

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Thị Nụ

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ
(Theo định hướng ứng dụng)

HÀ NỘI - 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Thị Nụ

NGHIÊN CỨU KIỂM THỬ TỰ ĐỘNG PHẦN MỀM VÀ ỨNG
DỤNG TẠI CÔNG TY A1 CONSULTING

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN
MÃ SỐ: 8.48.01.04

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ
(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC
TS. VŨ VĂN THỎA

HÀ NỘI – 2025

MỤC LỤC

MỤC LỤC.....	i
DANH MỤC BẢNG BIỂU.....	iii
MỞ ĐẦU.....	1
CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM VÀ CÁC VẤN ĐỀ LIÊN QUAN.....	2
1.1 Tổng quan về kiểm thử phần mềm.....	2
1.1.1 Khái niệm.....	2
1.1.2 Mục đích của quá trình kiểm thử	2
1.1.3 Các nguyên tắc cơ bản của kiểm thử phần mềm.....	2
1.2 Quy trình, kỹ thuật kiểm thử phần mềm và các vấn đề liên quan.....	3
1.2.3 Mức độ kiểm thử.....	3
1.2. 4 Thiết kế testcase	3
1.3 Tổng quan về kiểm thử phần mềm trong bối cảnh chuyển đổi số.....	3
1.4 Kết luận chương 1	4
CHƯƠNG 2: NGHIÊN CỨU KỸ THUẬT VÀ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG	5
2.1 Tổng quan về kiểm thử tự động.....	5
2.2.1 Khái niệm.....	5
2.1.2 Vai trò.....	5
2.2 Các kỹ thuật kiểm thử tự động.....	5
2.2.1. Kiểm thử hồi quy (Regression Testing).....	5
2.2.2. Kiểm thử API.....	6
2.2.3. Kiểm thử hiệu năng (Performance Testing).....	7
2.3. Các công cụ kiểm thử tự động	7
2.3.1 Selenium	7
2.3.2 Katalon Studio.....	7
2.3.3 Playwright.....	7
2.3.4 Cypress.....	7
2.3.5 Appium	7
2.3.6 Postman.....	7
2.3.7 Apache Jmeter	7
2.4 So sánh và đánh giá các công cụ kiểm thử.....	7
2.5 Lợi ích và thách thức khi áp dụng kiểm thử tự động	8
2.5.1 Lợi ích.....	8
2.5.2 Thách thức	8
2.6 Kết luận chương 2	9
CHƯƠNG 3: ỨNG DỤNG KIỂM THỬ TỰ ĐỘNG PHẦN MỀM TẠI CÔNG TY A1 CONSULTING.....	10

3.1 Giới thiệu về Công ty A1 Consulting.....	10
3.1.1 Lịch sử hình thành và phát triển.....	10
3.1.2. Lĩnh vực hoạt động và vị thế hiện tại.....	10
3.1.3. Thực trạng kiểm thử phần mềm tại A1 Consulting.....	10
3.2 Giới thiệu phần mềm lựa chọn để thực nghiệm kiểm thử.....	10
3.2.1. Giới thiệu phân hệ Sales và Purchase trong Odoo	10
3.2.1.1 <i>Phân hệ Sales (Bán hàng)</i>	10
3.2.1.2 <i>Phân hệ Purchase (Mua hàng)</i>	10
3.2.2 Yêu cầu và công cụ thử nghiệm.....	11
3.3 Triển khai thực nghiệm.....	11
3.3.1 Xây dựng kịch bản kiểm thử thủ công các tính năng ứng dụng kiểm thử tự động.....	12
3.3.2 Xây dựng kịch bản kiểm thử tự động theo các kịch bản thủ công đã xây dựng.	12
3.4 Kết quả kiểm thử.....	13
3.5 Đánh giá kết quả.....	13
3.6 Kết luận chương 3.....	14
KẾT LUẬN	15

DANH MỤC BẢNG BIỂU

Bảng 2-1: So sánh công cụ kiểm thử	8
Bảng 3-1: Công cụ thử nghiệm được sử dụng.....	11
Bảng 3-4: Đánh giá kết quả.....	13

MỞ ĐẦU

Trong bối cảnh chuyển đổi số mạnh mẽ, nhiều tổ chức số hóa quy trình để tối ưu hiệu suất và nắm bắt cơ hội phát triển. Kiểm thử phần mềm trở thành yếu tố then chốt giúp đảm bảo chất lượng và hiệu suất sản phẩm, thông qua kiểm thử thủ công, tự động và các công cụ như Selenium, Katalon. Kiểm thử giúp phát hiện lỗi, đáp ứng nhu cầu người dùng và nâng cao trải nghiệm. Tuy nhiên, quá trình này cũng đối mặt với thách thức từ hệ thống tích hợp phức tạp và tốc độ thay đổi công nghệ, đồng thời mở ra cơ hội cải tiến bằng tự động hóa và phân tích dữ liệu, góp phần vào thành công của chuyển đổi số.

Vì vậy đề tài “Nghiên cứu kiểm thử tự động phần mềm và ứng dụng tại công ty A1 Consulting” được thực hiện với mục đích nghiên cứu về vai trò và tầm quan trọng của kiểm thử phần mềm trong bối cảnh chuyển đổi số. Đề tài sẽ phân tích các phương pháp kiểm thử hiện có, đánh giá hiệu quả của chúng trong việc đảm bảo chất lượng sản phẩm và dịch vụ số, đồng thời xác định những thách thức mà các Doanh nghiệp gặp phải trong quá trình này.

Đối tượng nghiên cứu: Tập trung vào các phương pháp và công cụ kiểm thử phần mềm tự động, so sánh với kiểm thử thủ công.

Phạm vi nghiên cứu: Tập trung vào kiểm thử tự động cho ứng dụng Odoo ERP.

Phương pháp nghiên cứu: Nghiên cứu các tài liệu và sách chuyên ngành liên quan đến kiểm thử phần mềm tự động, các công cụ và phương pháp mới nhất. Xây dựng các kịch bản kiểm thử công, kịch bản kiểm thử tự động cho các chức năng chính của phần mềm, đảm bảo rằng chúng phản ánh đúng yêu cầu của người dùng và tiêu chuẩn chất lượng.

Từ mục tiêu, nhiệm vụ nghiên cứu, đề án sẽ được cấu trúc với ba chương nội dung chính như sau:

Chương 1: Tổng quan về kiểm thử phần mềm và các vấn đề liên quan.

Chương 2: Nghiên cứu kỹ thuật và công cụ kiểm thử tự động.

Chương 3: Ứng dụng kiểm thử tự động tại Công ty A1 Consulting.

CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM VÀ CÁC VẤN ĐỀ LIÊN QUAN

Chương 1 đề án sẽ khảo sát các vấn đề liên quan đến kiểm thử phần mềm, làm cơ sở cho các nghiên cứu chuyên sâu ở các chương sau.

1.1 Tổng quan về kiểm thử phần mềm

1.1.1 Khái niệm

Kiểm thử phần mềm là quá trình đánh giá nhằm xác định phần mềm có đáp ứng đúng yêu cầu hay không, đồng thời phát hiện và loại bỏ lỗi. Tester sử dụng cả kiểm thử thủ công và tự động để nâng cao hiệu quả, sau đó báo cáo kết quả cho nhóm phát triển. Mục tiêu chính là đảm bảo phần mềm hoạt động đúng như thiết kế, không phát sinh hành vi ngoài ý muốn. Đây là giai đoạn quan trọng trong phát triển phần mềm, giúp đánh giá mức độ hoàn thiện và phù hợp của hệ thống.

1.1.2 Mục đích của quá trình kiểm thử

Quá trình kiểm thử phần mềm nhằm các mục đích sau đây.

- Tìm các bug phát sinh do dev tạo ra khi code.
- Nhằm đảm bảo độ tin cậy và cung cấp cái nhìn rõ ràng về chất lượng của phần mềm.
- Để ngăn ngừa lỗi.
- Kiểm thử phần mềm giúp đảm bảo rằng sản phẩm được phát triển đúng với yêu cầu đã đề ra từ phía doanh nghiệp và người dùng cuối, đồng thời là cơ sở để tạo dựng lòng tin nơi khách hàng thông qua chất lượng phần mềm.

1.1.3 Các nguyên tắc cơ bản của kiểm thử phần mềm

Nguyên tắc 1: Kiểm thử phần mềm chứng minh sự hiện diện của lỗi

Nguyên tắc 2: Kiểm thử toàn bộ là không khả thi

Nguyên tắc 3: Kiểm thử càng sớm càng tốt

Nguyên tắc 4: Sự tập trung của lỗi

Nguyên tắc 5: Nghịch lý thuốc trừ sâu

Nguyên tắc 6: Kiểm thử phụ thuộc vào ngữ cảnh

Nguyên tắc 7: Không có lỗi – Sai lầm

1.2 Quy trình, kỹ thuật kiểm thử phần mềm và các vấn đề liên quan

1.2.1 Quy trình kiểm thử phần mềm

1.2.2 Các kỹ thuật kiểm thử phần mềm

1.2.3 Mức độ kiểm thử

Có 4 mức độ kiểm thử:

Kiểm thử đơn vị (Unit Test): Kiểm tra các thành phần nhỏ nhất của phần mềm như hàm, lớp, thủ tục để đảm bảo chúng hoạt động đúng.

Kiểm thử tích hợp (Integration Test): Đánh giá sự tương tác giữa các đơn vị sau khi được kết hợp, kiểm tra luồng dữ liệu và phối hợp chức năng.

Kiểm thử hệ thống (System Test): Đánh giá toàn bộ phần mềm sau khi tích hợp, bao gồm cả chức năng và phi chức năng như hiệu năng, bảo mật, khả năng sử dụng.

Kiểm thử chấp nhận sản phẩm (Acceptance Test): Xác nhận phần mềm đáp ứng đúng yêu cầu nghiệp vụ và được người dùng cuối chấp nhận sử dụng.

1.2. 4 Thiết kế testcase

Biểu mẫu testcase gồm các nội dung:

- Mục đích kiểm thử
- Các bước thực hiện
- Kết quả mong muốn
- Kết quả thực tế

Các kỹ thuật thiết kế testcase:

- Phân vùng tương đương
- Phân tích giá trị biên
- Bảng quyết định
- Đoán lỗi

1.3 Tổng quan về kiểm thử phần mềm trong bối cảnh chuyển đổi số

Chuyển đổi số là quá trình áp dụng toàn diện công nghệ số vào mọi lĩnh vực của tổ chức nhằm thay đổi mô hình vận hành truyền thống sang hiện đại, hiệu quả hơn. Quá trình này gồm ba khía cạnh:

- Công nghệ: Ứng dụng AI, Cloud, Big Data, IoT để nâng cao khả năng thu thập, phân tích, xử lý dữ liệu.

- Quy trình: Tăng cường tự động hóa, giảm thao tác thủ công, nâng cao hiệu suất.
- Con người: Đòi hỏi thay đổi tư duy, kỹ năng và cách phối hợp để thích nghi với mô hình mới.

1.4 Kết luận chương 1

Trong chương này, đề án đã trình bày những cơ sở lý thuyết nền tảng về kiểm thử phần mềm và chuyển đổi số, bao gồm các khái niệm, vai trò, quy trình thực hiện cũng như những thách thức liên quan trong thực tiễn.

Trên cơ sở nội dung đã nghiên cứu ở chương 1, chương tiếp theo của đề án sẽ tập trung vào việc nghiên cứu chuyên sâu về kỹ thuật và công cụ kiểm thử tự động nhằm nâng cao hiệu quả và chất lượng trong quá trình kiểm thử phần mềm.

CHƯƠNG 2: NGHIÊN CỨU KỸ THUẬT VÀ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG

2.1 Tổng quan về kiểm thử tự động

2.2.1 Khái niệm

Kiểm thử tự động là quá trình sử dụng phần mềm để thực hiện các kịch bản kiểm thử, so sánh kết quả thực tế với mong đợi và tạo báo cáo mà không cần thao tác thủ công. Sau khi thiết lập test script, chúng có thể chạy lặp lại nhanh chóng và chính xác. Khác với kiểm thử thủ công, kiểm thử tự động có thể mô phỏng nhiều hành động người dùng và đặc biệt hiệu quả trong kiểm thử hồi quy sau mỗi lần cập nhật phần mềm.

2.1.2 Vai trò

Kiểm thử tự động đóng vai trò chiến lược trong toàn bộ vòng đời phát triển phần mềm. Trong bối cảnh doanh nghiệp chuyển đổi số và các phần mềm được phát hành nhanh hơn (mỗi tuần hoặc mỗi ngày), việc kiểm thử thủ công sẽ không thể đáp ứng được tốc độ và độ tin cậy cần thiết.

Một số vai trò cốt lõi của kiểm thử tự động gồm:

- Tối ưu hóa thời gian kiểm thử
- Tăng độ chính xác
- Tích hợp kiểm thử vào CI/CD
- Giảm chi phí dài hạn
- Cải thiện độ bao phủ kiểm thử

2.2 Các kỹ thuật kiểm thử tự động

Trong lĩnh vực kiểm thử phần mềm, việc áp dụng đúng kỹ thuật kiểm thử là yếu tố quan trọng để đạt được hiệu quả và tính bao phủ cao. Đặc biệt trong kiểm thử tự động, các kỹ thuật được lựa chọn không chỉ dựa trên yêu cầu kiểm tra của hệ thống mà còn phụ thuộc vào khả năng tái sử dụng, mức độ tự động hóa và mức độ phức tạp của chức năng cần kiểm thử. Dưới đây là ba kỹ thuật kiểm thử tự động tiêu biểu và phổ biến nhất trong thực tế triển khai.

2.2.1. Kiểm thử hồi quy (Regression Testing)

Kiểm thử hồi quy là kỹ thuật được sử dụng nhằm đảm bảo rằng các chức năng đã hoạt động đúng trước đó sẽ tiếp tục hoạt động chính xác sau mỗi lần thay đổi mã nguồn. Sự thay đổi này có thể là do sửa lỗi, cập nhật tính năng hoặc thay đổi kiến trúc hệ thống.

Quy trình kiểm thử hồi quy

Quy trình kiểm thử hồi quy thường bao gồm các bước:

Bước 1: Xác định phạm vi kiểm thử hồi quy

Bước 2: Tái sử dụng bộ test case sẵn có

Bước 3: Thực thi kiểm thử

Bước 4: Đánh giá kết quả và phân tích lỗi

Tự động hóa trong kiểm thử hồi quy

Kiểm thử hồi quy cần lặp lại thường xuyên khi có thay đổi, nên việc tự động hóa là cần thiết để:

- Tiết kiệm thời gian và chi phí nhân sự
- Tăng độ bao phủ kiểm thử cho hệ thống lớn
- Đảm bảo độ chính xác và nhất quán
- Hỗ trợ kiểm thử liên tục trong DevOps

Các công cụ phổ biến gồm: Selenium, TestNG, Katalon Studio, JUnit, Appium, và tích hợp với CI/CD như Jenkins, GitLab CI.

2.2.2. Kiểm thử API

Quy trình kiểm thử API

Quy trình kiểm thử API tiêu chuẩn thường bao gồm 4 bước:

Bước 1: Xác định API cần kiểm thử: Endpoint, phương thức HTTP, dữ liệu và xác thực.

Bước 2: Thiết kế test case: Gồm kiểm thử đúng, sai, giá trị biên, dữ liệu thiếu/hỏng.

Bước 3: Thực thi kiểm thử: Gửi request, phân tích response, so sánh kết quả.

Bước 4: Đánh giá kết quả: Ghi nhận lỗi, kiểm tra hiệu suất và mức độ đáp ứng của API.

2.2.3. Kiểm thử hiệu năng (Performance Testing)

Kiểm thử hiệu năng là kỹ thuật phi chức năng nhằm đánh giá khả năng vận hành của phần mềm dưới các mức tải khác nhau, tập trung vào tốc độ phản hồi, độ ổn định, khả năng mở rộng và độ bền. Gồm các hình thức chính:

- Kiểm thử tải (Load Testing): Đánh giá hệ thống khi hoạt động ở mức tải thiết kế.
- Kiểm thử áp lực (Stress Testing): Kiểm tra khả năng phản hồi và phục hồi khi vượt ngưỡng tải tối đa.
- Kiểm thử độ bền (Soak Testing): Cho chạy liên tục thời gian dài để phát hiện rò rỉ bộ nhớ, suy giảm hiệu suất.
- Kiểm thử khả năng mở rộng (Scalability Testing): Đánh giá khả năng mở rộng mà vẫn giữ hiệu năng ổn định.

2.3. Các công cụ kiểm thử tự động

Việc lựa chọn công cụ kiểm thử phù hợp là một yếu tố quan trọng để triển khai kiểm thử tự động thành công. Trên thị trường hiện nay có nhiều công cụ hỗ trợ kiểm thử tự động, mỗi công cụ lại phục vụ cho các mục đích khác nhau như kiểm thử giao diện người dùng, kiểm thử API, hoặc kiểm thử hiệu năng. Tùy vào mục tiêu kiểm thử, quy mô hệ thống và năng lực kỹ thuật của đội ngũ mà doanh nghiệp có thể lựa chọn công cụ phù hợp.

Trong mục này, đề án giới thiệu một số công cụ kiểm thử tự động phổ biến bao gồm Selenium, Katalon, Postman và Jmeter.

2.3.1 Selenium

2.3.2 Katalon Studio

2.3.3 Playwright

2.3.4 Cypress

2.3.5 Appium

2.3.6 Postman

2.3.7 Apache Jmeter

2.4 So sánh và đánh giá các công cụ kiểm thử

Bảng so sánh tổng hợp một số công cụ phổ biến:

Bảng 0-1: So sánh công cụ kiểm thử

Công cụ	Ưu điểm	Hạn chế
Selenium	Mã nguồn mở, linh hoạt, hỗ trợ đa trình duyệt và ngôn ngữ, tích hợp CI dễ	Yêu cầu kỹ năng lập trình, không có giao diện người dùng trực quan
Katalon	Giao diện thân thiện, hỗ trợ đa nền tảng, dễ học	Một số tính năng nâng cao chỉ có ở bản thương mại
Postman	Dễ sử dụng, mạnh về kiểm thử API, hỗ trợ viết test script và xuất báo cáo	Không hỗ trợ kiểm thử giao diện người dùng
JMeter	Mạnh về kiểm thử tải, hiệu năng, có khả năng mở rộng tốt	Giao diện chưa thân thiện, cần thời gian làm quen

Trong số các công cụ trên, Selenium nổi bật với tính năng mạnh mẽ trong kiểm thử UI, khả năng mở rộng và tùy biến cao. Đây là công cụ được đề án lựa chọn vì phù hợp với mục tiêu kiểm thử giao diện web, dễ tích hợp vào quy trình DevOps và có cộng đồng người dùng đông đảo, tài liệu phong phú hỗ trợ triển khai thực tế.

2.5 Lợi ích và thách thức khi áp dụng kiểm thử tự động

2.5.1 Lợi ích

- Tăng hiệu quả kiểm thử
- Tăng tính nhất quán và độ tin cậy
- Hỗ trợ kiểm thử hồi quy hiệu quả
- Tích hợp vào quy trình DevOps/CI-CD
- Tiết kiệm chi phí về lâu dài

2.5.2 Thách thức

- Chi phí triển khai ban đầu cao
- Yêu cầu kỹ năng kỹ thuật cao
- Bảo trì test script tốn thời gian
- Không phù hợp với tất cả loại kiểm thử
- Sai lầm khi lạm dụng tự động hóa

2.6 Kết luận chương 2

Chương 2 đã trình bày tổng quan về kiểm thử tự động phần mềm, bao gồm khái niệm, vai trò, các kỹ thuật kiểm thử phổ biến như kiểm thử hồi quy, kiểm thử API và kiểm thử hiệu năng. Đồng thời, chương cũng phân tích các công cụ kiểm thử tự động đang được sử dụng rộng rãi như Selenium, Katalon Studio, Postman và Apache JMeter, kèm theo đánh giá so sánh dựa trên nhiều tiêu chí thực tiễn.

Ngoài ra, chương cũng chỉ ra những lợi ích nổi bật khi áp dụng kiểm thử tự động trong quy trình phát triển phần mềm hiện đại, song song với những thách thức mà tổ chức có thể gặp phải khi triển khai.

Những nội dung trong chương 2 sẽ là nền tảng để chương tiếp theo tập trung vào ứng dụng kiểm thử tự động với công cụ Selenium trong thực tế, từ đó đánh giá hiệu quả và đề xuất hướng áp dụng phù hợp trong chuyển đổi số tại doanh nghiệp.

CHƯƠNG 3: ỨNG DỤNG KIỂM THỬ TỰ ĐỘNG PHẦN MỀM TẠI CÔNG TY A1 CONSULTING

3.1 Giới thiệu về Công ty A1 Consulting

3.1.1 Lịch sử hình thành và phát triển

3.1.2. Lĩnh vực hoạt động và vị thế hiện tại

3.1.3. Thực trạng kiểm thử phần mềm tại A1 Consulting

Do triển khai phần mềm ERP tùy biến cao, A1 Consulting cần kiểm thử linh hoạt và chính xác. Trước đây, công ty chủ yếu dùng kiểm thử thủ công. Gần đây, nhằm nâng cao chất lượng và tối ưu chi phí, A1 đã bắt đầu áp dụng kiểm thử tự động cho các hệ thống nội bộ trên nền tảng Odoo, và mở rộng ra các dự án khách hàng.

3.2 Giới thiệu phần mềm lựa chọn để thực nghiệm kiểm thử

Trong phạm vi đề án này, phần mềm được lựa chọn để thực nghiệm kiểm thử tự động là Odoo, một hệ thống ERP mã nguồn mở có kiến trúc mô-đun linh hoạt và được sử dụng rộng rãi trong các tổ chức và doanh nghiệp trên toàn thế giới. Đề án tập trung kiểm thử trên hai phân hệ cốt lõi: Sales (Bán hàng) và Purchase (Mua hàng) - hai phân hệ đại diện cho quy trình vận hành chính trong chuỗi cung ứng của doanh nghiệp.

3.2.1. Giới thiệu phân hệ Sales và Purchase trong Odoo

3.2.1.1 Phân hệ Sales (Bán hàng)

Các chức năng chính của phân hệ Sales bao gồm:

- Quản lý thông tin khách hàng và điều khoản thanh toán
- Tạo, gửi và xác nhận báo giá
- Quản lý đơn hàng bán (Sales Order)
- Theo dõi trạng thái đơn hàng và tình trạng giao hàng
- Tự động hóa việc tạo hóa đơn và liên kết với phân hệ Kế toán
- Báo cáo và phân tích bán hàng

3.2.1.2 Phân hệ Purchase (Mua hàng)

Các chức năng chính của phân hệ Purchase bao gồm

- Quản lý danh sách nhà cung cấp và điều khoản thanh toán
- Tạo và xác nhận yêu cầu báo giá (Request for Quotation – RFQ)

- Tự động chuyển đổi RFQ thành đơn mua hàng (Purchase Order)
- Theo dõi quá trình nhận hàng và ghi nhận hóa đơn mua
- Tích hợp với phân hệ Kho để cập nhật tồn kho
- Phân tích hiệu suất nhà cung cấp và chi phí mua hàng

3.2.2 Yêu cầu và công cụ thử nghiệm

Công cụ thử nghiệm được sử dụng

Để đáp ứng yêu cầu kỹ thuật và đảm bảo khả năng mở rộng trong tương lai, đề án sử dụng kết hợp các công cụ kiểm thử sau:

Bảng 0-1: Công cụ thử nghiệm được sử dụng

Công cụ	Vai trò và ứng dụng
Selenium WebDriver	Tự động hóa các thao tác giao diện trên trình duyệt, đặc biệt hữu ích cho các bước nhập liệu, xác nhận đơn hàng và điều hướng giao diện người dùng trong Odoo
TestNG (Java)	Hỗ trợ quản lý và tổ chức bộ test case, cấu hình kiểm thử theo nhóm (suite), cung cấp báo cáo kết quả chi tiết
Postman (bổ trợ)	Được sử dụng để kiểm thử một số API nội bộ hoặc khi cần xác minh trạng thái hệ thống ở backend, chẳng hạn như xác thực việc tạo đơn hàng hoặc truy xuất dữ liệu sản phẩm
Dữ liệu thử nghiệm mẫu	Bao gồm các tập dữ liệu về khách hàng, nhà cung cấp, sản phẩm và đơn hàng để đảm bảo thử nghiệm lặp lại, nhất quán và dễ kiểm soát.

3.3 Triển khai thực nghiệm

Sau khi đánh giá tổng thể các đặc điểm của hệ thống cần kiểm thử – cụ thể là phân hệ Sales trong Odoo, với giao diện web động, có nhiều thao tác chuột, biểu mẫu phức tạp và luồng nghiệp vụ xuyên suốt – công cụ Selenium đã được lựa chọn để triển khai kiểm thử tự động.

Kiến trúc kiểm thử áp dụng:

- Page Classes: định nghĩa các locator và hành động trên từng trang cụ thể (ví dụ: LoginPage.java, CreateOrderPage.java);
- Test Classes: thực hiện kiểm thử theo luồng nghiệp vụ bằng cách gọi lại các Page đã định nghĩa;

- Base & Utility Classes: xử lý khởi tạo WebDriver, thiết lập báo cáo, log, chờ đợi;
- Test Framework: sử dụng TestNG để tổ chức test suite, báo cáo kết quả và thực hiện assertion.

3.3.1 Xây dựng kịch bản kiểm thử thủ công các tính năng ứng dụng kiểm thử tự động

Các chức năng chính được đưa vào kiểm thử gồm:

- Đăng nhập hệ thống
- Tạo mới yêu cầu bán hàng
- Xác nhận yêu cầu bán hàng
- Tìm kiếm đơn hàng
- Xác nhận phiếu Xuất kho
- Tạo yêu cầu mua hàng
- Xác nhận đơn mua hàng
- Xác nhận phiếu Nhập kho
- Quy trình Trả hàng

3.3.2 Xây dựng kịch bản kiểm thử tự động theo các kịch bản thủ công đã xây dựng.

Sau khi hoàn thiện hệ thống kịch bản kiểm thử thủ công cho từng chức năng, kịch bản kiểm thử tự động được xây dựng dựa trên các test case đã xác thực. Quá trình chuyển đổi từ kiểm thử thủ công sang kiểm thử tự động đảm bảo rằng:

- Các bước nghiệp vụ được mô phỏng đầy đủ và chính xác;
- Logic xử lý và dữ liệu đầu vào giống hệt kịch bản thủ công;
- Đầu ra (kết quả mong đợi) được kiểm tra tự động thông qua điều kiện assert.

Các kịch bản được chia thành hai nhóm chính:

- Kiểm thử frontend (UI): Các thao tác thông qua trình duyệt, thực hiện bởi người dùng cuối, bao gồm: đăng nhập, tạo đơn hàng, xác nhận phiếu kho, tìm kiếm báo giá...
- Kiểm thử backend (API): Các luồng xử lý hệ thống, không qua giao diện người dùng, bao gồm: login API, tạo đơn hàng qua API, xác nhận đơn, xử lý phiếu kho, trả hàng...

3.3.4 Thực thi kiểm thử

Quá trình thực thi được cấu hình thông qua tệp testng.xml, cho phép kiểm thử được triển khai linh hoạt theo từng lớp kiểm thử, từng chức năng nghiệp vụ riêng lẻ, hoặc toàn bộ hệ thống.

Đề án sử dụng framework TestNG để điều phối quá trình kiểm thử. Đây là công cụ hỗ trợ tổ chức và quản lý test case một cách hiệu quả, cho phép thiết lập các kịch bản chạy theo từng đơn vị kiểm thử (<test>), theo từng lớp (<class>) hoặc theo trình tự nghiệp vụ được xác định trước.

Các lớp kiểm thử được phân chia rõ ràng theo mục đích kiểm thử, bao gồm:

- Frontend (FE): kiểm thử các thao tác của người dùng thông qua trình duyệt với sự hỗ trợ của Selenium.
- Backend (BE): kiểm thử logic nghiệp vụ bằng cách gửi trực tiếp các yêu cầu HTTP (API) tới hệ thống Odoo.

Tất cả các lớp kiểm thử đều được khai báo và quản lý tập trung trong tệp testng.xml, đóng vai trò như tệp điều phối chính của toàn bộ quy trình kiểm thử tự động.

3.4 Kết quả kiểm thử

Sau khi hoàn thiện 28 kịch bản kiểm thử tự động cho các chức năng trọng yếu như đăng nhập, xử lý đơn hàng, kiểm tra kho và xác nhận dữ liệu, quá trình thực nghiệm được triển khai trong 8 tuần với tần suất thực thi 2 lần/tuần, tổng cộng 16 đợt chạy kiểm thử toàn bộ hệ thống.

3.5 Đánh giá kết quả

Bảng 0-2: Đánh giá kết quả

Tiêu chí	Trước khi áp dụng kiểm thử tự động	Sau khi áp dụng kiểm thử tự động	Mức cải thiện
Thời gian kiểm thử hồi quy mỗi lần	6 giờ (3h staging + 3h production)	2 giờ (1h tự động + 1h thủ công)	Giảm 66.7% (4 giờ/lần)
Tần suất kiểm thử hồi quy	2 lần/tuần	2 lần/tuần	Không đổi
Tổng thời gian kiểm thử hồi quy 8 tuần	96 giờ	32 giờ	Giảm 64 giờ
Số test case xây dựng tuần 1	13 test case đơn giản	-	-

Tổng số test case hoàn thiện tuần 3	0	28 test case	Hoàn thiện toàn bộ
Thời gian xây dựng trung bình/test case	1.5–2 giờ	~1 giờ	Tiết kiệm 0.5–1 giờ/test case
Thời gian bảo trì ban đầu (20 test case)	6 giờ	-	-
Thời gian bảo trì tuần 8 (28 test case)	-	4 giờ	Giảm >33%

3.6 Kết luận chương 3

Trong chương này, đề án trình bày quá trình ứng dụng kiểm thử tự động vào hệ thống Odoo ERP trong bối cảnh chuyển đổi số, với mục tiêu đánh giá hiệu quả và rút ra bài học thực tế. Kết quả thực nghiệm cho thấy kiểm thử tự động giúp phát hiện lỗi sớm, tăng độ ổn định hệ thống, và tiết kiệm thời gian so với phương pháp thủ công.

KẾT LUẬN

Kết quả đạt được của đề án

Đề án “Nghiên cứu kiểm thử tự động phần mềm và ứng dụng tại công ty A1 Consulting” đã bước đầu tiếp cận và triển khai kiểm thử tự động trong môi trường doanh nghiệp, góp phần nâng cao hiệu quả kiểm thử, rút ngắn thời gian và hỗ trợ kiểm soát chất lượng phần mềm tốt hơn. Cụ thể đề án đã đạt được các kết quả sau:

- Tiết kiệm thời gian kiểm thử hồi quy: Thời gian kiểm thử hồi quy giảm từ 6 giờ mỗi lần xuống còn 2 giờ, tương đương giảm 66.7%, tiết kiệm 64 giờ trong 8 tuần. Điều này giúp tăng tốc độ phát hành sản phẩm và giảm tải công việc cho đội ngũ kiểm thử.
- Hiệu quả xây dựng và bảo trì test case: Sử dụng mô hình Page Object Model phối hợp với Selenium và TestNG giúp tối ưu hóa việc tổ chức và tái sử dụng mã nguồn test case, giảm thời gian bảo trì hơn 33% sau 8 tuần vận hành, đồng thời nâng cao độ ổn định và khả năng mở rộng của bộ test.
- Phát hiện lỗi sớm, nâng cao chất lượng: Kiểm thử tự động hỗ trợ phát hiện lỗi sớm trong quá trình phát triển, giảm thiểu lỗi phát sinh và cải thiện độ ổn định của hệ thống.
- Giảm thiểu công sức kiểm thử thủ công: Giảm đáng kể khối lượng công việc thủ công, giúp đội kiểm thử tập trung vào các kịch bản phức tạp và sáng tạo hơn.
- Đánh giá tính khả thi của công cụ và mô hình: Mô hình Page Object Model và công cụ Selenium được chứng minh phù hợp và hiệu quả khi áp dụng trong môi trường thực tế doanh nghiệp.

Hướng phát triển tiếp theo

Để tiếp tục nâng cao hiệu quả và mở rộng phạm vi áp dụng của hệ thống kiểm thử tự động, đề án đề xuất một số hướng phát triển trong tương lai như sau:

- Mở rộng phạm vi kiểm thử: Phát triển thêm các test case để bao phủ toàn diện hơn các chức năng, bao gồm các tình huống đặc biệt (edge cases), dữ liệu biên (boundary values), cũng như các bài kiểm thử hiệu năng và bảo mật.

- Tích hợp quy trình kiểm thử vào hệ thống CI/CD: Thiết lập cơ chế kích hoạt kiểm thử tự động ngay khi có thay đổi trong mã nguồn (continuous testing), đảm bảo phản hồi sớm và liên tục trong quá trình phát triển phần mềm.
- Tối ưu hóa thời gian thực thi: Nghiên cứu áp dụng kiểm thử song song (parallel testing) và đa trình duyệt (cross-browser testing) nhằm rút ngắn thời gian thực hiện và đảm bảo khả năng tương thích của hệ thống trên nhiều môi trường khác nhau.
- Nâng cao trực quan hóa kết quả kiểm thử: Tích hợp các công cụ báo cáo nâng cao như Allure Report hoặc ExtentReports để hiển thị kết quả kiểm thử một cách trực quan, dễ theo dõi và thuận tiện cho việc đánh giá và phân tích.
- Xây dựng hệ thống kiểm thử hoàn toàn tự động: Tiến tới triển khai quy trình kiểm thử khép kín bao gồm khởi tạo dữ liệu đầu vào, thực thi kiểm thử, xác minh kết quả và dọn dẹp môi trường sau khi kiểm thử, qua đó giảm thiểu tối đa sự can thiệp thủ công.

Các định hướng trên không chỉ giúp hoàn thiện hệ thống kiểm thử tự động hiện có mà còn tạo nền tảng để nhân rộng và áp dụng ở quy mô lớn hơn, đáp ứng yêu cầu phát triển phần mềm hiện đại trong bối cảnh DevOps và Agile ngày càng phổ biến, nơi mà tốc độ và chất lượng đều là những yếu tố then chốt.