**BỘ CÔNG THƯƠNG**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**---------------------------**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN**

**KIỂM THỬ TỰ ĐỘNG VỚI SELENIUM TÍCH HỢP CI/CD BẰNG GITLAB CI**

**GVHD : PGS.TS. Trần Đăng Hưng**

**Sinh viên : Trần Đặng Phương Thảo**

**Mã số sinh viên : 2021601876**

#### 

**Hà Nội - 2025**

# 

# LỜI **CẢM** ƠN

Em xin chân thành cảm ơn quý thầy cô Trường Đại học Công Nghiệp Hà Nội đã tận tình giảng dạy và truyền đạt cho em những kiến thức nền tảng cũng như kinh nghiệm thực tiễn trong suốt quá trình học tập. Đây là hành trang quan trọng giúp em hoàn thành tốt chương trình học và thực hiện đồ án tốt nghiệp.

Em đặc biệt gửi lời cảm ơn đến thầy **PGS.TS. Trần Đăng Hưng** - người đã hướng dẫn, hỗ trợ và góp ý cho em trong suốt quá trình làm đồ án. Những đóng góp và định hướng của thầy đã giúp em nâng cao tư duy, kỹ năng làm việc và hoàn thành đề tài một cách hiệu quả.

Em cũng xin ghi nhận và cảm ơn sự hỗ trợ từ Nhà trường cùng các phòng ban chức năng đã tạo điều kiện thuận lợi về môi trường học tập, cơ sở vật chất và các hoạt động liên quan, giúp em có cơ hội tiếp cận và ứng dụng kiến thức vào thực tế.

Quãng thời gian bốn năm đại học đã mang lại cho em nhiều trải nghiệm giá trị, từ kiến thức chuyên môn đến kỹ năng làm việc và những mối quan hệ tích cực. Đây sẽ là nền tảng để em tiếp tục phát triển bản thân trong chặng đường sắp tới.

Cuối cùng, em xin kính chúc quý thầy cô sức khỏe, luôn giữ vững tinh thần nhiệt huyết với nghề. Chúc các bạn cùng khóa hoàn thành tốt đồ án tốt nghiệp và đạt được những mục tiêu mà mình đặt ra trong tương lai.

Em xin chân thành cảm ơn!

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc202130592)

[MỤC LỤC ii](#_Toc202130593)

[DANH MỤC HÌNH ẢNH iv](#_Toc202130594)

[DANH MỤC BẢNG BIỂU vi](#_Toc202130595)

[MỞ ĐẦU 1](#_Toc202130596)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 3](#_Toc202130597)

[1.1. Kiểm thử phần mềm 3](#_Toc202130598)

[1.1.1. Khái niệm 3](#_Toc202130599)

[1.1.2. Nguyên tắc kiểm thử phần mềm 3](#_Toc202130600)

[1.1.3. Quy trình kiểm thử phần mềm 5](#_Toc202130601)

[1.2. Kiểm thử tự động 9](#_Toc202130602)

[1.2.1. Khái niệm 9](#_Toc202130603)

[1.2.2. Một số công cụ kiểm thử tự động phổ biến 10](#_Toc202130604)

[1.2.3. Quy trình kiểm thử tự động 11](#_Toc202130605)

[1.2.4. Các trường hợp nên áp dụng kiểm thử tự động 11](#_Toc202130606)

[1.3. Kiểm thử tự động tích hợp CI/CD 12](#_Toc202130607)

[1.3.1. Giới thiệu về hệ thống CI/CD 12](#_Toc202130608)

[1.3.2. Luồng hoạt động của CI/CD 13](#_Toc202130609)

[1.3.3. Lợi ích của CI/CD 14](#_Toc202130610)

[1.4. Tổng kết chương 1 15](#_Toc202130611)

[CHƯƠNG 2. TÌM HIỂU VỀ SELENIUM WEBDRIVER VÀ WEBSITE KIỂM THỬ 16](#_Toc202130612)

[2.1. Công cụ kiểm thử tự động Selenium 16](#_Toc202130613)

[2.1.1. Giới thiệu chung về Selenium 16](#_Toc202130614)

[2.1.2. Các thành phần chính của Selenium 17](#_Toc202130615)

[2.1.3. Selenium WebDriver 18](#_Toc202130616)

[2.2. Tích hợp CI/CD bằng GitLab CI 19](#_Toc202130617)

[2.2.1. CI/CD 19](#_Toc202130618)

[2.2.2. GitLab 20](#_Toc202130619)

[2.2.3. Git Flow 21](#_Toc202130620)

[2.2.4. Pipeline 21](#_Toc202130621)

[2.2.5. Triển khai CI/CD với GitLab 22](#_Toc202130622)

[2.3. Tổng quan về website kiểm thử 24](#_Toc202130623)

[2.3.1. Giới thiệu 24](#_Toc202130624)

[2.3.2. Các chức năng của website 25](#_Toc202130625)

[2.4. Tổng kết chương 2 28](#_Toc202130626)

[CHƯƠNG 3. TRIỂN KHAI KIỂM THỬ VÀ ĐÁNH GIÁ 29](#_Toc202130627)

[3.1. Thiết lập môi trường 29](#_Toc202130628)

[3.1.1. Cài đặt cấu trúc Framework 29](#_Toc202130629)

[3.1.2. Thiết lập môi trường 31](#_Toc202130630)

[3.2. Kế hoạch kiểm thử 32](#_Toc202130631)

[3.2.1. Xác định các yêu cầu kiểm thử 32](#_Toc202130632)

[3.2.2. Xây dựng dữ liệu kiểm thử 33](#_Toc202130633)

[3.3. Xây dựng các test script 35](#_Toc202130634)

[3.4. Tích hợp CI/CD với Gitlab CI 48](#_Toc202130635)

[3.5. Báo cáo kiểm thử 50](#_Toc202130636)

[3.6. Tổng kết chương 3 54](#_Toc202130637)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI 55](#_Toc202130638)

[TÀI LIỆU THAM KHẢO 57](#_Toc202130639)

[PHỤ LỤC 58](#_Toc202130640)

# DANH MỤC HÌNH ẢNH

[Hình 1. 1. Kiểm thử phần mềm 3](#_Toc202129983)

[Hình 1. 2. Quy trình kiểm thử phần mềm 5](#_Toc202129984)

[Hình 1. 3. Kiểm thử tự động 9](#_Toc202129985)

[Hình 1. 4. Luồng hoạt động của CI/CD 13](#_Toc202129986)

[Hình 2. 1. Cấu trúc Selenium 16](#_Toc202129989)

[Hình 2. 2. Quy trình CI/CD 19](#_Toc202129990)

[Hình 2. 3. GitLab 20](#_Toc202129991)

[Hình 2. 4. Git Flow 21](#_Toc202129992)

[Hình 2. 5. Kết quả chạy job GitLab CI/CD 21](#_Toc202129993)

[Hình 2. 6. Đồ thị pipeline thông thường 22](#_Toc202129994)

[Hình 2. 7. Nội dung cơ bản của file “.gitlab-ci.yml” 23](#_Toc202129995)

[Hình 2. 8. Giao diện hệ thống 25](#_Toc202129996)

[Hình 3. 1. Cấu trúc framework dự án 30](#_Toc202129997)

[Hình 3. 2. Dữ liệu kiểm thử đăng ký 33](#_Toc202129998)

[Hình 3. 3. Dữ liệu kiểm thử đăng nhập 33](#_Toc202129999)

[Hình 3. 4. Dữ liệu kiểm thử đặt hàng khi đã login 34](#_Toc202130000)

[Hình 3. 5. Dữ liệu kiểm thử đặt hàng khi chưa login 34](#_Toc202130001)

[Hình 3. 6. Dữ liệu kiểm thử thêm sản phẩm 35](#_Toc202130002)

[Hình 3. 7. Mã nguồn kiểm thử chức năng đăng ký 36](#_Toc202130003)

[Hình 3. 8. Mã nguồn kiểm thử chức năng đăng nhập 37](#_Toc202130004)

[Hình 3. 9. Mã nguồn kiểm thử chức năng đăng xuất 38](#_Toc202130005)

[Hình 3. 10. Mã nguồn kiểm thử chức năng thêm sản phẩm vào giỏ hàng 39](#_Toc202130006)

[Hình 3. 11. Mã nguồn kiểm thử chức năng xóa sản phẩm khỏi giỏ hàng 40](#_Toc202130007)

[Hình 3. 12. Mã nguồn kiểm thử chức năng cập nhật sản phẩm trong giỏ hàng 41](#_Toc202130008)

[Hình 3. 13. Mã nguồn kiểm thử chức năng tìm kiếm sản phẩm 42](#_Toc202130009)

[Hình 3. 14. Mã nguồn kiểm thử chức năng xem đơn hàng đã đặt 43](#_Toc202130010)

[Hình 3. 15. Mã nguồn kiểm thử chức năng đặt hàng có đăng nhập 44](#_Toc202130011)

[Hình 3. 16. Mã nguồn kiểm thử chức năng đặt hàng không cần đăng nhập 45](#_Toc202130012)

[Hình 3. 17. Mã nguồn kiểm thử chức năng thêm sản phẩm(admin) 46](#_Toc202130013)

[Hình 3. 18. Mã nguồn kiểm thử chức năng xóa sản phẩm(admin) 47](#_Toc202130014)

[Hình 3. 19. Tổ chức mã nguồn kiểm thử tự động trên GitLab 48](#_Toc202130015)

[Hình 3. 20. Kết quả artifact được tạo ra từ job kiểm thử trên GitLab 49](#_Toc202130016)

[Hình 3. 21. Kết quả thực thi của chức năng Quản lý tài khoản 49](#_Toc202130017)

[Hình 3. 22. Report của chức năng Quản lý tài khoản 49](#_Toc202130018)

[Hình 3. 23. Kết quả thực thi của chức năng Quản lý giỏ hàng 50](#_Toc202130019)

[Hình 3. 24. Report của chức năng Quản lý giỏ hàng 50](#_Toc202130020)

[Hình 3. 25. Kết quả thực thi cho chức năng Xem đơn hàng đã đặt 50](#_Toc202130021)

[Hình 3. 26. Report của chức năng Xem đơn hàng đã đặt 51](#_Toc202130022)

[Hình 3. 27. Kết quả thực thi cho chức năng tìm kiếm sản phẩm 51](#_Toc202130023)

[Hình 3. 28. Report của chức năng tìm kiếm sản phẩm 52](#_Toc202130024)

[Hình 3. 29. Kết quả thực thi cho chức năng đặt hàng 52](#_Toc202130025)

[Hình 3. 30. Report của chức năng đặt hàng 52](#_Toc202130026)

[Hình 3. 31. Kết quả thực thi cho chức năng quản lý sản phẩm 53](#_Toc202130027)

[Hình 3. 32. Report của chức năng quản lý sản phẩm 53](#_Toc202130028)

# DANH MỤC BẢNG BIỂU

[Bảng 2. 1. Bảng mô tả các chức năng của hệ thống 25](#_Toc202130214)

[Bảng 2. 2. Bảng mô tả các yêu cầu phi chức năng của hệ thống 27](#_Toc202130215)

[Bảng 3. 1. Bảng Môi trường phát triển và quản lý dự án 31](#_Toc202130227)

[Bảng 3. 2. Bảng Thư viện và công cụ hỗ trợ 31](#_Toc202130228)

[Bảng 4. 1. Bảng testcase chức năng quản lý tài khoản 57](#_Toc202130218)

[Bảng 4. 2. Bảng testcase chức năng quản lý giỏ hàng 67](#_Toc202130219)

[Bảng 4. 3. Bảng testcase chức năng đặt hàng 69](#_Toc202130220)

[Bảng 4. 4. Bảng testcase chức năng quản lý sản phẩm 83](#_Toc202130221)

[Bảng 4. 5. Bảng testcase chức năng xem đơn hàng đã đặt 86](#_Toc202130222)

# 

# MỞ ĐẦU

Ngày nay, công nghệ thông tin nói chung và công nghệ phần mềm nói riêng đang chiếm một vị trí quan trọng trong tiến trình công nghiệp hoá, hiện đại hoá đất nước. Song song với việc phát triển công nghệ phần mềm luôn tiềm ẩn những thách thức cho dành các doanh nghiệp, nhà phát triển phần mềm trong việc kiểm soát lỗi, chất lượng đầu ra của sản phẩm. Tuy nhiên ở Việt Nam, số lượng các kiểm thử viên vẫn chưa đáp ứng được với nhu cầu của thị trường. Bên cạnh đó, xu hướng áp dụng tự động hoá đang được triển khai rộng rãi ở nhiều lĩnh vực, trong đó có kiểm thử phần mềm. Đặc biệt, khi kiểm thử phần mềm là công đoạn chiếm phần lớn thời gian trong quá trình phát triển dự án phần mềm thì sự ra đời của các công cụ kiểm thử tự động càng có ý nghĩa hơn bao giờ hết, giúp tiết kiệm thời gian, công sức và tiền bạc. Selenium là một công cụ hỗ trợ kiểm thử tự động dành cho các ứng dụng Web, hoạt động trên hầu hết các trình duyệt phổ biến hiện nay như Firefox, Chrome, Internet, Explorer, Safari, v.v. cũng như hỗ trợ số lượng lớn các ngôn ngữ lập trình Web phổ biến. Công cụ Selenium hiện được đánh giá là một trong những công cụ tốt nhất cho kiểm thử tự động các ứng dụng Web.

Với mong muốn được tìm hiểu sâu về lĩnh vực kiểm thử phần mềm cũng như trở thành một kỹ sư kiểm thử phần mềm sau khi tốt nghiệp đại học, em đã chọn đề tài ***“Kiểm thử tự động với Selenium tích hợp CI/CD bằng GitLab CI.”***. Trong quá trình làm đồ án, do còn hạn chế về thời gian và kinh nghiệm thực tế, em mong nhận được những góp ý chân thành từ thầy cô và các bạn.

Đồ án được tổ chức làm 3 phần như sau:

*Chương 1: Cơ sở lý thuyết*

Trình bày các kiến thức nền tảng về kiểm thử phần mềm, đồng thời giới thiệu kiểm thử tự động và cách kiểm thử tự động được tích hợp vào quy trình phát triển phần mềm thông qua hệ thống CI/CD.

*Chương 2: Tìm hiểu về Selenium Webdriver và website kiểm thử* Trình bày về công cụ kiểm thử tự động Selenium và phương pháp tích hợp kiểm thử vào quy trình CI/CD sử dụng GitLab CI. Đồng thời mô tả tổng quan hệ thống website bán hàng được lựa chọn làm đối tượng kiểm thử.

*Chương 3: Triển khai kiểm thử và đánh giá*

Trình bày cách xây dựng kế hoạch kiểm thử, viết test case. Cài đặt, thử nghiệm, đánh giá kết quả khi áp dụng kiểm thử website cụ thể.

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

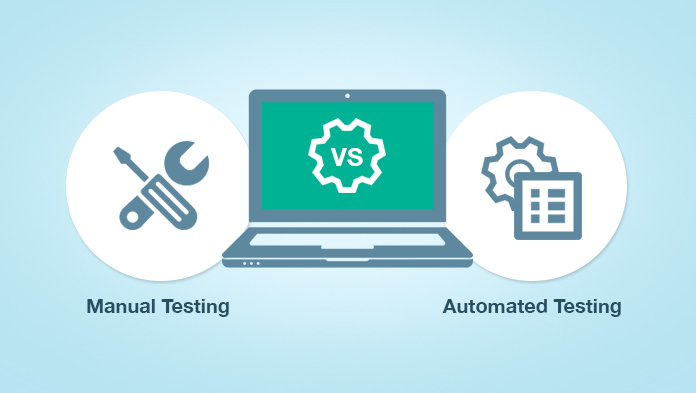
## 1.1. Kiểm thử phần mềm

### 1.1.1. Khái niệm

Kiểm thử phần mềm (Software Testing) là một quá trình bao gồm nhiều hoạt động nhằm đánh giá chất lượng các sản phẩm phần mềm và giảm thiểu rủi ro do lỗi gây ra trong quá trình vận hành khi đưa vào sử dụng thực tế.

Có 2 loại kiểm thử phần mềm: + Kiểm thử thủ công (Manual Testing)

+ Kiểm thử tự động (Automated Testing)



Hình 1. 1. Kiểm thử phần mềm

### 1.1.2. Nguyên tắc kiểm thử phần mềm

* Kiểm thử chứng minh sự hiện diện của lỗi

Kiểm thử nhằm mục đích tìm ra các lỗi, các vấn đề trong phần mềm. Khi một lỗi được tìm thấy, điều đó chứng minh rằng phần mềm có lỗi.

* Kiểm thử toàn bộ là không thể

Hầu hết các sản phẩm ngày nay rất đa dạng và phức tạp do được phát triển trên nhiều nền tảng, công nghệ phong phú cũng như khả năng lưu trữ kết nối dữ liệu lớn, khiến việc kiểm thử trở nên khó khăn và việc kiểm thử toàn bộ là gần như không thể. Kiểm thử với tất cả các kết hợp đầu vào và đầu ra, với tất cả các kịch bản là không thể trừ phi nó chỉ bao gồm ít trường hợp thì có thể kiểm thử toàn bộ. Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên sự mức độ ưu tiên chúng ta có thể tập trung việc kiểm thử vào một số điểm cần thiết, có nguy cơ lỗi cao hơn.

* Kiểm thử càng sớm càng tốt

Nguyên tắc này yêu cầu bắt đầu thử nghiệm phần mềm trong giai đoạn đầu của vòng đời phát triển phần mềm. Các hoạt động kiểm thử phần mềm từ giai đoạn đầu sẽ giúp phát hiện bug sớm hơn. Nó cho phép chuyển giao phần mềm theo yêu cầu đúng thời gian với chất lượng dự kiến. Việc phát hiện lỗi càng muộn bao nhiêu thì chi phí để sửa lỗi càng cao bấy nhiêu. Tương tự, việc thay đổi yêu cầu không đúng ngay từ đầu thường tốn ít chi phí thay đổi tính năng trong hệ thống.

* Lỗi thường được phân bố tập trung

Thông thường, phần lớn lỗi tập trung vào những module, thành phần chức năng chính của hệ thống. Điều này cũng thuận theo nguyên lý Pareto: 80% số lượng lỗi được tìm thấy trong 20% tính năng của hệ thống. Nếu bạn thành công xác định được điều này, bạn sẽ tập trung vào tìm kiếm lỗi quanh khu vực được xác định. Nó được coi là một trong những cách hiệu quả nhất để thực hiện kiểm tra hiệu quả.

* Nghịch lý thuốc trừ sâu

Trong kiểm thử phần mềm, nếu bạn cứ thực thi lặp đi lặp lại một bộ test case thì có khả năng rất thấp bạn sẽ tìm được lỗi từ những trường hợp kiểm thử này. Nguyên nhân là do khi hệ thống ngày càng hoàn thiện, những lỗi được tìm thấy lúc trước đã được sửa trong khi những trường hợp kiểm thử đã cũ. Do đó, khi một lỗi được sửa hay một tính năng mới được thêm vào, chúng ta nên tiến hành làm regression (kiểm thử hồi quy) nhằm mục đích đảm bảo những thay đổi này không ảnh hưởng đến những vùng khác của sản phẩm.

* Kiểm thử phụ thuộc vào ngữ cảnh

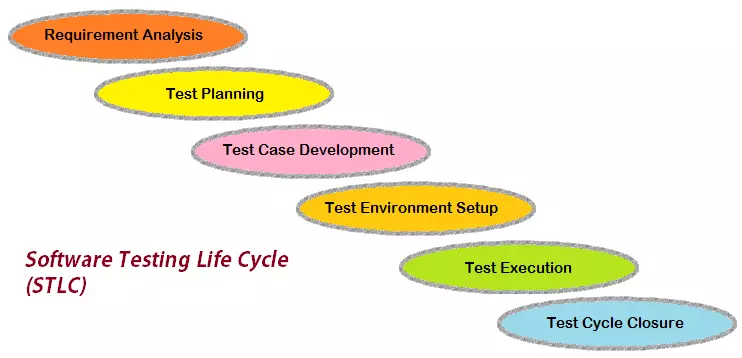
Theo nguyên tắc này thì nếu bạn đang kiểm thử ứng dụng web và ứng dụng di động bằng cách sử dụng chiến lược kiểm thử giống nhau, thì điều đó là sai lầm. Chiến lược để kiểm thử nên khác nhau và phụ thuộc vào chính ứng dụng đó.

* Quan niệm sai lầm về việc “hết lỗi”

Việc không tìm thấy lỗi trên sản phẩm không đồng nghĩa với việc sản phẩm đã sẵn sàng để tung ra thị trường. Việc không tìm thấy lỗi cũng có thể là do bộ trường hợp kiểm thử được tạo ra chỉ nhằm kiểm tra những tính năng được làm đúng theo yêu cầu thay vì nhằm tìm kiếm lỗi mới.

### 1.1.3. Quy trình kiểm thử phần mềm

Quy trình kiểm thử phần mềm là một phần không thể thiếu và quan trọng trong quy trình phát triển phần mềm. Đây là giai đoạn quyết định để đảm bảo rằng sản phẩm cuối cùng đáp ứng các yêu cầu chất lượng và hiệu suất được đặt ra.



Hình 1. 2. Quy trình kiểm thử phần mềm

**Giai đoạn 1: Requirement analysis – Phân tích yêu cầu**

Phân tích yêu cầu là giai đoạn đầu tiên trong vòng đời kiểm thử phần mềm. Ở giai đoạn này, QA team cần theo sát, nghiên cứu và phân tích từng yêu cầu trong các tài liệu đặc tả thiết kế dự án. Các yêu cầu được miêu tả trong tài liệu có thể là yêu cầu về chức năng hoặc phi chức năng như kiểm thử hiệu xuất, bảo mật...Trong quá trình nghiên cứu và phân tích, nếu có câu hỏi phát sinh, QA team sẽ đưa ra câu hỏi với các bên liên quan như Chuyên viên phân tích kinh doanh, Kiến trúc hệ thống, Khách hàng, Trưởng phòng kỹ thuật / Trưởng nhóm… để hiểu rõ hơn về yêu cầu chi tiết của phần mềm. Câu hỏi được lưu trữ vào một file Q&A (question and answer) và thường được đưa ra dưới dạng Yes/No question hoặc các câu hỏi lựa chọn đáp án nhằm giúp khách hàng tiết kiệm thời gian trả lời câu hỏi. Giai đoạn phân tích yêu cầu rất quan trọng trong quá trình kiểm thử bởi nó giúp ngăn chặn các bug xảy ra trong giai đoạn sớm, tiết kiệm thời gian và ngân sách.

Đầu vào của giai đoạn phân tích yêu cầu bao gồm tài liệu như tài liệu đặc tả yêu cầu, tài liệu thiết kế hệ thống, tài liệu về các tiêu chí chấp nhận sản phẩm...Đầu ra của giai đoạn phân tích yêu cầu bao gồm tài liệu chứa các câu hỏi và câu trả lời liên quan, tài liệu báo cáo tính khả thi của việc kiểm thử phần mềm.

**Giai đoạn 2: Test planning – Lập kế hoạch kiểm thử**

Lập kế hoạch kiểm thử nhằm xác định một số yếu tố quan trọng sau:

* Xác định phạm vi dự án: Bao gồm thời gian dự kiến cho các dự án, các công việc cụ thể trong từng giai đoạn, và lập lịch trình thực hiện để đảm bảo phù hợp với toàn bộ đội dự án.
* Xác định phương pháp tiếp cận**:** Đưa ra quyết định về cách tiếp cận kiểm thử dựa trên nhiều yếu tố như thời gian ước lượng, yêu cầu chất lượng từ phía khách hàng, công nghệ sử dụng, lĩnh vực của hệ thống/sản phẩm.
* Xác định các nguồn lực: Bao gồm con người (số lượng và kinh nghiệm của tester và nhóm phát triển) và thiết bị (số lượng và loại máy tính, thiết bị di động cần thiết).
* Lên kế hoạch thiết kế công việc kiểm thử: Bao gồm việc liệt kê các chức năng cần kiểm thử, thiết kế công việc kiểm thử cho mỗi chức năng, xác định điều kiện bắt đầu và kết thúc cho việc kiểm thử.

Đầu ra của giai đoạn lập kế hoạch kiểm thử bao gồm các tài liệu như kế hoạch kiểm thử (test plan), ước lượng kiểm thử (test estimation), lịch trình kiểm thử (test schedule).

**Giai đoạn 3: Test case development – Thiết kế kịch bản kiểm thử**

Giai đoạn thiết kế kịch bản kiểm thử diễn ra sau khi đã hoàn thành việc lập kế hoạch. Trong giai đoạn này, người kiểm thử viên bắt tay vào việc viết ra các test case (trường hợp kiểm thử, các kịch bản kiểm thử) chi tiết dựa vào việc vận dụng các test design technique (kỹ thuật thiết kế kịch bản kiểm thử). Test case được chia nhỏ ra theo các điều kiện miêu tả trong tài liệu đặc tả dự án nhằm mục đích bao phủ được tất cả các trường hợp kiểm thử có thể xảy ra. Cùng với việc tạo ra các test case chi tiết, đội kiểm thử cũng cần chuẩn bị trước các dữ liệu kiểm thử cho các trường hợp cần thiết. Sau khi hoàn thành, test case cần được kiểm tra lại bởi thành viên khác trong đội kiểm thử hoặc test leader nhằm tránh những sai sót trong tập hợp test case.

Đầu vào của giai đoạn này bao gồm tài liệu đặc tả dự án đã được cập nhật, các tài liệu đánh giá tính khả thi của việc kiểm thử tự động (nếu có). Đầu ra của giai đoạn này bao gồm test design (bản thiết kế kịch bản test), test case (kịch bản test chi tiết), test automation script (kịch bản test tự động).

**Giai đoạn 4: Test environment setup – Thiết lập môi trường kiểm thử**

Thiết lập môi trường kiểm thử là một phần quan trọng của vòng đời phát triển phần mềm. Về cơ bản, môi trường kiểm thử được quyết định dựa trên những điều kiện kiểm thử phần mềm. (Ví dụ server, client, network...). Đây là hoạt động độc lập và có thể được bắt đầu song song với giai đoạn thiết kế kịch bản kiểm thử. Kiểm thử viên cần chuẩn bị một vài test case để kiểm tra xem môi trường cài đặt đã sẵn sàng cho việc kiểm thử hay chưa.

Đầu vào của giai đoạn này là test plan, smoke test case, test data. Đầu ra của giai đoạn này là môi trường đã được cài đặt và sẵn sàng cho việc kiểm thử, các kết quả của smoke test case.

**Giai đoạn 5: Test execution – Thực hiện kiểm thử**

Giai đoạn thực hiện kiểm thử diễn ra sau khi đã hoàn thành giai đoạn viết test case và môi trường kiểm thử đã sẵn sàng. Trong giai đoạn này, người kiểm thử viên thực hiện kiểm thử phần mềm, đánh giá kết quả test, báo cáo các lỗi của phần mềm. Sau khi lập trình viên sửa lỗi xong, kiểm thử viên thực hiện test lại các trường hợp lỗi và theo dõi trạng thái của lỗi đến khi được sửa thành công.

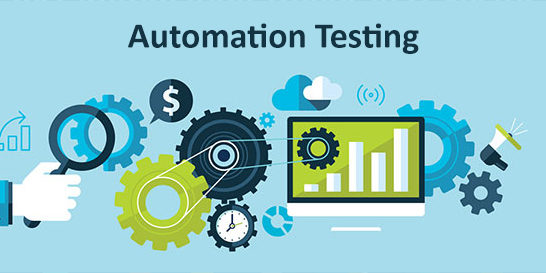
Đầu vào của giai đoạn này là tài liệu test plan, test case, test data. Đầu ra của giai đoạn này là danh sách lỗi, báo cáo thực hiện các trường hợp kiểm thử.

**Giai đoạn 6: Test cycle closure – Đóng chu trình kiểm thử**

Đây là giai đoạn cuối cùng trong vòng đời kiểm thử phần mềm. Ở giai đoạn này, đội kiểm thử viên họp bàn nhằm thực hiện tổng kết, báo cáo kết quả và đánh giá các tiêu chí hoàn thành như phạm vi kiểm tra, chất lượng, chi phí, thời gian, mục tiêu kinh doanh quan trọng. Ngoài ra, test cycle closure cũng bao gồm hoạt động thảo luận tất cả những điểm tốt, điểm chưa tốt và rút ra bài học kinh nghiệm cho những dự án sau, giúp cải thiện quy trình kiểm thử. Cùng với đó, kịch bản kiểm thử và lỗi sẽ được phân loại và đánh giá mức độ nghiêm trọng. Sau khi hoàn thành những hoạt động trên, báo cáo đóng quy trình kiểm thử và các số liệu kiểm thử sẽ được chuẩn bị dựa trên các thông số trên.

## 1.2. Kiểm thử tự động

### 1.2.1. Khái niệm



Hình 1. 3. Kiểm thử tự động

Trong kiểm thử tự động, tester viết code / test scripts để tự động thực hiện kiểm thử. Tester sử dụng các công cụ tự động hóa phù hợp để xây dựng các kịch bản kiểm thử và xác minh phần mềm hoạt động đúng hay không. Mục tiêu là hoàn thành kiểm thử trong một khoảng thời gian ngắn hơn.

Kiểm thử tự động hoàn toàn dựa vào kiểm thử trước khi chạy tự động kịch bản để so sánh kết quả thực tế với kết quả dự kiến. Điều này giúp tester xác định xem một ứng dụng có hoạt động đúng như mong đợi hay không.

**Ưu và nhược điểm:**

*Ưu điểm*

* Quy trình kiểm thử nhanh chóng và hiệu quả
* Quá trình tự động có thể được ghi lại. Điều này cho phép sử dụng lại và thực hiện cùng loại hoạt động kiểm thử.
* Kiểm thử tự động được thực hiện bằng các công cụ phần mềm, do đó, có thể hoạt động liên tục mà không biết mệt mỏi.
* Có thể dễ dàng tăng năng suất, cung cấp kết quả kiểm thử nhanh và chính xác
* Kiểm thử tự động hỗ trợ các ứng dụng khác nhau
* Phạm vi kiểm thử có thể được tăng lên vì công cụ kiểm thử tự động hóa không bỏ qua các đơn vị kiểm thử, ngay cả đơn vị nhỏ nhất

*Nhược điểm*

* Không có yếu tố con người, thật khó để hiểu sâu hơn về các khía cạnh trực quan của giao diện người dùng như màu sắc, phông chữ, kích thước, độ tương phản hoặc kích thước nút bấm, layout...
* Các công cụ để chạy kiểm thử tự động có thể tốn kém, có thể làm tăng chi phí kiểm thử của dự án.
* Công cụ kiểm thử tự động chưa phải là bằng chứng đầy đủ. Mỗi công cụ tự động hóa đều có những hạn chế của chúng làm, giảm phạm vi tự động hóa.
* Gỡ lỗi test scripts là một vấn đề lớn trong kiểm thử tự động. Kiểm thử bảo trì sẽ tốn kém.

### 1.2.2. Một số công cụ kiểm thử tự động phổ biến

* **Selenium**: Công cụ mã nguồn mở hỗ trợ kiểm thử tự động cho các ứng dụng web trên nhiều trình duyệt và hệ điều hành.
* **Katalon Studio**: Nền tảng kiểm thử toàn diện hỗ trợ cả kiểm thử web, mobile và API, dễ sử dụng cho cả tester không chuyên.
* **Appium**: Công cụ kiểm thử tự động dành cho ứng dụng di động (Android và iOS), hỗ trợ đa nền tảng.
* **Cypress**: Công cụ kiểm thử hiện đại cho ứng dụng web, nổi bật với tốc độ nhanh và dễ tích hợp CI/CD.
* **LambdaTest**: Nền tảng kiểm thử đám mây giúp chạy kiểm thử trên nhiều trình duyệt và thiết bị khác nhau.
* **Postman**: Công cụ phổ biến để kiểm thử API, hỗ trợ gửi yêu cầu, xác thực phản hồi và tự động hóa kiểm thử.
* **TestComplete**: Công cụ kiểm thử giao diện người dùng với khả năng tạo kịch bản kiểm thử bằng script hoặc không cần mã.
* **LoadTest**: Dùng để kiểm thử hiệu năng và đo lường khả năng chịu tải của hệ thống.
* **JMeter**: Công cụ kiểm thử hiệu năng mã nguồn mở, thường dùng để kiểm thử tải và đo lường hiệu suất của ứng dụng web và dịch vụ API.

### 1.2.3. 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

### 1.2.4. Các trường hợp nên áp dụng kiểm thử tự động

Không phải lúc nào cũng nên áp dụng kiểm thử tự động trong việc kiểm thử phần mềm, vì nhiều khi chi phí và thời gian cho việc kiểm thử tự động còn lớn hơn nhiều so với kiểm thử thủ công. Dưới đây là một số trường hợp áp dụng phương pháp kiểm thử tự động để đạt được hiệu quả cao về thời gian, chi phí cũng như chất lượng.

* Kiểm thử tự động rất hiệu quả khi áp dụng cho giai đoạn retest và test hồi quy. Hoặc cần chạy 1 số lượng testcases trong 1 thời gian ngắn.
* Việc phải test đi test lại nhiều không những gây tốn nguồn lực kiểm thử mà còn tạo cảm giác nhàm chán.
* Kiểm thử cần được thực hiện ở nhiều môi trường khác nhau.
* Những dự án có tính ổn định cao, đặc điểm kỹ thuật được xác định trước, test màn hình - chức năng không thay đổi trong tương lai.
* Những trường hợp kiểm thử xác nhận hoạt động cơ bản ( Di chuyển giữa các màn hình).
* Kiểm tra sự kết hợp giữa nhiều giá trị đầu vào ở 1 bước nào đó.
* Kiểm tra nhiều màn hình của dữ liệu đầu vào.
* Mục đầu vào của nhiều màn hình đăng ký.
* Khi muốn thực hiện performance test hoặc load test thì kiểm thử tự động gần như là biện pháp duy nhất.

## 1.3. Kiểm thử tự động tích hợp CI/CD

### 1.3.1. Giới thiệu về hệ thống CI/CD

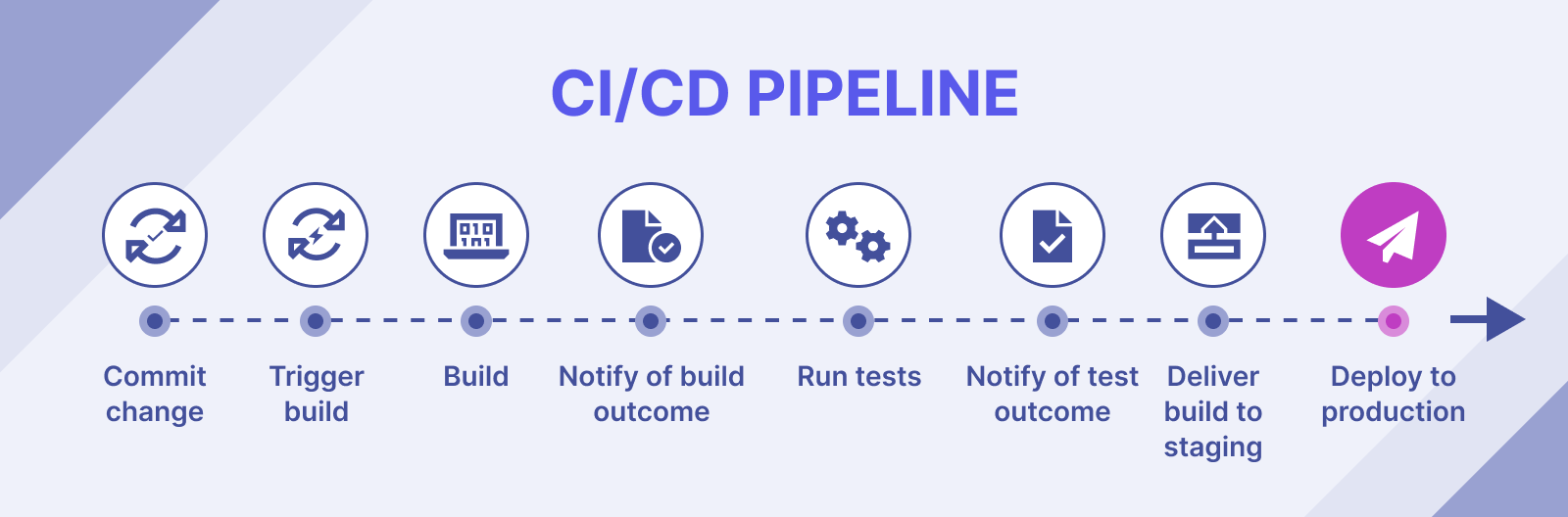
CI là viết tắt của **Continuous Integration** (tích hợp liên tục), CD là viết tắt của **Continuous Delivery** (chuyển giao liên tục) hoặc **Continuous Deployment** (triển khai liên tục).

Khái niệm CI/CD thường đề cập đến việc tự động hóa trong quy trình phát triển phần mềm và chuyển giao sản phẩm, giúp cho việc tích hợp diễn ra nhanh hơn và sản phẩm hoàn thiện được chuyển đến người dùng trong thời gian ngắn nhất.

Hiện nay, CI/CD đã được áp dụng rộng rãi vào quy trình làm việc của các doanh nghiệp làm trong lĩnh vực IT, song hành cùng với DevOps và Agile.

CI/CD thường được triển khai thông qua các công cụ như Jenkins, GitLab CI, GitHub Actions, CircleCI... và được tích hợp trực tiếp với hệ thống kiểm thử tự động như Selenium, JUnit, TestNG, ...

### 1.3.2. Luồng hoạt động của CI/CD



Hình 1. 4. Luồng hoạt động của CI/CD

Quy trình hoạt động của CI/CD được tự động hóa và diễn ra theo một chuỗi các bước liên tục nhằm đảm bảo chất lượng và tính ổn định của phần mềm. Một pipeline CI/CD điển hình thường bao gồm các bước sau:

*Tự động đồng bộ mã nguồn:* Mỗi khi lập trình viên đẩy (push) hoặc cập nhật mã nguồn lên hệ thống quản lý phiên bản như Git, pipeline sẽ tự động kích hoạt và lấy phiên bản mới nhất của mã.

*Thực hiện kiểm thử:* Trước khi được tích hợp, mã nguồn sẽ được kiểm tra thông qua các bộ kiểm thử tự động như unit test, integration test hoặc test chức năng. Mục tiêu là phát hiện sớm các lỗi logic hoặc xung đột trong mã.

*Tích hợp mã vào hệ thống chính*: Khi các bài kiểm thử được thực hiện thành công, mã sẽ được hợp nhất vào nhánh chính của dự án (thường là nhánh main hoặc develop), bảo đảm tính ổn định và liên tục của sản phẩm.

*Tiến hành xây dựng (build)*: Hệ thống sẽ biên dịch và đóng gói mã nguồn thành các file thực thi hoặc deployable artifacts. Đây là giai đoạn kiểm tra tính khả thi của phiên bản phần mềm mới.

*Triển khai tự động*: Nếu giai đoạn build thành công, hệ thống có thể tự động triển khai phần mềm lên môi trường thử nghiệm (staging) hoặc môi trường chính thức (production), tùy thuộc vào cấu hình pipeline.

*Giám sát và đánh giá hiệu năng*: Sau khi triển khai, phần mềm sẽ được theo dõi liên tục nhằm phát hiện các lỗi tiềm ẩn, cảnh báo sớm hoặc đánh giá hiệu suất vận hành trong thực tế.

### 1.3.3. Lợi ích của CI/CD

Việc tích hợp kiểm thử tự động với CI/CD mang lại nhiều lợi ích đáng kể trong phát triển phần mềm, bắt nguồn từ cách tiếp cận phát triển linh hoạt, lặp đi lặp lại:

*Tăng tính ổn định và đáng tin cậy*: Kiểm tra mã nguồn liên tục và triển khai phần mềm tự động giúp tăng tính ổn định và đáng tin cậy của sản phẩm phần mềm, giảm thiểu rủi ro cho quá trình triển khai.

*Giảm thiểu thời gian triển khai:* Tự động hóa quá trình triển khai phần mềm giúp giảm thiểu thời gian và công sức cho quá trình phát triển phần mềm.

*Đẩy nhanh quá trình phát triển*: Tự động hóa quá trình kiểm tra, tích hợp, xây dựng và triển khai giúp đẩy nhanh quá trình phát triển phần mềm.

*Tăng sự linh hoạt:* CI/CD cho phép triển khai các bản cập nhật nhỏ và thường xuyên hơn, giúp tăng sự linh hoạt cho quá trình phát triển phần mềm.

*Giảm thiểu lỗi:* Kiểm tra mã nguồn liên tục giúp phát hiện và sửa lỗi sớm hơn, giảm thiểu rủi ro cho quá trình triển khai.

*Tăng khả năng phục hồi:* Nếu có lỗi xảy ra trong quá trình triển khai, CI/CD cho phép phục hồi nhanh chóng bằng cách triển khai lại phiên bản trước đó.

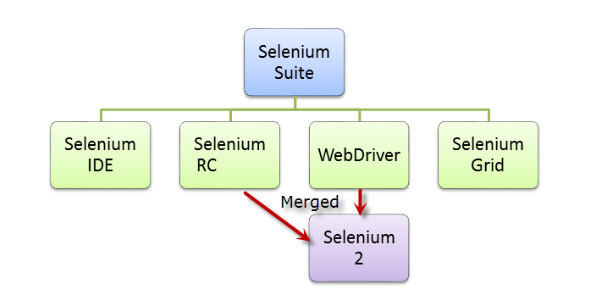
## 1.4. Tổng kết chương 1

Trong chương 1 em đã trình bày các kiến thức nền tảng về kiểm thử phần mềm, bao gồm khái niệm, nguyên tắc, quy trình kiểm thử phần mềm. Đồng thời, chương cũng giới thiệu kiểm thử tự động, các công cụ phổ biến và quy trình áp dụng hiệu quả. Bên cạnh đó, chương đã làm rõ vai trò của kiểm thử tự động trong quy trình CI/CD, giúp nâng cao chất lượng phần mềm và tối ưu quá trình phát triển. Những kiến thức này sẽ được vận dụng cụ thể trong Chương 2 khi tìm hiểu về công cụ Selenium WebDriver và hệ thống website dùng để kiểm thử.

# CHƯƠNG 2. TÌM HIỂU VỀ SELENIUM WEBDRIVER VÀ WEBSITE KIỂM THỬ

## 2.1. Công cụ kiểm thử tự động Selenium

### 2.1.1. Giới thiệu chung về Selenium



Hình 2. 1. Cấu trúc Selenium

Selenium là một phần mềm mã nguồn mở - một công cụ kiểm thử phần mềm tự động để kiểm thử các ứng dụng trên nền Web. Năm 2004, Selenium được phát triển bởi ThoughtWorks với cái tên ban đầu JavaScript Test Runner.

Đến năm 2007, tác giả Jason Huggins rời ThoughtWorks và gia nhập Selenium Team (thuộc Google), từ đó tiếp tục phát triển Selenium như hiện nay.

Tới thời điểm hiện tại, Selenium luôn nằm trong danh sách top 10 Testing Automation Tools nhờ các tính năng nổi bật:

* Selenium là mã nguồn mở. Do đó, mọi người có thể download source code về sử dụng, và thay đổi tùy theo nhu cầu.
* Cộng đồng sử dụng rộng rãi. Thường xuyên được phát triển và cải tiến mạnh mẽ.
* Hỗ trợ cho nhiều trình duyệt như Chrome, Firefox, Safari, Internet Explorer, Opera, và Microsoft Edge.
* Hỗ trợ trên nhiều hệ điều hành: MacOS, Linux, Windows và UNIX
* Các test case có thể được viết bằng nhiều ngôn ngư khác nhau như: Java, C#, Ruby, JavaScript, Python,…
* Với Selenium GRID, các test case có thể được thực thi trên nhiều máy khác nhau cùng một lúc, giúp cho quá trình thực thi test được diễn ra nhanh hơn.
* Có thể kết hợp với các hệ thống CI/CD như Jenkins, CircleCI để đẩy nhanh tiến độ kiểm tra, từ đó có thể rút ngắn thời gian release sản phẩm.
* Hỗ trợ mobile testing. Có thể kết hợp Selenium với một số tool như Appium, Selendroid để có thể tiến hành kiểm thử các ứng dụng mobile như: native, hybrid, và web mobile app.

### 2.1.2. Các thành phần chính của Selenium

Selenium không chỉ là một công cụ đơn lẻ mà là một bộ gồm 4 công cụ, mỗi công cụ đáp ứng nhu cầu kiểm thử khác nhau.

* Selenium IDE: Selenium Integrated Development Environment (IDE) là một plugin trên trình duyệt Chrome và Firefox. Ta có thể sử dụng chúng để ghi và phát lại (record and playback) các tương tác của người dùng theo một quy trình hay một test case nào đó.
* Selenium RC: Selenium Remote Control (RC), Selenium server khởi chạy và tương tác với trình duyệt web. Năm 2008, Selenium team đã quyết định gộp Selenium RC và WebDriver để tạo ra Selenium 2 với nhiều tính năng mạnh mẽ hơn, mà hiện nay phần lớn các project Selenium đều sử dụng.
* Selenium WebDriver: Selenium WebDriver là một automation testing tool dành riêng cho web, nó sẽ gửi lệnh khởi chạy và tương tác trực tiếp tới các trình duyệt.
* Selenium Grid: Selenium Grid được sử dụng để khởi chạy nhiều kịch bản test song song cùng một lúc, và có thể chạy trên nhiều máy, nhiều hệ điều hành và nhiều trình duyệt khác nhau.

### 2.1.3. Selenium WebDriver

Selenium WebDriver là một công cụ quan trọng trong bộ công cụ Selenium, được sử dụng để tự động hóa việc kiểm thử ứng dụng web trên nhiều trình duyệt khác nhau. Thành phần chính bao gồm:

* API (Application Programming Interface): Selenium WebDriver cung cấp một API cho phép lập trình viên tương tác với các phần tử trên trang web và thực hiện các hành động như nhấp chuột, nhập liệu, kiểm tra điều kiện. API này có thể được sử dụng với nhiều ngôn ngữ lập trình như Java, Python, C#.
* Driver: Selenium WebDriver sử dụng các driver tương ứng với từng trình duyệt như ChromeDriver cho Google Chrome, GeckoDriver cho Mozilla Firefox, WebDriver cho Microsoft Edge,... Driver này chịu trách nhiệm giao tiếp với trình duyệt và điều khiển các hành động từ mã kiểm thử.

Với nhiều ưu điểm vượt trội, Selenium WebDriver được ứng dụng rộng rãi trong kiểm thử tự động web. Selenium WebDriver cho phép kiểm tra chức năng của ứng dụng web bằng cách thực hiện các thao tác tương tự như người dùng thực hiện trên trình duyệt web. Hỗ trợ chạy kiểm thử trên nhiều trình duyệt web phổ biến như Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera, và các trình duyệt khác, điều này giúp đảm bảo rằng ứng dụng hoạt động đúng đắn trên mọi trình duyệt mà người dùng có thể sử dụng. Ngoài ra công cụ này có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux, cho phép kiểm tra tính nhất quán của ứng dụng trên nhiều môi trường khác nhau.

Selenium WebDriver có thể tích hợp với nhiều frameworks và công cụ kiểm thử như TestNG, JUnit, NUnit, để tự động hóa việc quản lý và thực thi bài kiểm thử, giúp tự động hóa quy trình kiểm thử và tạo ra các báo cáo chi tiết về kết quả kiểm thử.

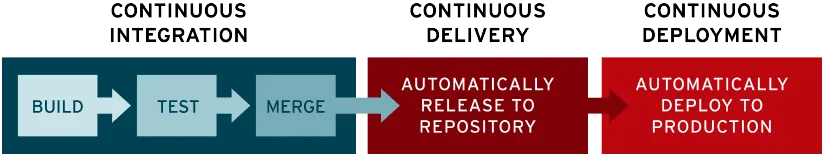
Selenium WebDriver có thể được sử dụng để kiểm tra hiệu suất và tương tác người dùng trên các ứng dụng web, từ đó giúp đảm bảo trải nghiệm người dùng cuối tốt nhất. Nó cung cấp khả năng kiểm tra thời gian phản hồi của trang, thao tác của người dùng, và các yếu tố khác ảnh hưởng đến trải nghiệm người

## 2.2. Tích hợp CI/CD bằng GitLab CI

### 2.2.1. CI/CD

**CI/CD** là một bộ đôi công việc, bao gồm CI (Continuous Integration) và CD (Continuous Delivery) hoặc Continuous Deployment (Triển khai liên tục), là quá trình tích hợp (integration) thường xuyên, nhanh chóng hơn khi code cũng như thường xuyên cập nhật phiên bản mới (delivery).

* **CI (Continuous Integration):** là một quy trình tự động hóa cho phép các nhà phát triển thường xuyên hợp nhất các thay đổi mã nguồn trở lại một nhánh chia sẻ hoặc “trunk”. Khi các cập nhật này được thực hiện, các bước kiểm tra tự động sẽ được kích hoạt để đảm bảo tính ổn định của các thay đổi mã nguồn đã hợp nhất.
* **CD ( Continuous Deployment ):** tự động hóa việc phát hành mã nguồn đã được xác nhận vào một kho lưu trữ sau khi tự động xây dựng và kiểm tra đơn vị và kiểm tra tích hợp trong CI. Vì vậy, để có quy trình Continuous Delivery hiệu quả, CI cần phải được tích hợp sẵn vào quy trình phát triển.



Hình 2. 2. Quy trình CI/CD

### 2.2.2. GitLab

GitLab là một hệ thống quản lý mã nguồn và dự án phần mềm có sự tăng cường của các công cụ quản lý DevOps. Nó cung cấp một nền tảng tích hợp để phát triển, kiểm thử, triển khai và theo dõi mã nguồn hiệu quả.

****

Hình 2. 3. GitLab

Các thành phần chính của GitLab CI/CD:

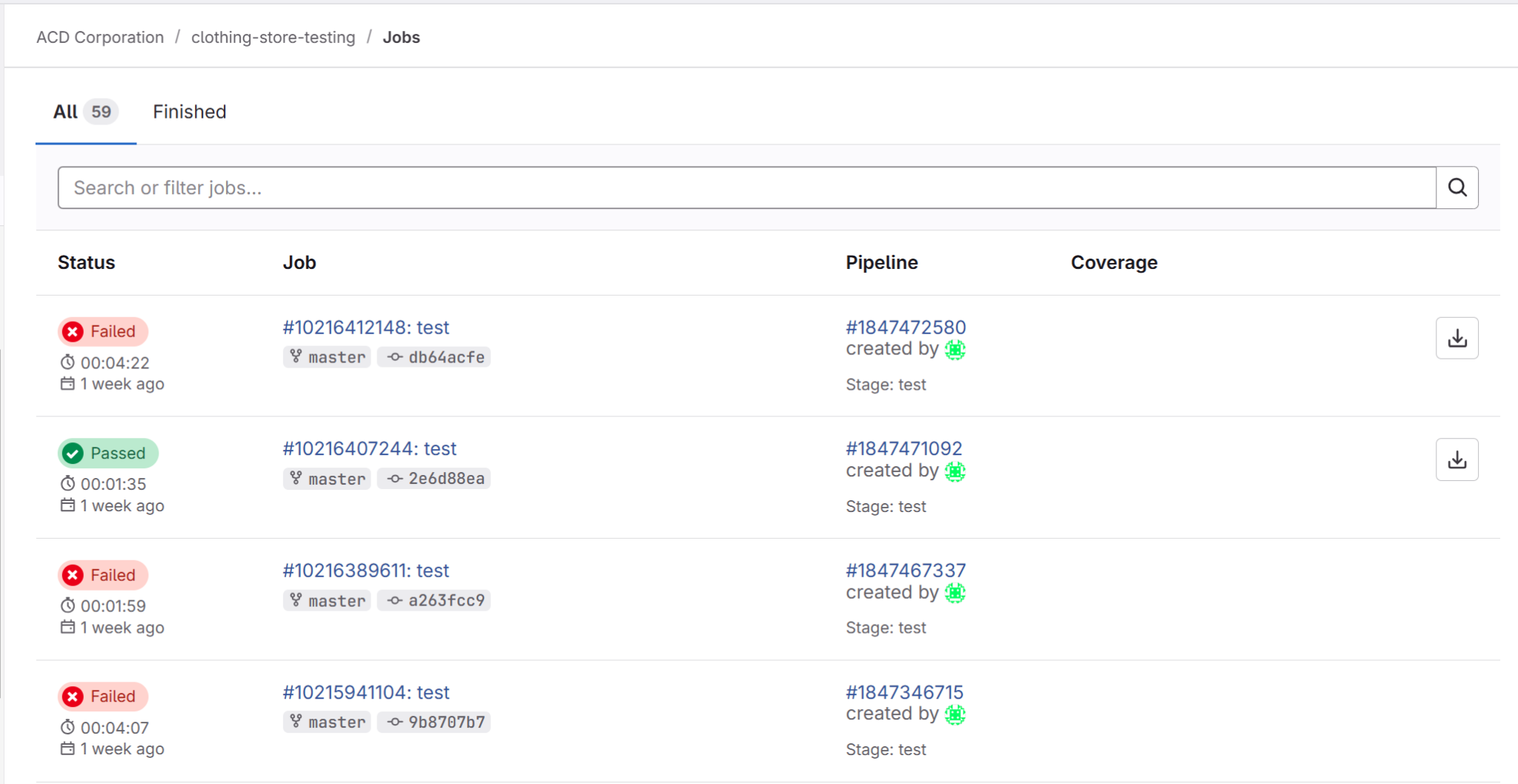
* Pipeline: Là một tập hợp các Jobs (công việc) được thực thi theo một thứ tự xác định. Mỗi pipeline có thể có nhiều stages (giai đoạn), và mỗi stage có thể có nhiều jobs.
* Job: Là một đơn vị công việc cụ thể, ví dụ như biên dịch mã nguồn, chạy kiểm thử, hoặc triển khai ứng dụng. Các jobs trong cùng một stage được thực thi song song, còn các stages được thực thi tuần tự.
* Runner: Là các agent chịu trách nhiệm thực thi các jobs. GitLab Runner có thể được cài đặt trên máy chủ riêng hoặc sử dụng các runners được cung cấp bởi GitLab.
* .gitlab-ci.yml: Là tệp cấu hình được lưu trữ trong kho mã nguồn (repository) của bạn. Tệp này xác định các pipelines, stages và jobs cần thực thi. Đây là nơi bạn định nghĩa logic CI/CD cho dự án của mình.

### 2.2.3. Git Flow



Hình 2. 4. Git Flow

Developer hoàn thành một task nào đó và push commit lên Gitlab để mọi người review. Khi đó Gitlab CI cũng bắt đầu thực hiện công việc mà nó được giao. Nó sử dụng file .gitlab-ci.yml nằm trong thư mục gốc của repo để cấu hình project sử dụng các Runner. Một pipeline CI sinh ra và report sẽ được hiển thị trên giao diện.



Hình 2. 5. Kết quả chạy job GitLab CI/CD

### 2.2.4. Pipeline

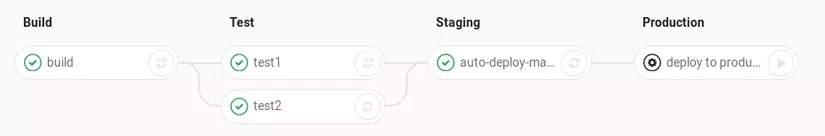
Pipeline bao gồm :

* Jobs : Các công việc được giao thực thi. ( Ví dụ : biên dịch mã hoặc chạy test )
* Stage : Xác định các thời điểm và cách thực hiện. ( Ví dụ : test chỉ chạy sau khi biên dịch thành công )

Pipeline hoạt động theo nguyên tắc sau :

* Tất cả các công việc trong cùng một stage được Runner thực hiện song song, nếu có đủ số lượng Runner đồng thời.
* Nếu Success, pipeline chuyển sang stage tiếp theo.
* Nếu Failed, pipeline sẽ dừng lại. Có một ngoại lệ là nếu job được đánh dấu làm thủ công, thì dù bị fail thì pipeline vẫn tiếp tục.

Bên dưới là ví dụ về đồ thị Pipeline thông thường :



Hình 2. 6. Đồ thị pipeline thông thường

Tóm lại, các bước để Gitlab CI hoạt động như sau :

* Thêm .gitlab-ci.yml vào thư mục gốc của repo
* Cấu hình Gitlab Runner

### 2.2.5. Triển khai CI/CD với GitLab

**Bước 1: Tạo Repository trên GitLab**

Đăng nhập vào GitLab <https://gitlab.com/users/sign_in/>

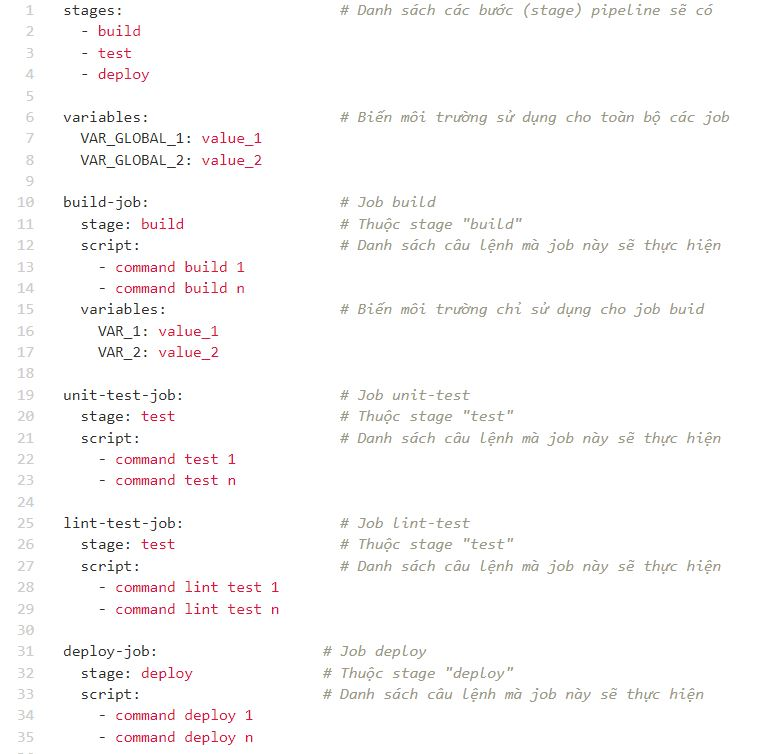
Tạo Repository mới: Nhấn vào nút "New Project" và điền thông tin cần thiết như tên dự án, mô tả, và chọn chế độ công khai hoặc riêng tư.

Cấu hình branch: Tạo các branch cần thiết cho quy trình phát triển, chẳng hạn như main cho mã nguồn chính và develop cho các tính năng đang phát triển.

**Bước 2: Cấu hình file .gitlab-ci.yml**

Tạo file .gitlab-ci.yml: Trong thư mục gốc của repository, tạo một file có tên .gitlab-ci.yml. Đây là file cấu hình chính cho pipeline CI/CD.

Định nghĩa Pipeline: Trong file này sẽ định nghĩa các bước cần thiết cho quy trình CI/CD. Ví dụ:



Hình 2. 7. Nội dung cơ bản của file “.gitlab-ci.yml”

**Bước 3: Thiết lập quyền truy cập**

Quản lý thành viên: Vào phần "Members" trong repository để thêm hoặc quản lý quyền truy cập cho các thành viên trong nhóm. Đảm bảo rằng họ có quyền thực hiện các thao tác cần thiết như push code và xem kết quả pipeline.

**Bước 4: Kiểm tra và chạy pipeline**

Commit và push code: Sau khi cấu hình xong, commit và push file .gitlab-ci.yml lên repository.

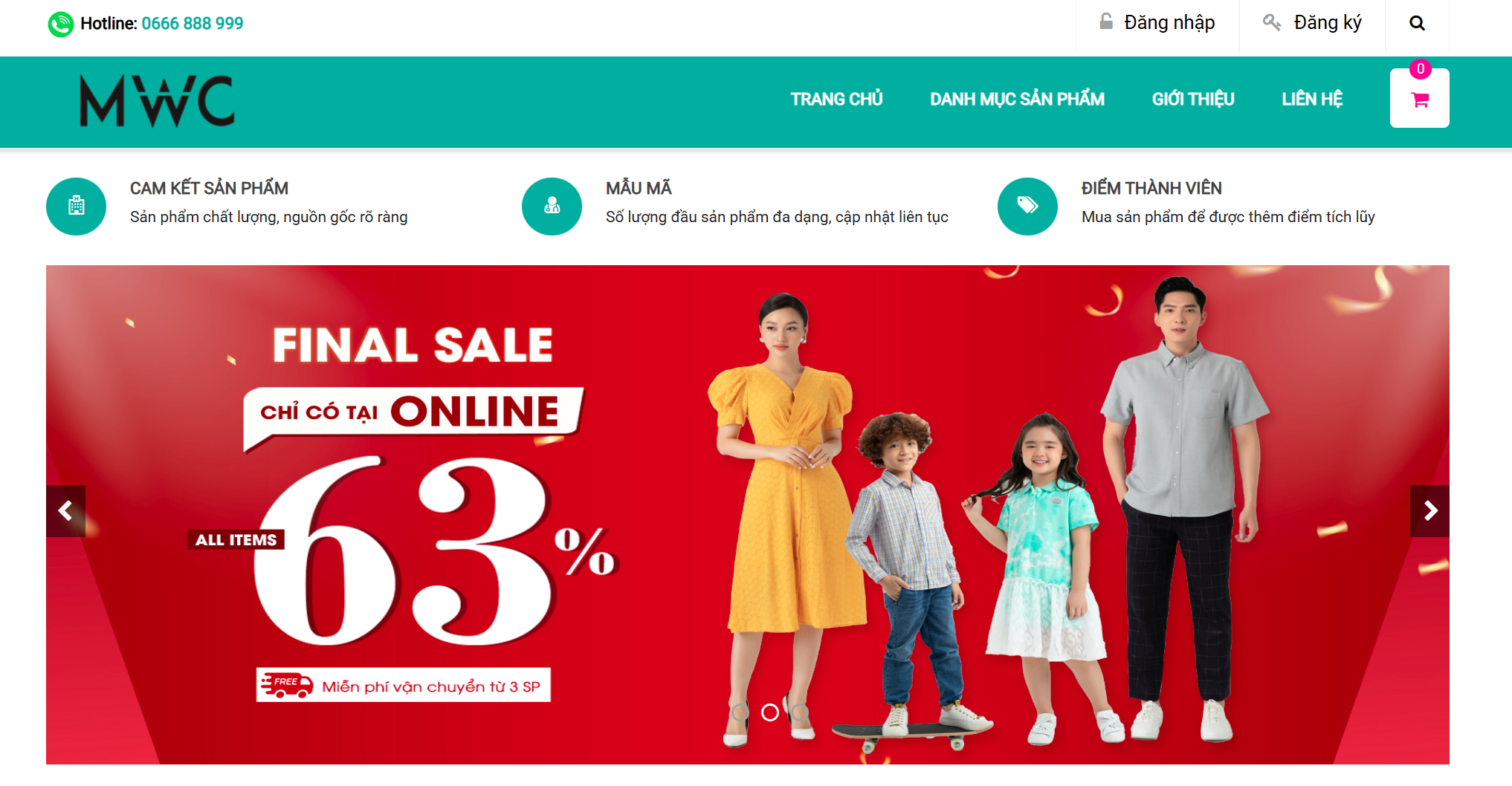
Theo dõi pipeline: Truy cập vào tab "CI/CD" trong repository để theo dõi trạng thái của pipeline sẽ thấy các bước build, test và deploy được thực hiện tự động.

## 2.3. Tổng quan về website kiểm thử

### 2.3.1. Giới thiệu

Trong thời đại công nghệ 4.0 phát triển bùng nổ và dần chi phối phần lớn cuộc sống con người như hiện nay, ứng dụng công nghệ và kinh tế chia sẻ trở thành một xu hướng chiến lược trong kinh doanh. Việc mua bán luôn là nhu cầu thiết yếu của cuộc sống và các trang web bán hàng với quy mô lớn nhỏ cũng lần lượt ra đời. Với ý tưởng thiết kế một marketplace phù hợp với việc tiếp cận các website bán hàng đơn lẻ và người sử dụng giúp quảng bá sản phẩm mua sắm, website bán quần áo được ra đời. Website là một marketplace trung gian kết nối người bán và người mua giúp người mua tìm kiếm sản phẩm mong muốn một cách dễ dàng hay chỉ đơn giản là lướt dạo xem các sản phẩm mà không bị nhàm chán. Và để đảm bảo chất lượng website bán quần áo hoạt động đúng với kỳ vọng của người dùng thì ta không thể bỏ qua bước kiểm thử trước khi triển khai nó.

Hệ thống em thực hiện kiểm thử là một website bán quần áo được xây dựng nhằm cung cấp một hệ thống bán hàng trực tuyến, giúp cho khách hàng dù ở xa vẫn có thể mua được sản phẩm mà không cần trực tiếp tới cửa hàng.

Hình 2. 8. Giao diện hệ thống

### 2.3.2. Các chức năng của website

Để đảm bảo hệ thống đáp ứng đầy đủ yêu cầu của người dùng và hoạt động ổn định, website được xây dựng với các chức năng chính chia theo từng nhóm vai trò, đồng thời tuân thủ các tiêu chí phi chức năng về hiệu năng, độ tin cậy và giao diện. Chi tiết được thể hiện trong các bảng sau:

Bảng 2. 1. Bảng mô tả các chức năng của hệ thống

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Loại** | **ID** | **Tên** | **Mô tả** |
| **A** | **Các chức năng chung** | | | |
| 1 | 1 | F.A.1 | Đăng ký | Nhập thông tin và tạo tài khoản người dùng. |
| 2 | 1 | F.A.2 | Đăng nhập | Đăng nhập vào hệ thống bằng tài khoản đã đăng ký. |
| 3 | 1 | F.A.3 | Đăng xuất | Đăng xuất khỏi hệ thống và quay về trang chưa đăng nhập. |
| 4 | 1 | F.A.4 | Cập nhật thông tin tài khoản | Thay đổi thông tin cá nhân đã đăng ký. |
| 5 | 1 | F.A.5 | Đổi mật khẩu | Thay đổi mật khẩu tài khoản người dùng. |
| **B** | **Các chức năng của Khách hàng** | | | |
| 6 | 3 | F.B.1 | Quản lý sản phẩm yêu thích | Thêm, xem, và xóa sản phẩm khỏi danh sách yêu thích. |
| 7 | 3 | F.B.2 | Quản lý giỏ hàng | Thêm, cập nhật số lượng, xóa sản phẩm trong giỏ hàng. |
| 8 | 1 | F.B.7 | Đặt hàng | Nhập thông tin và gửi đơn hàng. |
| 9 | 1 | F.B.8 | Xem lịch sử đặt hàng | Hiển thị danh sách các đơn hàng đã mua. |
| 10 | 1 | F.B.9 | Tìm kiếm sản phẩm | |  | | --- | |  |  |  | | --- | | Tìm sản phẩm theo từ khóa tên sản phẩm. | |
| 11 | 1 | F.B.10 | Lọc sản phẩm | Lọc sản phẩm theo nhà sản xuất hoặc khoảng giá. |
| **C** | **Các chức năng của Quản trị viên** | | | |
| 12 | 1 | F.C.1 | Quản lý danh mục sản phẩm | Xem, thêm, sửa, xóa, tìm kiếm danh mục sản phẩm. |
| 13 | 1 | F.C.2 | Quản lý sản phẩm | Xem, thêm, sửa, xóa, tìm kiếm sản phẩm trong hệ thống. |
| 14 | 3 | F.C.3 | Quản lý giao dịch | Xem, duyệt, hủy, xác nhận, tìm kiếm giao dịch. |
| 15 | 1 | F.C.4 | Quản lý khách hàng | Xem danh sách, chuyển trạng thái, tìm kiếm khách hàng. |

Bảng 2. 2. Bảng mô tả các yêu cầu phi chức năng của hệ thống

|  |  |  |
| --- | --- | --- |
| **STT** | **Tiêu chí** | **Mô tả** |
| 1 | Yêu cầu về hiệu năng | - Hệ thống cho phép truy cập dữ liệu thời gian thực. Các tác vụ thực hiện tức thời trong thời gian ngừng cho phép chấp nhận dưới 30s.  - Hệ thống đảm bảo phục vụ truy cập online ít nhất 80 người cùng một lúc. |
| 2 | Yêu cầu về độ tin cậy | - Yêu cầu mã hóa thông tin mật khẩu khi người dùng đăng nhập hệ thống, khai báo thông tin.  - Hệ thống đáp ứng các yêu cầu: Dữ liệu lưu trữ trong hệ thống được tự động sao lưu dự phòng theo lịch đã được lên trước, tránh mất dữ liệu. Dữ liệu hệ thống có thể kết xuất ra các thiết bị lưu trữ ngoài và phục hồi khi cần thiết. |
| 3 | Yêu cầu về giao diện | - Cung cấp một giao diện thân thiện phù hợp với quy trình nghiệp vụ hiện đang vận hành.  - Hệ thống đơn giản trong cài đặt và quản lý.  - Hệ thống hỗ trợ các trình duyệt phổ biến là Chrome, IE, Mozilla Firefox. |

## 2.4. Tổng kết chương 2

Trong Chương 2 em đã trình bày tổng quan về Selenium WebDriver - công cụ kiểm thử tự động mạnh mẽ cho ứng dụng web, cách cài đặt và cấu hình môi trường phát triển. Đồng thời giới thiệu quy trình tích hợp CI/CD bằng GitLab CI giúp tự động hóa kiểm thử hiệu quả. Cuối cùng là mô tả website bán quần áo được chọn làm đối tượng kiểm thử, làm nền tảng cho các bước triển khai ở chương 3.

# CHƯƠNG 3. TRIỂN KHAI KIỂM THỬ VÀ ĐÁNH GIÁ

## 3.1. Thiết lập môi trường

### 3.1.1. Cài đặt cấu trúc Framework

Framework kiểm thử tự động cho hệ thống bán hàng quần áo trực tuyến được xây dựng theo mô hình Data-driven kết hợp Page Object Model (POM).

Framework sử dụng công nghệ Selenium WebDriver, TestNG và Log4j2 để tổ chức, thực hiện và ghi log cho các ca kiểm thử.

*Page Object Model (POM):*

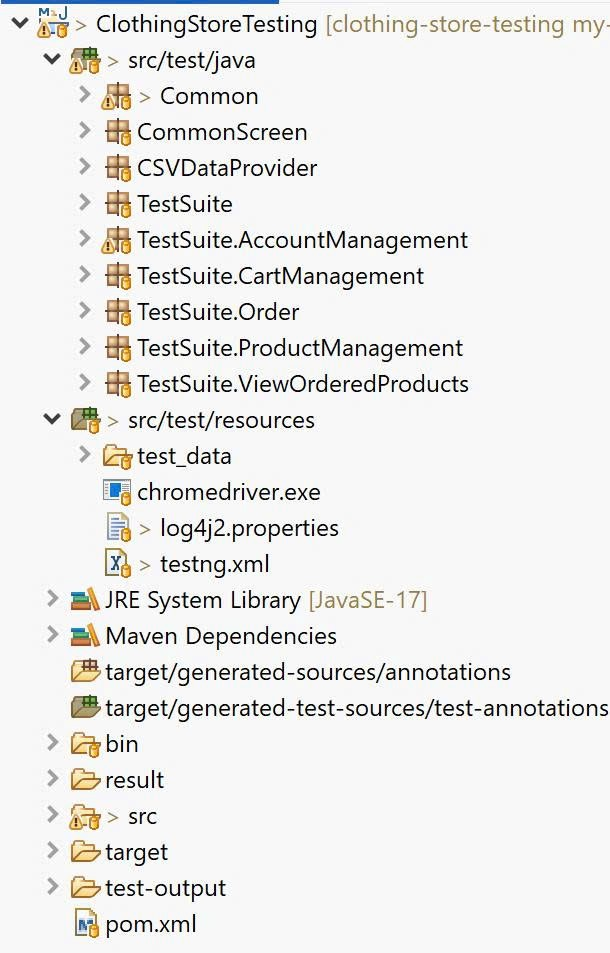
* Mỗi trang giao diện trong ứng dụng web được mô hình hóa dưới dạng một lớp Java.
* Giúp tái sử dụng mã nguồn, dễ bảo trì khi giao diện thay đổi.
* *Data-driven Testing:*
* Dữ liệu kiểm thử được tách riêng dưới dạng file CSV.
* Cho phép chạy cùng một test case với nhiều bộ dữ liệu khác nhau.
* */Common chứa các lớp:*
* Constant.java: Khai báo các hằng số (URL, thời gian chờ...).
* Initialization.java: Khởi tạo WebDriver và cấu hình môi trường chạy test.
* Utilities.java: Các hàm tiện ích như chờ, chụp ảnh màn hình, xử lý dữ liệu.

*/CommonScreen:* Chứa các lớp đại diện cho giao diện (Page Object)

* Lưu trữ các lớp màn hình như màn hình LoginScreen.java, CartScreen.java, RegisterScreen.java, ...
* Những lớp này sẽ lưu trữ thông tin của các phần tử trên giao diện như id, xpath, label, … và thao tác trên màn hình.

*/CSVDataProvider*: Lưu trữ các lớp dữ liệu, thực hiện đọc dữ liệu từ file CSV và cung cấp dữ liệu cho các ca kiểm thử tương ứng

*/TestSuite:* Lưu trữ các lớp kịch bản kiểm thử, thực thi các ca kiểm thử đã được thiết kế từ trước.

*/test\_data:* Chứa dữ liệu đầu vào cho các ca kiểm thử

Hình 3. 1. Cấu trúc framework dự án

### 3.1.2. Thiết lập môi trường

Bảng 3. 1. Bảng Môi trường phát triển và quản lý dự án

|  |  |  |
| --- | --- | --- |
| **STT** | **Công cụ / Công nghệ** | **Mô tả** |
| 1 | Eclipse IDE | Môi trường phát triển tích hợp để viết mã kiểm thử với giao diện thân thiện và hỗ trợ plugin. |
| 2 | Java | Ngôn ngữ lập trình được sử dụng để phát triển framework. |
| 3 | Apache Maven | Quản lý thư viện và cấu trúc dự án, sử dụng tệp pom.xml để khai báo phụ thuộc. |

Bảng 3. 2. Bảng Thư viện và công cụ hỗ trợ

|  |  |  |
| --- | --- | --- |
| **STT** | **Thư viện /Công cụ** | **Mô tả** |
| 1 | Selenium Java | Thư viện chính dùng để tự động hóa thao tác trên trình duyệt, phục vụ kiểm thử giao diện người dùng (UI Testing). |
| 2 | WebDriverManager | Hỗ trợ tự động tải và cấu hình các trình điều khiển cho các trình duyệt (ChromeDriver, GeckoDriver, v.v.), giúp đơn giản hóa quá trình cài đặt. |
| 3 | TestNG | Framework kiểm thử dùng để tổ chức, thực thi các test case, hỗ trợ tạo báo cáo kiểm thử. |
| 4 | Apache POI | Hỗ trợ đọc và ghi dữ liệu từ file Excel, CSV phục vụ kiểm thử hướng dữ liệu. |
| 5 | log4j2.properties | Ghi log đơn giản, hỗ trợ theo dõi kết quả kiểm thử hoặc lỗi phát sinh. |
| 6 | pom.xml | Quản lý thư viện, plugin, và thiết lập môi trường build cho Maven |

## 3.2. Kế hoạch kiểm thử

### 3.2.1. Xác định các yêu cầu kiểm thử

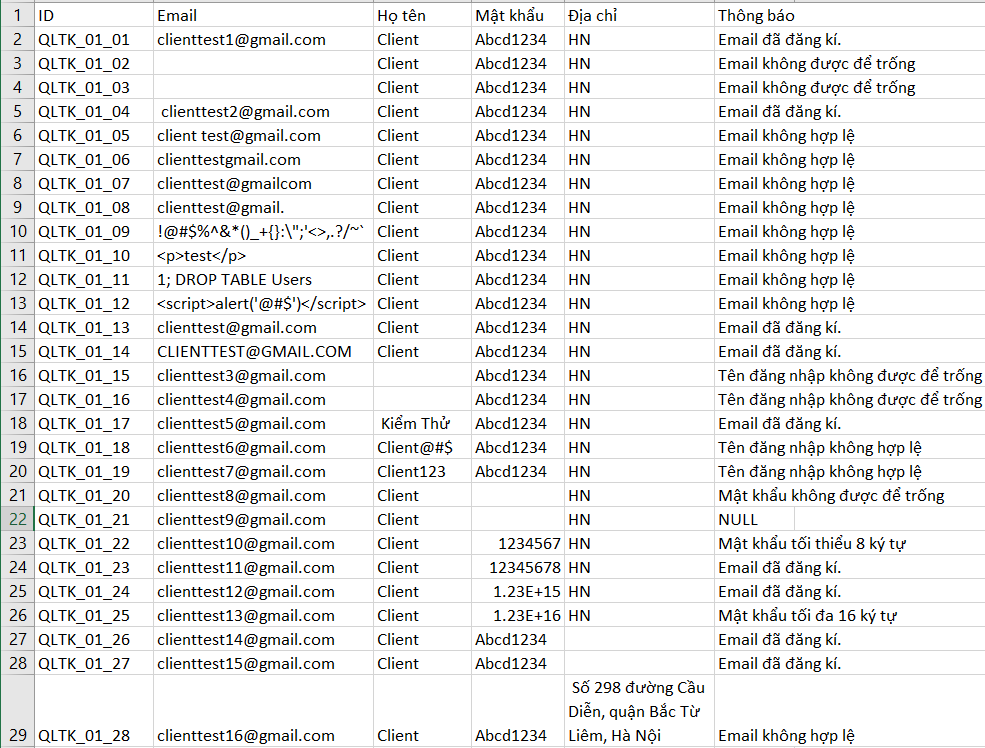
Để đảm bảo hệ thống website vận hành ổn định và đúng như mong đợi của người dùng, việc xác định rõ các yêu cầu kiểm thử cho từng chức năng là bước quan trọng trong kế hoạch kiểm thử. Các chức năng chính cần kiểm thử bao gồm:

* Đăng ký tài khoản
* Đăng nhập hệ thống
* Đăng xuất tài khoản
* Tìm kiếm sản phẩm
* Quản lý giỏ hàng
* Đặt hàng, xem lịch sử đặt hàng
* Quản lý sản phẩm (dành cho admin)

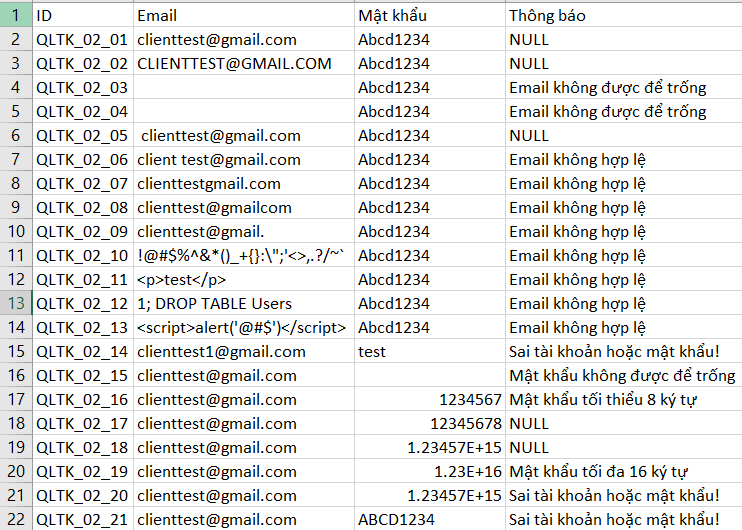
Với mỗi chức năng, các trường hợp kiểm thử (test case) đã được xây dựng chi tiết, bao gồm cả các tình huống hợp lệ và không hợp lệ, nhằm đảm bảo kiểm tra đầy đủ cả luồng chính và luồng ngoại lệ. Toàn bộ bảng test case tương ứng được trình bày ở **Phụ lục** để thuận tiện theo dõi và đối chiếu trong quá trình triển khai kiểm thử tự động.

### 3.2.2. Xây dựng dữ liệu kiểm thử

Dữ liệu kiểm thử được chuẩn bị dưới dạng file *.csv* nhằm phục vụ cho quá trình kiểm thử chức năng.



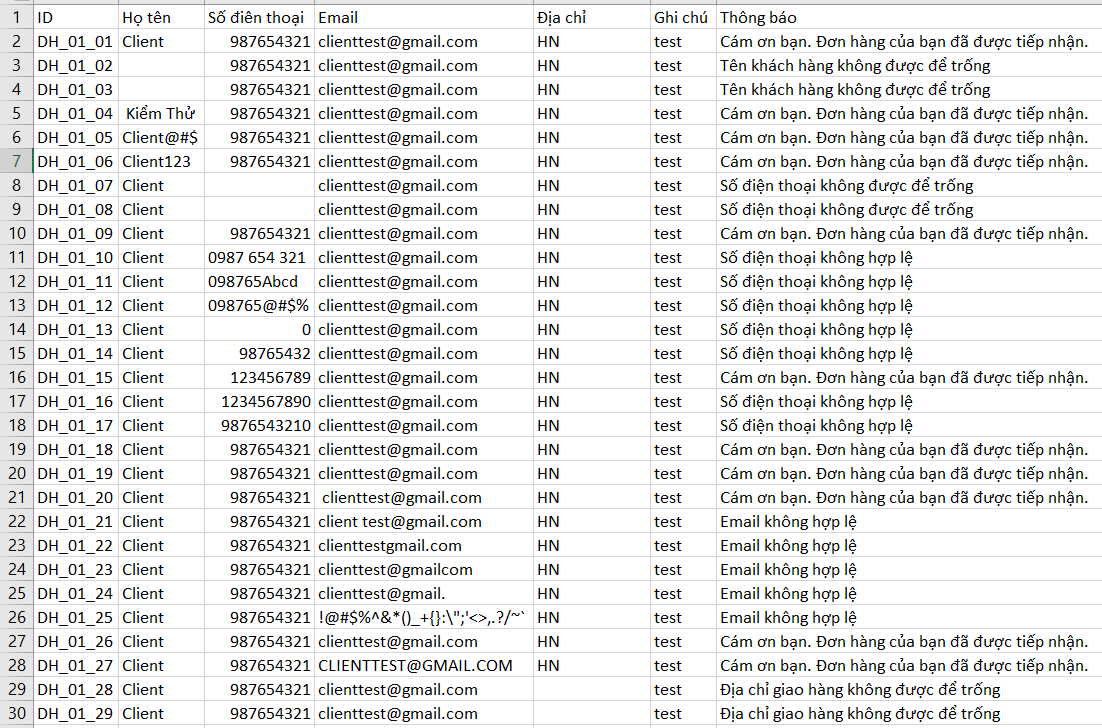
Hình 3. 2. Dữ liệu kiểm thử đăng ký



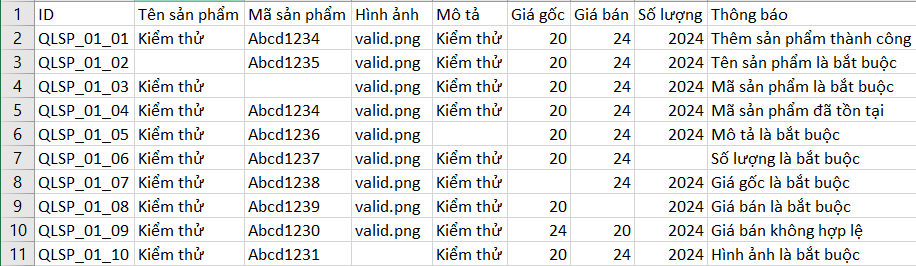
Hình 3. 3. Dữ liệu kiểm thử đăng nhập



Hình 3. 4. Dữ liệu kiểm thử đặt hàng khi đã login



Hình 3. 5. Dữ liệu kiểm thử đặt hàng khi chưa login



Hình 3. 6. Dữ liệu kiểm thử thêm sản phẩm

## 3.3. Xây dựng các test script

Để triển khai các ca kiểm thử tự động cho hệ thống quản trị website, em đã thực hiện theo mô hình POM (Page Object Model) nhằm tổ chức mã nguồn một cách rõ ràng và dễ bảo trì. Dưới đây là các bước cụ thể mà em đã thực hiện:

**Bước 1:** Tạo Page Object cho từng chức năng. Mỗi trang trong ứng dụng sẽ có một lớp Page Object riêng, chứa các phương thức để tương tác với các phần tử trên trang đó.

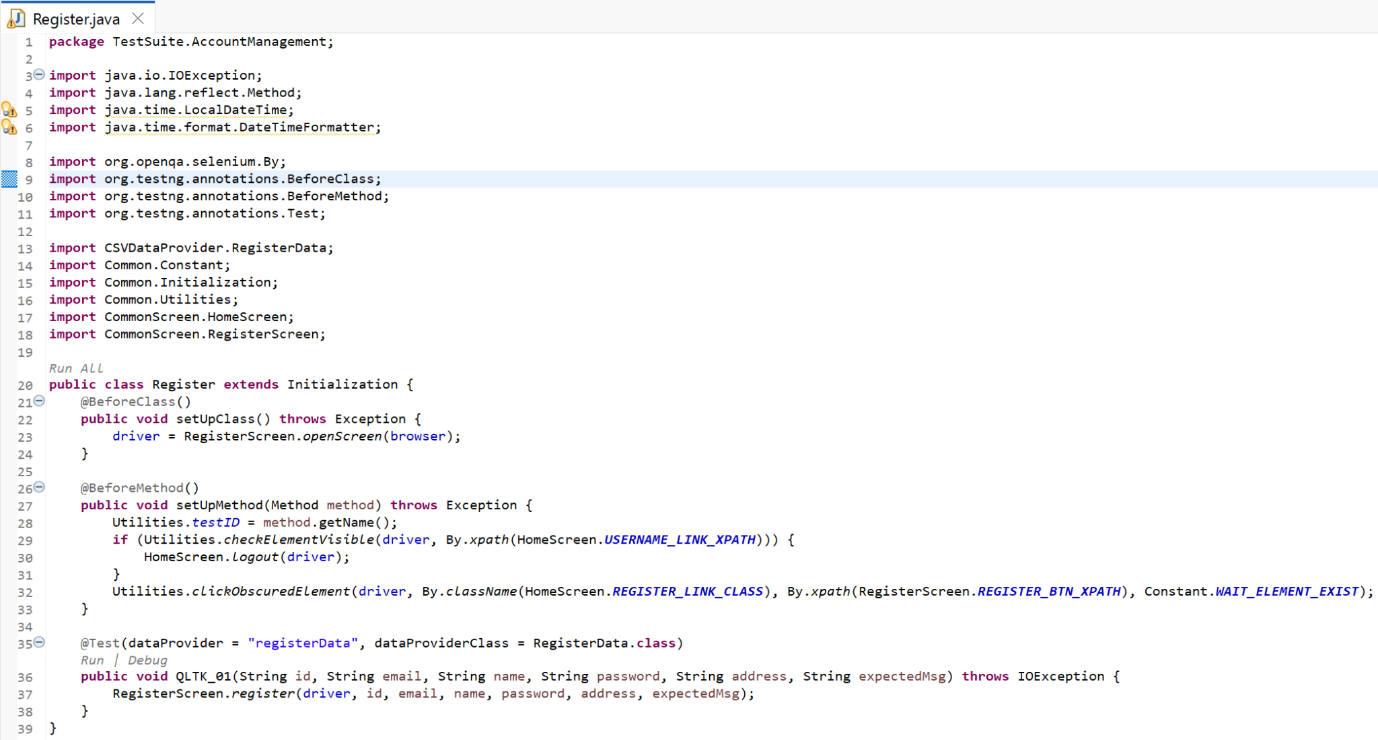
**Bước 2:** Tạo lớp kiểm thử sử dụng TestNG và triển khai các kịch bản kiểm thử, mỗi bước kiểm thử tương ứng với một phương thức Java.

**Bước 3:** Thực thi các ca kiểm thử, Sử dụng TestNG để thực thi các ca kiểm thử đã được định nghĩa. Kết quả kiểm thử sẽ được ghi lại và có thể xem trên giao diện trình duyệt hoặc trong các báo cáo được tạo ra.

**TestSuite.AccountManagement – Kiểm thử quản lý tài khoản**

***Kịch bản kiểm thử chức năng đăng ký***

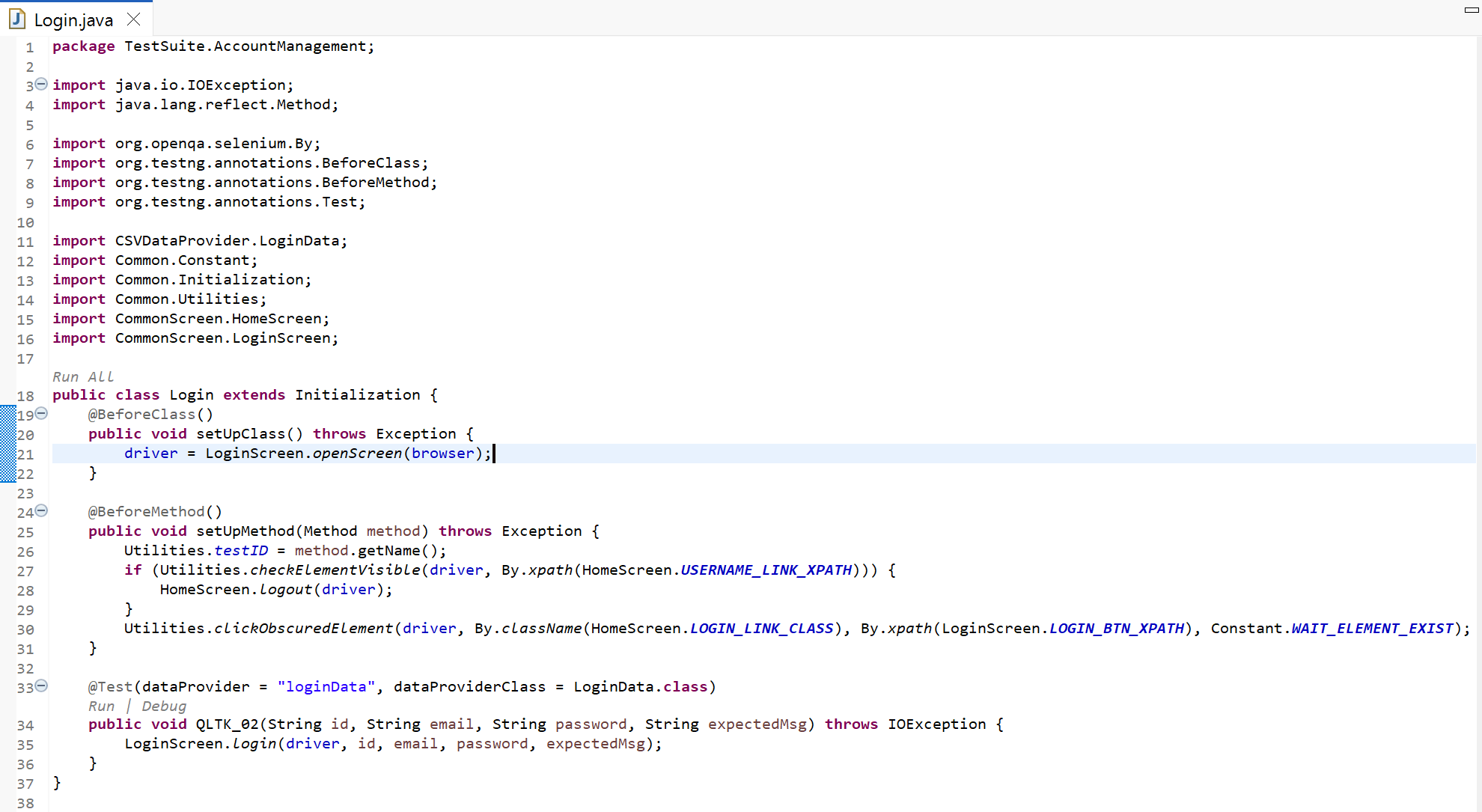
* **Bước 1:** Tạo lớp Register.java chứa kịch bản kiểm thử đăng ký tài khoản.
* **Bước 2:** Sử dụng RegisterScreen để mở giao diện đăng ký và xử lý hành động nhập liệu.
* **Bước 3:** Lấy dữ liệu kiểm thử từ tệp CSV (RegisterData.csv) chứa các trường hợp kiểm thử khác nhau.
* **Bước 4:** Thực hiện điền thông tin đăng ký như họ tên, email, mật khẩu, địa chỉ.
* **Bước 5:** Gửi form đăng ký và kiểm tra thông báo trả về có khớp với kết quả mong đợi (expectedMsg).



Hình 3. 7. Mã nguồn kiểm thử chức năng đăng ký

***Kịch bản kiểm thử chức năng đăng nhập***

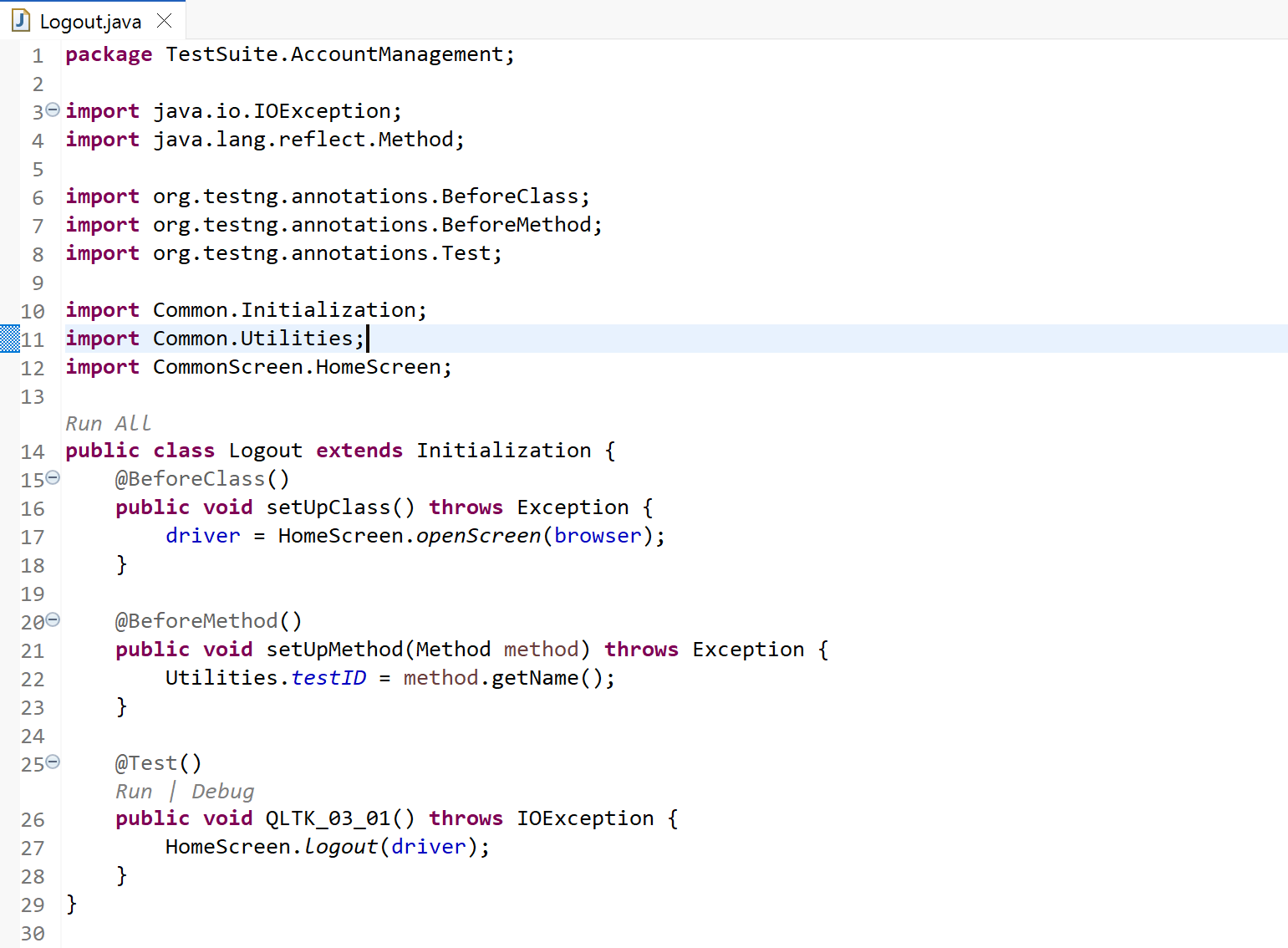
* **Bước 1:** Tạo lớp Login.java chứa kịch bản kiểm thử tính năng đăng nhập.
* **Bước 2:** Dùng LoginScreen.openScreen() để mở giao diện đăng nhập trên trình duyệt.
* **Bước 3:** Nếu đang đăng nhập → tự động đăng xuất để đảm bảo trạng thái ban đầu.
* **Bước 4:** Đọc dữ liệu kiểm thử từ LoginData.csv (email, mật khẩu, kết quả mong đợi).
* **Bước 5:** Điền thông tin đăng nhập và gửi form.
* **Bước 6:** So sánh thông báo hệ thống với expectedMsg để xác định Pass/Fail.



Hình 3. 8. Mã nguồn kiểm thử chức năng đăng nhập

***Kịch bản kiểm thử chức năng đăng xuất***

* **Bước 1:** Tạo lớp Logout.java để thực hiện kiểm thử chức năng đăng xuất.
* **Bước 2:** Sử dụng HomeScreen.openScreen() để mở giao diện chính sau khi đăng nhập.
* **Bước 3:** Gán mã test case tương ứng với tên phương thức (sử dụng method.getName()).
* **Bước 4:** Thực hiện thao tác đăng xuất bằng hàm HomeScreen.logout(driver).
* **Bước 5:** Xác nhận người dùng được chuyển về trang chưa đăng nhập (có thể là trang chủ hoặc trang login).

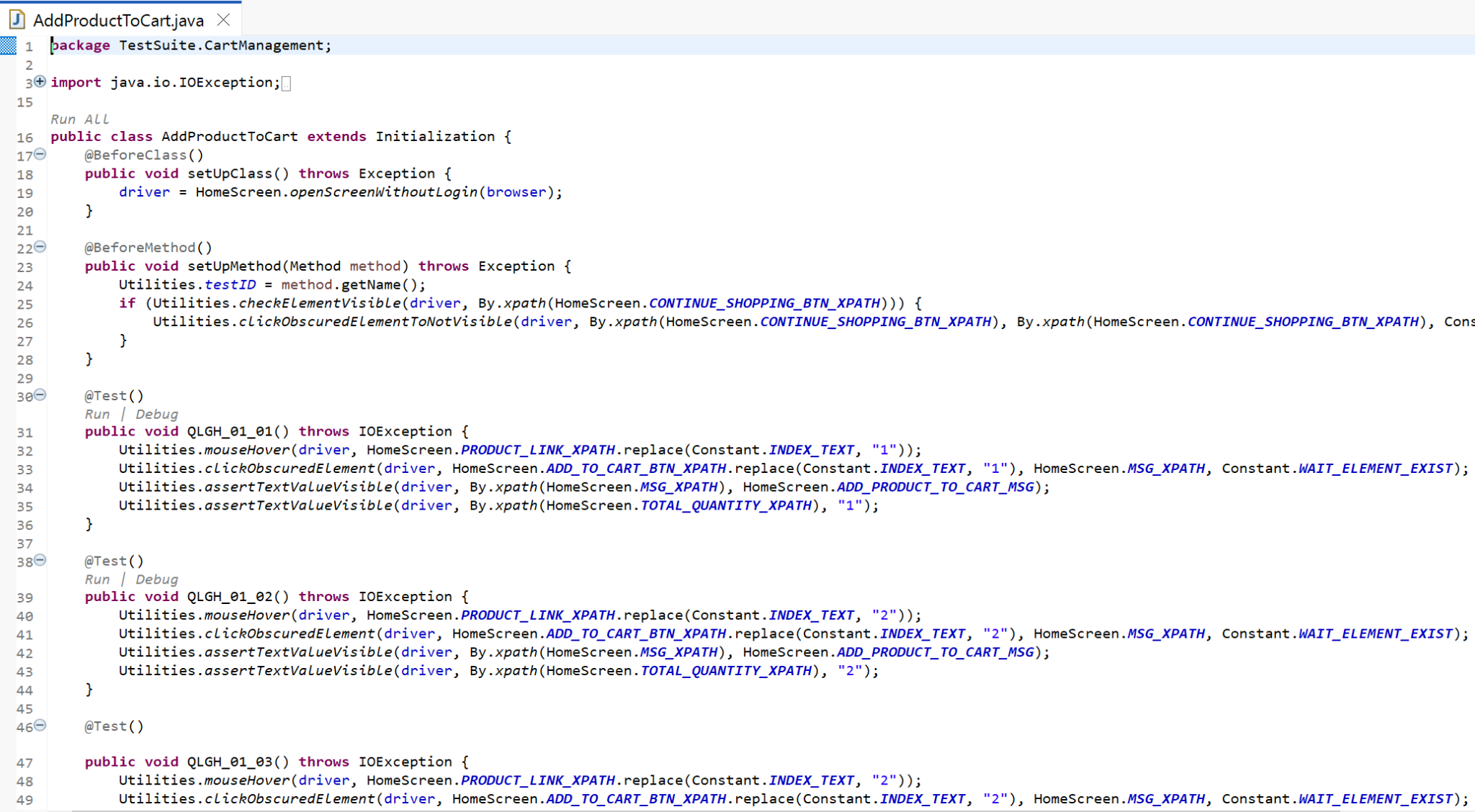


Hình 3. 9. Mã nguồn kiểm thử chức năng đăng xuất

**TestSuite.CartManagement – Kiểm thử chức năng đặt hàng**

***Kịch bản kiểm thử chức năng thêm sản phẩm vào giỏ hàng***

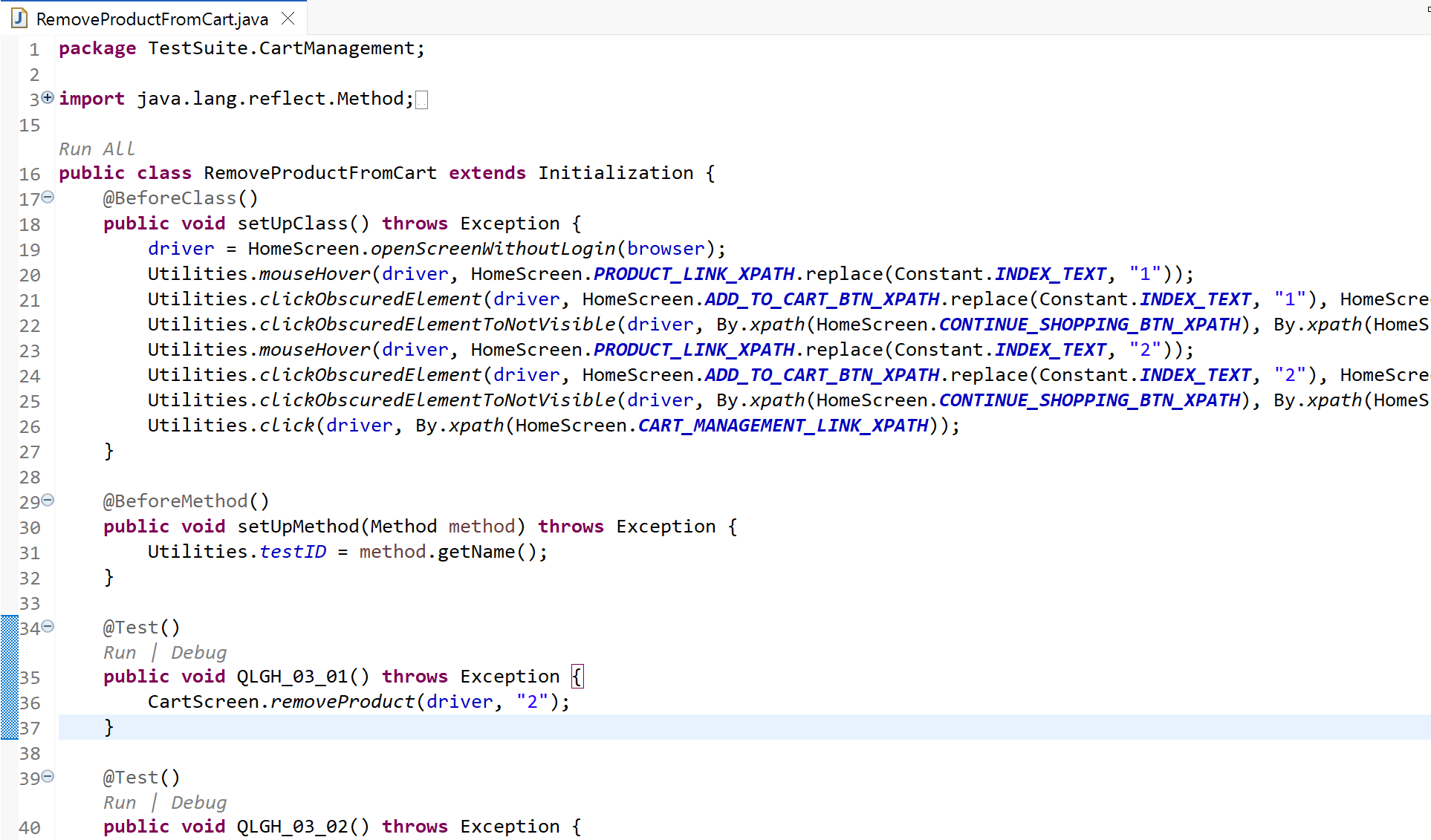
* **Bước 1:** Tạo lớp AddProductToCart.java kiểm thử thêm sản phẩm.
* **Bước 2:** Mở trang chủ không cần đăng nhập bằng openScreenWithoutLogin().
* **Bước 3:** Đóng popup “Tiếp tục mua sắm” nếu hiển thị.
* **Bước 4:** Hover chuột vào sản phẩm và nhấn "Thêm vào giỏ hàng".
* **Bước 5:** Kiểm tra thông báo và số lượng sản phẩm trong giỏ hàng.



Hình 3. 10. Mã nguồn kiểm thử chức năng thêm sản phẩm vào giỏ hàng

***Kịch bản kiểm thử chức năng xóa sản phẩm khỏi giỏ hàng***

* **Bước 1:** Tạo lớp RemoveProductFromCart.java kiểm thử chức năng xóa sản phẩm.
* **Bước 2:** Mở trang chủ thêm 2 sản phẩm vào giỏ hàng.
* **Bước 3:** Truy cập trang giỏ hàng.
* **Bước 4:** Thực hiện thao tác xóa sản phẩm bằng CartScreen.removeProduct().



Hình 3. 11. Mã nguồn kiểm thử chức năng xóa sản phẩm khỏi giỏ hàng

***Kịch bản kiểm thử chức năng cập nhật sản phẩm trong giỏ hàng***

* **Bước 1:** Tạo lớp UpdateProductInCart.java để kiểm thử cập nhật giỏ hàng.
* **Bước 2:** Mở trang chủ thêm sản phẩm đầu tiên vào giỏ hàng.
* **Bước 3:** Truy cập giao diện giỏ hàng.
* **Bước 4:** Gọi CartScreen.updateProduct() để tăng/giảm số lượng.



Hình 3. 12. Mã nguồn kiểm thử chức năng cập nhật sản phẩm trong giỏ hàng

***Kịch bản kiểm thử chức năng tìm kiếm sản phẩm***

* **Bước 1:** Tạo lớp SearchProduct.java để kiểm thử chức năng tìm kiếm.
* **Bước 2:** Mở trang chủ (không cần đăng nhập).
* **Bước 3:** Click biểu tượng tìm kiếm để hiển thị ô nhập từ khóa.
* **Bước 4:** Gọi hàm searchProduct trong HomeScreen với các đầu vào khác nhau: từ khóa rỗng, khoảng trắng, từ khóa hợp lệ và từ khóa sai.



Hình 3. 13. Mã nguồn kiểm thử chức năng tìm kiếm sản phẩm

***Kiểm thử chức năng xem đơn hàng đã đặt***

* **Bước 1:** Tạo lớp ViewOrderedProducts.java để kiểm thử chức năng xem đơn hàng.
* **Bước 2:** Trước mỗi ca kiểm thử, gán mã test bằng Utilities.testID = method.getName().
* **Bước 3:** Mở trang danh sách đơn hàng bằng OrderedProductsScreen.openScreen trong test XLSDH\_01\_01.



Hình 3. 14. Mã nguồn kiểm thử chức năng xem đơn hàng đã đặt

**TestSuite.Order**

***Kịch bản kiểm thử chức năng đặt hàng khi đã đăng nhập***

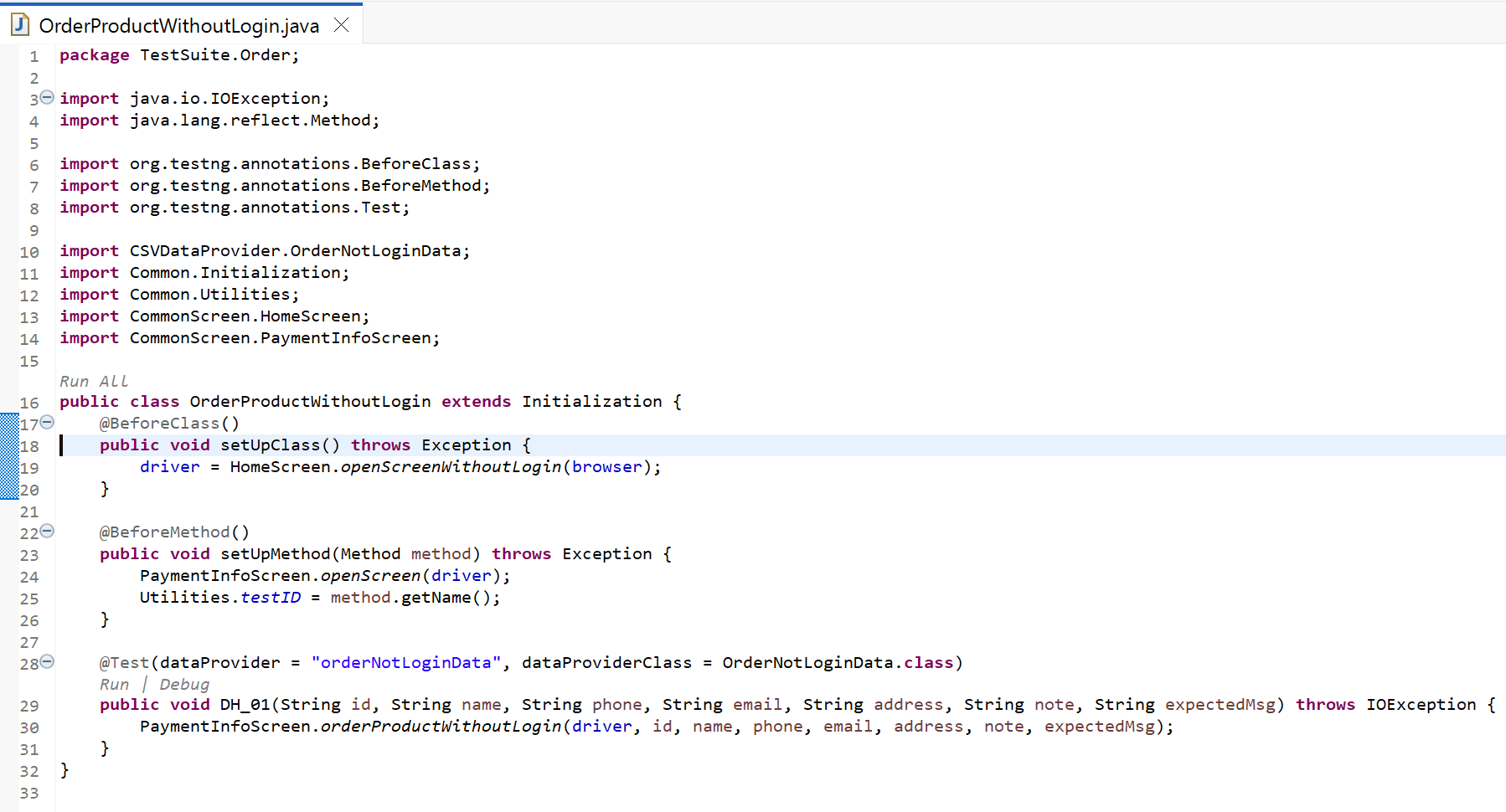
* **Bước 1:** Tạo lớp OrderProductWithLogin.java để kiểm thử đặt hàng.
* **Bước 2:** Mở trang chủ và đăng nhập
* **Bước 3:** Truy cập trang thanh toán bằng PaymentInfoScreen.openScreen.
* **Bước 4:** Đọc dữ liệu kiểm thử từ file CSV (OrderData.class).
* **Bước 5:** Gọi PaymentInfoScreen.orderProductWithLogin() để nhập thông tin và hoàn tất đơn hàng.



Hình 3. 15. Mã nguồn kiểm thử chức năng đặt hàng có đăng nhập

***Kịch bản kiểm thử chức năng đặt hàng không cần đăng nhập***

* **Bước 1:** Tạo lớp OrderProductWithoutLogin.java để kiểm thử đặt hàng không cần tài khoản.
* **Bước 2:** Mở trang chủ và truy cập trang thanh toán bằng PaymentInfoScreen.openScreen.
* **Bước 3:** Lấy dữ liệu từ file CSV (OrderNotLoginData.class).
* **Bước 4:** Gọi PaymentInfoScreen.orderProductWithoutLogin() để nhập thông tin và hoàn tất đơn hàng.

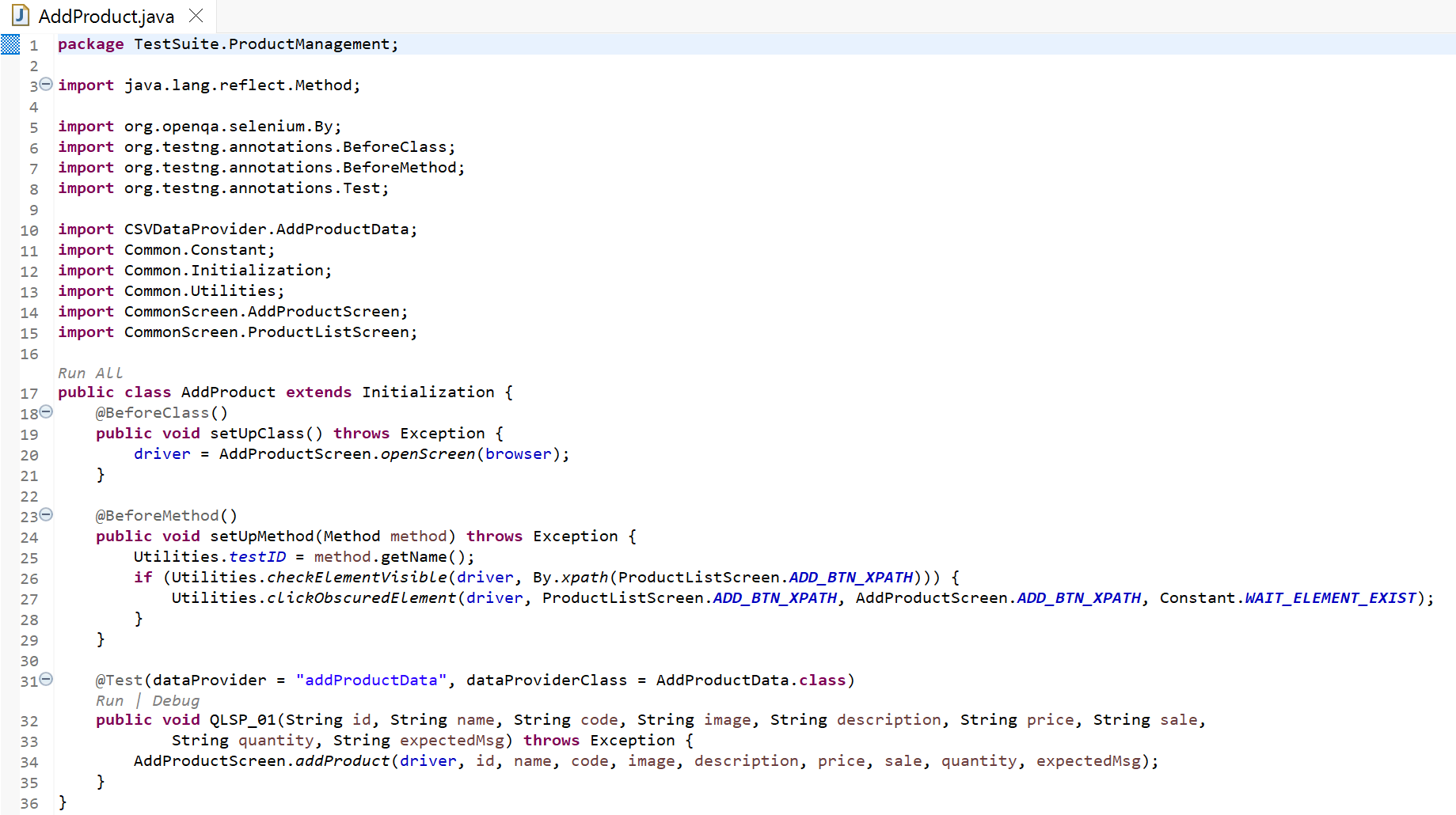


Hình 3. 16. Mã nguồn kiểm thử chức năng đặt hàng không cần đăng nhập

**Test Suite: ProductManagement – Kiểm thử quản lý sản phẩm (Admin)**

***Kịch bản kiểm thử chức năng thêm sản phẩm với tài khoản admin***

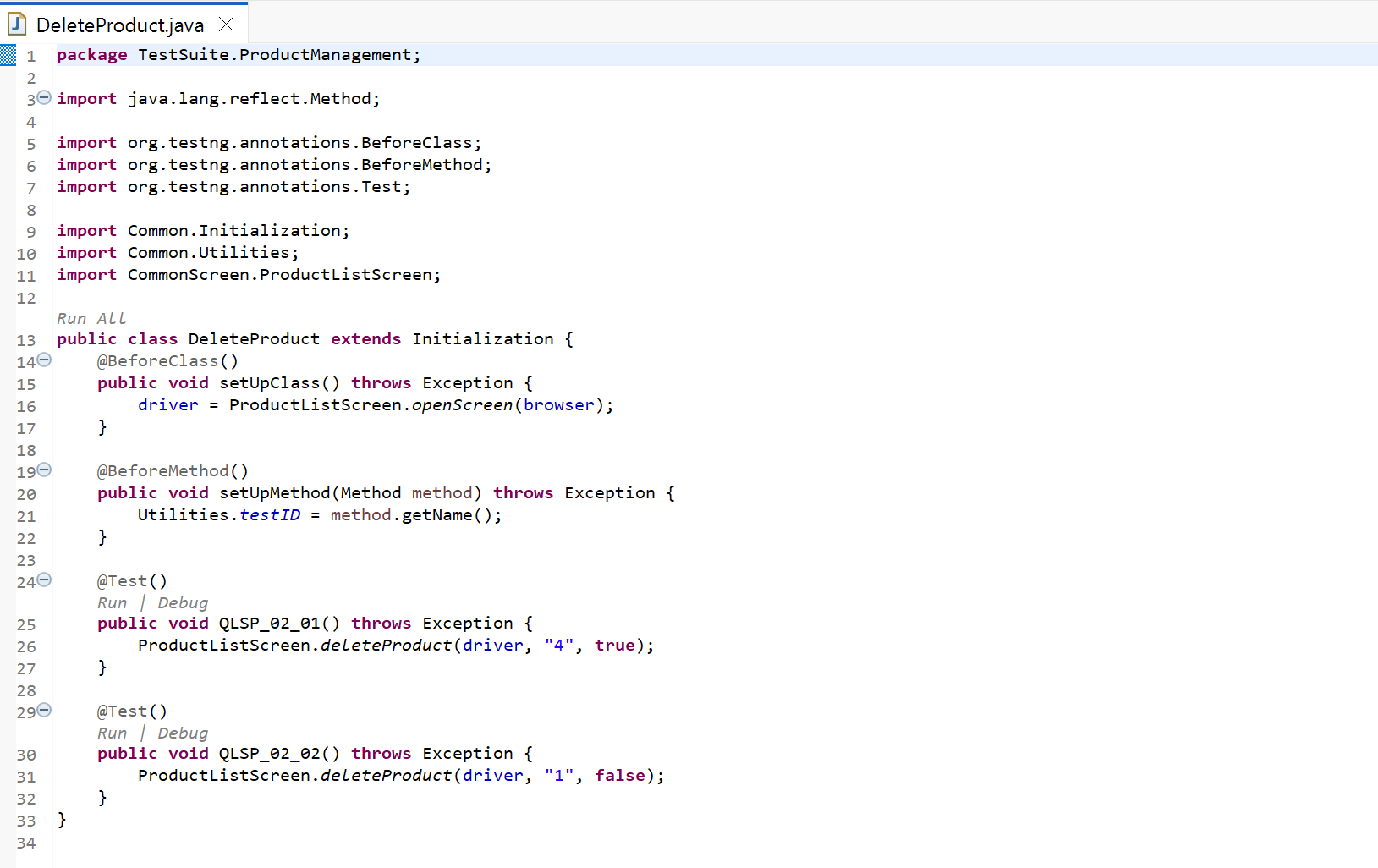
* **Bước 1:** Tạo lớp AddProduct.java để kiểm thử chức năng thêm sản phẩm trong vai trò quản trị viên.
* **Bước 2:** Đăng nhập vào hệ thống với quyền admin để truy cập khu vực quản lý sản phẩm.
* **Bước 3:** Mở giao diện thêm sản phẩm bằng AddProductScreen.openScreen.
* **Bước 4:** Kiểm tra nếu hiển thị nút “Thêm” → mở form thêm sản phẩm bằng Utilities.clickObscuredElement().
* **Bước 5:** Đọc dữ liệu từ file CSV (AddProductData.class) bao gồm các trường: tên sản phẩm, mã sản phẩm, ảnh, mô tả, giá bán, khuyến mãi, số lượng tồn kho, và thông báo mong đợi.
* **Bước 6:** Gọi AddProductScreen.addProduct() để nhập dữ liệu và xác nhận thêm sản phẩm.



Hình 3. 17. Mã nguồn kiểm thử chức năng thêm sản phẩm(admin)

***Kịch bản kiểm thử chức năng xóa sản phẩm với tài khoản admin***

* **Bước 1:** Tạo lớp DeleteProduct.java để kiểm thử chức năng xóa sản phẩm trong vai trò admin.
* **Bước 2:** Đăng nhập với tài khoản admin, sau đó mở danh sách sản phẩm bằng ProductListScreen.openScreen.
* **Bước 3:** Thiết lập mã test bằng Utilities.testID = method.getName().
* **Bước 4:** Gọi ProductListScreen.deleteProduct(driver, "index", confirm) để thực hiện hành động:
* "index": chỉ số sản phẩm trong danh sách.
* confirm = true: xác nhận xóa.
* confirm = false: hủy thao tác xóa.



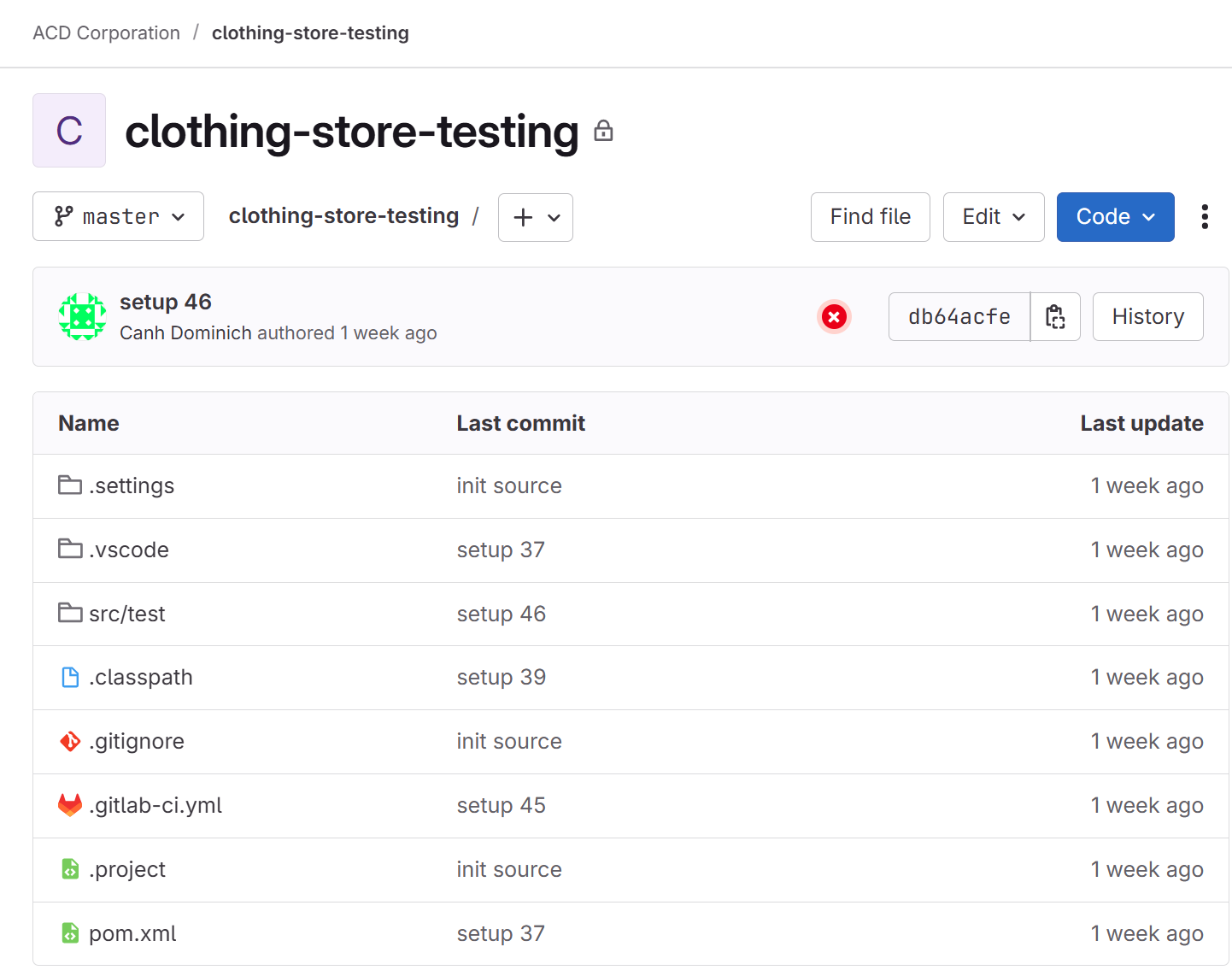
Hình 3. 18. Mã nguồn kiểm thử chức năng xóa sản phẩm(admin)

## 3.4. Tích hợp CI/CD với Gitlab CI

Để tích hợp kiểm thử tự động ứng dụng web bằng GitLab thực hiện như

sau:

* Tạo tệp cấu hình GitLab CI/CD (.gitlab-ci.yml)
* Commit và Push mã nguồn: Đảm bảo rằng commit và push file .gitlab-ci.yml lên GitLab.



Hình 3. 19. Tổ chức mã nguồn kiểm thử tự động trên GitLab

Pipeline sẽ tự động chạy khi có sự thay đổi trên nhánh master

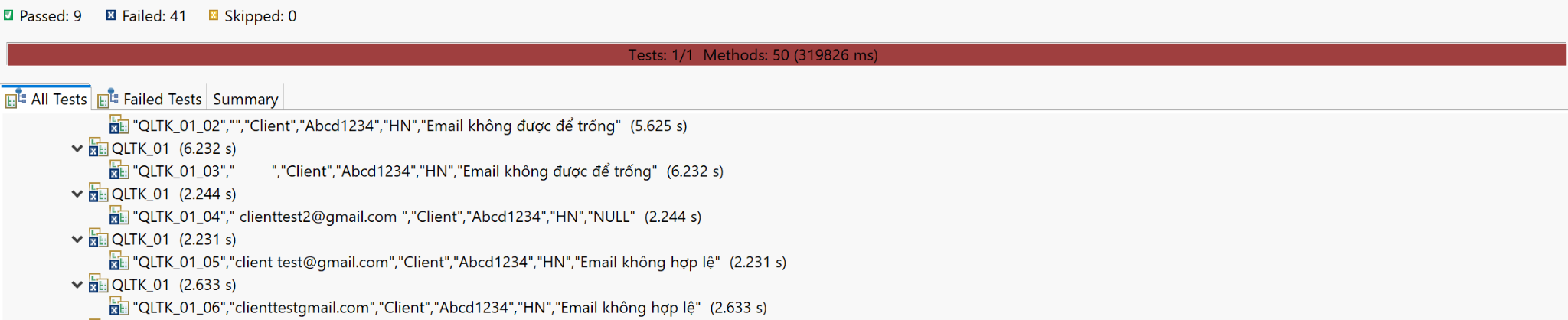
* Pipeline là một quy trình tự động trong GitLab, giúp kiểm tra và triển khai code mỗi khi commit rồi push code lên.
* Job là một bước nhỏ trong pipeline. Nhiều job sẽ tạo thành một pipeline.
* Còn artifacts sẽ là nơi để lưu kết quả chạy test.



Hình 3. 20. Kết quả artifact được tạo ra từ job kiểm thử trên GitLab

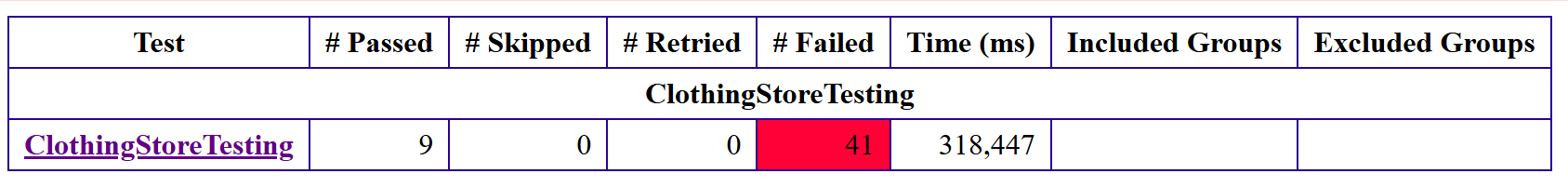
## 3.5. Báo cáo kiểm thử

***Kết quả thực thi kiểm thử cho chức năng Quản lý tài khoản bao gồm đăng ký, đăng nhập, đăng xuất:***



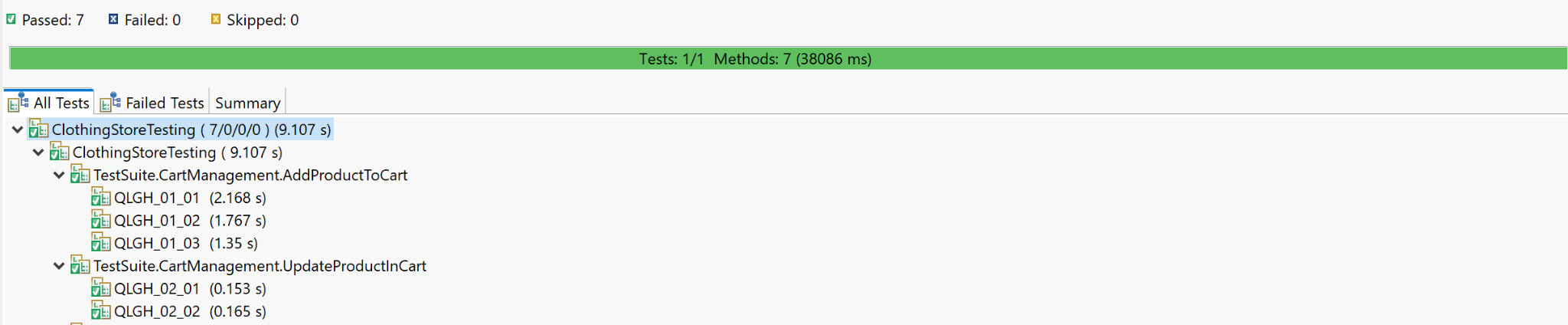
Hình 3. 21. Kết quả thực thi của chức năng Quản lý tài khoản

Nhìn vào kết quả ta thấy, sau khi test, tỷ lệ pass đạt 18% (9/50 test case) và fail đạt 82% (41/50 test case). Tổng thời gian thực hiện toàn bộ suite là 5 phút 18 giây, đảm bảo hiệu quả về mặt thời gian khi áp dụng kiểm thử tự động thay cho kiểm thử thủ công.



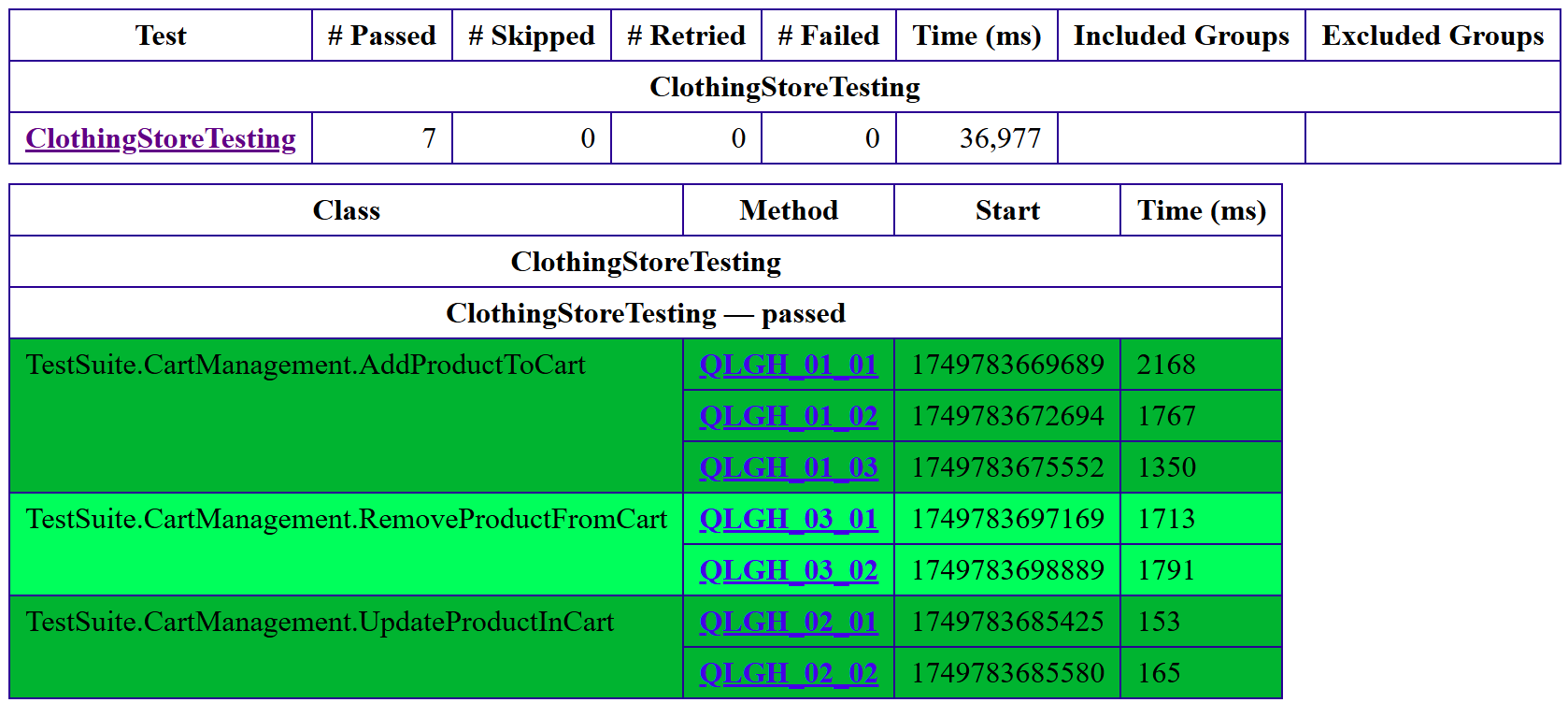
Hình 3. 22. Report của chức năng Quản lý tài khoản

***Kết quả thực thi kiểm thử cho chức năng Quản lý giỏ hàng bao gồm thêm, sửa, xóa sản phẩm trong giỏ hàng:***



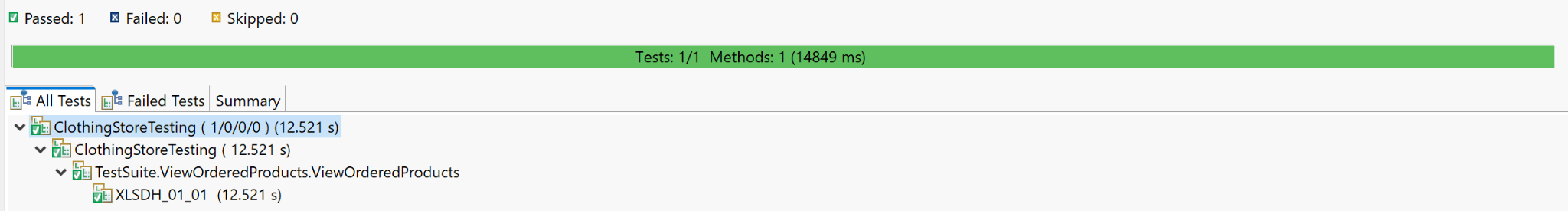
Hình 3. 23. Kết quả thực thi của chức năng Quản lý giỏ hàng

Nhìn vào kết quả ta thấy, sau khi test tỷ lệ pass đạt 100% (7 test case). Tổng thời gian thực hiện toàn bộ suite là 36.977 giây, đảm bảo hiệu quả về mặt thời gian khi áp dụng kiểm thử tự động thay cho kiểm thử thủ công.



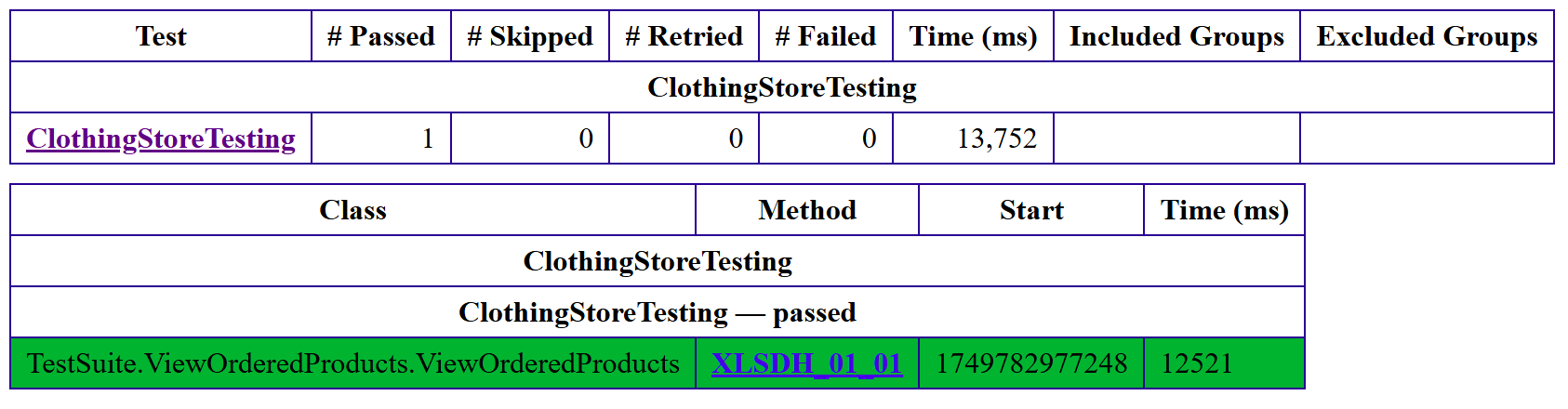
Hình 3. 24. Report của chức năng Quản lý giỏ hàng

***Kết quả thực thi kiểm thử cho chức năng xem đơn hàng đã đặt trong hệ thống:***



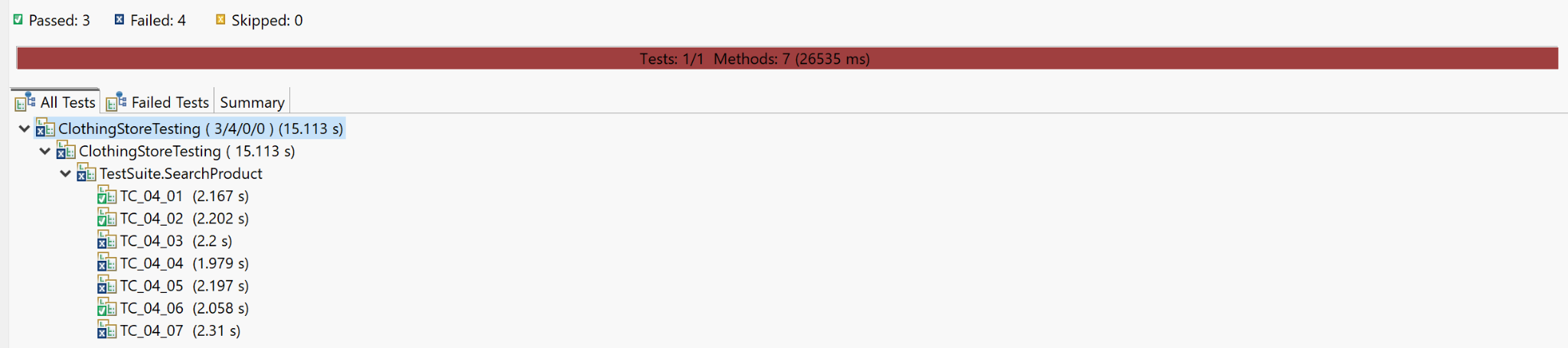
Hình 3. 25. Kết quả thực thi cho chức năng Xem đơn hàng đã đặt

Nhìn vào kết quả ta thấy, sau khi test tỷ lệ pass đạt 100% với 1 test case được chạy (XLSDH\_01\_01) thuộc class ViewOrderedProducts. Tổng thời gian thực hiện toàn bộ suite là 12521 giây cho thấy quá trình kiểm thử tự động hoạt động ổn định, nhanh chóng và chính xác.



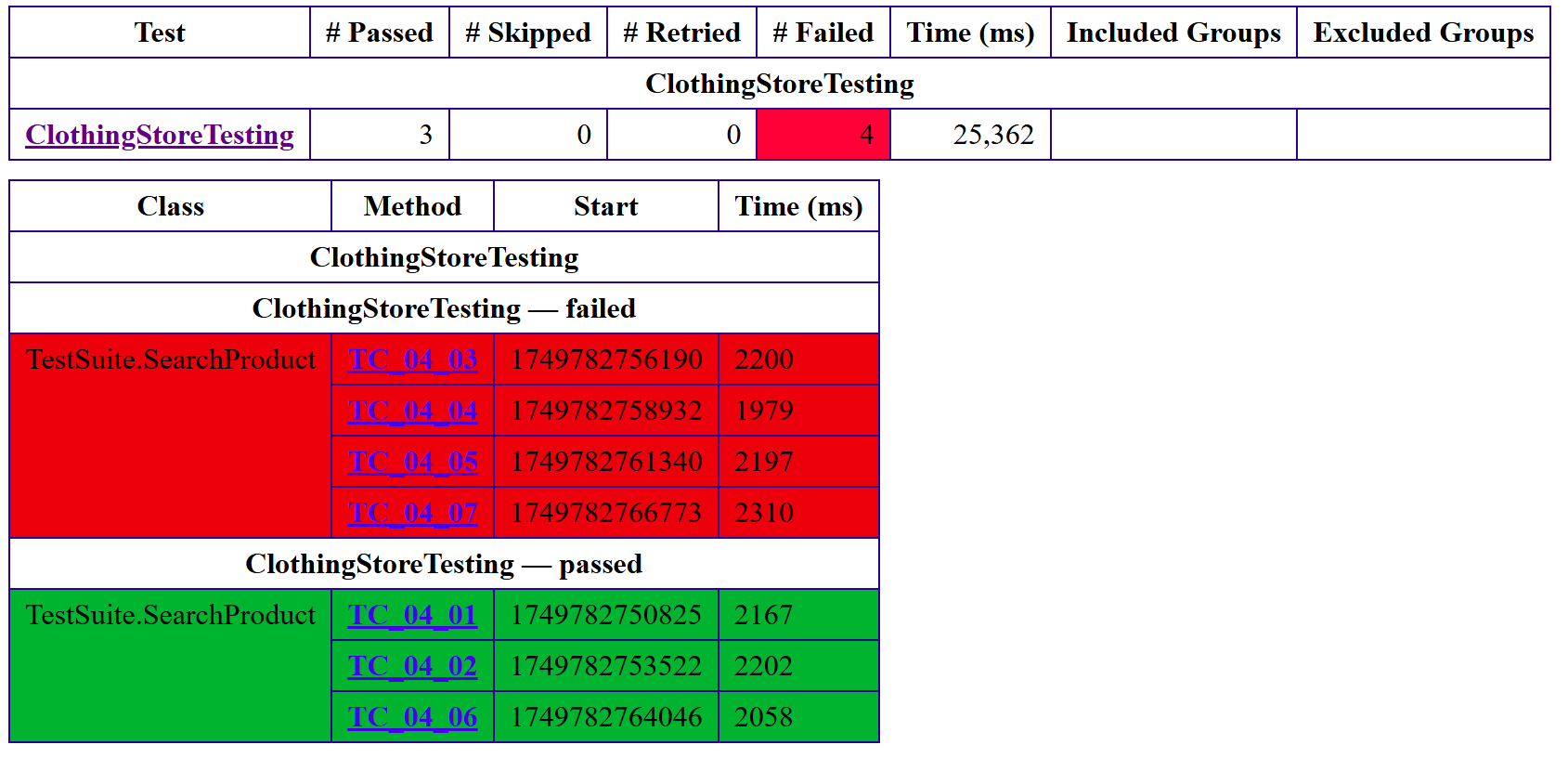
Hình 3. 26. Report của chức năng Xem đơn hàng đã đặt

***Kết quả thực thi kiểm thử cho chức năng tìm kiếm sản phẩm:***



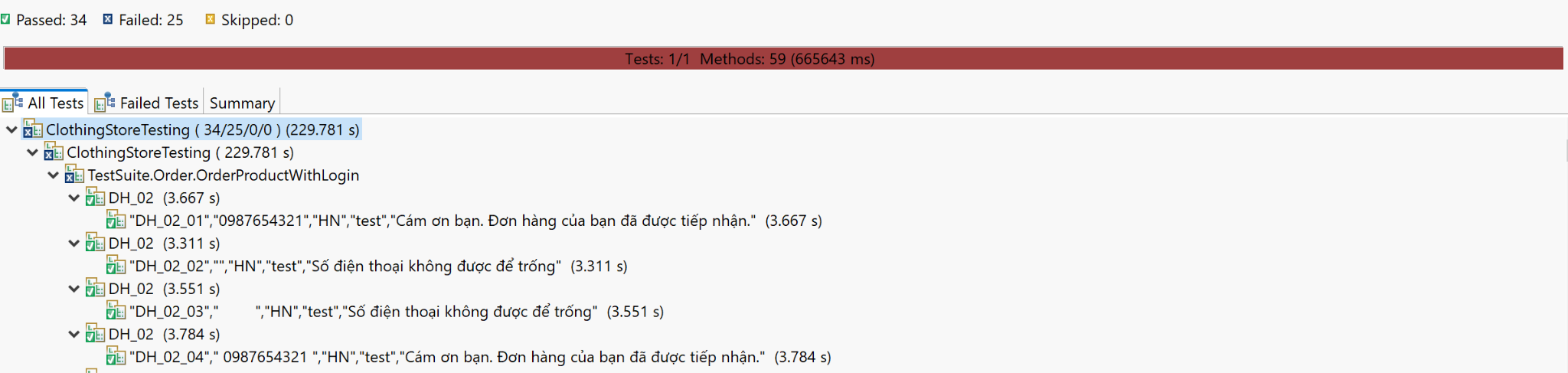
Hình 3. 27. Kết quả thực thi cho chức năng tìm kiếm sản phẩm

Nhìn vào kết quả ta thấy, sau khi test, tỷ lệ pass đạt khoảng 43% (3/7 test case) và fail đạt 57% (4/7 test case). Tổng thời gian thực hiện toàn bộ suite là 25,362 giây, cho thấy kiểm thử tự động giúp phát hiện lỗi nhanh chóng và tiết kiệm thời gian hơn so với kiểm thử thủ công.



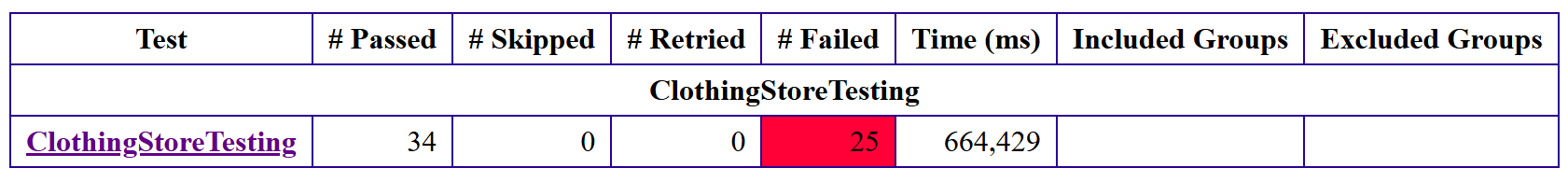
Hình 3. 28. Report của chức năng tìm kiếm sản phẩm

***Kết quả thực thi kiểm thử cho chức năng đặt hàng:***



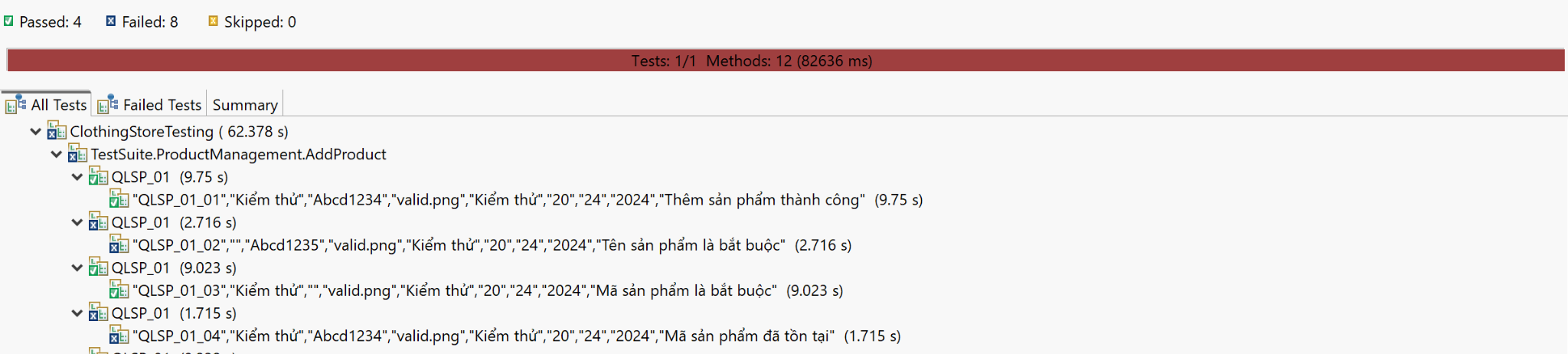
Hình 3. 29. Kết quả thực thi cho chức năng đặt hàng

Nhìn vào kết quả ta thấy, sau khi test, tỷ lệ pass đạt khoảng 58% (34/59 test case) và fail đạt 42% (25/59 test case). Tổng thời gian thực hiện toàn bộ suite là 11 phút 10 giây, cho thấy việc áp dụng kiểm thử tự động giúp tiết kiệm thời gian và phát hiện lỗi hiệu quả hơn so với kiểm thử thủ công.



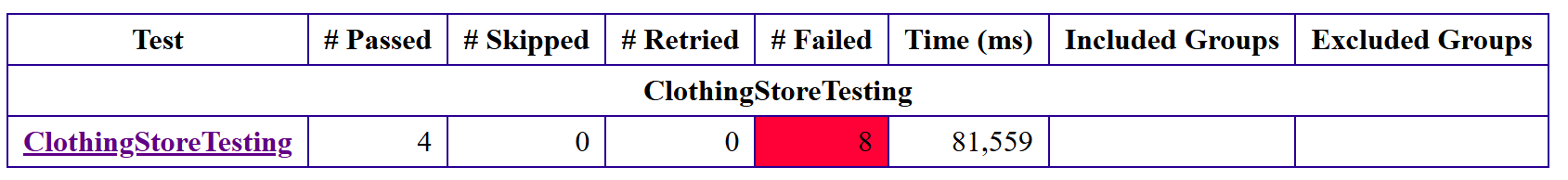
Hình 3. 30. Report của chức năng đặt hàng

***Kết quả thực thi kiểm thử cho chức năng quản lý sản phẩm bằng tài khoản admin bao gồm thêm và xóa sản phẩm:***



Hình 3. 31. Kết quả thực thi cho chức năng quản lý sản phẩm

Nhìn vào kết quả ta thấy, sau khi test, tỷ lệ pass đạt 33% (4/12 test case) và fail đạt 67% (8/12 test case). Tổng thời gian thực hiện toàn bộ suite là 1 phút 24 giây, cho thấy kiểm thử tự động giúp phát hiện chính xác các lỗi xảy ra trong quá trình thêm và xóa sản phẩm khi thao tác bằng tài khoản admin.



Hình 3. 32. Report của chức năng quản lý sản phẩm

## 3.6. Tổng kết chương 3

Trong Chương 3 em đã trình bày quy trình triển khai kiểm thử tự động cho website bán quần áo bằng Selenium WebDriver. Đồng thời, quy trình CI/CD được tích hợp bằng GitLab CI nhằm tự động hóa kiểm thử. Kết quả thực thi cho thấy kiểm thử tự động giúp tiết kiệm thời gian, phát hiện lỗi hiệu quả và đảm bảo chất lượng hệ thống.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

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 thì kết quả mà em thu được cụ thể như sau:

**Kết quả đạt được:**

Hoàn thành mục tiêu đồ án “Kiểm thử tự động với Selenium tích hợp CI/CD bằng GitLab CI”.

Xây dựng được các tài liệu test để phục vụ cho dự án bao gồm test case, test script, report.

Áp dụng các framework đã xây dựng được vào kiểm thử cho website bán quần áo.

Framework có cấu trúc rõ ràng, dễ sử dụng và bảo trì, đồng thời cho phép mở rộng để kiểm thử các chức năng khác trong tương lai.

Tích hợp kiểm thử tự động với Selenium vào quy trình CI/CD thông qua GitLab CI đã giúp tự động hóa toàn bộ quá trình kiểm thử phần mềm, từ khâu viết mã cho đến triển khai.

**Hạn chế:**

Chưa tích hợp các kỹ thuật AI để tối ưu kiểm thử như: tự động phát hiện lỗi, tạo test case, hoặc phân tích giao diện người dùng trong quá trình kiểm thử bằng Selenium.

Kịch bản test còn hạn chế, thiếu kinh nghiệm trong việc xây dựng kịch bản test.

Chưa tối ưu kiểm thử tự động trong việc phát hiện lỗi phần mềm.

Chưa mở rộng kiểm thử trên thiết bị di động.

**Hướng phát triển đề tài:**

Trong tương lai em sẽ tiếp tục nghiên cứu sửa đổi, mở rộng framework kiểm thử tự động hiện tại.

Phát triển framework để kiểm thử tự động cho các ứng dụng khác như ứng dụng di động, ứng dụng trên windows.

Kế thừa mô hình của hệ thống kiểm thử tự động hiện tại và ứng dụng thêm nhiều công cụ khác vào kiểm thử.

# TÀI LIỆU THAM KHẢO

[1] Hoàng Quang Huy, Phùng Đức Hòa, Nguyễn Đức Lưu, Trịnh Bá Quý, (2019), “Giáo trình kiểm thử phần mềm”, Nhà xuất bản Thống kê.

[2] Navneesh Garg. *Test Automation Using Selenium WebDriver with Java: Step by Step Guide*. BPB Publications, 2016.

[3] https://locker.io/vi/blog/gitlab-ci-cd

[4] https://docs.gitlab.com/ci/

[5] <https://viblo.asia/p/tich-hop-cicd-voi-gitlab-ORNZqB0el0n#_6-seeing-status-of-your-pipeline--jobs-5>

[6] <https://haucanit.com/kiem-thu-tu-dong-auto-testing/>

# PHỤ LỤC

Bảng 4. 1. Bảng testcase chức năng quản lý tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| **Đăng ký** |  |  |  |
| QLTK\_01\_01 | Kiểm tra đăng ký thành công | 1. Nhập dữ liệu hợp lệ vào các trường 2. Nhấn nút Đăng ký | 2.Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| QLTK\_01\_02 | Kiểm tra đăng ký với Email trống | 1. Không nhập dữ liệu cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không được để trống" |
| QLTK\_01\_03 | Kiểm tra đăng ký với Email toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không được để trống" |
| QLTK\_01\_04 | Kiểm tra đăng ký với Email có chứa ký tự khoảng trắng ở đầu và cuối | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở đầu và cuối cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| QLTK\_01\_05 | Kiểm tra đăng ký với Email có chứa ký tự khoảng trắng ở giữa | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở giữa cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_01\_06 | Kiểm tra đăng ký với Email thiếu ký tự '@' | 1. Nhập dữ liệu thiếu ký tự '@' cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_01\_07 | Kiểm tra đăng ký với Email thiếu ký tự '.' trước tên miền | 1. Nhập dữ liệu thiếu ký tự '.' trước tên miền cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_01\_08 | Kiểm tra đăng ký với Email thiếu tên miền | 1. Nhập dữ liệu thiếu tên miền cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_01\_09 | Kiểm tra đăng ký với Email chứa ký tự đặc biệt | 1. Nhập dữ liệu chứa ký tự đặc biệt cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_01\_10 | Kiểm tra đăng ký với Email là html | 1. Nhập dữ liệu html cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_01\_11 | Kiểm tra đăng ký với Email là sql | 1. Nhập dữ liệu sql cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_01\_12 | Kiểm tra đăng ký với Email là javascript | 1. Nhập dữ liệu javascript cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_01\_13 | Kiểm tra đăng ký với Email đã đăng ký | 1. Nhập dữ liệu đã đăng ký cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email đã đăng kí." |
| QLTK\_01\_14 | Kiểm tra đăng ký với Email đã đăng ký nhưng viết hoa | 1. Nhập dữ liệu đã đăng ký và viết hoa cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Email đã đăng kí." |
| QLTK\_01\_15 | Kiểm tra đăng ký với Tên đăng nhập trống | 1. Không nhập dữ liệu cho trường Tên đăng nhập 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Tên đăng nhập không được để trống" |
| QLTK\_01\_16 | Kiểm tra đăng ký với Tên đăng nhập toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Tên đăng nhập 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Tên đăng nhập không được để trống" |
| QLTK\_01\_17 | Kiểm tra đăng ký với Tên đăng nhập có chứa ký tự khoảng trắng | 1. Nhập dữ liệu có chứa ký tự khoảng trắng cho trường Tên đăng nhập 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| QLTK\_01\_18 | Kiểm tra đăng ký với Tên đăng nhập có chứa ký tự đặc biệt | 1. Nhập dữ liệu có chứa ký tự đặc biệt cho trường Tên đăng nhập 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Tên đăng nhập không hợp lệ" |
| QLTK\_01\_19 | Kiểm tra đăng ký với Tên đăng nhập có chứa ký tự số | 1. Nhập dữ liệu có chứa ký tự số cho trường Tên đăng nhập 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Tên đăng nhập không hợp lệ" |
| QLTK\_01\_20 | Kiểm tra đăng ký với Mật khẩu trống | 1. Không nhập dữ liệu cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Mật khẩu không được để trống" |
| QLTK\_01\_21 | Kiểm tra đăng ký với Mật khẩu toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| QLTK\_01\_22 | Kiểm tra đăng ký với Mật khẩu < 8 ký tự | 1. Nhập dữ liệu có 7 ký tự cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Mật khẩu tối thiểu 8 ký tự" |
| QLTK\_01\_23 | Kiểm tra đăng ký với Mật khẩu = 8 ký tự | 1. Nhập dữ liệu có 8 ký tự cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| QLTK\_01\_24 | Kiểm tra đăng ký với Mật khẩu = 16 ký tự | 1. Nhập dữ liệu có 16 ký tự cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| QLTK\_01\_25 | Kiểm tra đăng ký với Mật khẩu > 16 ký tự | 1. Nhập dữ liệu có 17 ký tự cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Hiển thị thông báo lỗi "Mật khẩu tối đa 16 ký tự" |
| QLTK\_01\_26 | Kiểm tra đăng ký với Địa chỉ trống | 1. Không nhập dữ liệu cho trường Địa chỉ 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| QLTK\_01\_27 | Kiểm tra đăng ký với Địa chỉ toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Địa chỉ 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| QLTK\_01\_28 | Kiểm tra đăng ký với Địa chỉ có chứa ký tự khoảng trắng | 1. Nhập dữ liệu có chứa ký tự khoảng trắng cho trường Địa chỉ 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đăng ký | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng ký |
| **Đăng nhập** |  |  |  |
| QLTK\_02\_01 | Kiểm tra đăng nhập thành công | 1. Nhập dữ liệu hợp lệ vào các trường 2. Nhấn nút Đăng nhập | 2. Chuyển đến màn hình Trang chủ với tài khoản đã đăng nhập |
| QLTK\_02\_02 | Kiểm tra đăng nhập thành công nhưng email viết hoa | 1. Nhập dữ liệu đã Đăng nhập và viết hoa cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng nhập |
| QLTK\_02\_03 | Kiểm tra đăng nhập với Email trống | 1. Không nhập dữ liệu cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không được để trống" |
| QLTK\_02\_04 | Kiểm tra đăng nhập với Email toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không được để trống" |
| QLTK\_02\_05 | Kiểm tra đăng nhập với Email có chứa ký tự khoảng trắng ở đầu và cuối | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở đầu và cuối cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng nhập |
| QLTK\_02\_06 | Kiểm tra đăng nhập với Email có chứa ký tự khoảng trắng ở giữa | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở giữa cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_02\_07 | Kiểm tra đăng nhập với Email thiếu ký tự '@' | 1. Nhập dữ liệu thiếu ký tự '@' cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_02\_08 | Kiểm tra đăng nhập với Email thiếu ký tự '.' trước tên miền | 1. Nhập dữ liệu thiếu ký tự '.' trước tên miền cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_02\_09 | Kiểm tra đăng nhập với Email thiếu tên miền | 1. Nhập dữ liệu thiếu tên miền cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_02\_10 | Kiểm tra đăng nhập với Email chứa ký tự đặc biệt | 1. Nhập dữ liệu chứa ký tự đặc biệt cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_02\_11 | Kiểm tra đăng nhập với Email là html | 1. Nhập dữ liệu html cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_02\_12 | Kiểm tra đăng nhập với Email là sql | 1. Nhập dữ liệu sql cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_02\_13 | Kiểm tra đăng nhập với Email là javascript | 1. Nhập dữ liệu javascript cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| QLTK\_02\_14 | Kiểm tra đăng nhập với Email chưa được Đăng nhập | 1. Nhập dữ liệu đã đăng nhập cho trường Email 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Sai tài khoản hoặc mật khẩu!" |
| QLTK\_02\_15 | Kiểm tra đăng nhập với Mật khẩu trống | 1. Không nhập dữ liệu cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Mật khẩu không được để trống" |
| QLTK\_02\_16 | Kiểm tra đăng nhập với Mật khẩu < 8 ký tự | 1. Nhập dữ liệu có 7 ký tự cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Mật khẩu tối thiểu 8 ký tự" |
| QLTK\_02\_17 | Kiểm tra đăng nhập với Mật khẩu = 8 ký tự | 1. Nhập dữ liệu có 8 ký tự cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng nhập |
| QLTK\_02\_18 | Kiểm tra đăng nhập với Mật khẩu = 16 ký tự | 1. Nhập dữ liệu có 16 ký tự cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Chuyển đến màn hình Trang chủ với tài khoản đã đăng nhập |
| QLTK\_02\_19 | Kiểm tra đăng nhập với Mật khẩu > 16 ký tự | 1. Nhập dữ liệu có 17 ký tự cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Mật khẩu tối đa 16 ký tự" |
| QLTK\_02\_20 | Kiểm tra đăng nhập với Mật khẩu không chính xác | 1. Nhập dữ liệu không chính xác cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Sai tài khoản hoặc mật khẩu!" |
| QLTK\_02\_21 | Kiểm tra đăng nhập với Mật khẩu chính xác nhưng viết hoa | 1. Nhập dữ liệu chính xác nhưng viết hoa cho trường Mật khẩu 2. Nhập dữ liệu hợp lệ cho trường khác 3. Nhấn nút Đăng nhập | 3. Hiển thị thông báo lỗi "Sai tài khoản hoặc mật khẩu!" |
| **Đăng xuất** |  |  |  |
| QLTK\_03\_01 | Kiểm tra đăng xuất thành công | 1. Nhấn chọn Tên đăng nhập 2. Nhấn chọn Đăng xuất | 2. Đăng xuất tài khoản của khách hàng ra khỏi hệ thống |

Bảng 4. 2. Bảng testcase chức năng quản lý giỏ hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| **Thêm sản phẩm vào giỏ hàng** |  |  |  |
| QLGH\_01\_01 | Kiểm tra thêm sản phẩm vào giỏ hàng trống | 1. Nhấn nút “Thêm vào giỏ hàng” của sản phẩm A | 1. - Sản phẩm A được thêm vào giỏ hàng thành công với số lượng là 1 - Số lượng sản phẩm trong giỏ hàng là 1 |
| QLGH\_01\_02 | Kiểm tra thêm sản phẩm khác vào giỏ hàng khi có sản phẩm trong giỏ hàng | 2. Nhấn nút “Thêm vào giỏ hàng” của sản phẩm B | 2. - Sản phẩm B được thêm vào giỏ hàng thành công với số lượng là 1 - Số lượng sản phẩm trong giỏ hàng là 2 |
| QLGH\_01\_03 | Kiểm tra thêm sản phẩm vào giỏ hàng khi sản phẩm đó có sản phẩm trong giỏ hàng | 3. Nhấn nút “Thêm vào giỏ hàng” của sản phẩm B | 3. - Sản phẩm B được thêm vào giỏ hàng thành công với số lượng là 2 - Số lượng sản phẩm trong giỏ hàng là 3 |
| **Cập nhật số lượng sản phẩm trong giỏ hàng** |  |  |  |
| QLGH\_02\_01 | Kiểm tra nhấn nút "+" | 1. Nhấn nút "+" 2. Nhấn nút "Thêm vào giỏ hàng" | 2. Cập nhật giỏ hàng như số lượng sản phẩm, thành tiền, tạm tính và tổng tiền |
| QLGH\_02\_02 | Kiểm tra nhấn nút "-" | 1. Nhấn nút "-" 2. Nhấn nút "Thêm vào giỏ hàng" | 2. Cập nhật giỏ hàng như số lượng sản phẩm, thành tiền, tạm tính và tổng tiền |
| **Xóa sản phẩm khỏi giỏ hàng** |  |  |  |
| QLGH\_03\_01 | Kiểm tra xóa sản phẩm khi giỏ hàng có nhiều hơn 1 sản phẩm | 1. Nhấn biểu tượng "Xóa" | 1. Cập nhật giỏ hàng như số lượng sản phẩm, thành tiền, tạm tính và tổng tiền |
| QLGH\_03\_02 | Kiểm tra xóa sản phẩm khi giỏ hàng có 1 sản phẩm | 1. Nhấn biểu tượng "Xóa" | 1. Hiển thị thông báo "Giỏ hàng của bạn đang trống. Mời bạn tiếp tục mua hàng!" |

Bảng 4. 3. Bảng testcase chức năng đặt hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| **Đặt hàng khi chưa đăng nhập** |  |  |  |
| DH\_01\_01 | Kiểm tra đặt mua thành công | 1. Nhập dữ liệu hợp lệ cho các trường 2. Nhấn nút Đặt mua | 2. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_02 | Kiểm tra đặt mua với Tên khách hàng trống | 1. Không nhập dữ liệu cho trường Tên khách hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Tên khách hàng không được để trống" |
| DH\_01\_03 | Kiểm tra đặt mua với Tên khách hàng toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Tên khách hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Tên khách hàng không được để trống" |
| DH\_01\_04 | Kiểm tra đặt mua với Tên khách hàng có chứa ký tự khoảng trắng | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở đầu và cuối cho trường Tên khách hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_05 | Kiểm tra đặt mua với Tên khách hàng có chứa ký tự đặc biệt | 1. Nhập dữ liệu có chứa ký tự đặc biệt cho trường Tên khách hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Tên khách hàng không hợp lệ" |
| DH\_01\_06 | Kiểm tra đặt mua với Tên khách hàng có chứa ký tự số | 1. Nhập dữ liệu có chứa ký tự số cho trường Tên khách hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Tên khách hàng không hợp lệ" |
| DH\_01\_07 | Kiểm tra đặt mua với Số điện thoại trống | 1. Không nhập dữ liệu cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không được để trống” |
| DH\_01\_08 | Kiểm tra đặt mua với Số điện thoại toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không được để trống” |
| DH\_01\_09 | Kiểm tra đặt mua với Số điện thoại có chứa ký tự khoảng trắng ở đầu và cuối | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở đầu và cuối cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận.” |
| DH\_01\_10 | Kiểm tra đặt mua với Số điện thoại có chứa ký tự khoảng trắng ở giữa | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở giữa cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_01\_11 | Kiểm tra đặt mua với Số điện thoại có chứa ký tự chữ | 1. Nhập dữ liệu có chứa ký tự chữ cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_01\_12 | Kiểm tra đặt mua với Số điện thoại có chứa ký tự đặc biệt | 1. Nhập dữ liệu có chứa ký tự đặc biệt cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_01\_13 | Kiểm tra đặt mua với Số điện thoại toàn số 0 | 1. Nhập toàn số 0 cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_01\_14 | Kiểm tra đặt mua với Số điện thoại < 10 ký tự số | 1. Nhập dữ liệu có 9 ký tự số cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_01\_15 | Kiểm tra đặt mua với Số điện thoại = 10 ký tự số | 1. Nhập dữ liệu có 10 ký tự số cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận.” |
| DH\_01\_16 | Kiểm tra đặt mua với Số điện thoại = 10 ký tự nhưng không phải số điện thoại | 1. Nhập dữ liệu có 10 ký tự số nhưng không phải số điện thoại cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_01\_17 | Kiểm tra đặt mua với Số điện thoại > 10 ký tự số | 1. Nhập dữ liệu có 11 ký tự số cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_01\_18 | Kiểm tra đặt mua với Email trống | 1. Không nhập dữ liệu cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_19 | Kiểm tra đặt mua với Email toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_20 | Kiểm tra đặt mua với Email có chứa ký tự khoảng trắng ở đầu và cuối | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở đầu và cuối cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_21 | Kiểm tra đặt mua với Email có chứa ký tự khoảng trắng ở giữa | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở giữa cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| DH\_01\_22 | Kiểm tra đặt mua với Email thiếu ký tự '@' | 1. Nhập dữ liệu thiếu ký tự '@' cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| DH\_01\_23 | Kiểm tra đặt mua với Email thiếu ký tự '.' trước tên miền | 1. Nhập dữ liệu thiếu ký tự '.' trước tên miền cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| DH\_01\_24 | Kiểm tra đặt mua với Email thiếu tên miền | 1. Nhập dữ liệu thiếu tên miền cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| DH\_01\_25 | Kiểm tra đặt mua với Email chứa ký tự đặc biệt | 1. Nhập dữ liệu chứa ký tự đặc biệt cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Email không hợp lệ" |
| DH\_01\_26 | Kiểm tra đặt mua với Email đã đặt mua | 1. Nhập dữ liệu đã đặt mua cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_27 | Kiểm tra đặt mua với Email đã đặt mua nhưng viết hoa | 1. Nhập dữ liệu đã đặt mua và viết hoa cho trường Email 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_28 | Kiểm tra đặt mua với Địa chỉ giao hàng trống | 1. Không nhập dữ liệu cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Địa chỉ giao hàng không được để trống" |
| DH\_01\_29 | Kiểm tra đặt mua với Địa chỉ giao hàng toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Địa chỉ giao hàng không được để trống" |
| DH\_01\_30 | Kiểm tra đặt mua với Địa chỉ giao hàng có chứa ký tự khoảng trắng | 1. Nhập dữ liệu có chứa ký tự khoảng trắng cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_31 | Kiểm tra đặt mua với Địa chỉ giao hàng trên nhiều dòng | 1. Nhập dữ liệu trên nhiều dòng cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_32 | Kiểm tra đặt mua với Địa chỉ giao hàng chứa ký tự xuống dòng | 1. Nhập dữ liệu chứa ký tự xuống dòng cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_33 | Kiểm tra đặt mua với Ghi chú cho đơn hàng trống | 1. Không nhập dữ liệu cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_34 | Kiểm tra đặt mua với Ghi chú cho đơn hàng toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_35 | Kiểm tra đặt mua với Ghi chú cho đơn hàng có chứa ký tự khoảng trắng | 1. Nhập dữ liệu có chứa ký tự khoảng trắng cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_36 | Kiểm tra đặt mua với Ghi chú cho đơn hàng trên nhiều dòng | 1. Nhập dữ liệu trên nhiều dòng cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_01\_37 | Kiểm tra đặt mua với Ghi chú cho đơn hàng chứa ký tự xuống dòng | 1. Nhập dữ liệu chứa ký tự xuống dòng cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| **Đặt hàng khi đã đăng nhập** |  |  |  |
| DH\_02\_01 | Kiểm tra đặt mua thành công | 1. Nhập dữ liệu hợp lệ cho các trường 2. Nhấn nút Đặt mua | 2. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_02\_02 | Kiểm tra đặt mua với Số điện thoại trống | 1. Không nhập dữ liệu cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không được để trống” |
| DH\_02\_03 | Kiểm tra đặt mua với Số điện thoại toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không được để trống” |
| DH\_02\_04 | Kiểm tra đặt mua với Số điện thoại có chứa ký tự khoảng trắng ở đầu và cuối | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở đầu và cuối cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận.” |
| DH\_02\_05 | Kiểm tra đặt mua với Số điện thoại có chứa ký tự khoảng trắng ở giữa | 1. Nhập dữ liệu có chứa ký tự khoảng trắng ở giữa cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_02\_06 | Kiểm tra đặt mua với Số điện thoại có chứa ký tự chữ | 1. Nhập dữ liệu có chứa ký tự chữ cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_02\_07 | Kiểm tra đặt mua với Số điện thoại có chứa ký tự đặc biệt | 1. Nhập dữ liệu có chứa ký tự đặc biệt cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_02\_08 | Kiểm tra đặt mua với Số điện thoại toàn số 0 | 1. Nhập toàn số 0 cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_02\_09 | Kiểm tra đặt mua với Số điện thoại < 10 ký tự số | 1. Nhập dữ liệu có 9 ký tự số cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_02\_10 | Kiểm tra đặt mua với Số điện thoại = 10 ký tự số | 1. Nhập dữ liệu có 10 ký tự số cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận.” |
| DH\_02\_11 | Kiểm tra đặt mua với Số điện thoại = 10 ký tự nhưng không phải số điện thoại | 1. Nhập dữ liệu có 10 ký tự số nhưng không phải số điện thoại cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_02\_12 | Kiểm tra đặt mua với Số điện thoại > 10 ký tự số | 1. Nhập dữ liệu có 11 ký tự số cho trường Số điện thoại 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo “Số điện thoại không hợp lệ” |
| DH\_02\_13 | Kiểm tra đặt mua với Địa chỉ giao hàng trống | 1. Không nhập dữ liệu cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Địa chỉ giao hàng không được để trống" |
| DH\_02\_14 | Kiểm tra đặt mua với Địa chỉ giao hàng toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo lỗi "Địa chỉ giao hàng không được để trống" |
| DH\_02\_15 | Kiểm tra đặt mua với Địa chỉ giao hàng có chứa ký tự khoảng trắng | 1. Nhập dữ liệu có chứa ký tự khoảng trắng cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_02\_16 | Kiểm tra đặt mua với Địa chỉ giao hàng trên nhiều dòng | 1. Nhập dữ liệu trên nhiều dòng cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_02\_17 | Kiểm tra đặt mua với Địa chỉ giao hàng chứa ký tự xuống dòng | 1. Nhập dữ liệu chứa ký tự xuống dòng cho trường Địa chỉ giao hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_02\_18 | Kiểm tra đặt mua với Ghi chú cho đơn hàng trống | 1. Không nhập dữ liệu cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_02\_19 | Kiểm tra đặt mua với Ghi chú cho đơn hàng toàn ký tự khoảng trắng | 1. Nhập toàn ký tự khoảng trắng cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_02\_20 | Kiểm tra đặt mua với Ghi chú cho đơn hàng có chứa ký tự khoảng trắng | 1. Nhập dữ liệu có chứa ký tự khoảng trắng cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_02\_21 | Kiểm tra đặt mua với Ghi chú cho đơn hàng trên nhiều dòng | 1. Nhập dữ liệu trên nhiều dòng cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |
| DH\_02\_22 | Kiểm tra đặt mua với Ghi chú cho đơn hàng chứa ký tự xuống dòng | 1. Nhập dữ liệu chứa ký tự xuống dòng cho trường Ghi chú cho đơn hàng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Đặt mua | 3. Hiển thị thông báo "Cám ơn bạn. Đơn hàng của bạn đã được tiếp nhận." |

Bảng 4. 4. Bảng testcase chức năng quản lý sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| **Thêm sản phẩm** |  |  |  |
| QLSP\_01\_01 | Kiểm tra thêm sản phẩm thành công | 1. Nhập dữ liệu hợp lệ vào các trường 2. Nhấn nút Thêm mới | 2. Hiển thị thông báo "Thêm sản phẩm thành công" |
| QLSP\_01\_02 | Kiểm tra thêm sản phẩm với Tên sản phẩm trống | 1. Không nhập dữ liệu cho trường Tên sản phẩm 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Tên sản phẩm là bắt buộc" |
| QLSP\_01\_03 | Kiểm tra thêm sản phẩm với Mã sản phẩm trống | 1. Không nhập dữ liệu cho trường Mã sản phẩm 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Mã sản phẩm là bắt buộc" |
| QLSP\_01\_04 | Kiểm tra thêm sản phẩm với Mã sản phẩm đã tồn tại | 1. Nhập dữ liệu đã tồn tại cho trường Mã sản phẩm 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Mã sản phẩm đã tồn tại" |
| QLSP\_01\_05 | Kiểm tra thêm sản phẩm với Mô tả trống | 1. Không nhập dữ liệu cho trường Mô tả 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Mô tả là bắt buộc" |
| QLSP\_01\_06 | Kiểm tra thêm sản phẩm với Số lượng trống | 1. Không nhập dữ liệu cho trường Số lượng 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Số lượng là bắt buộc" |
| QLSP\_01\_07 | Kiểm tra thêm sản phẩm với Giá gốc trống | 1. Không nhập dữ liệu cho trường Giá gốc 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Giá gốc là bắt buộc" |
| QLSP\_01\_08 | Kiểm tra thêm sản phẩm với Giá bán trống | 1. Không nhập dữ liệu cho trường Giá bán 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Giá bán là bắt buộc" |
| QLSP\_01\_09 | Kiểm tra thêm sản phẩm với Giá bán nhỏ hơn giá gốc | 1. Nhập dữ liệu cho trường Giá bán > Giá gốc 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Giá bán không hợp lệ" |
| QLSP\_01\_10 | Kiểm tra thêm sản phẩm với Hình ảnh trống | 1. Không chọn dữ liệu cho trường Hình ảnh 2. Nhập dữ liệu hợp lệ cho các trường khác 3. Nhấn nút Thêm mới | 3. Hiển thị thông báo lỗi "Hình ảnh là bắt buộc" |
| **Xóa sản phẩm** |  |  |  |
| QLSP\_02\_01 | Kiểm tra xóa sản phẩm thành công | 1. Nhấn nút Xóa tại dòng chứa sản phẩm cần xóa 2. Nhấn nút OK | 2. Tải lại màn hình Danh sách sản phẩm không còn sản phẩm đã chọn |
| QLSP\_02\_02 | Kiểm tra hủy xóa sản phẩm thành công | 1. Nhấn nút Xóa tại dòng chứa sản phẩm cần xóa 2. Nhấn nút Hủy | 2. Tải lại màn hình Danh sách sản phẩm mà vẫn còn sản phẩm đã chọn |

Bảng 4. 5. Bảng testcase chức năng xem đơn hàng đã đặt

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mục tiêu kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** |
| XLSDH\_01\_01 | Kiểm tra xem danh sách đơn hàng thành công | 1. Nhấn chọn Tên đăng nhập 2. Nhấn chọn Đơn hàng của tôi | 2. Chuyển đến màn hình Đơn hàng của tôi |