm:
* Thông tin sản phẩm: Tên, giá, hình ảnh, mô tả
* Đánh giá, bình luận sản phẩm
* Cho sản phẩm vào giỏ hàng



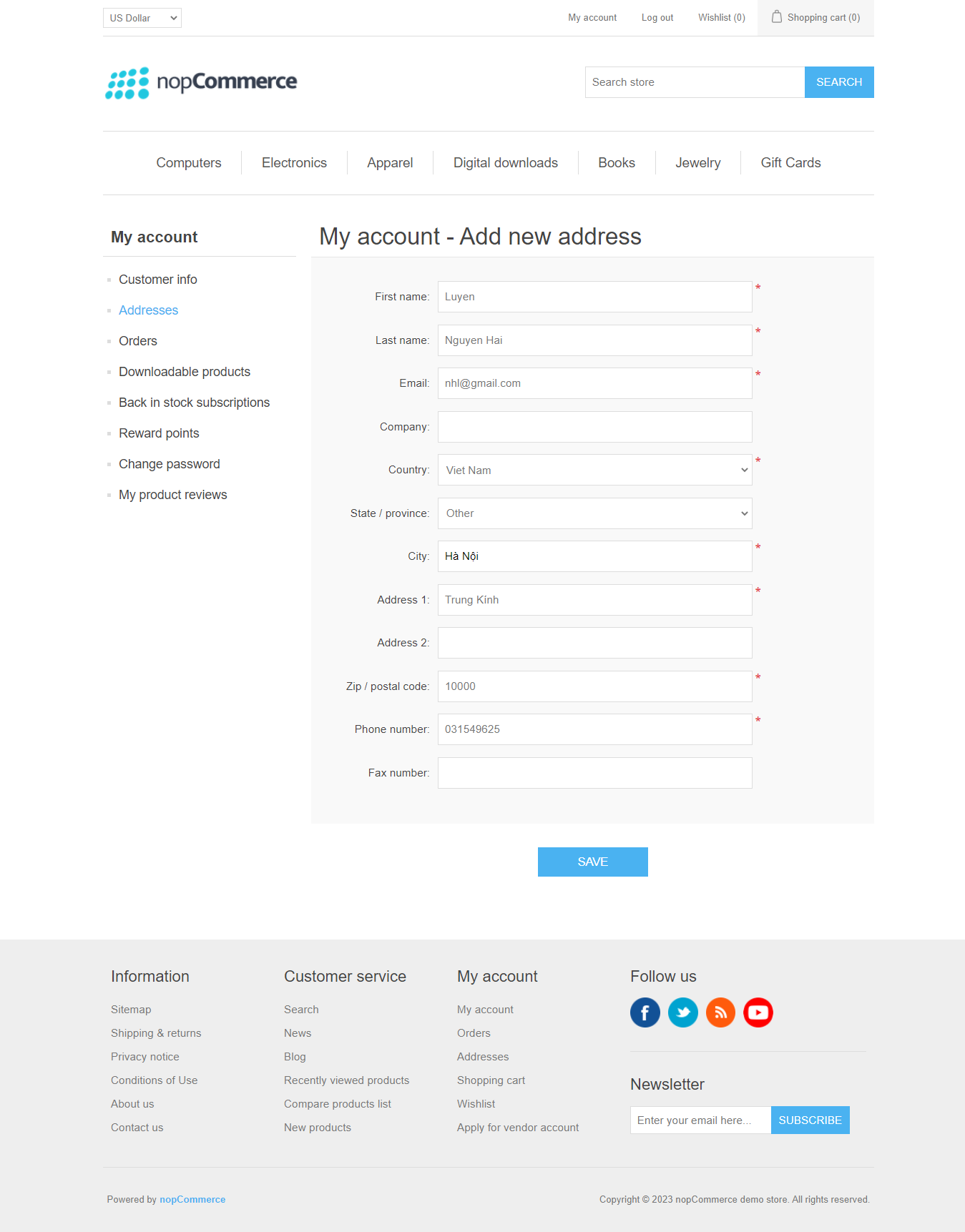
1. Quản lý địa chỉ

a. Thêm địa chỉ

* Trên thanh điều hướng, kích vào nút “My account”
* Kích vào tab “Addresses” và click vào button 

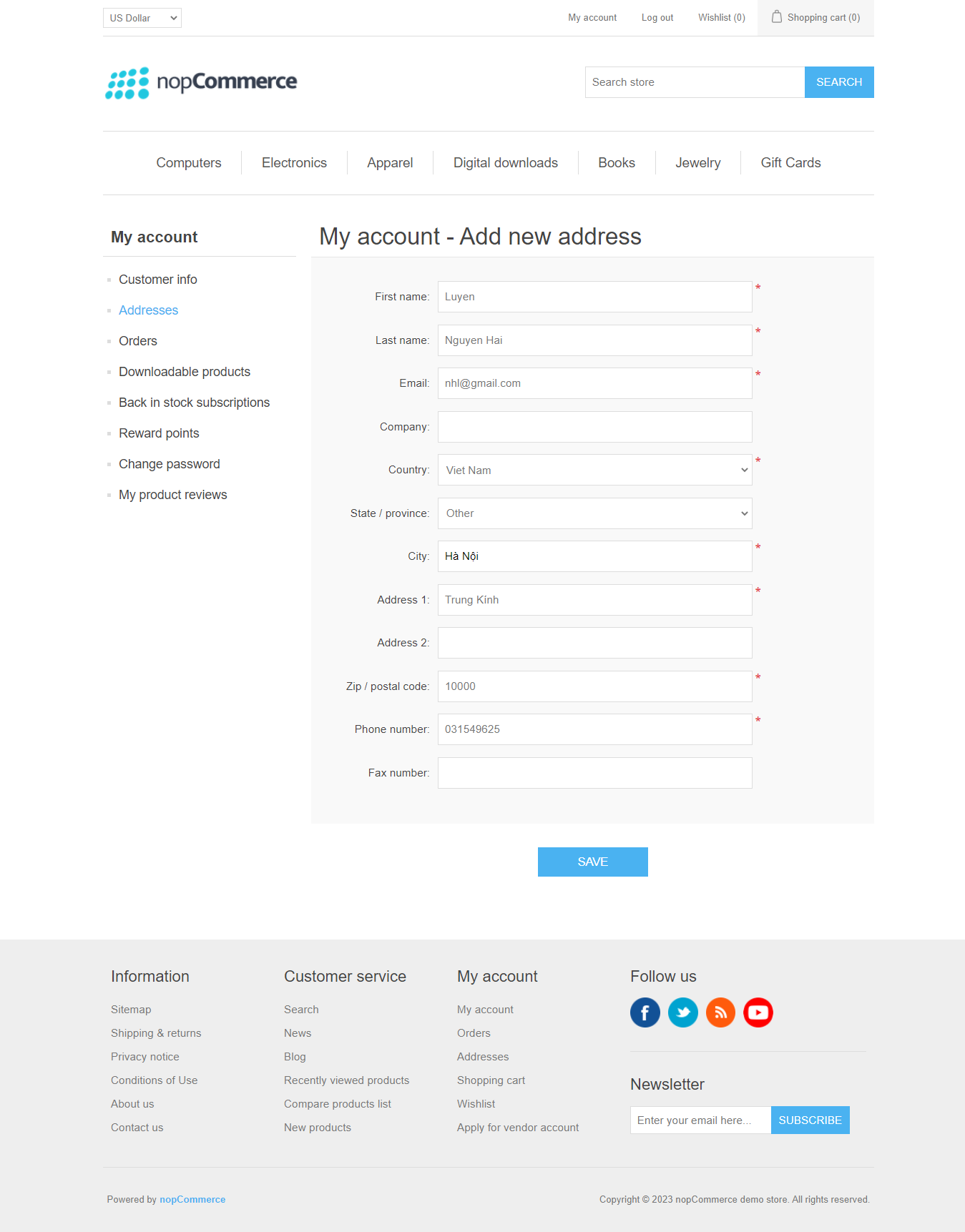


* Điền đẩy đủ thông tin rồi kích nút 



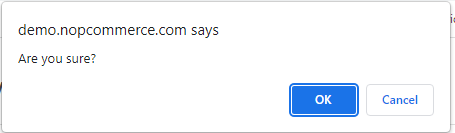
b. Sửa địa chỉ

* Tại trang “Addresses”, click vào nút 
* Điền đẩy đủ thông tin rồi kích nút 



c. Xóa địa chỉ

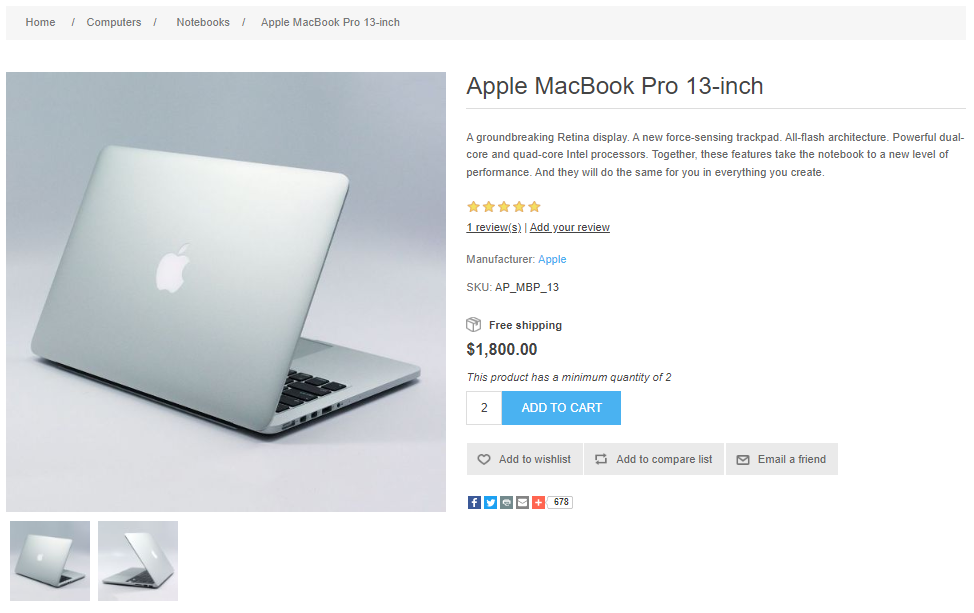
* Tại trang “Addresses”, click vào nút 
* Kích chọn OK



1. Quản lý giỏ hàng

a. Thêm sản phẩm vào giỏ hàng

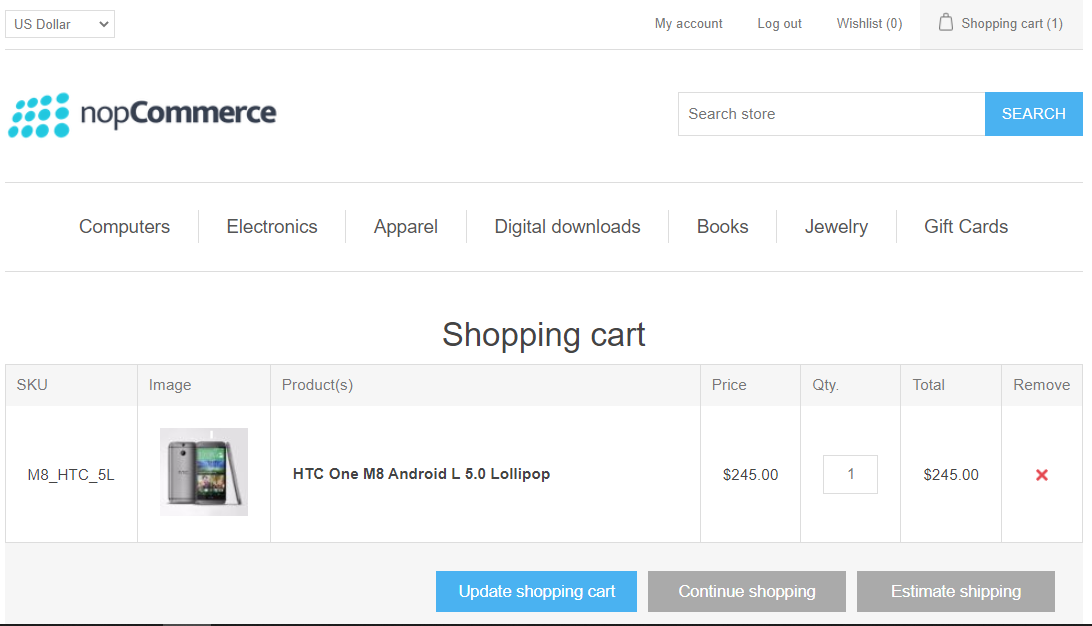
* Vào trang sản phẩm cần mua



* Điền số lượng rồi nhấn nút 

b. Thay đổi số lượng sản phẩm

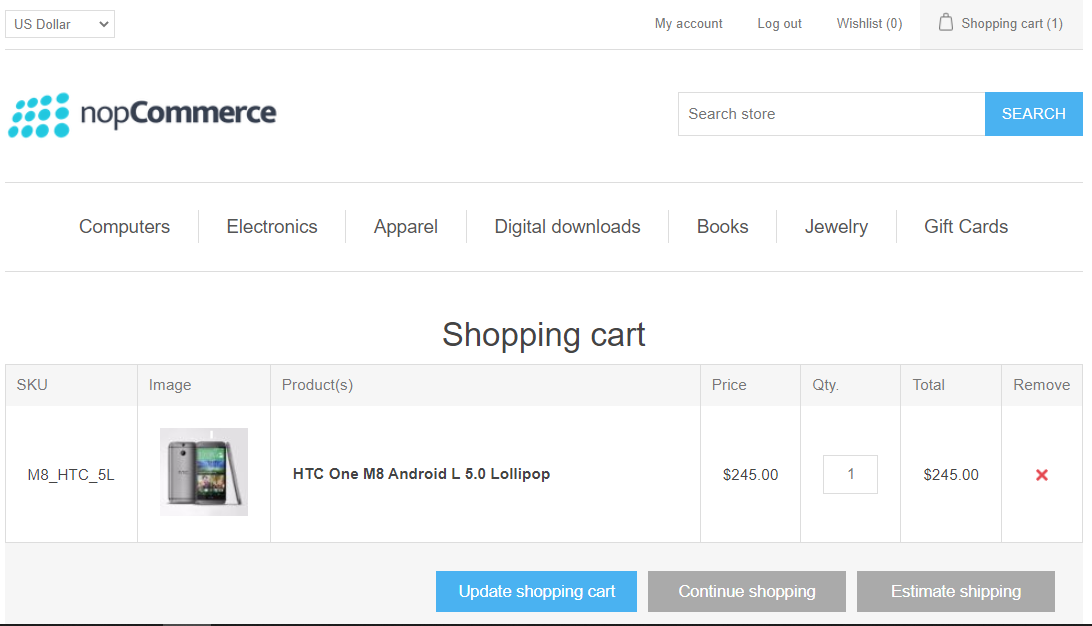
* Trên thanh điều hướng, click vào nút “Shopping Cart”



* Điền số lượng sản phẩm cần thay đổi và ấn nút 

c. Xóa sản phẩm khỏi giỏ hàng

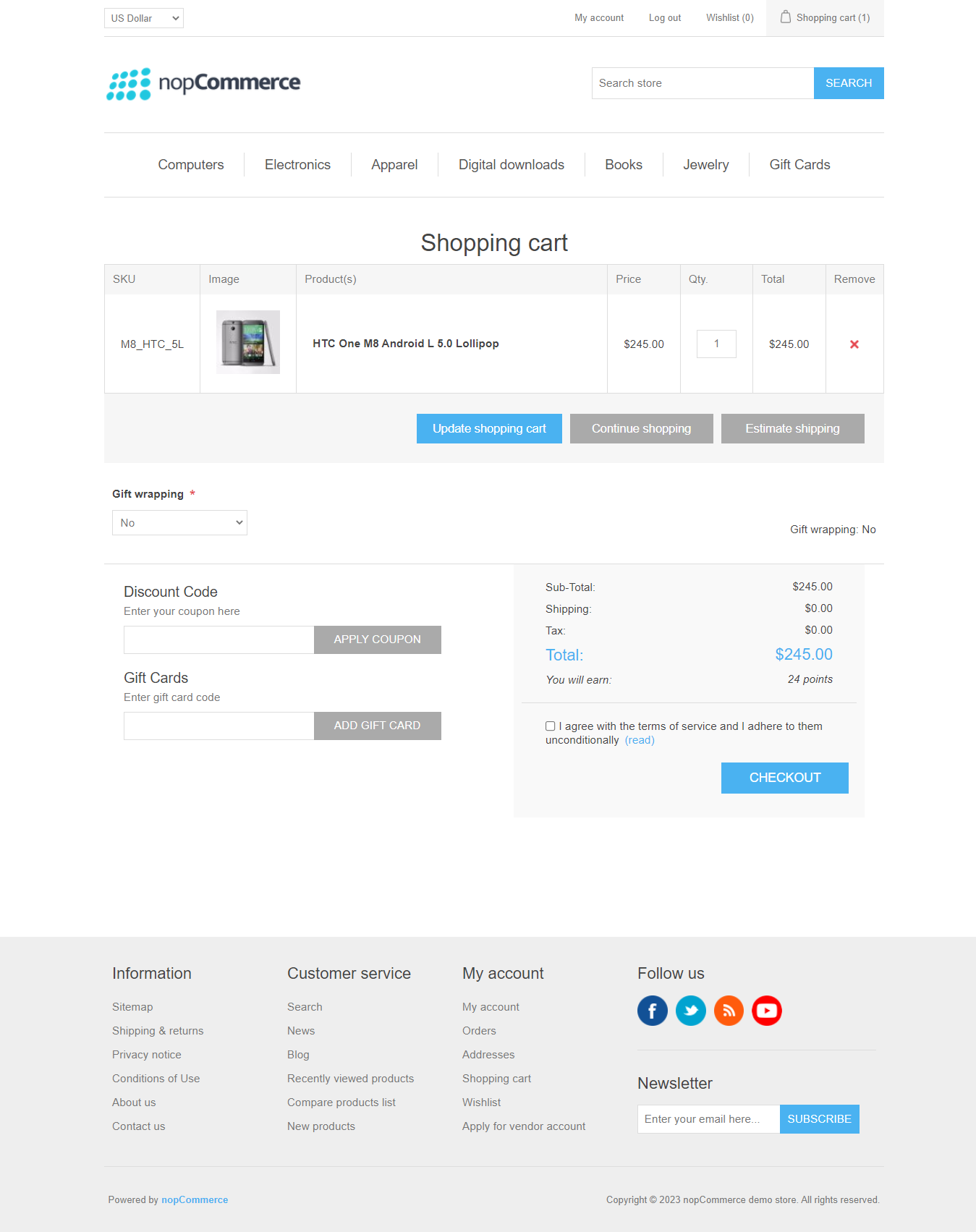
* Trên thanh điều hướng, click vào nút “Shopping Cart”



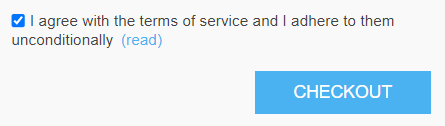
* Chọn sản phẩm cần xóa và ấn nút 

1. Đặt hàng

* Trên thanh điều hướng, click vào nút “Shopping Cart”



* Kích vào nút checkbox và ấn nút 



* Điền các thông tin và ấn nút 

