

ĐOÀN HƯƠNG LY

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN

ĐOÀN HƯƠNG LY

KIỂM THỬ TỰ ĐỘNG CHO WEBSITE CAMERA
SAMTECH SỬ DỤNG CÔNG CỤ KATALON STUDIO VÀ
JMETER

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

HƯNG YÊN - 2022

2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN

ĐOÀN HƯƠNG LY

KIỂM THỬ TỰ ĐỘNG CHO WEBSITE CAMERA
SAMTECH SỬ DỤNG CÔNG CỤ KATALON STUDIO VÀ
JMETER

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

NGƯỜI HƯỚNG DẪN
NGÔ THANH HUYỀN

HƯNG YÊN - 2021

NHẬN XÉT

Nhận xét của giảng viên hướng dẫn:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan đồ án tốt nghiệp “Kiểm thử tự động cho website camera Samtech sử dụng công cụ Katalon Studio và Jmeter” là công trình nghiên cứu của bản thân. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai em xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của Bộ môn và Nhà trường đề ra.

Hưng Yên, ngày ... tháng ... năm.....

Sinh viên

.....

MỤC LỤC

MỤC LỤC.....	5
DANH SÁCH CÁC THUẬT NGỮ.....	8
DANH SÁCH BẢNG BIỂU.....	9
DANH SÁCH HÌNH VẼ.....	10
CHƯƠNG 1: MỞ ĐẦU.....	12
1.1 Lý do chọn đề án.....	12
1.2 Mục tiêu của đề án.....	12
1.2.1 Mục tiêu tổng quát.....	12
1.2.2 Mục tiêu cụ thể.....	12
1.3 Giới hạn và phạm vi của đề án.....	13
1.3.1 Đối tượng nghiên cứu.....	13
1.3.2 Phạm vi nghiên cứu.....	13
1.4 Nội dung thực hiện.....	13
1.5 Phương pháp tiếp cận.....	13
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	14
2.1 Tổng quan về kiểm thử phần mềm.....	14
2.1.1 Các cấp độ kiểm thử phần mềm.....	14
2.1.2 Kỹ thuật kiểm thử phần mềm.....	16
2.2 Kiểm thử tự động phần mềm.....	16
2.3 Kiểm thử chức năng.....	24
2.3.1 Kiểm thử chức năng.....	24
2.4 Hướng kiểm thử tự động.....	25

2.5 Công cụ áp dụng.....	29
2.5.1 Công cụ Katalon Studio	29
2.5.2 Công cụ Jmeter	34
CHƯƠNG 3: ĐẶC TẢ HỆ THỐNG PHẦN MỀM	40
3.1 Giới thiệu về hệ thống phần mềm	40
3.2 Các yêu cầu chức năng.....	41
3.2.2 Chức năng đăng nhập.....	42
3.2.3 Đăng ký	44
3.2.4 Chức năng tìm kiếm.....	46
3.2.5 Chức năng quản lý giỏ hàng	47
3.2.6 Chức năng đặt hàng.....	48
3.3 Các yêu cầu phi chức năng.....	50
CHƯƠNG 4: TRIỂN KHAI KIỂM THỬ TỰ ĐỘNG	51
4.1 Thiết kế các yêu cầu kiểm thử.....	51
4.1.1 Thiết kế kiểm thử tổng quan hệ thống	51
4.1.2 Thiết kế kiểm thử chức năng.....	52
4.1.3 Thiết kế kiểm thử hiệu năng.....	58
4.2 Xây dựng ca kiểm thử	59
4.2.1 Test Case Đăng Nhập.....	59
4.2.2 Test Case Đăng Nhập.....	62
4.2.3 Test Case Giỏ Hàng.....	67
4.2.4 Test Case Tìm kiếm.....	69
4.2.5 Test Case Đặt hàng.....	71
4.2.6 Test Case hiệu năng.....	77

4.3	Xây dựng dữ liệu kiểm thử.....	80
4.3.1	Dữ liệu đăng nhập	80
4.3.2	Dữ liệu đăng ký	81
4.3.4	Dữ liệu đặt hàng	82
4.4	Xây dựng kịch bản kiểm thử	83
4.4.1	Keyword Đăng ký	83
4.4.2	Keyword Đăng nhập.....	85
4.4.3	Keyword Tìm kiếm	87
4.4.4	Keyword Đặt hàng.....	88
4.4.5	Test scrip giỏ hàng	94
4.5	Thực thi và báo cáo kiểm thử	95
4.5.1	Thực thi kiểm thử katalon studio	95
4.5.2	Thực thi kiểm thử Jmeter	100
4.5.3	Báo cáo trên katalon studio	108
4.5.4	Báo cáo Jmeter	111
4.5.5	Test Report	122
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		123
Kết quả đạt được.....		123
Hạn chế của đề tài		123
Hướng phát triển của đề tài		123
TÀI LIỆU THAM KHẢO.....		124

DANH SÁCH CÁC THUẬT NGỮ

Từ viết tắt	Từ đầy đủ	Giải thích
SRS	Software Requirement Specification	Đặc tả yêu cầu phần mềm
AVG	average	Giá trị trung bình
CMMI	Capability Maturity Model Integration	Khuôn khổ cải tiến quy trình dành cho các doanh nghiệp phần mềm
ISO	International Organization for Standardization	Tổ chức tiêu chuẩn hóa Quốc tế

DANH SÁCH BẢNG BIỂU

Bảng 2. 1 So sánh katalon studio và selenium	32
Bảng 2. 2 So sánh Jmeter và HP LoadRunner	38
Bảng 3. 1 Bảng mô tả use case	41
Bảng 3. 2 Yêu cầu validate của chức năng [Tìm kiếm]	46
Bảng 3. 3 Nội dung thông báo của chức năng [Tìm kiếm]	46
Bảng 3. 4 Nội dung thông báo của chức năng [Quản lý giỏ hàng]	47
Bảng 3. 5 Yêu cầu validate của chức năng [Đặt hàng]	49
Bảng 3. 6 Nội dung thông báo của chức năng [Đặt hàng]	50
Bảng 4. 1 Dữ liệu đăng nhập	80
Bảng 4. 2 Dữ liệu đăng ký	81
Bảng 4. 3 Dữ liệu tìm kiếm	82
Bảng 4. 4 Dữ liệu đặt hàng	82
Bảng 4. 5 Báo Cáo Jmeter	111
Bảng 4. 6 Bảng hình ảnh minh chứng Jmeter	119

DANH SÁCH HÌNH VẼ

Hình 2. 1 Các cấp độ kiểm thử phần mềm.....	14
Hình 2. 2 Hình ảnh minh họa về công cụ Katalon Studio	29
Hình 2. 3 Hình ảnh minh họa công cụ Jmeter.....	34
Hình 2. 4 Quy trình làm việc của công cụ Jmeter.....	36
Hình 2. 5 Ưu điểm của công cụ Jmeter.....	36
Hình 3. 1 Use case tổng quát.....	41
Hình 3. 2 Use case đăng nhập	42
Hình 3. 3 Màn hình đăng nhập hệ thống	43
Hình 3. 4 Use case Đăng ký	44
Hình 3. 5 Màn hình đăng ký tài khoản.....	45
Hình 3. 6 Màn hình hiển thị chức năng [Tìm kiếm]	46
Hình 3. 7 Màn hình hiển thị chức năng [Quản lý giỏ hàng]	47
Hình 3. 8 Màn hình hiển thị giỏ hàng khi chưa đăng nhập.....	48
Hình 3. 9 Màn hình hiển thị giỏ hàng khi đã đăng nhập.....	48
Hình 4. 1 Test design “Tổng quát”.....	51
Hình 4. 2 Test design “Đăng nhập”	52
Hình 4. 3 Test design “Đăng ký”	53
Hình 4. 4 Test design “Quản lý Giỏ hàng”	54
Hình 4. 5 Test design “Tìm kiếm”	55
Hình 4. 6 Test design “Đặt hàng”	56
Hình 4. 7 Test design “System Test Đặt hàng”	57

Hình 4. 8 Test design “Hiệu năng”	58
Hình 4. 9 : Kết quả chạy Test Suite Đăng nhập trên Katalon Studio	95
Hình 4. 10 :Kết quả chạy Test Suite Đăng ký trên Katalon Studio	96
Hình 4. 11 : Kết quả chạy Test Suite Tìm kiếm trên Katalon Studio	97
Hình 4. 12 Kết quả chạy Test Suite Giỏ Hàng trên Katalon Studio	98
Hình 4. 13 : Kết quả chạy Test Suite Đặt Hàng trên Katalon Studio	99
Hình 4. 14 Kết quả chạy Đăng ký trên jmeter	100
Hình 4. 15 Kết quả chạy Đăng nhập trên jmeter.....	101
Hình 4. 16 Kết quả chạy Giỏ Hàng trên jmeter	102
Hình 4. 17 Kết quả chạy Đặt Hàng trên jmeter.....	103
Hình 4. 18: 10 người truy cập 1s sau 20 người truy cập 3s sau 50 người truy cập (cùng trang web).....	104
Hình 4. 19. 10 người truy cập trang đăng nhập 1s sau 20 người truy cập trang chủ 3s sau 50 người truy cập giỏ hàng	105
Hình 4. 20 Kiểm tra 10 ng truy cập đồng thời các trang web(đăng nhập, đăng ký, giỏ hàng, trang chủ).....	106
Hình 4. 21: Kiểm tra 300 ng truy cập trong 1s	107
Hình 4. 22 Báo cáo kết quả Test Suite trên katalon -Test Ops	108
Hình 4. 23 Báo cáo Test Run-Execution Result trên Katalon-Test Ops.....	109
Hình 4. 24. Báo cáo chi tiết Test Run Trên Katalon- Test Ops	110

CHƯƠNG 1: MỞ ĐẦU

1.1 Lý do chọn đề án

Công nghệ thông tin đang ngày càng phát triển nhanh chóng, kéo theo đó là hệ thống mạng và các phần mềm cũng gia tăng cả về số lượng theo quy mô và cả về chất lượng phần mềm theo chiều sâu. Nhưng cũng từ đó đã nảy sinh ra nhiều vấn đề về lỗi hỏng hóc phần mềm không đáng có gây ra các ảnh hưởng nghiêm trọng đến xã hội, kinh tế,... Vì vậy, để kiểm thử được một phần mềm lớn sẽ tốn rất nhiều công sức và thời gian nếu làm thủ công, chưa kể đến chất lượng kiểm thử sẽ không cao và thật chính xác phù hợp cho yêu cầu. Theo nhiều tính toán thì công việc kiểm thử đóng vai trò hết sức quan trọng trong quy trình phát triển phần mềm, nó đóng góp tới 40% tổng toàn bộ chi phí cho phép ta thực hiện được các công việc một cách nhanh chóng và độ an toàn, chính xác cao nhất có thể. Do vậy, nhu cầu tự động hoá quy trình kiểm thử phần mềm cũng được đặt ra. Qua thực tế cho thấy, việc áp dụng kiểm thử tự động hợp lý sẽ mang lại thành công cho hoạt động kiểm thử phần mềm. Kiểm thử tự động giúp giảm bớt công sức thực hiện, tăng độ tin cậy, giảm sự nhầm lẫn và rèn luyện kỹ năng lập trình cho người kiểm thử. Đó là lý do để em lựa chọn đề tài “Kiểm thử tự động cho *website camera Samtech*” để nghiên cứu, tìm kiếm và đề ra các giải pháp mới để cải tiến quy trình kiểm thử như hiện nay sao cho có năng suất cao nhất.

1.2 Mục tiêu của đề án

1.2.1 Mục tiêu tổng quát

Hoàn thiện website so với yêu cầu của người sử dụng. Đảm bảo rằng Website hoàn toàn và chắc chắn rằng nó hoạt động tốt và theo đúng các thông số kỹ thuật.

1.2.2 Mục tiêu cụ thể

- Kiểm thử bao phủ các lĩnh vực khác nhau như: hoạt động của các chức năng, khả năng tương thích cả website với các trình duyệt, hiệu năng của website...
- Kiểm tra độ thân thiện với người dùng.
- Đảm bảo rằng website đáp ứng các yêu cầu khác nhau, bao gồm: chức năng, hiệu suất, độ tin cậy, an toàn, khả năng sử dụng.

- Xây dựng bản test case tìm ra các lỗi trong phần mềm.
- Xây dựng bản thiết kế kiểm thử.
- Báo cáo lỗi phần mềm.
- Xây dựng bản đặc tả chi tiết cho ứng dụng website.

1.3 Giới hạn và phạm vi của đề án

1.3.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu: website camera Samtech

Khách thể nghiên cứu: Khách hàng

1.3.2 Phạm vi nghiên cứu

- Phạm vi không gian: Cửa hàng camera Samtech (tại tòa nhà 389, Trường Chinh, Hà Nội).
- Phạm vi thời gian: được thực hiện trong 4 tháng.
- Ý nghĩa khoa học và thực tiễn của đề tài: vận dụng kiến thức kiểm thử phần mềm và sử dụng kiểm thử tự động cho website camera Samtech

1.4 Nội dung thực hiện

Để đảm bảo cho website làm đúng đủ các yêu cầu và hạn chế lỗi, việc kiểm thử phải diễn ra theo các giai đoạn sau:

- Tìm hiểu cơ sở lý thuyết
- Đặc tả hệ thống phần mềm
- Xây dựng Test design, Test case
- Xây dựng Test script

1.5 Phương pháp tiếp cận

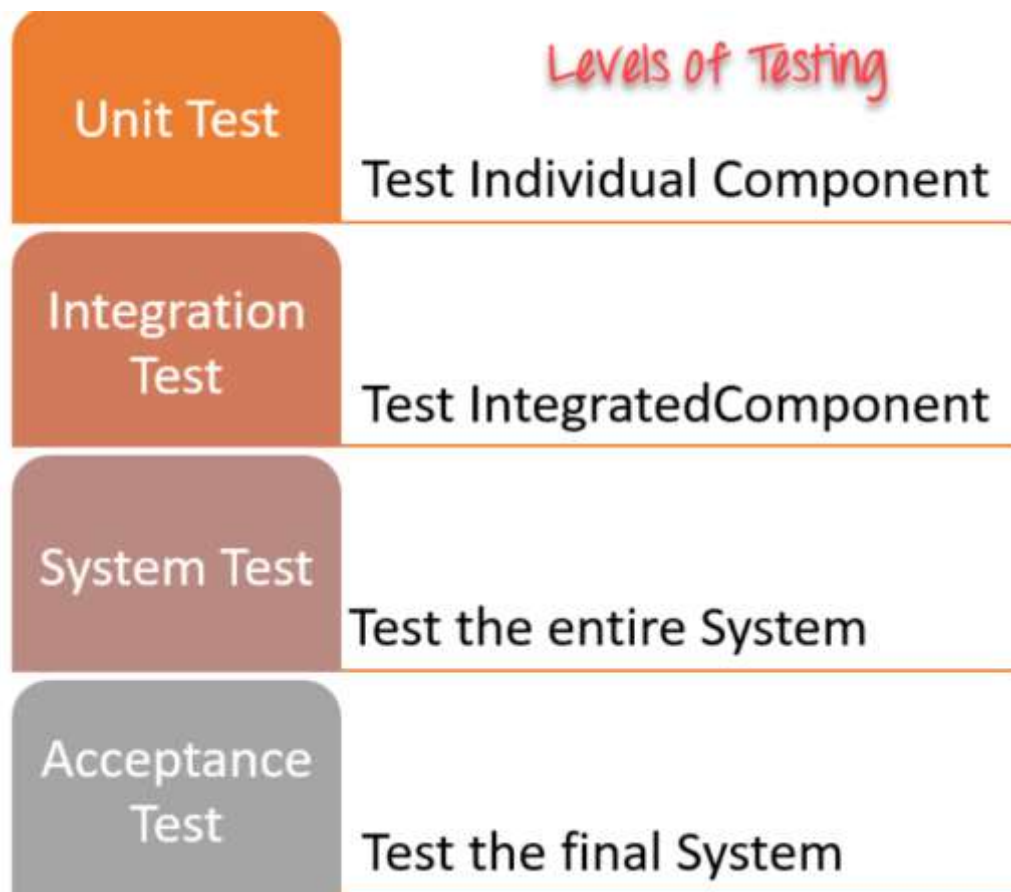
- Cách tiếp cận: Nghiên cứu các chuẩn đảm bảo chất lượng phần mềm: CMM/CMMI, ISO 9216,...
- Khảo sát một số website thời trang nữ và các hệ thống bán hàng thực tế.
- Sử dụng các phương pháp kiểm thử: Kiểm thử hộp đen, kiểm thử hộp trắng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về kiểm thử phần mềm

Kiểm thử phần mềm là quá trình thực thi một hệ thống phần mềm để xác định xem phần mềm có đúng với đặc tả không và thực hiện trong môi trường như mong đợi hay không. Mục đích của kiểm thử phần mềm là tìm ra lỗi chưa được phát hiện, tìm một cách sớm nhất và bảo đảm rằng lỗi sẽ được sửa. Mục tiêu của kiểm thử phần mềm là thiết kế tài liệu kiểm thử một cách có hệ thống và thực hiện nó sao cho có hiệu quả, nhưng tiết kiệm được thời gian, công sức và chi phí.

2.1.1 Các cấp độ kiểm thử phần mềm



Hình 2. 1 Các cấp độ kiểm thử phần mềm

Kiểm thử đơn vị (Unit Test)

Một đơn vị (Unit) là một thành phần phần mềm nhỏ nhất mà ta có thể kiểm thử được, ví dụ: các hàm (Function), thủ tục (Procedure), lớp (Class), hoặc các phương thức (Method).

Kiểm thử tích hợp (Integration Test)

Kiểm thử tích hợp kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành. Trong khi kiểm thử đơn vị kiểm tra các thành phần và Unit riêng lẻ thì kiểm thử tích hợp kết hợp chúng lại với nhau và kiểm tra sự giao tiếp giữa chúng.

Kiểm thử hệ thống (System Test)

Mục đích của kiểm thử hệ thống là kiểm thử xem thiết kế và toàn bộ hệ thống (sau khi tích hợp) có thỏa mãn yêu cầu đặt ra hay không.

Kiểm thử hệ thống kiểm tra cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật.

Kiểm thử hệ thống bắt đầu khi tất cả các bộ phận của phần mềm đã được tích hợp thành công.

Điểm khác nhau then chốt giữa kiểm thử tích hợp và kiểm thử hệ thống là kiểm thử hệ thống chú trọng các hành vi và lỗi trên toàn hệ thống, còn kiểm thử tích hợp chú trọng sự giao tiếp giữa các đơn thể hoặc đối tượng khi chúng làm việc cùng nhau. Thông thường ta phải thực hiện kiểm thử đơn vị và kiểm thử tích hợp để bảo đảm mọi Unit và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện kiểm thử hệ thống.

Kiểm thử chấp nhận (Acceptance Test)

Mục đích của kiểm thử chấp nhận là kiểm thử khả năng chấp nhận cuối cùng để chắc chắn rằng sản phẩm là phù hợp và thỏa mãn các yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm.

Trong giai đoạn kiểm thử chấp nhận thì người kiểm tra là khách hàng. Khách hàng sẽ đánh giá phần mềm với mong đợi theo những thao tác sử dụng quen thuộc

của họ. Việc kiểm tra ở giai đoạn này có ý nghĩa hết sức quan trọng tránh cho việc hiểu sai yêu cầu cũng như sự mong đợi của khách hàng.[1]

2.1.2 Kỹ thuật kiểm thử phần mềm

Mục tiêu của kiểm thử là phải thiết kế các trường hợp kiểm thử có khả năng cao nhất trong việc phát hiện nhiều lỗi với thời gian và công sức tối thiểu. Do đó có thể chia các kỹ thuật kiểm thử thành hai loại:

- Kỹ thuật kiểm thử hộp đen (Black – box Testing) hay còn gọi là kỹ thuật kiểm thử chức năng (Functional Testing).
- Kỹ thuật kiểm thử hộp trắng (White – box Testing) hay còn gọi là kỹ thuật kiểm thử cấu trúc (Structural Testing)

Kỹ thuật kiểm thử hộp đen (Black – box Testing)

Kiểm thử hộp đen còn được gọi là kiểm thử hướng dữ liệu (data - driven) hay là kiểm thử hướng vào/ra (input/output driven).

Trong kỹ thuật này, người kiểm thử xem phần mềm như là một hộp đen. Người kiểm thử hoàn toàn không quan tâm đến cấu trúc và hành vi bên trong của chương trình. Người kiểm thử chỉ cần quan tâm đến việc tìm các hiện tượng mà phần mềm không hành xử theo đúng đặc tả của nó. Do đó, dữ liệu kiểm thử sẽ xuất phát từ đặc tả.

Kỹ thuật kiểm thử hộp trắng (White – box Testing)

Kiểm thử hộp trắng hay còn gọi là kiểm thử hướng logic, cho phép kiểm tra cấu trúc bên trong của phần mềm với mục đích bảo đảm rằng tất cả các câu lệnh và điều kiện sẽ được thực hiện ít nhất một lần. Người kiểm thử truy nhập vào mã nguồn chương trình và có thể kiểm tra nó, lấy đó làm cơ sở để hỗ trợ việc kiểm thử.

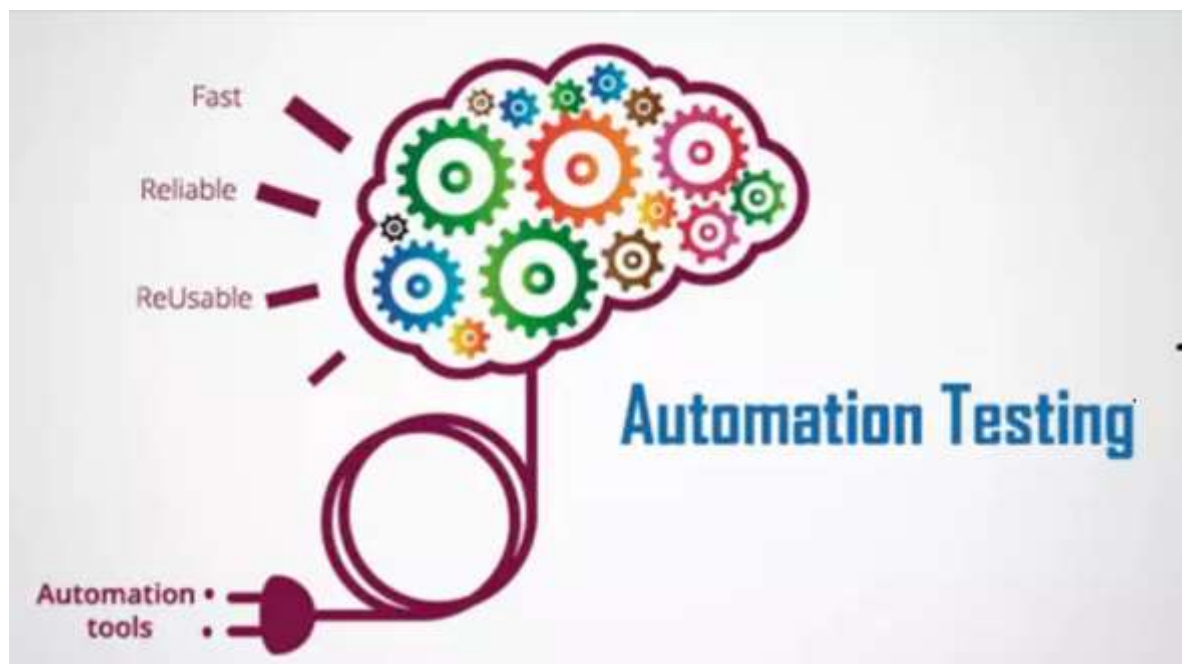
2.2 Kiểm thử tự động phần mềm

Kiểm thử tự động là một kỹ thuật sử dụng một ứng dụng để thực hiện toàn bộ vòng đời của phần mềm trong thời gian ngắn hơn và sử dụng được nhiều lần cho test hồi quy.

Kiểm thử tự động là một kỹ thuật tự động hóa, trong đó người kiểm thử tự viết các tập lệnh và sử dụng phần mềm phù hợp để kiểm thử phần mềm. Về cơ bản nó là một quá trình tự động hóa của một quy trình thủ công. Giống như kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để kiểm tra ứng dụng theo quan điểm load, performance và stress test.

Nói cách khác, kiểm thử tự động sử dụng các công cụ tự động hóa để viết và thực hiện các test case, không cần sự tham gia thủ công trong khi thực hiện một test suite. Thông thường, người kiểm thử viết test script và test case bằng cách sử dụng công cụ tự động hóa và sau đó nhóm thành các test suite.

Mục tiêu chính của kiểm thử tự động là tăng hiệu quả kiểm thử và phát triển giá trị phần mềm.



Quy trình kiểm thử tự động



Bước 1: Phân tích khả năng áp dụng kiểm thử tự động.

Hiển nhiên, chúng ta không thể tự động hoá mọi việc trong kiểm thử phần mềm được. Có những phần mềm mới hay công nghệ viết ra phần mềm mà những công cụ kiểm thử tự động hiện tại chưa hỗ trợ hoặc chỉ hỗ trợ một phần. Ví dụ rõ ràng nhất là khi chúng ta kiểm thử một trang Web trên một trình duyệt mới, và lúc đó, công cụ kiểm thử tự động chưa có phiên bản mới hỗ trợ trên trình duyệt đó. Hay, một ví dụ khác về chương trình SAP, các nhà phát triển SAP đã đưa ra một lựa chọn ngăn chặn việc chạy script tự động trên nó, và để có thể kiểm thử tự động trên SAP, chúng ta cần yêu cầu các nhà phát triển chương trình SAP gỡ bỏ lựa chọn này.

Bước 2: Lựa chọn công cụ kiểm thử tự động thích hợp.

Sau khi xác định được sản phẩm hiện tại có thể làm Kiểm Thử Tự Động hay không, bước kế tiếp, chúng ta cần xác định nên sử dụng công cụ kiểm thử tự động nào. Công cụ nào hỗ trợ kiểm thử tự động cho công nghệ mà sản phẩm sử dụng? Ưu nhược điểm của từng công cụ? Ngôn ngữ kịch bản nào mà công cụ kiểm thử sử dụng? Nhân sự hiện tại có quen thuộc với công cụ đó hay không?

Bước 3: Xây dựng môi trường làm việc.

Môi trường làm việc bao gồm các khái niệm, chu trình, thủ tục và môi trường mà kịch bản kiểm thử tự động được thiết kế và viết ra. Bên cạnh đó, nó cũng nên bao gồm luôn cấu trúc thư mục, lưu trữ các kịch bản kiểm thử cũng như các mối quan hệ logic giữa các thành phần.

Bước 4: Viết kịch bản kiểm thử, thực thi và phân tích kết quả.

Dựa trên các kịch bản kiểm thử đã được tạo ra bằng kiểm thử thủ công, dựa vào ngôn ngữ kịch bản mà công cụ kiểm thử tự động hỗ trợ, chúng ta viết các đoạn mã tương tác với sản phẩm phần mềm trên các môi trường và thực thi nó. Sau khi thực thi các đoạn mã, chúng ta cần phân tích các kết quả đạt được và ghi lại các vấn đề của sản phẩm, nếu có.

Một số công cụ hỗ trợ kiểm thử tự động

Selenium

Selenium là một công cụ kiểm thử phần mềm tự động mã nguồn mở miễn phí cho các ứng dụng web trên nhiều trình duyệt và nền tảng khác nhau như Windows, Mac và Linux. Selenium giúp Tester thực hiện kiểm thử bằng nhiều ngôn ngữ lập trình khác nhau như Java, PHP, C#, Python, Groovy, Ruby và Perl.

Selenium hiện có 3 loại: Selenium Webdriver, Selenium IDE, Selenium Grid. Tùy vào kỹ năng, nền tảng và yêu cầu mà bạn có thể lựa chọn sử dụng loại Selenium phù hợp.

Công cụ này phổ biến với tất cả các trình duyệt nổi tiếng hiện tại như Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari, Opera. Vì vậy, Selenium chắc chắn là nền tảng cho hầu hết các công cụ kiểm thử phần mềm khác.



TestingWhiz

TestingWhiz là công cụ kiểm thử phần mềm tự động với phiên bản Enterprise cung cấp một gói hoàn chỉnh gồm nhiều giải pháp test tự động khác nhau. Trong đó bao gồm: test web, test phần mềm, test database (cơ sở dữ liệu), test API, test ứng dụng di động, bảo trì bộ kiểm tra hồi quy, tối ưu hóa và tự động hóa cũng như kiểm thử trên nhiều trình duyệt.

- Ngoài ra TestingWhiz cung cấp nhiều tính năng quan trọng khác nhau như:
 - Kiểm thử theo hướng từ khóa (key-word driven), theo hướng dữ liệu (data driven) và kiểm thử phân tán (distributed)
 - Kiểm thử tiện ích mở rộng trong trình duyệt
 - Object Eye Internal Recorder
 - SMTP Integration
 - Tích hợp với các công cụ theo dõi lỗi như Jira, Mantis, TFS và FogBugz
 - Tích hợp với các công cụ quản lý kiểm thử như HP Quality Center, Zephyr, TestRail và Microsoft VSTS

- Centralized Object Repository (Kho lưu trữ đối tượng tập trung)
- Version Control System Integration (Tích hợp hệ thống kiểm soát phiên bản)
- Customized Recording Rule (Quy tắc ghi tùy chỉnh)

HPE Unified Functional Testing (HP – UFT formerly QTP)

HPEU cung cấp tính năng tự động hóa kiểm thử để kiểm thử chức năng và kiểm thử hồi quy cho các ứng dụng phần mềm. Ngôn ngữ script Visual Basic Scripting Edition được ứng dụng bởi công cụ này để đăng ký các quá trình kiểm thử và vận hành các đối tượng và điều khiển khác nhau trong việc test các ứng dụng.

- Ngoài ra, QTP cung cấp các tính năng khác như:
 - Tích hợp với Mercury Business Process Testing và Mercury Quality Center
 - Nhận dạng Unique Smart Object
 - Cơ chế xử lý lỗi
 - Tạo các tham số cho đối tượng, checkpoint và bảng điều hướng dữ liệu
 - Tài liệu tự động

TestComplete

TestComplete là một nền tảng kiểm thử chức năng cung cấp các giải pháp khác nhau để tự động kiểm thử. Công cụ này sử dụng cho máy tính để bàn, web và các ứng dụng di động.

- TestComplete cung cấp các tính năng sau:
 - GUI testing
 - Hỗ trợ ngôn ngữ test – JavaScript, Python, VBScript, JScript, DelphiScript, C++ Script & C# Script
 - Kiểm thử trình hiển thị

- Kiểm thử theo script (Scripted testing)
- Kiểm thử ghi và phát lại (Test recording and playback)

Ranorex

Ranorex Studio cung cấp các công cụ tự động hóa testing khác nhau bao gồm việc test tất cả các ứng dụng máy tính để bàn, web và thiết bị di động.

- Cụ thể hơn, Ranorex cung cấp các tính năng sau:

- Kiểm thử GUI
- Có thể tái sử dụng test code
- Phát hiện bug
- Tích hợp với nhiều công cụ khác nhau
- Ghi và phát lại

Sahi

Sahi là một công cụ kiểm thử phần mềm tự động hóa áp dụng cho việc test các ứng dụng web. Mã nguồn mở Sahi được viết bằng ngôn ngữ lập trình Java và JavaScript.

- Sahi cung cấp các tính năng sau:

- Thực hiện kiểm thử nhiều trình duyệt cùng lúc
- Hỗ trợ các framework ExtJS, ZK, Dojo, YUI, v.v.

Apache JMeter

Apache JMeter được dùng để kiểm thử khả năng chịu tải và kiểm thử hiệu năng cho các ứng dụng Web và một số ứng dụng khác. Công cụ này hỗ trợ kiểm thử hiệu năng của các mã nguồn được viết bằng các ngôn ngữ khác nhau như PHP, Java, ASP.NET, . . . Apache JMeter mô phỏng khả năng chịu tải của các máy chủ trên máy sử dụng để kiểm thử hệ thống. Công cụ này hỗ trợ giao diện đồ họa giúp phân

tích tốt hiệu suất khi kiểm thử đồng thời nhiều ca kiểm thử. Ngoài ra, Apache JMeter còn hỗ trợ thêm nhiều tiện ích khác.

Tosca Testsuite

Tosca Testsuite là một công cụ phần mềm để thực hiện tự động kiểm thử phần mềm chức năng và hồi quy. Ngoài chức năng tự động hóa thử nghiệm, TOSCA bao gồm quản lý kiểm thử tích hợp, giao diện người dùng đồ họa (GUI), giao diện dòng lệnh (CLI) và giao diện lập trình ứng dụng (API).

- Tosca Testsuite đi kèm với các tính năng sau:
 - Lập kế hoạch và thiết kế trường hợp thử nghiệm
 - Kiểm tra cung cấp dữ liệu
 - Dịch vụ mạng ảo hóa
 - Kiểm tra ứng dụng di động
 - Quản lý tích hợp
 - Bảo hiểm rủi ro

Telerik TestStudio

Telerik TestStudio cung cấp giải pháp để tự động kiểm thử ứng dụng trên máy tính để bàn, web và thiết bị di động bao gồm kiểm thử giao diện người dùng, load và hiệu suất.

- Công cụ này cung cấp nhiều khả năng tương thích khác nhau như:
 - Hỗ trợ các ngôn ngữ lập trình như HTML, AJAX, ASP.NET, JavaScript, Silverlight, WPF và MVC
 - Tích hợp với Visual Basic Studio 2010 và 2012
 - Ghi và phát lại
 - Kiểm thử trên nhiều trình duyệt
 - Kiểm thử thủ công

- Tích hợp với các công cụ theo dõi bug

Katalon Studio

Katalon Studio là một công cụ kiểm thử phần mềm tự động hóa miễn phí được phát triển bởi Katalon LLC. Công cụ này được xây dựng dựa trên các framework tự động hóa mã nguồn mở Selenium, Appium với giao diện IDE chuyên biệt để kiểm tra API, web và thiết bị di động. Công cụ này bao gồm một gói đầy đủ các tính năng mạnh mẽ giúp dễ dàng tự động hóa kiểm thử giao diện người dùng web.

- Katalon Studio bao gồm các tính năng sau:
 - Kho lưu trữ đối tượng tích hợp, XPath, nhận dạng lại đối tượng
 - Hỗ trợ các ngôn ngữ script Java / Groovy
 - Hỗ trợ tích hợp cho kiểm thử dựa trên hình ảnh
 - Hỗ trợ các công cụ Tích hợp liên tục như Jenkins & TeamCity
 - Hỗ trợ Duel-editor Interface
 - Quy trình thực thi có thể tùy chỉnh

2.3 Kiểm thử chức năng

2.3.1 Kiểm thử chức năng

Định nghĩa

Kiểm thử chức năng là một loại kiểm thử hộp đen (black box) và test case của nó được dựa trên đặc tả của ứng dụng phần mềm/thành phần đang kiểm thử. Các chức năng được kiểm thử bằng cách nhập vào các giá trị nhập và kiểm tra kết quả đầu ra, và ít quan tâm đến cấu trúc bên trong của ứng dụng.

Nó là một quy trình cố gắng tìm ra các khác biệt giữa đặc tả bên ngoài của phần mềm và thực tế mà phần mềm cung cấp. Với các đặc tả bên ngoài của phần mềm là đặc tả chính xác về hành vi của phần mềm theo góc nhìn của người dùng.

Mục đích

Với kiểm thử đơn vị ta phát hiện sự khác biệt giữa đặc tả giao tiếp của đơn vị và thực tế mà đơn vị này cung cấp.

Với kiểm thử hệ thống ta chỉ ra rằng chương trình không tương thích với các mục tiêu ban đầu của nó. Thì:

Với kiểm thử chức năng ta sẽ hoàn thiện nốt phần cần xác minh còn lại là chỉ ra rằng chương trình không tương thích với các đặc tả bên ngoài của nó.

Các lợi ích: Tránh kiểm thử dư thừa. Ngăn chặn sự quan tâm nhiều vào quá nhiều loại lỗi tại từng thời điểm.

Các kỹ thuật thường dùng trong kiểm thử chức năng:

- Kỹ thuật phân lớp tương đương (Equivalence Class Partitioning).
- Kỹ thuật dùng các bảng quyết định (Decision Tables).
- Kỹ thuật kiểm thử các bộ thân kỳ (Pairwise).
- Kỹ thuật phân tích vùng miền (domain analysis).
- Kỹ thuật dựa trên đặc tả Use Case (Use case).

2.4 Hướng kiểm thử tự động

Những Mô hình phổ biến hiện nay

- Tuyến tính
- Hướng module
- Hướng dữ liệu
- Hướng từ khóa
- Hỗn hợp

Tuyến tính – Linear Framework

Đặc điểm cơ bản

- Mọi thứ liên quan đến mã đều được định nghĩa bên trong phương thức kiểm thử
- Không quan tâm đến việc lãng phí và trùng lặp các câu lệnh - Việc record/playback thường xuyên sinh ra mã tuyến tính - Dễ dàng để bắt đầu
- Khó khăn trong việc chỉnh sửa

Mô hình này có thể được dùng trong dự án nhỏ, nơi mà không có quá nhiều màn hình giao diện cũng như chức năng. Mặc khác, chúng ta cũng hay dùng Mô hình này khi sử dụng một công cụ kiểm thử tự động lần đầu tiên. Thông qua việc record/playback và phát sinh mã, chúng ta có thể học được cách công cụ tương tác với ứng dụng như thế nào. Ngoài hai lý do trên, Mô hình này không được khuyến khích sử dụng trong kiểm thử tự động.

Hướng modul – Modularity

Đặc điểm cơ bản

- Các đối tượng được định nghĩa một lần và tái sử dụng trong các phương thức kiểm thử.
- Các phương thức nhỏ và có mục đích được tạo ra cho những chức năng riêng biệt
- Kịch bản kiểm thử tự động là một tập hợp các phương thức nhỏ và các đối tượng được định nghĩa từ trước
- Cho phép chúng ta dễ dàng viết các mã dễ dàng được chỉnh sửa

Mục đích chính của Mô hình này là việc chỉnh sửa dễ dàng. Nếu có bất kỳ thay đổi nào trên giao diện, chúng ta chỉ cần chỉnh sửa trong các phương thức và đối tượng. Mã kiểm thử chính của chúng ta vẫn hoạt động chính xác. Mô hình POM (Page Object Model) – thường được dùng với Selenium – là một dạng của việc ứng dụng Mô hình hướng modul. Toàn bộ trang web sẽ được chia nhỏ thành các trang. Các đối tượng UI của từng trang được định nghĩa bên trong từng lớp của trang đó. Nếu có bất kỳ thay đổi nào trên ứng dụng web, chúng ta chỉ cần chỉnh sửa lớp của

trang đó, những lớp của trang khác vẫn giữ nguyên. Kết quả cuối cùng chúng ta sẽ có những đoạn mã được bảo trì tốt hơn và dễ đọc hơn.

Điểm yếu của mô hình này là nó yêu cầu một mức độ kỹ năng lập trình và hiểu sâu về hướng đối tượng. Nếu bạn có nó, khuôn mẫu này được khuyến khích sử dụng.

Hướng dữ liệu – Data driven

Đặc điểm cơ bản

- Dữ liệu kiểm thử (giá trị đầu vào và đầu ra) được tách khỏi mã nguồn và lưu trong một tập tin bên ngoài. Nó có thể là một tập tin CSV, một bảng Excel hay một cơ sở dữ liệu.

- Khi mã kiểm thử thực thi, các giá trị này được lấy ra từ tập tin, chứa vào biến và thay thế các giá trị cứng (nếu có) trong mã nguồn.

- Thực sự hữu ích khi mà cùng một kịch bản kiểm thử cần thực thi với nhiều dữ liệu đầu vào khác nhau.

Có vài ưu điểm khi áp dụng mô hình này. Tất cả các giá trị kiểm thử được lưu bên ngoài mã nguồn, do đó, bất kỳ thay đổi nào xảy ra trong quá trình phát

triển ứng dụng, chúng ta chỉ cần thay đổi dữ liệu trong tập tin bên ngoài, và mã kiểm thử tự động của chúng ta vẫn được giữ nguyên.

Một ưu điểm khác là, khả năng sử dụng một kịch bản kiểm thử cho nhiều dữ liệu khác nhau. Ví dụ như, bạn đang làm một kịch bản đăng nhập hệ thống với 100 user. Bạn có thể viết 1 đoạn mã và một tập tin lưu trữ thông tin của 100 user. Sau đó, bạn chỉ cần thực thi 1 lần, và đi qua cả 100 bộ dữ liệu. Bạn dễ dàng phát hiện, với kiểu dữ liệu nào thì đoạn mã Fail. Đây cũng là một thế mạnh khi bạn đang làm kiểm thử phủ định – Negative Test.

Hướng từ khóa – Keyword driven

Đặc điểm cơ bản

- Cả dữ liệu và chức năng được định nghĩa bên ngoài mã nguồn - Cần phát triển các từ khóa cho nhiều chức năng khác nhau

- Mã kiểm thử tự động đôi khi được lưu trữ ở một tập tin bên ngoài mã nguồn giống như mô hình hướng dữ liệu. Các bước của kịch bản kiểm thử được viết từng bước với định dạng bảng, nơi mà sử dụng các từ khóa và dữ liệu kiểm thử

- Mã nguồn chính sẽ đọc các bước trong định dạng bảng và thực thi các chức năng tương ứng

- Cho phép các kỹ sư kiểm thử thủ công, nhưng người không biết về lập trình, có thể là một phần, ở một mức độ, của nhóm kiểm thử tự động

Ưu điểm của mô hình hướng từ khóa

Mô hình này rất hữu dụng trong những trường hợp mà kịch bản kiểm thử có quá nhiều thay đổi. Nếu bất kỳ bước nào trong kịch bản kiểm thử bị thay đổi, chúng ta không cần phải chỉnh sửa mã nguồn. Chúng ta chỉ cần chỉnh sửa tập tin bên ngoài và như vậy, kịch bản kiểm thử tự động sẽ được chỉnh sửa theo.

Chúng ta định nghĩa toàn bộ kịch bản ở tập tin và đưa cho kỹ sư kiểm thử thủ công, họ sẽ thêm các đoạn văn bản (text) hoặc chỉnh sửa cái có sẵn. Bằng cách này, kỹ sư kiểm thử tự động cũng có thể trở thành một phần của nhóm kiểm thử tự động bởi vì họ không cần phải lập trình gì cả. Họ chỉ cần chỉnh sửa tập tin ở những vị trí cần thiết và kịch bản kiểm thử tự động sẽ được chỉnh sửa một cách tự động.

Một lợi ích khác của mô hình này là, kịch bản kiểm thử của bạn trở thành một công cụ độc lập. Bạn chỉ cần bảo trì kịch bản kiểm thử trong một tập tin và nếu bạn cần thay đổi công cụ kiểm thử tự động ở điểm nào đó, bạn có thể dễ dàng chỉnh sửa bằng cách viết lại cách đọc và thực thi tập tin với công cụ mới.

Mặc khác, khuyết điểm của mô hình này là, bạn cần phát triển các từ khóa cho các chức năng khác nhau. Trong một dự án lớn, có thể có rất nhiều từ khóa mà bạn cần phải nhớ và tổ chức nó hợp lý. Bản thân việc này có thể sẽ làm một công việc nặng nhọc cho quá trình phát triển kiểm thử tự động.

Ở vài trường hợp phức tạp, khi mà các đối tượng UI không thể được xác định dễ dàng, chúng ta phải sử dụng nhiều kỹ thuật khác nhau để xử lý, mô hình này không hữu dụng cho lắm.

Mô hình hướng từ khóa là một mô hình ưa thích của nhiều kỹ sư kiểm thử tự động. Robot Framework – công cụ kiểm thử tự động được phát triển bởi Google – là một công cụ phổ biến đi theo hướng từ khóa. Những công cụ đi theo hướng từ khóa này còn có Test Architect hay Katalon Studio

Hướng hỗn hợp – Hybrid

Đặc điểm cơ bản

- Là sự kết hợp của hai hoặc nhiều kỹ thuật ở trên, kế thừa thế mạnh và loại bỏ những điểm yếu của các mô hình khác.
- Mô hình này sử dụng cách tiếp cận theo hướng modul, kết hợp với hướng dữ liệu hoặc hướng từ khóa
- Mô hình này có thể dùng mã nguồn để xử lý những công việc đặc biệt mà quá khó để tạo ra với cách làm từ khóa

Một cách đơn giản, mô hình này sử dụng kết hợp nhiều kỹ thuật với nhau. Chúng ta có thể sử dụng hướng dữ liệu đồng thời với hướng modul. Trong vài trường hợp, chúng ta có thể dùng từ khóa song song với modul. Cơ bản, khi nào chúng ta sử dụng nhiều hơn một mô hình, đó là lúc chúng ta sử dụng hỗn hợp – Hybrid >

2.5 Công cụ áp dụng

2.5.1 Công cụ Katalon Studio



Hình 2. 2 Hình ảnh minh họa về công cụ Katalon Studio

Khái niệm: Là một bộ công cụ toàn diện cho kiểm thử tự động hóa ứng dụng trên web và điện thoại di động. Công cụ này bao gồm một gói đầy đủ các tính năng mạnh mẽ giúp vượt qua những thách thức phổ biến trong tự động hóa thử nghiệm giao diện web, ví dụ như popup, iFrame và wait-time. Giải pháp thân thiện và linh hoạt này giúp tester thực hiện công tác kiểm tra tốt hơn, làm việc nhanh hơn và khởi chạy phần mềm chất lượng cao nhờ vào sự thông minh mà nó cung cấp cho toàn bộ quá trình tự động hóa kiểm thử.

Các tính năng chính của Katalon Studio:

Simple deployment: Một gói triển khai duy nhất, gắn kết chứa mọi thứ bạn cần để triển khai một công cụ kiểm tra tự động mạnh mẽ

Quick & easy set-up: Không chỉ cung cấp sự cài đặt đơn giản, Katalon Studio cũng giúp bạn dễ dàng thiết lập môi trường. Tester có thể chạy test script đầu tiên của họ khá nhanh bằng cách sử dụng mẫu được xây dựng trước và các test scripts, chẳng hạn như object repositories và keyword libraries.

Faster & Better results: Tích hợp sẵn mẫu với hướng dẫn rõ ràng giúp tester nhanh chóng xây dựng và chạy các test scripts tự động hóa. Họ có thể thực hiện từng bước với tốc độ và hiệu quả, từ thiết lập dự án, tạo ra thử nghiệm, thực hiện, tạo báo cáo và bảo trì.

Flexible modes: Một tester mới có thể sử dụng recording và keywords để xây dựng các bài kiểm tra tự động hóa, trong khi các chuyên gia kiểm tra có một IDE hoàn chỉnh để xây dựng các kịch bản nâng cao.

Ease of use: Nó không thể được dễ dàng hơn, ngay cả hướng dẫn sử dụng với kinh nghiệm lập trình tối thiểu cũng có thể khai thác lợi ích của nó một cách dễ dàng.

Cross-browser application: Katalon Studio hỗ trợ nhiều nền tảng: Windows 32 và 64 (7, 8 và 10) và OS X 10.5+.

Quy trình làm việc của Katalon Studio:

Điều hành một luồng công việc tuyến tính:

Khởi tạo:

Các mẫu dự án tích hợp: bằng cách cung cấp các mẫu dựng sẵn để tổ chức các trường hợp kiểm thử, kho lưu trữ đối tượng và từ khóa, Katalon Studio giúp việc kiểm thử dễ dàng hơn cho người kiểm tra.

Nhiều khả năng: hỗ trợ đầy đủ kiểm thử cho Web, Android, iOS, và API trên tất cả các hệ điều hành.

Tích hợp trên công cụ không rắc rối: dễ dàng kết hợp với Jenkins, GIT và JIRA với các plug-in gốc.

Triển khai:

Tạo kiểm thử tự động: ghi lại các hành động và tạo các kịch bản tự động bằng các từ khóa tích hợp.

Tập lệnh Hi-end: cho phép xây dựng kịch bản kiểm thử nâng cao hoặc từ khóa tùy chỉnh dễ dàng và hiệu quả.

Chụp đối tượng thông minh: máy ghi âm tiên tiến phát hiện các thuộc tính đối tượng một cách hiệu quả để tối đa hóa nhận dạng.

Vận hành:

Thực hiện kiểm thử mạnh mẽ: chạy các trường hợp kiểm thử hoặc bộ kiểm thử bằng cách sử dụng nhiều cấu hình và bộ dữ liệu.

Tính linh hoạt trong thực thi: cung cấp bảng điều khiển tích hợp CI với các tham số khác nhau để thực hiện từ xa. Chạy thử nghiệm trên nhiều trình duyệt và hệ điều hành OS cục bộ hoặc với Sauce Labs và BrowserStack.

Xử lý lỗi linh hoạt và thực hiện lại tự động: bao gồm các quy tắc thời gian chạy để tự động xử lý các luồng thực thi phức tạp.

Báo cáo:

Báo cáo có sẵn trong một số định dạng: với ghi nhật ký nâng cao, dữ liệu gỡ lỗi và ảnh chụp màn hình.

Báo cáo thực hiện Bespoke: được tích hợp với quy trình thông báo của bạn.

Nhật ký Selen và Appium cải tiến: với các tính năng phân tích được cải tiến để cải thiện chiến lược tự động hóa.

Bảo trì:

Bảo trì đối tượng kiểm thử thông minh: tự động cập nhật tất cả các trường hợp và bộ kiểm thử liên quan khi các đối tượng được thay đổi.

Tổ chức kiểm thử hiệu quả: cho phép dễ dàng quản lý và bảo trì các bài kiểm thử, dữ liệu và từ khóa.

Bảng 2. 1 So sánh katalon studio và selenium

Những điểm chính	Katalon	Selenium
Bản chất	Công cụ tích hợp thư viện Selenium và Appium	Thư viện để tích hợp (Katalon dùng)
Khả năng tùy biến	Giới hạn theo sản phẩm và chi phí	Tùy ý mở rộng với các ngôn ngữ lập trình và nền tảng của hệ thống công ty.
Nền tảng	Đa nền tảng	Đa nền tảng
Ứng dụng	Windows desktop, Web apps, Mobile apps, API/Web service	Web apps
Ngôn ngữ lập trình	Java/Groovy	Java, C#, Python, JavaScript, Ruby, PHP, Perl, VBA

Những điểm chính	Katalon	Selenium
Thời gian tạo test script	Nhanh	Chậm
Cài đặt và sử dụng	Dễ dàng cài đặt và chạy	Yêu cầu cài đặt và tích hợp các công cụ khác nhau (môi trường, IDE,...)
Test hình ảnh	Hỗ trợ tích hợp	Yêu cầu cài đặt thêm thư viện bổ sung
Tích hợp DevOps / ALM	Có	Không (yêu cầu thư viện bổ sung)
CI (tích hợp hệ thống liên tục)	Có Các công cụ CI (ví dụ: Jenkins, Teamcity)	Có Các công cụ CI (ví dụ: Jenkins, Cruise Control)
Kỹ năng lập trình	Không yêu cầu. Đối với các kịch bản nâng cao mới yêu cầu có kỹ năng lập trình	Cần có kỹ năng lập trình ở mức nâng cao để tích hợp các công cụ khác nhau
Phân tích kết quả kiểm tra	Katalon TestOps	Không (phải dùng thư viện report riêng)
Hỗ trợ sản phẩm	Cộng đồng, dịch vụ hỗ trợ doanh nghiệp	Cộng đồng nguồn mở

Những điểm chính	Katalon	Selenium
Kiểu giấy phép	Mất phí	Nguồn mở (Apache 2.0)
Giá cả	Katalon Studio có bản miễn phí và trả phí. Bản tính phí là Enterprise	Miễn phí

2.5.2 Công cụ Jmeter

Khái niệm:



Hình 2. 3 Hình ảnh minh họa công cụ Jmeter

Jmeter là ứng dụng viết trên mã nguồn mở (open source), được viết 100% trên java và là ứng dụng dành cho việc test tải (load test) tốc độ thực thi của ứng dụng web. Ban đầu nó được thiết kế để đơn thuần kiểm tra cho ứng dụng web, sau đã được mở rộng ra nhiều chức năng khác.

Apache JMeter có thể được sử dụng để kiểm tra hiệu suất cả trên các tài nguyên tĩnh, động và các ứng dụng Web. Nó có thể được sử dụng để mô phỏng một lượng người dùng ảo, request lớn trên một máy chủ, nhóm máy chủ, mạng hoặc đối tượng để kiểm tra về độ tải của nó hoặc để phân tích thời gian phản hồi tổng thể dưới các loại tải khác nhau.

Apache jmeter bao gồm các tính năng:

Khả năng và thử nghiệm hiệu suất nhiều ứng dụng, server và protocol khác nhau:

Web -HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET)

SOAP/REST Web Services

FTP

LDAP

Database via JDBC

Message-oriented middleware (MOM) via JMS

Mail -SMTP(s), POP3(S) and IMAP(S)

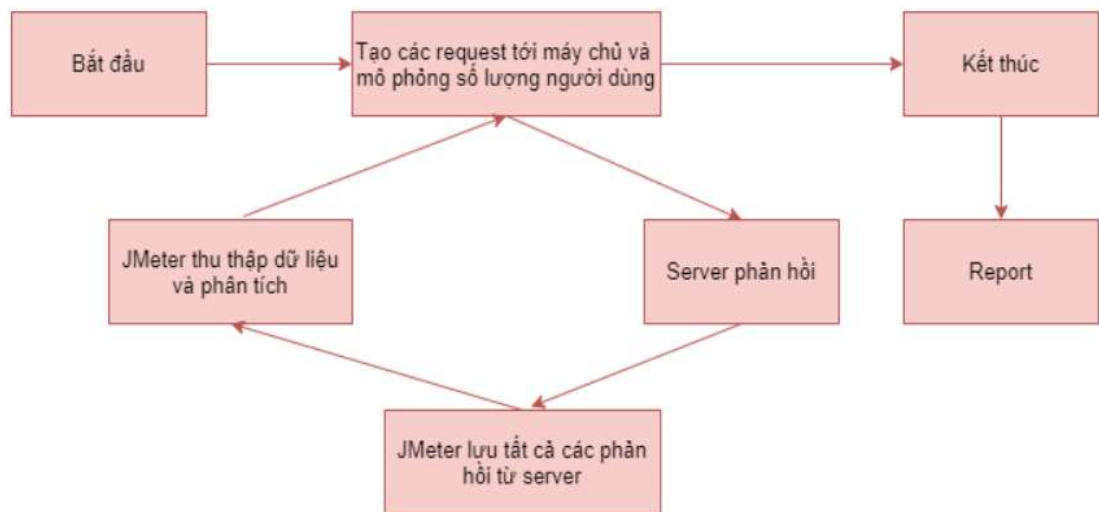
Native commands or shell scripts

TCP

Java Objects

Quy trình làm việc của Jmeter

Khi chúng ta bắt đầu tải hoặc thực hiện kiểm tra một ứng dụng, JMeter tạo các yêu cầu đến máy chủ mục tiêu và mô phỏng số lượng người dùng gửi yêu cầu đến máy chủ đích. Ngay khi máy chủ bắt đầu phản hồi các yêu cầu, JMeter bắt đầu lưu tất cả các phản hồi. Trên cơ sở dữ liệu / phản hồi JMeter tập hợp dữ liệu để tính toán thông tin thống kê. Cuối cùng, bằng cách sử dụng thông tin thống kê này, JMeter chuẩn bị một báo cáo cho biết về hiệu suất của quá trình kiểm tra.



Hình 2. 4 Quy trình làm việc của công cụ Jmeter

Ưu điểm của Jmeter



Hình 2. 5 Ưu điểm của công cụ Jmeter

Mã nguồn mở: Jmeter là một phần mềm mã nguồn mở. Điều này có nghĩa là nó có thể được tải xuống miễn phí. Nó cũng là một ứng dụng Java thuần túy 100%. Nhà phát triển có thể sử dụng mã nguồn của nó, có thể sửa đổi và tùy chỉnh nó theo yêu cầu của họ. Họ cũng có thể đóng góp code của họ để làm nên một JMeter tốt hơn.

Dễ sử dụng: Người dùng có thể cài đặt và sử dụng JMeter một cách dễ dàng. Chỉ cần tải về từ internet, cài đặt và chạy. Như một ứng dụng Java thuần túy, nó sẵn sàng để sử dụng với các cài đặt mặc định. Nó không yêu cầu bạn phải có bất kỳ kỹ năng cụ thể nào hoặc kiến thức nền tảng để sử dụng nó.

Nền tảng độc lập: JMeter được phát triển bằng Java, đây là ngôn ngữ lập trình phổ biến nhất trên thế giới. Do đó, nó có thể chạy trong mọi hệ điều hành có thể là Window, Linux hoặc Mac.

Báo cáo mạnh mẽ: JMeter có thể tạo báo cáo hiệu quả. Kết quả kiểm tra có thể được xem lại bằng cách sử dụng Graph, Chart, and Tree View. Jmeter hỗ trợ các định dạng khác nhau của báo cáo như text, XML, HTML and JSON.

Thử nghiệm cuối cùng: Với Jmeter, người dùng có thể thực hiện bất kỳ loại kiểm thử nào mà bạn muốn. Load Test, Stress Test, Functional Test, Distributed Test, tất cả trong một công cụ

Tính linh hoạt: Bạn có thể tùy chỉnh JMeter theo yêu cầu của bạn và áp dụng thử nghiệm tự động cho JMeter. Bạn có thể tiết kiệm công sức của việc thực hiện các trường hợp kiểm tra thủ công.

Hỗ trợ đa giao thức: JMeter hỗ trợ một vài giao thức như HTTP, FTP, SOAP, JDBC, JMS và LDAP. Nó cũng có thể được sử dụng để kiểm thử hiệu suất của cơ sở dữ liệu của bạn.

Nhược điểm của Jmeter

Tiêu thụ bộ nhớ: JMeter có thể mô phỏng tải nặng và trực quan hóa báo cáo thử nghiệm. Điều này có thể tiêu tốn rất nhiều bộ nhớ và có thể dẫn ra khỏi bộ nhớ dưới tải nặng.

Chỉ áp dụng cho ứng dụng web: JMeter là công cụ tốt để thử nghiệm ứng dụng web nhưng nó không phải là công cụ phù hợp để thử nghiệm ứng dụng máy tính để bàn.

Thiếu hỗ trợ cho JavaScript: JMeter không phải là một trình duyệt, vì vậy nó không thể chạy JavaScript trong ứng dụng web. Nó có hỗ trợ hạn chế để xử lý JavaScript hoặc Ajax, điều này có thể ảnh hưởng đến độ chính xác của mô phỏng.

Bảng 2. 2 So sánh Jmeter và HP LoadRunner

Những điểm chính	Apache Jmeter	HP LoadRunner
Tính khả dụng và thực hiện dễ dàng	Jmeter là một công cụ kiểm thử ứng dụng mã nguồn mở có sẵn. Bạn tải về java, Jmeter được sử dụng để tải lên kịch bản Jmeter của bạn và chạy kiểm thử.	LoadRunner phải thanh toán và đắt đỏ. Việc thực hiện có thể phức tạp bởi vì nó tạo trên chủ đề cho mỗi người dùng.
Thế hệ tải	Jmeter có thể tạo ra tải không giới hạn cho kiểm thử tải chính xác.	LoadRunner có thể tạo ra một lượng tải hạn chế cho kiểm thử tải.
Phân tích	Kết quả được tạo ra bởi Jmeter đơn giản và dễ dàng để hiểu cho những kỹ sư kiểm thử ít kinh nghiệm.	Mặt khác, LoadRunner tạo ra báo cáo thô và được phân tích bởi những kỹ sư riêng
Cộng đồng hỗ trợ	Jmeter có một cộng đồng lớn và hỗ trợ vấn đề phải đối mặt bởi người dùng.	LoadRunner bị giới hạn người dùng do đó cộng đồng của nó nhỏ hơn.

Yêu cầu kỹ năng	Jmeter có thể được sử dụng mà không cần biết một từ nào về mã hóa và kịch bản.	Mặt khác LoadRunner yêu cầu hiểu biết kịch bản chính xác để chạy kiểm thử hiệu suất của nó.
Bổ sung các yếu tố	Các yếu tố có thể được thêm dễ dàng hoặc loại bỏ từ Jmeter.	Trái lại, LoadRunner phức tạp hơn nhiều để tinh chỉnh một số yếu tố trong kịch bản của nó.

CHƯƠNG 3: ĐẶC TẢ HỆ THỐNG PHẦN MỀM

3.1 Giới thiệu về hệ thống phần mềm

Để đáp ứng yêu cầu người dùng, hệ thống website thời trang nữ Vascara bao gồm 2 phân hệ chính: phân hệ cho người quản trị để quản trị nội dung và quản trị hoạt động kinh doanh online; phân hệ cho người dùng để cho khách hàng có thể xem thông tin về các mặt hàng kinh doanh và đặt hàng online.

Trong Đồ án này, em sẽ đi sâu hơn về kiểm thử phân hệ cho người dùng.

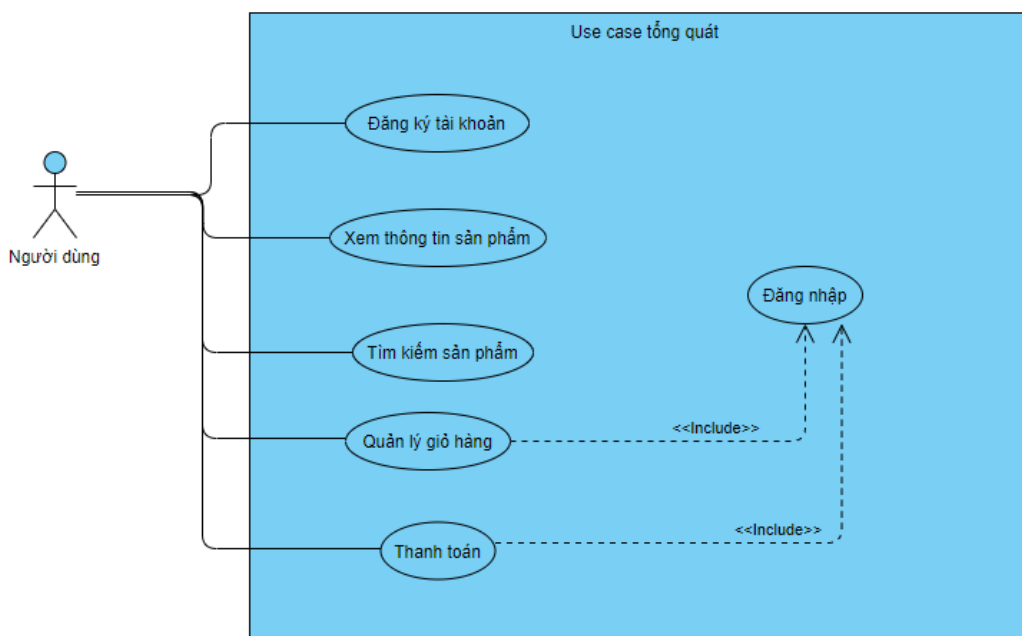
❖ Hệ thống website bán hàng của cửa hàng cần đạt được những mục tiêu sau:

1. Xây dựng giao diện sinh động, dễ sử dụng.
2. Hệ thống cho phép tìm kiếm thông tin các sản phẩm.
3. Hệ thống cho phép tài khoản đăng ký, đăng nhập.
4. Hệ thống cho phép khách hàng quản lý giỏ hàng của mình.
5. Hệ thống cho phép khách hàng thanh toán trước qua tài khoản điện tử.

3.2 Các yêu cầu chức năng

a) Chức năng của phân hệ người dùng (nếu có)

Các chức năng chính của hệ thống được mô tả như sơ đồ dưới đây.



Hình 3. 1 Use case tổng quát

Use case được mô tả như bảng dưới:

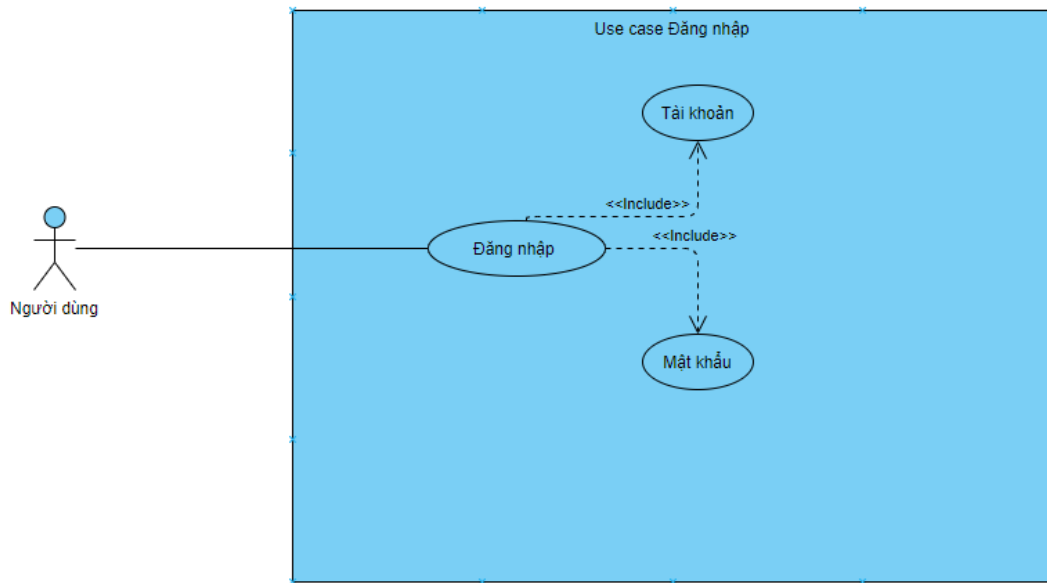
Bảng 3. 1 Bảng mô tả use case

STT	Use case	Mô tả
1	Đăng nhập	Use case này mô tả chức năng đăng nhập vào hệ thống của người dùng.
2	Đăng ký tài khoản	Use case này mô tả chức năng đăng ký tài khoản làm thành viên của người dùng.
3	Quản lý giỏ hàng	Use case mô tả chức năng thêm, sửa, xoá sản phẩm trong giỏ hàng.
4	Thanh toán	Use case này mô tả chức năng thanh toán sản phẩm của người dùng.
5	Xem thông tin Sản phẩm	Use case này mô tả chức năng xem thông tin sản phẩm có trên website của người dùng.

6	Tìm kiếm Sản phẩm	Use case này mô tả chức năng tìm kiếm các sản phẩm trên website của người dùng.
---	-------------------	---------------------------------------------------------------------------------

3.2.2 Chức năng đăng nhập

- Biểu đồ use case



Hình 3. 2 Use case đăng nhập

- Tóm tắt
 - Tác nhân: Người dùng.
 - Mô tả: Người dùng đăng nhập vào hệ thống.
 - Điều kiện cần: Người dùng phải nhập email và mật khẩu chính xác.
 - Điều kiện đủ: Không
- Dòng sự kiện
 - Dòng sự kiện chính:*
 - Người dùng nhập email và mật khẩu của mình
 - Click nút “Đăng nhập”
 - Hệ thống kiểm tra dữ liệu các thông tin đã nhập. Nếu đúng sẽ lưu vào cookie thông tin về người dùng và hiển thị thông tin tài khoản.
 - Dòng sự kiện phụ:*
 - Nếu thông tin nhập sai sẽ thông báo cho người dùng yêu cầu nhập lại.
- Yêu cầu đặc biệt: không.
- Mô tả chức năng

THÔNG TIN CÁ NHÂN

Email: *

Mật khẩu: *

[Bạn quên mật khẩu](#)

Đăng nhập

Hình 3. 3 Màn hình đăng nhập hệ thống

Chức năng: [Đăng nhập]: cho phép đăng nhập tài khoản vào hệ thống.

Validate:

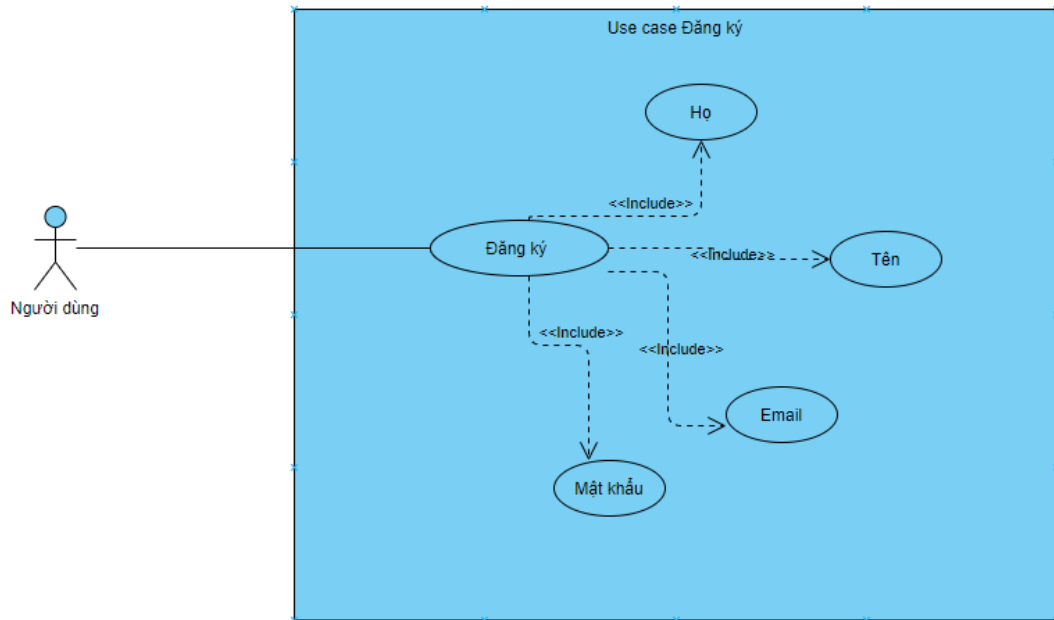
- Email: length(50), khác rỗng , định dạng
- Mật khẩu: kiểu mật khẩu(“*****”), length(4-20), khác rỗng

Chức năng: Người dùng nhập thông tin vào các ô nhập liệu và nhấn nút [Đăng nhập]

- Nếu có trường không hợp lệ về validate: hiện thông báo không hợp lệ.
- Nếu [Email] không tồn tại: hiện label với nội dung: “Email hoặc mật khẩu không chính xác!”.
- Nếu [Email] tồn tại nhưng mật khẩu sai, hiện label với nội dung: “Email hoặc mật khẩu không chính xác!”
- Nếu [Đăng nhập] thành công, hiển thị ra đăng nhập thành công với thông tin tài khoản.
- Khi đăng nhập, người dùng sẽ nhập mật khẩu và phần mật khẩu sẽ được mã hóa (MD5) rồi kiểm tra với CSDL.

3.2.3 Đăng ký

Biểu đồ Use case



Hình 3. 4 Use case Đăng ký

- Tóm tắt
 - Tác nhân: Người dùng
 - Mô tả: Người dùng đăng ký tài khoản trên hệ thống.
 - Điều kiện cần: Người dùng cần phải nhập đầy đủ và chính xác thông tin.
 - Điều kiện đủ: Không
- Dòng sự kiện
 - Dòng sự kiện chính:*
 - Khách hàng cần phải nhập họ tên, email, mật khẩu, số điện thoại Click nút “Đăng ký”
 - Hệ thống kiểm tra dữ liệu các thông tin đã nhập. Nếu đúng sẽ lưu vào cookie thông tin về khách hàng.
 - Dòng sự kiện phụ:*
 - Nếu thông tin nhập không hợp lệ sẽ thông báo cho người dùng yêu cầu nhập lại.
- Yêu cầu đặc biệt: không.

Hình 3. 5 Màn hình đăng ký tài khoản

Các chức năng: [Đăng ký]

- Chức năng [Đăng ký]: cho phép đăng ký tài khoản người dùng (tài khoản khách hàng) để có thể đăng nhập vào hệ thống.
- Validate:
 - Họ và Tên: length(1-50), khác rỗng
 - Email: length(50), kiểu email (“xyz@gmail.com”)
 - Điện thoại: length(3-32), kiểu number, khác rỗng
 - Mật khẩu: kiểu mật khẩu(“*****”), length(4-20), khác rỗng
- Chức năng: Người dùng nhập thông tin vào các ô nhập liệu và nhấn nút [Tiếp tục]
 - Nếu có trường không hợp lệ về validate: hiện label màu đỏ bên dưới ô nhập liệu.
 - Nếu [Email] đã có người sử dụng để đăng ký: hiện label màu đỏ bên cạnh ô nhập liệu với nội dung: “Email đã có người sử dụng”.
 - Khi đăng ký thành công: hệ thống sẽ hiện thị trang với nội dung “Tài khoản của bạn đã được tạo!”
 - Mật khẩu sau khi đăng kí sẽ được mã hóa bằng MD5 và lưu mật khẩu sau khi mã hóa vào CSDL.

3.2.4 Chức năng tìm kiếm

❖ Màn hình hiển thị



Hình 3. 6 Màn hình hiển thị chức năng [Tìm kiếm]

❖ Mô tả chức năng

Khi muốn tìm kiếm một mặt hàng, khách hàng sẽ nhập thông tin tương đối hoặc tuyệt đối vào trường tìm kiếm rồi nhấn tìm kiếm.

❖ Yêu cầu validate

Bảng 3. 2 Yêu cầu validate của chức năng [Tìm kiếm]

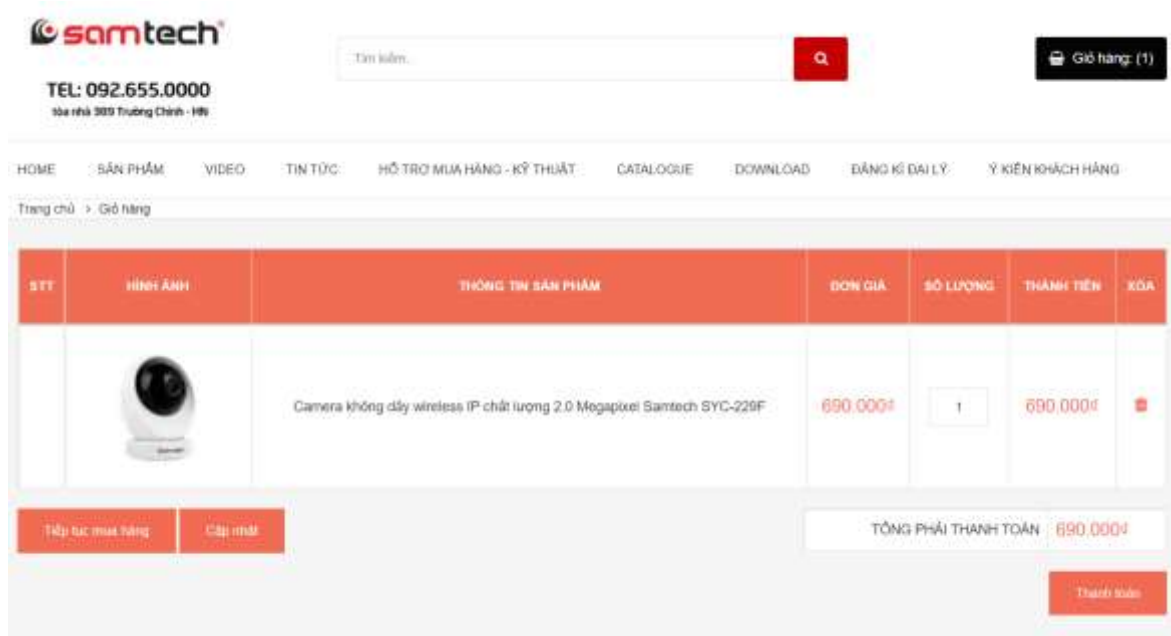
Tên trường	Mô tả	Kiểu	Độ dài tối đa	Bắt buộc
Tìm kiếm	Gồm thông tin cần tìm kiếm	Textbox		

Bảng 3. 3 Nội dung thông báo của chức năng [Tìm kiếm]

STT	Tình huống	Thông báo
1	Tìm kiếm tương đối	Hiển thị ra tất cả sản phẩm liên quan
2	Tìm kiếm tuyệt đối	Hiển thị ra sản phẩm đúng chính xác

3.2.5 Chức năng quản lý giỏ hàng

❖ Màn hình hiển thị



Hình 3. 7 Màn hình hiển thị chức năng [Quản lý giỏ hàng]

❖ Mô tả chức năng

Khi người dùng muốn xem, xóa, cập nhật giỏ hàng, thì người dùng có thể bấm vào giỏ hàng.

❖ Yêu cầu validate

Không có validate

❖ Nội dung thông báo

Bảng 3. 4 Nội dung thông báo của chức năng [Quản lý giỏ hàng]

STT	Tình huống	Thông báo
1	Khi chọn 2 sản phẩm giống nhau	Hiển thị 1 sản phẩm với số lượng 2
2	Khi chọn 2 sản phẩm khác nhau.	Hiển thị 2 sản phẩm với số lượng 1
3	Khi không có sản phẩm nào trong giỏ hàng	Hiển thị 0 sản phẩm

3.2.6 Chức năng đặt hàng

❖ Màn hình hiển thị

The screenshot shows the checkout page for a guest user. The page is divided into three main sections: 'Thông tin nhận hàng' (Shipping Information), 'Vận chuyển' (Shipping), and 'Đơn hàng (1 sản phẩm)' (Order (1 product)).

Thông tin nhận hàng: This section contains several input fields for the user's details. The email field is pre-filled with 'doanhuong@gmail.com'. Other fields include 'Họ và tên' (Last Name) with 'Doãn Hương Ly', 'Số điện thoại' (Phone Number) with '0965646298', 'Địa chỉ' (Address) with 'TX Mỹ Hòa- Hưng Yên', 'Tỉnh thành' (Province/City) with 'Hưng Yên', 'Quận huyện (tùy chọn)' (District/City (optional)) with 'Huyện Mỹ Hòa', and 'Ghi chú (tùy chọn)' (Note (optional)).

Vận chuyển: This section has a red error message: 'Khu vực không hỗ trợ vận chuyển' (Area not supported for shipping). Below this is the 'Thanh toán' (Payment) section, which shows 'Thanh toán khi giao hàng (COD)' (Payment on delivery (COD)) as the selected option.

Đơn hàng (1 sản phẩm): This section displays the product being ordered: 'Camera không dây wireless IP chất lượng 2.0 Megapixel Samtech SYC-229F' with a price of 690,000. It includes a 'Nhập mã giảm giá' (Enter discount code) field and an 'Áp dụng' (Apply) button. Below this, a summary table shows: 'Tạm tính' (Subtotal) 690,000, 'Phí vận chuyển' (Shipping fee) 0, and 'Tổng cộng' (Total) 690,000. At the bottom, there is a 'Quay về giỏ hàng' (Return to cart) link and a 'ĐẶT HÀNG' (Place Order) button.

Hình 3. 8 Màn hình hiển thị giỏ hàng khi chưa đăng nhập

The screenshot shows the checkout page for a logged-in user. The page layout is similar to the guest user view, but with some differences in the 'Thông tin nhận hàng' and 'Vận chuyển' sections.

Thông tin nhận hàng: This section contains several input fields for the user's details. The email field is pre-filled with 'lye@gmail.com'. Other fields include 'Họ và tên' (Last Name) with 'Doãn Minh', 'Số điện thoại' (Phone Number) with '0965646298', 'Địa chỉ' (Address) with 'TX Mỹ Hòa- Hưng Yên', 'Tỉnh thành' (Province/City) with 'Hưng Yên', 'Quận huyện (tùy chọn)' (District/City (optional)) with 'Huyện Mỹ Hòa', and 'Ghi chú (tùy chọn)' (Note (optional)).

Vận chuyển: This section has a blue message: 'Vui lòng nhập thông tin giao hàng' (Please enter shipping information). Below this is the 'Thanh toán' (Payment) section, which shows 'Thanh toán khi giao hàng (COD)' (Payment on delivery (COD)) as the selected option.

Đơn hàng (1 sản phẩm): This section displays the product being ordered: 'Camera wifi 1080P (MOU - 60 phút gọi Free HD 2.0 Megapixel)' with a price of 590,000. It includes a 'Nhập mã giảm giá' (Enter discount code) field and an 'Áp dụng' (Apply) button. Below this, a summary table shows: 'Tạm tính' (Subtotal) 590,000, 'Phí vận chuyển' (Shipping fee) 0, and 'Tổng cộng' (Total) 590,000. At the bottom, there is a 'Quay về giỏ hàng' (Return to cart) link and a 'ĐẶT HÀNG' (Place Order) button.

Hình 3. 9 Màn hình hiển thị giỏ hàng khi đã đăng nhập

❖ *Mô tả chức năng*

Khi người dùng muốn đặt hàng, thì người dùng sẽ sang trang thanh toán. Trang này cho phép người dùng chỉnh sửa địa chỉ, tên, số điện thoại. Và yêu cầu đăng nhập tài khoản.

❖ *Yêu cầu validate*

Bảng 3. 5 Yêu cầu validate của chức năng [Đặt hàng]

Tên trường	Mô tả	Kiểu	Độ dài tối đa	Bắt buộc
Họ và tên	Không rỗng, độ dài từ 3-100 kí tự chữ. Giá trị mặc định là:nhập tên khách hàng	Textbox	100	Y
Số điện thoại	Gồm 10 số.	Textbox	10	y
Tên email	Bao gồm cả kí tự đặc biệt, từ 10-50 kí tự. Giá trị mặc định là:Nhập email	Textbox	50	
Mật khẩu	Lớn hơn 6 kí tự và không đc rỗng	Textbox		y
Địa chỉ	Địa chỉ của khách hàng từ 1-255 kí tự. Giá trị mặc định là:địa chỉ khách hàng	Textbox		Y
Ghi chú	Mô tả về hóa đơn của khách hàng.	Texbox		
Phương thức thanh toán	Phương thức để thanh toán	Radio button		y

Bảng 3. 6 Nội dung thông báo của chức năng [Đặt hàng]

STT	Tình huống	Thông báo
1	Chưa nhập đủ thông tin	Thông báo thiếu thông tin
2	Nhập sai thông tin	Thông báo sai thông tin

3.3 Các yêu cầu phi chức năng

Performace

Yêu cầu thời gian mở website con bất kỳ không được chậm hơn 3 giây, và xử lý lưu thông tin không được chậm hơn 4 giây.

CHƯƠNG 4: TRIỂN KHAI KIỂM THỬ TỰ ĐỘNG

4.1 Thiết kế các yêu cầu kiểm thử

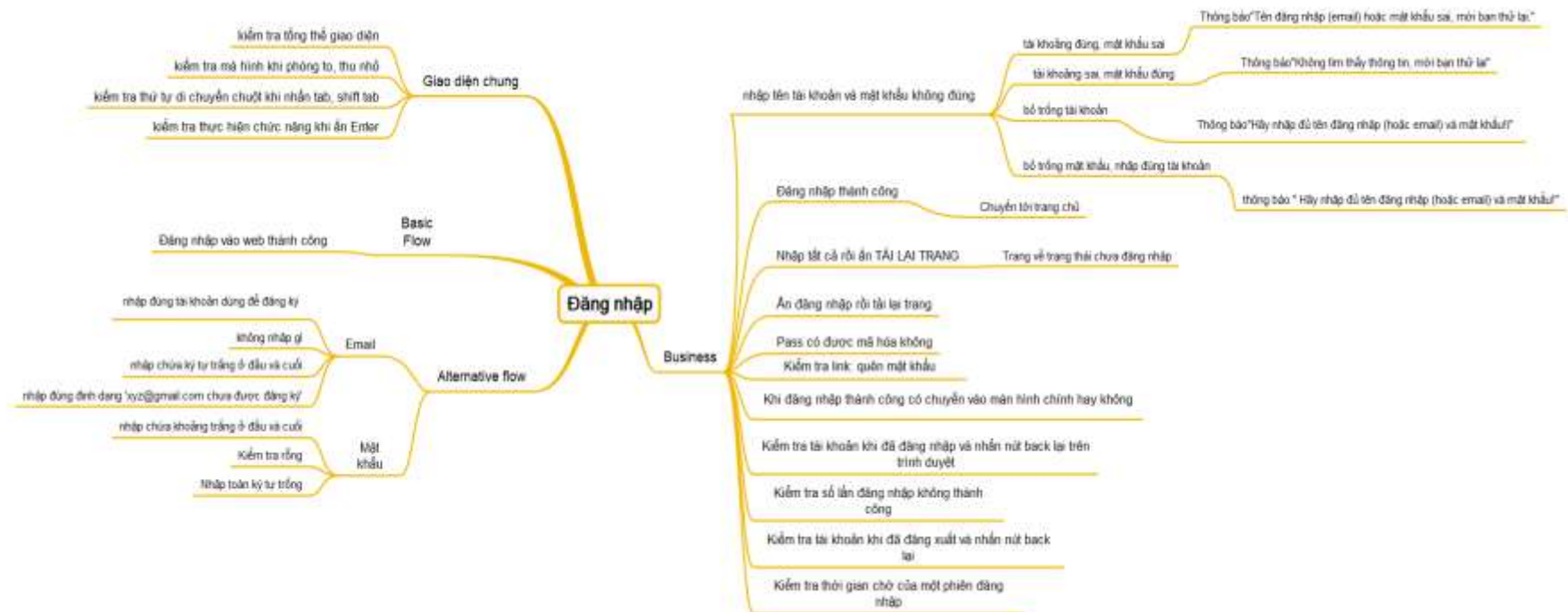
4.1.1 Thiết kế kiểm thử tổng quan hệ thống



Hình 4. 1 Test design “Tổng quát”

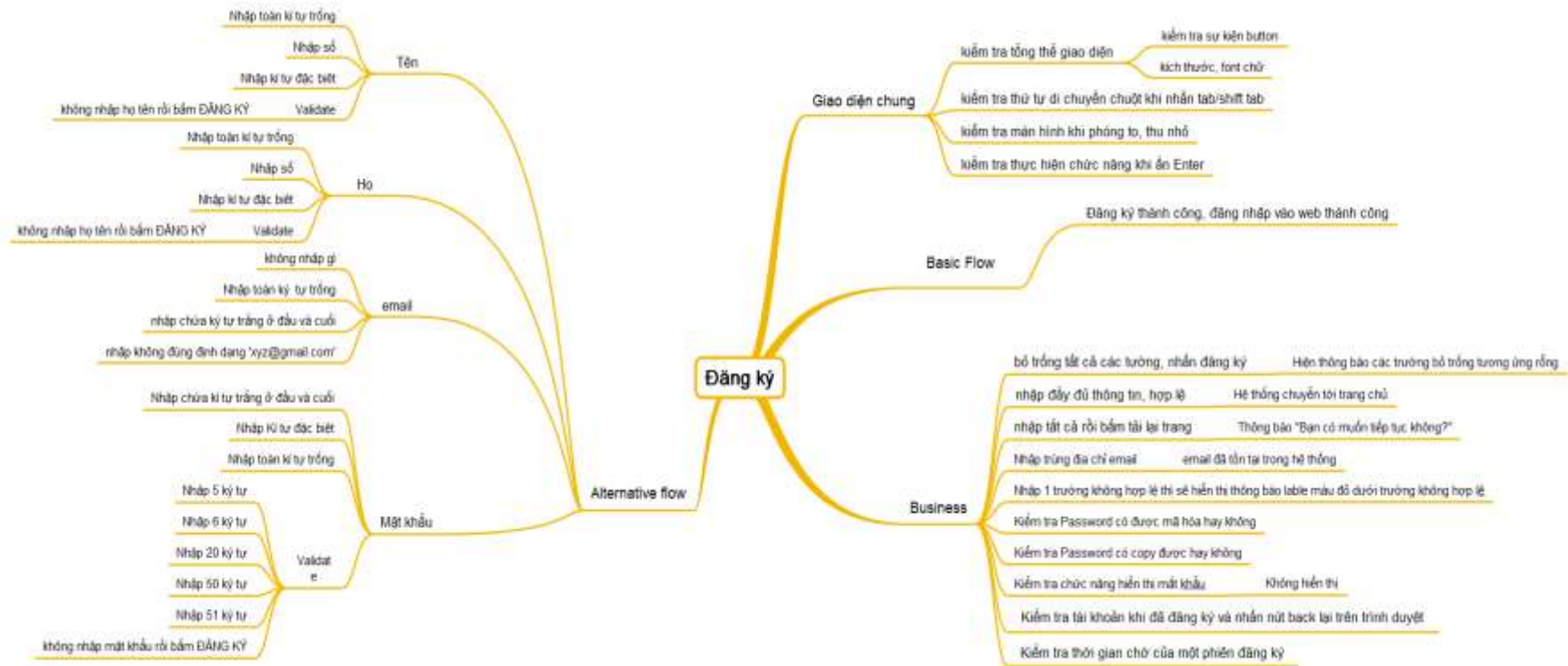
4.1.2 Thiết kế kiểm thử chức năng

Thiết kế kiểm thử chức năng “Đăng nhập hệ thống”



Hình 4. 2 Test design “Đăng nhập”

Thiết kế kiểm thử chức năng “Đăng ký”



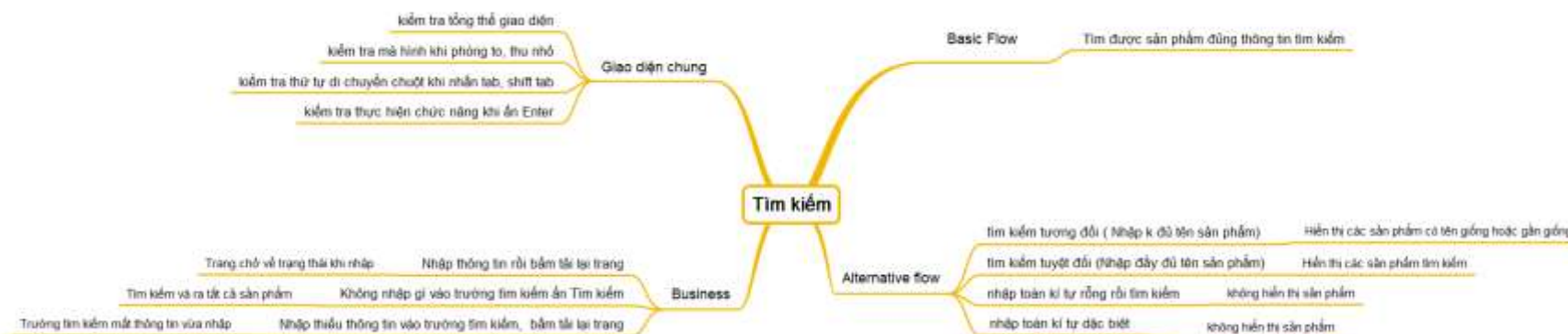
Hình 4. 3 Test design “Đăng ký”

Thiết kế kiểm thử chức năng “Giỏ hàng”



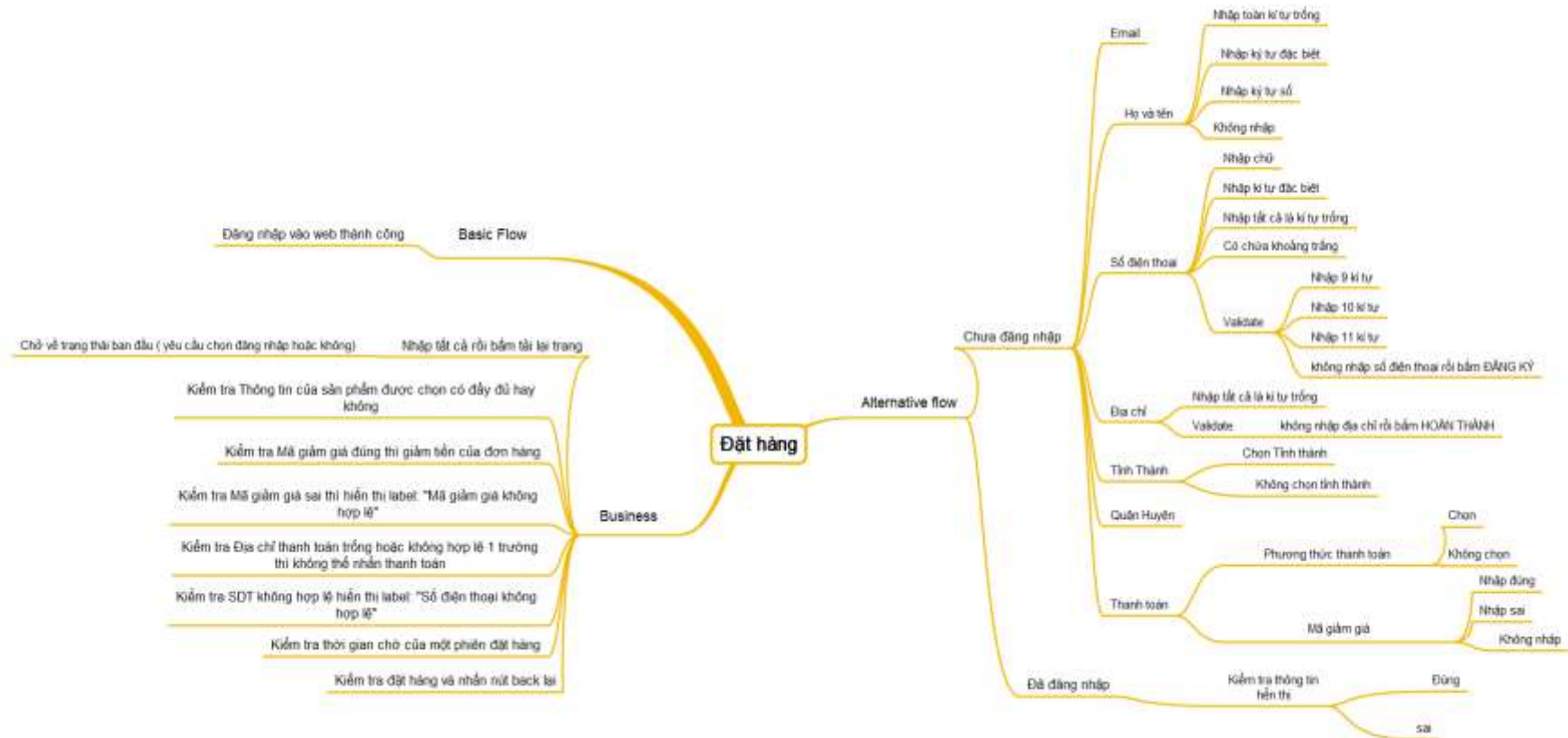
Hình 4. 4 Test design “Quản lý Giỏ hàng”

Thiết kế kiểm thử chức năng “Tìm kiếm”

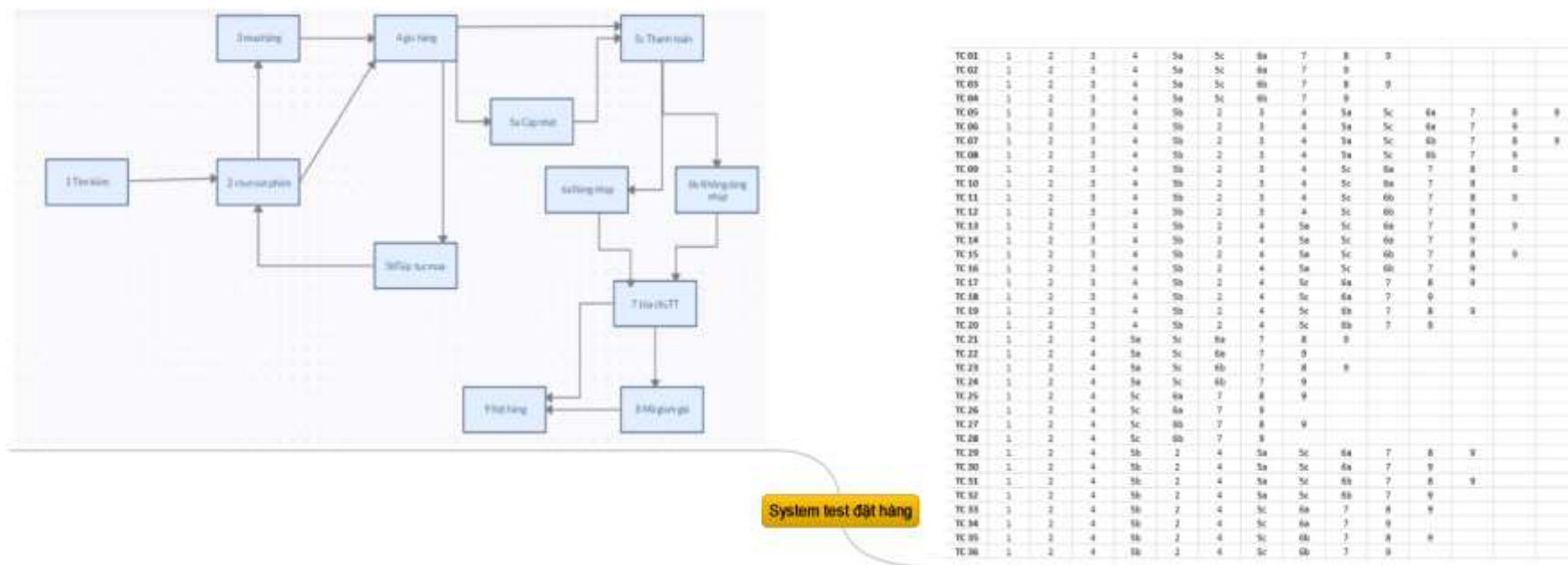


Hình 4. 5 Test design “Tìm kiếm”

Thiết kế kiểm thử chức năng “Đặt hàng”

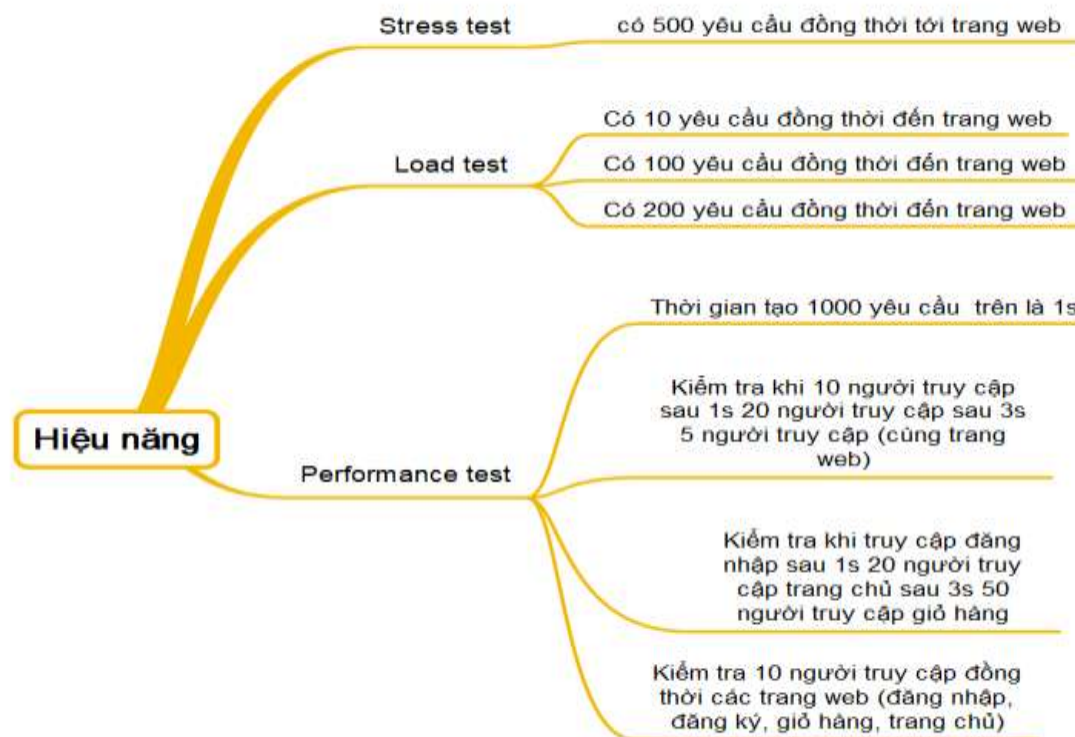


Hình 4. 6 Test design “Đặt hàng”



Hình 4. 7 Test design “System Test Đặt hàng”

4.1.3 Thiết kế kiểm thử hiệu năng



Hình 4. 8 Test design “Hiệu năng”

4.2 Xây dựng ca kiểm thử

4.2.1 Test Case Đăng Nhập

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
Đăng nhập				
DN-1	Kiểm tra giá trị trường trường email		1. Nhập email đúng tài khoản dùng để đăng ký 2. Nhập trường còn lại đúng quy tắc 3. Ấn ĐĂNG NHẬP	3. Đăng nhập thành công, load ra màn thông tin tài khoản
DN-2			1. Không nhập email 2. Nhập trường còn lại đúng quy tắc 3. Ấn ĐĂNG NHẬP	1. Hiện thị thông báo " Vui lòng nhập Email".
DN-3			1. Nhập email chứa ký tự trắng ở đầu và cuối 2. Nhập trường còn lại đúng quy tắc 3. Ấn ĐĂNG NHẬP	3. Đăng nhập thành công, ra màn thông tin tài khoản

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DN-4			1.Nhập email đúng định dạng 'xyz@gmail.com' nhưng chưa đăng ký 2. Nhập trường còn lại đúng quy tắc 3. Ấn ĐĂNG NHẬP	3. Hiện thị thông báo " Thông tin đăng nhập không chính xác".
DN-5	Kiểm tra giá trị trường MẬT KHẨU		1. Nhập MẬT KHẨU chứa kí tự trắng 2. Nhập trường còn lại đúng quy tắc 3. Ấn ĐĂNG NHẬP	3. Hiện thị thông báo " Thông tin đăng nhập không chính xác".
DN-6			1. Không nhập MẬT KHẨU 2. Nhập trường còn lại đúng quy tắc 3. Ấn ĐĂNG NHẬP	3. Hiện thị thông báo " Vui lòng nhập Mật khẩu".
DN-7	Business		1.Nhập tất cả các trường 2. Bấm tải lại trang	2. Các trường trống dữ liệu
DN-8			1.Nhập trường Email sai 2. Nhập Trường MẬT KHẨU đúng 3. Ấn ĐĂNG NHẬP	3. Thông báo "Thông tin đăng nhập không chính xác".

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DN-9			1. Nhập trường Email đúng 2. Nhập Trường MẬT KHẨU sai 3. Ấn ĐĂNG NHẬP	3. Thông báo "Mật khẩu không đúng".
DN-10			1. Đăng nhập 2. Ấn tải lại trang	3. load ra màn trang chủ
DN-11			1. Nhập MẬT KHẨU 2. Kiểm tra mã hóa	3. MẬT KHẨU được mã hóa
DN-12			1. Click Quên mật khẩu	3. load ra màn Quên mật khẩu
DN-13			1. Đăng nhập 2. Ấn nút back	3. Hiện thị màn đăng nhập không có dữ liệu đã nhập
DN-14			1. Đăng xuất 2. Ấn nút back	3. Hiện thị màn thông tin tài khoản

4.2.2 Test Case Đăng Nhập

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
Đăng ký				
DK-1	Kiểm tra giá trị trường HỌ		1. Nhập HỌ toàn kí tự trống 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Vui lòng nhập họ"
DK-2			1. Nhập HỌ toàn kí đặc biệt 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Họ không bao gồm kí tự đặc biệt và số!!!"
DK-3			1. Nhập HỌ toàn kí tự số 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Họ không bao gồm kí tự đặc biệt và số!!!"
DK-4			1. Không nhập HỌ 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Vui lòng nhập họ"
DK-5	Kiểm tra giá trị trường Tên		1. Nhập Tên toàn kí tự trống 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Vui lòng nhập Tên"

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DK-6			1. Nhập Tên toàn kí đặc biệt 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Tên không bao gồm kí tự đặc biệt và số!!!"
DK-7			1. Nhập Tên toàn kí tự số 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Tên không bao gồm kí tự đặc biệt và số!!!"
DK-8			1. Không nhập TÊN 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Vui lòng nhập Tên"
DK-9	Kiểm tra giá trị trường email		1. Nhập email toàn kí tự trống 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "vui lòng nhập email "
DK-10			1. Nhập các trường còn lại hợp lệ, nhập email thiếu @ 2. Click ĐĂNG KÝ	3. Hiện thị thông báo "vui lòng bao gồm '@' trong địa chỉ email".
DK-11			1. Nhập các trường còn lại hợp lệ, nhập toàn kí tự trắng đứng trước	3. Hiện thị thông báo "vui lòng nhập phần đứng trước '@'"

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
			@ 2.Click ĐĂNG KÝ	
DK-12			1.Nhập các trường còn lại hợp lệ, nhập email không đúng định dạng 'xyz@gmail.com' 2.Click ĐĂNG KÝ	3. Hiện thị thông báo "Email sai định dạng"
DK-13	Kiểm tra giá trị trường MẬT KHẨU		1. Nhập MẬT KHẨU toàn kí tự trống 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Vui lòng nhập mật khẩu"
DK-14			1. Nhập MẬT KHẨU chứa ký tự đặt biệt 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo " Mật khẩu không được chứa dấu cách, dấu nháy, ký tự >".
DK-15			1. Nhập MẬT KHẨU chứa ký trắng ở đầu và cuối 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo " Mật khẩu không được chứa dấu cách, dấu nháy, ký tự >".

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DK-16			1. Nhập MẬT KHẨU 5 ký tự 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Mật khẩu dài từ 6 đến 50 ký tự"
DK-17			1. Nhập MẬT KHẨU 6 ký tự 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Đăng kí thành công
DK-18			1. Nhập MẬT KHẨU 20 ký tự 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Đăng kí thành công
DK-19			1. Nhập MẬT KHẨU 50 ký tự 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Đăng kí thành công
DK-20			1. Nhập MẬT KHẨU 51 ký tự 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐĂNG KÝ	3. Hiện thị thông báo "Mật khẩu dài từ 6 đến 50 ký tự"
DK-21			1. Nhập các trường còn lại đúng quy tắc	3. Hiện thị thông báo "Vui lòng nhập mật khẩu"

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
			2. Không nhập MẬT KHẨU 3. Ấn ĐĂNG KÝ	
DK-22	Business		1. Bỏ trống tất cả các trường 2. Ấn ĐĂNG KÝ	3. Hiện thị thông báo Yêu cầu nhập
DK-23			1. Nhập thông tin hợp lệ 2. Ấn ĐĂNG KÝ	3. Đăng kí thành công, laod ra trang chủ
DK-24			1. Nhập thông tin 2. Ấn tải lại trang	3. Hiện thị thông báo sẽ bị mất nội dung đã nhập
DK-25			1. Nhập thông tin trùng với tài khoản đã được đăng ký 2. Ấn tải lại trang	3. Hiện thị thông báo trùng
DK-26			1. Kiểm tra mật khẩu có được mã hóa không	2. Mật khẩu được mã hóa
DK-27			1. Kiểm tra mật khẩu có được hiển thị không	2. Không hiển thị
DK-28			1. Đăng ký thành công 2. Ấn back	3. Hiện thị thông tin không hiển thị mật khẩu

4.2.3 Test Case Giỏ Hàng

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
Giỏ Hàng				
GH-1	Kiểm tra tăng giảm số lượng		1. Ấn tăng số lượng lên 1 2. Kiểm tra giá trị số lượng	2. Số lượng tăng lên 1
GH-2			1. Ấn số lượng giảm 1 2. Kiểm tra giá trị số lượng	2. Số lượng giảm 1
GH-3			1. Nhập SỐ LƯỢNG chứa kí tự rỗng, chữ, đặc biệt 2. ấn CẬP NHẬT GIỎ HÀNG	2. Không nhập được, hệ thống chỉ cho phép nhập số
GH-6	BUSINESS		1. Thêm 2 sản phẩm giống nhau vào giỏ hàng 2. Kiểm tra GIỎ HÀNG	2. Hiện thị 1 sản phẩm với số lượng 2
GH-7			1. Thêm 2 sản phẩm khác nhau vào giỏ hàng 2. Kiểm tra GIỎ HÀNG	2. Hiện thị 2 sản phẩm với mỗi sản phẩm có số lượng 1

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
GH-8			1. Không thêm sản phẩm nào trong giỏ hàng 2. Kiểm tra GIỎ HÀNG	2. Hiện tại không có sản phẩm nào trong giỏ hàng của bạn
GH-9			1. Xóa hết sản phẩm trong giỏ 2. Kiểm tra GIỎ HÀNG	2. Hiện tại không có sản phẩm nào trong giỏ hàng của bạn

4.2.4 Test Case Tìm kiếm

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
Tìm kiếm				
TK-1	Kiểm tra giá trị trường TÌM KIẾM		1. Nhập thiếu thông tin cần tìm vào trường TÌM KIẾM 2. Ấn nút TÌM KIẾM	2. Tìm kiếm và ra các sản phẩm liên quan tới từ khóa vừa tìm
TK-2			1. Nhập chính xác thông tin cần tìm vào trường TÌM KIẾM 2. Ấn nút TÌM KIẾM	2. Tìm kiếm và ra các sản phẩm chính xác thông tin cần tìm
TK-3			1. Nhập trường TÌM KIẾM toàn kí tự trống 2. Ấn nút TÌM KIẾM	2. không hiển thị sản phẩm nào
TK-4			1. Nhập trường TÌM KIẾM toàn kí tự đặc biệt 2. Ấn nút TÌM KIẾM	2. Không ra sản phẩm nào
TK-5	BUSINESS		1. Không nhập gì vào trường TÌM KIẾM 2. Ấn nút TÌM KIẾM	2. không hiển thị sản phẩm nào

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
TK-6			1.Nhập đúng thông tin vào trường TÌM KIẾM	2. Tìm kiếm và ra kết quả cần tìm
TK-7			1. Nhập thông thiếu thông tin cần tìm vào trường TÌM KIẾM 2. Ấn nút tải lại trang	2. Trường tìm kiếm mất thông tin vừa nhập

4.2.5 Test Case Đặt hàng

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
Đặt hàng khi chưa đăng nhập				
DH-1	Kiểm tra giá trị trường HỌ VÀ TÊN		1. Nhập HỌ VÀ TÊN toàn kí tự trống 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Vui lòng nhập họ và tên"
DH-2			1. Nhập HỌ VÀ TÊN toàn kí đặc biệt 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Họ không bao gồm kí tự đặc biệt và số!!!"
DH-3			1. Nhập HỌ VÀ TÊN toàn kí tự số 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Họ không bao gồm kí tự đặc biệt và số!!!"

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DH-4			1. Không nhập HỌ VÀ TÊN 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Vui lòng nhập họ và tên"
DH-5	Kiểm tra giá trị trường SỐ ĐIỆN THOẠI		1. Nhập chữ vào trường SỐ ĐIỆN THOẠI 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "số điện thoại không hợp lệ."
DH-6			1. Nhập SỐ ĐIỆN THOẠI toàn kí tự đặc biệt 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Số điện thoại không hợp lệ."
DH-7			1. Nhập SỐ ĐIỆN THOẠI toàn kí tự trống 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Vui lòng nhập số điện thoại."

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DH-8			1. Nhập SỐ ĐIỆN THOẠI 9 kí tự số 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Số điện thoại không hợp lệ."
DH-9			1. Nhập SỐ ĐIỆN THOẠI 10 kí tự đúng 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Đặt hàng thành công
DH-10			1. Nhập SỐ ĐIỆN THOẠI 11 kí tự số 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Số điện thoại không hợp lệ"
DH-11			1. Nhập SỐ ĐIỆN THOẠI có chứa khoảng trắng 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Đặt hàng thành công

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DH-12			1. Không nhập SỐ ĐIỆN THOẠI 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Số điện thoại không hợp lệ"
DH-13	Kiểm tra giá trị trường trường ĐỊA CHỈ		1. Nhập ĐỊA CHỈ toàn kí tự trống 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Vui lòng nhập địa chỉ"
DH-14			1. Không nhập ĐỊA CHỈ 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Vui lòng nhập địa chỉ"
DH-15	Kiểm tra giá trị trường Tỉnh thành		1. Không chọn tỉnh thành 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	3. LABEL xuất hiện "Bạn chưa chọn tỉnh thành"

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DH-16	Kiểm tra giá trị trường Phương thức thanh toán		1. Không chọn Phương thức thanh toán 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo "Bạn cần chọn phương thức thanh toán"
DH-17			1.Chọn Phương thức thanh toán 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Đặt hàng thành công
DH-18	Kiểm tra giá trị trường Mã giảm giá		1. Nhập đúng mã giảm giá 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Đặt hàng thành công
DH-19			1. Nhập sai mã giảm giá 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Hiện thị thông báo " Mã khuyến mãi không hợp lệ"

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output
DH-20			1. Không nhập mã giảm giá 2. Nhập các trường còn lại đúng quy tắc 3. Ấn ĐẶT HÀNG	4. Đặt hàng thành công
Đặt hàng khi đã đăng nhập				
DH-21	Kiểm tra các thông tin hiển thị khi đã ĐĂNG NHẬP		1. Kiểm tra thông tin đặt hàng tự điền có đúng với thông tin tài khoản đã đăng ký (Email, họ tên)	2. Hiện thị đúng với thông tin đã Đăng Ký
DH-22	Business		1. Kiểm tra thông tin sản phẩm	4. Hiện thị đầy đủ thông tin của sản phẩm
DH-23			1. Đặt hàng 2. Ấn back lại	3. Hiện thị màn đặt hàng mất hết dữ liệu đã đặt hàng
DH-24			1. Nhập tất cả các trường 2. Bấm tải lại trang	2. Các trường trống dữ liệu

4.2.6 Test Case hiệu năng

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output	Actual Output
HIỆU NĂNG					
HN-1	Load test		1. Chạy kịch bản test với 10 người dùng cùng truy cập đăng ký	1. Thời gian phản hồi là 1s	Pass
HN-2			1. Chạy kịch bản test với 100 người dùng cùng truy cập đăng ký	1. Thời gian phản hồi là 3s	Pass
HN-3	Stress test		1. Chạy kịch bản test với 500 người dùng cùng truy cập đăng ký	1. Thời gian phản hồi là 20s	Fail
HN-4	Load test		1. Chạy kịch bản test với 10 người dùng cùng truy cập đăng nhập	1. Thời gian phản hồi là 1s	Pass
HN-5			1. Chạy kịch bản test với 100 người dùng cùng truy cập đăng nhập	1. Thời gian phản hồi là 2s	Fail
HN-6	Stress test		1. Chạy kịch bản test với 500 người dùng cùng truy cập đăng nhập	1. Thời gian phản hồi là 21s	Fail

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output	Actual Output
HN-7	Load test		1. Chạy kịch bản test với 10 người dùng cùng truy cập giỏ hàng	1. Thời gian phản hồi là 1s	Pass
HN-8			1. Chạy kịch bản test với 100 người dùng cùng truy cập giỏ hàng	1. Thời gian phản hồi là 1s	Fail
HN-9	Stress test		1. Chạy kịch bản test với 500 người dùng cùng truy cập giỏ hàng	1. Thời gian phản hồi là 21s	Fail
HN-10	Load test		1. Chạy kịch bản test với 10 người dùng cùng truy cập đặt hàng	1. Thời gian phản hồi là 1s	Pass
HN-11			1. Chạy kịch bản test với 100 người dùng cùng truy cập đặt hàng	1. Thời gian phản hồi là 1s	Fail
HN-12	Stress test		1. Chạy kịch bản test với 500 người dùng cùng truy cập đặt hàng	1. Thời gian phản hồi là 21s	Fail

ID	Test Case Description	Pre-condition	Test Case Procedure	Expected Output	Actual Output
HN-13	Performance test		1. Chạy kịch bản test với 10 người truy cập 1s sau 20 người truy cập 3s sau 50 người truy cập (cùng trang web)	1. Thời gian phản hồi là 4s	Pass
HN-14			1. Chạy kịch bản test với 10 người truy cập trang đăng nhập 1s sau 20 người truy cập trang chủ 3s sau 50 người truy cập giỏ hàng	1. Thời gian phản hồi là 4s	Fail
HN-15			1. Chạy kịch bản test với 10 người truy cập đồng thời các trang web(đăng nhập, đăng ký, giỏ hàng, trang chủ)	1. Thời gian phản hồi là 2s	Pass
HN-16			1. Chạy kịch bản test với 1000 người truy cập trang chủ trong 1s	1. Thời gian phản hồi là 22s	Fail

4.3 Xây dựng dữ liệu kiểm thử

4.3.1 Dữ liệu đăng nhập

Bảng 4. 1 Dữ liệu đăng nhập

Email	MK	Verify
	123123	Vui lòng nhập Email
lyly@gmail.com		Vui lòng nhập Mật khẩu
lin1234@gmail.com	123123	Thông tin đăng nhập không chính xác.
	123123	Vui lòng nhập Email
lyly@gmail.com	123123	Thông tin đăng nhập không chính xác.
lyly@gmail.com	1111111	Thông tin đăng nhập không chính xác.
lyly@gmail.com	123123	
lyly@gmail.com	123123	

4.3.2 Dữ liệu đăng ký

Bảng 4. 2 Dữ liệu đăng ký

Ho	Ten	Email	Mật Khẩu
	a	Tranngoan155@gmail.com	123123
a	a	Tranngoan1@gmail.com	123123
			1234567890asdfghjklzxcvbnmq wertyuiopasdfghjklzxcvb
a	ly	lylym2@gmail.com	
a	ly	lylym3@gmail.com	zxcvbnmzxc vbnmmzxcvb
a		Tranngoan12@gmail.com	123123
a		Tranngoan123456@gmail.com	123123
a	ly		123123
a	ly	lylygmail.com	123123
a	ly	lyly@	123123
a	ly	lyly@gmail.com	123123
a	ly	@gmail.com	123123
a	ly	lyly12@gmail.com	709 7
a	ly	lyly12@gmail.com	1.>3444
a	ly	lyly12@gmail.com	12312
		lyly12@gmail.com	1234567890asdfghjklzxcvb nmqwertyuiopasdfghjklzxc vb1
a	ly	lyly12@gmail.com	
a	12345	lyly06@gmail.com	123123
a>< @	lyly07@gmail.com	123123
	mai	ly1@gmail.com	123123
.. @	mai	ly2@gmail.com	123123
12312 3	mai	ly3@gmail.com	123123
	mai	ly4@gmail.com	123123

4.3.3 Dữ liệu tìm kiếm

Bảng 4. 3 Dữ liệu tìm kiếm

SP	Verify
	Không còn sản phẩm nào thỏa mãn.
Camera	Kết quả tìm kiếm cho "camera"
Camera giám sát vỏ kim loại STZ-3203	Kết quả tìm kiếm cho "Camera giám sát vỏ kim loại STZ-3203"
??><<<	Kết quả tìm kiếm cho "??><<<"
	Kết quả tìm kiếm cho " "

4.3.4 Dữ liệu đặt hàng

Bảng 4. 4 Dữ liệu đặt hàng

HoTen	SĐT	DiaChi	TT	MGG	verify
	0987654321	abc	Hà Nội	ADF	Vui lòng nhập họ tên
	0987654323	abc	Hà Nội	ADF	Vui lòng nhập họ tên
hai		abc	Hà Nội	ADF	Vui lòng nhập số điện thoại
hai	abcf	abc	Hà Nội	ADF	Số điện thoại không hợp lệ
hai		abc	Hà Nội	ADF	Vui lòng nhập số điện thoại
hai	><....	abc	Hà Nội	ADF	Số điện thoại không hợp lệ
hai	09876543256	abc	Hà Nội	ADF	Số điện thoại không hợp lệ
hai	098765432	abc	Hà Nội	ADF	Số điện thoại không hợp lệ
hai	0987654321		Hà Nội	ADF	
hai	0987654322		Hà Nội	ADF	

4.4 Xây dựng kịch bản kiểm thử

4.4.1 Keyword Đăng ký

```
package dangky

import static
com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
import static
com.kms.katalon.core.testobject.ObjectRepository.findTestObject
import static
com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject

import com.kms.katalon.core.annotation.Keyword
import com.kms.katalon.core.checkpoint.Checkpoint
import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as
CucumberKW
import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as
Mobile
import com.kms.katalon.core.model.FailureHandling
import com.kms.katalon.core.testcase.TestCase
import com.kms.katalon.core.testdata.TestData
import com.kms.katalon.core.testobject.TestObject
import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as
Windows
import internal.GlobalVariable
import com.kms.katalon.core.testng.keyword.TestNGBuiltInKeywords as
TestNGKW
import org.openqa.selenium.Keys as Keys
public class dangky_1 {
    @Keyword
    def DangKy_ktc (String lastName, String firstName, String
password, String email ) {
        WebUI.openBrowser('')

        WebUI.navigateToUrl('https://samtech.vn/')

        WebUI.click(findTestObject('DK/DK'))

        WebUI.setText(findTestObject('DK/Page_ng k ti
khon/input_lastName'), lastName)

        WebUI.setText(findTestObject('DK/Page_ng k ti
khon/input_firstName'), firstName)

        WebUI.setText(findTestObject('DK/Page_ng k ti
khon/input_password'), password)

        WebUI.setText(findTestObject('DK/Page_ng k ti
khon/input_email'), email)

        WebUI.click(findTestObject('DK/Page_ng k ti
khon/button_DK'))
    }
}
```

```
        String checkURL = WebUI.getUrl()

        WebUI.verifyEqual(checkURL,
'https://samtech.vn/account/register')

        WebUI.closeBrowser()
    }
    @Keyword
    def DangKy_tc (String lastName, String firstName, String password,
String email ) {
        WebUI.openBrowser('')

        WebUI.navigateToUrl('https://samtech.vn/')

        WebUI.click(findTestObject('DK/DK'))

        WebUI.setText(findTestObject('DK/Page_ng k ti
khon/input_lastName'), lastName)

        WebUI.setText(findTestObject('DK/Page_ng k ti
khon/input_firstName'), firstName)

        WebUI.setText(findTestObject('DK/Page_ng k ti
khon/input_password'), password)

        WebUI.setText(findTestObject('DK/Page_ng k ti
khon/input_email'), email)

        WebUI.click(findTestObject('DK/Page_ng k ti
khon/button_DK'))

        String checkURL = WebUI.getUrl()

        WebUI.verifyEqual(checkURL, 'https://samtech.vn/')

        WebUI.closeBrowser()
    }
}
```

4.4.2 Keyword Đăng nhập

```
package dangnhap

import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
import static
com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject

import com.kms.katalon.core.annotation.Keyword
import com.kms.katalon.core.checkpoint.Checkpoint
import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as
CucumberKW
import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile
import com.kms.katalon.core.model.FailureHandling
import com.kms.katalon.core.testcase.TestCase
import com.kms.katalon.core.testdata.TestData
import com.kms.katalon.core.testobject.TestObject
import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows

import groovy.json.StringEscapeUtils
import internal.GlobalVariable

public class dangnhap_1 {
    //public void String email, String password
    @Keyword
    def dangnhap_pass(String email, String password) {
        WebUI.openBrowser('')

        WebUI.navigateToUrl('https://samtech.vn/')

        WebUI.click(findTestObject('Object Repository/Page_Phn phi thit b
in thng minh Smart home IOT/a_ng nhp'))

        WebUI.setText(findTestObject('Object Repository/Page_ng nhp ti
khon/input__email'), email)

        WebUI.setText(findTestObject('Object Repository/Page_ng nhp ti
khon/input__password'), password)

        WebUI.click(findTestObject('Object Repository/Page_ng nhp ti
khon/button_ng nhp'))

        WebUI.verifyElementText(findTestObject('Object
Repository/Page_Trang khch hng/img'), '')
        WebUI.closeBrowser()
    }
    @Keyword
    def dangnhap_fail(String email, String password) {
        WebUI.openBrowser('')

        WebUI.navigateToUrl('https://samtech.vn/')
```

```
WebUI.click(findTestObject('Object Repository/Page_Phn phi thit b
in thng minh Smart home IOT/a_ng nhp'))

WebUI.setText(findTestObject('Object Repository/Page_ng nhp ti
khon/input__email'), email)

WebUI.setText(findTestObject('Object Repository/Page_ng nhp ti
khon/input__password'), password)

WebUI.click(findTestObject('Object Repository/Page_ng nhp ti
khon/button_ng nhp'))

WebUI.verifyElementText(findTestObject('Object
Repository/Page_Trang khch hng/img'), '')
WebUI.closeBrowser()
}
}
```

4.4.3 Keyword Tìm kiếm

```
package timkiem

import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
import static
com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject

import com.kms.katalon.core.annotation.Keyword
import com.kms.katalon.core.checkpoint.Checkpoint
import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as
CucumberKW
import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile
import com.kms.katalon.core.model.FailureHandling
import com.kms.katalon.core.testcase.TestCase
import com.kms.katalon.core.testdata.TestData
import com.kms.katalon.core.testobject.TestObject
import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows

import internal.GlobalVariable

public class Keyword_TK {
    @Keyword
    def TK(String SP,String ketqua) {
        WebUI.openBrowser('')

        WebUI.navigateToUrl('https://samtech.vn/')

        WebUI.click(findTestObject('GioHang/Page_Phn phi thit b in thng
minh Smart home IOT/img'))

        if(SP.length() == 0) {

            WebUI.click(findTestObject('Object Repository/Page_Phn phi
thit b in thng minh Smart home IOT/button_Kinh Doanh 02462927193_search-
form__submit'))
            WebUI.verifyElementText(findTestObject('Object
Repository/tim-kiem/Page_Kt qu tm kim cho/kq-tk'), "Không còn sản phẩm nào thỏa
mãn.")
        }else {
            WebUI.setText(findTestObject('Object Repository/Page_Phn phi
thit b in thng minh Smart home IOT/input_Kinh Doanh 02462927193_query'),SP)

            WebUI.click(findTestObject('Object Repository/Page_Phn phi
thit b in thng minh Smart home IOT/button_Kinh Doanh 02462927193_search-
form__submit'))

            WebUI.verifyElementText(findTestObject('Object
Repository/Page_Kt qu tm kim cho camera/li_Kt qu tm kim cho camera'), ketqua)
        }

        WebUI.closeBrowser()
    }
}
```

4.4.4 Keyword Đặt hàng

```
package dathang

import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
import static
com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject

import com.kms.katalon.core.annotation.Keyword
import com.kms.katalon.core.checkpoint.Checkpoint
import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as
CucumberKW
import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile
import com.kms.katalon.core.model.FailureHandling
import com.kms.katalon.core.testcase.TestCase
import com.kms.katalon.core.testdata.TestData
import com.kms.katalon.core.testobject.TestObject
import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows

import groovy.json.StringEscapeUtils
import internal.GlobalVariable
import java.awt.Robot
import java.awt.event.KeyEvent as KeyEvent
import org.openqa.selenium.Keys as Keys

public class dathang_1 {
    @Keyword
    def diachi(String billingName,billingPhone, billingAddress,
billingProvince, reductionCode) {

        WebUI.callTestCase(findTestCase('Giỏ hàng/GH-CALL'), [:],
FailureHandling.STOP_ON_FAILURE)

        WebUI.click(findTestObject('GioHang/Page_Gi hng (1)/a_Thanh ton'))

        WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Name'),billingName)

        WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Phone'),billingPhone)

        WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Address'),billingAddress)

        WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/Tinhthanh'))

        WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input__TT'), billingProvince)
```



```
WebUI.sendKeys(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/input__TT'), Keys.chord(Keys.ENTER))

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/PTTT'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/input_Mgg'),reductionCode)

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/button_mgg'))

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/button_DH'))

WebUI.verifyElementText(findTestObject('DH/Verify/DC'), 'Vui lòng nhập địa chỉ')

WebUI.closeBrowser()
}

@Keyword
def ten(String billingName,billingPhone, billingAddress, billingProvince, reductionCode, ketqua) {

WebUI.callTestCase(findTestCase('Giỏ hàng/GH-CALL'), [:], FailureHandling.STOP_ON_FAILURE)

WebUI.click(findTestObject('GioHang/Page_Gi hng (1)/a_Thanh ton'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/input_Name'),billingName)

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/input_Phone'),billingPhone)

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/input_Address'),billingAddress)

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/Tinhthanh'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/input__TT'), billingProvince)

WebUI.sendKeys(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/input__TT'), Keys.chord(Keys.ENTER))

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/PTTT'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/input_Mgg'),reductionCode)

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh ton n hng/button_mgg'))
```

```
WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/button_DH'))

WebUI.verifyElementText(findTestObject('DH/Verify/Hoten'), ketqua)

WebUI.closeBrowser()
}
@Keyword
def sdt(String billingName,billingPhone, billingAddress, billingProvince,
reductionCode, ketqua) {

WebUI.callTestCase(findTestCase('Giỏ hàng/GH-CALL'), [:],
FailureHandling.STOP_ON_FAILURE)

WebUI.click(findTestObject('GioHang/Page_Gi hng (1)/a_Thanh ton'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Name'),billingName)

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Phone'),billingPhone)

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Address'),billingAddress)

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/Tinhthanh'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input__TT'), billingProvince)

WebUI.sendKeys(findTestObject('DH/Page_in thng minh Samtech -
Thanh ton n hng/input__TT'), Keys.chord(Keys.ENTER))

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/PTTT'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Mgg'),reductionCode)

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/button_mgg'))

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/button_DH'))

WebUI.verifyElementText(findTestObject('DH/Verify/Sdt'), ketqua)

WebUI.closeBrowser()
}
@Keyword
def mgg(String billingName,billingPhone, billingAddress, billingProvince,
reductionCode, ketqua) {

WebUI.callTestCase(findTestCase('Giỏ hàng/GH-CALL'), [:],
FailureHandling.STOP_ON_FAILURE)
WebUI.click(findTestObject('GioHang/Page_Gi hng (1)/a_Thanh ton'))
```

```
WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Name'),billingName)

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Phone'),billingPhone)

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Address'),billingAddress)

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/Tinhthanh'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input__TT'), billingProvince)

WebUI.sendKeys(findTestObject('DH/Page_in thng minh Samtech -
Thanh ton n hng/input__TT'), Keys.chord(Keys.ENTER))

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/PTTT'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Mgg'),reductionCode)

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/button_mgg'))

WebUI.verifyElementText(findTestObject('DH/Verify/MGG'), ketqua)

WebUI.closeBrowser()
}
@Keyword
def pttt(String billingName,billingPhone, billingAddress,
billingProvince, reductionCode, ketqua) {

WebUI.callTestCase(findTestCase('Giỏ hàng/GH-CALL'), [:],
FailureHandling.STOP_ON_FAILURE)

WebUI.click(findTestObject('GioHang/Page_Gi hng (1)/a_Thanh ton'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Name'),billingName)

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Phone'),billingPhone)

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input_Address'),billingAddress)

WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/Tinhthanh'))

WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/input__TT'), billingProvince)
```

```
WebUI.sendKeys(findTestObject('DH/Page_in thng minh Samtech -  
Thanh ton n hng/input__TT'), Keys.chord(Keys.ENTER))  
  
// WebUI.click(findTestObject('DH/Page_in thng minh  
Samtech - Thanh ton n hng/PTTT'))  
  
WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/input_Mgg'),reductionCode)  
  
WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/button_mgg'))  
  
WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/button_DH'))  
  
WebUI.verifyElementText(findTestObject('DH/Verify/PTTT'), ketqua)  
  
WebUI.closeBrowser()  
}  
@Keyword  
def tt(String billingName,billingPhone, billingAddress, billingProvince,  
reductionCode, ketqua) {  
  
WebUI.callTestCase(findTestCase('Giỏ hàng/GH-CALL'), [:],  
FailureHandling.STOP_ON_FAILURE)  
  
WebUI.click(findTestObject('GioHang/Page_Gi hng (1)/a_Thanh ton'))  
  
WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/input_Name'),billingName)  
  
WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/input_Phone'),billingPhone)  
  
WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/input_Address'),billingAddress)  
  
// WebUI.click(findTestObject('DH/Page_in thng minh  
Samtech - Thanh ton n hng/Tinhthanh'))  
  
// WebUI.setText(findTestObject('DH/Page_in thng  
minh Samtech - Thanh ton n hng/input__TT'), billingProvince)  
  
// WebUI.sendKeys(findTestObject('DH/Page_in thng  
minh Samtech - Thanh ton n hng/input__TT'), Keys.chord(Keys.ENTER))  
  
WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/PTTT'))  
  
WebUI.setText(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/input_Mgg'),reductionCode)  
  
WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh  
ton n hng/button_mgg'))
```

```
WebUI.click(findTestObject('DH/Page_in thng minh Samtech - Thanh
ton n hng/button_DH'))

WebUI.verifyElementText(findTestObject('DH/Verify/Tinhthanh'),
ketqua)

WebUI.closeBrowser()
}
```

4.4.5 Test scrip giỏ hàng

```
import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
import static
com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject
import com.kms.katalon.core.checkpoint.Checkpoint as Checkpoint
import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as
CucumberKW
import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile
import com.kms.katalon.core.model.FailureHandling as FailureHandling
import com.kms.katalon.core.testcase.TestCase as TestCase
import com.kms.katalon.core.testdata.TestData as TestData
import com.kms.katalon.core.testng.keyword.TestNGBuiltinKeywords as TestNGKW
import com.kms.katalon.core.testobject.TestObject as TestObject
import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows
import internal.GlobalVariable as GlobalVariable
import org.openqa.selenium.Keys as Keys

WebUI.openBrowser('')

WebUI.navigateToUrl('https://samtech.vn/')

WebUI.setText(findTestObject('Page_Phn phi thit b in thng minh Smart home
IOT/input_tk'),
    'camera')

WebUI.click(findTestObject('TK/button_TK'))

WebUI.click(findTestObject('GioHang/them2sp-giong/span_Cho vo gi hng1'))

WebUI.click(findTestObject('GioHang/them2sp-giong/span_Cho vo gi hng1'))

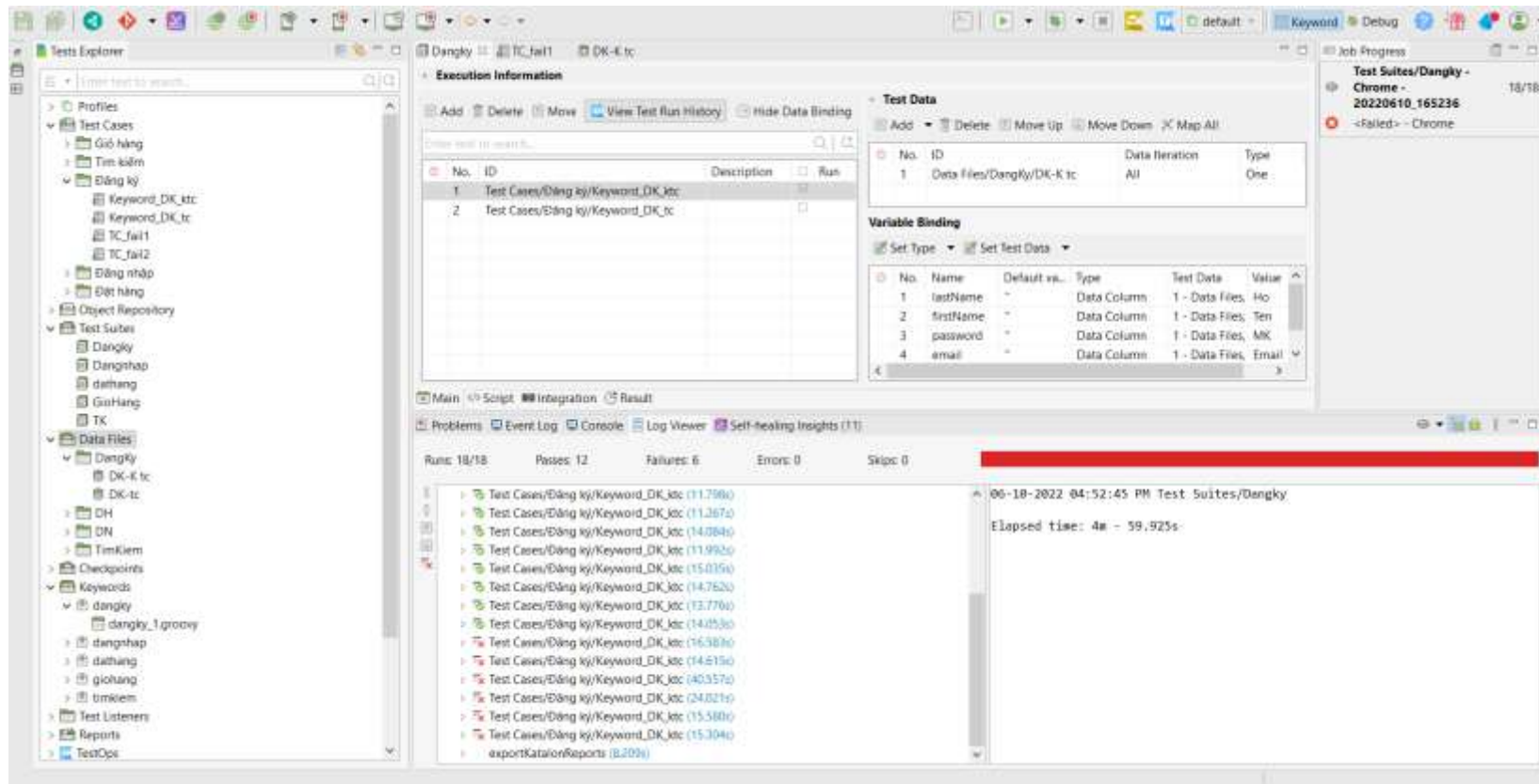
WebUI.click(findTestObject('GioHang/them2sp-giong/a_Gi hng (0)'))

WebUI.verifyElementText(findTestObject('GioHang/Page_Gi hng (1)/sl-
2spgionghau'), '')

WebUI.closeBrowser()
```

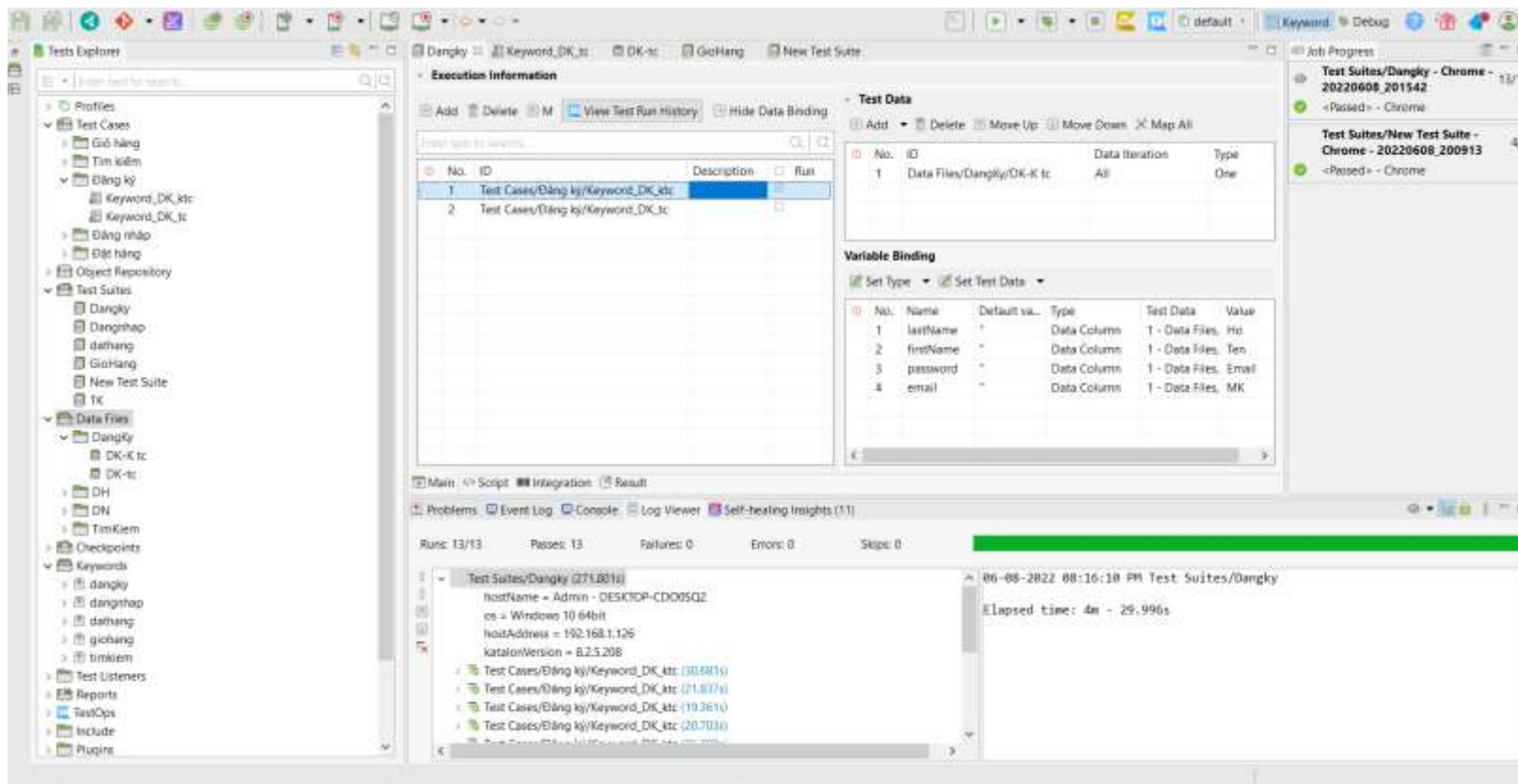
4.5 Thực thi và báo cáo kiểm thử

4.5.1 Thực thi kiểm thử katalon studio



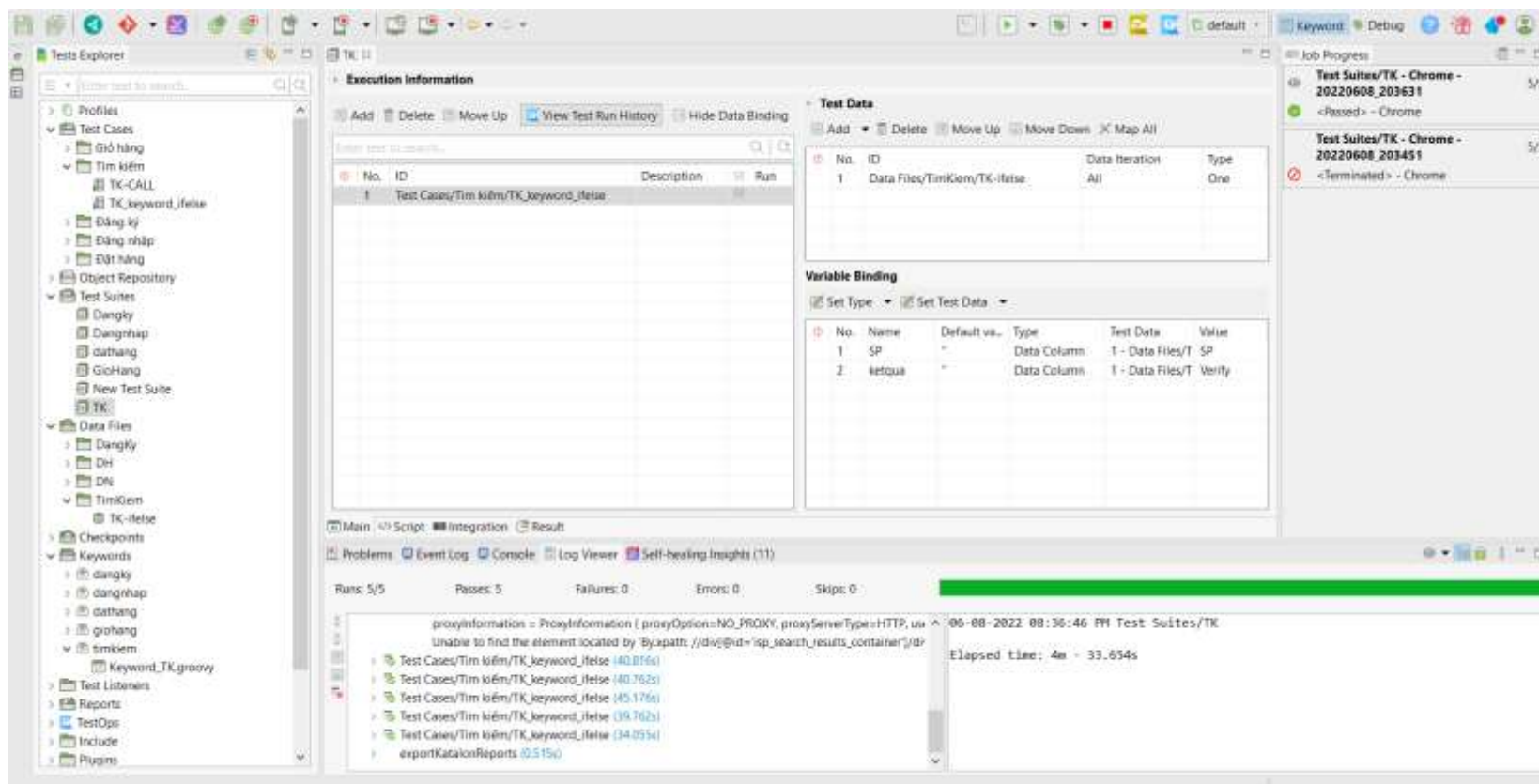
Hình 4. 9 : Kết quả chạy Test Suite Đăng nhập trên Katalon Studio

Kiểm thử tự động cho website camera Samtech sử dụng Katalon studio và Jmeter



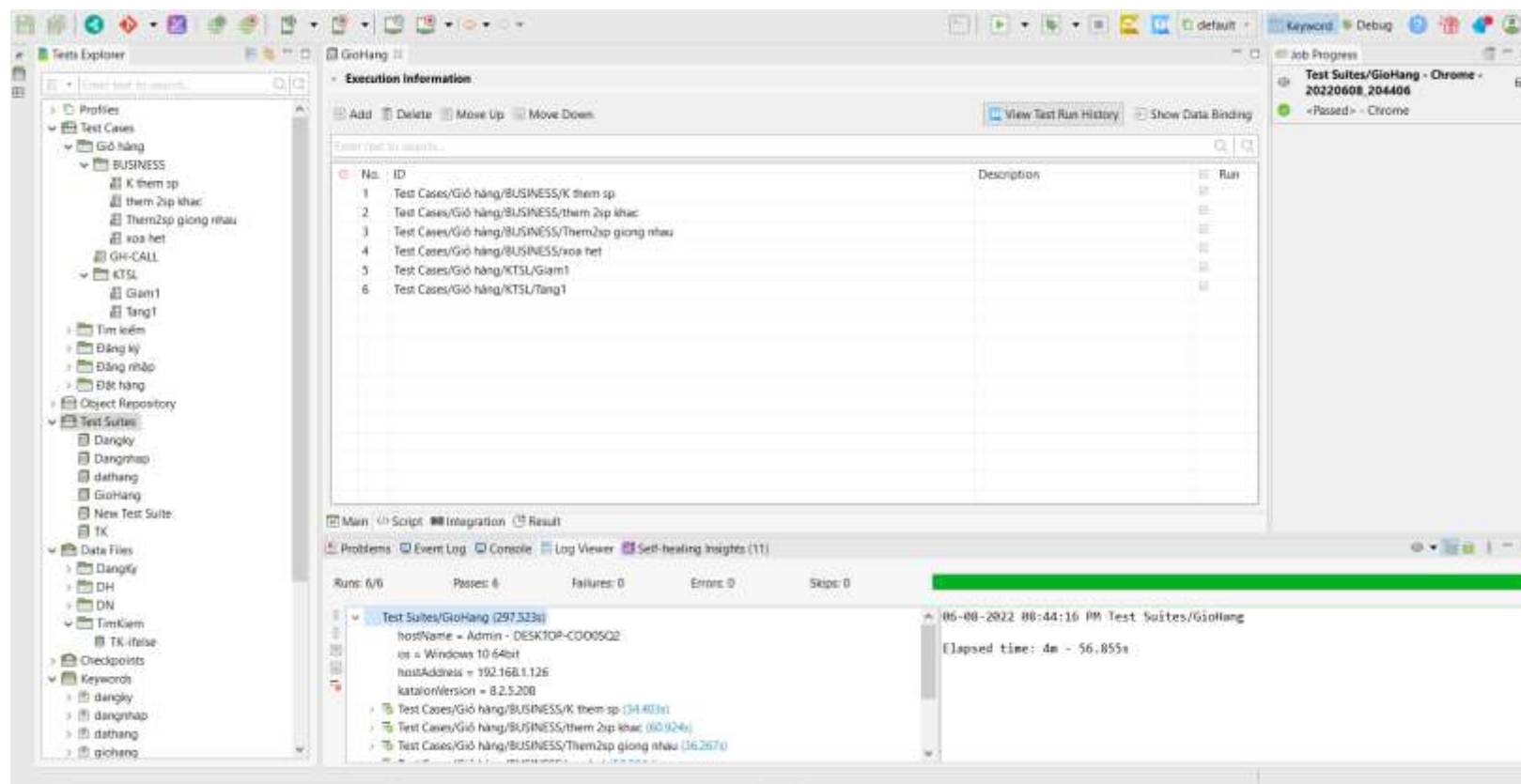
Hình 4. 10 :Kết quả chạy Test Suite Đăng ký trên Katalon Studio

Kiểm thử tự động cho website camera Samtech sử dụng Katalon studio và Jmeter



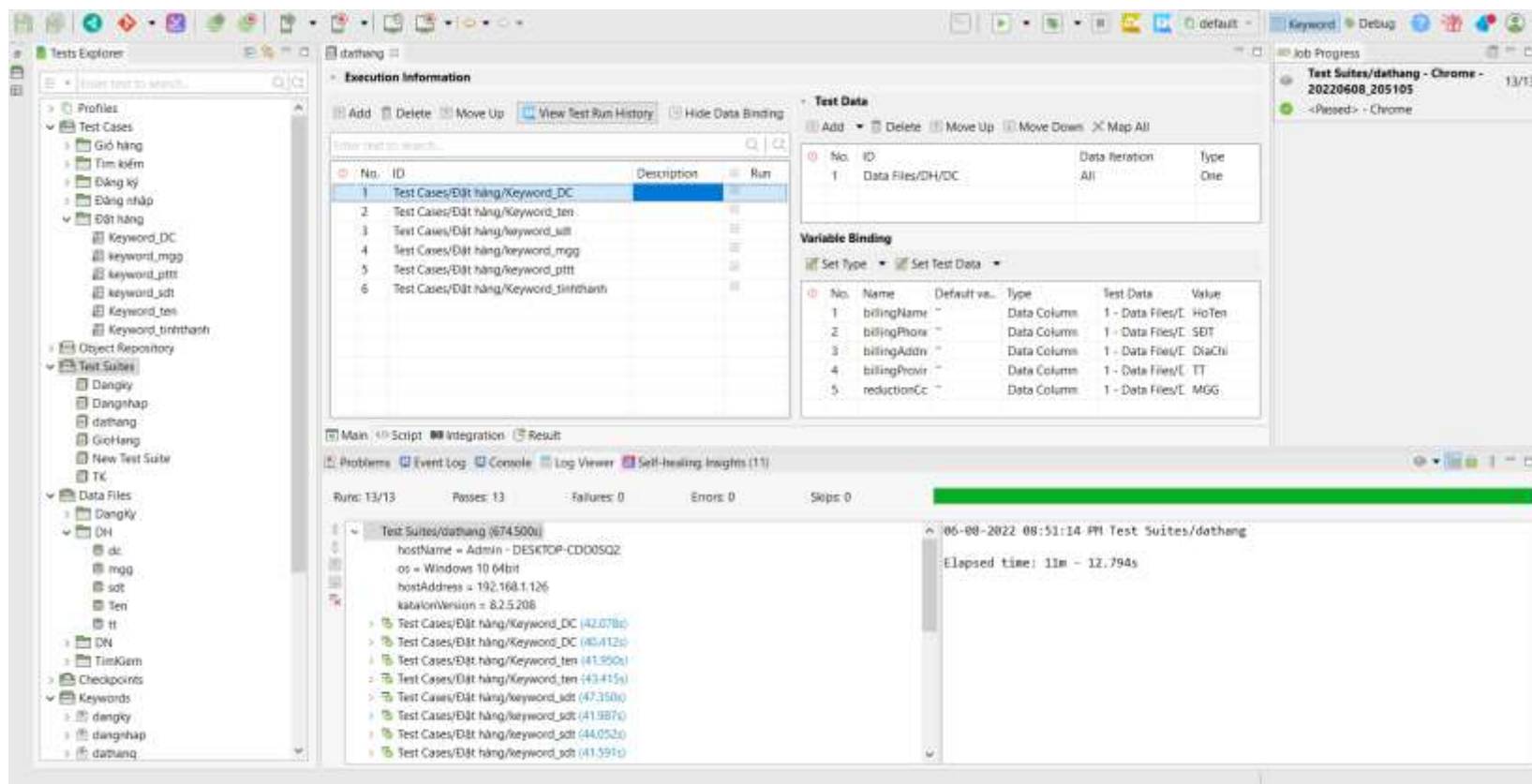
Hình 4. 11 : Kết quả chạy Test Suite Tìm kiếm trên Katalon Studio

Kiểm thử tự động cho website camera Samtech sử dụng Katalon studio và Jmeter



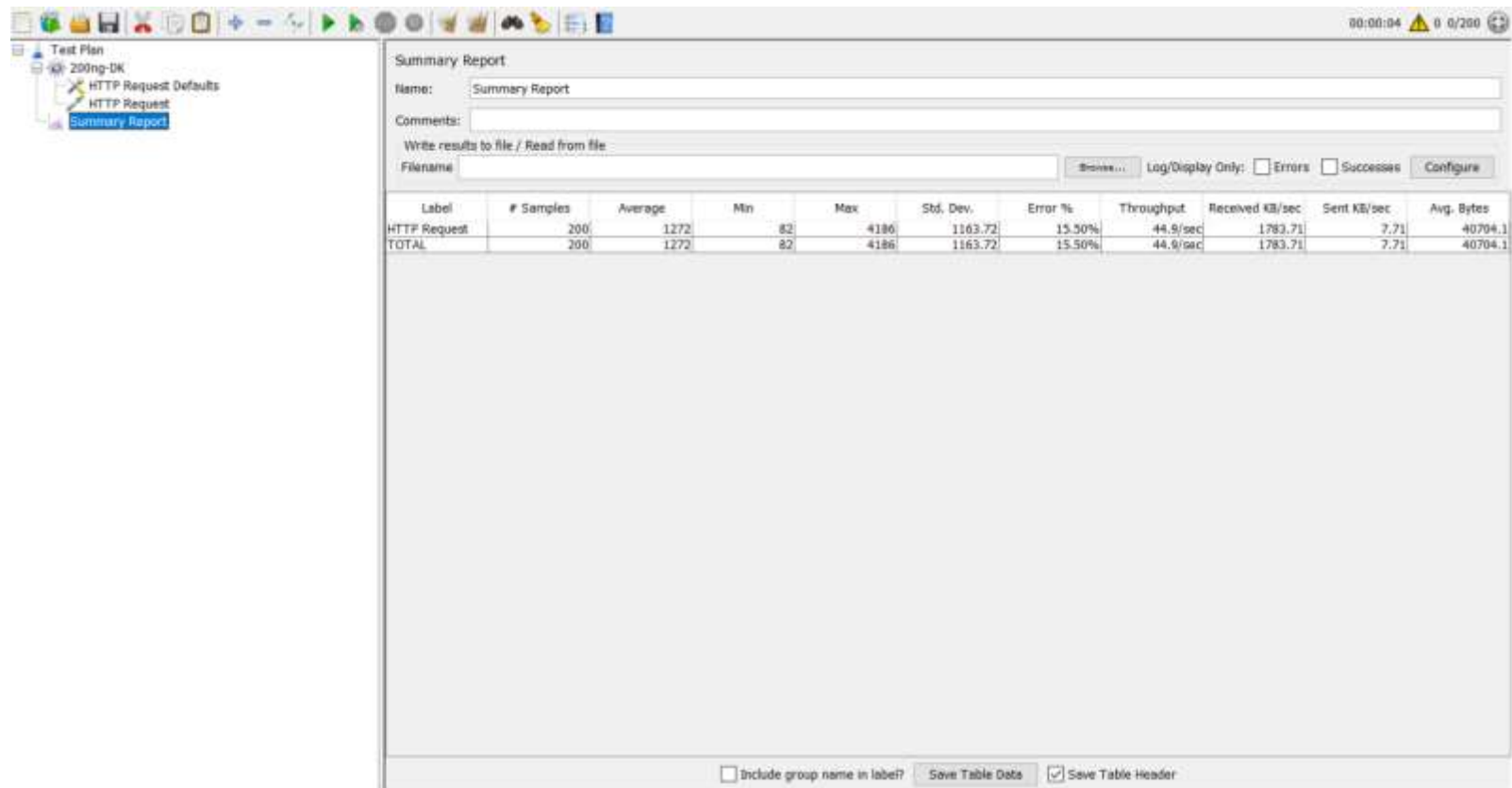
Hình 4. 12 Kết quả chạy Test Suite Giỏ Hàng trên Katalon Studio

Kiểm thử tự động cho website camera Samtech sử dụng Katalon studio và Jmeter



Hình 4. 13 : Kết quả chạy Test Suite Đặt Hàng trên Katalon Studio

4.5.2 Thực thi kiểm thử Jmeter



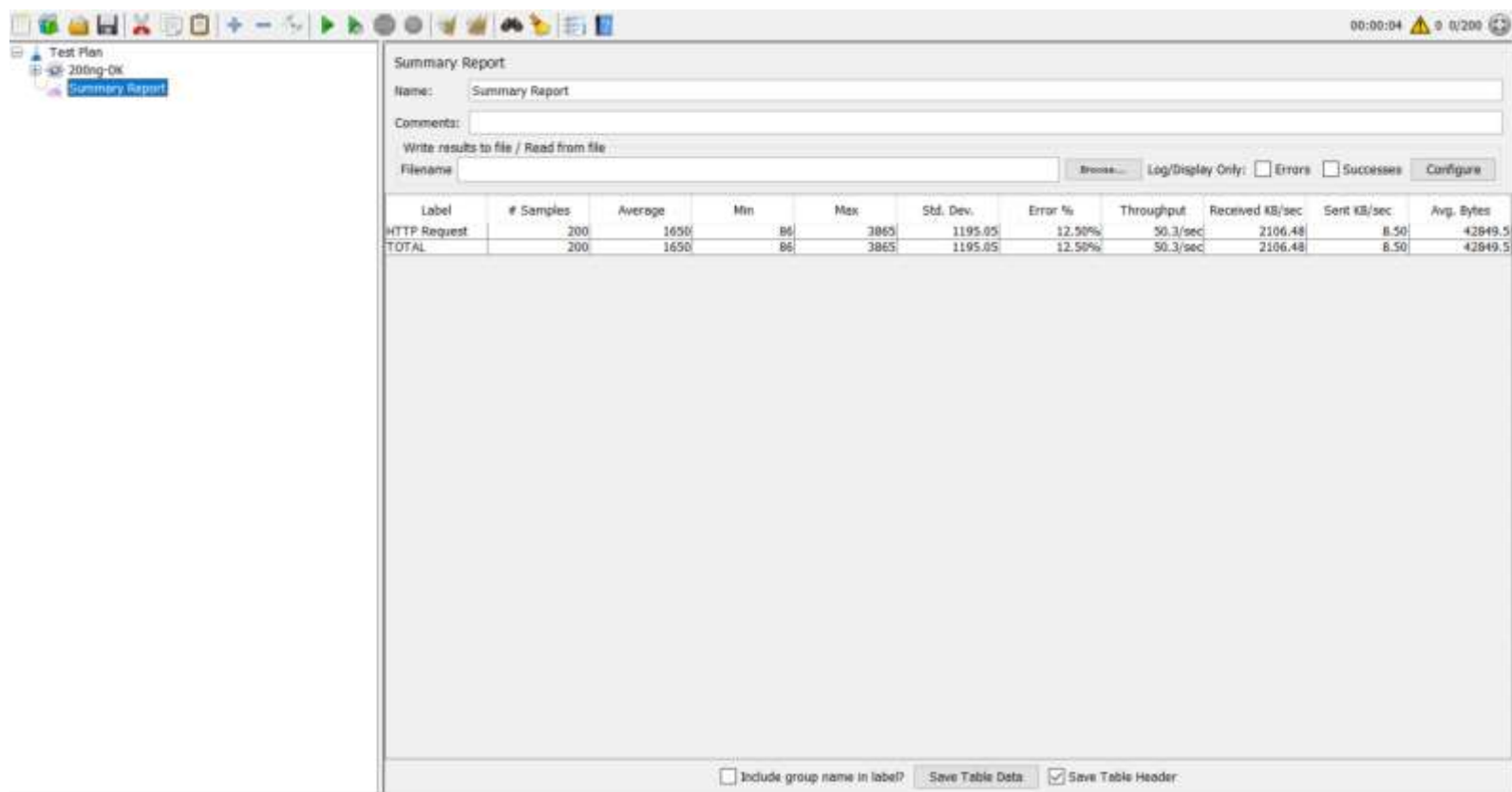
The screenshot shows the JMeter Summary Report window. The left sidebar displays the test plan structure: 'Test Plan' -> '200ng-DK' -> 'HTTP Request Defaults' -> 'HTTP Request' -> 'Summary Report'. The main area is titled 'Summary Report' and contains the following information:

Name: Summary Report
Comments:
Write results to file / Read from file
Filename: Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	200	1272	82	4186	1163.72	15.50%	44.9/sec	1783.71	7.71	40704.1
TOTAL	200	1272	82	4186	1163.72	15.50%	44.9/sec	1783.71	7.71	40704.1

At the bottom, there are checkboxes for 'Include group name in label?' (unchecked), 'Save Table Data' (unchecked), and 'Save Table Header' (checked).

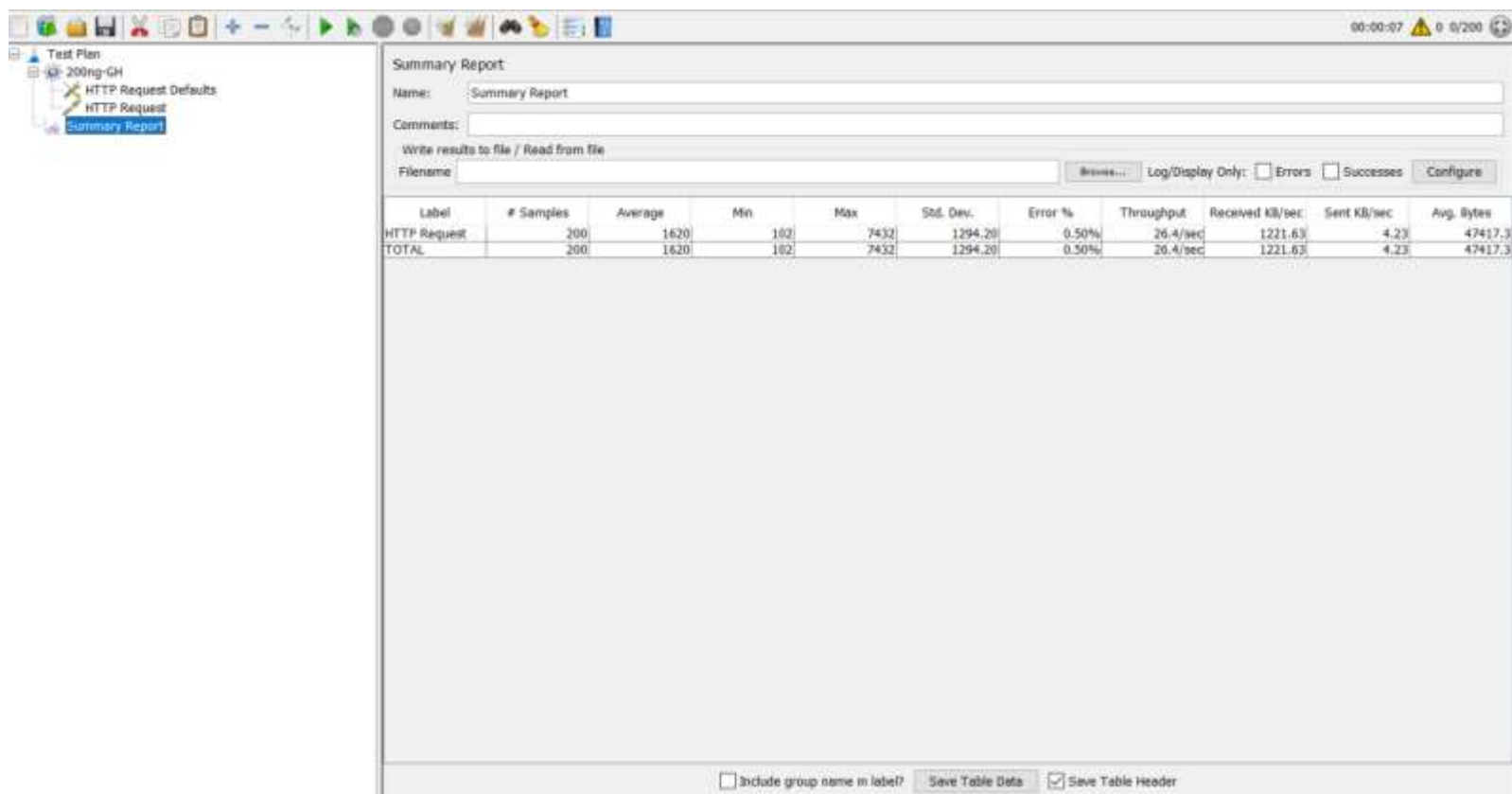
Hình 4. 14 Kết quả chạy Đăng ký trên jmeter



The screenshot shows the JMeter Summary Report window. The report is titled 'Summary Report' and includes fields for Name, Comments, and a 'Write results to file / Read from file' section with a 'Browse...' button. Below these fields is a table with 11 columns: Label, # Samples, Average, Min, Max, Std. Dev., Error %, Throughput, Received KB/sec, Sent KB/sec, and Avg. Bytes. The table contains two rows: 'HTTP Request' and 'TOTAL'. The 'HTTP Request' row shows 200 samples, an average of 1650, a minimum of 86, a maximum of 3865, a standard deviation of 1195.05, an error rate of 12.50%, a throughput of 50.3/sec, received data of 2106.48 KB/sec, sent data of 8.50 KB/sec, and an average of 42849.5 bytes. The 'TOTAL' row shows the same values as the 'HTTP Request' row. At the bottom of the window, there are checkboxes for 'Include group name in label?', 'Save Table Data', and 'Save Table Header'.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	200	1650	86	3865	1195.05	12.50%	50.3/sec	2106.48	8.50	42849.5
TOTAL	200	1650	86	3865	1195.05	12.50%	50.3/sec	2106.48	8.50	42849.5

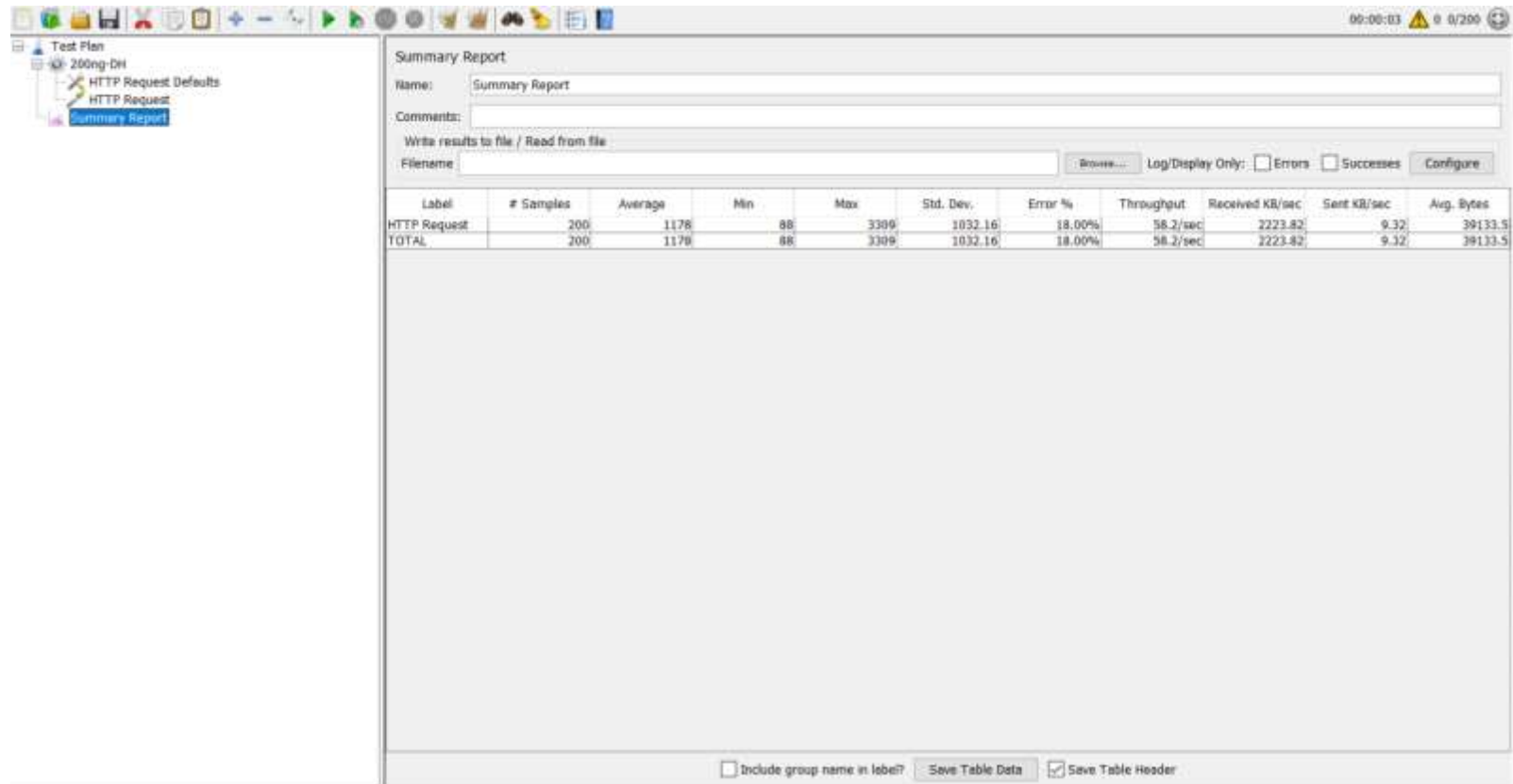
Hình 4. 15 Kết quả chạy Đăng nhập trên jmeter



The screenshot shows the JMeter Summary Report window. The left sidebar displays the test plan structure: Test Plan, 200mg-GH, HTTP Request Defaults, HTTP Request, and Summary Report. The main area shows the Summary Report for the selected test. The report includes fields for Name, Comments, and a section for writing results to a file. Below this is a table with 11 columns: Label, # Samples, Average, Min, Max, Std. Dev., Error %, Throughput, Received KB/sec, Sent KB/sec, and Avg. Bytes. The table contains two rows: HTTP Request and TOTAL. At the bottom, there are checkboxes for 'Include group name in label?' and 'Save Table Header', and a 'Save Table Data' button.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	200	1620	102	7432	1294.20	0.50%	26.4/sec	1221.63	4.23	47417.3
TOTAL	200	1620	102	7432	1294.20	0.50%	26.4/sec	1221.63	4.23	47417.3

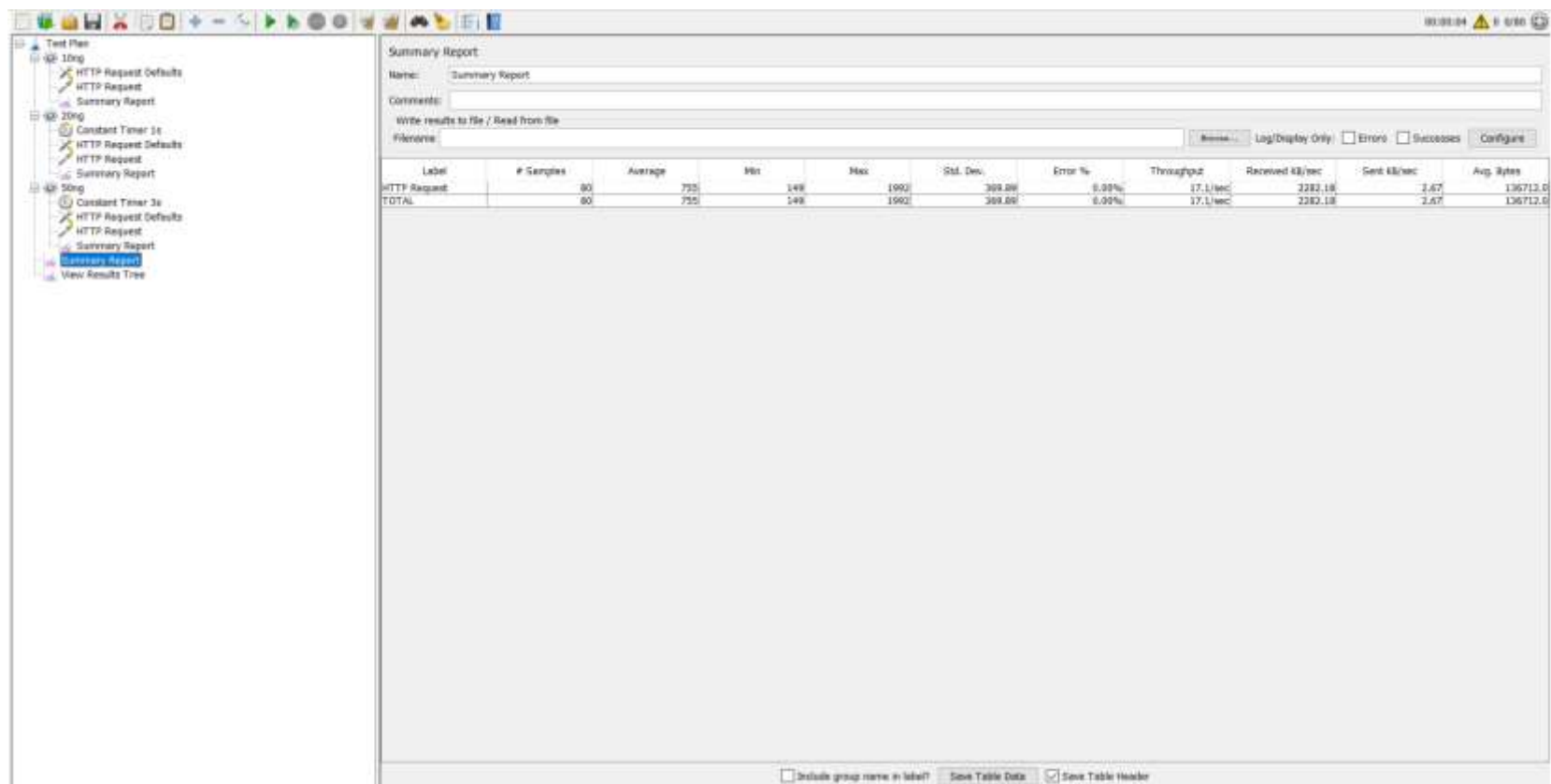
Hình 4. 16 Kết quả chạy Giả Hàng trên jmeter



The screenshot shows the JMeter Summary Report window. The left sidebar displays the test plan structure: Test Plan, 200mg-DH, HTTP Request Defaults, HTTP Request, and Summary Report. The main area shows the Summary Report for the selected test. The report includes fields for Name (Summary Report), Comments, and a section for writing results to a file. Below this is a table with 11 columns: Label, # Samples, Average, Min, Max, Std. Dev., Error %, Throughput, Received KB/sec, Sent KB/sec, and Avg. Bytes. The table contains two rows: HTTP Request and TOTAL. The bottom of the window has checkboxes for 'Include group name in label?', 'Save Table Data', and 'Save Table Header'.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	200	1178	88	3399	1032.16	18.00%	58.2/sec	2223.82	9.32	39133.5
TOTAL	200	1178	88	3399	1032.16	18.00%	58.2/sec	2223.82	9.32	39133.5

Hình 4. 17 Kết quả chạy Đặt Hàng trên jmeter

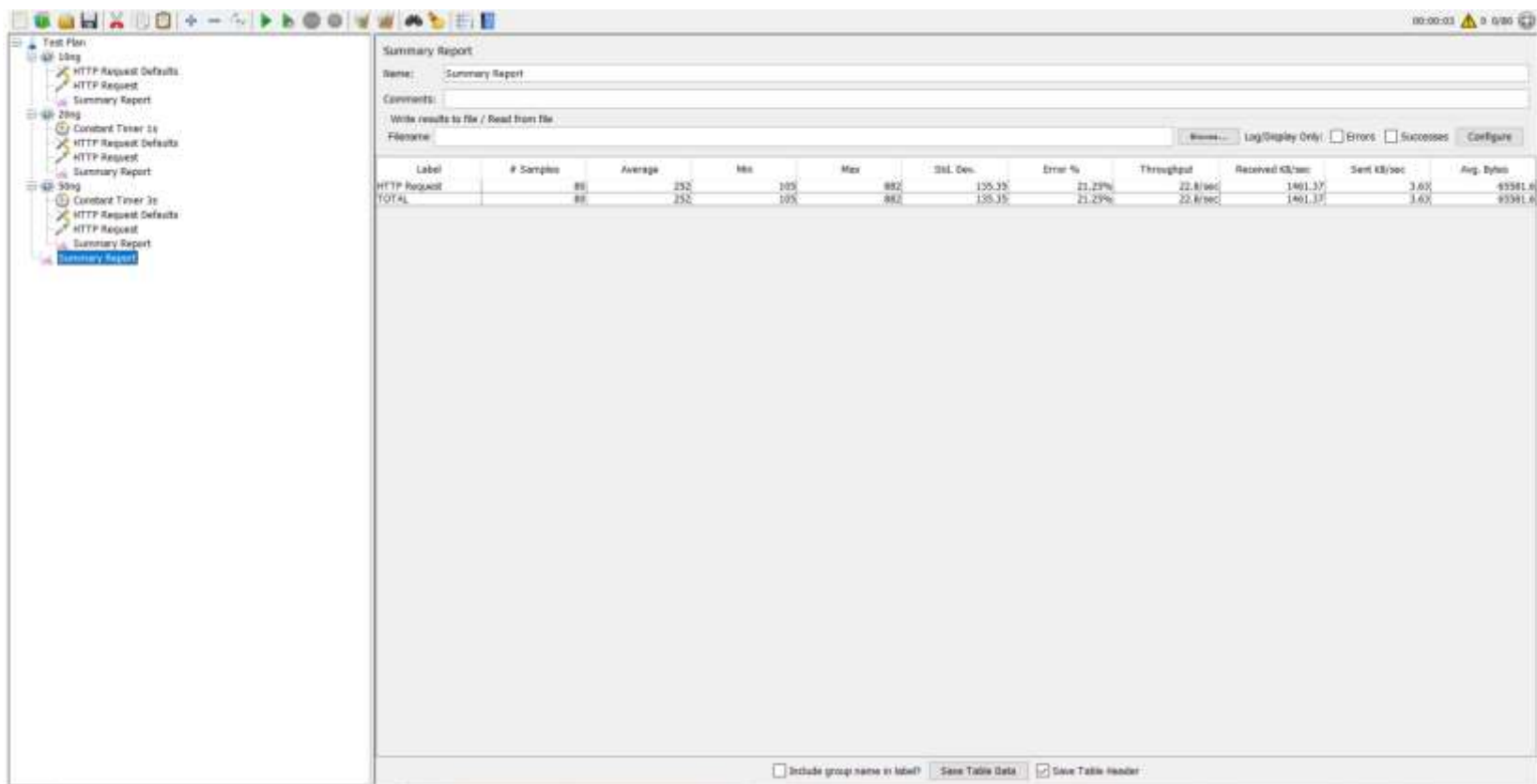


The screenshot shows the JMeter Summary Report window. The left sidebar displays the test plan structure with nodes for '10s', '20s', and '50s' groups, each containing 'Constant Timer', 'HTTP Request Defaults', 'HTTP Request', and 'Summary Report'. The main area shows the 'Summary Report' for the '10s' group. The report includes a table with the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received kB/sec	Sent kB/sec	Avg. Bytes
HTTP Request	80	735	148	1992	388.88	0.00%	17.1/sec	2282.18	2.67	136713.0
TOTAL	80	735	148	1992	388.88	0.00%	17.1/sec	2282.18	2.67	136713.0

At the bottom of the window, there are checkboxes for 'Include group name in label?' (unchecked), 'Save Table Data' (checked), and 'Save Table Header' (checked).

Hình 4. 18: 10 người truy cập 1s sau 20 người truy cập 3s sau 50 người truy cập (cùng trang web)

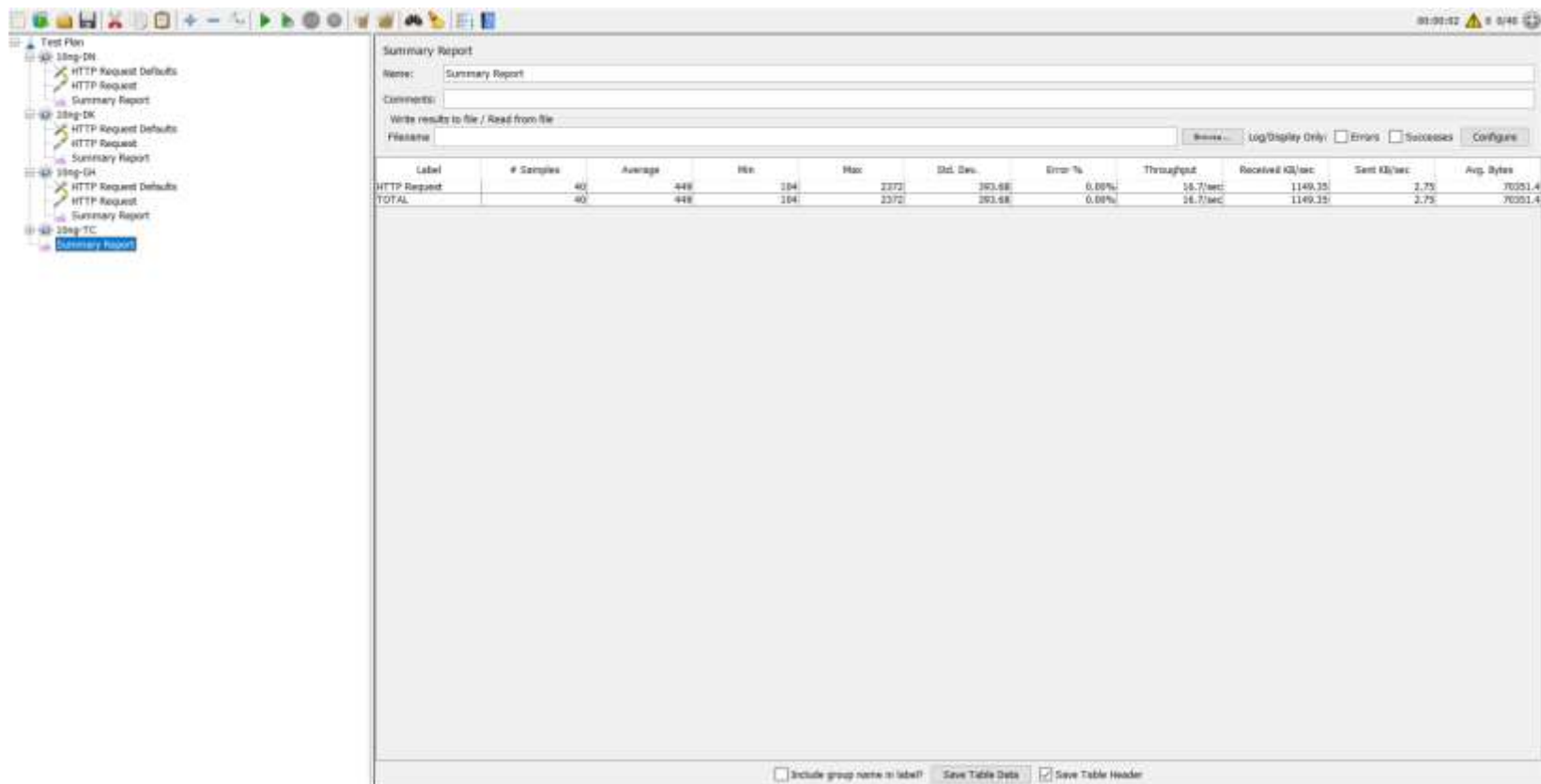


The screenshot shows the JMeter Summary Report window. The left sidebar displays the test plan structure with a tree view. The main area shows the 'Summary Report' for a test named 'Summary Report'. The report includes a table with the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	88	252	105	882	155.35	21.23%	22.8/sec	1461.37	3.63	45581.6
TOTAL	88	252	105	882	155.35	21.23%	22.8/sec	1461.37	3.63	45581.6

At the bottom of the window, there are checkboxes for 'Include group name in label?', 'Save Table Data', and 'Save Table Header'.

Hình 4. 19. 10 người truy cập trang đăng nhập 1s sau 20 người truy cập trang chủ 3s sau 50 người truy cập giỏ hàng



Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

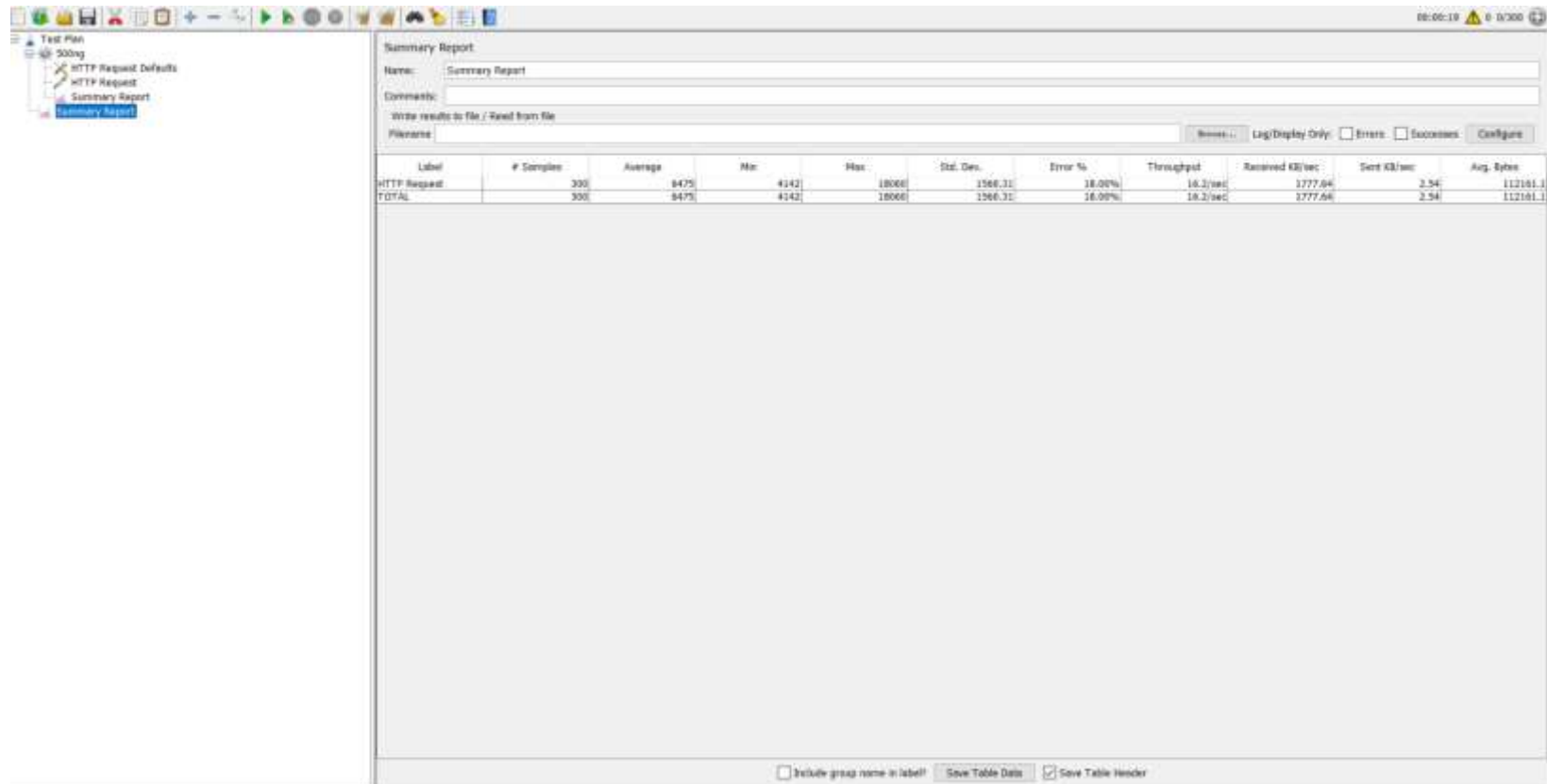
Filename: ☐ Log/Display Only ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	40	448	104	2372	303.68	0.00%	16.7/sec	1149.35	2.75	70251.4
TOTAL	40	448	104	2372	303.68	0.00%	16.7/sec	1149.35	2.75	70251.4

☐ Include group name in label? ☒ Save Table Header

Hình 4. 20 Kiểm tra 10 ng truy cập đồng thời các trang web(đăng nhập, đăng ký, giỏ hàng, trang chủ)

Kiểm thử tự động cho website camera Samtech sử dụng Katalon studio và Jmeter



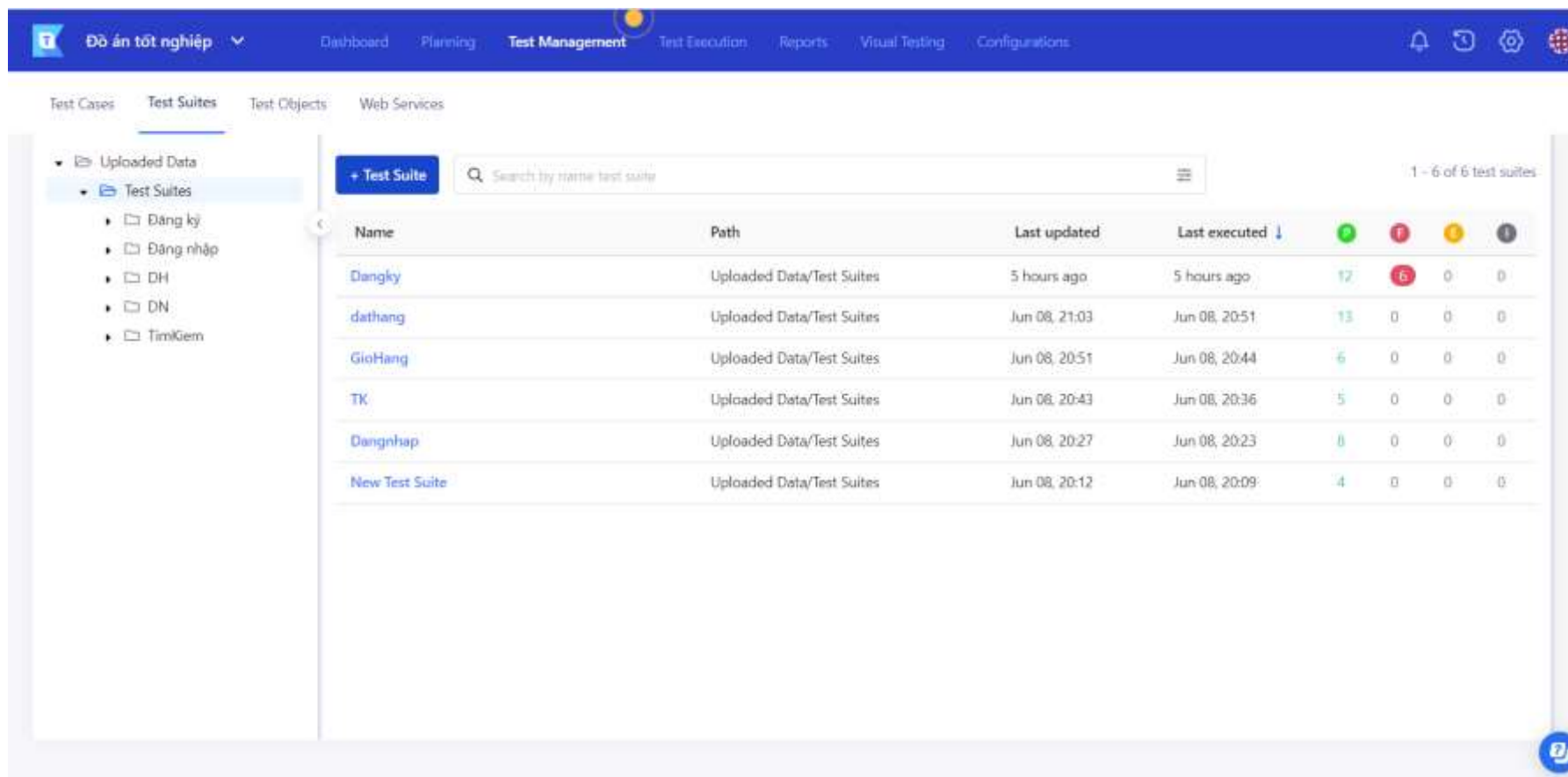
The screenshot shows the JMeter Summary Report window. The left sidebar contains a tree view with 'Test Plan', '300ng', 'HTTP Request Defaults', 'HTTP Request', and 'Summary Report'. The main area displays the 'Summary Report' for the '300ng' test. The report includes a table with the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	300	8475	4142	18066	1566.31	18.00%	16.2/sec	1777.64	2.54	112161.1
TOTAL	300	8475	4142	18066	1566.31	18.00%	16.2/sec	1777.64	2.54	112161.1

At the bottom of the window, there are checkboxes for 'Include group name in label?' (unchecked), 'Save Table Data' (checked), and 'Save Table Header' (checked).

Hình 4. 21: Kiểm tra 300 ng truy cập trong 1s

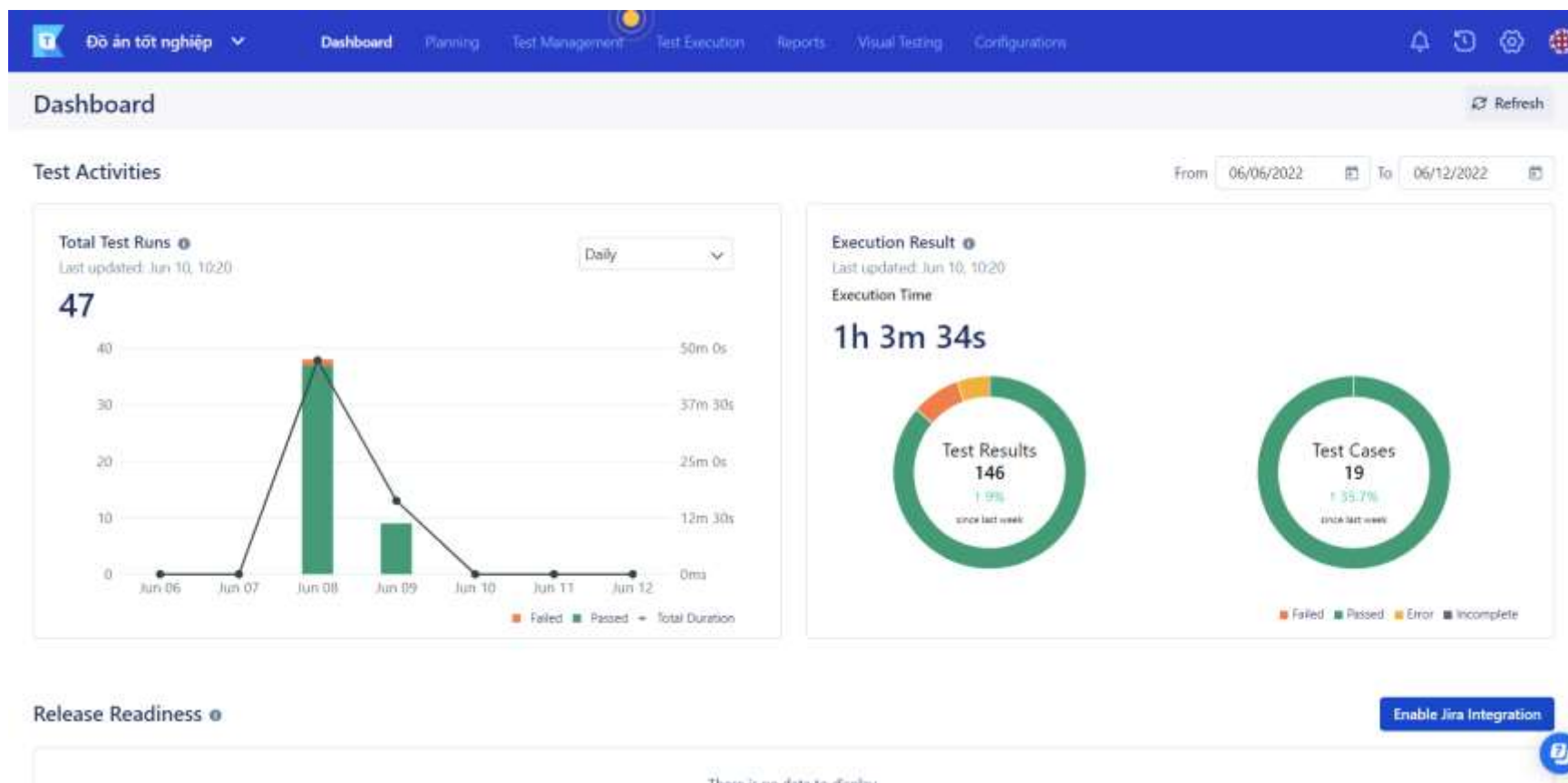
4.5.3 Báo cáo trên katalon studio



The screenshot displays the Katalon Studio interface, specifically the 'Test Suites' tab. The left sidebar shows a tree view of 'Uploaded Data' containing folders for 'Đăng ký', 'Đăng nhập', 'DH', 'DN', and 'TimiGem'. The main area shows a table of test suites. The table has columns for Name, Path, Last updated, Last executed, and a summary of results (Pass, Fail, Error, Skip). The results are shown as colored circles with counts: Green for Pass, Red for Fail, Yellow for Error, and Grey for Skip.

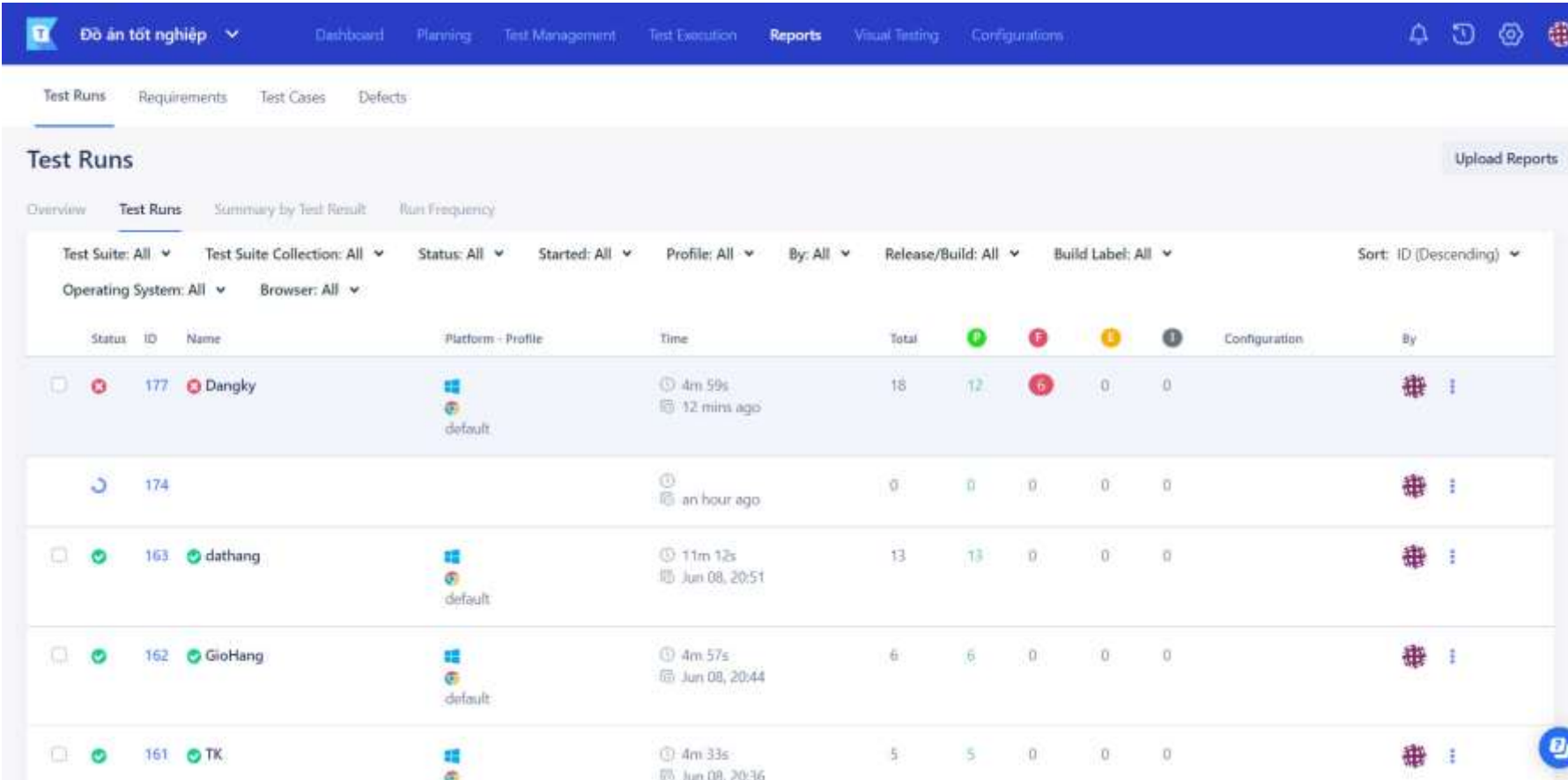
Name	Path	Last updated	Last executed	Pass	Fail	Error	Skip
Dangky	Uploaded Data/Test Suites	5 hours ago	5 hours ago	12	6	0	0
dathang	Uploaded Data/Test Suites	Jun 08, 21:03	Jun 08, 20:51	13	0	0	0
GioHang	Uploaded Data/Test Suites	Jun 08, 20:51	Jun 08, 20:44	6	0	0	0
TK	Uploaded Data/Test Suites	Jun 08, 20:43	Jun 08, 20:36	5	0	0	0
Dangnhap	Uploaded Data/Test Suites	Jun 08, 20:27	Jun 08, 20:23	0	0	0	0
New Test Suite	Uploaded Data/Test Suites	Jun 08, 20:12	Jun 08, 20:09	4	0	0	0

Hình 4. 22 Báo cáo kết quả Test Suite trên katalon -Test Ops



Hình 4. 23 Báo cáo Test Run-Execution Result trên Katalon-Test Ops

Kiểm thử tự động cho website camera Samtech sử dụng Katalon studio và Jmeter



The screenshot displays the Katalon Test Ops interface, specifically the 'Reports' section under 'Test Runs'. The top navigation bar includes links for Dashboard, Planning, Test Management, Test Execution, Reports (active), Visual Testing, and Configurations. Below the navigation bar, there are tabs for Test Runs, Requirements, Test Cases, and Defects. The 'Test Runs' tab is selected, and the 'Overview' sub-tab is active. The main content area shows a list of test runs with various filters and a table of results.

Filters:

- Test Suite: All
- Test Suite Collection: All
- Status: All
- Started: All
- Profile: All
- By: All
- Release/Build: All
- Build Label: All
- Sort: ID (Descending)
- Operating System: All
- Browser: All

Test Runs Table:

Status	ID	Name	Platform - Profile	Time	Total	P	F	E	I	Configuration	By
	177	Dangky	Windows default	4m 59s 12 mins ago	18	12	6	0	0		
	174			an hour ago	0	0	0	0	0		
	163	dathang	Windows default	11m 12s Jun 08, 20:51	13	13	0	0	0		
	162	GioHang	Windows default	4m 57s Jun 08, 20:44	6	6	0	0	0		
	161	TK	Windows default	4m 33s Jun 08, 20:36	5	5	0	0	0		

Hình 4. 24. Báo cáo chi tiết Test Run Trên Katalon- Test Ops

4.5.4 Báo cáo Jmeter

Bảng 4. 5 Báo Cáo Jmeter

STT		Tên Chức Năng	Số Lượng Người Dùng	Set the ramp-up period	Tiêu Chí Hiệu Năng	Kết quả thực tế			
						Lần 1	Lần 2	Lần 3	AVG
1	Load test	Đăng ký	10	0	Min Response Time	104	110	112	108.66666667
					Max Response Time	154	137	322	204.33333333
					Average Response time	129	127	176	144
					Throughput (KB/sec)	64.9	72.5	30.8	56.06666667
					Error Rate (%)	0	0	0	0
					Pass				
2	Load test	Đăng ký	100	0	Min Response Time	234	201	206	213.66666667
					Max Response Time	4166	1846	1931	2647.666667
					Average Response time	585	514	625	574.66666667
					Throughput (KB/sec)	23.1	48.9	48.9	40.3
					Error Rate (%)	0	0	0	0
					Pass				

STT		Tên Chức Năng	Số Lượng Người Dùng	Set the ramp-up period	Tiêu Chí Hiệu Năng	Kết quả thực tế			
						Lần 1	Lần 2	Lần 3	AVG
3	Stress test	Đăng ký	500	0	Min Response Time	324	107	270	233.6666667
					Max Response Time	21060	21075	21048	21061
					Average Response time	7115	6832	7324	7090.333333
					Throughput (KB/sec)	23.3	23.3	23.3	23.3
					Error Rate (%)	18.2	19.8	17.6	18.53333333
			Fail						
4	Load test	Đăng nhập	10	0	Min Response Time	100	99	96	98.33333333
					Max Response Time	332	125	332	263
					Average Response time	239	118	216	191
					Throughput (KB/sec)	30.1	78.7	29.9	46.23333333
					Error Rate (%)	0	0	0	0
			Pass						

STT		Tên Chức Năng	Số Lượng Người Dùng	Set the ramp-up period	Tiêu Chí Hiệu Năng	Kết quả thực tế			
						Lần 1	Lần 2	Lần 3	AVG
5	Load test	Đăng nhập	100	0	Min Response Time	394	165	92	217
					Max Response Time	1553	1781	4177	2503.666667
					Average Response time	939	668	977	861.3333333
					Throughput (KB/sec)	63	55.1	23.8	47.3
					Error Rate (%)	3	1	1	1.666666667
					Fail				
6	Stress test	Đăng nhập	500	0	Min Response Time	289	110	137	178.6666667
					Max Response Time	21044	21052	21056	21050.66667
					Average Response time	7745	7975	1650	5790
					Throughput (KB/sec)	23.5	23.6	23.4	23.5
					Error Rate (%)	33.2	37.4	33.6	34.73333333
					Fail				

STT		Tên Chức Năng	Số Lượng Người Dùng	Set the ramp-up period	Tiêu Chí Hiệu Năng	Kết quả thực tế			
						Lần 1	Lần 2	Lần 3	AVG
7	Load test	Giỏ hàng	10	0	Min Response Time	67	63	66	65.33333333
					Max Response Time	88	279	79	148.6666667
					Average Response time	80	211	72	121
					Throughput (KB/sec)	111.1	35.7	126.6	91.13333333
					Error Rate (%)	0	0	0	0
					Pass				
8	Load test	Giỏ hàng	100	0	Min Response Time	164	160	537	287
					Max Response Time	1111	1130	1120	1120.333333
					Average Response time	474	420	746	546.6666667
					Throughput (KB/sec)	83.8	83.1	85.4	84.1
					Error Rate (%)	15	18	15	16
					Fail				

STT		Tên Chức Năng	Số Lượng Người Dùng	Set the ramp-up period	Tiêu Chí Hiệu Năng	Kết quả thực tế			
						Lần 1	Lần 2	Lần 3	AVG
9	Stress test	Giỏ hàng	500	0	Min Response Time	256	219	233	236
					Max Response Time	21050	21066	21038	21051.33333
					Average Response time	7598	7767	7899	7754.666667
					Throughput (KB/sec)	23.5	23.6	23.5	23.53333333
					Error Rate (%)	34.2	35.2	32.8	34.06666667
			Fail						
10	Load test	Đặt hàng	10	0	Min Response Time	115	97	100	104
					Max Response Time	821	110	136	355.6666667
					Average Response time	198	102	108	136
					Throughput (KB/sec)	12.2	90.1	73.5	58.6
					Error Rate (%)	0	0	0	0
			Pass						

STT		Tên Chức Năng	Số Lượng Người Dùng	Set the ramp-up period	Tiêu Chí Hiệu Năng	Kết quả thực tế			
						Lần 1	Lần 2	Lần 3	AVG
11	Load test	Đặt hàng	100	0	Min Response Time	137	144	194	158.3333333
					Max Response Time	1142	1770	1196	1369.333333
					Average Response time	555	573	566	564.6666667
					Throughput (KB/sec)	80.1	54.2	78.8	71.03333333
					Error Rate (%)	6	12	9	9
					Fail				
12	Stress test	Đặt hàng	500	0	Min Response Time	177	142	337	218.6666667
					Max Response Time	21057	21065	21041	21054.33333
					Average Response time	7464	7891	7396	7583.666667
					Throughput (KB/sec)	23.4	23.5	23.5	23.46666667
					Error Rate (%)	29.6	30.4	29.2	29.73333333
					Fail				

STT		Tên Chức Năng	Số Lượng Người Dùng	Set the ramp-up period	Tiêu Chí Hiệu Năng	Kết quả thực tế			
						Lần 1	Lần 2	Lần 3	AVG
13	Performance test	10 người truy cập 1s sau 20 người truy cập 3s sau 50 người truy cập (cùng trang web)	80	0s-1s-3s	Min Response Time	149	141	167	152.3333333
					Max Response Time	1992	15389	15892	11091
					Average Response time	755	2336	1495	1528.666667
					Throughput (KB/sec)	17.1	4.3	4.7	8.7
					Error Rate (%)	0	0	0	0
					Pass				
14	Performance test	10 người truy cập trang đăng nhập 1s sau 20 người truy cập trang chủ 3s sau 50 người truy cập giờ hàng	80	0s-1s-3s	Min Response Time	96	87	94	92.33333333
					Max Response Time	1070	3563	1530	2054.333333
					Average Response time	348	475	538	453.6666667
					Throughput (KB/sec)	19.6	15.1	20.7	18.46666667
					Error Rate (%)	13.75	12.5	16.25	14.16666667
					Fail				

STT		Tên Chức Năng	Số Lượng Người Dùng	Set the ramp-up period	Tiêu Chí Hiệu Năng	Kết quả thực tế			
						Lần 1	Lần 2	Lần 3	AVG
15	Performance test	Kiểm tra 10 ng truy cập đồng thời các trang web(đăng nhập, đăng ký, giỏ hàng, trang chủ)	40	0	Min Response Time	104	113	85	100.6666667
					Max Response Time	2372	1305	1214	1630.333333
					Average Response time	449	269	571	429.6666667
					Throughput (KB/sec)	16.7	30.3	32.7	26.56666667
					Error Rate (%)	0	0	0	0
					Pass				
16	Performance test	Kiểm tra 1000 ng truy cập trang chủ trong 1s	1000	1	Min Response Time	106	260	472	279.3333333
					Max Response Time	21060	21052	28453	23521.66667
					Average Response time	12111	12078	12174	12121
					Throughput (KB/sec)	45.4	45.4	34.4	41.73333333
					Error Rate (%)	36.9	37.6	35	36.5
					Fail				

Bảng 4. 6 Bảng hình ảnh minh chứng Jmeter

Đăng ký	
10 ng lần 1	https://prnt.sc/lMaDQrLc7oPs
10 ng lần 2	https://prnt.sc/LPPnFjBZoUm4
10 ng lần 3	https://prnt.sc/c-CWM0BWvAtH
100 ng lần 1	https://prnt.sc/B8LbasgZhKXq
100 ng lần 2	https://prnt.sc/XMufm-9mhxJH
100 ng lần 3	https://prnt.sc/QUlJfeWiR2-A
500ng lần 1	https://prnt.sc/G9YpBbajRUOJ
500ng lần 2	https://prnt.sc/oDrRznL-BL_Z
500ng lần 3	https://prnt.sc/xM-mVbZ4gWrz
Đăng nhập	
10 ng lần 1	https://prnt.sc/VQG4rN_oA7rb
10 ng lần 2	https://prnt.sc/LA8h45fHmztO
10 ng lần 3	https://prnt.sc/YybC9z7FDxJ2
100 ng lần 1	https://prnt.sc/nlLRwazYYYwP
100 ng lần 2	https://prnt.sc/akQs5T52ygd3
100 ng lần 3	https://prnt.sc/6npBB2H8vhnF
500ng lần 1	https://prnt.sc/Tt5q1BA1mxlv

500ng lần 2	https://prnt.sc/Qrw4AKCfoET0
500ng lần 3	https://prnt.sc/hyclCZeskr_W
Giỏ hàng	
10 ng lần 1	https://prnt.sc/6TvzOetsJJQT
10 ng lần 2	https://prnt.sc/f443ivzYhS6g
10 ng lần 3	https://prnt.sc/OFabllqmO2Fh
100 ng lần 1	https://prnt.sc/RTz1YeXOCiYs
100 ng lần 2	https://prnt.sc/3QQdxRkYozVS
100 ng lần 3	https://prnt.sc/SJQjRslisNqr
500ng lần 1	https://prnt.sc/SZZiKOYhzwew
500ng lần 2	https://prnt.sc/VVSXtCw6ZG60
500ng lần 3	https://prnt.sc/rM4KAZyz8H3f
Đặt hàng	
10 ng lần 1	https://prnt.sc/Pueb6WA0cfSl
10 ng lần 2	https://prnt.sc/AZT56vqGST2D
10 ng lần 3	https://prnt.sc/FMZ70SfNdyJF
100 ng lần 1	https://prnt.sc/Uf2nFk4vf5Y-
100 ng lần 2	https://prnt.sc/wttehVMyyL1k
100 ng lần 3	https://prnt.sc/QDVss_ctGPN2
500ng lần 1	https://prnt.sc/hX5zZ-aEJO2e
500ng lần 2	https://prnt.sc/WnA6ullCqaKC
500ng lần 3	https://prnt.sc/x1IHfpZ4QKEg

10 người truy cập 1s sau 20 người truy cập 3s sau 50 người truy cập (cùng trang web)	
Lần 1	https://prnt.sc/AAre6MH48nn8
Lần 2	https://prnt.sc/Plo-Vu7n-K9d
Lần 3	https://prnt.sc/BWJPOOkbSQxg
10 người truy cập trang đăng nhập 1s sau 20 người truy cập trang chủ 3s sau 50 người truy cập giỏ hàng	
lần 1	https://prnt.sc/GWvxVXndpd4W
lần 2	https://prnt.sc/-WFdLoz0UWDi
lần 3	https://prnt.sc/1jw_yfENxfHl
Kiểm tra 10 ng truy cập đồng thời các trang web(đăng nhập, đăng ký, giỏ hàng, trang chủ)	
Lần 1	https://prnt.sc/0vfyl6MMXVJ
Lần 2	https://prnt.sc/rTagkaOF0Syb
Lần 3	https://prnt.sc/8O3Kq2nBKDOL
Kiểm tra 1000 ng truy cập trong 1s	
Lần 1	https://prnt.sc/jLAyD4YQnHgH
Lần 2	https://prnt.sc/FDiUIZI3bwsg
Lần 3	https://prnt.sc/WGhPVJEC68K7

4.5.5 Test Report

TEST REPORT

Project Name	<i>Kiểm thử tự động cho website camera Samtech sử dụng Katalon studio và Jmeter</i>	Creator	<i>Đoàn Hương Ly</i>
Project Code	<i>Đoàn Hương Ly</i>	Reviewer/ Approver	<i>Ngô Thanh Huyền</i>
Document Name		Issue Date	<i>1/6/2022</i>
Notes			

No	Module code	Pass	Fail	Untested	N/A	Number of test cases
1	TC Đăng ký	21	7	0	0	28
2	TC Giỏ Hàng	6	0	0	1	7
3	TC Tìm Kiếm	5	0	0	1	6
4	TC Đăng Nhập	9	0	0	5	14
5	TC Đặt Hàng	15	0	0	9	24
6	TC Hiệu Năng	7	9	0	0	16
Sub total		63	16	0	16	95

Test coverage 100.00 %

Test successful coverage 66.32 %

Test fail coverage 16.84 %

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết quả đạt được

Sau quá trình làm việc, đề tài đã kiểm thử tự động được một số chức năng cơ bản của trang web, sử dụng được công cụ kiểm thử tự động vào kiểm tra. Vận dụng các kiến thức đã học để thực thi kiểm thử 1 trang web bao gồm tất cả các giai đoạn. Các chức năng đã chạy được kiểm thử tự động: Đăng nhập, Đăng ký, Giỏ hàng, Tìm kiếm, Đặt hàng.

Hạn chế của đề tài

Do trình độ, khả năng và thời gian còn hạn chế nên báo cáo còn tồn tại một số điểm hạn chế: - Dễ nảy sinh lỗi trong code, và rất khó sửa.

- Mặc dù đã tìm tòi, cũng như chủ động trong việc tìm kiếm nguồn tài liệu, báo cáo nước ngoài bằng tiếng Anh để nghiên cứu, nhưng việc thể hiện lại bằng tiếng Việt còn nhiều bối rối khi chưa truyền tải chính xác các thuật ngữ chuyên ngành. - Trong khuôn khổ báo cáo chưa thể trình bày đầy đủ các tính năng, và nghiên cứu các kỹ thuật nâng cao được sử dụng trong Katalon Studio như test API.

Hướng phát triển của đề tài

Hướng phát triển tiếp theo em sẽ tiếp tục tìm hiểu và áp dụng công cụ kiểm thử Katalon đầy đủ để thực hiện việc kiểm thử toàn bộ ứng dụng web, mobile một cách tổng thể không chỉ là kiểm thử chức năng, kiểm thử hiệu năng mà còn kiểm thử giao diện. Cần nghiên cứu sâu hơn về Katalon Studio trong lĩnh vực an toàn thông tin - Kết hợp công cụ kiểm thử tự động Katalon Studio với công cụ kiểm thử khác để có thể cải tiến được quy trình kiểm thử. Đảm bảo ứng dụng web có thể được kiểm tra về chức năng

TÀI LIỆU THAM KHẢO

- [1] Bộ môn CNPM, Đề cương “Kiểm thử phần mềm tự động”, ĐHSPKT Hưng Yên, 2016.
- [2] Bộ môn CNPM, Đề cương “Kiểm thử phần mềm”, ĐHSPKT Hưng Yên, 2016.
- [3] Bộ môn CNPM, Đề cương “Đảm bảo chất lượng phần mềm”, ĐHSPKT Hưng Yên, 2016.
- [4] <https://viblo.asia/p/cac-mo-hinh-phat-trien-phan-mem-GrLZDwbGKk0>
- [5] <https://viblo.asia/p/tim-hieu-ve-framework-gioi-thieu-mot-so-frameworkpho-bien-m68Z0p2NZkG>
- [6] <https://viblo.asia/p/kiem-thu-tu-dong-va-cac-cong-cu-duoc-su-dung-trongkiem-thu-tu-dong-aWj531qbZ6m>
- [7] <https://www.guru99.com/keyword-driven-testing.html>
- [8] <https://www.softwaretestinghelp.com/keyword-driven-framework-inselenium/>
- [9] <https://www.devpro.edu.vn/kiem-thu-tu-dong-la-gi>
- [10] <https://vntesters.com/tu-dong-kiem-thu-thanh-cong-2>
- [11] Nguyễn Văn Vy - Nguyễn Việt Hà, Giáo trình Kỹ nghệ phần mềm, Nhà xuất bản Giáo dục Việt Nam, 2009
- [12] <https://viblo.asia/p/kiem-thu-tu-dong-va-kiem-thu-thu-cong-nensudung-khi-nao-EoDkQqEqkbV>