LỜI CẢM ƠN

Đồ án tốt nghiệp là bước cuối cùng đánh dấu sự trưởng thành của một sinh viên ở giảng đường đại học. Trong quá trình thực hiện đồ án, em đã nhận được sự giúp đỡ của các thầy cô và các bạn, từ phía em đã hoàn thành đồ án đúng theo quy định. Nay em xin được gửi lời cảm ơn sâu sắc và chân thành đến:

Các thầy cô giáo trong khoa Công nghệ thông tin trường Đại học Công nghiệp Hà Nội đã truyền đạt cho chúng em những kiến thức chuyên môn làm cơ sở để thực hiện đồ án tốt nghiệp và tạo điều kiện cho em hoàn thành tốt đồ án.

Xin gửi lời cảm ơn đến bạn Lê Quang Trường, người đã làm ra website được sử dụng để chạy thử nghiệm trong đồ án này. Nhờ có bạn mà em đã có thể thực hiện và hoàn thành được các thử nghiệm của mình.

Đặc biệt, em xin gửi lời tri ân sâu sắc đến giảng viên Ts Đỗ Mạnh Hùng là người đã trực tiếp hướng dẫn em trong đồ án này. Trong suốt quá trình thực hiện, thầy đã tận tình hướng dẫn, giúp em giải quyết các vấn đề khúc mắc trong quá trình làm đồ án tốt nghiệp. Em xin chân thành cảm ơn thầy!

Trong quá trình thực hiện đồ án, do kinh nghiệm và thời gian làm đồ án tốt nghiệp còn nhiều hạn chế, em sẽ không tránh khỏi những thiếu sót. Em rất mong được sự góp ý từ các thầy cô để đồ án tốt nghiệp của em có thể hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc145336876)

[MỤC LỤC ii](#_Toc145336877)

[DANH MỤC TỪ VIẾT TẮT , THUẬT NGỮ v](#_Toc145336878)

[DANH MỤC HÌNH VẼ vi](#_Toc145336879)

[DANH MỤC BẢNG BIỂU vii](#_Toc145336880)

[LỜI NÓI ĐẦU 1](#_Toc145336881)

[CHƯƠNG 1: GIỚI THIỆU VỀ KIỂM THỬ PHẦN MỀM 2](#_Toc145336882)

[1.1 Phần mềm 2](#_Toc145336883)

[1.1.1 Khái niệm 2](#_Toc145336884)

[1.1.2 Phân loại 2](#_Toc145336885)

[1.2 Quy trình phát triển phần mềm 2](#_Toc145336886)

[1.3 Các mô hình phát triển phần mềm 5](#_Toc145336887)

[1.3.1 Waterfall model (Mô hình thác nước) 5](#_Toc145336888)

[1.3.2 V model (Mô hình chữ V) 5](#_Toc145336889)

[1.3.3 Spiral model (Mô hình xoắn ốc) 6](#_Toc145336890)

[1.3.4 Agile model & Scrum Process (Mô hình Agile và quy trình Scrum) 7](#_Toc145336891)

[1.4 Lỗi phần mềm 9](#_Toc145336892)

[1.4.1 Khái niệm 9](#_Toc145336893)

[1.4.2 Trạng thái của lỗi 9](#_Toc145336894)

[1.4.3 Vòng đời của lỗi 11](#_Toc145336895)

[1.5 Khái niệm về kiểm thử phần mềm 12](#_Toc145336896)

[1.6 Quy trình kiểm thử phần mềm 13](#_Toc145336897)

[1.7 Các cấp độ kiểm thử 15](#_Toc145336898)

[1.7.1 Unit Testing (Kiểm thử đơn vị) 15](#_Toc145336899)

[1.7.2 Integration Testing (Kiểm thử tích hợp) 16](#_Toc145336900)

[1.7.3 System Testing (Kiểm thử hệ thống) 16](#_Toc145336901)

[1.7.4 Acceptance Testing (Kiểm thử chấp nhận) 16](#_Toc145336902)

[1.8 Các phương pháp kiểm thử phần mềm 17](#_Toc145336903)

[1.8.1 Kiểm thử hộp trắng (White box testing) 17](#_Toc145336904)

[1.8.2 Black box testing (Kiểm thử hộp đen) 18](#_Toc145336905)

[1.8.4 Gray box testing (Kiểm thử hộp xám) 18](#_Toc145336906)

[1.8.4 So sánh các phương pháp kiểm thử phần mềm 19](#_Toc145336907)

[1.9 Các kỹ thuật thiết kế test case 21](#_Toc145336908)

[1.9.1 Equivalence Class Partitioning (Phân vùng tương đương) 21](#_Toc145336909)

[1.9.2 Boundary Value Analysis (Phân tích giá trị biên) 22](#_Toc145336910)

[1.9.3 Decision Table (Bảng quyết định) 24](#_Toc145336911)

[1.9.4 Đoán lỗi (Error Guessing) 25](#_Toc145336912)

[CHƯƠNG 2: TÌM HIỂU VỀ KIỂM THỬ TỰ ĐỘNG VỚI SELENIUM 26](#_Toc145336913)

[2.1 Khái niệm về kiểm thử tự động 26](#_Toc145336914)

[2.2 Quy trình kiểm thử tự động 26](#_Toc145336915)

[2.3 Mục đích và khi nào cần kiểm thử tự động? 26](#_Toc145336916)

[2.4 Giới thiệu về Selenium 27](#_Toc145336917)

[2.5 Selenium WebDriver 29](#_Toc145336918)

[2.5.1 Khái niệm 29](#_Toc145336919)

[2.5.2 Tại sao nên dùng Selenium WebDriver 30](#_Toc145336920)

[2.6 Cách xác định locators 31](#_Toc145336921)

[2.6.1 Cách xác định locators của element trên website 31](#_Toc145336922)

[2.6.2 Cách lấy xpath của element trên website 32](#_Toc145336923)

[2.7 Ngôn ngữ lập trình Java 34](#_Toc145336924)

[CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ 38](#_Toc145336925)

[3.1 Giới thiệu về website kiểm thử 38](#_Toc145336926)

[3.1.1 Mô tả bài toán 38](#_Toc145336927)

[3.1.2 Đặc tả use case 39](#_Toc145336928)

[3.2 Thiết kế ca kiểm thử 51](#_Toc145336929)

[3.3 Xây dựng test script 66](#_Toc145336930)

[3.4 Kết quả kiểm thử 70](#_Toc145336931)

[KẾT LUẬN 81](#_Toc145336932)

[TÀI LIỆU THAM KHẢO 82](#_Toc145336933)

DANH MỤC TỪ VIẾT TẮT , THUẬT NGỮ

|  |  |
| --- | --- |
| DSD | Đặc tả kỹ thuật thiết kế |
| SDLC  (Software Development Life Cycle) | Quy trình phát triển phần mềm |
| CSDL | Cơ sở dữ liệu |
| Test case | Kịch bản kiểm thử |

DANH MỤC HÌNH VẼ

[Hình 1.1: Quy trình phát triển phần mềm 3](#_Toc145336934)

[Hình 1.2: Mô hình thác nước 5](#_Toc145336935)

[Hình 1.3:Mô hình chữ V 6](#_Toc145336936)

[Hình 1.4: Mô hình Agile 8](#_Toc145336937)

[Hình 1.5: Quy trình Scrum 8](#_Toc145336938)

[Hình 1.6: Các trạng thái của lỗi 9](#_Toc145336939)

[Hình 1.7: Vòng đời của lỗi 11](#_Toc145336940)

[Hình 1.8: Quy trình kiểm thử phần mềm 13](#_Toc145336941)

[Hình 1.9: Kiểm thử hộp trắng (White box testing) 17](#_Toc145336942)

[Hình 1.10: Kiểm thử hộp đen (Black box testing) 18](#_Toc145336943)

[Hình 1.11: Kiểm thử hộp xám (Gray box testing) 19](#_Toc145336944)

[Hình 2.1: Cấu trúc của Selenium 28](#_Toc145336945)

[Hình 2.2: Selenium WebDriver 30](#_Toc145336946)

[Hình 2.3: Xpath 33](#_Toc145336947)

[Hình 2.4: Cách lấy Xpath 34](#_Toc145336948)

[Hình 3.1: Tạo project 67](#_Toc145336949)

[Hình 3.2: Tạo file chức các biến và phương thức chung 67](#_Toc145336950)

[Hình 3.3: Tạo file chứa các đường dẫn xpath và các hàm thao tác kiểm tra website 68](#_Toc145336951)

[Hình 3.4: Tạo file chứa các ca kiểm thử của từng chức năng 68](#_Toc145336952)

[Hình 3.5: Tạo file chứa các chức năng cần kiểm thử 69](#_Toc145336953)

[Hình 3.6: Chạy các ca kiểm thử 69](#_Toc145336954)

DANH MỤC BẢNG BIỂU

[Bảng 1.1: So sánh các phương pháp kiểm thử phần mềm 19](#_Toc145336955)

[Bảng 2.1: Bảng sau liệt kê tất cả cú pháp Java để định vị các phần tử trong Selenium WebDriver 32](#_Toc145336956)

[Bảng 3.1: Đặc tả use case “Đăng ký” 39](#_Toc145336957)

[Bảng 3.2: Đặc tả use case “Đăng nhập” 42](#_Toc145336958)

[Bảng 3.3: Đặc tả use case “Tìm kiếm sản phẩm” 43](#_Toc145336959)

[Bảng 3.4: Đặc tả use case “Xem thông tin chi tiết sản phẩm” 44](#_Toc145336960)

[Bảng 3.5: Đặc tả use case “Gửi thông tin liên hệ” 44](#_Toc145336961)

[Bảng 3.6: Đặc tả use case “Cập nhật thông tin cá nhân” 46](#_Toc145336962)

[Bảng 3.7: Đặc tả use case “Xem lịch sử đặt hàng” 47](#_Toc145336963)

[Bảng 3.8:Đặc tả use case “Đăng xuất” 48](#_Toc145336964)

[Bảng 3.9: Đặc tả use case “Thêm vào giỏ hàng” 49](#_Toc145336965)

[Bảng 3.10: Đặc tả use case “Đặt hàng” 50](#_Toc145336966)

[Bảng 3.11: Bảng các ca kiểm thử cho “Đăng ký” 51](#_Toc145336967)

[Bảng 3.12: Bảng các ca kiểm thử cho “Đăng nhập” 56](#_Toc145336968)

[Bảng 3.13: Bảng các ca kiểm thử cho “Tìm kiếm sản phẩm” 59](#_Toc145336969)

[Bảng 3.14: Bảng các ca kiểm thử cho “Xem thông tin chi tiết sản phẩm” 59](#_Toc145336970)

[Bảng 3.15: Bảng các ca kiểm thử cho “Gửi liên hệ” 60](#_Toc145336971)

[Bảng 3.16: Bảng các ca kiểm thử cho “Cập nhật thông tin cá nhân” 61](#_Toc145336972)

[Bảng 3.17: Bảng các ca kiểm thử cho “Xem lịch sử đặt hàng” 63](#_Toc145336973)

[Bảng 3.18: Bảng các ca kiểm thử cho “Đăng xuất” 64](#_Toc145336974)

[Bảng 3.19: Bảng các ca kiểm thử cho “Thêm vào giỏ hàng” 64](#_Toc145336975)

[Bảng 3.20: Bảng các ca kiểm thử cho “Đặt hàng” 65](#_Toc145336976)

[Bảng 3.21: Bảng kết quả kiểm thử cho “Đăng ký” 70](#_Toc145336977)

[Bảng 3.22: Bảng kết quả kiểm thử cho “Đăng nhập” 73](#_Toc145336978)

[Bảng 3.23: Bảng kết quả kiểm thử cho “Tìm kiếm sản phẩm” 74](#_Toc145336979)

[Bảng 3.24: Bảng kết quả kiểm thử cho “Xem thông tin chi tiết sản phẩm” 75](#_Toc145336980)

[Bảng 3.25: Bảng kết quả kiểm thử cho “Gửi liên hệ” 75](#_Toc145336981)

[Bảng 3.26: Bảng kết quả kiểm thử cho “Cập nhật thông tin cá nhân” 76](#_Toc145336982)

[Bảng 3.27: Bảng kết quả kiểm thử cho “Xem lịch sử đặt hàng” 77](#_Toc145336983)

[Bảng 3.28: Bảng kết quả kiểm thử cho “Đăng xuất” 78](#_Toc145336984)

[Bảng 3.29: Bảng kết quả kiểm thử cho “Thêm vào giỏ hàng” 78](#_Toc145336985)

[Bảng 3.30: Bảng kết quả kiểm thử cho “Đặt hàng” 79](#_Toc145336986)

[Bảng 3.31: Bảng tổng hợp kết quả kiểm thử 80](#_Toc145336987)

LỜI NÓI ĐẦU

Cuộc sống ngày nay càng hiện đại và công nghệ tiên tiến được áp dụng ngày càng nhiều, tạo nên những bước đột phá mạnh mẽ trong nhiều lĩnh vực và ngành nghề, một trong những số đó không thể không nói đến sự phát triển của các sàn giao dịch Thương mại điện tử mua sắm online. Đặc biệt là trong những năm vừa qua, do ảnh hưởng của đại dịch Covid diễn biến phức tạp đã khiến toàn dân buộc phải cách ly tại nhà, tránh tiếp xúc để chống lây lan dịch bệnh, khiến nhu cầu mua sắm của người dùng không thể đáp ứng, các cửa hàng bán thực phẩm chức năng rơi vào nguy cơ phải ngừng hoạt động, phá sản. Hiện nay, với các công ty tin học không ngừng phát triển đã cho ra đời các phần mềm, các trang mạng nhằm cho phép tiến hành thương mại hóa trên Internet để khắc phục, cải thiện, nâng cao cũng như đáp ứng nhu cầu của người mua sắm, là phương án cứu giúp các cửa hàng kinh doanh bán hàng thời trang có thể tránh khỏi trình trạng phải đóng cửa cửa hàng. Khi tạo ra được các website như vậy thì cũng không thể thiếu được việc kiểm thử phần mềm. Kiểm thử nhằm đảm bảo kiểm soát lỗi, chất lượng đầu ra của sản phẩm. Đặc biệt, khi kiểm thử phần mềm là công đoạn chiếm phần lớn thời gian trong quá trình phát triển dự án phần mềm.

Từ những phân tích trên, ý tưởng cho một ứng dụng website bán hàng hình thành nhằm mục đích đáp ứng người tiêu dùng, chỉ với những thao tác đơn giản trên máy tính hoặc điện thoại có kết nối internet, người mua sắm sẽ có tận tay những sản phẩm (thông qua hình thức giao hàng tại nhà) mà không cần phải ra trực tiếp cửa hàng. Điều này vẫn có thể giữ vững được sự cung cầu giữa người mua sắm và người bán, góp phần giữ vững nền kinh tế nước nhà.

Vì vậy, em đã chọn đề tài ***“Kiểm thử tự động bằng Selenium WebDriver cho website bán hàng”*** làm đồ án tốt nghiệp. Trong báo cáo đồ án này, em sẽ tập trung trình bày những nghiên cứu kiến thức cơ bản về quy trình kiểm thử phần mềm. Vai trò, mục tiêu và các kỹ thuật kiểm thử phần mềm, đi sâu hơn vào kiểm thử chức năng, các cách tiếp cận, quy trình và phương pháp kiểm thử chức năng. Phân tích và kiểm thử chức năng website bằng cách triển khai các bước: lập kế hoạch test, thiết kế test, tạo các testcase, thực hiện test và báo cáo kết quả test.

Trong quá trình tìm hiểu lý thuyết và áp dụng những kiến thức, kinh nghiệm của em có được vào hệ thống cũng chưa nhiều và còn nhiều hạn chế cũng như sai sót mong thầy cô và các bạn đóng góp ý kiến để đề tài được hoàn thiện hơn.

# GIỚI THIỆU VỀ KIỂM THỬ PHẦN MỀM

## 1.1 Phần mềm

### 1.1.1 Khái niệm

Phần mềm máy tính (Computer Software) hay gọi tắt là Phần mềm (Software) là một tập hợp những câu lệnh hoặc chỉ thị (Instruction) được viết bằng một hoặc nhiều ngôn ngữ lập trình theo một trật tự xác định, và các dữ liệu hay tài liệu liên quan nhằm tự động thực hiện một số nhiệm vụ hay chức năng hoặc giải quyết một vấn đề cụ thể nào đó.

Phần mềm thực hiện các chức năng của nó bằng cách gửi các chỉ thị trực tiếp đến phần cứng (hay phần cứng máy tính, Computer Hardware) hoặc bằng cách cung cấp dữ liệu để phục vụ các chương trình hay phần mềm khác.

### 1.1.2 Phân loại

*- Phần mềm hệ thống:* Dùng để quản lý hành vi phần cứng máy tính, cung cấp các chức năng cơ bản được người dùng yêu cầu hoặc phần mềm khác chạy đúng (nếu có). Được thiết kế để cung cấp một nền tảng để chạy phần mềm ứng dụng như hệ điều hành Windows, iOS, Android, trình điều khiển thiết bị Driver …

*- Phần mềm ứng dụng:*

* Sử dụng hệ thống máy tính để thực hiện các chức năng đặc biệt hoặc cung cấp các chức năng giải trí ngoài hoạt động cơ bản của chính máy tính.
* Phổ biến trong các phần mềm văn phòng như Microsoft Office, phần mềm game, các công cụ và tiện ích khác …

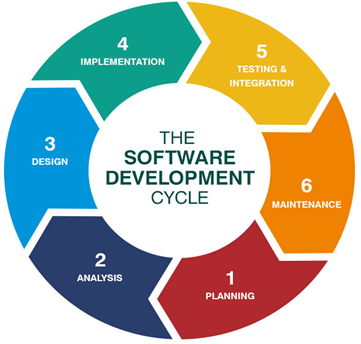
*- Phần mềm dịch mã (trình biên dịch và trình thông dịch):* Dịch các câu lệnh từ mã nguồn của ngôn ngữ lập trình sang dạng ngôn ngữ máy sao cho thiết bị thực thi có thể hiểu được và hiện thực.

*- Nền tảng ứng dụng (ASP.NET, PHP …):* Dựa vào nền tảng ứng dụng Web của Microsoft tạo ra các ứng dụng Web, dịch vụ Web (Web Service)

## 1.2 Quy trình phát triển phần mềm

Theo định nghĩa từ *Indeed*, quy trình phát triển phần mềm (SDLC - Software Development Life Cycle) là toàn bộ quá trình xây dựng lên sản phẩm đáp ứng các thông số kỹ thuật và yêu cầu của người dùng. SDLC cung cấp một tiêu chuẩn quốc tế mà các công ty sản xuất phần mềm có thể sử dụng để xây dựng và cải tiến sản phẩm công nghệ. Một quy trình tốt sẽ luôn tạo ra những sản phẩm đạt tiêu chuẩn.

Quy trình được chia thành 6 bước và mỗi giai đoạn đều có sự tham gia của đội ngũ phát triển phần mềm. Quy trình giúp tương tác hóa các hoạt động và yếu tố với nhau một cách nhịp nhàng, đem lại hiệu quả trong quá trình sản xuất phần mềm.



Hình 1.1: Quy trình phát triển phần mềm

*Giai đoạn 1: Needs identification (Xác định nhu cầu)*

Needs identification là giai đoạn nghiên cứu thị trường và brainstorming (phương pháp động não) của quy trình. Trước khi xây dựng phần mềm, công ty cần thực hiện nghiên cứu sâu rộng thị trường để xác định khả năng tồn tại của sản phẩm. Developers phải xác định các chức năng và dịch vụ mà phần mềm nên cung cấp được cho người tiêu dùng để họ cảm thấy sản phẩm cần.

Ngoài ra, Developers cũng nên thảo luận cùng các bộ phận khác trong công ty về: Điểm mạnh, điểm yếu và cơ hội của sản phẩm. Quá trình phát triển phần mềm chỉ bắt đầu nếu sản phẩm thỏa mãn được mọi thông số nhất thiết để thành công.

*Giai đoạn 2: Requirements Analytics (Phân tích yêu cầu)*

Requirements Analytics là giai đoạn thực hiện khảo sát chi tiết yêu cầu, mong muốn của khách hàng. Sau đó, thông tin sẽ được tổng hợp vào tài liệu đặc tả yêu cầu (Prototype). Tài liệu đặc tả phải đầy đủ các yêu cầu về chức năng, phi chức năng và giao diện. Ngoài ra, tài liệu còn cung cấp một bản phác thảo chi tiết về thành phần, phạm vi, nhiệm vụ của developers và các thông số thử nghiệm tạo ra sản phẩm chất lượng.

Phân tích yêu cầu cũng là giai đoạn mà các developers lựa chọn cách tiếp cận phát triển phần mềm như: Mô hình chữ V (V Model) hay mô hình thác nước (Waterfall).

*Giai đoạn 3: Design (Thiết kế)*

Sau khi đã xác định & phân tích kỹ lưỡng về yêu cầu, chúng ta sẽ chuyển sang giai đoạn nắm vai trò quan trọng thiết yếu của Quy trình phát triển phần mềm – Design (Thiết kế). Tại đây, các kiến trúc sư và nhà phát triển phần mềm sẽ đưa ra các thông số kỹ thuật tiên tiến mà họ tạo ra sản phẩm theo yêu cầu. Vấn đề cần được thảo luận thêm giữa các bên bao gồm: Mức độ rủi ro, thành phần nhóm, công nghệ áp dụng, thời gian, ngân sách, giới hạn của dự án, phương pháp và thiết kế kiến trúc.

Tài liệu DSD (Đặc điểm kỹ thuật thiết kế) sẽ là kết quả cuối cùng của giai đoạn. DSD chỉ định kiến trúc, thành phần, giao tiếp, đại diện front-end và luồng người dùng của sản phẩm.

*Giai đoạn 4: Development (Lập trình)*

Tại giai đoạn 4, developers sẽ lập trình và triển khai thông số thiết kế. Lập trình viễn sẽ coding dựa trên thông số kỹ thuật và yêu cầu của sản phẩm đã được thống nhất trong các giai đoạn trước. Sau khi coding hoàn tất, developers sẽ deploy sản phẩm trong môi trường phát triển (development environment). Lập trình viên sẽ thử nghiệm phiên bản đã tạo ra và điều chỉnh lại cho phù hợp với yêu cầu.

*Giai đoạn 5: Testing (Kiểm thử)*

Sau khi developers đã hoàn thành giai đoạn lập trình, tester sẽ tiếp nhận sản phẩm và tiến hành testing. Tester sẽ tạo test case (Kịch bản kiểm thử) dựa trên tài liệu giải pháp tạo ở giai đoạn 2 và tiến hành kiểm tra. Tester sẽ cập nhật kết quả test vào tool quản lý và thông báo bug (lỗi) đến developers. Tester và developers sẽ cùng nhau phối hợp xử lý các bug và cập nhật trên hệ thống quản lý lỗi. Trong thực tế, tùy theo mô hình phát triển phần mềm mà hoạt động Develop và Kiểm Thử có thể diễn ra song song hoặc tiến hành lần lượt.

Ví dụ như ở mô hình Waterfall, lập trình được thực hiện xong mới đến giai đoạn kiểm thử.

*Giai đoạn 6: Deployment & Maintenance (Triển khai & bảo trì)*

Tại giai đoạn này khi lỗi đã được xử lý xong, nhà phát triển phần mềm sẽ cung cấp sản phẩm hoàn chỉnh đến tay khách hàng. Testing vẫn được diễn ra ở giai đoạn triển khai để đảm bảo sản phẩm luôn có mức độ hoàn hảo cao. Sau khi phát hành, công ty sẽ tạo ra một nhóm bảo trì để quản lý các vấn đề mà khách hàng gặp phải khi sử dụng sản phẩm. Bảo trì giúp khắc phục nhanh các vấn đề nhỏ xảy ra trong quá trình sử dụng phần mềm.

## 1.3 Các mô hình phát triển phần mềm

### 1.3.1 Waterfall model (Mô hình thác nước)

Waterfall model được coi là mô hình phát triển phần mềm đầu tiên được sử dụng. Đây là mô hình áp dụng theo tính tuần tự của các giai đoạn phát triển phần mềm; giai đoạn sau chỉ được phép thực hiện tiếp khi giai đoạn trước đã kết thúc.

Ưu điểm:

- Dễ sử dụng, dễ tiếp cận.

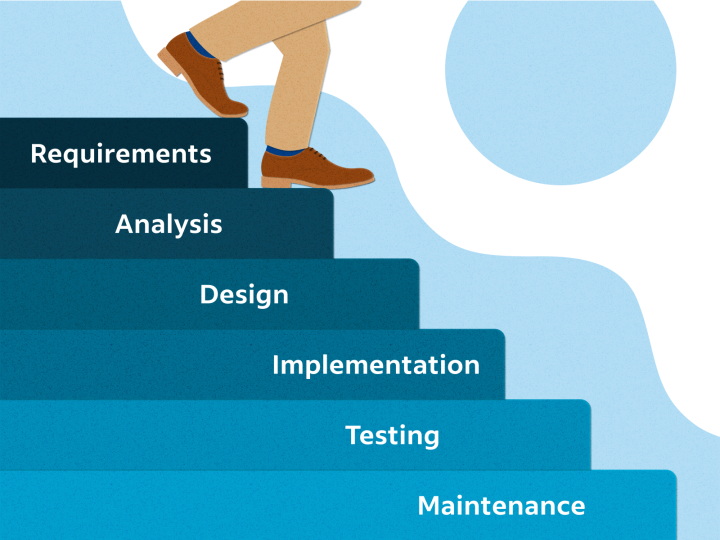
- Các giai đoạn và hoạt động được xác định rõ ràng.

- Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm các lỗi.

Nhược điểm:

- Rất khó để quay lại giai đoạn nào khi nó đã kết thúc

- Ít tính linh hoạt và phạm vi điều chỉnh của nó khá là khó khăn, tốn kém.



Hình 1.2: Mô hình thác nước

### 1.3.2 V model (Mô hình chữ V)

V model là quy trình được sử dụng nhiều tại các công ty sản xuất phần mềm. Khi áp dụng V model, toàn bộ quy trình phát triển phần mềm được chia thành 2 giai đoạn

tiến hành song song tương ứng nhau: Phát triển và Kiểm thử. Trong mô hình chữ V, việc kiểm thử được diễn ra ngay từ giai đoạn lấy yêu cầu nên lỗi được tìm ra ngay từ sớm để khắc phục. Muốn áp dụng được mô hình chữ V thì yêu cầu phần mềm phải xác định rõ ràng; công nghệ phần mềm và các công cụ phải được tìm hiểu kỹ.

Ưu điểm:

- Quá trình phát triển và quy trình quản lý có tính tổ chức và hệ thống.

- Hoạt động tốt cho các dự án có quy mô vừa và nhỏ.

- Kiểm tra bắt đầu từ khi bắt đầu phát triển vì vậy sự mơ hồ được xác định ngay từ đầu.

- Dễ dàng quản lý vì mỗi giai đoạn có các mục tiêu và mục tiêu được xác định rõ ràng.

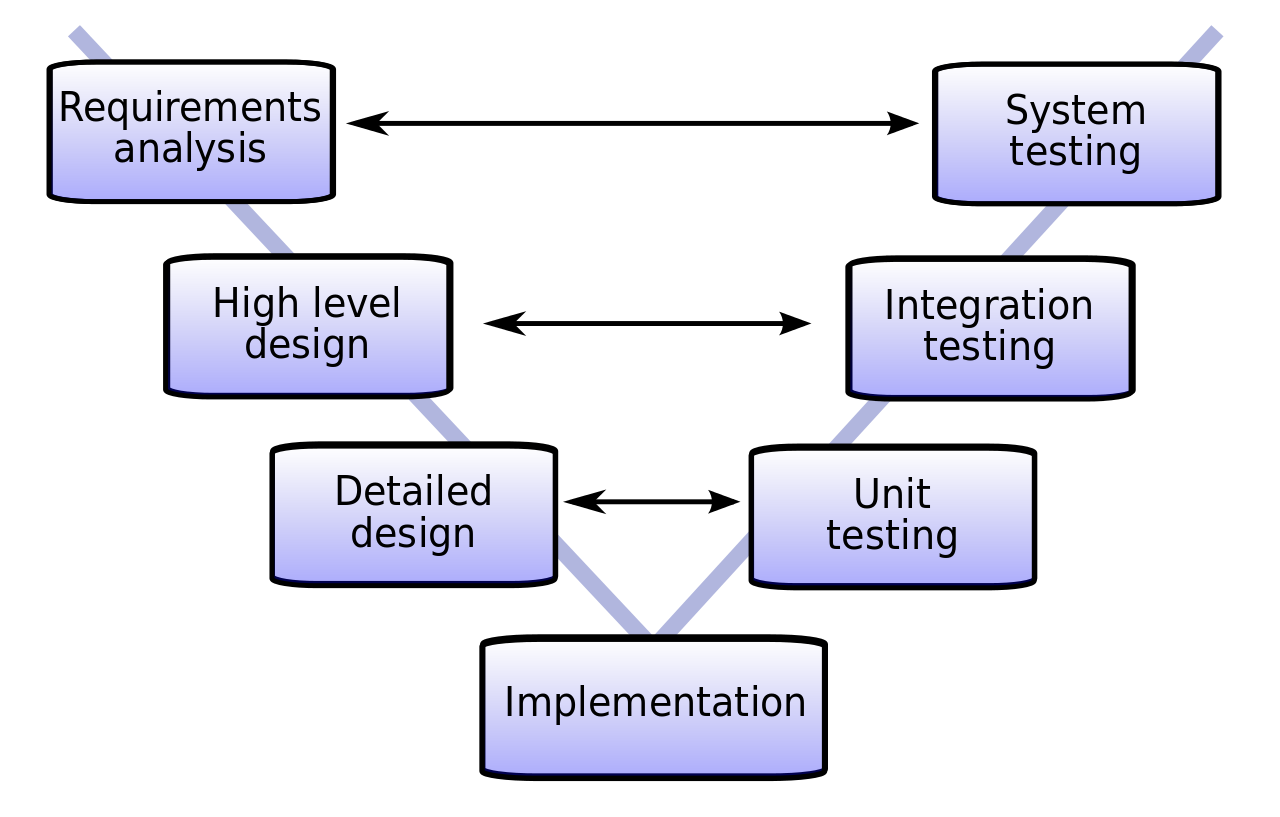
Nhược điểm:

- Không thích hợp cho các dự án lớn và phức tạp

- Không phù hợp nếu các yêu cầu thường xuyên thay đổi.

- Không có phần mềm làm việc được sản xuất ở giai đoạn trung gian.

- Không có điều khoản cho việc phân tích rủi ro nên có sự không chắc chắn và có tính rủi ro.



Hình 1.3:Mô hình chữ V

### 1.3.3 Spiral model (Mô hình xoắn ốc)

Spiral model là quy trình phát triển định hướng rủi ro cho các dự án phần mềm. Mô hình chú trọng vào phân tích rủi ro dự án, bắt đầu với yêu cầu/mục tiêu thiết kế và kết thúc với việc khách hàng kiểm tra tiến độ của từng giai đoạn. Mô hình xoắn ốc là một cách tiếp cận thực tế để phát triển các sản phẩm phần mềm quy mô lớn. Ngoài ra, nhà phát triển và khách hàng hiểu rõ hơn và phản ứng với các rủi ro ở mỗi cấp độ phát triển.

Ưu điểm:

- Hội tụ các tính năng tốt và khắc phục các yếu điểm của nhiều mô hình phát triển khác gặp phải.

- Giám sát dự án dễ dàng và hiệu quả.

- Phù hợp với dự án có nguy cơ thay đổi trong quá trình thực hiện dự án để giảm thiểu rủi ro.

- Dự đoán về thời hạn và chi phí sát với thực tế.

Nhược điểm:

- Phân tích rủi ro khá tốn kém. Chủ yếu áp dụng cho dự án lớn, có tiềm lực về tài chính.

- Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn, phức tạp, cần có đội ngũ chuyên gia về phân tích rủi ro.

### 1.3.4 Agile model & Scrum Process (Mô hình Agile và quy trình Scrum)

Mô hình Agile và quy trình Scrum đang là xu hướng được các công ty phần mềm hướng tới sử dụng trong thời gian gần đây.

*a. Agile model*

Agile model được tạo ra dựa trên 2 mô hình: Iterative (Lặp lại) và Incremental (Tăng dần). Mô hình Agile có thể được sử dụng với bất kỳ loại hình dự án nào, nhưng cần sự tham gia và tính tương tác của khách hàng. Agile được sử dụng khi khách hàng yêu cầu chức năng sẵn sàng trong khoảng thời gian ngắn như 3 - 4 tuần.

Ưu điểm:

- Giảm thời gian cần thiết để tận dụng một số tính năng của hệ thống.

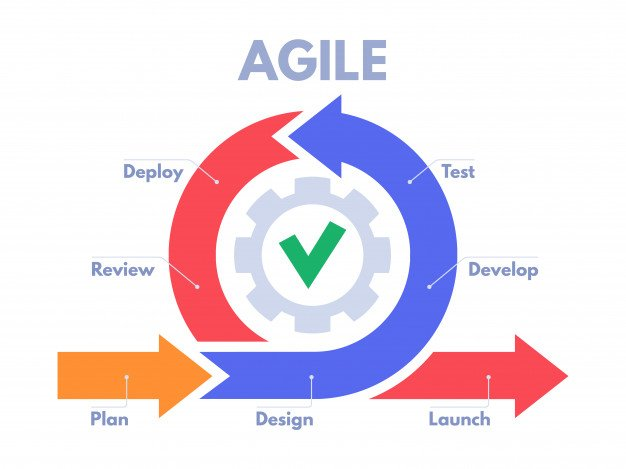
- Kết quả cuối cùng là phần mềm chất lượng cao trong thời gian ít nhất có thể và sự hài lòng của khách hàng.

Nhược điểm:

- Phụ thuộc vào kỹ năng của người phát triển phần mềm.

- Tài liệu được thực hiện ở giai đoạn sau.

- Cần một team có kinh nghiệm.

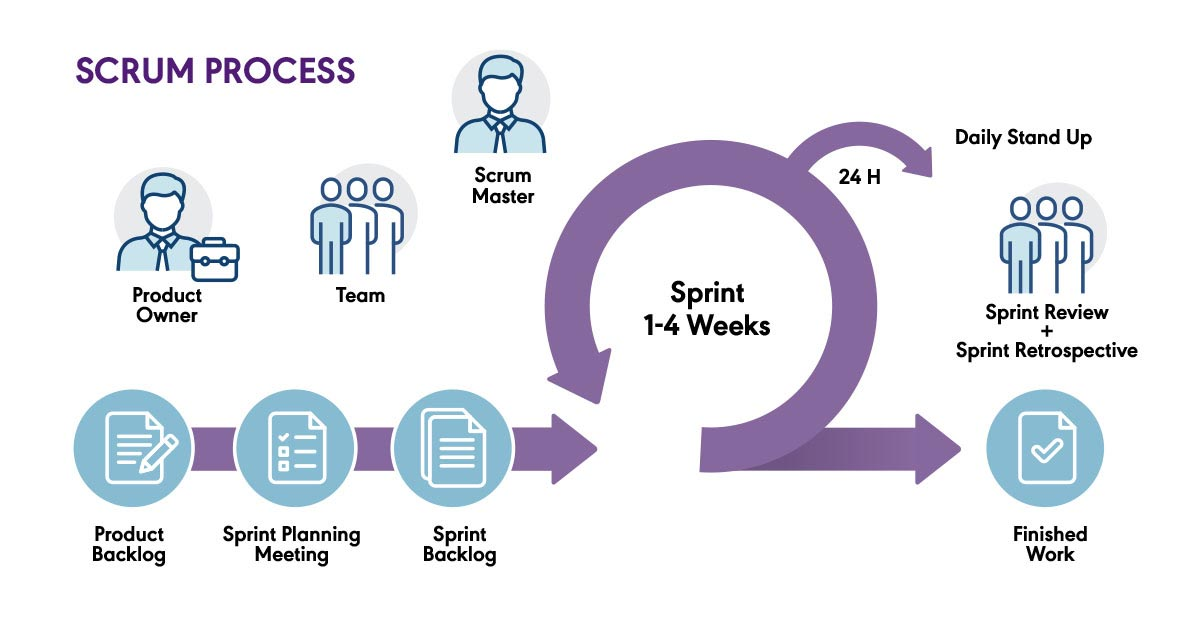


Hình 1.4: Mô hình Agile

*b. Scrum Process*

Scrum là một “khung quản lý dự án” được áp dụng rất rộng rãi từ những dự án đơn giản với một nhóm phát triển nhỏ cho đến những dự án có yêu cầu rất phức tạp với hàng trăm người tham gia. Ngoài ra, Scrum Process cũng phù hợp với những dự án đòi hỏi khung thời gian cố định.

Trong Scrum, công việc sẽ được chia nhỏ thành nhiều giai đoạn gọi là Sprint. Mỗi Sprint chỉ kéo dài từ 1 đến 4 tuần, không quá một tháng. Đầu Sprint sẽ lên kế hoạch làm những yêu cầu nào rồi thực hiện code và test. Cuối Sprint là một sản phẩm hoàn thiện cả code lẫn test có thể demo và chạy được.



Hình 1.5: Quy trình Scrum

## 1.4 Lỗi phần mềm

### 1.4.1 Khái niệm

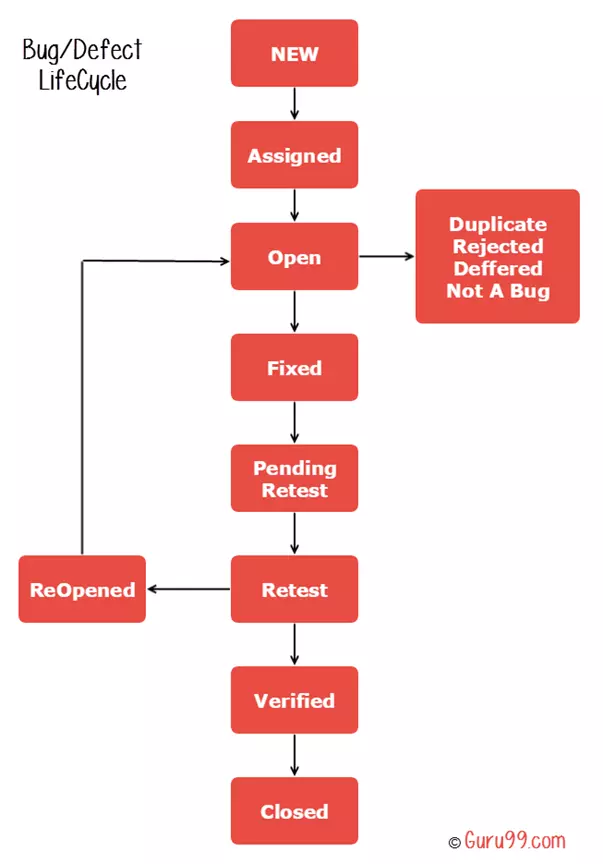
Testing là công việc không thể thiếu trong quá trình xây dựng sản phẩm phần mềm. Cho dù sản phẩm của bạn là một đoạn chương trình, một chức năng hay toàn bộ ứng dụng thì bạn đều phải thực hiện testing trước khi bàn giao. Đó là lúc kiểm tra lại xem sản phẩm làm có đúng yêu cầu khách hàng không? Có vận hành ổn định trên nhiều tình huống giả định không? Có lỗi phát sinh nào không, nguyên nhân là gì để biết cách khắc phục và hoàn thiện sản phẩm.

Lỗi phần mềm được gọi là Defect hoặc Bug hoặc Fault trong tiếng anh.

Không phải tất cả các Lỗi gây ra đều xảy ra do code, lỗi có thể đến từ Đặc tả yêu cầu, thiết kế..

### 1.4.2 Trạng thái của lỗi

Các trạng thái của một bug/defect thường sẽ thay đổi tùy từng dự án. Dưới đây là sơ đồ vòng đời của một bug/defect, bao gồm tất cả các trạng thái có thể:



Hình 1.6: Các trạng thái của lỗi

- New: Khi một lỗi mới được ghi lại và đăng lần đầu tiên. Nó được gán một trạng thái là “New”.

- Assigned: Một khi bug đã được đăng bởi tester thì test leader sẽ phê duyệt lỗi và chuyển giao lỗi cho nhóm phát triển.

- Open: Dev bắt đầu phân tích và thực hiện sửa lỗi. Cũng có khả năng vấn đề có vẻ không phù hợp, trong trường hợp đó thì Dev có thể chuyển vấn đề sang bốn trạng thái sau dựa trên các lý do cụ thể:

* Duplicate: Nếu lỗi được lặp lại hai lần hoặc lỗi tương ứng với cùng một khái niệm về lỗi, trạng thái được thay đổi thành “Duplicate/trùng lặp”.
* Rejected: Nếu dev cảm thấy lỗi không phải là khiếm khuyết thực sự thì nó sẽ thay đổi lỗi thành “Rejected/Loại bỏ”.
* Deferred: Nếu lỗi hiện tại không phải là ưu tiên chính và nếu dự kiến sẽ được sửa trong bản phát hành tiếp theo, thì trạng thái “Deferred/Trì hoãn” được gán cho các lỗi đó
* Not a bug: Nếu nó không ảnh hưởng đến chức năng của ứng dụng thì trạng thái được gán cho lỗi là “Not a bug/Không phải là lỗi”.

- Fixed: Khi Dev hiện đã sửa xong lỗi bằng cách sửa code và đã xác nhận là sửa xong, bug có thể được chuyển sang trạng thái “Fixed/Đã sửa”.

- Pending Retest: Sau khi sửa lỗi, dev bàn giao lại bug cho bên tester. Vì quá trình kiểm thử vẫn đang được diễn ra bởi các tester nên trạng thái được chỉ định là “pending retest/kiểm tra lại đang chờ xử lý”.

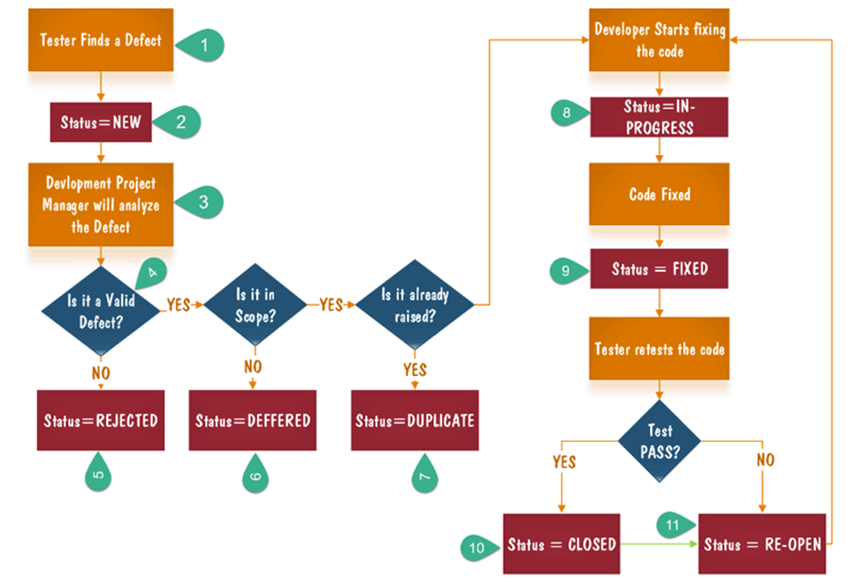
- Retest: Tester thực hiện test lại chương trình ở giai đoạn này để kiểm tra xem lỗi đã được fix hay chưa và thay đổi trạng thái thành “Retest/Kiểm tra lại”.

- Verified: Tester kiểm tra lại lỗi sau khi dev đã fixed. Nếu không có lỗi được phát hiện trong phần mềm, thì lỗi đã được sửa và trạng thái được gán là “Verified/đã được xác minh”.

- Reopen: Nếu lỗi vẫn tồn tại ngay cả sau khi dev đã sửa lỗi, tester sẽ thay đổi trạng thái thành “Reopen/mở lại”. Bug 1 lần nữa quay lại chu kỳ mới.

- Closed: Nếu lỗi không còn tồn tại thì tester sẽ gán trạng thái "Closed/Đã đóng"."

### 1.4.3 Vòng đời của lỗi



Hình 1.7: Vòng đời của lỗi

1. Tester tìm thấy bug/defect

2. Gán trạng thái cho bug: New/Mới

3. Chuyển bug sang cho Quản lý dự án để phân tích

4. Quản lý dự án quyết định xem bug có hợp lệ không

5. Nếu như lỗi không hợp lệ, trạng thái sẽ được chuyển thành “Rejected/Đã từ chối”

6. Nếu lỗi không bị rejected thì bước tiếp theo là kiểm tra xem nó có nằm trong phạm vi không. Giả sử chúng ta có một chức năng khác - chức năng email cho cùng một ứng dụng và bạn thấy có vấn đề với điều đó. Nhưng nó không nằm trong scope của lần phát hành ứng dụng lần này, trạng thái của bug đó có thể chuyển thành “Postponed/hoãn”.

7. Tiếp theo, người quản lý cần xác minh xem đã có bug nào tương tự đã được tìm ra trước đó hay chưa. Nếu đã có rồi, bug này được chuyển trạng thái thành “Duplicate/trùng lặp”

8. Nếu không có vấn đề gì phát sinh trong khi dev fix bug thì bug này được chuyển sang trạng thái là “In- progress/đang tiến hành”

9. Khi code được fixed. Bug sẽ được gán trạng thái là “Fixed/đã sửa xong”

10. Tiếp theo, tester sẽ test lại phần code vừa được sửa. Nếu như các phần test cases liên quan đều passed thì bug đó được đóng lại hay được chuyển trạng thái thành “Closed”. Nếu các trường hợp kiểm thử thất bại một lần nữa, lỗi được mở lại/re-opened và lại được chuyển giao sang cho dev

11. Hãy xem xét một tình huống trong lần release đầu tiên, một lỗi được tìm thấy theo thứ tự Fix đã được sửa và gán trạng thái đóng. Trong lần nâng cấp thứ hai, lỗi tương tự lại xuất hiện trở lại. Trong những trường hợp như vậy, một khiếm khuyết kín sẽ được mở lại

## 1.5 Khái niệm về kiểm thử phần mềm

Kiểm thử phần mềm là quá trình thực thi chương trình với mục đích tìm ra lỗi. 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.

Các hoạt động kiểm thử tồn tại trước và sau khi thực hiện kiểm thử. Các hoạt động này bao gồm lập kế hoạch và kiểm soát, lựa chọn điều kiện thử nghiệm, thiết kế và thực hiện các trường hợp thử nghiệm, kiểm tra kết quả, đánh giá các tiêu chí thoát, báo cáo về quá trình thử nghiệm và kiểm thử hệ thống, và hoàn thiện hoặc hoàn thành các hoạt động đóng sau khi giai đoạn thử nghiệm đã hoàn thành. Kiểm tra cũng bao gồm việc xem xét các tài liệu (bao gồm cả nguồn) và phân tích tĩnh.

Kiểm thử có thể là công việc bao gồm các mục đích sau:

- Tìm kiếm khuyết điểm

- Đạt được niềm tin về mức độ chất lượng

- Cung cấp thông tin để đưa ra quyết định

- Ngăn chặn các lỗi

Quá trình suy nghĩ và các hoạt động liên quan đến việc thiết kế kiểm thử sớm trong vòng đời sản phẩm (xác minh cơ sở thử nghiệm thông qua thiết kế thử nghiệm) có thể giúp ngăn ngừa các khiếm khuyết được đưa vào mã. Việc xem xét các kiến thức tài liệu (ví dụ: yêu cầu) và xác định và giải quyết các vấn đề cũng giúp ngăn ngừa các khiếm khuyết xuất hiện trong mã.

Các quan điểm khác nhau trong thử nghiệm có tính đến các mục tiêu khác nhau:

- Trong kiểm thử phát triển (ví dụ: kiểm thử thành phần, tích hợp và hệ thống), mục tiêu chính có thể là gây ra càng nhiều lỗi càng tốt để các lỗi trong phần mềm được xác định và có thể được sửa chữa.

- Trong thử nghiệm chấp nhận, mục tiêu chính có thể là xác nhận rằng hệ thống hoạt động như mong đợi, để có được sự tin tưởng rằng nó đã đáp ứng các yêu cầu.

- Trong một số trường hợp, mục tiêu chính của thử nghiệm có thể là đánh giá chất lượng của phần mềm này (không có ý định sửa chữa các khiếm khuyết), cung cấp thông tin cho các bên liên quan về nguy cơ giải phóng hệ thống tại một thời điểm nhất định.

- Kiểm tra bảo trì thường bao gồm kiểm tra không có sản phẩm mới nào được đưa ra trong quá trình phát triển các thay đổi.

- Trong quá trình thử nghiệm hoạt động, mục tiêu chính có thể là đánh giá các đặc tính của hệ thống như độ tin cậy hoặc tính khả dụng.

Gỡ lỗi và kiểm thử là khác nhau. Thử nghiệm động có thể cho thấy lỗi do các khuyết tật gây ra. Gỡ lỗi là hoạt động phát triển để tìm, phân tích và loại bỏ nguyên nhân gây ra lỗi. Kiểm tra lại sau đó người kiểm tra đảm bảo việc sửa lỗi thực sự khắc phục được lỗi. Trách nhiệm đối với các hoạt động thường là tester kiểm tra và gỡ lỗi nhà phát triển.

## 1.6 Quy trình kiểm thử phần mềm

Về cơ bản, quy trình kiểm thử phần mềm gồm 5 giai đoạn: Planning & Control (lập kế hoạch và kiểm soát), Analysis & Design (phân tích và thiết kế), Implementation & Execution (hoàn thiện và thực hiện kiểm thử), Test Report & Evaluation (đánh giá điều kiện kết thúc và báo cáo) và Test Closure (các hoạt động kết thúc việc kiểm thử).

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Hình 1.8: Quy trình kiểm thử phần mềm

*Lập kế hoạch và kiểm soát (planning and control)*

- Planning: Xác định mục tiêu, nhiệm vụ và các rủi ro của dự án

- Planning bao gồm các nội dung:

* Xác định phạm vi (scope), rủi ro (risk), và mục đích việc kiểm thử.
* Phương pháp kiểm thử (test approach).
* Xác định tài nguyên kiểm thử (testing resource) như: Con người, môi trường…
* Lên kế hoạch phân tích kiểm thử, thiết kế kiểm thử, thực hiện kiểm thử và đánh giá.
* Xác định điều kiện kết thúc (exit criteria).

- Control:

* So sánh tiến độ thực tế so với kế hoạch, thực hiện báo cáo tình trạng sai lệch so với kế hoạch.
* Đồng thời đưa ra các hành động cần thiết để đáp ứng được mục tiêu và nhiệm vụ của dự án.

*Phân tích và Thiết kế (analyst and design)*

- Bao gồm các nội dung:

* Kiểm tra lại các thông tin kiểm thử cơ bản (test basic) bao gồm: Yêu cầu, tài liệu thiết kế, tài liệu phân tích rủi ro ...
* Xác định các điều kiện kiểm thử.
* Thiết kế test case.
* Thiết kế môi trường kiểm thử, xác định cơ sở hạ tầng và công cụ cần thiết.

*Hoàn thiện và thực hiện kiểm thử (Test implement and execution)*

- Implement:

* Phát triển và đánh độ ưu tiên cho các Test case và tạo dữ liệu test (test data).
* Tạo ra bộ kiểm thử (test suites) từ các Test case.
* Hoàn thiện và kiểm tra môi trường.

- Execution:

* Thực hiện bộ test suites hoặc từng Test case riêng lẻ, đảm bảo theo trình tự thời gian của dự án.
* Ghi lại kết quả của việc thực hiện test, phiên bản phần mềm sử dụng.
* So sánh kết quả thực tế (actual results) và kết quả mong đợi (expected results)

*Đánh giá điều kiện kết thúc và báo cáo (Evaluating exit criteria and reporting)*

- Evaluating exit criteria: là quá trình đánh giá xem khi nào sẽ dừng việc test.

- Exit criteria sẽ được thực hiện khi:

* Một lượng lớn test case được thực hiện với tỷ lệ % pass nhất định.
* Tỷ lệ lỗi ở dưới mức nhất định.
* Khi đến hạn (deadlines).

- Exit criteria có nhiệm vụ:

* Đánh giá xem có cần test thêm hoặc có cần thay đổi điều kiện kết thúc.
* Dùng để viết báo cáo tóm tắt kiểm thử (test summary report) cho các bên liên quan.

*Các hoạt động kết thúc việc kiểm thử (Test closure activities)*

- Test closure activities: Là các hoạt động kết thúc việc kiểm thử được thực hiện khi phần mềm sẵn sàng chuyển giao.

- Nhiệm vụ:

* Đảm bảo rằng tất cả các sản phẩm đã được plan đã được giao (deliver) và tất cả các báo cáo về sự cố đã được giải quyết.
* Hoàn thiện và lưu trữ các bộ dữ liệu test: Test script, tài liệu về môi trường test … để sử dụng sau này.
* Chuyển giao lại bộ dữ liệu test cho đội bảo trì
* Rút ra bài học cho các dự án sau này.

## 1.7 Các cấp độ kiểm thử

### 1.7.1 Unit Testing (Kiểm thử đơn vị)

- Kiểm thử đơn vị là cấp độ kiểm thử cơ bản, thực hiện test từng module nhỏ trong hệ thống.

- Kiểm thử đơn vị có thể được thực hiện tách biệt với phần còn lại của hệ thống tùy thuộc vào mô hình vòng đời phát triển được chọn cho ứng dụng cụ thể đó.

- Mục đích: để xác nhận mỗi thành phần của phần mềm thực hiện đúng với thiết kế.

- Kiểm thử đơn vị thường do lập trình viên thực hiện.

### 1.7.2 Integration Testing (Kiểm thử tích hợp)

- Kiểm thử tích hợp có nghĩa là kiểm thử kết hợp. Một dự án phần mềm được kết hợp bởi nhiều module riêng lẻ khác nhau và được code bởi nhiều lập trình viên khác nhau. Chính vì thế kiểm thử tích hợp là tích hợp kiểm tra các module riêng lẻ với nhau thành một nhóm.

- Tích hợp kiểm tra việc truyền dữ liệu giữa các module, tích hợp kiểm tra các hàm lại với nhau, các màn hình với nhau theo từng module hoặc theo chức năng.

- Mục đích: để đảm bảo rằng hệ thống tích hợp đã sẵn sàng để thử nghiệm hệ thống.

- Kiểm thử tích hợp được thực hiện sau khi kiểm tra đơn vị và trước khi kiểm tra hệ thống.

- Integration testing được thực hiện bởi một người thử nghiệm cụ thể hoặc một nhóm kiểm thử.

- Một số phương pháp kiểm thử tích hợp:

* Phương pháp kiểm thử Bigbang
* Phương pháp kiểm thử Topdown
* Phương pháp kiểm thử Bottom up
* Phương pháp kiểm thử Sandwich

### 1.7.3 System Testing (Kiểm thử hệ thống)

- System Testing là thực hiện kiểm thử một hệ thống đã được tích hợp hoàn chỉnh để xác minh rằng nó đúng yêu cầu của phần mềm.

- Kiểm thử hệ thống nằm trong phạm vi kiểm thử hộp đen và do đó, không yêu cầu kiến ​​thức về thiết kế bên trong của mã hoặc logic.

- Kiểm thử hệ thống thường là thử nghiệm cuối cùng để xác minh rằng hệ thống được phân phối đáp ứng các đặc điểm kỹ thuật và mục đích của nó.

- Kiểm thử hệ thống nên thực hiện kiểm thử chức năng và phi chức năng và được thực hiện bởi tester.

### 1.7.4 Acceptance Testing (Kiểm thử chấp nhận)

- Sau khi kiểm tra hệ thống đã sửa tất cả hoặc hầu hết các lỗi, hệ thống sẽ được gửi đến người dùng hoặc khách hàng để kiểm tra chấp nhận. Về cơ bản kiểm thử chấp nhận cũng khá giống kiểm thử hệ thống nhưng được thực hiện bởi khách hàng.

- Mục đích: đảm bảo phần mềm đáp ứng đúng yêu cầu của khách hàng. Sản phẩm nhận được sự chấp nhận từ khách hàng/ người dùng cuối.

- Kiểm thử chấp nhận được chia thành 2 mức khác nhau:

* Kiểm thử alpha: được thực hiện tại nơi phát triển phần mềm bởi những người trong tổ chức nhưng không tham gia phát triển phần mềm.
* Kiểm thử beta: được thực hiện tại bởi khách hàng/ người dùng cuối tại địa điểm của người dùng cuối.

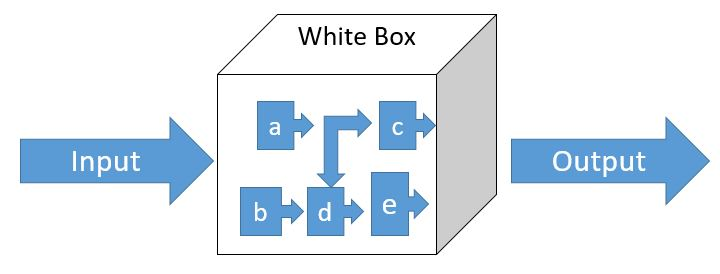
## 1.8 Các phương pháp kiểm thử phần mềm

### 1.8.1 Kiểm thử hộp trắng (White box testing)

- Kiểm thử hộp trắng (White box testing) là phương pháp kiểm thử phần mềm nhằm kiểm tra thuật toán, cấu trúc code bên trong của sản phẩm. Hộp trắng tượng trưng cho khả năng nhìn xuyên qua lớp vỏ bên ngoài của phần mềm để thấy được hoạt động bên trong của chúng.

- Mục đích của White box testing là đảm bảo tất cả các câu lệnh và điều kiện sẽ được thực hiện đúng, kết quả đầu ra như mong đợi. Tester sẽ xác minh luồng hoạt động cho ứng dụng bằng cách kiểm tra một loạt đầu vào (Input) đã được xác định từ trước có dẫn đến đầu ra (Output) như dự kiến không? Nếu Output không khớp với kỳ vọng, có nghĩa là sản phẩm đang bị lỗi.

- Am hiểu về lập trình, có kiến thức về công nghệ thông tin là điều kiện tiên quyết để tester có thể tiến hành kiểm thử hộp trắng. Thông thường, nhiệm vụ thực thi White box Testing sẽ do chính các Developers đảm nhiệm.



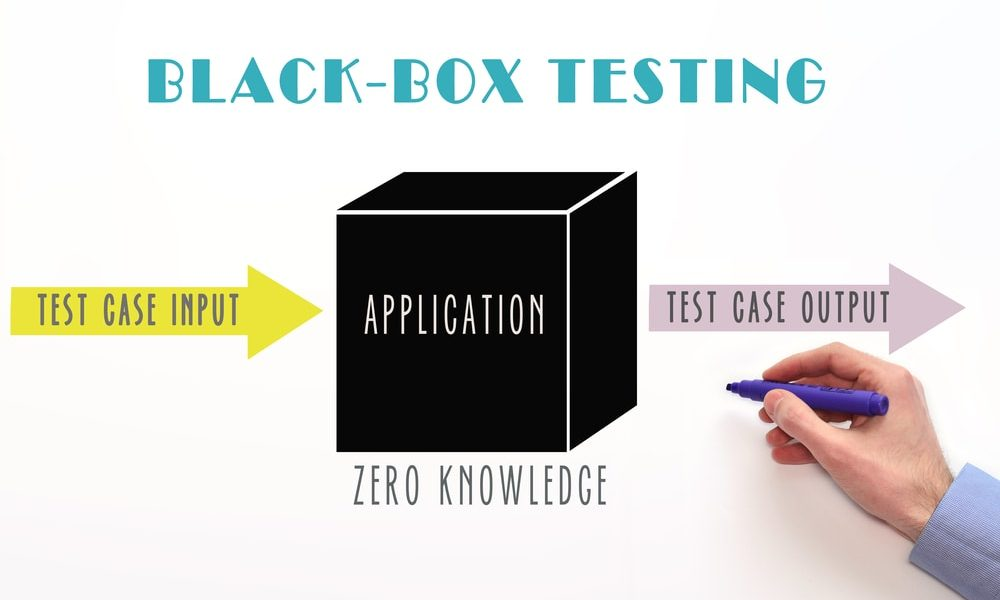
Hình 1.9: Kiểm thử hộp trắng (White box testing)

### 1.8.2 Black box testing (Kiểm thử hộp đen)

- Trái ngược với kiểm thử hộp trắng, kiểm thử hộp đen (Black box testing) là phương pháp test dựa trên đầu vào và đầu ra của chương trình mà không quan tâm tới code bên trong được viết ra sao. Tester sẽ xem phần mềm như là một hộp đen, chỉ nhìn được lớp vỏ bên ngoài mà không nhìn được cấu trúc bên trong.

- Ưu điểm của kiểm thử hộp đen chính là các tester không cần có nền tảng công nghệ, không cần biết ngôn ngữ lập trình đều có thể thực hiện test được. Người kiểm thử khi vận dụng Black box testing sẽ là một phương diện độc lập, có cái nhìn khách quan về sản phẩm.

- Phương pháp kiểm thử hộp đen sẽ cố gắng tìm ra lỗi ở các vấn đề sau: Chức năng không chính xác hoặc thiếu, lỗi giao diện, lỗi trong cấu trúc dữ liệu hoặc truy cập cơ sở bên ngoài, lỗi về hiệu suất …



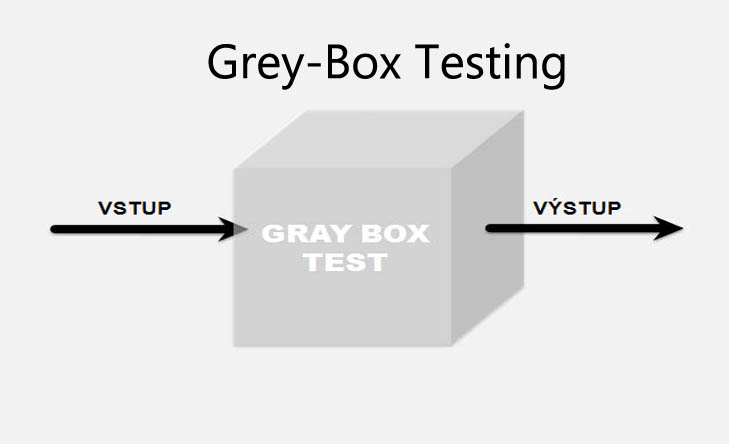
Hình 1.10: Kiểm thử hộp đen (Black box testing)

### 1.8.4 Gray box testing (Kiểm thử hộp xám)

- Kiểm thử hộp xám (Gray box testing) là phương pháp kiểm thử phần mềm kết hợp giữa kiểm thử hộp đen và kiểm thử hộp trắng. Hộp xám trong kiểm thử phần mềm giống như bán trong suốt, nhìn vào sẽ thấy lớp vỏ bên ngoài và một phần cấu trúc bên trong.

- Trong kiểm thử hộp xám, Tester có thể truy cập vào cấu trúc dữ liệu bên trong và thuật toán của chương trình với mục đích là để thiết kế test case, nhưng khi test thì test như là người dùng cuối hoặc là ở mức hộp đen.

- Ưu điểm của kiểm thử hộp xám là sử dụng kỹ thuật đơn giản của kiểm thử hộp đen kết nối với mã code hệ thống kiểm thử hộp trắng để tìm ra vấn đề của sản phẩm. Các tester được quyền truy cập vào cấu trúc dữ liệu bên trong cũng như các thuật toán được sử dụng trong sản phẩm.



Hình 1.11: Kiểm thử hộp xám (Gray box testing)

### 1.8.4 So sánh các phương pháp kiểm thử phần mềm

Bảng 1.1: So sánh các phương pháp kiểm thử phần mềm

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Kiểm thử hộp trắng** | **Kiểm thử hộp đen** | **Kiểm thử hộp xám** |
| **Hành động** | Kiểm tra thuật toán, cấu trúc code bên trong của sản phẩm. | Không quan tâm đến cấu trúc code bên trong, tập trung tìm ra lỗi ở các vấn đề sau: Chức năng không chính xác hoặc thiếu, lỗi giao diện, lỗi trong cấu trúc dữ liệu hoặc truy cập cơ sở bên ngoài, lỗi về hiệu suất … | Tìm kiếm và xác định các khiếm khuyết do cấu trúc mã không phù hợp hoặc sử dụng ứng dụng không đúng cách. |
| **Đối tượng kiểm tra** | Mã code, câu lệnh | Giao diện, chức năng, hiệu suất … | Cấu trúc mã, cách sử dụng ứng dụng |
| **Mục đích** | Đảm bảo tất cả các câu lệnh và điều kiện sẽ được thực hiện đúng, kết quả đầu ra như mong đợi. | Đảm bảo phần mềm hoạt động đúng như dự kiến và đáp ứng được sự mong đợi của khách hàng. | Đảm bảo cấu trúc mã của sản phẩm phù hợp và ứng dụng được sử dụng đúng cách đem lại hiệu quả. |
| **Nhân lực** | Thường là Developers | Tester (Không cần có nền tảng IT vẫn có thể test được) | Tester có nền tảng IT |
| **Các loại kiểm thử** | - Kiểm thử đơn vị (Unit Testing)  - Kiểm thử rò rỉ bộ nhớ ( Testing for Memory Leaks) | - Kiểm thử đơn vị (Unit Testing)  - Kiểm thử rò rỉ bộ nhớ ( Testing for Memory Leaks)  - Kiểm thử chức năng (Functional testing)  - Kiểm thử phi chức năng (Non-Functional testing)  - Kiểm thử hồi quy (Regression testing) | - Kiểm thử tích hợp (Integration Testing)  - Kiểm thử thâm nhập (Penetration Testing) |
| **Kỹ thuật** | Phân tích độ phủ mã (Coverage Testing)  - Bao phủ câu lệnh (Statement Coverage)  - Bao phủ nhánh (Branch Coverage)  - Bao phủ đường đi (Path Coverage) | - Phân vùng tương đương ( Equivalence partitioning)  - Phân tích giá trị biên (Boundary value analysis)  - Bảng quyết định (Decision table)  - Đoán lỗi (Error Guessing) | - Kiểm thử ma trận (Matrix testing)  - Kiểm tra hồi quy (Regression testing)  - Kiểm tra mảng trực giao hoặc OAT (Orthogonal array testing or OAT)  - Kiểm tra mẫu (Pattern testing) |

## 1.9 Các kỹ thuật thiết kế test case

### 1.9.1 Equivalence Class Partitioning (Phân vùng tương đương)

- Phân vùng lớp tương đương cho phép bạn phân chia tập hợp các điều kiện kiểm tra thành một phân vùng nên được coi là giống nhau.

- 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.

- Với các giá trị đầu vào chia thành các vùng tương đương:

* 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 trạng thái khác của hệ thống: sai, thiếu, không đúng,...

- Mục đích : Giảm đáng kể số lượng test case cần phải thiết kế vì với mỗi lớp tương đương ta chỉ cần test trên các phần tử đại diện.

- Thiết kế Test-case bằng phân lớp tương đương tiến hành theo 2 bước:

* Xác định các lớp tương đương
* Xác định các ca kiểm thử

- Nguyên tắc:

* 1 lớp các giá trị lớn hơn
* 1 lớp các giá trị nhỏ hơn
* n lớp các giá trị hợp lệ

*Ví dụ:* Thiết kế testcase cho ô text chỉ cho nhập số nguyên với độ dài ký tự thuộc [1-10] hoặc [20-30]

- Với yêu cầu trên ta có các vùng:

* nhỏ hơn 1 : vùng không hợp lệ
* [1-10] : vùng hợp lệ
* lớn hơn 10 và nhỏ hơn 20 : vùng không hợp lệ
* [20-30] : vùng hợp lệ
* lớn hơn 30: vùng không hợp lệ
* Nhập các ký tự không phải số nguyên : vùng không hợp lệ

=> Vì vậy ta có các case sau:

- Case hợp lệ:

* Nhập 5 ký tự
* Nhập 25 ký tự số

- Case không hợp lệ:

* Không nhập vào trường
* Nhập 15 ký tự
* Nhập số thập phân
* Nhập 35 kí tự
* Nhập ký tự chữ: Tiếng việt, Tiếng anh, Full-size, Half-size
* Nhập ký tự đặc biệt, space, kí tự Enter
* Nhập câu lệnh SQL injection, HTML, XSS

### 1.9.2 Boundary Value Analysis (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.

- Phân tích giá trị biên sẽ chọn các giá trị:

* Giá trị ngay dưới giá trị nhỏ nhất
* Giá trị nhỏ nhất
* Giá trị ngay trên giá trị nhỏ nhất
* Giá trị ngay dưới giá trị lớn nhất
* Giá trị lớn nhất
* Giá trị ngay trên giá trị lớn nhất

*Ví dụ:* Với ví dụ trên ta có các case test sau:

* Không nhập ký tự nào (Giá trị ngay dưới giá trị nhỏ nhất vùng 1)
* Nhập 1 ký tự (giá trị nhỏ nhất vùng 1)
* Nhập 2 ký tự (giá trị ngay trên giá trị nhỏ nhất vùng 1)
* Nhập 9 ký tự (giá trị ngay dưới giá trị lớn nhất vùng 1)
* Nhập 10 ký tự (giá trị lớn nhất vùng 1)
* Nhập 11 ký tự (giá trị ngay trên giá trị lớn nhất vùng 1)
* Nhập 19 ký tự (giá trị ngay dưới giá trị nhỏ nhất vùng 2)
* Nhập 20 ký tự (giá trị nhỏ nhất vùng 2)
* Nhập 21 ký tự (giá trị ngay trên giá trị nhỏ nhất vùng 2)
* Nhập 29 ký tự (giá trị ngay dưới giá trị lớn nhất vùng 2)
* Nhập 30 ký tự (giá trị lớn nhất vùng 2)
* Nhập 31 ký tự (giá trị ngay trên giá trị lớn nhất vùng 2)

=> Kết hợp kỹ thuật phân vùng tương đương với phân tích giá trị biên ta có các case test sau:

* Không nhập ký tự nào
* Nhập 1 ký tự
* Nhập 5 ký tự
* Nhập 10 ký tự
* Nhập 11 ký tự
* Nhập 19 ký tự
* Nhập 20 ký tự
* Nhập 21 ký tự
* Nhập 25 ký tự
* Nhập 30 ký tự
* Nhập 31 ký tự
* Nhập số thập phân
* Nhập ký tự chữ: Tiếng việt, Tiếng anh, Full-size, Half-size
* Nhập ký tự đặc biệt, space, kí tự Enter
* Nhập câu lệnh SQL injection, HTML, XSS

### 1.9.3 Decision Table (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.

*Ví dụ:* Nút Submit phải được enable nếu người dùng đã nhập tất cả các trường bắt buộc

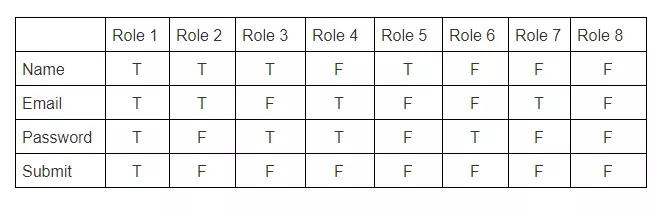
- Đầu tiên là xác định đấu ra của các chức năng có phụ thuộc vào sự kết hợp của các đầu vào. Nếu có tập hợp kết hợp đầu vào lớn, thì hãy chia nó thành các tập hợp nhỏ hơn hữu ích cho việc quản lý bảng quyết định.

- Đối với mọi chức năng, cần tạo một bảng và liệt kê tất cả các loại kết hợp đầu vào và đầu ra tương ứng. 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 cùng, ghi chú đầu ra so với kết hợp đầu vào.

*Ví dụ:* Nút Submit chỉ được enable khi tất cả các đầu vào được nhập bởi người dùng cuối



### 1.9.4 Đoán lỗi (Error Guessing)

- 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.

# TÌM HIỂU VỀ KIỂM THỬ TỰ ĐỘNG VỚI SELENIUM

## 2.1 Khái niệm về kiểm thử tự động

- Kiểm thử tự động: Là xử lý một cách tự động các bước thực hiện các testcase, kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử.

- Kiểm thử tự động: là một kỹ thuật tự động trong đó người kiểm thử tự viết các tập lệnh và sử dụng phần mềm phù hợp để kiểm thử phần mềm. Nó về cơ bản là một quá trình tự động hóa của một quy trình kiểm thử thủ công. Giống như kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để kiểm thử ứng dụng theo quan điểm tải, hiệu năng và ứng suất.

- Kiểm thử tự động giúp giảm chi phí kiểm thử bằng cách hỗ trợ quá trình kiểm thử thông qua các công cụ phần mềm.

- Kiểm thử tự động hay sử dụng phần mềm để kiểm thử với các ưu điểm:

* Có thể thực hiện các kiểm thử một cách liên tục, lặp lại và giảm chi phí cho nhân lực kiểm thử.
* Luôn đảm bảo hoạt động theo một kịch bản duy nhất – không bị ảnh hưởng như với kiểm thử viên.

## 2.2 Quy trình kiểm thử tự động

- Quy trình kiểm thử tự động bao gồm: tester sử dụng các kịch bản tự động (automation scripts) và thực thi các script để chạy ứng dụng với sự giúp sức của các automation tool. Một khi script đã sẵn sàng thì việc thực thi kiểm thử có thể diễn ra nhanh chóng và hiệu quả.

- Các hoạt động của kiểm thử tự động:

* Phân tích yêu cầu/Xác định môi trường/công cụ
* Xác định tiêu chí đầu ra
* Lên kế hoạch và kiểm soát
* Thiết lập môi trường kiểm thử
* Triển khai thiết kế kiểm thử
* Thực thi kiểm thử
* Phân tích, báo cáo

## 2.3 Mục đích và khi nào cần kiểm thử tự động?

- Kiểm thử tự động với các mục đích:

* Giảm bớt công sức và thời gian thực hiện quá trình kiểm thử
* Tăng độ tin cậy.
* Giảm sự nhàm chán cho con người
* Rèn luyện kỹ năng lập trình cho kiểm thử viên
* Giảm chi phí cho tổng quá trình kiểm thử

- Khi nào cần kiểm thử tự động:

* Không đủ tài nguyên: Khi số lượng test case quá nhiều mà kiểm thử viên không thể hoàn tất trong thời gian cụ thể.
* Kiểm tra hồi quy: Nâng cấp phần mềm, kiểm tra lại các tính năng đã chạy tốt và những tính năng đã sửa. Tuy nhiên, việc này khó đảm bảo về mặt thời gian.
* Kiểm tra khả năng vận hành phần mềm trong môi trường đặc biệt (Đo tốc độ trung bình xử lý một yêu cầu của Web server, xác định cấu hình máy thấp nhất mà phần mềm vẫn có thể hoạt động tốt).

## 2.4 Giới thiệu về Selenium

Selenium (thường được viết tắt là SE) là một công cụ kiểm thử phần mềm tự động, được phát triển bởi ThoughtWorks từ năm 2004 với tên ban đầu là JavaScript Test Runner. Đến năm 2007, tác giả Jason Huggins rời ThoughtWorks và gia nhập Selenium team, một phần của Google và phát triển thành Selenium như hiện nay.

Selenium là một công cụ hỗ trợ kiểm tra tự động cho các ứng dụng chạy trên nền web. Selenium hỗ trợ kiểm tra hầu hết trên các trình duyệt phổ biến hiện nay như Firefox, Internet Explorer, Safari,…cũng như các hệ điều hành chủ yếu như Windows, Linux, Mac,…

Selenium hỗ trợ một số lớn các ngôn ngữ lập trình như C#, Java, Perl, PHP, Python, Ruby,…

Selenium có thể kết hợp thêm một số công cụ khác như Bromien, Junit nhưng với người dùng thông thường chỉ cần chạy tự động mà không cần cài thêm các công cụ hỗ trợ.

Những tính năng của Selenium bao gồm:

- Selenium là một công cụ mã nguồn mở/ framework để kiểm tra web, website phiên bản di động.

- Selenium IDE hỗ trợ tính năng playback giúp bạn có thể sử dụng các bài test của người khác mà không cần phải biết ngôn ngữ lập trình nào.

- Selenium là một nền tảng kiểm thử dựa trên cloud giúp tester có thể lưu lại thao tác và xuất ra dưới dạng script đơn giản, dễ hiểu.

- Selenium hỗ trợ nhiều hệ điều hành, ngôn ngữ lập trình và trình duyệt khác nhau.

- Selenium có thể chạy cùng lúc nhiều bài test để giảm bớt thời gian và tăng hiệu quả trong công việc.

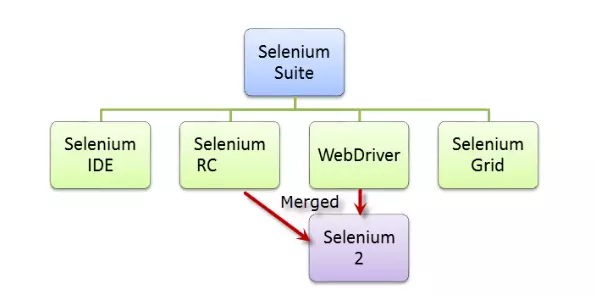
- Selenium có thể tích hợp với các framework khác như Ant và Maven để biên dịch mã nguồn.

- Quá trình của Selenium hao tốn ít tài nguyên và yêu cầu cấu hình thiết bị thấp hơn các công cụ khác.

- Test script của Selenium có thể trực tiếp tương tác với trình duyệt mà không yêu cầu cài đặt server.

- Selenium Remote Control kết hợp với WebDriver API để trở thành phiên bản Selenium 2.0 hỗ trợ những trang web động và Ajax.

Tính đến thời điểm hiện tại, Selenium không phải là một công cụ duy nhất, nó có tới 4 công cụ để bạn có thể tìm ra mục đích sử dụng và lựa chọn thích hợp.



Hình 2.1: Cấu trúc của Selenium

Bốn công cụ đó là:

*- Selenium IDE:* là một công cụ cho phép người dùng ghi lại một kịch bản kiểm thử và tái sử dụng lại chính kịch bản đó. Nó hoạt động như một Add-on của trình duyệt Mozilla Firefox với giao diện trực quan, dễ sử dụng ngay cả với những kiểm thử viên không biết về code. Selenium IDE cho phép người dùng tái sử dụng lại kịch bản kiểm thử trên nhiều trình duyệt khác nhau như internet explorer, google chrome, …

*- Selenium Remote Control (Selenium RC):* công cụ cho phép các nhà phát triển tự động hóa quá trình kiểm thử bằng cách sử dụng bất kỳ ngôn ngữ lập trình nào, phát huy tối đa thế mạnh của Selenium trong kiểm thử đơn vị. Để dễ dàng hơn cho việc kiểm thử, Selenium RC cung cấp các API và thư viện cho mỗi ngôn ngữ được hỗ trợ như HTML, Java, Perl, PHP, Ruby, Python, C#.

*- Selenium WebDriver:* là phiên bản kế nhiệm của Selenium RC. Do vậy nên Selenium WebDriver cũng hỗ trợ viết các kịch bản kiểm thử bằng các ngôn ngữ khác nhau như Java, Perl, PHP, Ruby, Python, C# và kiểm thử viên có thể dùng các điều kiện if-else hay các vòng lặp để tăng tính chính xác cho kịch bản kiểm thử. Selenium WebDriver có kiến trúc khá đơn giản, điều khiển trình duyệt trực tiếp từ hệ điều hành.

*- Selenium Grid:* là một hệ thống hỗ trợ kiểm thử viên thực thi kịch bản kiểm thử trên nhiều máy. Khi mới ra đời, Selenium Grid chỉ dùng hỗ trợ cho Selenium RC, sau này đội ngũ phát triển Selenium đã phát triển công cụ này cho cả Selenium WebDriver. Selenium Grid không chỉ thực thi kiểm thử trên nhiều máy với nhiều trình duyệt khác nhau, mà nó còn cho phép người dùng thực thi kiểm thử với chế độ phân tán.

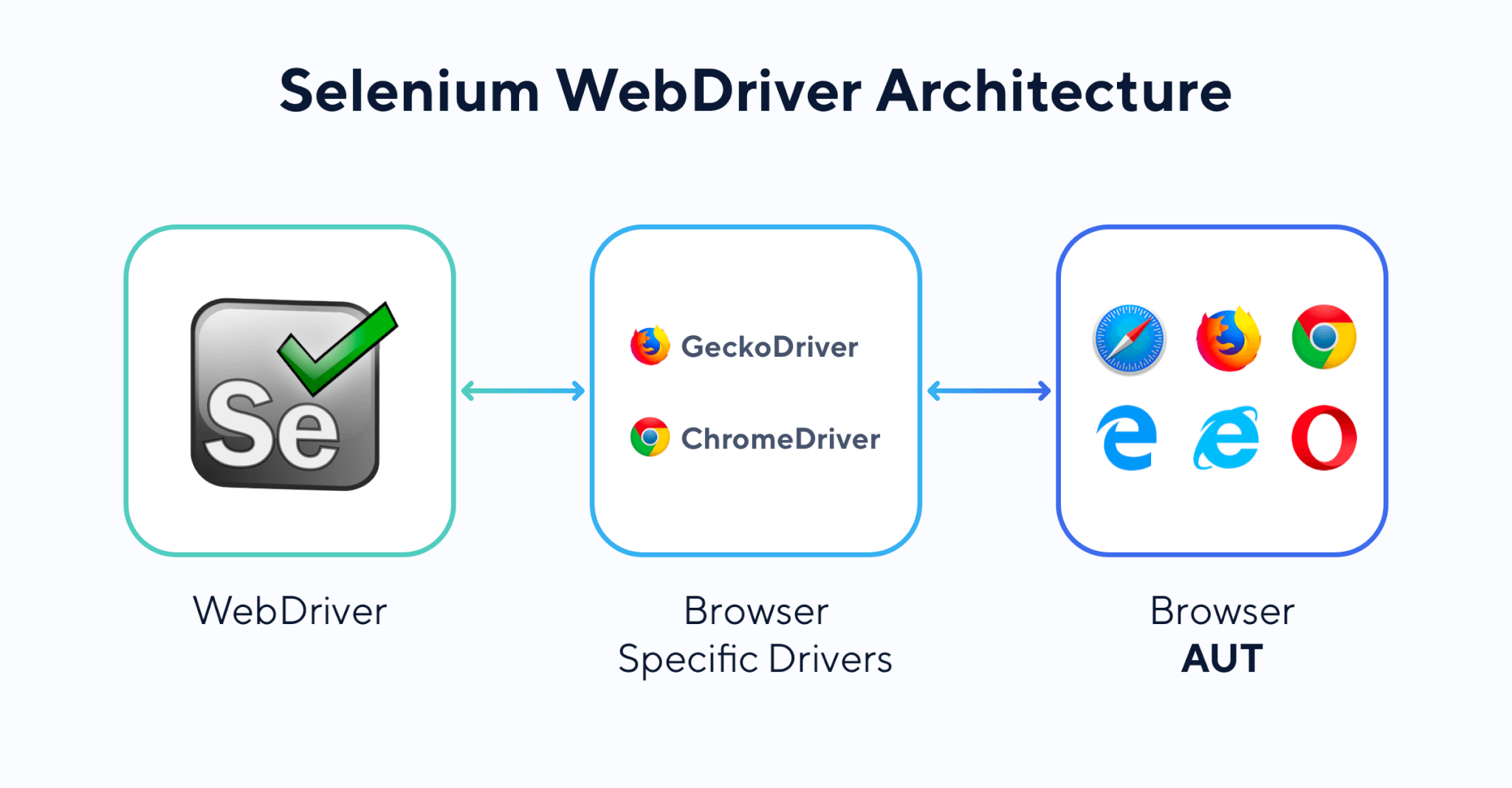
## 2.5 Selenium WebDriver

### 2.5.1 Khái niệm

Selenium WebDriver (gọi tắt là WebDriver) là một automation framework của web, cho phép thực thi các test của người dùng đối với các trình duyệt khác nhau không chỉ Firefox, Chrome.

Webdriver cho phép người dùng sử dụng các ngôn ngữ lập trình trong việc tạo test script của họ. Người dùng có thể sử dụng Conditional operations như If-else hay là Switch-case hoặc vòng lặp Do-while.

WebDriver hỗ trợ nhiều ngôn ngữ lập trình khác nhau như Java, .Net, PHP, Python, Perl, Ruby,... Không nhất thiết người dùng phải am hiểu tất cả các ngôn ngữ trên nhưng để sử dụng WebDriver một cách hiệu quả hơn, nên biết ít nhất một số trong những ngôn ngữ lập trình trên.



Hình 2.2: Selenium WebDriver

### 2.5.2 Tại sao nên dùng Selenium WebDriver

Ưu điểm khi dùng Selenium WebDriver đó là người dùng có thể sử dụng đến 7 ngôn ngữ lập trình (Java, C#, PHP, Ruby, Perl, Python và .Net), hỗ trợ thử nghiệm trên nhiều trình duyệt khác nhau (Firefox, Chrome, IE, Safari) và chạy trên nhiều hệ điều hành khác nhau (Window, Mac, Linux, Android, IOS). Khắc phục được hạn chế của Selenium 1.0 như tải lên tệp, tải xuống, cửa sổ bật lên và hộp thoại.

Với Selenium WebDriver, người dùng có thể dùng miễn phí, kiến trúc đơn giản chỉ với điều khiển trình duyệt từ hệ điều hành, IDE và trình duyệt là xong. Selenium WebDriver hỗ trợ viết kịch bản kiểm thử bằng các ngôn ngữ lập trình khác nhau, có thể sử dụng được các điều kiện, các vòng lặp để tăng tính chính xác cho kịch bản kiểm thử.

Về mặt tốc độ thì khi so sánh với các công cụ khác của Selenium, Selenium WebDriver là công cụ nhanh nhất trong số tất cả do tương tác trực tiếp từ hệ điều hành tới trình duyệt.

Hỗ trợ bộ câu lệnh đơn giản nhưng lại phổ biến, có thể dùng lặp đi lặp lại nhiều lần như:

- Driver.get(“URL"): điều hướng đến một trang web.

- Element.sendKeys(“text”): nhập văn bản vào input textbox.

- Element.clear(): xóa nội dung khỏi input textbox.

- Select.deselectAll(): bỏ chọn tất cả option từ select đầu tiên trên trang web.

- Select.selectByVisibleText(“text”): chọn option với đầu vào được chỉ định bởi người dùng.

- Driver.switchTo().window(“windowName”): di chuyển con trỏ chuột từ cửa số này sang cửa sổ khác.

- Driver.quit(): thoát driver và đóng tất cả các cửa sổ liên quan đến driver đó.

- Driver.switchTo().frame(“frameName”): di chuyển con trỏ chuột từ frame này sang frame khác.

- Driver.refresh(): tải lại trang hiện tại.

- Driver.navigation().back(): chuyển hướng về trang trước.

- Driver.navigation().to(“URL”): chuyển hướng tới URL.

- Driver.navigation().forward(): chuyển hướng đến trang tiếp theo.

- Driver.close(): đóng trình duyệt hiện tại và các liên kết đến với driver.

- Driver.switchTo().alert(): di chuyển con trỏ chuột tới alert.

Cách làm việc của Selenium WebDriver: để Selenium WebDriver có thể mô phỏng hành động như click chuột hay điền một giá trị, điều đầu tiên là phải xác định được vị trí của element (phần tử) trên trang web mà nó sẽ tương tác (dựa vào CSS Selector hoặc Xpath). Sau khi đã xác định vị trí thì tiếp theo Selenium WebDriver sẽ xác định hành động đối với phần tử kiểm thử đó (thông qua các API được viết sẵn trong các gói thư viện của Selenium WebDriver), điều này làm tăng độ chính xác của máy móc khi tạo ra hoặc mô phỏng lại các hành động của con người lên đối tượng kiểm thử.

## 2.6 Cách xác định locators

### 2.6.1 Cách xác định locators của element trên website

Định vị các phần tử trong Selenium WebDriver được thực hiện với sự trợ giúp của các phương thức findElement() và findElements() được cung cấp bởi lớp WebDriver và WebElement.

- findElement(): trả về một đối tượng WebElement.

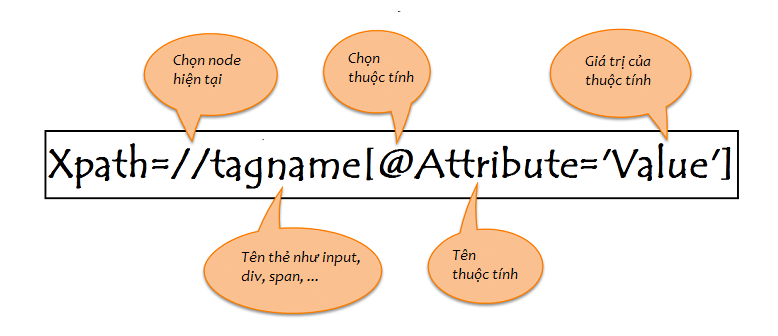
- findElements(): trả về một danh sách các đối tượng thuộc WebElement.

Bảng 2.1: Bảng sau liệt kê tất cả cú pháp Java để định vị các phần tử trong Selenium WebDriver

|  |  |  |
| --- | --- | --- |
| **Phương thức** | **Cú pháp** | **Mô tả** |
| Theo ID | driver.findElement(By.id (<element ID>)) | Định vị một phần tử bằng thuộc tính ID |
| Theo name | driver.findElement(By.name (<element name>)) | Định vị một phần tử bằng cách sử dụng thuộc tính Name |
| Theo class name | driver.findElement(By.className (<element class>)) | Định vị một phần tử bằng thuộc tính Class |
| Theo tag name | driver.findElement(By.tagName (<htmltagname>)) | Định vị một phần tử bằng cách sử dụng thẻ HTML |
| Bằng văn bản liên kết | driver.findElement(By.linkText (<linktext>)) | Tìm liên kết bằng văn bản liên kết |
| Bằng văn bản liên kết một phần | driver.findElement(By.partialLinkText (<linktext>)) | Định vị liên kết bằng văn bản một phần của liên kết |
| Theo CSS | driver.findElement(By.cssSelector (<css selector>)) | Định vị một phần tử bằng cách sử dụng bộ chọn CSS |
| Theo XPath | driver.findElement(By.xpath (<xpath>)) | Định vị một phần tử bằng cách sử dụng truy vấn XPath |

### 2.6.2 Cách lấy xpath của element trên website

Xpath được sử dụng để xác định các phần tử web dựa trên path XML của nó. Đây được xem là kỹ thuật hiệu quả nhất, có thể xác định tất cả các phần tử trên trang web thường là các element không định vị được bằng id, name, class. Tuy nhiên khi có sự thay đổi trong cấu trúc trang thì bạn sẽ cần phải update lại thông tin các element liên quan.



Hình 2.3: Xpath

Có 2 loại Xpath: Xpath tuyệt đối và Xpath tương đối.

- Xpath tuyệt đối:

* Xpath tuyệt đối bắt đầu bằng dấu gạch chéo đơn "/", cho phép xác định một đường dẫn tuyệt đối đến đối tượng UI.
* Xpath tuyệt đối được xem là là cách tìm kiếm phần tử dễ dàng nhất, tuy nhiên nhược điểm của nó là nếu có bất kỳ thay đổi nào trên đường dẫn của element thì Xpath lấy sẽ sai.

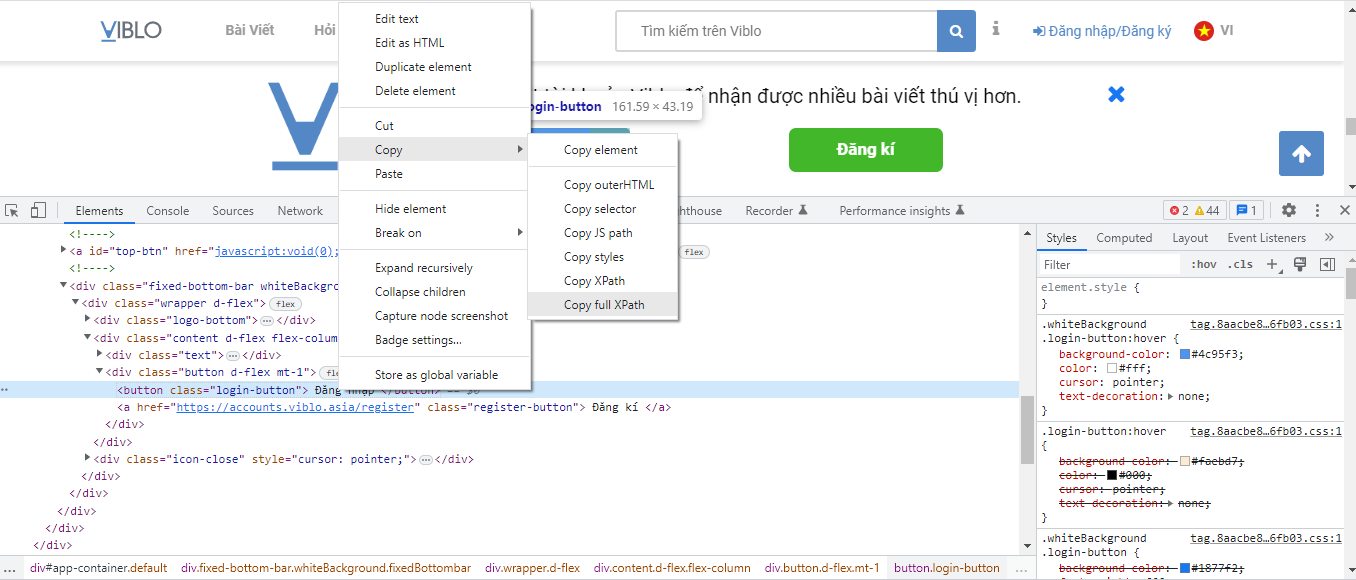
- Xpath tương đối: Xpath tương đối bắt đầu bằng 2 dấu gạch chéo "//", cho phép xác định một đối tượng UI ở bất kỳ đâu trên trang web, không cần bắt đầu bởi thẻ html trong đường dẫn.

Cách lấy Xpath từ Inspect của Chrome:

- Bước 1: Chạy và khởi động trình duyệt.

- Bước 2: Trỏ chuột vào vị trí muốn lấy. Sau đó, bấm chuột phải rồi chọn Inspect.

- Bước 3: Kiểm tra lại xem vị trí đã được xác định đúng chưa. Sau đó, bấm chuột phải rồi chọn Copy và chọn Copy Xpath.

****

Hình 2.4: Cách lấy Xpath

## 2.7 Ngôn ngữ lập trình Java

Java là một ngôn ngữ lập trình hướng đối tượng, mạnh mẽ và phổ biến, được biên dịch thành mã máy trước khi thực thi. Ngôn ngữ lập trình này được tạo ra với mục tiêu tạo ra một hệ thống linh hoạt, an toàn và hiệu quả cho việc phát triển phần mềm. Java được phát triển bởi James Gosling và nhóm của ông tại Sun Microsystems vào những năm đầu của thập kỷ 1990. Đến ngày nay, cộng đồng phát triển và sử dụng Java đã trở nên rất lớn mạnh, đặc biệt sau khi Sun Microsystems chính thức công bố mã nguồn mở của Java vào năm 2006.

Java là một ngôn ngữ lập trình mạnh mẽ, linh hoạt và phổ biến, có khả năng đáp ứng nhiều nhu cầu khác nhau trong quá trình phát triển phần mềm.

Các lý do chọn Java là ngôn ngữ để thực hiện kiểm thử tự động:

- Đa năng và phổ biến: Java là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. Nó được sử dụng rộng rãi trong nhiều lĩnh vực phát triển phần mềm, từ ứng dụng di động đến phần mềm máy tính và cả trong lĩnh vực kiểm thử tự động.

- Dễ học và sử dụng: Java có cú pháp rõ ràng và gần gũi với nhiều ngôn ngữ lập trình khác, làm cho việc học và sử dụng nó dễ dàng đối với nhiều lập trình viên.

- Thư viện và framework đa dạng: Java cung cấp nhiều thư viện và framework mạnh mẽ cho việc kiểm thử tự động như Selenium, TestNG, JUnit, Cucumber và nhiều công cụ khác. Những công cụ này giúp việc xây dựng và quản lý các kịch bản kiểm thử trở nên dễ dàng hơn.

- Hỗ trợ đa nền tảng: Java là một ngôn ngữ đa nền tảng, có thể chạy trên nhiều hệ điều hành khác nhau. Điều này giúp bạn tạo ra các kịch bản kiểm thử có khả năng chạy trên nhiều môi trường khác nhau mà không cần viết lại mã nguồn.

- Cộng đồng và tài liệu đa dạng: Với sự phổ biến của Java, cộng đồng lập trình viên rất lớn và tích cực. Điều này đồng nghĩa rằng bạn có thể tìm thấy rất nhiều tài liệu, hướng dẫn và giải pháp cho các vấn đề phát sinh trong quá trình kiểm thử tự động.

- Khả năng tích hợp và mở rộng: Java có khả năng tích hợp tốt với các công cụ và dịch vụ khác trong quy trình phát triển phần mềm. Bạn có thể kết hợp kiểm thử tự động với các công cụ quản lý mã nguồn, CI/CD, và các dịch vụ khác một cách dễ dàng.

- Tính ổn định và độ tin cậy: Java là một ngôn ngữ có tính ổn định và độ tin cậy cao, giúp việc xây dựng và duy trì các kịch bản kiểm thử được thực hiện một cách hiệu quả và bền vững theo thời gian.

Với Java unit test (còn được gọi là JUnit) là một nền tảng kiểm thử (testing framework) được phát triển dành cho Java có một số annotaion và assert statement trong unit test:

- @Test: Đánh dấu phương thức kiểm thử.

- @Before: Được thực thi trước khi thực thi mỗi function test, nó được sử dụng để chuẩn bị môi trường cho việc test (đọc, input data, khởi tạo class, ...).

- @After: Được thực thi sau mỗi function test, nó được sử dụng để làm sạch môi trường test.

- @BeforeClass: Được thực thi một lần trước khi bắt đầu thực thi tất cả các function test. Các phương thức được đánh dấu bằng annotaion này thì cần được xác định là static để làm việc với JUnit.

- @AfterClass: Được thực thi một lần, sau khi tất cả các function test kết thúc, nó được sử dụng để dọn dẹp một số thứ đã chuẩn bị trước đó cho việc test (connect database, ...). Các phương thức được đánh dấu bằng annotaion này thì cần được xác định là static để làm việc với JUnit.

- @Ignore or @Ignore("Why disabled"): Đánh dấu test đó là disabled. Nó hữu ích trong trường hợp code thay đổi mà function test vẫn chưa được điều chỉnh.

- @Test (expected = Exception.class): Failed nếu method không ném ra được tên một exception.

- @Test(timeout=100): Failed nếu phương thức kiểm thử đó chạy quá 100 ms.

- assertTrue(message, boolean condition): Đánh giá biểu thức luận lý. Test sẽ được chấp nhận nếu biểu thức đó đúng.

- assertFalse(message, boolean condition): Đánh giá biểu thức luận lý. Test sẽ được chấp nhận nếu biểu thức đó sai.

- assertNull(message, object): So sánh tham chiếu của một đối tượng với giá trị null. Test sẽ được chấp nhận nếu tham chiếu đối tượng đó là null.

- assertNotNull(message, object): So sánh tham chiếu của một đối tượng với null. Test sẽ được chấp nhận nếu tham chiếu đối tượng đó khác null.

- assertSame(message,expected, actual): So sánh địa chỉ vùng nhớ của 2 tham chiếu đối tượng bằng cách sử dụng toán tử ==. Test sẽ được chấp nhận nếu cả 2 đều tham chiếu đến cùng một đối tượng.

- assertNotSame(message,expected, actual): So sánh địa chỉ vùng nhớ của 2 tham chiếu đối tượng bằng cách sử dụng toán tử ==. Test sẽ được chấp nhận nếu cả 2 đều tham chiếu đến các đối tượng khác nhau.

- assertThat(): So sánh một giá trị thực tế có thỏa mãn với 1 Matcher được xác định hay không. Với matchers có thể kiểm tra kết quả của một string, number, collections …

- fail(): Phương thức này làm cho test hiện hành thất bại, phương thức này thường được sử dụng khi xử lý các ngoại lệ. Mặc dù chúng ta có thể chỉ cần sử dụng phương thức assertTrue() cho gần như hầu hết các test case, tuy nhiên thì việc sử dụng một trong các phương thức assertXXX() cụ thể sẽ làm cho các test dễ hiểu hơn và cung cấp các thông điệp khi một test bị fail rõ ràng hơn.

Mặc dù cả Java và Python đều có thể được sử dụng để thực hiện kiểm thử tự động với Selenium, nhưng có một số điểm mà Java có thể được coi là có lợi hơn so với Python trong môi trường kiểm thử với Selenium:

- Hiệu năng: Java thường có hiệu năng tốt hơn so với Python trong nhiều trường hợp. Điều này đặc biệt quan trọng khi bạn cần thực hiện các kịch bản kiểm thử phức tạp và yêu cầu xử lý dữ liệu lớn.

- Đa luồng và đồng bộ hóa: Java hỗ trợ quản lý đa luồng và đồng bộ hóa tốt hơn Python. Trong kiểm thử tự động, việc thao tác đồng thời trên nhiều thành phần của ứng dụng thường cần sự quản lý đa luồng và đồng bộ hóa.

- Khả năng xử lý lỗi: Java có hệ thống xử lý lỗi và quản lý ngoại lệ (exception handling) mạnh mẽ hơn Python. Điều này giúp bạn quản lý các tình huống bất thường trong quá trình thực hiện kiểm thử một cách dễ dàng hơn.

- Phong cách lập trình chặt chẽ: Java thường thúc đẩy phong cách lập trình chặt chẽ hơn, với việc kiểm tra kiểu dữ liệu tại thời điểm biên dịch. Điều này có thể giúp bạn phát hiện lỗi trước khi thực thi chương trình.

- Hệ thống kiểu dữ liệu mạnh mẽ: Java có hệ thống kiểu dữ liệu mạnh mẽ hơn Python, điều này có thể giúp bạn dễ dàng quản lý dữ liệu và tránh những sai sót phát sinh từ kiểu dữ liệu không rõ ràng.

- Thư viện kiểm thử phong phú: Java có nhiều thư viện và framework kiểm thử mạnh mẽ như TestNG và JUnit, cùng với Selenium, giúp bạn xây dựng và quản lý các kịch bản kiểm thử một cách chuyên nghiệp.

# THỰC NGHIỆM VÀ ĐÁNH GIÁ

## 3.1 Giới thiệu về website kiểm thử

### 3.1.1 Mô tả bài toán

Website Functional Food Store do bạn Lê Quang Trường, đã xây dựng lên là một trang web cho phép người dùng có thể mua sắm online với các mặt hàng thực phẩm chức năng như Người dùng có thể đăng ký để tạo tài khoản cá nhân, từ đó đăng nhập vào website và tiến hành mua hàng online.

Giao diện trang chủ website

Khi tìm hiểu về các mặt hàng trên website, người dùng có thể đặt hàng ngay trên website. Sau khi đặt hàng thành công, người dùng có thể vào mục Đơn hàng của tôi để tiện theo dõi tình trạng của đơn hàng.

Các tính năng của website Functional Food Store bao gồm:

- Đăng ký

- Đăng nhập

- Đăng xuất

- Tìm kiếm sản phẩm

- Xem thông tin chi tiết sản phẩm

- Gửi thông tin liên hệ

- Cập nhật thông tin cá nhân

- Thêm vào giỏ hàng

- Đặt hàng

- Xem lịch sử đặt hàng

Website hiện đang được phát triển và các tính năng vẫn đang được hoàn thiện qua từng giai đoạn phát triển. Đây cũng là website mà đồ án này sẽ lấy làm mẫu demo cho mục tiêu nghiên cứu - kiểm thử tự động với công cụ Selenium cho website.

### 3.1.2 Đặc tả use case

Bảng 3.1: Đặc tả use case “Đăng ký”

|  |  |
| --- | --- |
| **Use case** | Đăng ký |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện đăng ký tài khoản |
| **Tiền điều kiện** | Khách hàng truy cập vào hệ thống |
| **Hậu điều kiện** | Khách hàng đăng ký tài khoản thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn Đăng ký | 2. Hiển thị trang Đăng ký | | 3. Nhập số điện thoại và nhấn nút Đăng ký | 4. Kiểm tra thông tin đã nhập. Nếu hợp lệ, hiển thị trang Cập nhật thông tin cá nhân | | 5. Nhập thông tin cá nhân và nhấn nút Cập nhật thông tin | 6. Kiểm tra thông tin đã nhập. Nếu hợp lệ, lưu lại thông tin vào CSDL và hiển thị thông báo “Thông tin tài khoản được cập nhật thành công” rồi đăng nhập luôn vào hệ thống | |
| **Ngoại lệ** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 3.1 Không nhập số điện thoại và nhấn nút Đăng ký | 3.1.1 Hiển thị thông báo “Bạn chưa nhập số điện thoại” | | 3.2 Nhập số điện thoại đã đăng ký và nhấn nút Đăng ký | 3.2.1 Hiển thị thông báo “Số điện thoại này đã đăng kí” | | 3.3 Nhập số điện thoại không hợp lệ và nhấn nút Đăng ký | 3.3.1 Hiển thị thông báo “Số điện thoại này không hợp lệ” | | 5.1 Không nhập tên và nhấn nút Cập nhật thông tin | 5.1.1 Hiển thị thông báo “Bạn chưa nhập tên” | | 5.2 Nhập tên vượt quá 50 ký tự và nhấn nút Cập nhật thông tin | 5.2.1 Hiển thị thông báo “Họ tên vượt quá 50 kí tự” | | 5.3 Không nhập ngày sinh và nhấn nút Cập nhật thông tin | 5.3.1 Hiển thị thông báo “Bạn chưa nhập ngày sinh” | | 5.4 Nhập ngày sinh không hợp lệ và nhấn nút Cập nhật thông tin | 5.4.1 Hiển thị thông báo “Ngày sinh không hợp lệ” | | 5.5 Không chọn giới tính và nhấn nút Cập nhật thông tin | 5.5.1 Hiển thị thông báo “Bạn chưa chọn giới tính” | | 5.6 Không nhập địa chỉ và nhấn nút Cập nhật thông tin | 5.6.1 Hiển thị thông báo “Bạn chưa nhập địa chỉ” | | 5.7 Không nhập mật khẩu và nhấn nút Cập nhật thông tin | 5.7.1 Hiển thị thông báo “Bạn chưa nhập mật khẩu” | | 5.8 Nhập mật khẩu khác ký tự chữ và số và nhấn nút Cập nhật thông tin | 5.8.1 Hiển thị thông báo “Mật khẩu không đúng định dạng” | | 5.9 Nhập mật khẩu vượt quá 16 ký tự và nhấn nút Cập nhật thông tin | 5.9.1 Hiển thị thông báo “Mật khẩu tối đa 16 kí tự” | | 5.10 Nhập mật khẩu nhỏ hơn 8 ký tự và nhấn nút Cập nhật thông tin | 5.10.1 Hiển thị thông báo “Mật khẩu cần ít nhất 8 kí tự” | | 5.11 Không nhập lại mật khẩu và nhấn nút Cập nhật thông tin | 5.11.1 Hiển thị thông báo “Bạn chưa nhập lại mật khẩu” | | 5.12 Nhập mật khẩu nhập lại khác ký tự chữ và số và nhấn nút Cập nhật thông tin | 5.12.1 Hiển thị thông báo “Mật khẩu nhập lại không đúng định dạng” | | 5.13 Nhập mật khẩu nhập lại nhỏ hơn 8 ký tự và nhấn nút Cập nhật thông tin | 5.13.1 Hiển thị thông báo “Mật khẩu nhập lại cần ít nhất 8 kí tự” | | 5.14 Nhập mật khẩu nhập lại vượt quá 16 ký tự và nhấn nút Cập nhật thông tin | 5.14.1 Hiển thị thông báo “Mật khẩu nhập lại có tối đa 16 kí tự” | | 5.15 Không nhập địa chỉ và nhấn nút Cập nhật thông tin | 5.15.1 Hiển thị thông báo “Bạn chưa nhập địa chỉ” | | 5.16 Nhập mật khẩu với mật khẩu nhập lại không khớp và nhấn nút Cập nhật thông tin | 5.16.1 Hiển thị thông báo “Mật khẩu không khớp” | |

Bảng 3.2: Đặc tả use case “Đăng nhập”

|  |  |
| --- | --- |
| **Use case** | Đăng nhập |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện đăng nhập vào hệ thống |
| **Tiền điều kiện** | Khách hàng truy cập vào hệ thống |
| **Hậu điều kiện** | Khách hàng đăng nhập vào hệ thống thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn Đăng nhập | 2. Hiển thị trang Đăng nhập | | 3. Nhập thông tin đăng nhập và nhấn nút Đăng nhập | 6. Kiểm tra thông tin đã nhập. Nếu hợp lệ, đăng nhập vào hệ thống và hiển thị Trang chủ | |
| **Ngoại lệ** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 3.1 Không nhập số điện thoại và nhấn nút Đăng nhập | 3.1.1 Hiển thị thông báo “Bạn chưa nhập số điện thoại” | | 3.2 Nhập số điện thoại không hợp lệ và nhấn nút Đăng nhập | 3.2.1 Hiển thị thông báo “Số điện thoại này không hợp lệ” | | 3.3 Không nhập mật khẩu và nhấn nút Đăng nhập | 3.3.1 Hiển thị thông báo “Bạn chưa nhập mật khẩu” | | 3.4 Nhập thông tin không chính xác và nhấn nút Đăng nhập | 3.4.1 Hiển thị thông báo “Thông tin đăng nhập không chính xác” | |

Bảng 3.3: Đặc tả use case “Tìm kiếm sản phẩm”

|  |  |
| --- | --- |
| **Use case** | Tìm kiếm sản phẩm |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện tìm kiếm sản phẩm |
| **Tiền điều kiện** | Khách hàng truy cập vào hệ thống |
| **Hậu điều kiện** | Khách hàng tìm kiếm sản phẩm thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhập từ khóa và nhấn nút Tìm kiếm | 2. Tìm kiếm sản phẩm theo từ khóa và hiển thị danh sách sản phẩm tìm thấy | |
| **Ngoại lệ** | Không có |

Bảng 3.4: Đặc tả use case “Xem thông tin chi tiết sản phẩm”

|  |  |
| --- | --- |
| **Use case** | Xem thông tin chi tiết sản phẩm |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện xem thông tin chi tiết sản phẩm |
| **Tiền điều kiện** | Khách hàng truy cập vào hệ thống |
| **Hậu điều kiện** | Khách hàng xem thông tin chi tiết sản phẩm thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn sản phẩm bất kỳ | 2. Hiển thị trang Thông tin chi tiết sản phẩm | |
| **Ngoại lệ** | Không có |

Bảng 3.5: Đặc tả use case “Gửi thông tin liên hệ”

|  |  |
| --- | --- |
| **Use case** | Gửi thông tin liên hê |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện gửi thông tin liên hệ |
| **Tiền điều kiện** | Khách hàng truy cập vào hệ thống |
| **Hậu điều kiện** | Khách hàng gửi thông tin liên hệ thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn Liên hệ | 2. Hiển thị trang Liên hệ | | 3. Nhập thông tin liên hệ và nhấn nút Gửi | 4. Kiểm tra thông tin đã nhập. Nếu hợp lệ, lưu lại thông tin vào CSDL và hiển thị thông báo “Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công” | |
| **Ngoại lệ** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 3.1 Không nhập tên và nhấn nút Gửi | 3.1.1 Hiển thị thông báo “Bạn chưa nhập tên” | | 3.2 Nhập tên vượt quá 50 ký tự và nhấn nút Gửi | 3.2.1 Hiển thị thông báo “Họ tên vượt quá 50 kí tự” | | 3.3 Không nhập email và nhấn nút Gửi | 3.3.1 Hiển thị thông báo “Bạn chưa nhập email” | | 3.4 Nhập email không hợp lệ và nhấn nút Gửi | 3.4.1 Hiển thị thông báo “Email không đúng định dạng” | | 3.5 Không nhập nội dung và nhấn nút Gửi | 3.5.1 Hiển thị thông báo “Bạn chưa nhập nội dung” | |

Bảng 3.6: Đặc tả use case “Cập nhật thông tin cá nhân”

|  |  |
| --- | --- |
| **Use case** | Cập nhật thông tin cá nhân |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện cập nhật thông tin cá nhân |
| **Tiền điều kiện** | Khách hàng đã đăng nhập vào hệ thống |
| **Hậu điều kiện** | Khách hàng cập nhật thông tin cá nhân thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn Tên đăng nhập | 2. Hiển thị menu sổ xuống có chứa Thông tin cá nhân | | 3. Nhấn chọn Thông tin cá nhân | 4. Hiển thị trang Cập nhật thông tin cá nhân | | 5. Nhập thông tin cá nhân và nhấn nút Lưu thay đổi | 6. Kiểm tra thông tin đã nhập. Nếu hợp lệ, lưu lại thông tin vào CSDL và hiển thị thông báo “Thông tin tài khoản được cập nhật thành công” | |
| **Ngoại lệ** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 5.1 Không nhập tên và nhấn nút Lưu thay đổi | 5.1.1 Hiển thị thông báo “Bạn chưa nhập tên” | | 5.2 Nhập tên vượt quá 50 ký tự và nhấn nút Lưu thay đổi | 5.2.1 Hiển thị thông báo “Họ tên vượt quá 50 kí tự” | | 5.3 Không nhập ngày sinh và nhấn nút Lưu thay đổi | 5.3.1 Hiển thị thông báo “Bạn chưa nhập thông tin ngày sinh” | | 5.4 Nhập ngày sinh không hợp lệ và nhấn nút Lưu thay đổi | 5.4.1 Hiển thị thông báo “Ngày sinh không hợp lệ” | | 5.5 Không chọn giới tính và nhấn nút Lưu thay đổi | 5.5.1 Hiển thị thông báo “Bạn chưa chọn giới tính” | | 5.6 Không nhập địa chỉ và nhấn nút Lưu thay đổi | 5.6.1 Hiển thị thông báo “Bạn chưa nhập địa chỉ” | |

Bảng 3.7: Đặc tả use case “Xem lịch sử đặt hàng”

|  |  |
| --- | --- |
| **Use case** | Xem lịch sử đặt hàng |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện xem lịch sử đặt hàng |
| **Tiền điều kiện** | Khách hàng truy cập vào hệ thống |
| **Hậu điều kiện** | Khách hàng xem lịch sử đặt hàng thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn Tên đăng nhập | 2. Hiển thị menu sổ xuống có chứa Lịch sử đặt hàng | | 3. Nhấn chọn Lịch sử đặt hàng | 4. Hiển thị trang Lịch sử đặt hàng | |
| **Ngoại lệ** | Không có |

Bảng 3.8:Đặc tả use case “Đăng xuất”

|  |  |
| --- | --- |
| **Use case** | Đăng xuất |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện đăng xuất khỏi hệ thống |
| **Tiền điều kiện** | Khách hàng đã đăng nhập vào hệ thống |
| **Hậu điều kiện** | Khách hàng đăng xuất khỏi hệ thống thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn Tên đăng nhập | 2. Hiển thị menu sổ xuống có chứa Đăng xuất | | 3. Nhấn chọn Đăng xuất | 4. Đăng xuất khỏi hệ thống và hiển thị Trang chủ | |
| **Ngoại lệ** | Không có |

Bảng 3.9: Đặc tả use case “Thêm vào giỏ hàng”

|  |  |
| --- | --- |
| **Use case** | Thêm vào giỏ hàng |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện thêm vào giỏ hàng |
| **Tiền điều kiện** | Khách hàng đã đăng nhập vào hệ thống |
| **Hậu điều kiện** | Khách hàng thêm vào giỏ hàng thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn sản phẩm bất kỳ | 2. Hiển thị trang Thông tin chi tiết sản phẩm | | 3. Nhập thông tin số lượng sản phẩm và nhấn nút Thêm vào giỏ hàng | 4. Kiểm tra thông tin đã nhập. Nếu hợp lệ, lưu lại thông tin vào CSDL và hiển thị thông báo “Sản phẩm của bạn đã được thêm vào giỏ hàng” | |
| **Ngoại lệ** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 3.1 Không nhập số lượng hoặc nhập số lượng không hợp lệ và nhấn nút Thêm vào giỏ hàng | 3.1.1 Hiển thị thông báo “Số lượng sản phẩm không hợp lệ” | |

Bảng 3.10: Đặc tả use case “Đặt hàng”

|  |  |
| --- | --- |
| **Use case** | Đặt hàng |
| **Tác nhân** | Khách hàng |
| **Mô tả** | Khách hàng thực hiện đặt hàng |
| **Tiền điều kiện** | Khách hàng đã đăng nhập vào hệ thống |
| **Hậu điều kiện** | Khách hàng đặt hàng thành công |
| **Luồng sự kiện chính** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 1. Nhấn chọn Giỏ hàng | 2. Hiển thị trang Giỏ hàng | | 3. Nhấn nút Tiến hàng đặt hàng | 4. Hiển thị trang Thông tin thanh toán | | 5. Nhập thông tin thanh toán và nhấn nút Đặt mua | 6. Kiểm tra thông tin đã nhập. Nếu hợp lệ, lưu lại thông tin vào CSDL và hiển thị thông báo “Đơn hàng của bạn đã được tiếp nhận” | |
| **Ngoại lệ** | |  |  | | --- | --- | | **Khách hàng** | **Hệ thống** | | 5.1 Không nhập địa chỉ và nhấn nút Lưu thay đổi | 5.1.1 Hiển thị thông báo “Bạn chưa nhập địa chỉ” | |

## 3.2 Thiết kế ca kiểm thử

Bảng 3.11: Bảng các ca kiểm thử cho “Đăng ký”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra nhập số điện thoại hợp lệ | 1. Nhập số điện thoại hợp lệ  2. Nhấn nút "Đăng ký" | 2. Hiển thị trang "Cập nhật thông tin cá nhân" |
| TEST\_02 | Kiểm tra không nhập số điện thoại | 1. Không nhập số điện thoại  2. Nhấn nút "Đăng ký" | 2. Hiển thị thông báo "Bạn chưa nhập số điện thoại" |
| TEST\_03 | Kiểm tra nhập số điện thoại đã đăng ký | 1. Nhập số điện thoại đã đăng ký  2. Nhấn nút "Đăng ký" | 2. Hiển thị thông báo "Số điện thoại này đã đăng kí" |
| TEST\_04 | Kiểm tra nhập số điện thoại có chứa ký tự chữ | 1. Nhập số điện thoại có chứa ký tự chữ  2. Nhấn nút "Đăng ký" | 2. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_05 | Kiểm tra nhập số điện thoại có chứa ký tự đặc biệt | 1. Nhập số điện thoại có chứa ký tự đặc biệt  2. Nhấn nút "Đăng ký" | 2. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_06 | Kiểm tra nhập số điện thoại có chứa ký tự khoảng trắng | 1. Nhập số điện thoại có chứa ký tự khoảng trắng  2. Nhấn nút "Đăng ký" | 2. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_07 | Kiểm tra nhập số điện thoại < 10 ký tự số | 1. Nhập số điện thoại có 9 ký tự số  2. Nhấn nút "Đăng ký" | 2. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_08 | Kiểm tra nhập số điện thoại = 10 ký tự số nhưng không đúng định dạng số điện thoại | 1. Nhập số điện thoại có 10 ký tự số nhưng không đúng định dạng số điện thoại  2. Nhấn nút "Đăng ký" | 2. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_09 | Kiểm tra nhập số điện thoại > 10 ký tự số | 1. Nhập số điện thoại có 11 ký tự số  2. Nhấn nút "Đăng ký" | 2. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_10 | Kiểm tra nhập thông tin các trường hợp lệ | 1. Nhập thông tin các trường hợp lệ  2. Nhấn nút "Cập nhật thông tin" | 2. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_11 | Kiểm tra không nhập tên | 1. Không nhập tên  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Bạn chưa nhập tên" |
| TEST\_12 | Kiểm tra nhập tên < 50 ký tự | 1. Nhập tên có 49 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_13 | Kiểm tra nhập tên = 50 ký tự | 1. Nhập tên có 50 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_14 | Kiểm tra nhập tên > 50 ký tự | 1. Nhập tên có 51 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Họ tên vượt quá 50 kí tự" |
| TEST\_15 | Kiểm tra không nhập ngày sinh | 1. Không nhập ngày sinh  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Bạn chưa nhập ngày sinh" |
| TEST\_16 | Kiểm tra nhập ngày sinh có chứa ký tự chữ | 1. Nhập ngày sinh có chứa ký tự chữ  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 1. Chặn các ký tự chữ  3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_17 | Kiểm tra nhập ngày sinh có chứa ký tự đặc biệt | 1. Nhập ngày sinh có chứa ký tự đặc biệt  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 1. Chặn các ký tự đặc biệt  3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_18 | Kiểm tra nhập ngày sinh < ngày hiện tại | 1. Nhập ngày sinh là ngày hiện tại - 1  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_19 | Kiểm tra nhập ngày sinh = ngày hiện tại | 1. Nhập ngày sinh là ngày hiện tại  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_20 | Kiểm tra nhập ngày sinh > ngày hiện tại | 1. Nhập ngày sinh là ngày hiện tại + 1  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Ngày sinh không hợp lệ" |
| TEST\_21 | Kiểm tra chọn giới tính khác giá trị mặc định | 1. Chọn giới tính khác giá trị mặc định  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_22 | Kiểm tra không nhập địa chỉ | 1. Không nhập địa chỉ  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Bạn chưa nhập địa chỉ" |
| TEST\_23 | Kiểm tra không nhập mật khẩu | 1. Không nhập mật khẩu  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Bạn chưa nhập mật khẩu" |
| TEST\_24 | Kiểm tra nhập mật khẩu có chứa ký tự đặc biệt | 1. Nhập mật khẩu có chứa ký tự đặc biệt  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu không đúng định dạng" |
| TEST\_25 | Kiểm tra nhập mật khẩu có chứa ký tự khoảng trắng | 1. Nhập mật khẩu có chứa ký tự khoảng trắng  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu không đúng định dạng" |
| TEST\_26 | Kiểm tra nhập mật khẩu < 8 ký tự | 1. Nhập mật khẩu có 7 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu cần ít nhất 8 kí tự" |
| TEST\_27 | Kiểm tra nhập mật khẩu = 8 ký tự | 1. Nhập mật khẩu có 8 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_28 | Kiểm tra nhập mật khẩu = 16 ký tự | 1. Nhập mật khẩu có 16 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_29 | Kiểm tra nhập mật khẩu > 16 ký tự | 1. Nhập mật khẩu có 17 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu tối đa 16 kí tự" |
| TEST\_30 | Kiểm tra không nhập mật khẩu nhập lại | 1. Không nhập mật khẩu nhập lại  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Bạn chưa nhập lại mật khẩu" |
| TEST\_31 | Kiểm tra nhập mật khẩu nhập lại có chứa ký tự đặc biệt | 1. Nhập mật khẩu nhập lại có chứa ký tự đặc biệt  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu nhập lại không đúng định dạng" |
| TEST\_32 | Kiểm tra nhập mật khẩu nhập lại có chứa ký tự khoảng trắng | 1. Nhập mật khẩu nhập lại có chứa ký tự khoảng trắng  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu nhập lại không đúng định dạng" |
| TEST\_33 | Kiểm tra nhập mật khẩu nhập lại < 8 ký tự | 1. Nhập mật khẩu nhập lại có 7 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu nhập lại cần ít nhất 8 kí tự" |
| TEST\_34 | Kiểm tra nhập mật khẩu nhập lại > 16 ký tự | 1. Nhập mật khẩu nhập lại có 17 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu nhập lại tối đa 16 kí tự" |
| TEST\_35 | Kiểm tra nhập mật khẩu với mật khẩu nhập lại không khớp | 1. Nhập mật khẩu với mật khẩu nhập lại không khớp  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Cập nhật thông tin" | 3. Hiển thị thông báo "Mật khẩu không khớp" |

Bảng 3.12: Bảng các ca kiểm thử cho “Đăng nhập”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra nhập thông tin các trường hợp lệ | 1. Nhập thông tin các trường hợp lệ  2. Nhấn nút "Đăng nhập" | 2.  - Đăng nhập vào hệ thống thành công  - Hiển thị "Trang chủ" |
| TEST\_02 | Kiểm tra không nhập số điện thoại | 1. Không nhập số điện thoại  2. Nhập mật khẩu chính xác  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Bạn chưa nhập số điện thoại" |
| TEST\_03 | Kiểm tra nhập số điện thoại chưa đăng ký | 1. Nhập số điện thoại chưa đăng ký  2. Nhập mật khẩu chính xác  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" |
| TEST\_04 | Kiểm tra nhập số điện thoại có chứa ký tự chữ | 1. Nhập số điện thoại có chứa ký tự chữ  2. Nhập mật khẩu chính xác  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_05 | Kiểm tra nhập số điện thoại có chứa ký tự đặc biệt | 1. Nhập số điện thoại có chứa ký tự đặc biệt  2. Nhập mật khẩu chính xác  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_06 | Kiểm tra nhập số điện thoại có chứa ký tự khoảng trắng | 1. Nhập số điện thoại có chứa ký tự khoảng trắng  2. Nhập mật khẩu chính xác  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_07 | Kiểm tra nhập số điện thoại < 10 ký tự số | 1. Nhập số điện thoại có 9 ký tự số  2. Nhập mật khẩu chính xác  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_08 | Kiểm tra nhập số điện thoại = 10 ký tự số nhưng không đúng định dạng số điện thoại | 1. Nhập số điện thoại có 10 ký tự số nhưng không đúng định dạng số điện thoại  2. Nhập mật khẩu chính xác  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_09 | Kiểm tra nhập số điện thoại > 10 ký tự số | 1. Nhập số điện thoại có 11 ký tự số  2. Nhập mật khẩu chính xác  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Số điện thoại này không hợp lệ" |
| TEST\_10 | Kiểm tra không nhập mật khẩu | 1. Không nhập mật khẩu  2. Nhập số điện thoại đã đăng ký  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Bạn chưa nhập mật khẩu" |
| TEST\_11 | Kiểm tra nhập mật khẩu có chứa ký tự đặc biệt | 1. Nhập mật khẩu có chứa ký tự đặc biệt  2. Nhập số điện thoại đã đăng ký  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Mật khẩu không đúng định dạng" |
| TEST\_12 | Kiểm tra nhập mật khẩu có chứa ký tự khoảng trắng | 1. Nhập mật khẩu có chứa ký tự khoảng trắng  2. Nhập số điện thoại đã đăng ký  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Mật khẩu không đúng định dạng" |
| TEST\_13 | Kiểm tra nhập mật khẩu < 8 ký tự | 1. Nhập mật khẩu có 7 ký tự  2. Nhập số điện thoại đã đăng ký  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Mật khẩu cần ít nhất 8 kí tự" |
| TEST\_14 | Kiểm tra nhập mật khẩu = 8 ký tự | 1. Nhập mật khẩu có 8 ký tự  2. Nhập số điện thoại đã đăng ký  3. Nhấn nút "Đăng nhập" | 2.  - Đăng nhập vào hệ thống thành công  - Hiển thị "Trang chủ" |
| TEST\_15 | Kiểm tra nhập mật khẩu = 16 ký tự | 1. Nhập mật khẩu có 16 ký tự  2. Nhập số điện thoại đã đăng ký  3. Nhấn nút "Đăng nhập" | 2.  - Đăng nhập vào hệ thống thành công  - Hiển thị "Trang chủ" |
| TEST\_16 | Kiểm tra nhập mật khẩu > 16 ký tự | 1. Nhập mật khẩu có 17 ký tự  2. Nhập số điện thoại đã đăng ký  3. Nhấn nút "Đăng nhập" | 3. Hiển thị thông báo "Mật khẩu tối đa 16 kí tự" |
| TEST\_17 | Kiểm tra nhập mật khẩu không chính xác | 1. Nhập số điện thoại đã đăng ký  2. Nhập mật khẩu không chính xác  3. Nhấn nút "Đăng nhập" | 2. Hiển thị thông báo "Thông tin đăng nhập không chính xác" |

Bảng 3.13: Bảng các ca kiểm thử cho “Tìm kiếm sản phẩm”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra không nhập từ khóa | 1. Không nhập từ khóa  2. Nhấn nút "Tìm kiếm" | 2. Hiển thị toàn bộ sản phẩm trong hệ thống |
| TEST\_02 | Kiểm tra tìm kiếm thành công | 1. Nhập tên sản phẩm có trong hệ thống  2. Nhấn nút "Tìm kiếm" | 2. Hiển thị toàn bộ sản phẩm có tên giống với từ khóa đã nhập |
| TEST\_03 | Kiểm tra tìm kiếm không thành công | 1. Nhập tên sản phẩm không có trong hệ thống  2. Nhấn nút "Tìm kiếm" | 2. Không hiển thị sản phẩm nào |
| TEST\_04 | Kiểm tra tìm kiếm không phân biệt chữ hoa với chữ thường | 1. Nhập tên sản phẩm có trong hệ thống nhưng viết hoa  2. Nhấn nút "Tìm kiếm" | 2. Hiển thị toàn bộ sản phẩm có tên giống với từ khóa đã nhập |
| TEST\_05 | Kiểm tra nhập toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng  2. Nhấn nút "Tìm kiếm" | 2. Hiển thị toàn bộ sản phẩm trong hệ thống |

Bảng 3.14: Bảng các ca kiểm thử cho “Xem thông tin chi tiết sản phẩm”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra xem thông tin chi tiết sản phẩm thành công | 1. Nhấn chọn sản phẩm bất kỳ | 2. Chuyển đến màn hình Thông tin chi tiết sản phẩm |

Bảng 3.15: Bảng các ca kiểm thử cho “Gửi liên hệ”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra nhập thông tin hợp lệ | 1. Nhập thông tin các trường hợp lệ  2. Nhấn nút "Gửi" | 2. Hiển thị thông báo "Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công" |
| TEST\_02 | Kiểm tra không nhập tên | 1. Không nhập tên  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Bạn chưa nhập tên" |
| TEST\_03 | Kiểm tra nhập tên < 50 ký tự | 1. Nhập tên có 49 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_04 | Kiểm tra nhập tên = 50 ký tự | 1. Nhập tên có 50 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_05 | Kiểm tra nhập tên > 50 ký tự | 1. Nhập tên có 51 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Họ tên vượt quá 50 kí tự" |
| TEST\_06 | Kiểm tra không nhập email | 1. Không nhập email  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Bạn chưa nhập email" |
| TEST\_07 | Kiểm tra nhập email không đúng định dạng | 1. Nhập email không đúng định dạng  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Email không đúng định dạng" |
| TEST\_08 | Kiểm tra không nhập nội dung | 1. Không nhập nội dung  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Bạn chưa nhập nội dung" |
| TEST\_09 | Kiểm tra nhập nội dung trên một dòng | 1. Nhập nội dung trên một dòng  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công" |
| TEST\_10 | Kiểm tra nhập nội dung trên nhiều dòng | 1. Nhập nội dung trên nhiều dòng  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Gửi" | 3. Hiển thị thông báo "Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công" |

Bảng 3.16: Bảng các ca kiểm thử cho “Cập nhật thông tin cá nhân”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra nhập thông tin các trường hợp lệ | 1. Nhập thông tin các trường hợp lệ  2. Nhấn nút "Lưu thay đổi" | 2. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_02 | Kiểm tra không nhập tên | 1. Không nhập tên  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Bạn chưa nhập tên" |
| TEST\_03 | Kiểm tra nhập tên < 50 ký tự | 1. Nhập tên có 49 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_04 | Kiểm tra nhập tên = 50 ký tự | 1. Nhập tên có 50 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_05 | Kiểm tra nhập tên > 50 ký tự | 1. Nhập tên có 51 ký tự  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Họ tên vượt quá 50 kí tự" |
| TEST\_06 | Kiểm tra không nhập ngày sinh | 1. Không nhập ngày sinh  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Bạn chưa nhập ngày sinh" |
| TEST\_07 | Kiểm tra nhập ngày sinh có chứa ký tự chữ | 1. Nhập ngày sinh có chứa ký tự chữ  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 1. Chặn các ký tự chữ  3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_08 | Kiểm tra nhập ngày sinh có chứa ký tự đặc biệt | 1. Nhập ngày sinh có chứa ký tự đặc biệt  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 1. Chặn các ký tự đặc biệt  3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_09 | Kiểm tra nhập ngày sinh < ngày hiện tại | 1. Nhập ngày sinh là ngày hiện tại - 1  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_10 | Kiểm tra nhập ngày sinh = ngày hiện tại | 1. Nhập ngày sinh là ngày hiện tại  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_11 | Kiểm tra nhập ngày sinh < ngày hiện tại | 1. Nhập ngày sinh là ngày hiện tại + 1  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Ngày sinh không hợp lệ" |
| TEST\_12 | Kiểm tra chọn giới tính khác giá trị đã chọn | 1. Chọn giới tính khác giá trị đã chọn  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" |
| TEST\_13 | Kiểm tra không nhập địa chỉ | 1. Không nhập địa chỉ  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Lưu thay đổi" | 3. Hiển thị thông báo "Bạn chưa nhập địa chỉ" |

Bảng 3.17: Bảng các ca kiểm thử cho “Xem lịch sử đặt hàng”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra xem lịch sử đặt hàng thành công | 1. Nhấn chọn "Tên đăng nhập"  2. Nhấn chọn "Lịch sử đặt hàng" | 2. Hiển thị trang "Lịch sử đặt hàng" |

Bảng 3.18: Bảng các ca kiểm thử cho “Đăng xuất”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra đăng xuất thành công | 1. Nhấn chọn "Tên đăng nhập"  2. Nhấn chọn "Đăng xuất" | 2.  - Đăng xuất khỏi hệ thống thành công  - Hiển thị "Trang chủ" |

Bảng 3.19: Bảng các ca kiểm thử cho “Thêm vào giỏ hàng”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra với số lượng mặc định | 1. Nhấn nút "Thêm vào giỏ hàng" | 1. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" |
| TEST\_02 | Kiểm tra nhập số lượng hợp lệ | 1. Nhập số lượng hợp lệ  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" |
| TEST\_03 | Kiểm tra nhập số lượng chứa ký tự chữ | 1. Nhập số lượng chứa ký tự chữ  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" |
| TEST\_04 | Kiểm tra nhập số lượng chứa ký tự đặc biệt | 1. Nhập số lượng chứa ký tự đặc biệt  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" |
| TEST\_05 | Kiểm tra nhập số lượng chứa ký tự khoảng trắng | 1. Nhập số lượng chứa ký tự khoảng trắng  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" |
| TEST\_06 | Kiểm tra với số lượng trống | 1. Xóa số lượng  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" |
| TEST\_07 | Kiểm tra nhập số lượng là toàn ký tự khoảng trắng | 1. Nhập số lượng là toàn ký tự khoảng trắng  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" |
| TEST\_08 | Kiểm tra nhập số lượng là số âm | 1. Nhập số lượng là số âm  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" |
| TEST\_09 | Kiểm tra nhập số lượng bằng 0 | 1. Nhập số lượng bằng 0  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" |
| TEST\_10 | Kiểm tra nhập số lượng có dấu "+" ở đầu | 1. Nhập số lượng có dấu "+" ở đầu  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" |
| TEST\_11 | Kiểm tra nhập số lượng là số thập phân | 1. Nhập số lượng là số thập phân  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" |
| TEST\_12 | Kiểm tra nhấn nút "+" | 1. Nhấn nút "+"  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" |
| TEST\_13 | Kiểm tra nhấn nút "-" | 1. Nhấn nút "-"  2. Nhấn nút "Thêm vào giỏ hàng" | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" |

Bảng 3.20: Bảng các ca kiểm thử cho “Đặt hàng”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| TEST\_01 | Kiểm tra nhập thông tin các trường hợp lệ | 1. Nhập thông tin các trường hợp lệ  2. Nhấn nút "Đặt mua" | 2. Hiển thị thông báo "Đơn hàng của bạn đã được tiếp nhận" |
| TEST\_02 | Kiểm tra không nhập địa chỉ | 1. Không nhập địa chỉ  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Đặt mua" | 3. Hiển thị thông báo "Bạn chưa nhập địa chỉ" |
| TEST\_03 | Kiểm tra không nhập ghi chú | 1. Không nhập ghi chú  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Đặt mua" | 3. Hiển thị thông báo "Đơn hàng của bạn đã được tiếp nhận" |
| TEST\_04 | Kiểm tra nhập ghi chú trên một dòng | 1. Nhập ghi chú trên một dòng  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Đặt mua" | 3. Hiển thị thông báo "Đơn hàng của bạn đã được tiếp nhận" |
| TEST\_05 | Kiểm tra nhập ghi chú trên nhiều dòng | 1. Nhập ghi chú trên nhiều dòng  2. Nhập thông tin các trường khác hợp lệ  3. Nhấn nút "Đặt mua" | 3. Hiển thị thông báo "Đơn hàng của bạn đã được tiếp nhận" |

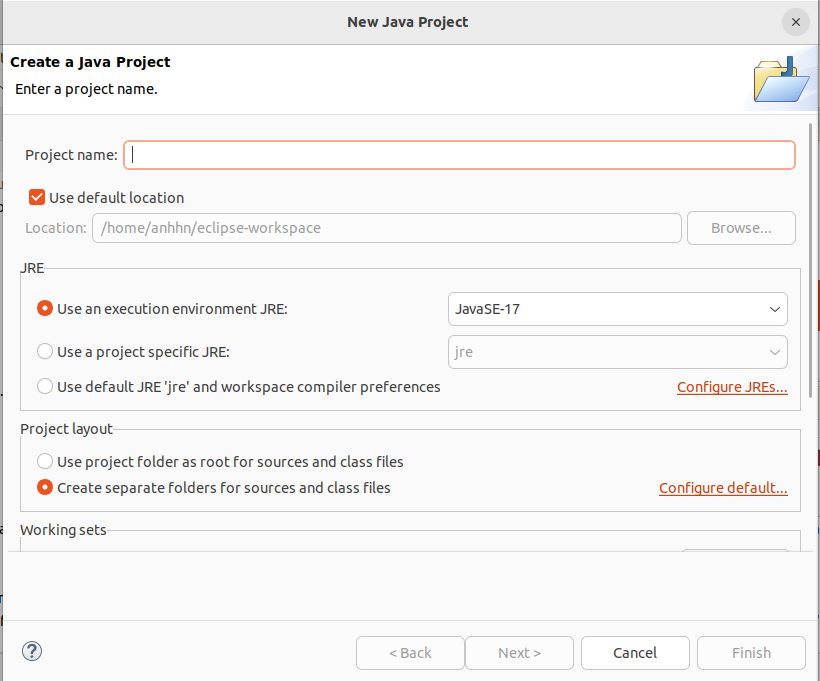
## 3.3 Xây dựng test script

Để xây dựng test script, ban đầu người dùng cần phải tải xuống bản webdriver tương thích với mỗi browser để có thể chạy được các auto testscript. Download công cụ Eclipse để thực hiện xây dựng và thực thi câu lệnh (tùy vào nhu cầu mỗi người dùng, có thể viết và thực thi câu lệnh trên các công cụ khác nhau có hỗ trợ ngôn ngữ và thư viện java).

Với mỗi browser sẽ có một bản webdriver riêng như với Chrome - ChromeDriver, với Firefox - GeckoDriver, …

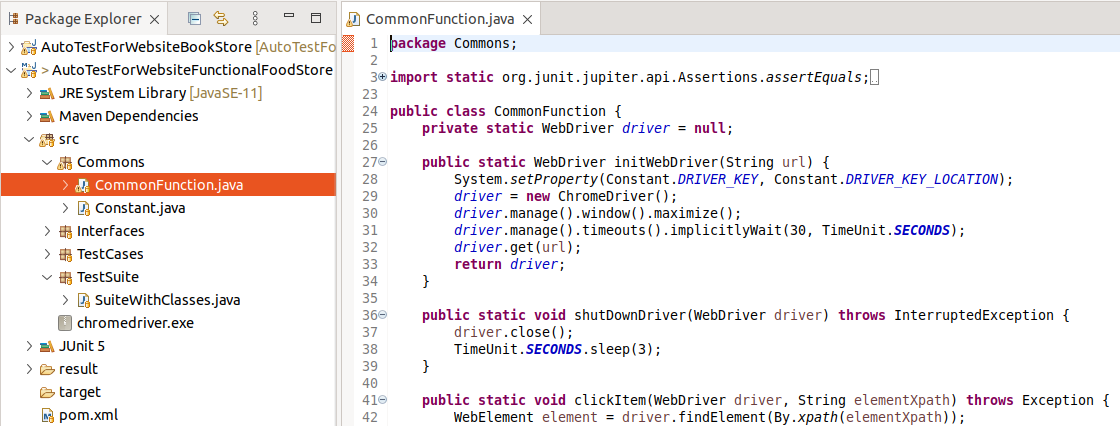
Các bước triển khai:

- Bước 1: Tạo mới một project: chọn File → New → Java Project → Nhập tên project → Nhấn Finish



Hình 3.1: Tạo project

- Bước 2: Tạo package Commons để lưu trữ các biến và phương thức chung



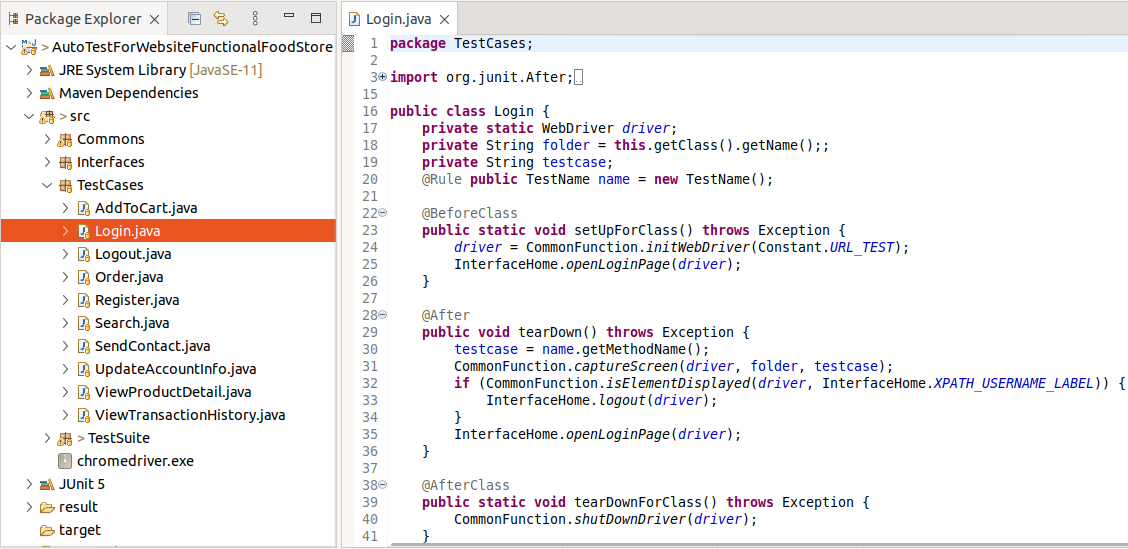
Hình 3.2: Tạo file chức các biến và phương thức chung

- Bước 3: Tạo package Interfaces để lưu trữ các đường dẫn xpath và các hàm thao tác kiểm tra website



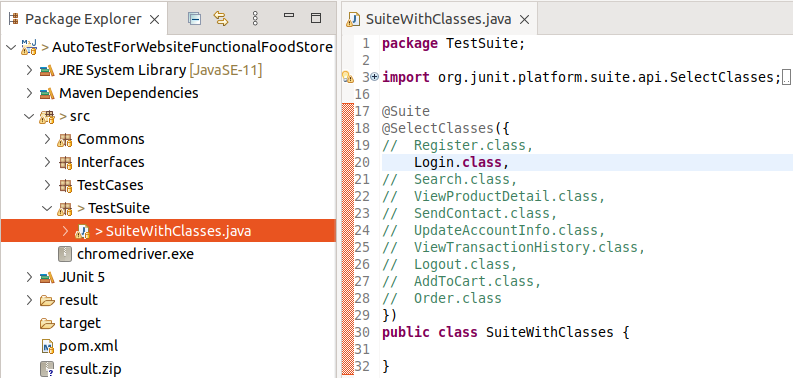
Hình 3.3: Tạo file chứa các đường dẫn xpath và các hàm thao tác kiểm tra website

- Bước 4: Tạo package TestCases để lưu trữ các ca kiểm thử



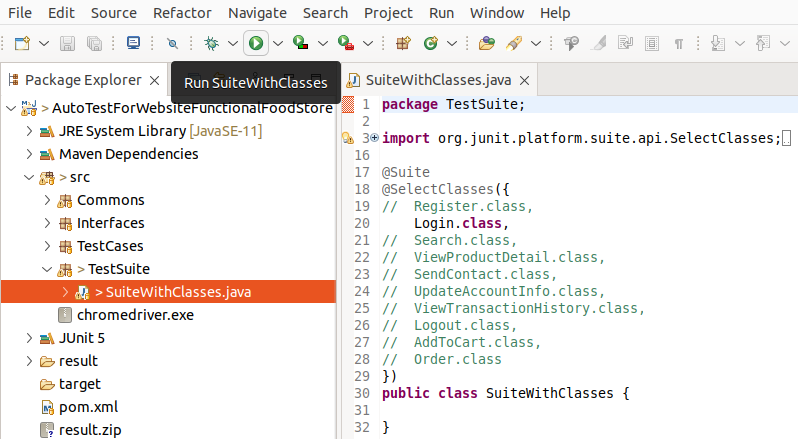
Hình 3.4: Tạo file chứa các ca kiểm thử của từng chức năng

- Bước 5: Tạo package TestSuite để lưu trữ các bộ test case cần kiểm thử



Hình 3.5: Tạo file chứa các chức năng cần kiểm thử

- Bước 6: Để chạy toàn bộ các ca kiểm thử của các chức năng thì mở file SuiteWithClasses.java rồi nhấn nút  trên thanh menu



Hình 3.6: Chạy các ca kiểm thử

## 3.4 Kết quả kiểm thử

Bảng 3.21: Bảng kết quả kiểm thử cho “Đăng ký”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra nhập số điện thoại hợp lệ | 2. Hiển thị trang "Cập nhật thông tin cá nhân" | Pass |
| TEST\_02 | Kiểm tra không nhập số điện thoại | 2. Hiển thị thông báo "Bạn chưa nhập số điện thoại" | Pass |
| TEST\_03 | Kiểm tra nhập số điện thoại đã đăng ký | 2. Hiển thị thông báo "Số điện thoại này đã đăng kí." | Fail |
| TEST\_04 | Kiểm tra nhập số điện thoại có chứa ký tự chữ | 2. Hiển thị trang "Cập nhật thông tin cá nhân" | Fail |
| TEST\_05 | Kiểm tra nhập số điện thoại có chứa ký tự đặc biệt | 2. Hiển thị trang "Cập nhật thông tin cá nhân" | Fail |
| TEST\_06 | Kiểm tra nhập số điện thoại có chứa ký tự khoảng trắng | 2. Hiển thị trang "Cập nhật thông tin cá nhân" | Fail |
| TEST\_07 | Kiểm tra nhập số điện thoại < 10 ký tự số | 2. Hiển thị trang "Cập nhật thông tin cá nhân" | Fail |
| TEST\_08 | Kiểm tra nhập số điện thoại = 10 ký tự số nhưng không đúng định dạng số điện thoại | 2. Hiển thị trang "Cập nhật thông tin cá nhân" | Fail |
| TEST\_09 | Kiểm tra nhập số điện thoại > 10 ký tự số | 2. Hiển thị trang "Cập nhật thông tin cá nhân" | Fail |
| TEST\_10 | Kiểm tra nhập thông tin các trường hợp lệ | 2. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_11 | Kiểm tra không nhập tên | 3. Hiển thị thông báo "Bạn chưa nhập tên" | Pass |
| TEST\_12 | Kiểm tra nhập tên < 50 ký tự | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_13 | Kiểm tra nhập tên = 50 ký tự | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_14 | Kiểm tra nhập tên > 50 ký tự | 3. Hiển thị thông báo "Họ tên vượt quá 50 kí tự" | Pass |
| TEST\_15 | Kiểm tra không nhập ngày sinh | 3. Hiển thị thông báo "Bạn chưa nhập thông tin ngày sinh" | Fail |
| TEST\_16 | Kiểm tra nhập ngày sinh có chứa ký tự chữ | 1. Chặn các ký tự chữ  3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_17 | Kiểm tra nhập ngày sinh có chứa ký tự đặc biệt | 1. Chặn các ký tự đặc biệt  3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_18 | Kiểm tra nhập ngày sinh < ngày hiện tại | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_19 | Kiểm tra nhập ngày sinh = ngày hiện tại | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_20 | Kiểm tra nhập ngày sinh > ngày hiện tại | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Fail |
| TEST\_21 | Kiểm tra chọn giới tính khác giá trị mặc định | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_22 | Kiểm tra không nhập địa chỉ | 3. Hiển thị thông báo "Bạn chưa nhập địa chỉ" | Pass |
| TEST\_23 | Kiểm tra không nhập mật khẩu | 3. Hiển thị thông báo "Bạn chưa nhập mật khẩu" | Pass |
| TEST\_24 | Kiểm tra nhập mật khẩu có chứa ký tự đặc biệt | 3. Hiển thị thông báo "Mật khẩu không đúng định dạng" | Pass |
| TEST\_25 | Kiểm tra nhập mật khẩu có chứa ký tự khoảng trắng | 3. Hiển thị thông báo "Mật khẩu không đúng định dạng" | Pass |
| TEST\_26 | Kiểm tra nhập mật khẩu < 8 ký tự | 3. Hiển thị thông báo "Mật khẩu cần ít nhất 8 kí tự" | Pass |
| TEST\_27 | Kiểm tra nhập mật khẩu = 8 ký tự | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_28 | Kiểm tra nhập mật khẩu = 16 ký tự | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_29 | Kiểm tra nhập mật khẩu > 16 ký tự | 3. Hiển thị thông báo "Mật khẩu tối đa 16 kí tự" | Pass |
| TEST\_30 | Kiểm tra không nhập mật khẩu nhập lại | 3. Hiển thị thông báo "Bạn chưa nhập lại mật khẩu" | Pass |
| TEST\_31 | Kiểm tra nhập mật khẩu nhập lại có chứa ký tự đặc biệt | 3. Hiển thị thông báo "Mật khẩu nhập lại không đúng định dạng" | Pass |
| TEST\_32 | Kiểm tra nhập mật khẩu nhập lại có chứa ký tự khoảng trắng | 3. Hiển thị thông báo "Mật khẩu nhập lại không đúng định dạng" | Pass |
| TEST\_33 | Kiểm tra nhập mật khẩu nhập lại < 8 ký tự | 3. Hiển thị thông báo "Mật khẩu nhập lại cần ít nhất 8 kí tự" | Pass |
| TEST\_34 | Kiểm tra nhập mật khẩu nhập lại > 16 ký tự | 3. Hiển thị thông báo "Mật khẩu nhập lại có tối đa 16 kí tự" | Fail |
| TEST\_35 | Kiểm tra nhập mật khẩu với mật khẩu nhập lại không khớp | 3. Hiển thị thông báo "Mật khẩu không khớp" | Pass |

Bảng 3.22: Bảng kết quả kiểm thử cho “Đăng nhập”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra nhập thông tin các trường hợp lệ | 2.  - Đăng nhập vào hệ thống thành công  - Hiển thị "Trang chủ" | Pass |
| TEST\_02 | Kiểm tra không nhập số điện thoại | 3. Hiển thị thông báo "Bạn chưa nhập tài khoản" | Fail |
| TEST\_03 | Kiểm tra nhập số điện thoại chưa đăng ký | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Pass |
| TEST\_04 | Kiểm tra nhập số điện thoại có chứa ký tự chữ | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_05 | Kiểm tra nhập số điện thoại có chứa ký tự đặc biệt | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_06 | Kiểm tra nhập số điện thoại có chứa ký tự khoảng trắng | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_07 | Kiểm tra nhập số điện thoại < 10 ký tự số | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_08 | Kiểm tra nhập số điện thoại = 10 ký tự số nhưng không đúng định dạng số điện thoại | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_09 | Kiểm tra nhập số điện thoại > 10 ký tự số | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_10 | Kiểm tra không nhập mật khẩu | 3. Hiển thị thông báo "Bạn chưa nhập mật khẩu" | Pass |
| TEST\_11 | Kiểm tra nhập mật khẩu có chứa ký tự đặc biệt | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_12 | Kiểm tra nhập mật khẩu có chứa ký tự khoảng trắng | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_13 | Kiểm tra nhập mật khẩu < 8 ký tự | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_14 | Kiểm tra nhập mật khẩu = 8 ký tự | 2.  - Đăng nhập vào hệ thống thành công  - Hiển thị "Trang chủ" | Pass |
| TEST\_15 | Kiểm tra nhập mật khẩu = 16 ký tự | 2.  - Đăng nhập vào hệ thống thành công  - Hiển thị "Trang chủ" | Pass |
| TEST\_16 | Kiểm tra nhập mật khẩu > 16 ký tự | 3. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Fail |
| TEST\_17 | Kiểm tra nhập mật khẩu không chính xác | 2. Hiển thị thông báo "Thông tin đăng nhập không chính xác" | Pass |

Bảng 3.23: Bảng kết quả kiểm thử cho “Tìm kiếm sản phẩm”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra không nhập từ khóa | 2. Hiển thị toàn bộ sản phẩm trong hệ thống | Pass |
| TEST\_02 | Kiểm tra tìm kiếm thành công | 2. Hiển thị toàn bộ sản phẩm có tên giống với từ khóa đã nhập | Pass |
| TEST\_03 | Kiểm tra tìm kiếm không thành công | 2. Không hiển thị sản phẩm nào | Pass |
| TEST\_04 | Kiểm tra tìm kiếm không phân biệt chữ hoa với chữ thường | 2. Hiển thị toàn bộ sản phẩm có tên giống với từ khóa đã nhập | Pass |
| TEST\_05 | Kiểm tra nhập toàn ký tự khoảng trắng | 2. Hiển thị toàn bộ sản phẩm trong hệ thống | Pass |

Bảng 3.24: Bảng kết quả kiểm thử cho “Xem thông tin chi tiết sản phẩm”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra xem thông tin chi tiết sản phẩm thành công | 2. Chuyển đến màn hình Thông tin chi tiết sản phẩm | Pass |

Bảng 3.25: Bảng kết quả kiểm thử cho “Gửi liên hệ”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra nhập thông tin hợp lệ | 2. Hiển thị thông báo "Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công" | Pass |
| TEST\_02 | Kiểm tra không nhập tên | 3. Hiển thị thông báo "Bạn chưa nhập tên" | Pass |
| TEST\_03 | Kiểm tra nhập tên < 50 ký tự | 2. Hiển thị thông báo "Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công" | Pass |
| TEST\_04 | Kiểm tra nhập tên = 50 ký tự | 2. Hiển thị thông báo "Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công" | Pass |
| TEST\_05 | Kiểm tra nhập tên > 50 ký tự | 3. Hiển thị thông báo "Họ tên vượt quá 50 kí tự" | Pass |
| TEST\_06 | Kiểm tra không nhập email | 3. Hiển thị thông báo "Bạn chưa nhập email" | Pass |
| TEST\_07 | Kiểm tra nhập email không đúng định dạng | 3. Hiển thị thông báo "Email không đúng định dạng" | Pass |
| TEST\_08 | Kiểm tra không nhập nội dung | 3. Hiển thị thông báo "Bạn chưa nhập nội dung" | Pass |
| TEST\_09 | Kiểm tra nhập nội dung trên một dòng | 3. Hiển thị thông báo "Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công" | Pass |
| TEST\_10 | Kiểm tra nhập nội dung trên nhiều dòng | 3. Hiển thị thông báo "Cám ơn bạn đã liên hệ với chúng tôi. Chúc bạn sức khỏe và thành công" | Pass |

Bảng 3.26: Bảng kết quả kiểm thử cho “Cập nhật thông tin cá nhân”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra nhập thông tin các trường hợp lệ | 2. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_02 | Kiểm tra không nhập tên | 3. Hiển thị thông báo "Bạn chưa nhập tên" | Pass |
| TEST\_03 | Kiểm tra nhập tên < 50 ký tự | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_04 | Kiểm tra nhập tên = 50 ký tự | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_05 | Kiểm tra nhập tên > 50 ký tự | 3. Hiển thị thông báo "Họ tên vượt quá 50 kí tự" | Pass |
| TEST\_06 | Kiểm tra không nhập ngày sinh | 3. Hiển thị thông báo "Bạn chưa nhập thông tin ngày sinh" | Fail |
| TEST\_07 | Kiểm tra nhập ngày sinh có chứa ký tự chữ | 1. Chặn các ký tự chữ  3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_08 | Kiểm tra nhập ngày sinh có chứa ký tự đặc biệt | 1. Chặn các ký tự đặc biệt  3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_09 | Kiểm tra nhập ngày sinh < ngày hiện tại | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_10 | Kiểm tra nhập ngày sinh = ngày hiện tại | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_11 | Kiểm tra nhập ngày sinh < ngày hiện tại | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Fail |
| TEST\_12 | Kiểm tra chọn giới tính khác giá trị đã chọn | 3. Hiển thị thông báo "Thông tin tài khoản được cập nhật thành công" | Pass |
| TEST\_13 | Kiểm tra không nhập địa chỉ | 3. Hiển thị thông báo "Bạn chưa nhập địa chỉ" | Pass |

Bảng 3.27: Bảng kết quả kiểm thử cho “Xem lịch sử đặt hàng”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra xem lịch sử đặt hàng thành công | 2. Hiển thị trang "Lịch sử đặt hàng" | Pass |

Bảng 3.28: Bảng kết quả kiểm thử cho “Đăng xuất”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra đăng xuất thành công | 2.  - Đăng xuất khỏi hệ thống thành công  - Hiển thị "Trang chủ" | Pass |

Bảng 3.29: Bảng kết quả kiểm thử cho “Thêm vào giỏ hàng”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra với số lượng mặc định | 1. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" | Pass |
| TEST\_02 | Kiểm tra nhập số lượng hợp lệ | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" | Pass |
| TEST\_03 | Kiểm tra nhập số lượng chứa ký tự chữ | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" | Pass |
| TEST\_04 | Kiểm tra nhập số lượng chứa ký tự đặc biệt | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" | Pass |
| TEST\_05 | Kiểm tra nhập số lượng chứa ký tự khoảng trắng | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" | Fail |
| TEST\_06 | Kiểm tra với số lượng trống | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" | Pass |
| TEST\_07 | Kiểm tra nhập số lượng là toàn ký tự khoảng trắng | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" | Pass |
| TEST\_08 | Kiểm tra nhập số lượng là số âm | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" | Pass |
| TEST\_09 | Kiểm tra nhập số lượng bằng 0 | 2. Hiển thị thông báo "Số lượng sản phẩm không hợp lệ" | Pass |
| TEST\_10 | Kiểm tra nhập số lượng có dấu "+" ở đầu | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" | Pass |
| TEST\_11 | Kiểm tra nhập số lượng là số thập phân | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" | Fail |
| TEST\_12 | Kiểm tra nhấn nút "+" | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" | Pass |
| TEST\_13 | Kiểm tra nhấn nút "-" | 2. Hiển thị thông báo "Sản phẩm của bạn đã được thêm vào giỏ hàng" | Pass |

Bảng 3.30: Bảng kết quả kiểm thử cho “Đặt hàng”

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Kết quả thực tế** | **Kết quả kiểm thử** |
| TEST\_01 | Kiểm tra nhập thông tin các trường hợp lệ | 2. Hiển thị thông báo "Đơn hàng của bạn đã được tiếp nhận" | Pass |
| TEST\_02 | Kiểm tra không nhập địa chỉ | 3. Hiển thị thông báo "Bạn chưa nhập địa chỉ" | Pass |
| TEST\_03 | Kiểm tra không nhập ghi chú | 3. Hiển thị thông báo "Đơn hàng của bạn đã được tiếp nhận" | Pass |
| TEST\_04 | Kiểm tra nhập ghi chú trên một dòng | 3. Hiển thị thông báo "Đơn hàng của bạn đã được tiếp nhận" | Pass |
| TEST\_05 | Kiểm tra nhập ghi chú trên nhiều dòng | 3. Hiển thị thông báo "Đơn hàng của bạn đã được tiếp nhận" | Pass |

Bảng 3.31: Bảng tổng hợp kết quả kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tổng số TCs** | **Số TCs Pass** | **Số TCs Fail** |
| **Đăng ký** | 35 | 25 | 10 |
| **Đăng nhập** | 17 | 6 | 11 |
| **Tìm kiếm sản phẩm** | 5 | 5 | 0 |
| **Xem thông tin chi tiết sản phẩm** | 1 | 1 | 0 |
| **Gửi liên hệ** | 8 | 10 | 0 |
| **Cập nhật thông tin cá nhân** | 13 | 11 | 2 |
| **Xem lịch sử đặt hàng** | 1 | 1 | 0 |
| **Đăng xuất** | 1 | 1 | 0 |
| **Thêm vào giỏ hàng** | 13 | 11 | 2 |
| **Đặt hàng** | 5 | 5 | 0 |
| **TỔNG** | **99** | **76** | **25** |

KẾT LUẬN

Kiểm thử phần mềm hiện nay giữ vai trò hết sức quan trọng trong việc đảm bảo chất lượng của một sản phẩm phần mềm. Mặc dù việc sử dụng các công cụ kiểm thử tự động đã giúp giảm thời gian, kinh phí, nhân lực trong quá trình phát triển phần mềm. Nhưng cũng có những việc mà công cụ kiểm thử tự động không thể thay thế được kiểm thử viên, hoặc nếu có thì việc cấu hình cho nó khó khăn hơn rất nhiều so với kiểm thử thủ công. Do đó, kiểm thử viên giỏi phải là người nhận biết được khi nào nên sử dụng công cụ kiểm thử tự động và khi nào nên kiểm thử thủ công.

Sau thời gian thực hiện đồ án dưới sự hướng dẫn của Ts Đỗ Mạnh Hùng, kết quả mà em thu được cụ thể như sau:

**Kết quả đạt được:**

- Trình bày được các kiến thức cơ bản về kiểm thử phần mềm nói chung và kiểm thử phần mềm tự động cho các ứng dụng web nói riêng.

- Giới thiệu được các đặc điểm, thành phần của công cụ kiểm thử tự động Selenium. Kết hợp công cụ Selenium WebDriver và framework JUnit trong kiểm thử tự động ứng dụng web.

- Áp dụng các kiến thức đã tìm hiểu vào thực hiện kiểm thử tự động các chức năng chính của ứng dụng web bán hàng online Functional Food Store.

**Hạn chế:**

- Chưa thể giả lập nhiều người dùng ảo chạy đồng thời để kiểm thử khả năng xử lý xung đột khi có nhiề khách hàng cùng đặt hàng vào một thời điểm.

- Chưa nghiên cứu được phương pháp kiểm thử hướng từ khóa với Selenium.

- Mới kiểm thử với ứng dụng nhỏ, chưa kiểm thử với ứng dụng thực tế.

**Hướng phát triển đề tài:**

Đồ án có thể tiếp tục tìm hiểu sâu hơn về các vấn đề của kiểm thử phần mềm và đặc biệt là công cụ kiểm thử tự động Selenium để có thể vận dụng kiểm thử các ứng dụng lớn hơn trong thực tiễn.

TÀI LIỆU THAM KHẢO

1. Hoàng Quang Huy (2016), “*Giáo trình kiểm thử phần mềm*”, Nhà xuất bản Thống kê.
2. Rex Black, Erik Van Veenendaal, Dorothy Graham (2012), “Foundations of Software Testing ISTQB Certification” , Cengage Learning
3. Tài liệu về Selenium Webdriver <https://www.selenium.dev/documentation/webdriver/>
4. Tài liệu về JUnit: <https://junit.org/junit5/docs/current/user-guide/>
5. Tài liệu về Take Screenshot: <https://www.guru99.com/take-screenshot-selenium-webdriver.html>
6. Tài liệu về Create a Selenium Maven project: <https://www.geeksforgeeks.org/how-to-create-a-selenium-maven-project-with-eclipse-to-open-chrome-browser/>