**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

****

**ĐỒ ÁN 4**

**KIỂM THỬ TỰ ĐỘNG VÀ XÂY DỰNG FRAMEWORK DỰA TRÊN SELENIUM CHO WEBSITE CHUS**

NGÀNH: KỸ THUẬT PHẦN MỀM

CHUYÊN NGÀNH: KIỂM THỬ VÀ

ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

SINH VIÊN: **HOÀNG MINH NGUYỆT**

MÃ LỚP: **12522T.1CN**

HƯỚNG DẪN: **TS ĐÀO ANH HIỂN**

**HƯNG YÊN - 2025**

NHẬN XÉT

Nhận xét của giảng viên hướng dẫn:

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

**GIẢNG VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

LỜI CAM ĐOAN

Em xin cam đoan đồ án “Kiểm thử tự động và xây dựng framework dựa trên Selenium cho website Chus” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của thầy Đào Anh Hiển.

Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong đồ án và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước Khoa và Nhà trường.

|  |  |
| --- | --- |
|  | *Hưng Yên, ngày … tháng … năm…..*  Sinh viên |

LỜI CẢM ƠN

Để có thể hoàn thành đồ án này, đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin - Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện đồ án môn học này.

Đặc biệt em xin chân thành cảm ơn thầy Đào Anh Hiển đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện đồ án vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được đồ án này.

Mặc dù em đã có cố gắng, nhưng với kiến thức còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy giáo, Cô giáo về những kết quả triển khai trong đồ án.

Em xin trân trọng cảm ơn!MỤC LỤC

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 13](#_Toc201206399)

[1.1 Lý do chọn đề tài 13](#_Toc201206400)

[1.2 Mục tiêu của đề tài 14](#_Toc201206401)

[1.2.1 Mục tiêu tổng quát 14](#_Toc201206402)

[1.2.2 Mục tiêu cụ thể 14](#_Toc201206403)

[1.3 Giới hạn và phạm vi của đề tài 14](#_Toc201206404)

[1.3.1 Đối tượng nghiên cứu 14](#_Toc201206405)

[1.3.2 Phạm vi nghiên cứu 15](#_Toc201206406)

[1.4 Nội dung thực hiện 15](#_Toc201206407)

[1.5 Phương pháp tiếp cận 16](#_Toc201206408)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 18](#_Toc201206409)

[2.1 Tổng quan về kiểm thử tự động 18](#_Toc201206410)

[2.1.1 Định ngĩa, vai trò của kiểm thử tự động 18](#_Toc201206411)

[2.1.2 Phân loại kiểm thử tự động 19](#_Toc201206412)

[2.1.3 Quy trình kiểm thử tự động 21](#_Toc201206413)

[2.1.4 Ưu, nhược điểm kiểm thử tự động 24](#_Toc201206414)

[2.2 Giới thiệu các công cụ kiểm thử: 25](#_Toc201206415)

[2.2.1 Selenium 25](#_Toc201206416)

[2.3 Các kiến trúc của các Framework kiểm thử tự động 26](#_Toc201206417)

[2.3.1 Định nghĩa 26](#_Toc201206418)

[2.3.2 Ưu, nhược điểm 27](#_Toc201206419)

[2.3.3 Một số framework kiểm thử tự động 28](#_Toc201206420)

[2.4 Test Script và phương pháp viết Test Script 29](#_Toc201206421)

[2.4.1 Định nghĩa 29](#_Toc201206422)

[2.4.2 Mục tiêu của viết Test Script 30](#_Toc201206423)

[2.4.3 Phương pháp viết Test Script 30](#_Toc201206424)

[2.4.4 Nguyên tắc viết Test Script tốt 32](#_Toc201206425)

[CHƯƠNG 3: PHÂN TÍCH YÊU CẦU VÀ LẬP KẾ HOẠCH KIỂM THỬ 33](#_Toc201206426)

[3.1 Phân tích yêu cầu kiểm thử 33](#_Toc201206427)

[3.1.1 Yêu cầu chức năng 34](#_Toc201206428)

[3.1.2 Yêu cầu phi chức năng 43](#_Toc201206429)

[3.1.3 Giới hạn và phạm vi đề tài 43](#_Toc201206430)

[3.2 Kế hoạch kiểm thử 44](#_Toc201206431)

[3.2.1 Lịch trình công việc 44](#_Toc201206432)

[3.2.1. Phương pháp kiểm thử 45](#_Toc201206433)

[3.2.2 Môi trường kiểm thử 46](#_Toc201206434)

[3.3 Xây dựng Framework 47](#_Toc201206435)

[3.3.1 Mục tiêu của Framework 47](#_Toc201206436)

[3.3.2 Mô hình kiến trúc áp dụng 47](#_Toc201206437)

[3.3.3 Cấu trúc thư mục của framework 49](#_Toc201206438)

[CHƯƠNG 4: TRIỂN KHAI KIỂM THỬ ỨNG DỤNG 50](#_Toc201206439)

[4.1 Thiết kế kịch bản kiểm thử 50](#_Toc201206440)

[4.1.1 Xây dựng mô hình Page Object cho website Chus 50](#_Toc201206441)

[4.1.2 Cài đặt lớp BaseSetup 54](#_Toc201206442)

[4.1.3 Kiểm thử hướng dữ liệu 55](#_Toc201206443)

[4.1.4 Tích hợp tiện ích hỗ trợ 56](#_Toc201206444)

[4.1.5 Ghi kết quả báo cáo 57](#_Toc201206445)

[4.1.6 Quản lý kiểm thử và thư viện 60](#_Toc201206446)

[4.2 Thiết kế kịch bản kiểm thử chức năng 61](#_Toc201206447)

[4.3 Xây dựng và thực thi các trường hợp kiểm thử 64](#_Toc201206448)

[4.3.1 Xây dựng và thực thi các trường hợp kiểm thử chức năng Đăng nhập 64](#_Toc201206449)

[4.3.2 Xây dựng và thực thi các trường hợp Kiểm thử chức năng 65](#_Toc201206450)

[4.3.3 Xây dựng và thực thi các trường hợp Kiểm thử hướng dữ liệu 67](#_Toc201206451)

[4.3.4 Xây dựng và thực thi các trường hợp kiểm thử chức năng Đặt hàng 68](#_Toc201206452)

[4.3 Báo cáo và phân tích kết quả kiểm thử 69](#_Toc201206453)

[4.4.1 Báo cáo kiểm thử chức năng Đăng nhập 69](#_Toc201206454)

[4.4.2 Báo cáo kiểm thử chức năng Đăng ký 70](#_Toc201206455)

[4.4.3 Báo cáo kiểm thử chức năng Tìm kiếm 71](#_Toc201206456)

[4.4.4 Báo cáo kiểm thử chức năng Đặt hàng 71](#_Toc201206457)

[4.4.5 Báo cáo kiểm thử chức năng website Chus 72](#_Toc201206458)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 73](#_Toc201206459)

[5.1 Kết quả đạt được của đề tài 73](#_Toc201206460)

[5.2 Hạn chế của đề tài 74](#_Toc201206461)

[5.3 Hướng phát triển của đề tài 74](#_Toc201206462)

[TÀI LIỆU THAM KHẢO 76](#_Toc201206463)

DANH MỤC CÁC THUẬT NGỮ

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Cụm từ tiếng anh | Diễn giải |
| 1 | SRS | Software Requirement Specification | Đặc tả yêu cầu phần mềm |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

DANH MỤC CÁC BẢNG

[Bảng 3. 1: Danh sách yêu cầu chức năng 31](#_Toc201186649)

[Bảng 3. 2: Phương pháp kiểm thử 42](#_Toc201186650)

[Bảng 3. 4 Môi trường kiểm thử 43](#_Toc201186651)

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 3. 1: Giao diện Chức năng Đăng ký 35](#_Toc201187110)

[Hình 3. 2: Giao diện Chức năng Đăng nhập 36](#_Toc201187111)

[Hình 3. 3: Giao diện Chức năng Quên mật khẩu 37](#_Toc201187112)

[Hình 3. 4: Giao diện Chức năng Tìm kiếm 38](#_Toc201187113)

[Hình 3. 5: Giao diện Chức năng Chi tiết 39](#_Toc201187114)

[Hình 3. 6: Giao diện Chức năng Giỏ hàng 40](#_Toc201187115)

[Hình 3. 7: Giao diện Chức năng Đặt hàng 41](#_Toc201187116)

[Hình 3. 8: Giao diện Chức năng Thông tin tài khoản 42](#_Toc201187117)

[Hình 3. 9: Giao diện Thêm sổ địa chỉ 42](#_Toc201187118)

[Hình 4. 1: Mã nguồn lớp LoginPage đại diện trang đăng nhập trong mô hình Page Object Model (POM) 50](#_Toc201186671)

[Hình 4. 2: Mã nguồn lớp HomePage đại diện trang trang chủ trong mô hình Page Object Model (POM) 51](#_Toc201186672)

[Hình 4. 3**:** Mã nguồn lớp OrderPage mô phỏng trang thanh toán website 52](#_Toc201186673)

[Hình 4. 4: Mã nguồn lớp SearchPage.java mô tả hành vi tìm kiếm sách trên website Chus 53](#_Toc201186674)

[Hình 4. 5: Mã nguồn lớp SignupPage.java mô tả hành vi tìm kiếm sách trên website Chus 54](#_Toc201186675)

[Hình 4. 6: Mã nguồn BaseSetup.java website Chus 54](#_Toc201186676)

[Hình 4. 7: Mã nguồn QlWebdriver website Chus 55](#_Toc201186677)

[Hình 4. 8: Lớp đọc dữ liệu từ ExcelReader 56](#_Toc201186678)

[Hình 4. 9: Hỗ trợ thao tác mở, đọc, ghi dữ liệu Excel – ExcelHelpers 57](#_Toc201186679)

[Hình 4. 10: Ghi báo cáo Login 58](#_Toc201186680)

[Hình 4. 11: Ghi báo cáo Signup 58](#_Toc201186681)

[Hình 4. 12: Ghi báo cáo Search 59](#_Toc201186682)

[Hình 4. 13: Ghi báo cáo Order 59](#_Toc201186683)

[Hình 4. 14: Cài đặt testng.xml 60](#_Toc201186684)

[Hình 4. 15: Cài đặt pom.xml 61](#_Toc201186685)

[Hình 4. 16: Kịch bản kiểm thử chức năng website Chus 61](#_Toc201186686)

[Hình 4. 17: Kịch bản kiểm thử chức năng Đăng nhập 62](#_Toc201186687)

[Hình 4. 18: Kịch bản kiểm thử Chức năng Đăng ký 63](#_Toc201186688)

[Hình 4. 19: Kịch bản kiểm thử Chức năng Tìm kiếm 63](#_Toc201186689)

[Hình 4. 20: Kịch bản kiểm thử Chức năng Đặt hàng 64](#_Toc201186690)

[Hình 4. 21: Kịch bản kiểm thử chức năng Đăng nhập 65](#_Toc201186691)

[Hình 4. 22: Dữ liệu chức năng Đăng nhập 65](#_Toc201186692)

[Hình 4. 23: Kịch bản kiểm thử chức năng Đăng ký 66](#_Toc201186693)

[Hình 4. 24: Dữ liệu chức năng Đăng ký 67](#_Toc201186694)

[Hình 4. 25: Kịch bản kiểm thử chức năng Tìm kiếm sản phẩm 68](#_Toc201186695)

[Hình 4. 26: Dữ liệu chức năng Tìm kiếm sản phẩm 68](#_Toc201186696)

[Hình 4. 27: Tổng hợp kết quả kiểm thử chức năng Đăng nhập 69](#_Toc201186697)

[Hình 4. 28: Kết quả kiểm thử chức năng Đăng nhập 69](#_Toc201186698)

[Hình 4. 29: Tổng hợp kết quả kiểm thử chức năng Đăng ký 70](#_Toc201186699)

[Hình 4. 30: Kết quả kiểm thử chức năng Đăng nhập 70](#_Toc201186700)

[Hình 4. 31: Tổng hợp kết quả kiểm thử chức năng Đăng ký 71](#_Toc201186701)

[Hình 4. 32: Kết quả kiểm thử chức năng Tìm kiếm 71](#_Toc201186702)

[Hình 4. 33: Tổng hợp kết quả kiểm thử chức năng Giỏ hàng 72](#_Toc201186703)

[Hình 4. 34: Kết quả kiểm thử chức năng Giỏ hàng 72](#_Toc201186704)

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Lý do chọn đề tài

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, mua sắm trực tuyến đã trở thành xu hướng phổ biến và quen thuộc đối với người tiêu dùng. Đặc biệt, sách là một trong những mặt hàng được lựa chọn mua online nhiều nhờ sự tiện lợi, nhanh chóng và kho sản phẩm đa dạng. Nắm bắt xu thế đó, hàng loạt nền tảng bán sách trực tuyến đã ra đời, trong đó có website Chus - một nền tảng phân phối sách trực tuyến với số lượng sản phẩm lớn và nhiều chức năng phục vụ nhu cầu người dùng.

Để đảm bảo mang lại trải nghiệm mua sắm tốt nhất cho khách hàng, website Chus cần vận hành ổn định, nhanh chóng, chính xác và đảm bảo an toàn bảo mật. Vì vậy, công tác kiểm thử phần mềm đóng vai trò then chốt trong việc phát hiện, khắc phục lỗi và duy trì chất lượng hệ thống trước khi sản phẩm được đưa vào sử dụng thực tế. Tuy nhiên, phương pháp kiểm thử thủ công hiện nay vẫn còn tồn tại nhiều hạn chế như tốn nhiều thời gian, nhân lực, dễ mắc sai sót và khó tái sử dụng các kịch bản kiểm thử khi hệ thống thay đổi.

Để khắc phục những hạn chế đó, kiểm thử tự động được áp dụng nhờ những lợi ích vượt trội so với kiểm thử thủ công. Cụ thể, kiểm thử tự động giúp tiết kiệm thời gian và chi phí, tăng độ chính xác, dễ dàng tái sử dụng và mở rộng, nâng cao hiệu suất kiểm thử**,** hỗ trợ kiểm thử liên tục.

Đặc biệt với website Chus, người dùng thường xuyên thực hiện các thao tác như đăng nhập, đăng ký, tìm kiếm sách, thêm sách yêu thích, thêm vào giỏ hàng và đặt mua. Nếu thực hiện kiểm thử thủ công cho các thao tác này nhiều lần sẽ rất tốn thời gian và dễ xảy ra sai sót. Do đó, việc xây dựng một framework kiểm thử tự động cho website thương mại điện tử Chus là thực sự cần thiết nhằm đảm bảo hệ thống hoạt động ổn định, tiết kiệm chi phí, nâng cao chất lượng kiểm thử và tối ưu trải nghiệm cho người dùng.

1.2 Mục tiêu của đề tài

1.2.1 Mục tiêu tổng quát

Kiểm thử tự động website Chus dựa trên framework Selenium.

1.2.2 Mục tiêu cụ thể

* Tìm hiểu và phân tích quy trình kiểm thử tự động cho website Chus, lựa chọn phương pháp và công cụ kiểm thử tự động phù hợp, cụ thể là công cụ Selenium.
* Đưa ra các giải pháp nhằm tăng chất lượng kiểm thử tự động, giảm thiểu thời gian kiểm thử, nâng cao độ chính xác và hiệu suất hoạt động của hệ thống.
* Xây dựng và triển khai các kịch bản kiểm thử tự động cho các chức năng trong website Chus như: chức năng đăng nhập, đăng ký, tìm kiếm sách, thêm vào yêu thích, thêm vào giỏ hàng và thực hiện đặt mua.
* Thiết kế và xây dựng framework kiểm thử tự động trên nền tảng Selenium, đảm bảo tính linh hoạt, dễ dàng mở rộng và tái sử dụng cho các đợt kiểm thử tiếp theo.
* Thực hiện kiểm thử, thu thập kết quả và so sánh hiệu quả giữa kiểm thử thủ công và kiểm thử tự động về các tiêu chí: thời gian thực hiện, độ chính xác và khả năng tái sử dụng.

1.3 Giới hạn và phạm vi của đề tài

1.3.1 Đối tượng nghiên cứu

* Đối tượng nghiên cứu: Kiểm thử tự động website Chus, tập trung vào các phương pháp, công cụ và kỹ thuật kiểm thử chức năng, kiểm thử hiệu năng.
* Khách thể nghiên cứu: Người dùng website Chus, chuyên gia phát triển phần mềm, chuyên gia kiểm thử phần mềm.

1.3.2 Phạm vi nghiên cứu

* Phạm vi không gian: Website Chus.
* Phạm vi thời gian: Thu thập dữ liệu, triển khai và đánh giá trong khoảng thời gian từ 09/2025 - 06/2025.
* Ý nghĩa khoa học và thực tiễn của đề tài

**Về mặt khoa học:**

* Đề tài góp phần hệ thống hóa các kiến thức lý thuyết và thực tiễn liên quan đến kiểm thử phần mềm.
* Cung cấp một giải pháp framework kiểm thử tự động dựa trên Selenium WebDriver, giúp làm rõ quy trình thiết kế, triển khai và vận hành hệ thống kiểm thử tự động cho các website thương mại điện tử.
* Là tài liệu tham khảo hữu ích cho các nghiên cứu và đề tài sau này liên quan đến kiểm thử tự động và kiểm thử website thương mại điện tử.

**Về mặt thực tiễn:**

* Framework kiểm thử tự động do đề tài xây dựng giúp tiết kiệm thời gian, chi phí kiểm thử, đồng thời nâng cao độ chính xác và hiệu quả kiểm thử cho website Chus.
* Hỗ trợ phát hiện và xử lý lỗi kịp thời, đảm bảo hệ thống vận hành ổn định, an toàn, góp phần nâng cao chất lượng trải nghiệm người dùng.
* Có thể mở rộng và áp dụng cho các dự án website thương mại điện tử khác có quy mô và tính năng tương tự.

1.4 Nội dung thực hiện

* Nghiên cứu tài liệu về kiểm thử phần mềm, kiểm thử tự động, các công cụ kiểm thử phổ biến hiện nay.
* Khảo sát và phân tích website Chus để xác định các chức năng cần kiểm thử như: đăng ký, đăng nhập, tìm kiếm sách, thêm vào giỏ hàng, đặt mua, yêu thích sách…
* Lựa chọn công cụ kiểm thử phù hợp, cụ thể là Selenium WebDriver, dựa trên yêu cầu hệ thống và khả năng đáp ứng các chức năng kiểm thử tự động.
* Xây dựng bộ kịch bản kiểm thử tự động, bao gồm các trường hợp kiểm thử chức năng và kiểm thử bảo mật cho các chức năng chính của website.
* Thiết kế và triển khai framework kiểm thử tự động cho website Chus, thực hiện kiểm thử thực tế, thu thập và phân tích kết quả kiểm thử.
* Đánh giá hiệu quả kiểm thử tự động so với kiểm thử thủ công, về các tiêu chí: thời gian thực hiện, độ chính xác, khả năng tái sử dụng và hiệu suất hệ thống.
* Viết báo cáo tổng kết quá trình nghiên cứu, trình bày kết quả đạt được, khó khăn gặp phải và đề xuất hướng phát triển trong tương lai.

1.5 Phương pháp tiếp cận

Để thực hiện đề tài “Xây dựng framework kiểm thử tự động cho website thương mại điện tử Chus bằng Selenium”, nhóm thực hiện lựa chọn phương pháp tiếp cận trực tiếp và từng bước như sau:

* Tiếp cận trực tiếp với người sử dụng hệ thống, thu thập yêu cầu và phản hồi từ người dùng, quản trị viên hệ thống và bộ phận kỹ thuật nhằm nắm rõ các chức năng quan trọng, các lỗi thường gặp cũng như nhu cầu kiểm thử hiện tại của website Chus.
* Khảo sát hệ thống website hiện tại, phân tích quy trình kiểm thử thủ công đang áp dụng để nhận diện các điểm hạn chế và những khu vực chức năng cần kiểm thử liên tục.
* Nghiên cứu, đề xuất và lựa chọn phương án kiểm thử tự động phù hợp, trong đó tập trung vào công cụ Selenium nhờ khả năng hỗ trợ kiểm thử giao diện web, đa trình duyệt và dễ tích hợp vào quy trình kiểm thử liên tục (CI/CD).
* Lên kế hoạch xây dựng framework kiểm thử tự động, thiết kế các kịch bản kiểm thử dựa trên các chức năng trọng yếu của hệ thống, đồng thời đảm bảo tính mở rộng và tái sử dụng cho các đợt kiểm thử về sau.
* Tiến hành thử nghiệm framework kiểm thử tự động trên hệ thống thực tế, ghi nhận kết quả kiểm thử và đánh giá hiệu suất hệ thống.
* Đề xuất các phương án cải tiến, hoàn thiện framework kiểm thử, đồng thời kiến nghị hướng áp dụng lâu dài vào quy trình phát triển và vận hành website.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về kiểm thử tự động

2.1.1 Định nghĩa, vai trò của kiểm thử tự động

Trong quá trình phát triển phần mềm, kiểm thử đóng vai trò vô cùng quan trọng nhằm đảm bảo sản phẩm hoạt động đúng chức năng, ổn định và đáp ứng được yêu cầu của người sử dụng. Trong đó, kiểm thử tự động (Automated Testing) là một phương pháp kiểm thử hiện đại, sử dụng các công cụ và phần mềm để tự động hóa việc thực thi các kịch bản kiểm thử thay cho phương pháp kiểm thử thủ công truyền thống.

Kiểm thử tự động giúp kiểm tra phần mềm một cách nhanh chóng, chính xác và hiệu quả, đặc biệt trong những dự án có quy mô lớn, nhiều chức năng hoặc thường xuyên thay đổi, cập nhật tính năng. Các kịch bản kiểm thử được lập trình sẵn và có thể thực hiện lặp đi lặp lại nhiều lần với cùng một tập dữ liệu, giúp phát hiện lỗi sớm và đảm bảo hệ thống luôn vận hành ổn định.

Mục tiêu chính của kiểm thử tự động là:

* Tăng hiệu quả và tốc độ thực hiện kiểm thử.
* Giảm thiểu sai sót do yếu tố con người trong quá trình kiểm tra.
* Tiết kiệm thời gian và chi phí cho việc kiểm tra các chức năng lặp lại nhiều lần.
* Hỗ trợ kiểm thử liên tục (Continuous Testing) trong các quy trình phát triển phần mềm hiện đại như CI/CD.
* Dễ dàng tái sử dụng, chỉnh sửa và mở rộng các kịch bản kiểm thử khi phần mềm thay đổi.

Những lợi ích nổi bật của kiểm thử tự động gồm:

* Tăng tốc độ kiểm thử: Thời gian thực thi các kịch bản kiểm thử được rút ngắn đáng kể so với kiểm thử thủ công.
* Giảm lỗi và nâng cao độ chính xác: Các bước kiểm thử được thực hiện nhất quán, giảm thiểu các sai sót do con người.
* Tái sử dụng và mở rộng: Các kịch bản kiểm thử tự động có thể sử dụng lại cho nhiều phiên bản phần mềm khác nhau.
* Hỗ trợ kiểm thử liên tục: Giúp kiểm tra phần mềm ngay khi có thay đổi mới, đảm bảo sản phẩm luôn ổn định trước khi đến tay người dùng.
* Tiết kiệm chi phí lâu dài: Mặc dù chi phí ban đầu để thiết lập kiểm thử tự động cao hơn, nhưng về lâu dài sẽ tiết kiệm nhân lực và chi phí vận hành.

2.1.2 Phân loại kiểm thử tự động

Có các loại kiểm thử tự động khác nhau tùy vào mục đích phạm vi kiểm thử.

1. **Kiểm thử đơn vị (Unit Testing)**

Là quá trình kiểm tra từng thành phần nhỏ nhất của phần mềm như hàm, phương thức, module hoặc đối tượng. Mục đích nhằm đảm bảo từng đơn vị hoạt động chính xác độc lập trước khi tích hợp vào hệ thống lớn hơn. Unit test giúp phát hiện lỗi sớm, giảm chi phí sửa lỗi và tăng khả năng tái sử dụng mã nguồn.

Các loại Unit test phổ biến:

* Black Box Testing: Kiểm tra input và output mà không xét logic bên trong.
* White Box Testing: Kiểm tra logic, cấu trúc nội bộ của mã nguồn.
* Gray Box Testing: Kết hợp hai phương pháp trên*.*

1. **Kiểm thử tích hợp (Integration Testing)**

Là kiểm tra quá trình kết hợp các module, thành phần của hệ thống để đảm bảo chúng hoạt động đồng bộ, truyền dữ liệu chính xác và không phát sinh lỗi khi tích hợp. Loại kiểm thử này giúp phát hiện sớm lỗi liên quan đến giao tiếp giữa các module.

Kỹ thuật kiểm thử tích hợp:

* Big Bang: Tích hợp tất cả module rồi kiểm thử một lần.
* Top-Down: Tích hợp dần từ trên xuống dưới.
* Bottom-Up: Tích hợp dần từ dưới lên trên.
* Sandwich (Hybrid): Kết hợp cả Top-Down và Bottom-Up.

1. **Kiểm thử chức năng (Functional Testing)**

Kiểm tra xem phần mềm có đáp ứng đúng các yêu cầu chức năng đặt ra hay không. Thường sử dụng phương pháp Black Box Testing, tập trung vào đầu vào, đầu ra mà không cần biết đến cấu trúc bên trong.

1. **Kiểm thử hồi quy (Regression Testing)**

Là kiểm tra lại các tính năng của hệ thống sau khi có thay đổi về mã nguồn (sửa lỗi, nâng cấp tính năng) nhằm đảm bảo các chức năng cũ vẫn hoạt động tốt và không phát sinh lỗi mới. Đây là loại kiểm thử quan trọng trong quá trình bảo trì phần mềm.

1. **Kiểm thử chấp nhận (Acceptance Testing):**

Kiểm thử chấp nhận là quá trình kiểm tra tính đúng đắn và đáp ứng yêu cầu của người dùng. Nó được thực hiện bởi người dùng cuối và đảm bảo các tính năng của phần mềm đã đáp ứng chính xác các tiêu chí chấp thuận được đề ra từ ban đầu, dưới góc nhìn của người dùng.

1. **Kiểm thử hoạt động (Operational Testing):**

Kiểm thử hoạt động tập trung vào việc đảm bảo rằng hệ thống hoạt động một cách đáng tin cậy trong môi trường sản xuất. Bằng cách này, ta có thể phát hiện và khắc phục các vấn đề liên quan đến hiệu suất, bảo mật và khả năng chịu tải của hệ thống.

1. **Kiểm thử phi chức năng (Non-functional Testing):**

Kiểm thử phi chức năng là quá trình kiểm tra các yếu tố không liên quan đến tính năng của phần mềm như trải nghiệm người dùng, hiệu suất, bảo mật, khả năng mở rộng và khả năng tương thích. Điều này giúp đảm bảo tính linh hoạt và hiệu quả của hệ thống trong các tình huống thực tế.

2.1.3 Quy trình kiểm thử tự động



1. **Lập kế hoạch và thiết kế kịch bản kiểm thử**

Ở giai đoạn đầu tiên, nhóm kiểm thử sẽ tiến hành lập kế hoạch tổng thể cho quá trình kiểm thử tự động. Công việc chính bao gồm:

Xác định mục tiêu và phạm vi kiểm thử tự động, những chức năng, module hoặc thành phần nào của phần mềm sẽ được tự động hóa.

Phân tích các yêu cầu hệ thống để xác định những chức năng có tính ổn định cao, ít thay đổi và có khả năng tự động hóa tốt.

Thiết kế các kịch bản kiểm thử chi tiết dựa trên các yêu cầu chức năng và phi chức năng đã phân tích. Các kịch bản này cần đầy đủ thông tin về:

* Dữ liệu đầu vào
* Các bước thực hiện
* Điều kiện đầu vào và đầu ra mong đợi

Việc thiết kế kịch bản kỹ lưỡng giúp quá trình tự động hóa sau này hiệu quả và giảm thiểu các lỗi phát sinh.

1. **Lựa chọn và thiết lập công cụ kiểm thử**

Tiếp theo, nhóm kiểm thử sẽ tiến hành:

* Khảo sát và lựa chọn công cụ kiểm thử tự động phù hợp với công nghệ phần mềm, ngân sách và nhu cầu dự án. Một số công cụ phổ biến như: Selenium, Katalon, TestComplete, Appium, JMeter,…
* Cài đặt và thiết lập môi trường kiểm thử tự động bao gồm các framework, thư viện và các công cụ hỗ trợ cần thiết.
* Cấu hình tích hợp công cụ kiểm thử với hệ thống quản lý phiên bản (Git), hệ thống CI/CD (Jenkins, GitLab CI), và các công cụ quản lý kiểm thử khác nếu có.

Việc lựa chọn đúng công cụ sẽ giúp tối ưu hóa chi phí và thời gian cho dự án kiểm thử tự động.

1. **Phát triển và tự động hóa kịch bản kiểm thử**

Sau khi có công cụ và môi trường, nhóm kiểm thử sẽ:

* Tiến hành viết mã kiểm thử tự động hoặc thiết lập các bước kiểm thử bằng giao diện tùy thuộc vào loại công cụ.
* Chuyển các kịch bản kiểm thử từ bản thiết kế sang các đoạn mã hoặc test script.
* Áp dụng các kỹ thuật kiểm thử như:
* Data-Driven Testing: kiểm thử với nhiều bộ dữ liệu khác nhau.
* Keyword-Driven Testing: kiểm thử theo các từ khóa đại diện cho hành động.
* Modular Testing: chia nhỏ kịch bản kiểm thử thành các module độc lập để dễ quản lý và tái sử dụng.

Đồng thời tổ chức các test suite để nhóm các test case cùng loại với nhau, phục vụ cho quá trình chạy kiểm thử tự động.

1. **Thực thi và theo dõi kết quả kiểm thử**

Ở bước này, các test suite và test case sẽ được chạy tự động:

* Thực thi các kịch bản kiểm thử tự động đã phát triển.
* Theo dõi quá trình thực thi và ghi lại log, báo cáo kết quả kiểm thử.
* Phân tích kết quả thực thi để xác định các lỗi phát sinh, điểm chưa đạt hoặc các bất thường.
* Thực hiện báo cáo chi tiết cho nhóm phát triển về các lỗi gặp phải và phối hợp xử lý.

Việc chạy kiểm thử tự động giúp phát hiện lỗi nhanh chóng và liên tục sau mỗi lần cập nhật hệ thống.

1. **Bảo trì và tối ưu hóa kiểm thử tự động**

Cuối cùng, để đảm bảo bộ kiểm thử luôn hoạt động chính xác:

* Cập nhật, chỉnh sửa các test script khi phần mềm có sự thay đổi về chức năng hoặc giao diện.
* Xóa bỏ những kịch bản không còn cần thiết, thêm mới những kịch bản cho chức năng mới.
* Tối ưu hóa test case, test suite để giảm thời gian thực thi và tăng độ ổn định.
* Đánh giá hiệu quả của quá trình kiểm thử tự động để có các điều chỉnh phù hợp.

Bảo trì kiểm thử tự động là một công việc định kỳ và rất quan trọng để đảm bảo tính chính xác và khả năng tái sử dụng của hệ thống kiểm thử.

2.1.4 Ưu, nhược điểm kiểm thử tự động

**a. Ưu điểm:**

* Độ tin cậy cao: Công cụ kiểm thử tự động có sự ổn định cao vì được hoạt động dựa trên một quy trình sẵn có. Đặc biệt, trong trường hợp nhiều test case sẽ tránh được tình trạng lỗi do con người tạo ra khi tiến hành kiểm tra thủ công.
* Khả năng lặp: Bạn có thể dễ dàng test các phần mềm xử lý (tính năng/ hiệu năng) khi gặp tình huống chạy lặp đi lặp lại nhiều lần trên cùng script test. Điều này giúp các tester xử lý được trường hợp lặp lại các thao tác như nhập dữ liệu, check kết quả, click,… Đây còn được gọi là performance hoặc Load testing.
* Khả năng tái sử dụng: Với automation testing bạn có thể tái sử dụng ứng dụng trên nhiều phiên bản ngay cả khi có sự thay đổi giao diện. Có thể test ở nhiều môi trường khác như như: beta, production,..
* Tốc độ cao: Automation test giúp chạy test với tốc độ nhanh hơn ít nhất 10 lần so với chạy thủ công bằng tay. Bạn chỉ cần mất khoảng 30s để thực thi việc này một cách tự động thay vì mất 5 phút cho test tay thủ công.
* Chi phí thấp: Nếu biết áp dụng kiểm thử tự động đúng cách, đúng mục đích thì bạn có thể tiết kiệm được rất nhiều chi phí cả về thời gian và nhân lực. Bởi kiểm thử tự động nhanh hơn nhiều so với thủ công. Hơn nữa, nhân lực cần để thực hiện và bảo trì scripts không nhiều.

**b. Nhược điểm:**

* Khó mở rộng, khó bảo trì: Để mở rộng phạm vi cho kiểm thử tự động khó hơn nhiều so với thủ công trong cùng một dự án. Nguyên nhân là bởi vì cập nhật hay chỉnh sửa yêu cầu nhiều công việc như debug, thay đổi dữ liệu đầu vào và cập nhật code mới.
* Khả năng bao phủ thấp: Do đòi hỏi nhiều kỹ năng lập trình và khó mở rộng, nên độ bao phủ của kiểm thử tự động khi xét trên góc nhìn toàn dự án sẽ thấp hơn
* Đối với vấn đề công cụ và nhân lực: Hiện nay, mặc dù có rất nhiều công cụ kiểm thử tự động nhằm hỗ trợ được đánh giá cao, tuy nhiên vẫn còn nhiều hạn chế. Ngoài ra, nhân lực đạt yêu cầu cũng không nhiều.
* Tốn thời gian: Để có thể áp dụng một cách thành thạo các công cụ kiểm thử đòi hỏi thời gian chuẩn bị dài, thiết kế, cài đặt kỹ càng trước khi chạy dự án.

2.2 Giới thiệu các công cụ kiểm thử:

2.2.1 Selenium

Selenium là một framework kiểm thử tự động mã nguồn mở, được sử dụng phổ biến để kiểm thử các ứng dụng web trên nhiều trình duyệt và nền tảng khác nhau. Không giống như các framework kiểm thử chạy trong trình duyệt như Cypress, Selenium điều khiển trình duyệt thông qua WebDriver, cho phép tương tác với trang web tương tự như người dùng thật, thực hiện các thao tác như click, nhập liệu, kiểm tra dữ liệu hiển thị, và xác thực nội dung trang web.

Selenium đặc biệt thích hợp cho các dự án cần kiểm thử chức năng, kiểm thử giao diện người dùng (UI Testing) và kiểm thử End-to-End (E2E), nhất là với các website thương mại điện tử như Chus — nơi có nhiều thao tác và chức năng người dùng cần kiểm tra.

Đặc điểm nổi bật:

* Hỗ trợ đa
* Điều khiển trình duyệt qua WebDriver: Selenium sử dụng WebDriver API để điều khiển trình trình duyệt: Selenium hỗ trợ kiểm thử trên nhiều trình duyệt phổ biến như Chrome, Firefox, Edge, Safari…, giúp kiểm tra khả năng tương thích giao diện và tính năng.duyệt từ bên ngoài, mô phỏng hành vi người dùng thật như click, nhập văn bản, kéo thả…
* Hỗ trợ nhiều ngôn ngữ lập trình: Selenium cho phép viết kịch bản kiểm thử bằng các ngôn ngữ phổ biến như Java, Python, C#, JavaScript, Ruby…, thuận tiện cho việc tích hợp vào các dự án khác nhau.
* Tích hợp CI/CD: Selenium dễ dàng tích hợp với các công cụ quản lý kiểm thử và hệ thống CI/CD như Jenkins, GitLab CI, Bamboo…, hỗ trợ kiểm thử liên tục.
* Thư viện hỗ trợ phong phú: Với cộng đồng người dùng đông đảo, Selenium có rất nhiều thư viện và tài liệu hướng dẫn, plugin, giúp mở rộng tính năng kiểm thử và gỡ lỗi.
* Có thể kết hợp với nhiều công cụ khác: Selenium có thể kết hợp với TestNG, JUnit (Java) hoặc PyTest (Python) để tổ chức và quản lý các test case, đồng thời tích hợp với Allure hoặc ExtentReports để tạo báo cáo kết quả kiểm thử chuyên nghiệp.

2.3 Các kiến trúc của các Framework kiểm thử tự động

2.3.1 Định nghĩa

Framework là tập hợp các đoạn mã, thư viện, mô hình thiết kế, API và các quy tắc đã được đóng gói sẵn nhằm hỗ trợ lập trình viên hoặc kiểm thử viên trong quá trình phát triển và kiểm thử phần mềm. Framework giúp tiêu chuẩn hóa quy trình, giảm thiểu thời gian và công sức cho các thao tác lặp lại, đồng thời đảm bảo tính nhất quán và dễ bảo trì cho dự án.

Nếu ví von, framework giống như bộ khung nhà đã xây sẵn nền móng, cột trụ — lập trình viên và tester chỉ cần xây dựng tiếp phần nội thất, trang trí và chức năng theo ý muốn mà không phải thiết kế lại từ đầu.

2.3.2 Ưu, nhược điểm

**a. Ưu điểm**

|  |  |
| --- | --- |
| Ưu điểm | Mô tả |
| Tiết kiệm thời gian và công sức | Framework cung cấp sẵn các thành phần, thư viện và công cụ giúp lập trình viên, kiểm thử viên không phải xây dựng mọi thứ từ đầu |
| Tăng hiệu suất làm việc | Giúp tăng tốc quá trình phát triển và kiểm thử phần mềm nhờ các tính năng hỗ trợ và các công cụ tích hợp sẵn. |
| Tính tái sử dụng cao | Các đoạn mã và thành phần trong framework có thể tái sử dụng nhiều lần cho các dự án khác nhau. |
| Giảm thiểu lỗi phát sinh | Do sử dụng các thành phần chuẩn hóa, đã được kiểm tra kỹ lưỡng nên giảm thiểu rủi ro lỗi và tăng độ ổn định của sản phẩm.  Hỗ trợ bảo trì, mở rộng dễ dàng  Kiến trúc rõ ràng, phân tách các lớp, các module giúp việc bảo trì và mở rộng hệ thống dễ dàng hơn. |
| Cộng đồng hỗ trợ lớn | Hầu hết các framework phổ biến đều có cộng đồng người dùng và tài liệu phong phú, dễ tìm kiếm khi gặp sự cố hoặc cần học hỏi. |

**b. Nhược điểm**

|  |  |
| --- | --- |
| Nhược điểm | Mô tả |
| Hạn chế về tính linh hoạt | Do phải tuân theo kiến trúc và quy tắc của framework nên đôi khi khó tùy biến cho các dự án quá đặc thù. |
| Tốn thời gian học tập ban đầu | Để sử dụng thành thạo framework, lập trình viên và kiểm thử viên phải dành thời gian tìm hiểu cấu trúc, cách vận hành và các tính năng của framework đó. |
| Hiệu năng bị ảnh hưởng nếu framework cồng kềnh | Một số framework có thể làm giảm hiệu suất ứng dụng nếu không tối ưu hóa tốt hoặc quá nặng nề. |
| Phụ thuộc vào framework | Khi đã xây dựng hệ thống dựa trên một framework cụ thể, việc chuyển sang framework khác có thể gây khó khăn, mất thời gian và tốn kém |
| Cập nhật và tương thích | Các framework thường xuyên nâng cấp phiên bản. Nếu không theo kịp, có thể dẫn đến lỗi tương thích với dự án cũ. |

2.3.3 Một số framework kiểm thử tự động

* ***Linear Scripting Framework***

Linear Scripting Framework là một khung tự động hóa kiểm tra mức cơ bản ở dạng 'Ghi và Phát lại' theo kiểu tuyến tính. Khung này còn được gọi là khung 'Ghi và Phát lại'. Loại khuôn khổ này được sử dụng để kiểm tra các ứng dụng có kích thước nhỏ. Trong loại này, việc tạo và thực thi các tập lệnh thử nghiệm được thực hiện riêng lẻ cho từng trường hợp thử nghiệm riêng lẻ. Người kiểm tra nắm bắt từng bước kiểm tra như duyệt, điều hướng, đầu vào của người dùng, thực thi các điểm kiểm tra. Sau đó, người kiểm tra sẽ phát các tập lệnh để thực hiện các bài kiểm tra.

* ***Module-Based Testing Framework***

Với framework này, thì ta sẽ xây dựng các test script độc lập, tương ứng với từng module, các compoment hoặc các function của phần mềm ứng dụng. Việc tránh sử dụng các script phụ thuộc nhau là một yếu tố quan trọng đối với sự ổn định và khả năng bảo trì của framework này.

Nói một cách dễ hiểu, người kiểm thử chia ứng dụng thành nhiều mô-đun và tạo các tập lệnh kiểm tra riêng lẻ. Các tập lệnh thử nghiệm riêng lẻ này có thể được kết hợp để tạo ra các tập lệnh thử nghiệm lớn hơn bằng cách sử dụng một tập lệnh chính để đạt được các kịch bản yêu cầu. Tập lệnh chính này được sử dụng để gọi các mô-đun riêng lẻ để chạy các kịch bản thử nghiệm từ đầu đến cuối.

Lý do chính để sử dụng khuôn khổ này là xây dựng một lớp trừu tượng để bảo vệ mô-đun chính khỏi bất kỳ thay đổi nào được thực hiện trong các thử nghiệm riêng lẻ.

* ***Library Architecture Tesitng Framework***

Khung kiểm tra kiến ​​trúc thư viện hay còn gọi là “Kịch bản có cấu trúc” hoặc “Phân rã chức năng” nó dựa trên khuôn khổ mô-đun (Module Based Framework) với một số lợi thế bổ sung. Trong khuôn khổ kiểm thử mô-đun, chúng ta chia ứng dụng đang được kiểm tra thành các mô-đun trong khi ở đây chúng ta xác định các nhiệm vụ chung và nhóm chúng thành các chức năng. Khi các chức năng được nhóm lại thì các nhóm này sẽ được lưu giữ trong thư viện. Các kịch bản thử nghiệm sử dụng lại các thư viện này để tạo các trường hợp thử nghiệm mới.

2.4 Test Script và phương pháp viết Test Script

2.4.1 Định nghĩa

Test Script là một tập hợp các câu lệnh chi tiết được lập trình nhằm tự động hóa quy trình kiểm thử phần mềm. Mục đích của Test Script là mô phỏng và thực hiện các thao tác của người dùng trên phần mềm để kiểm tra tính đúng đắn, độ ổn định và khả năng phản hồi của hệ thống.

**Đặc điểm của Test Script:**

* Thường được lập trình bằng ngôn ngữ hỗ trợ của công cụ kiểm thử (Java, Python, C#, Groovy…)
* Tự động hóa các hành động như click, nhập liệu, chuyển trang, xác minh kết quả…
* Chuyển các bước Test Case thủ công thành mã hóa tự động

2.4.2 Mục tiêu của viết Test Script

* Tự động hóa kiểm thử lặp lại, giảm thiểu công việc thủ công.
* Tăng hiệu quả và tính chính xác trong quá trình kiểm thử.
* Giảm chi phí và nhân sự kiểm thử, đặc biệt với các dự án lớn, nhiều môi trường.
* Hỗ trợ kiểm thử hồi quy (Regression Testing) nhanh chóng khi có phiên bản cập nhật.
* Tích hợp vào quy trình CI/CD (Continuous Integration/Continuous Delivery), giúp kiểm thử liên tục.
* Đảm bảo chất lượng phần mềm ở mức cao nhất nhờ phát hiện lỗi sớm.

2.4.3 Phương pháp viết Test Script

+ Bước 1: Phân tích Test Case và Thiết kế Precondition

* + Xác định chức năng cần test.
  + Xác định điều kiện ban đầu (Precondition): trình duyệt mở, người dùng chưa đăng nhập,...
  + Viết mã khởi tạo driver, mở URL, và chờ page load.

+ Bước 2: “Code hóa” các bước kiểm thử

* Mỗi bước trong test case được “dịch” thành mã:
* Nhập dữ liệu: sendKeys
* Tương tác UI: click, select, hover...
* Chờ load hoặc xử lý bất đồng bộ: wait, sleep, fluentWait...

+ Bước 3: Xác minh kết quả (Verification/Validation)

Dùng các lệnh kiểm tra:

* Assert.assertEquals() – Kiểm tra kết quả mong đợi và thực tế.
* Assert.assertTrue() – Kiểm tra điều kiện có đúng không.

Checkpoint nên đặt ở các phần ổn định như: tiêu đề trang, thông báo xác nhận...

+ Bước 4: Xử lý dữ liệu kiểm thử (Test Data)

Test data nên tách khỏi mã nguồn, lưu ở:

* File Excel, JSON, CSV
* Database

Tránh hard-code dữ liệu vào script.

+ Bước 5: Tạo báo cáo và log

Ghi lại:

* Kết quả pass/fail từng bước.
* Ảnh chụp màn hình nếu lỗi.
* Ghi log chi tiết bằng thư viện log (log4j, slf4j…).

Tùy chỉnh báo cáo bằng: Allure, ExtentReport, TestNG Reports…

+ Bước 6: Dọn dẹp (Teardown)

Sau khi kiểm thử:

* Đóng trình duyệt (driver.quit())
* Hủy kết nối DB (nếu có)
* Xóa dữ liệu test ảo nếu cần

2.4.4 Nguyên tắc viết Test Script tốt

|  |  |
| --- | --- |
| Nguyên tắc | Giải thích |
| Rõ ràng | Code dễ đọc, tên biến rõ ràng, chia hàm hợp lý. |
| Tái sử dụng | Tách cách hành vi dùng lại nhiều lần vào hàm hoặc Page Object. |
| Thêm sử lý lỗi | Dùng try-catch, chụp ảnh khi lỗi, timeout chờ xử lý. |
| Bảo trì trễ | Tách phần tử UI ra file riêng hoặc Page Object để dễ cập nhật |
| Đảm bảo độc lập | Mỗi test case nên chạy độc lập, không phụ thuộc lẫn nhau. |

CHƯƠNG 3: PHÂN TÍCH YÊU CẦU VÀ LẬP KẾ HOẠCH KIỂM THỬ

3.1 Phân tích yêu cầu kiểm thử

Bảng 3. 1: Danh sách yêu cầu chức năng

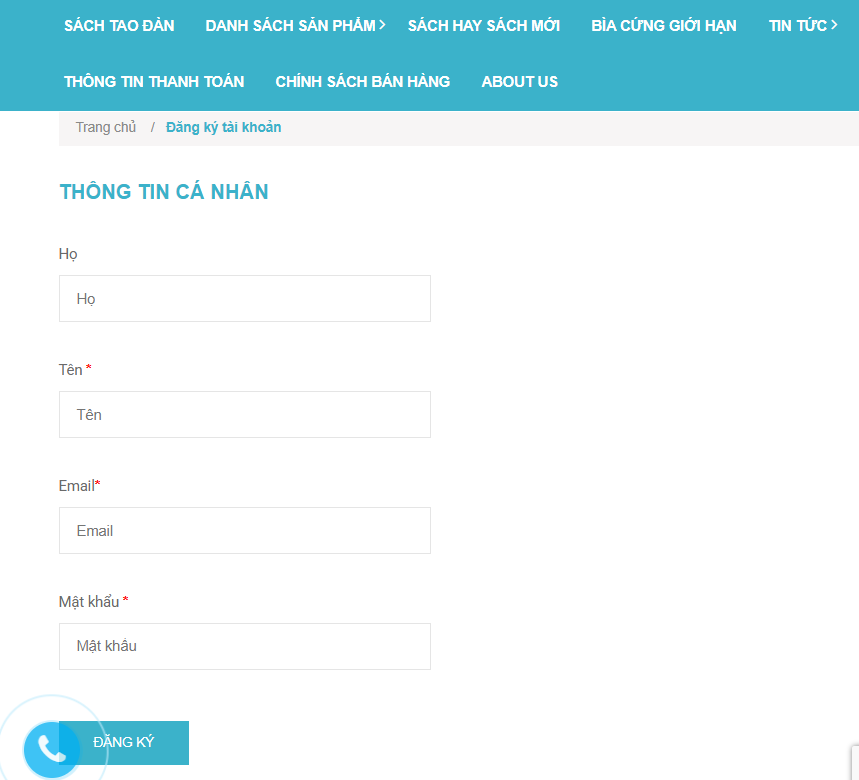
|  |  |  |
| --- | --- | --- |
| Mục | Tên chức năng | Mô tả |
| 1 | Đăng ký | Chức năng đăng ký cho phép người dùng đăng ký tài khoản mới nếu chưa có tài khoản bao gồm các trường Họ, tên, email, mật khẩu để sử dụng các dịch vụ đặt hàng, theo dõi đơn hàng của cửa hàng. |
| 2 | Đăng nhập | Chức năng đăng nhập dùng đăng nhập hệ thống khi người dùng đã đăng ký tài khoản để đặt hàng, theo dõi đơn hàng, quản lý thông tin cá nhân, sổ địa chỉ. |
| 3 | Đăng xuất | Chức năng đăng xuất trên website Chus dùng để thoát khỏi tài khoản người dùng. Khi bạn hoàn tất việc sử dụng tài khoản, việc đăng xuất giúp bảo vệ thông tin cá nhân và bảo mật tài khoản, đặc biệt nếu bạn đang sử dụng thiết bị công cộng hoặc chia sẻ với người khác. Sau khi đăng xuất, bạn sẽ không còn quyền truy cập vào các tính năng của tài khoản cho đến khi đăng nhập lại. |
| 4 | Tìm kiếm sản phẩm | Cho phép người dùng tìm kiếm sản phẩm bằng từ khóa qua thanh tìm kiếm, danh mục sản phẩm, danh mục tin tức. Kết quả hiển thị bao gồm: ảnh bìa, tên sách, giá bán, khuyến mại. |
| 5 | Chi tiết sản phẩm | Cho phép người dùng xem chi tiết sản phẩm bao gồm: ảnh minh họa, tên sách, tình trạng hàng, giá sách, thông tin sản phẩm. |
| Mục | Tên chức năng | Đăng nhập |
| 6 | Giỏ hàng | Cho phép người dùng quản lý giỏ hàng:   * Tăng/giảm số lượng sản phẩm * Xem danh sách các sản phẩm trong giỏ hàng * Xóa sản phẩm khỏi giỏ hàng * Mua hàng |
| 7 | Đặt hàng | Chức năng đặt hàng cho phép người dùng đặt hàng sau khi điền đầy đủ thông tin mua hàng bao gồm: họ và tên, số điện thoại, địa chỉ, cách thanh toán. |
| 8 | Trang tài khoản | Chức năng này cho phép người dùng quản lý thông tin tài khoản của mình như sổ địa chỉ, thông tin đơn hàng. |

3.1.1 Yêu cầu chức năng

Website Chus gồm 8 chức năng chính là: Đăngký, Đăng nhập, Giỏ hàng, Chi tiết sản phẩm, Tìm kiếm, Đặt hàng, Quên mật khẩu, Trang tài khoản, Đăng xuất.

1. **Chức năng Đăng ký**

Chức năng đăng ký trên website Chus giúp người dùng tạo tài khoản mới để truy cập và sử dụng các dịch vụ của trang web.



Hình 3. 1: Giao diện Chức năng Đăng ký

* **Input**

+ Họ và tên: Không được để trống hoặc toàn kí tự trắng và số lượng kí tự phải ít hơn hoặc bằng 255. Có thể chứa chữ cái thường, chữ cái in hoa, số, kí tự đặc biệt.

+ Địa chỉ email: Không được để trống hoặc toàn kí tự trắng và số lượng kí tự phải ít hơn hoặc bằng 255 và phải chứa địa chỉ email hợp lệ. Có thể chứa chữ cái thường, chữ cái in hoa, số, kí tự đặc biệt.

+ Mật khẩu: Không được để trống hoặc toàn kí tự trắng và số lượng kí tự phải lớn hơn hoặc bằng 6 kí tự nhỏ hơn 24. Có thể chứa chữ cái thường, chữ cái in hoa, số, kí tự đặc biệt.

+ Nhập lại mật khẩu: Phải giống hệt mật khẩu không được để trống hoặc toàn kí tự trắng. Có thể chứa chữ cái thường, chữ cái in hoa, số, kí tự đặc biệt.

* **Action**:

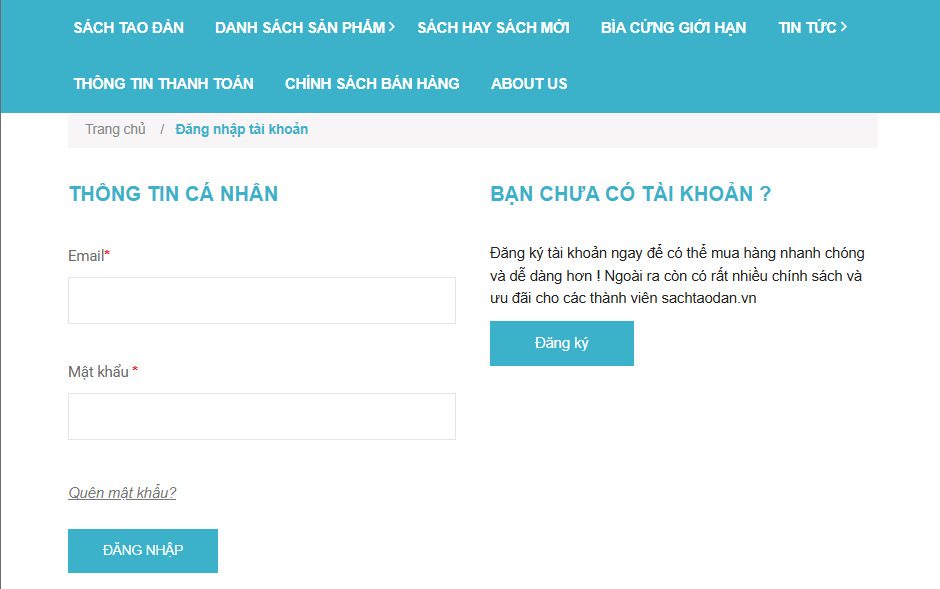
+ Đăng kí nhận thông tin qua email: có thể chọn hoặc không chọn.

+ Nút “Đăng ký” dùng để đăng ký tài khoản khi chưa có tài khoản, khi đăng ký thành công sẽ chuyển sang giao diện Trang chủ nếu Đăng ký thất bại hệ thống thông báo lỗi.

+ Nút Đăng nhập: dùng để đăng nhập nếu đã có tài khoản truy cập hệ thống, khi đăng nhập thành công sẽ chuyển sang giao diện Trang chủ nếu Đăng nhập thất bại hệ thống thông báo lỗi.

1. **Chức năng Đăng nhập**

Chức năng Đăng nhập trên website Chus giúp người dùng truy cập vào tài khoản cá nhân của họ để đặt hàng, theo dõi đơn hàng.



Hình 3. 2: Giao diện Chức năng Đăng nhập

* **Input**

+ Email: Không được để trống hoặc toàn kí tự trắng và số lượng kí tự phải ít hơn hoặc bằng 255. Có thể chứa chữ cái thường, chữ cái in hoa, số, kí tự đặc biệt và phải là địa chỉ email đã được đăng kí trên website Chus.

+ Mật khẩu: Không được để trống hoặc toàn kí tự trắng và sô lượng kí tự phải lớn hơn hoặc bằng 6 kí tự và nhỏ hơn 24 ký tự. Có thể chứa chữ cái thường, chữ cái in hoa, số, kí tự đặc biệt và phải là mật khẩu đã sử dụng khi đăng kí email trên website Chus.

* **Action**:

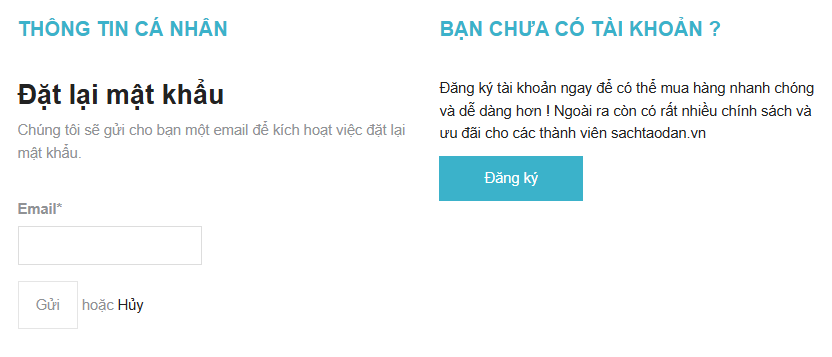
+ Nút “Đăng ký” dùng để đăng ký tài khoản khi chưa có tài khoản, khi đăng ký thành công sẽ chuyển sang giao diện Trang chủ nếu Đăng ký thất bại hệ thống thông báo lỗi.

+ Nút Đăng nhập: dùng để đăng nhập nếu đã có tài khoản truy cập hệ thống, khi đăng nhập thành công sẽ chuyển sang giao diện Trang chủ nếu Đăng nhập thất bại hệ thống thông báo lỗi.

+ Quên tài khoản: dùng để thiết lập mật khẩu mới khi bị quên mật khẩu.

1. **Chức năng Quên mật khẩu**

Chức năng quên mật khẩu trên website Chus giúp người dùng khôi phục mật khẩu khi quên.



Hình 3. 3: Giao diện Chức năng Quên mật khẩu

* **Input**

+ Email: Email: Không được để trống hoặc toàn kí tự trắng và số lượng kí tự phải ít hơn hoặc bằng 255. Có thể chứa chữ cái thường, chữ cái in hoa, số, kí tự đặc biệt và phải là địa chỉ email đã được đăng kí trên website Chus.

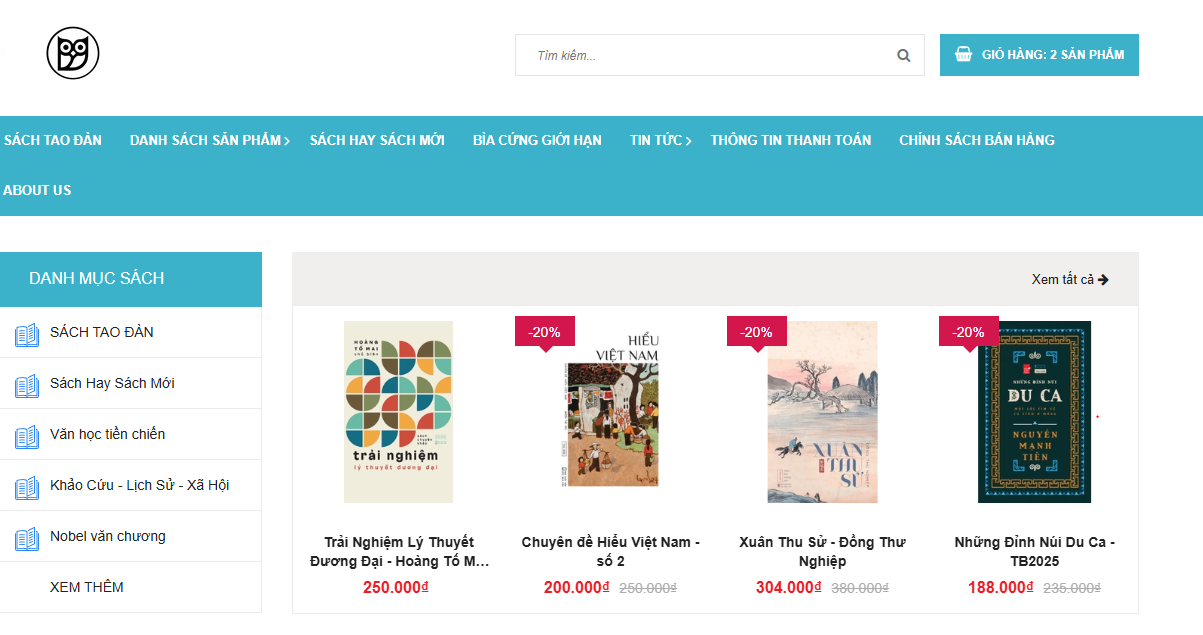
* **Action**

+ Nút “Gửi” : dùng để gửi yêu cầu về hệ thống nếu địa chỉ Email đúng thì sẽ gửi yêu cầu về hệ thống thành công nếu Email sai thông báo lỗi.

+ Nút “Hủy”: dùng để hủy yêu cầu khi không cần thiết lập mật khẩu mới khi quên mật khẩu.

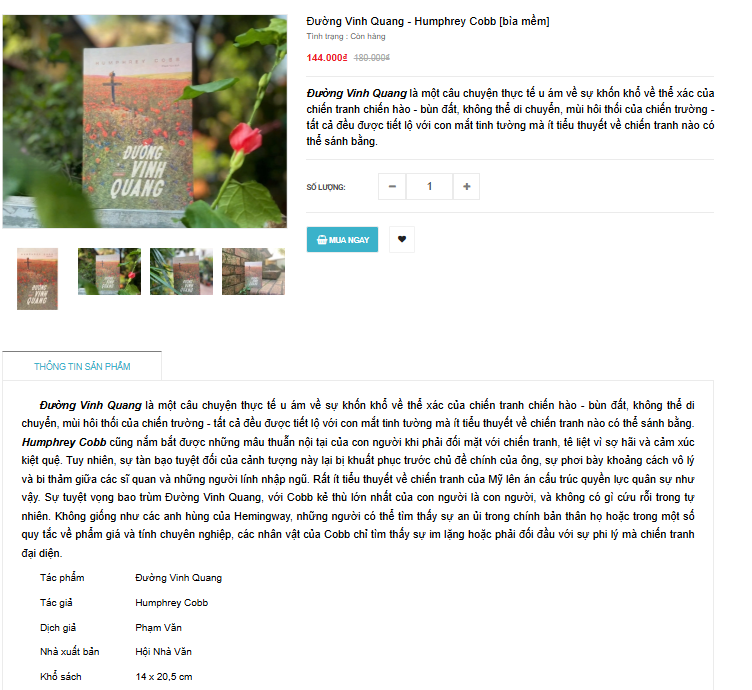
1. **Chức năng Tìm kiếm**

Cho phép người dùng tìm kiếm sản phẩm bằng từ khóa qua thanh tìm kiếm



Hình 3. 4: Giao diện Chức năng Tìm kiếm

1. **Chức năng chi tiết**



Hình 3. 5: Giao diện Chức năng Chi tiết

* **Input**

Chức năng Xem chi tiết sản phẩm cho phép người dùng xem chi tiết sản phẩm bao gồm: tên sách, tình trạng (còn hàng/hết hàng), giá bán.

* **Action**

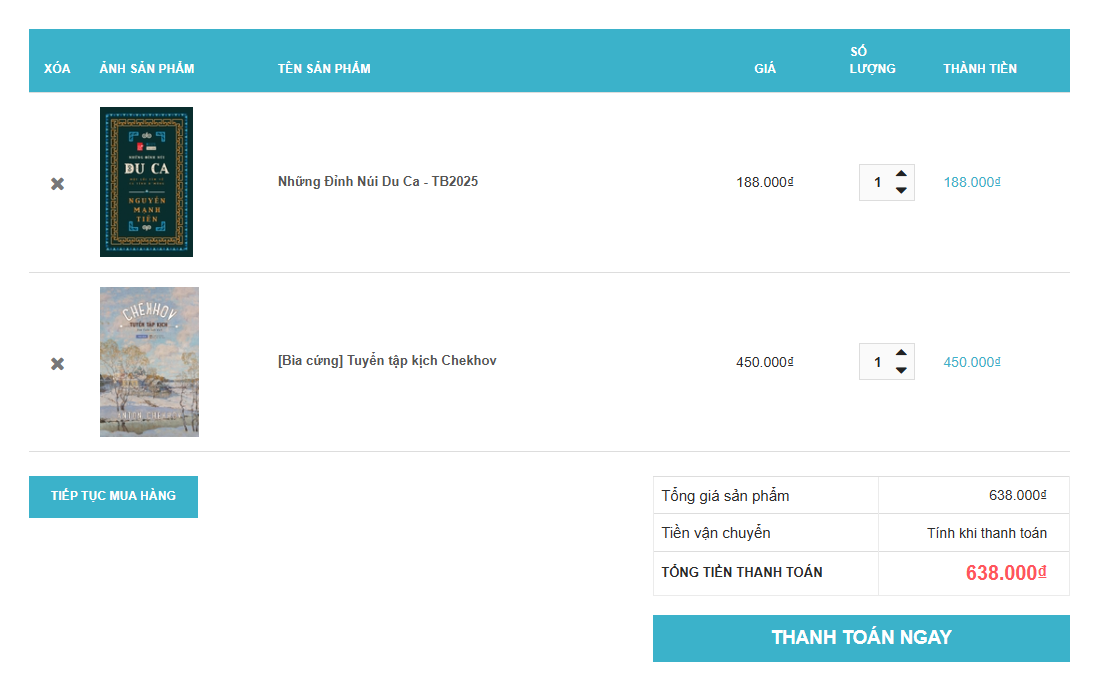
+ Nút + để thêm số lượng.

+ Nút – để giảm số lượng

+ Nút Mua ngay để thêm sản phẩm vào giỏ hàng hoặc tiến hành thanh toán.

1. **Chức năng giỏ hàng**

Chức năng giỏ hàng trên website Chus giúp người dùng lưu trữ và quản lý các sản phẩm mà họ muốn mua.



Hình 3. 6: Giao diện Chức năng Giỏ hàng

Mô tả chi tiết chức năng:

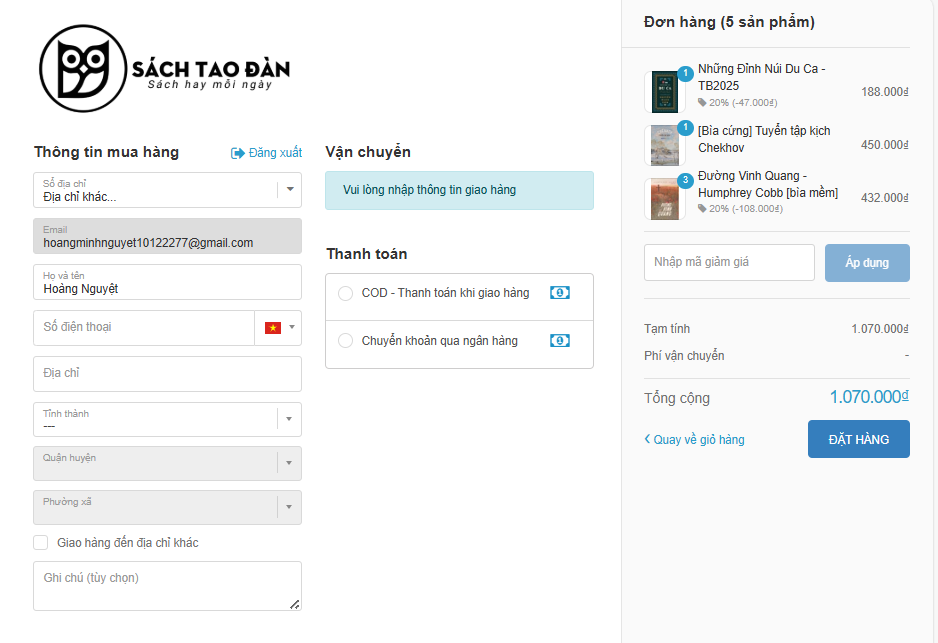
+ Số lượng: phải lớn hơn hoặc bằng 1 và phải nhỏ hơn tổng số lượng còn trong kho.

+ Nút “Tiếp tục mua sắm” : dùng dể chuyển sang giao diện Tất cả sản phẩm.

+ Nút “Xóa” : dùng đẻ xóa sản phẩm khỏi giỏ hàng khi không có nhu cầu mua.

+ Nút “Thanh toán ngay” : dùng để đặt hàng.

1. **Chức năng Đặt hàng**



Hình 3. 7: Giao diện Chức năng Đặt hàng

Chức năng Đặt hàng cho phép người dùng mua sắm trực tuyến qua hệ thống.Người dùng chọn sản phẩm muốn mua tại giỏ hàng, đuền đầy đủ các thông tin mua hàng, chọn phương thức thanh toán, nhập mã giảm giá nếu có, rồi chọn đặt hàng.

+ Họ và tên: Nhập đúng tên người nhận, không được để trống hoặc toàn khoảng trắng có thể chứa chữ hoa, chữ thường, số, ký tư đặc biệt.

+ SĐT: nhập đúng SĐT nhận hàng, không để trống hoặc toàn khoảng trắng, gồm 10 số và bắt đầu bằng số 0, không có chữ cái hay ký tự đặc biệt.

+ Địa chỉ: nhập đúng địa chỉ nhận hàng, không để trống hoặc chỉ toàn khoảng trắng, tuân thủ về độ dài và ký tự đặc biệt.

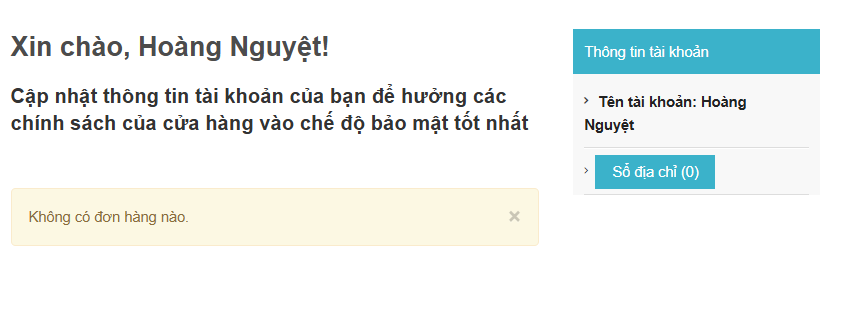
+ Tỉnh thành/Quận huyện/Phường xã: không bỏ trống chọn đúng địa chỉ nhận hàng.

+ Ghi chú: nhập vào nếu có.

+ Mã giảm giá: nhập vào nếu có.

+ Thanh toán: có 2 phương thức cod hoặc chuyển khoản qua ngân hàng.

1. **Chức năngTrang tài khoản**



Hình 3. 8: Giao diện Chức năng Thông tin tài khoản

Chức năng này cho phép người dùng quản lý thông tin tài khoản của mình như sổ địa chỉ, thông tin đơn hàng.

+ Sổ địa chỉ: Cho phép người dùng thêm địa chỉ nhận hàng.



Hình 3. 9: Giao diện Thêm sổ địa chỉ

3.1.2 Yêu cầu phi chức năng

* Hiệu năng (Performance): Hệ thống phải phản hồi trong vòng 3 giây với các chức năng chính, khi có tối đa 20 người dùng đồng thời.
* Bảo mật (Security): Thông tin đăng nhập, tài khoản và đơn hàng của người dùng phải được bảo vệ an toàn, không để lộ dữ liệu. Các trường email, mật khẩu cần được kiểm tra hợp lệ và mã hóa trong hệ thống.
* Khả năng sử dụng (Usability): Giao diện thân thiện, dễ thao tác với người dùng phổ thông; các nút chức năng dễ nhận biết. Phông chữ và màu sắc đảm bảo dễ đọc.
* Tính tương thích (Compatibility): Website phải hoạt động tốt trên các trình duyệt phổ biến như Chrome, Firefox, Edge, độ phân giải tối thiểu 1366x768.
* Khả năng mở rộng (Scalability): Hệ thống cần cho phép mở rộng chức năng (ví dụ: thêm phương thức thanh toán, tích hợp vận chuyển) mà không ảnh hưởng đến tính ổn định hiện tại.
* Độ tin cậy (Reliability): Website phải duy trì hoạt động liên tục với tỷ lệ lỗi thấp; không được xảy ra lỗi nghiêm trọng gây gián đoạn giao dịch.
* Khảnăng phục hồi (Recoverability): Khi có sự cố mất kết nối tạm thời, hệ thống cần có khả năng khôi phục dữ liệu giỏ hàng và phiên người dùng nếu có thể.

3.1.3 Giới hạn và phạm vi đề tài

Trong phạm vi đồ án này, nhóm tập trung triển khai kiểm thử tự động cho các chức năng cốt lõi sau:

* Đăng ký tài khoản
* Đăng nhập tài khoản
* Tìm kiếm sản phẩm
* Đặt hàng mua sản phẩm

Các chức năng như đăng xuất, quản lý giỏ hàng, xem chi tiết sản phẩm, quên mật khẩu và quản lý tài khoản chưa được kiểm thử tự động trong đồ án này, tuy nhiên em có kế hoạch mở rộng kiểm thử ở giai đoạn tiếp theo nếu có thời gian và tài nguyên phù hợp.

3.2 Kế hoạch kiểm thử

3.2.1 Lịch trình công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Các mốc quan trọng | Mục tiêu | Thời gian | Ngày bắt đầu | Ngày kết thúc |
| Tìm hiểu, lựa chọn đề tài môi trường và công cụ kiểm thử | Công cụ đã được cài đặt |  | 10/02/2025 | 16/02/2025 |
| Mô tả các chức năng chính của website | Tài liệu Phân tích yêu cầu chức năng |  | 17/02/2025 | 17/2/2025 |
| Viết TestPlan | Tài liệu TestPlan |  | 18/2/2025 | 23/2/2025 |
| Rà soát lại các tài liệu | Báo cáo đồ án |  | 24/2/2025 | 25/2/2025 |
| Viết testcases kiểm thử chức năng | Tài liệu Testcase |  | 26/2/2025 | 30/4/2025 |
| Thực thi các testcase bằng công cụ kiểm thử tự động | Các script và report của công cụ |  | 11/5/2025 | 14/5/2025 |
| Tổng kết và ghi nhận các kết quả kiểm thử thủ công và tự động | Tài liệu Testcase, script và report |  | 15/5/2025 | 18/5/2025 |

3.2.1. Phương pháp kiểm thử

Bảng 3. 2: Phương pháp kiểm thử

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đảm bảo các chức năng được kiểm tra hoạt động chính xác theo đặc tả yêu cầu |
| Kỹ thuật | Thực thi tất cả các trường hợp có thể có cho mỗi nhóm chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định: - Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng - Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng |
| Tiêu chuẩn dừng | Tất cả các testcase đã được thiết kế đều được thực thi. Tất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho developer khắc phục. |
| Chịu trách nhiệm kiểm thử | Tester |
| Cách kiểm thử | - Kiểm thử tự động, dùng Selenium + Java - Mô hình: POM (Page Object Model) - Framework: tự xây dựng, hướng data-driven |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử. |

3.2.2 Môi trường kiểm thử

Bảng 3. 4 Môi trường kiểm thử

| Nội dung | Mô tả |
| --- | --- |
| Hệ điều hành | Windows 10 |
| Phần cứng | CPU: Intel Core i7, Intel iRISx  RAM: 8G  Ổ cứng: SSD 256G |
| Công cụ kiểm thử | Selenium WebDriver |
| Framework tổ chức | TestNG |
| Công cụ quản lý dependency | Maven/NuGet |
| Trình duyệt web | Google Chorme, Mozilla Firefox, Microsoft Edge |
| Công cụ báo cáo | ExtentReports |
| Ngôn ngữ lập trình | Java |
| Quản lý mã nguồn | Github |
| IDE phát triển | IntelliJ IDEA |
| Đường truyền | Kết nối **Internet liên tục, ổn định** để đảm bảo quá trình kiểm thử các chức năng của website không bị gián đoạn. |
| Trang web kiểm thử | https://sachtaodan.vn/ |

3.3 Xây dựng Framework

3.3.1 Mục tiêu của Framework

Việc xây dựng một framework kiểm thử tự động là bước quan trọng giúp nâng cao hiệu quả kiểm thử phần mềm, đặc biệt đối với các website thương mại điện tử như Chus. Các mục tiêu chính của framework bao gồm:

* Tự động hóa kiểm thử chức năng: Hỗ trợ kiểm thử các chức năng cốt lõi như đăng ký, đăng nhập, tìm kiếm, giỏ hàng, đặt hàng... giúp tiết kiệm thời gian và giảm công sức so với kiểm thử thủ công.
* Hỗ trợ kiểm thử hướng dữ liệu (Data-Driven Testing): Cho phép chạy một test case với nhiều bộ dữ liệu khác nhau (như thông tin đăng nhập, địa chỉ, thông tin đặt hàng...), tăng độ phủ kiểm thử.
* Tái sử dụng mã nguồn và giảm trùng lặp: Áp dụng mô hình phân lớp và Page Object Model (POM) để tách riêng logic giao diện và kiểm thử, giúp mã dễ bảo trì, mở rộng.
* Tạo báo cáo kết quả kiểm thử rõ ràng: Tích hợp module ExcelReport giúp ghi lại kết quả test dưới dạng file .xlsx. Dù không trực quan như HTML Report (ExtentReports), vẫn phục vụ tốt việc tổng hợp và phân tích kết quả test.
* Khả năng mở rộng – bảo trì – tích hợp CI/CD: Cấu trúc framework linh hoạt, có thể dễ dàng mở rộng thêm chức năng, thêm case test mới hoặc tích hợp với Jenkins/GitHub Actions để kiểm thử tự động theo chu kỳ.

3.3.2 Mô hình kiến trúc áp dụng

Framework áp dụng mô hình Page Object Model (POM) – một mô hình thiết kế chuẩn trong kiểm thử tự động với Selenium, nhằm tổ chức mã nguồn rõ ràng, dễ bảo trì và dễ mở rộng.

Trong mô hình POM, mỗi trang hoặc chức năng của website được ánh xạ thành một lớp Java riêng (gọi là Page Object). Mỗi lớp này sẽ:

* Khai báo các phần tử giao diện web dưới dạng biến (ví dụ: nút đăng nhập, ô tìm kiếm...).
* Định nghĩa các hành động người dùng (như click, nhập dữ liệu, kiểm tra hiển thị,...) thành các phương thức tương ứng.

Lớp kiểm thử (test class) sẽ không thao tác trực tiếp với các phần tử web, mà chỉ gọi các phương thức đã được đóng gói trong Page Object. Cách tổ chức này giúp mã nguồn kiểm thử dễ đọc, dễ tái sử dụng và dễ bảo trì.

Luồng hoạt động tổng quát của framework:

Test Case (TestNG)

Page Object (LoginPage, HomePage)

age, ...)

Utils(ExcelReader,ExcelReport, ExcelLogger)

Base (BaseSetup.java khởi tạo WebDriver, teardown, ...)

Việc kết hợp mô hình POM với các lớp tiện ích (utils) và lớp nền (base) giúp tách biệt rõ ràng giữa các vai trò, hỗ trợ việc phát triển, bảo trì và mở rộng framework khi chức năng hệ thống thay đổi hoặc khi dự án được mở rộng.

3.3.3 Cấu trúc thư mục của framework

Framework được tổ chức theo cấu trúc thư mục sau:

DoAn3/

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ ├── base/ # Cấu hình và khởi tạo trình duyệt

│ │ │ │ ├── BaseSetup.java

│ │ │ │ └── QlWebdriver.java

│ │ │ ├── config/ # Lưu thông tin cấu hình như URL

│ │ │ │ └── AppURL.java

│ │ │ ├── pages/ # Các Page Object đại diện cho các trang web

│ │ │ │ ├── HomePage.java

│ │ │ │ ├── LoginPage.java

│ │ │ │ ├── OrderPage.java

│ │ │ │ ├── SearchPage.java

│ │ │ │ └── SignupPage.java

│ │ │ ├── utils/ # Các class tiện ích hỗ trợ

│ │ │ │ ├── DateTimeUtils.java

│ │ │ │ ├── ExcelHelpers.java

│ │ │ │ ├── ExcelLogger.java

│ │ │ │ ├── ExcelReader.java

│ │ │ │ └── ExcelReport.java

│ │ └── resources/ # File cấu hình bổ sung( chưa dùng)

│

├── test/

│ ├── java/

│ │ └── test/ # Chứa các class kiểm thử chính (TestNG)

│ │ ├── LoginTest.java

│ │ ├── OrderTest.java

│ │ ├── SearchTest.java

│ │ └── SignupTest.java

│ └── resources/ # Chứa dữ liệu test

│

├── test-output/ # Báo cáo sau khi chạy test

│ └── report.xlsx # File kết quả xuất từ ExcelReport

│

├── pom.xml # Quản lý thư viện (TestNG, Selenium, POI,...)

├── testng.xml # File khai báo suite test cho TestNG

└── .gitignore / target /

CHƯƠNG 4: TRIỂN KHAI KIỂM THỬ ỨNG DỤNG

4.1 Thiết kế kịch bản kiểm thử

4.1.1 Xây dựng mô hình Page Object cho website Chus

Mô hình Page Object Model (POM) được áp dụng trong framework nhằm tách biệt rõ ràng giữa phần mã kiểm thử (test logic) và phần mã xử lý giao diện người dùng (UI interaction). Mỗi trang (hoặc thành phần chức năng) trên website được ánh xạ với một lớp Java trong thư mục pages/, nơi chứa các phần tử giao diện (Web Elements) và phương thức thao tác với các phần tử đó.

**Ví dụ 1: Lớp LoginPage** quản lý các thành phần giao diện như trường nhập email, mật khẩu, nút đăng nhập và xử lý các tình huống đăng nhập thành công hoặc thất bại.

package pages;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
  
  
public class LoginPage {  
 WebDriver driver;  
 By emailInput = By.id("email");  
 By passwordInput = By.id("pass");  
 By loginButton = By.name("send");  
 By messageLabel = By.xpath("//li[contains(@style, 'color:red') and contains(@style, 'margin-bottom:10px')]");  
  
 public LoginPage(WebDriver driver) {  
 this.driver = driver;  
 }  
  
 public void enterEmail(String email) {  
 this.driver.findElement(this.emailInput).clear();  
 this.driver.findElement(this.emailInput).sendKeys(email);  
 }  
  
 public void enterPassword(String password) {  
 this.driver.findElement(this.passwordInput).clear();  
 this.driver.findElement(this.passwordInput).sendKeys(password);  
 }

Hình 4. 1: Mã nguồn lớp LoginPage đại diện trang đăng nhập trong mô hình Page Object Model (POM)

**Ví dụ 2: Lớp HomePage** quản lý các thao tác chính từ trang chủ của website Chus như tìm kiếm sách, chọn sách theo tiêu đề, thay đổi số lượng, thêm vào giỏ hàng, xác nhận popup hiển thị và chuyển đến trang thanh toán. Các thành phần giao diện như ô tìm kiếm, nút "Mua ngay", số lượng, popup, đều được định nghĩa và xử lý thông qua các phương thức tương ứng để mô phỏng hành vi người dùng.

package pages;  
  
import org.openqa.selenium.\*;  
import org.openqa.selenium.support.ui.\*;  
  
import java.time.Duration;  
  
public class HomePage {  
 private WebDriver driver;  
 private WebDriverWait wait;  
  
 private By searchBox = By.xpath("//input[@placeholder='Tìm kiếm...']");  
 private By quantityInput = By.id("qty"); // đúng theo HTML bạn cung cấp  
 private By increaseQtyBtn = By.cssSelector("button.increase.count");  
 private By decreaseQtyBtn = By.cssSelector("button.reduced.count");  
 private By addToCartBtn = By.xpath("//button[contains(.,'Mua ngay')]");  
 private By popupContent = By.xpath("//div[@class='content']");  
 private By proceedToCheckoutBtn = By.xpath("//button[contains(text(),'Tiến hành thanh toán')]");  
  
 public HomePage(WebDriver driver) {  
 this.driver = driver;  
 this.wait = new WebDriverWait(driver, Duration.ofSeconds(10));  
 }  
  
 // Gõ từ khóa tìm sách vào ô tìm kiếm và nhấn Enter  
 public void searchBook(String keyword) {  
 WebElement searchInput = wait.until(ExpectedConditions.visibilityOfElementLocated(searchBox));  
 searchInput.clear();  
 searchInput.sendKeys(keyword);  
 searchInput.sendKeys(Keys.*ENTER*);  
 }

Hình 4. 2: Mã nguồn lớp HomePage đại diện trang trang chủ trong mô hình Page Object Model (POM)

**Ví dụ 3: Lớp OrderPage** xử lý toàn bộ quy trình đặt hàng, bao gồm nhập thông tin người nhận, địa chỉ giao hàng, ghi chú, mã giảm giá, chọn phương thức thanh toán và gửi đơn hàng. Lớp này sử dụng kỹ thuật xử lý dropdown động (Select2), kiểm tra nút đặt hàng sẵn sàng, và xác thực kết quả đặt hàng thành công. Tất cả hành vi đều được tự động hóa để mô phỏng chính xác thao tác của người dùng cuối.

package pages;  
  
import org.openqa.selenium.\*;  
import org.openqa.selenium.support.ui.\*;  
  
import java.time.Duration;  
  
public class OrderPage {  
 private WebDriver driver;  
 private WebDriverWait wait;  
  
 public OrderPage(WebDriver driver) {  
 this.driver = driver;  
 this.wait = new WebDriverWait(driver, Duration.ofSeconds(15));  
 }  
  
 // Điền form thanh toán  
 public void fillCheckoutForm(String email, String fullName, String phone, String address,  
 String province, String district, String ward,  
 String note, String coupon, String paymentMethod) {  
  
 // Nhập các thông tin cơ bản  
 driver.findElement(By.name("email")).clear();  
 driver.findElement(By.name("email")).sendKeys(email);  
  
 driver.findElement(By.name("billingName")).clear();  
 driver.findElement(By.name("billingName")).sendKeys(fullName);

Hình 4. 3**:** Mã nguồn lớp OrderPage mô phỏng trang thanh toán website

Chus

**Ví dụ 4:** **Lớp SearchPage** đại diện cho chức năng tìm kiếm sản phẩm trên website. Lớp này quản lý ô nhập từ khóa và nút tìm kiếm, đồng thời cung cấp phương thức search() để thực hiện hành vi tìm kiếm sách giống với thao tác của người dùng thực tế.

package pages;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
  
public class SearchPage {  
 private WebDriver driver;  
  
 public SearchPage(WebDriver driver) {  
 this.driver = driver;  
 }  
  
 private By searchInput = By.cssSelector("input[name='query']");  
 private By searchButton = By.cssSelector("button[type='submit']");  
  
 public void enterKeyword(String keyword) {  
 driver.findElement(searchInput).clear();  
 driver.findElement(searchInput).sendKeys(keyword);  
 }  
  
 public void clickSearch() {  
 driver.findElement(searchButton).click();  
 }  
  
 public void search(String keyword) {  
 enterKeyword(keyword);  
 clickSearch();  
 }  
}

Hình 4. 4: Mã nguồn lớp SearchPage.java mô tả hành vi tìm kiếm sách trên website Chus

**Ví dụ 5:** **Lớp SignupPage** đại diện cho trang đăng ký tài khoản. Lớp này quản lý các thành phần giao diện như ô nhập họ tên, email, mật khẩu và nút “Đăng ký”. Ngoài ra, lớp cũng hỗ trợ lấy thông báo lỗi khi đăng ký không hợp lệ, giúp kiểm thử xác thực đầu vào hiệu quả hơn.

package pages;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class SignupPage {  
 WebDriver driver;  
  
 // Locator  
 By firstNameInput = By.id("first\_name");  
 By lastNameInput = By.id("last\_name");  
 By emailInput = By.id("email");  
 By passwordInput = By.id("creat\_password");  
 By registerButton = By.cssSelector("input[type='submit'][value='Đăng ký']");  
 By errorMessages = By.cssSelector("ul.disc > li");  
  
 public SignupPage(WebDriver driver) {  
 this.driver = driver;  
 }  
  
 // === Hành động nhập dữ liệu ===  
 public void enterFirstName(String firstName) {  
 driver.findElement(firstNameInput).clear();  
 driver.findElement(firstNameInput).sendKeys(firstName);  
 }  
  
 public void enterLastName(String lastName) {  
 driver.findElement(lastNameInput).clear();  
 driver.findElement(lastNameInput).sendKeys(lastName);

Hình 4. 5: Mã nguồn lớp SignupPage.java mô tả hành vi tìm kiếm sách trên website Chus

4.1.2 Cài đặt lớp BaseSetup

Lớp BaseSetup là lớp cơ sở dùng để khởi tạo và quản lý vòng đời của WebDriver, phục vụ cho toàn bộ ca kiểm thử. Thay vì khởi tạo trực tiếp, BaseSetup sử dụng lớp QlWebdriver để đảm bảo trình điều khiển được quản lý tập trung, chỉ khởi tạo một lần duy nhất (singleton pattern). Ngoài ra, lớp này cũng tích hợp việc ghi báo cáo kiểm thử sau mỗi lần thực thi.

Cấu trúc lớp BaseSetup:

package base;  
  
import org.openqa.selenium.WebDriver;  
import utils.ExcelReport;  
  
public class BaseSetup {  
 protected WebDriver driver;  
  
 public void initializeDriver() {  
 driver = QlWebdriver.getDriver(); // Lấy driver đã quản lý từ QlWebdriver  
 }  
  
 public void closeDriver() {  
 QlWebdriver.closeDriver(); // Đóng driver an toàn  
 }  
 public void tearDownAfterTest() {  
 ExcelReport.saveReport();  
 closeDriver();  
 }  
}

Hình 4. 6: Mã nguồn BaseSetup.java website Chus

Lớp QlWebdriver thực hiện khởi tạo WebDriver duy nhất, cấu hình cửa sổ trình duyệt và timeout mặc định:

package base;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
  
import java.time.Duration;  
  
public class QlWebdriver {  
 private static WebDriver *driver*;  
  
 public static WebDriver getDriver() {  
 if (*driver* == null) {  
 System.setProperty("webdriver.chrome.driver", "C:\\Users\\Admin\\Downloads\\chromedriver-win64\\chromedriver.exe");  
 *driver* = new ChromeDriver();  
 *driver*.manage().window().maximize();  
 *driver*.manage().timeouts().implicitlyWait(Duration.ofSeconds(10)); // Thêm timeout mặc định  
 }  
 return *driver*;  
 }  
  
 public static void closeDriver() {  
 if (*driver* != null) {  
 *driver*.quit();  
 *driver* = null; // Đặt lại để tạo mới nếu cần sau này  
 }  
 }  
  
}

Hình 4. 7: Mã nguồn QlWebdriver website Chus

* + 1. Kiểm thử hướng dữ liệu

Kiểm thử hướng dữ liệu là kỹ thuật cho phép chạy cùng một kịch bản kiểm thử nhiều lần với các bộ dữ liệu khác nhau được cung cấp từ bên ngoài (thường là file Excel, CSV, cơ sở dữ liệu...). Trong framework kiểm thử tự động của website Chus, phương pháp này được áp dụng cho các chức năng như: đăng nhập, đăng ký, tìm kiếm, đặt hàng,… để đảm bảo độ bao phủ rộng, linh hoạt và dễ bảo trì. Lớp đọc dữ liệu từ file Excel - ExcelReader.java

package utils;  
  
import java.io.FileInputStream;  
import java.util.ArrayList;  
import java.util.List;  
import org.apache.poi.ss.usermodel.\*;  
  
  
public class ExcelReader {  
  
 // Số cột cho từng loại dữ liệu  
 private static final int *LOGIN\_COLS* = 3;  
 private static final int *SEARCH\_COLS* = 2;  
 private static final int *SIGNUP\_COLS* = 5;  
 private static final int *ORDER\_COLS* = 13;  
  
 // Hàm dùng chung: đọc file Excel và trả về danh sách dòng với số cột cố định  
 private static List<String[]> readExcel(String filePath, int expectedColumns) {  
 List<String[]> data = new ArrayList<>();  
  
 try (FileInputStream fis = new FileInputStream(filePath);  
 Workbook workbook = WorkbookFactory.create(fis)) {

Hình 4. 8: Lớp đọc dữ liệu từ ExcelReader

* + 1. Tích hợp tiện ích hỗ trợ

Hỗ trợ thao tác mở – đọc – ghi dữ liệu Excel

package utils;  
  
import java.awt.Color;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import org.apache.poi.ss.usermodel.Cell;  
import org.apache.poi.ss.usermodel.CellStyle;  
import org.apache.poi.ss.usermodel.DateUtil;  
import org.apache.poi.ss.usermodel.Row;  
import org.apache.poi.ss.usermodel.Sheet;  
import org.apache.poi.ss.usermodel.Workbook;  
import org.apache.poi.ss.usermodel.WorkbookFactory;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
  
public class ExcelHelpers {  
 private FileInputStream fis;  
 private FileOutputStream fileOut;  
 private Workbook wb;  
 private Sheet sh;  
 private Cell cell;  
 private Row row;  
 private CellStyle cellstyle;  
 private Color mycolor;  
 private String excelFilePath;  
 private Map<String, Integer> columns = new HashMap();  
  
 public void setExcelFile(String ExcelPath, String SheetName) throws Exception {  
 try {  
 File f = new File(ExcelPath);  
 if (!f.exists()) {  
 f.createNewFile();  
 System.*out*.println("File doesn't exist, so created!");  
 }  
  
 this.fis = new FileInputStream(ExcelPath);  
 this.wb = WorkbookFactory.create(this.fis);  
 this.sh = this.wb.getSheet(SheetName);  
 if (this.sh == null) {  
 this.sh = this.wb.createSheet(SheetName);  
 }  
  
 this.excelFilePath = ExcelPath;  
 Row headerRow = this.sh.getRow(0);  
 if (headerRow != null) {  
 for(Cell c : headerRow) {  
 this.columns.put(c.getStringCellValue(), c.getColumnIndex());  
 }  
 }  
 } catch (Exception e) {  
 System.*out*.println(e.getMessage());  
 }  
  
 }

Hình 4. 9: Hỗ trợ thao tác mở, đọc, ghi dữ liệu Excel – ExcelHelpers

* + 1. Ghi kết quả báo cáo

Framework sử dụng hình thức ghi kết quả vào file excel: File Report.xlsx: Ghi lại kết quả thực thi (PASS/FAIL), gồm:

* Các trường nhập liệu
* Kết quả mong đợi
* Kết quả thực tế
* Trạng thái (PASS/FAIL)
* Thời gian thực hiện

Báo cáo này giúp người dùng phi kỹ thuật cũng có thể theo dõi quá trình kiểm thử của hệ thống Chus

Ghi báo cáo – ExcelReport

public static void startNewLoginTest(String sheetName) {  
 resetSTT();  
 *currentSheetName* = sheetName;  
 setupSheetWithHeaders(new String[]{  
 "STT", "Thời gian", "Email", "Password", "Kết quả mong đợi", "Kết quả thực tế", "Trạng thái"  
 });  
}  
  
public static void writeLoginReport(String email, String password, String expected, String actual, String status) {  
 Row row = *sheet*.createRow(*currentRow*++);  
 row.createCell(0).setCellValue(*stt*++);   
 row.createCell(1).setCellValue(getCurrentTime());  
 row.createCell(2).setCellValue(email);  
 row.createCell(3).setCellValue(password);  
 row.createCell(4).setCellValue(expected);  
 row.createCell(5).setCellValue(actual);  
 row.createCell(6).setCellValue(status);  
}

Hình 4. 10: Ghi báo cáo Login

public static void startNewSignupTest(String sheetName) {  
 resetSTT();  
 *currentSheetName* = sheetName;  
 setupSheetWithHeaders(new String[]{  
 "STT", "Thời gian", "Họ", "Tên", "Email", "Mật khẩu", "Kết quả mong đợi", "Kết quả thực tế", "Trạng thái"  
 });  
}  
  
public static void writeSignupReport(String lastName, String firstName, String email, String password,  
 String expected, String actual, String status) {  
 Row row = *sheet*.createRow(*currentRow*++);  
 row.createCell(0).setCellValue(*stt*++);  
 row.createCell(1).setCellValue(getCurrentTime());  
 row.createCell(2).setCellValue(lastName);  
 row.createCell(3).setCellValue(firstName);  
 row.createCell(4).setCellValue(email);  
 row.createCell(5).setCellValue(password);  
 row.createCell(6).setCellValue(expected);  
 row.createCell(7).setCellValue(actual);  
 row.createCell(8).setCellValue(status);  
}

Hình 4. 11: Ghi báo cáo Signup

public static void startNewSearchTest(String sheetName) {  
 resetSTT();  
 *currentSheetName* = sheetName;  
 setupSheetWithHeaders(new String[]{  
 "STT", "Thời gian", "Từ khóa", "Kết quả mong đợi", "Kết quả thực tế", "Trạng thái"  
 });  
}  
  
public static void writeSearchReport(String keyword, String expected, String actual, String status) {  
 Row row = *sheet*.createRow(*currentRow*++);  
 row.createCell(0).setCellValue(*stt*++);  
 row.createCell(1).setCellValue(getCurrentTime());  
 row.createCell(2).setCellValue(keyword);  
 row.createCell(3).setCellValue(expected);  
 row.createCell(4).setCellValue(actual);  
 row.createCell(5).setCellValue(status);  
}

Hình 4. 12: Ghi báo cáo Search

public static void startNewOrderTest(String sheetName) {  
 resetSTT();  
 *currentSheetName* = sheetName;  
 setupSheetWithHeaders(new String[]{  
 "STT", "Thời gian", "Tên sách", "Họ tên", "SĐT", "Địa chỉ", "Tỉnh", "Huyện", "Xã",  
 "Ghi chú", "Mã giảm giá", "PTTT", "Kết quả mong đợi", "Kết quả thực tế", "Trạng thái"  
 });  
}  
  
public static void writeOrderReport(String bookTitle, String fullName, String phone, String address, String province,  
 String district, String ward, String note, String coupon, String payment,  
 String expected, String actual, String status) {  
 Row row = *sheet*.createRow(*currentRow*++);  
 int col = 0;  
 row.createCell(col++).setCellValue(*stt*++);  
 row.createCell(col++).setCellValue(getCurrentTime());  
 row.createCell(col++).setCellValue(bookTitle);  
 row.createCell(col++).setCellValue(fullName);  
 row.createCell(col++).setCellValue(phone);  
 row.createCell(col++).setCellValue(address);  
 row.createCell(col++).setCellValue(province);  
 row.createCell(col++).setCellValue(district);  
 row.createCell(col++).setCellValue(ward);  
 row.createCell(col++).setCellValue(note);  
 row.createCell(col++).setCellValue(coupon);  
 row.createCell(col++).setCellValue(payment);  
 row.createCell(col++).setCellValue(expected);  
 row.createCell(col++).setCellValue(actual);  
 row.createCell(col).setCellValue(status);  
}

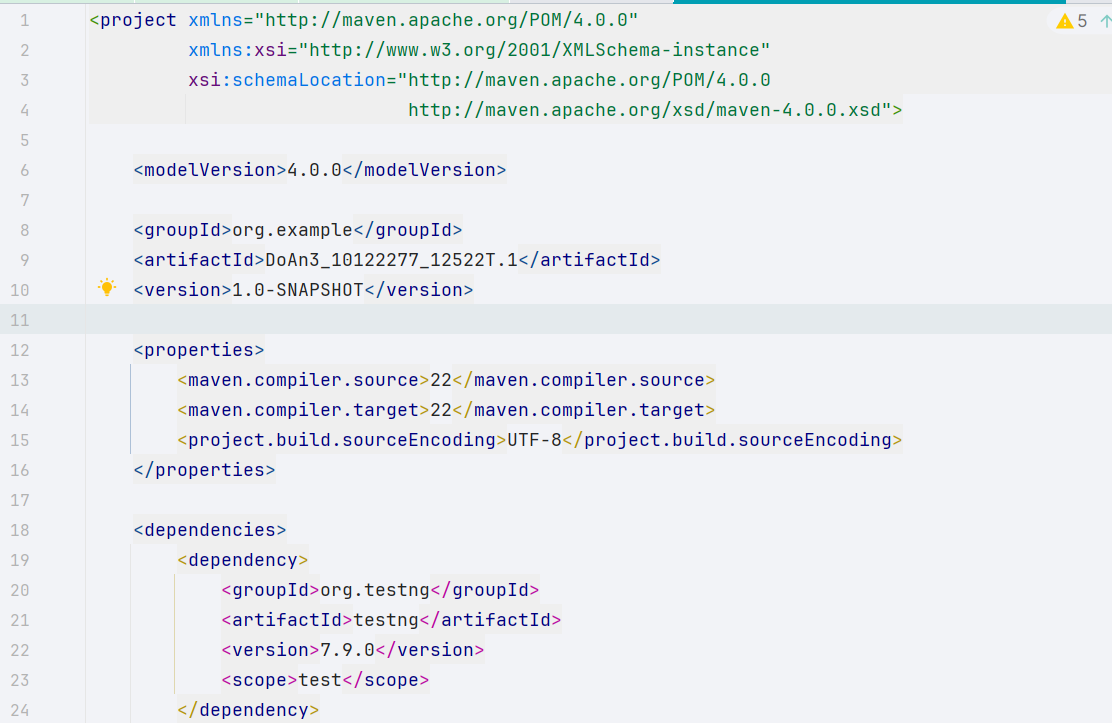
Hình 4. 13: Ghi báo cáo Order

* + 1. Quản lý kiểm thử và thư viện
* testng.xml: dùng để định nghĩa nhóm test, chạy tuần tự hoặc song song.



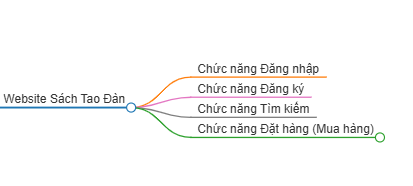
Hình 4. 14: Cài đặt testng.xml

* pom.xml: quản lý toàn bộ thư viện sử dụng (Selenium, TestNG, POI, ExtentReports...) thông qua Maven.

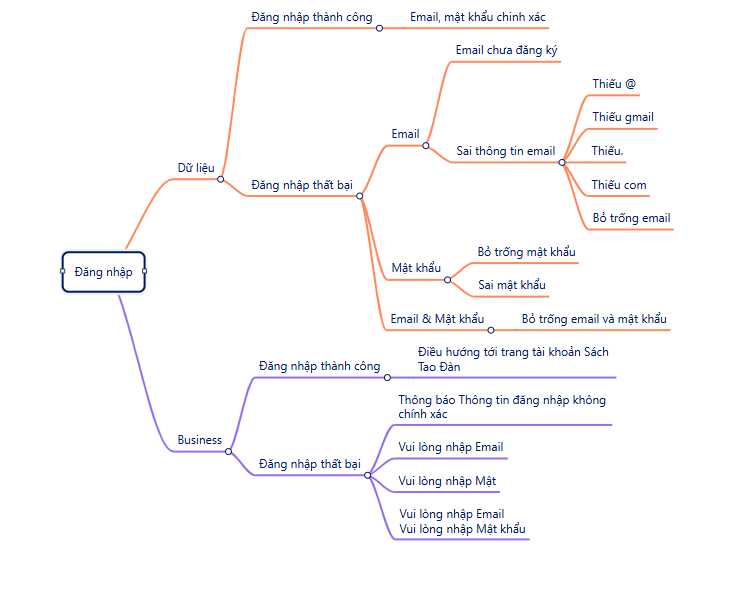


Hình 4. 15: Cài đặt pom.xml

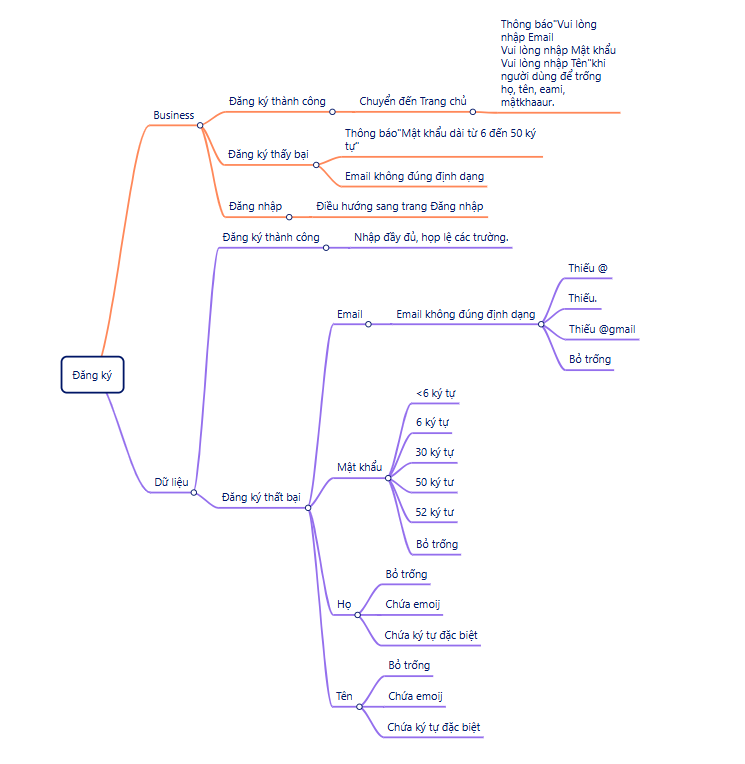
4.2 Thiết kế kịch bản kiểm thử chức năng



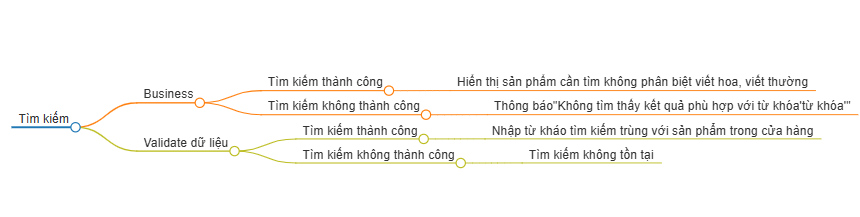
Hình 4. 16: Kịch bản kiểm thử chức năng website Chus

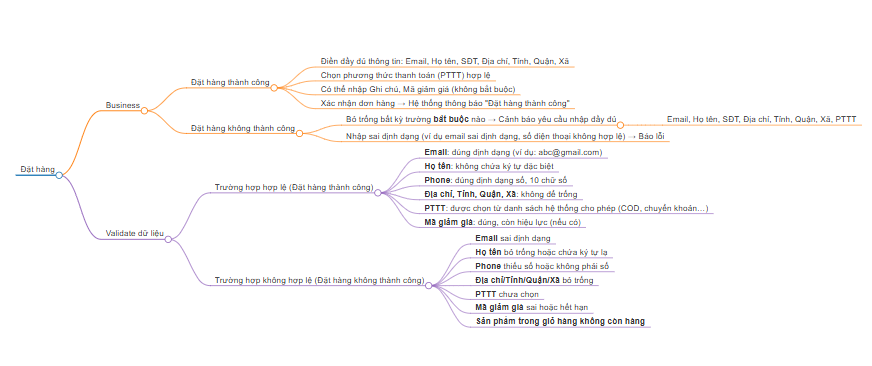


Hình 4. 17: Kịch bản kiểm thử chức năng Đăng nhập



Hình 4. 18: Kịch bản kiểm thử Chức năng Đăng ký

Hình 4. 19: Kịch bản kiểm thử Chức năng Tìm kiếm



Hình 4. 20: Kịch bản kiểm thử Chức năng Đặt hàng

4.3 Xây dựng và thực thi các trường hợp kiểm thử

4.3.1 Xây dựng và thực thi các trường hợp kiểm thử chức năng Đăng nhập

Tên class: LoginTest.java

Nguồn dữ liệu: Sheet “Login” trong file InputData.xlsx

Logic thực hiện:

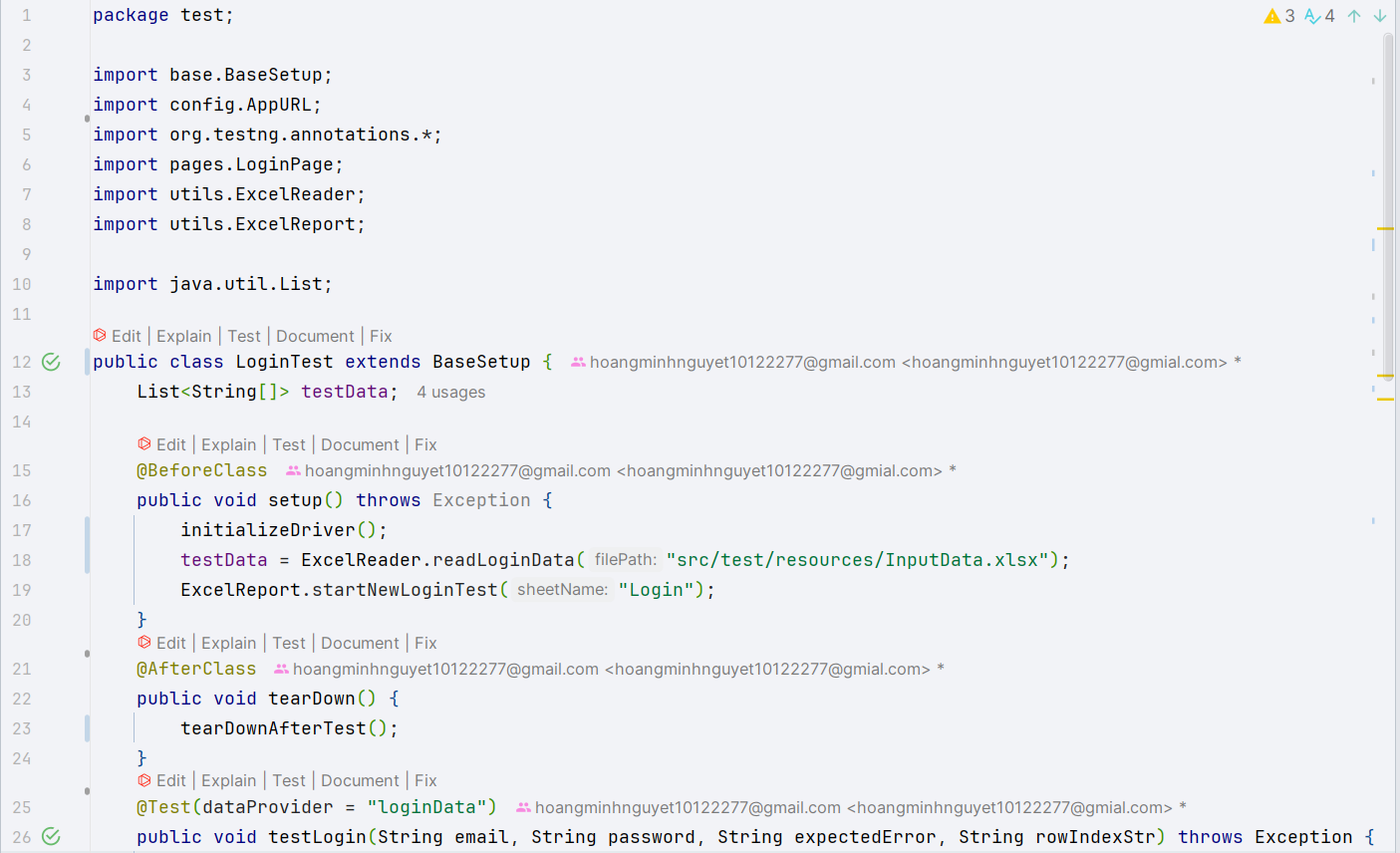
• Đọc từng dòng dữ liệu gồm: Email, Mật khẩu, Kết quả mong đợi

• Thực hiện thao tác đăng nhập trên giao diện

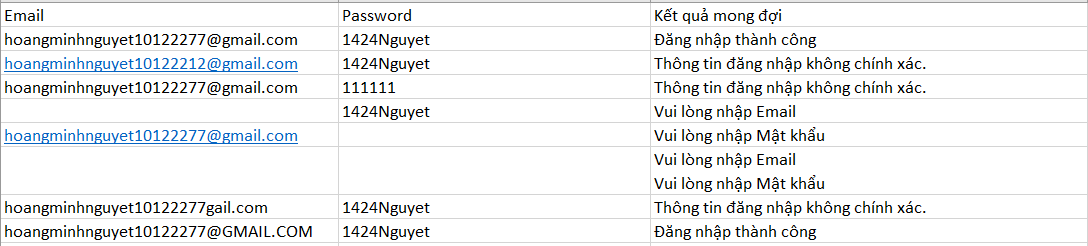
• So sánh kết quả thực tế:

* Nếu thiếu Email/Password → kiểm tra thông báo lỗi
* Nếu sai thông tin → kiểm tra thông báo lỗi
* Nếu đúng → kiểm tra tài khoản hiển thị

Ghi lại kết quả bằng ExcelReport.java



Hình 4. 21: Kịch bản kiểm thử chức năng Đăng nhập



Hình 4. 22: Dữ liệu chức năng Đăng nhập

4.3.2 Xây dựng và thực thi các trường hợp Kiểm thử chức năng

Tên lớp: SignupTest.java

Nguồn dữ liệu: Sheet “Signup” trong file InputData.xlsx

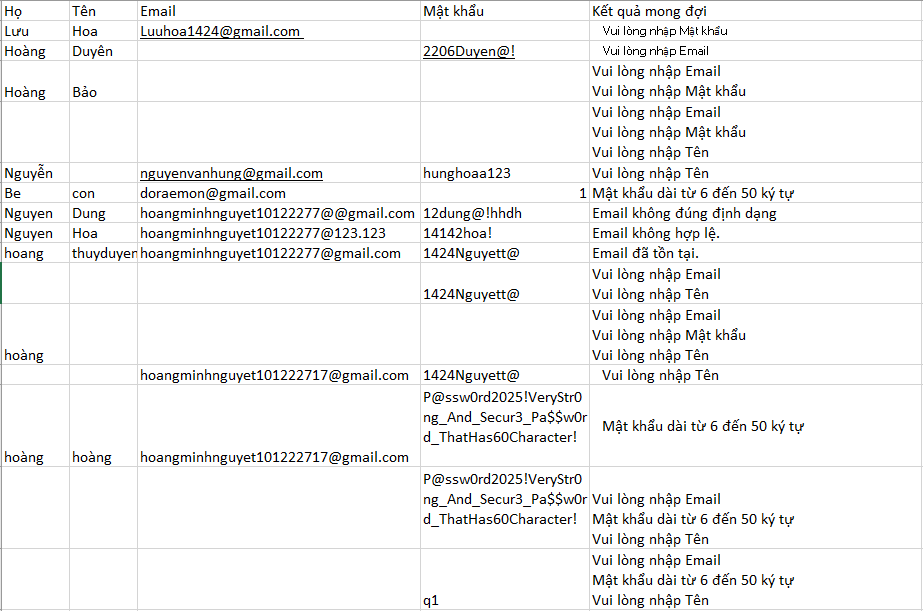
Quy trình kiểm thử:

* Nhập dữ liệu Email, Mật khẩu, Nhập lại mật
* Kiểm tra các tình huống:
* Thiếu trường dữ liệu → Hệ thống hiển thị thông báo
* Nhập sai quy tắc các trường → Hệ thống hiển thị thông báo
* Đăng ký thành công → kiểm tra tài khoản hiển thị

Ghi kết quả kiểm thử bằng ExcelReport.java



Hình 4. 23: Kịch bản kiểm thử chức năng Đăng ký



Hình 4. 24: Dữ liệu chức năng Đăng ký

4.3.3 Xây dựng và thực thi các trường hợp Kiểm thử hướng dữ liệu

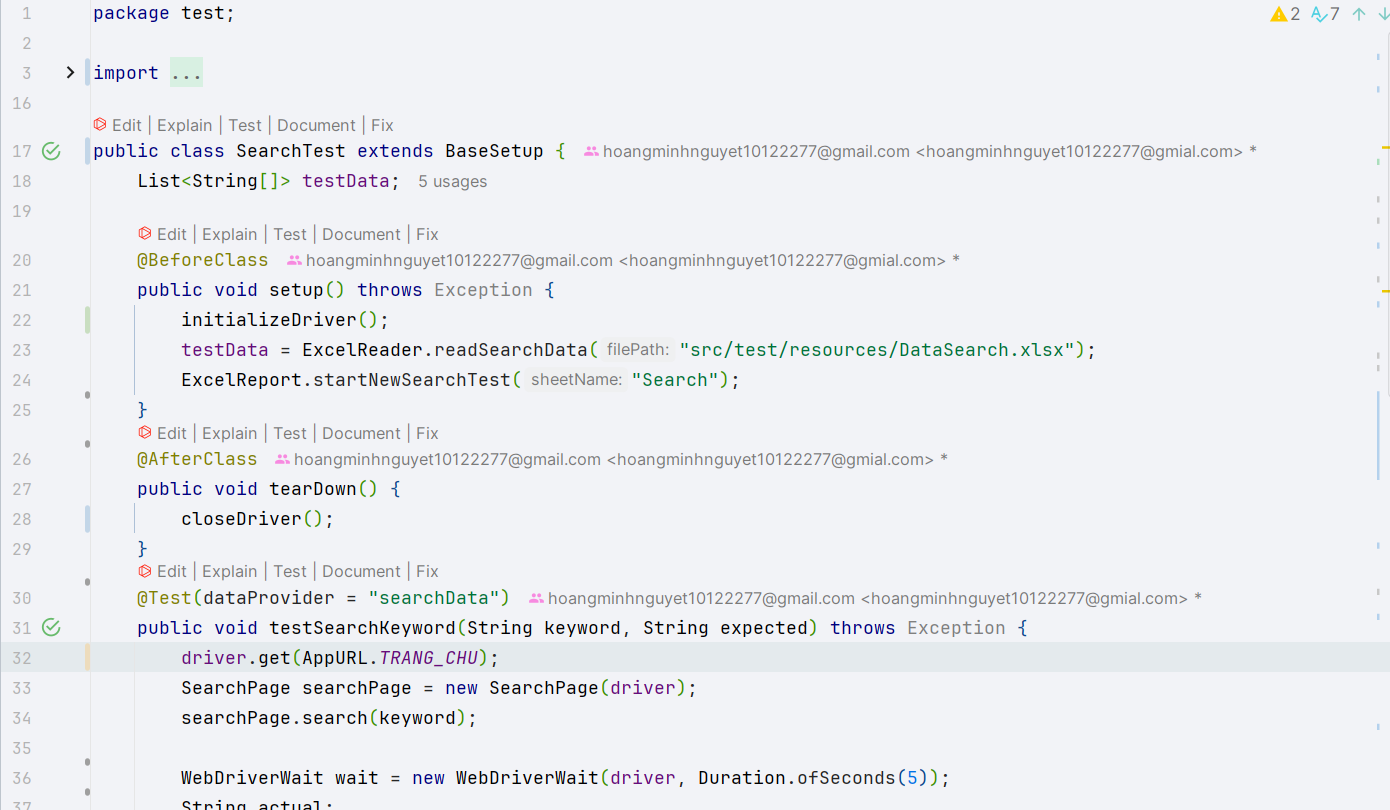
Tên lớp: SearchTest.java

Nguồn dữ liệu: Sheet “Search” trong file InputData.xlsx

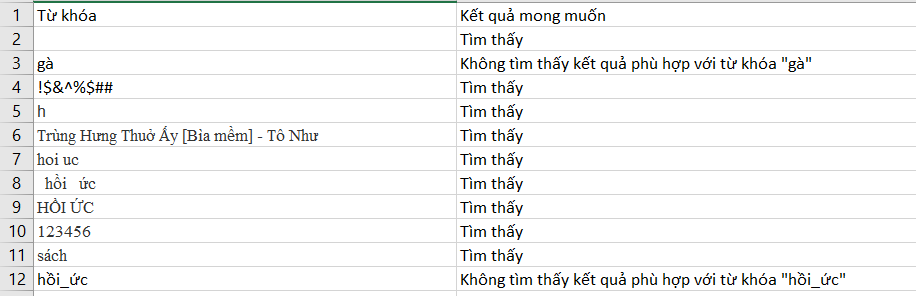
Thực hiện kiểm thử:

* + Nhập từ khóa vào ô tìm kiếm
  + Kiểm tra kết quả trả về:
* Có sản phẩm: từ khóa xuất hiện trong tiêu đề sản phẩm
* Không có: hiển thị thông báo “Không tìm thấy…”

Kết quả được ghi bởi ExcelReport.java



Hình 4. 25: Kịch bản kiểm thử chức năng Tìm kiếm sản phẩm



Hình 4. 26: Dữ liệu chức năng Tìm kiếm sản phẩm

4.3.4 Xây dựng và thực thi các trường hợp kiểm thử chức năng Đặt hàng

Tên lớp: OrderTest.java

Nguồn dữ liệu: Sheet “Order” trong file InputData.xlsx

Logic kiểm thử:

* Chọn một sản phẩm cụ thể trong danh sách
* Thay đổi số lượng sản phẩm.
* Click “Mua ngay” chọn “Tiến hành thanh toán”
* Nhập đầy đủ Email, Họ và tên, Số điện thoại, Địa chỉ, Chọn Tỉnh/Thành phố, Quận/Huyện, Phường/Xã, Ghi chú, Mã giảm giá, Phương thức thanh toán.
* Kiểm tra các trường hợp:
* Thành công: hiển thị thông báo đặt hàng thành công
* Thiếu hoặc sai định dạng trường dữ liệu → báo lỗi

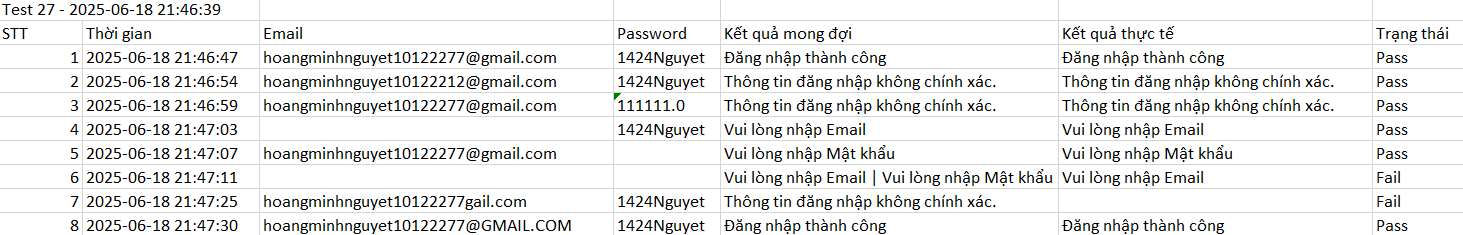
Kết quả được ghi bởi ExcelReport.java

4.3 Báo cáo và phân tích kết quả kiểm thử

4.4.1 Báo cáo kiểm thử chức năng Đăng nhập



Hình 4. 27: Tổng hợp kết quả kiểm thử chức năng Đăng nhập

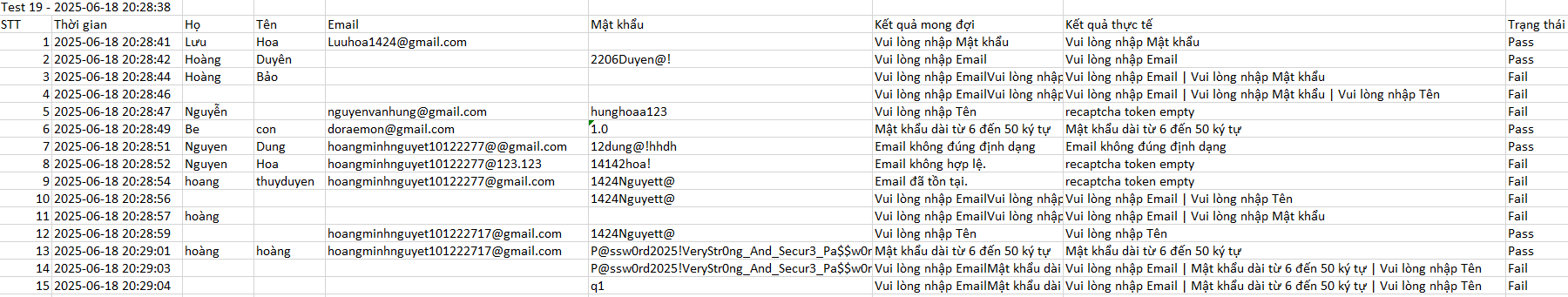


Hình 4. 28: Kết quả kiểm thử chức năng Đăng nhập

4.4.2 Báo cáo kiểm thử chức năng Đăng ký



Hình 4. 29: Tổng hợp kết quả kiểm thử chức năng Đăng ký

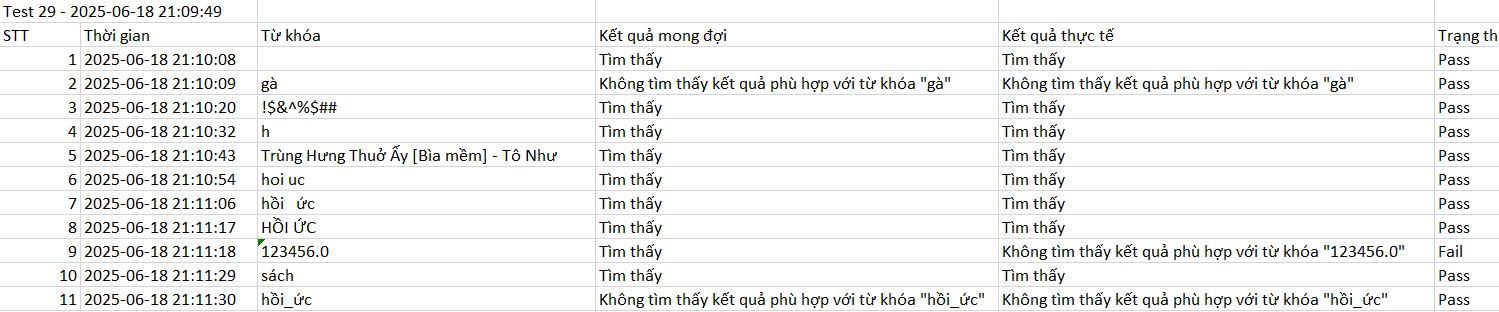


Hình 4. 30: Kết quả kiểm thử chức năng Đăng nhập

4.4.3 Báo cáo kiểm thử chức năng Tìm kiếm

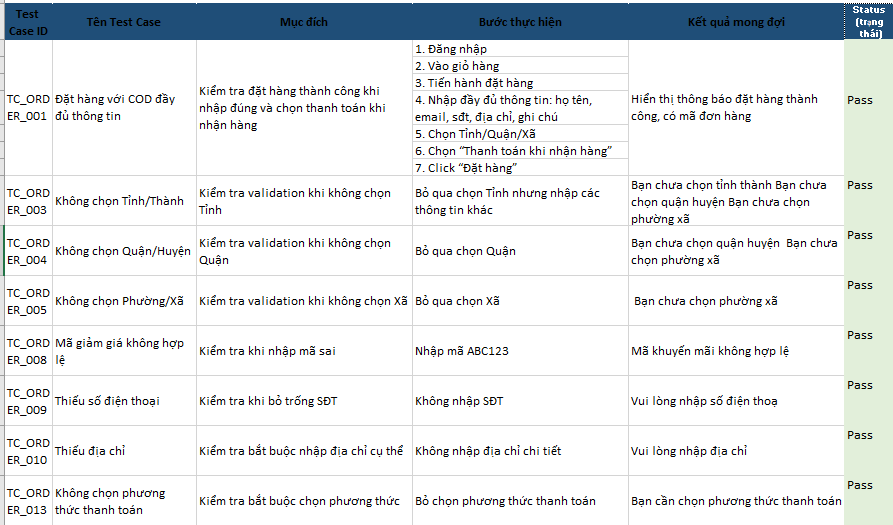


Hình 4. 31: Tổng hợp kết quả kiểm thử chức năng Đăng ký

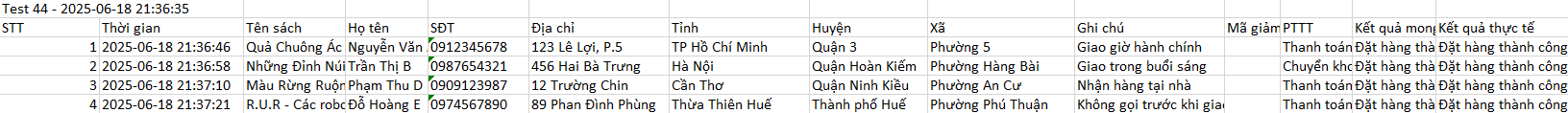


Hình 4. 32: Kết quả kiểm thử chức năng Tìm kiếm

4.4.4 Báo cáo kiểm thử chức năng Đặt hàng



Hình 4. 33: Tổng hợp kết quả kiểm thử chức năng Giỏ hàng



Hình 4. 34: Kết quả kiểm thử chức năng Giỏ hàng

4.4.5 Báo cáo kiểm thử chức năng website Chus

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Chức năng | Tổng số Test cas | Số test Pass | Số test Fail | Auto | Manual |
| 1 | Đăng nhập | 8 | 6 | 2 | 8 | 0 |
| 2 | Đăng ký | 15 | 12 | 3 | 15 | 0 |
| 3 | Tìm kiếm | 11 | 11 | 0 | 11 | 0 |
| 4 | Đặt hàng | 12 | 2 | 10 | 4 | 0 |
|  | | | | | | |
| TỔNG | | 46  (100%) | 33  (71,73%) | 15  (32,6%) | 46  (100%) | 0 |

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả đạt được của đề tài

Sau quá trình nghiên cứu, phân tích và triển khai, đề tài “Kiểm thử tự động và xây dựng framework dựa trên Selenium cho website Chus” đã đạt được một số kết quả nổi bật như sau:

* Thiết kế và xây dựng thành công một framework kiểm thử tự động có cấu trúc rõ ràng, dễ mở rộng và bảo trì, sử dụng các công nghệ phổ biến như Java, Selenium WebDriver, TestNG, Maven và Apache POI, kết hợp mô hình thiết kế Page Object Model (POM).
* Ứng dụng kiểm thử hướng dữ liệu (Data-Driven Testing) để tăng mức độ bao phủ của kiểm thử và nâng cao khả năng tái sử dụng kịch bản kiểm thử cho nhiều tập dữ liệu khác nhau.
* Tổ chức đầy đủ các thành phần cốt lõi của một framework kiểm thử chuyên nghiệp, bao gồm lớp thiết lập môi trường BaseSetup, trình điều phối WebDriver QlWebdriver, các lớp Page Object đại diện cho từng trang chức năng, và các tiện ích hỗ trợ như đọc file Excel, ghi log, chụp màn hình, xuất báo cáo.
* Xây dựng và thực thi kiểm thử tự động cho các chức năng chính của hệ thống như: đăng ký tài khoản, đăng nhập, tìm kiếm sản phẩm, thêm vào giỏ hàng, đặt hàng, khôi phục mật khẩu, với khả năng ghi nhận đầy đủ kết quả kiểm thử cho từng trường hợp.
* Tạo báo cáo kết quả kiểm thử chi tiết dưới dạng bảng, giúp cả người phát triển và quản lý dự án dễ dàng theo dõi tiến độ và đánh giá độ ổn định của hệ thống.

Những kết quả này cho thấy kiểm thử tự động mang lại hiệu quả đáng kể trong việc đảm bảo chất lượng phần mềm, đặc biệt với các hệ thống thương mại điện tử như Chus, nơi có số lượng chức năng và dữ liệu lớn, đòi hỏi quy trình kiểm thử thường xuyên, chính xác và nhanh chóng.

5.2 Hạn chế của đề tài

Bên cạnh những kết quả đạt được, đề tài vẫn tồn tại một số điểm hạn chế cần được khắc phục trong các giai đoạn phát triển tiếp theo:

* Chưa tích hợp với các hệ thống CI/CD (Continuous Integration/Continuous Deployment), nên việc thực thi kiểm thử vẫn đang phụ thuộc vào thao tác thủ công từ môi trường IDE.
* Chưa thực hiện kiểm thử đa trình duyệt (cross-browser testing) và trên thiết bị di động, điều này làm hạn chế phạm vi đánh giá tính tương thích của website.
* Hệ thống báo cáo còn đơn giản, mới chỉ hỗ trợ xuất kết quả kiểm thử ra file Excel, chưa áp dụng các công cụ báo cáo nâng cao như ExtentReports hoặc Allure để cải thiện tính trực quan và khả năng phân tích.
* Chưa tích hợp kiểm thử phi chức năng, như kiểm thử hiệu năng, bảo mật, khả năng chịu tải – vốn rất quan trọng với các nền tảng thương mại điện tử.

5.3 Hướng phát triển của đề tài

Để nâng cao năng lực và mức độ hoàn thiện của hệ thống kiểm thử, trong tương lai, đề tài có thể được mở rộng theo các hướng sau:

* Tích hợp quy trình kiểm thử vào hệ thống CI/CD, sử dụng các công cụ như Jenkins, GitHub Actions hoặc GitLab CI để tự động hóa kiểm thử sau mỗi lần cập nhật mã nguồn, từ đó rút ngắn chu kỳ phát hành và nâng cao chất lượng liên tục.
* Thực hiện kiểm thử đa nền tảng, bao gồm kiểm thử đa trình duyệt (Chrome, Firefox, Edge, Safari) và kiểm thử trên thiết bị di động thông qua Appium hoặc Selenium Grid, nhằm đảm bảo website hoạt động ổn định trong nhiều môi trường khác nhau.
* Nâng cấp hệ thống báo cáo bằng cách tích hợp các thư viện như ExtentReports hoặc Allure Report, cho phép trình bày kết quả kiểm thử dưới dạng biểu đồ, bảng trạng thái, và ảnh chụp màn hình kèm mô tả lỗi chi tiết.
* Bổ sung kiểm thử bảo mật tự động (Security Testing) bằng cách tích hợp với các công cụ chuyên biệt như OWASP ZAP hoặc Burp Suite, giúp phát hiện lỗ hổng sớm trong quá trình phát triển.
* Phát triển kiểm thử hiệu năng (Performance Testing) thông qua các công cụ như Apache JMeter hoặc Gatling để đo lường khả năng đáp ứng, độ trễ và hiệu suất của hệ thống khi có nhiều người dùng đồng thời.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Apache POI – *Java API for Microsoft Documents* – <https://poi.apache.org/> |
| [2] | Nguyễn Hữu Thắng (2021). *Kiểm thử phần mềm – Nguyên lý và thực hành*. NXB Khoa học và Kỹ thuật. |
| [3] | Trường Đại học Sư Phạm Kỹ thuật Hưng Yên, “Đề cương Kiểm thử ứng dụng Web & Di động,” Hưng Yên, 2025. . |

|  |  |
| --- | --- |
| [4] | A. Tester, “Anh Tester,” [Trực tuyến]. Available: https://anhtester.com/blog/cac-test-automation-framework-pho-bien-hien-nayb454.html. |
| [5] | Website kiểm thử: *Chus – Hệ thống thương mại điện tử phân phối sách*. Truy cập tại: <https://sachtaodan.vn/> |