**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**BÀI TẬP LỚN**

**KIỂM THỬ PHẦN MỀM**

**Tên đề tài: KIỂM THỬ PHẦN MỀM – WEBSITE BÁN SÁCH ADCBOOK.NET.VN**

|  |  |
| --- | --- |
| Giáo viên: | Th.s Nguyễn Thái Cường |
| Lớp: | IT6084.3 |
| Nhóm: | 7 |
| Thành viên: | Nguyễn Mỹ Hạnh |
|  | Nguyễn Hồng Quân |
|  | Nguyễn Trọng Trường |
|  | Nguyễn Quang Vũ |
|  |  |

Hà Nội, năm 2024

LỜI CẢM ƠN

Ngày nay, với sự phát triển như vũ bão của công nghệ thông tin, những ứng dụng phần mềm đã trở thành một phần quan trọng của cuộc sống hàng ngày. Các phần mềm được áp dụng phổ biến trong mọi ngành nghề và cả trong đời sống. Để có những phần mềm, ứng dụng chất lượng không thể thiếu việc kiểm thử phần mềm trong quá trình phát triển phần mềm. Kiểm thử có thể coi là một khâu không thể thiếu trong mỗi giai đoạn phát triển mỗi chức năng của một phần mềm hay ứng dụng di động. Do đó, chúng em đã quyết định thực hiện đề tài: “Kiểm thử phần mềm – website bán sách adcbook.net.vn”.

Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến quý thầy cô khoa Công nghệ thông tin – trường Đại học Công nghiệp Hà Nội đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian diễn ra môn học. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của chúng em mới có thể hoàn thiện tốt đẹp.

Đặc biệt, chúng em xin chân thành cảm ơn thầy Nguyễn Thái Cường – người thầy đã trực tiếp dạy bảo, giúp đỡ tận tình, quan tâm, hướng dẫn hết mình và tạo mọi điều kiện thuận lợi cho nhóm trong suốt quá trình học tập và hoàn thiện đề tài nghiên cứu.

Kiến thức của chúng em còn hạn chế nên không tránh khỏi những thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp quý báu của thầy để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao kiến thức của mình.

*Chúng em xin chân thành cảm ơn!*

MỤC LỤC

[Chương 1. Giới thiệu kiểm thử phần mềm 3](#_Toc166017525)

[1.1 Tổng quan về kiểm thử phần mềm 3](#_Toc166017526)

[1.2 Quy trình kiểm thử phần mềm 3](#_Toc166017527)

[1.3 Các cấp độ kiểm thử 4](#_Toc166017528)

[1.4 Các loại hình kiểm thử 6](#_Toc166017529)

[1.5 Các kỹ thuật kiểm thử 7](#_Toc166017530)

[Chương 2. Giới thiệu công cụ LoadForcus 10](#_Toc166017531)

[2.1 Giới thiệu công cụ 10](#_Toc166017532)

[2.2 Đặc điểm 10](#_Toc166017533)

[2.3 Cài đặt và sử dụng công cụ LoadForcus 13](#_Toc166017534)

[Chương 3. Giới thiệu phần mềm adcbook.net.vn 15](#_Toc166017535)

[3.1 Lý do chọn đề tài 15](#_Toc166017536)

[3.2 Phân tích thiết kế hệ thống 15](#_Toc166017537)

[3.2.1 Đặc tả hệ thống 15](#_Toc166017538)

[3.2.2 Thiết kế hệ thống 15](#_Toc166017539)

[Chương 4. Kiểm thử phần mềm adcbook.net.vn 16](#_Toc166017540)

[4.1 Lập kế hoạch kiểm thử (test plan) 16](#_Toc166017541)

[4.2 Thực hiện kiểm thử 16](#_Toc166017542)

[4.2.1 Nguyễn Mỹ Hạnh - <Tên chức năng kiểm thử> 16](#_Toc166017543)

[4.2.2 Nguyễn Hồng Quân – 16](#_Toc166017544)

[4.2.3 Nguyễn Trọng Trường - 16](#_Toc166017545)

[4.2.4 Nguyễn Quang Vũ - 16](#_Toc166017546)

[Phụ lục 17](#_Toc166017547)

[Phụ lục hình ảnh 17](#_Toc166017548)

[Phụ lục bảng biểu 17](#_Toc166017549)

[Tài liệu tham khảo 18](#_Toc166017550)

# Giới thiệu kiểm thử phần mềm

## Tổng quan về kiểm thử phần mềm

Kiểm thử phần mềm là quá trình đánh giá một phần mềm hoặc một hệ thống để xác định xem chúng hoạt động như mong đợi hay không.

Trong quy trình phát triển phần mềm xuất hiện 3 yếu tố chính ảnh hưởng đến chất lượng phần mềm (con người, quy trình, chất lượng) gồm: khoảng cách giữa yêu cầu người dùng và bản đặc tả, khoảng cách giữa bản đặc tả và sản phẩm, và khoảng cách giữa yêu cầu người dùng và sản phẩm. Ví dụ, khi có sự thay đổi yêu cầu trong quá trình phát triển nhưng không được cập nhật vào bản đặc tả. Điều này có thể dẫn đến việc phần mềm không đáp ứng đúng yêu cầu của khách hàng. Để đảm bảo chất lượng phần mềm, kiểm thử phần mềm là rất quan trọng:

* Kiểm thử giúp tìm ra nhiều lỗi trong giới hạn thời gian và nguồn lực có sẵn.
* Nó chứng minh tính phù hợp của sản phẩm với đặc tả và đảm bảo chất lượng với chi phí và nỗ lực tối thiểu.
* Thiết kế tài liệu kiểm thử có hệ thống và thực hiện một cách hiệu quả và tiết kiệm thời gian.

## Quy trình kiểm thử phần mềm

Quy trình kiểm thử phần mềm một tập các hoạt động, các phương thức mà con người phải làm để thực hiện việc kiểm thử cho một phần mềm hay một hệ thống phần mềm.

Quy trình kiểm thử phần mềm được chia thành 4 giai đoạn:

* Lập kế hoạch kiểm thử (Test planning): là quá trình tạo ra bản kế hoạch kiểm thử.
* Phân tích và thiết kế kiểm thử (Test analysis and design): là hoạt động chuyển các mục tiêu của kiểm thử thành các trường hợp kiểm thử cụ thể.
* Thực hiện kiểm thử (Test executing): là quá trình mà các bộ kiểm thử được thực hiện trên phần mềm thực tế. Điều này bao gồm chạy các kịch bản kiểm thử, ghi lại kết quả kiểm thử, xác minh các kết quả và ghi lại các lỗi và vấn đề phát hiện được.
* Báo cáo và đánh giá kiểm thử (Test report and evaluation): là hoạt động đánh giá kết quả kiểm thử so với mục tiêu đã đề ra.

Cấu trúc của bản kế hoạch kiểm thử cơ bản bao gồm 7 thành phần chính:

* Giới thiệu chung (Introduction).
* Các tiêu chí chấp nhận sản phẩm (Acceotance criteria).
* Các yêu cầu cần kiểm thử (Requirements for test).
* Chiến lược kiểm thử (Test strategy).
* Nguồn lực dành cho kiểm thử (Resources for testing).
* Các mốc kiểm thử (Test milestones).
* Sản phẩm cần bàn giao (Deliverables of test).

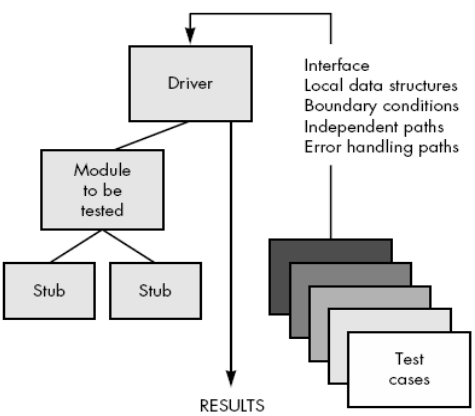
## Các cấp độ kiểm thử

Có 4 cấp độ kiểm thử, gồm:

* Kiểm thử đơn vị (Unit testing).
* Kiểm thử tích hợp (Integration testing).
* Kiểm thử hệ thống (System testing).
* Kiểm thử chấp nhận (Acceptance testing).

*Kiểm thử đơn vị (Unit testing):*

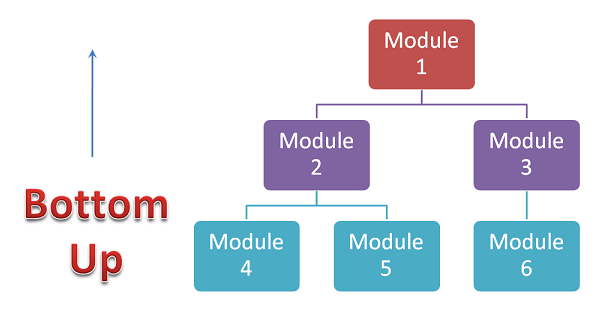
- Nhằm kiểm tra đơn vị thiết kế nhỏ nhất- một module phần mềm. Một module hoạt động thường có trao đổi thông tin với module mức dưới và mức trên nó, do đó phạm vi phát hiện lỗi liên quan chặt chẽ tới module này.



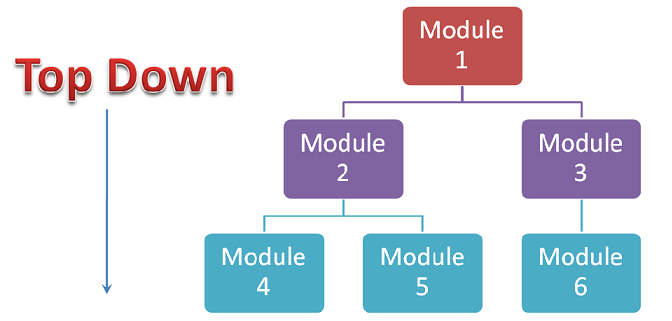
Hình 1.1 Mô hình kiểm thử đơn vị

*Kiểm thử tích hợp (Integration testing):*

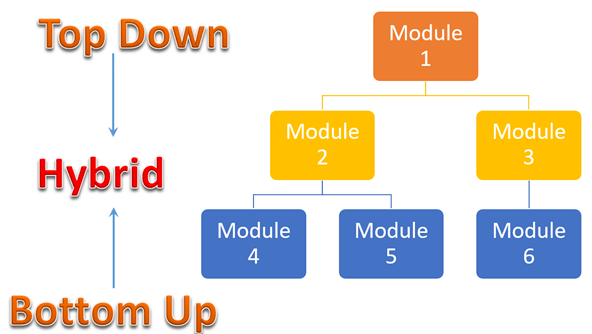
- Nhận các môđun đầu vào đã được kiểm thử đơn vị, nhóm chúng vào những tập hợp lớn hơn, áp dụng các ca kiểm thử đã được định nghĩa trong kế hoạch kiểm thử tích hợp vào tập hợp đó, và cung cấp đầu ra cho hệ thống tích hợp.



Hình 1.2 Mô hình chiến lược tích hợp từ dưới lên (Bottom-up testing)

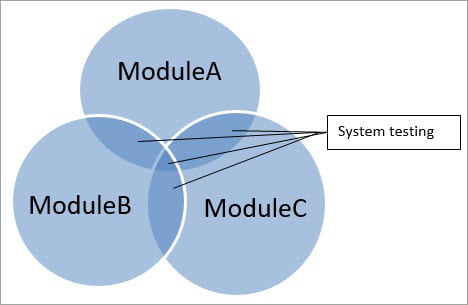


Hình 1.3 Mô hình chiến lược tích hợp từ trên xuống (Top-down testing)



Hình 1.4 Mô hình chiến lược kết hợp (Sandwich testing)

*Kiểm thử hệ thống (System testing):* Kiểm thử thiết kế và toàn bộ hệ thống (sau khi tích hợp) có thỏa mãn yêu cầu đặt ra hay không tìm.



Hình 1.5 Mô hình kiểm thử hệ thống

*Kiểm thử chấp nhận (Acceptance testing):* Vận hành hệ thống trong môi trường của người sử dụng.

A diagram of a flowchart

Description automatically generated

Hình 1.6 Quy trình các bước thực hiện kiểm thử chấp nhận

## Các loại hình kiểm thử

*Kiểm thử chức năng:*

- Là 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. Đặc tả về bên ngoài của phần mềm là đặc tả chính xác về hành vi của phần mềm theo góc nhìn của người dùng thấy.

- Mục tiêu: đảm bảo đúng mục tiêu của kiểm thử chức năng nhập dữ liệu – xử lý – lấy và kiểm tra kết quả trả về.

- Các loại kiểm thử chức năng:

* Kiểm thử chức năng của hệ thống.
* Kiểm thử tích hợp dữ liệu và cơ sở dữ liệu.
* Kiểm thử vòng lặp công việc.
* Kiểm thử truy cập.
* Kiểm thử giao diện.

*Kiểm thử phi chức năng:*

- Tập trung vào kiểm thử sản phẩm, hệ thống phần mềm cần kiểm thử có những đặc tính tốt như thế nào.

- Sử dụng hiệu quả nhất trong kiểm thử hệ thống và kiểm thử chấp nhận sản phẩm.

- Có 4 loại kiểm thử phi chức năng thường dùng:

* Kiểm thử hiệu năng.
* Kiểm thử tải trọng.
* Kiểm thử tập trung.
* Kiểm thử với dữ liệu lớn.

*Kiểm thử liên quan đến sự thay đổi:*

- Thực hiện hoạt động kiểm thử khi có sự thay đổi trên hoặc trong sản phẩm phần mềm.

- Sự thay đổi của sản phẩm phần mềm có thể là:

* Sửa chữa các lỗi tìm được.
* *S*ản phẩm được nâng cấp được thay đổi về chức năng.

- Gồm: kiểm thử lại và kiểm thử hồi quy.

## Các kỹ thuật kiểm thử

*Kiểm thử hộp đen:*

- Kiểm thử hộp đen (Black box testing) là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó.

- Đối tượng được kiểm thử*:* Là thành phần phần mền (TPPM) có thể là 1 hàm chức năng, 1 modul chức năng, 1 phân hệ chức năng…

- Phương pháp thử nghiệm*:* Dựa vào chức năng Kiểm thử hộp đen (Black box test) có thể được áp dụng hầu như đến mọi cấp độ của kiểm thử phần mềm:

* Kiểm thử đơn vị (Unit testing).
* Kiểm thử tích hợp (Intergration testing).
* Kiểm thử hệ thống (System testing).
* Kiểm thử chấp nhận (Acceptance test).

- Đặc điểm:

* Là chiến lược kiểm thử TPPM dựa vào thông tin duy nhất là các đặc tả về yêu cầu chức năng của TPPM tương ứng.
* Người kiểm thử không cần thiết phải có kiến thức về việc mã hoá, cấu trúc bên trong của TPPM, cũng như không yêu cầu phải biết lâp trình phần mềm.
* Việc kiểm thử được tiến hành dựa vào việc kiểm thử TPPM làm được gì, có phù hợp với yêu cầu của người dùng hay không. Các tester nhập số liệu vào phần mềm và chỉ cần xem kết quả của phần mềm và các mục tiêu kiểm tra.
* Tester phải viết test case đầy đủ trước khi test; khi test, đơn giản chỉ cần thực hiện theo các bước mô tả trong test case thao tác và nhập data vào, sau đó xem kết quả trả về hoặc hành vi của phần mềm, rồi so sánh với kết quả mong đọi được viết trong testcase.
* Khi viết test case: Dựa vào yêu cầu và giao diện bên ngoài của chương trình (Không can thiệp vào bên trong code của chương trình).
* Khi thực hiện test: Thực hiện trên giao diện của chương trình (yêu cầu chương trình phải chạy được mới test được, không can thiệp vào code).

*Kiểm thử hộp trắng:*

- Kiểm thử hộp trắng (While box testing) là phương pháp thử nghiệm phần mềm, trong đó các thiết kế, cấu trúc giải thuật bên trong, và việc thực hiện các công việc đều được biết đến.

- Đối tượng kiểm thử: Là 1 thành phần của phần mềm (1 chức năng, 1 module chức năng, 1 phân hệ chức năng....).

- Phương pháp thử nghiệm: Dựa vào thuật giải Kiểm thử hộp trắng dựa vào thuật giải cụ thể, vào cấu trúc dữ liệu bên trong của đơn vị phần mềm cần kiểm thử để xác Định đơn vị phần mềm đó có thực hiện đúng không. Kỹ thuật white box test thích hợp dùng để kiểm thử đơn vị (Unit test).

- Đặc điểm:

* Là chiến lược kiểm thử TPPM dựa vào giải thuật, cấu trúc bên trong chức năng của TPPM tương ứng.
* Người kiểm thử phải có kiến thức nhất định về việc mã hoá, cấu trúc bên trong của chức năng, biết lâp trình phần mềm.
* Việc kiểm thử được tiến hành dựa vào việc kiểm xem giải thuật, mã lệnh đã làm có đúng không.
* Mức test này thường yêu cầu các tester phải viết test case đầy đủ các nhánh trong code; khi test, sẽ set điều kiện và data để chạy vào đủ tất cả các nhánh trong giải thuật, đảm bảo thực hiện đầy đủ.
* Khi viết test case: Dựa vào yêu cầu và nội dung Source Code (can thiệp vào bên trong Code của chương trình).
* Khi thực hiện test: Thực thi test trong code (không cần thực thi chương trình, vì thực hiện test white box sẽ sử dụng framework nào đó hỗ trợ (Ví dụ như test kiểu debug).
* Trong kiểm tra này, đòi hỏi người tester phải có kiến thức và kỹ năng nhất định về ngôn ngữ lập trình được dùng, hiểu thuật giải trong thành phần phần mềm, để có thể hiểu được chi tiết về đoạn code cần kiểm thử.

# Giới thiệu công cụ LoadForcus

## Giới thiệu công cụ

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

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

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

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

Công cụ kiểm thử LoadFocus là nền tảng thử nghiệm đám mây tất cả trong một dành cho các trang web và API với chức năng kiểm tra tải Apache JMeter và theo dõi tốc độ trang.

## Đặc điểm

Công cụ kiểm thử LoadFocus là một công cụ mã nguồn mở. LoadFocus hỗ trợ 3 loại kiểm thử chủ yếu:

*LoadTesting:*

* Load testing là một loại kiểm thử hiệu năng được sử dụng để đánh giá hiệu suất của một trang web, ứng dụng hoặc giao diện lập trình ứng dụng (API) dưới điều kiện tải bình thường và tải cao. Mục đích của load testing là xác định hành vi và thời gian phản hồi của hệ thống khi phải xử lý số lượng người dùng hoặc yêu cầu tải lớn, đảm bảo tính ổn định, đáng tin cậy và khả năng mở rộng của nó.
* Kiểm thử tải bao gồm mô phỏng một lượng lớn người dùng đồng thời, yêu cầu hoặc giao dịch để đánh giá hiệu suất hệ thống và xác định các vấn đề về hiệu suất hoặc hạn chế tiềm ẩn. Bằng cách tạo ra các loại tải khác nhau, kiểm thử tải có thể đo lường các thông số như thời gian phản hồi, khả năng xử lý, tài nguyên sử dụng và tính ổn định của hệ thống dưới các mức tải khác nhau.
* Kiểm thử tải có thể được áp dụng cho các trang web, ứng dụng web, ứng dụng di động, API hoặc bất kỳ hệ thống nào tương tác với người dùng hoặc xử lý số lượng lớn dữ liệu. Nó giúp phát hiện các vấn đề liên quan đến hiệu suất như thời gian phản hồi chậm, rò rỉ bộ nhớ, vấn đề cơ sở dữ liệu, yếu tố cấu hình yếu kém hoặc phân bổ tài nguyên không đủ.
* Quá trình kiểm thử load testing thông thường bao gồm các bước sau:

1. Lập kế hoạch kiểm thử
2. Thiết kế kiểm thử
3. Thực hiện kiểm thử
4. Phân tích hiệu suất
5. Tối ưu hóa và kiểm thử lại

* Kiểm thử Load Testing mang lại nhiều lợi ích:
* Xác nhận hiệu suất: Kiểm thử tải giúp xác nhận các đặc điểm hiệu suất của một hệ thống, đảm bảo thời gian phản hồi và khả năng xử lý theo yêu cầu trong các tình huống tải khác nhau.
* Đánh giá khả năng mở rộng: Bằng cách mô phỏng các tải tăng dần, kiểm thử tải giúp đánh giá khả năng mở rộng và khả năng xử lý yêu cầu người dùng gia tăng. Nó giúp xác định khả năng tối đa của hệ thống và quyết định khi nào cần thêm tài nguyên hoặc nâng cấp cơ sở hạ tầng.
* Xác định điểm hạn chế: Kiểm thử tải xác định các vấn đề về hiệu suất, chẳng hạn như các truy vấn cơ sở dữ liệu chậm, thuật toán không hiệu quả hoặc vấn đề về độ trễ mạng. Thông tin này giúp nhà phát triển tối ưu hóa hệ thống để cải thiện hiệu suất.
* Tính đáng tin cậy và ổn định: Kiểm thử tải phát hiện các vấn đề về ổn định có thể xảy ra khi tải nặng, chẳng hạn như rò rỉ bộ nhớ, sự cố hoặc việc sử dụng tài nguyên quá mức. Bằng cách xác định và khắc phục các vấn đề này, hệ thống trở nên đáng tin cậy và ổn định hơn.

1. *Jmeter Load Testing:*

* JMeter Load Testing (Kiểm thử tải bằng JMeter) là một phương pháp kiểm thử hiệu năng sử dụng công cụ mã nguồn mở Apache JMeter để đánh giá hiệu suất và khả năng mở rộng của một ứng dụng web hoặc API. JMeter được sử dụng rộng rãi cho kiểm thử tải vì nó cho phép người dùng tạo và thực thi các kịch bản kiểm thử để mô phỏng các hành động của người dùng và tạo ra một lượng lớn yêu cầu.
* Một tính năng của JMeter Load Testing là khả năng chạy các kịch bản kiểm thử từ nhiều địa điểm khác nhau. Tính năng này có lợi khi đánh giá hiệu suất và thời gian phản hồi của ứng dụng từ các địa điểm địa lý khác nhau, cung cấp thông tin về cách ứng dụng hoạt động dưới các điều kiện mạng và địa lý đa dạng.
* Dưới đây là một tổng quan về cách JMeter Load Testing hoạt động:
* Tạo kịch bản kiểm thử: Trong JMeter, bạn có thể tạo các kịch bản kiểm thử mô phỏng các hành động của người dùng như điều hướng qua các trang web, gửi biểu mẫu hoặc tương tác với các API. Các kịch bản này xác định hành vi người dùng và tải mà bạn muốn tạo ra trong quá trình kiểm thử.
* Cấu hình kiểm thử: Trong JMeter, bạn có thể cấu hình các tham số khác nhau như số lượng người dùng đồng thời, thời gian tăng dần và thời gian thực thi kiểm thử. Các thiết lập này xác định mức tải và thời lượng kiểm thử.
* Kiểm thử phân tán: JMeter cho phép phân tán tải trên nhiều máy chủ, cho phép bạn chạy các kịch bản kiểm thử từ các địa điểm khác nhau cùng một lúc. Mỗi máy chủ đại diện cho một địa điểm từ đó tải được tạo ra.
* Cấu hình mạng: Để thực thi kiểm thử từ nhiều địa điểm, bạn cần cấu hình mạng một cách phù hợp. Điều này bao gồm thiết lập cơ sở hạ tầng mạng để đảm bảo giao tiếp giữa các phiên bản JMeter đang chạy trên các máy chủ khác nhau.
* Thực thi kiểm thử: Sau khi kịch bản kiểm thử và cấu hình mạng đã sẵn sàng, bạn có thể bắt đầu kiểm thử tải. JMeter sẽ thực thi kịch bản từ mỗi địa điểm, mô phỏng các hành động của người dùng và tạo ra mức tải mong muốn trên ứng dụng hoặc API mục tiêu.
* Phân tích kết quả: Sau khi kiểm thử tải hoàn thành, JMeter cung cấp các chỉ số hiệu suất và kết quả chi tiết. Các chỉ số này bao gồm thời gian phản hồi, công suất xử lý, tỷ lệ lỗi và sử dụng tài nguyên. Bạn có thể phân tích các kết quả này để xác định các vấn đề về hiệu suất, hạn chế hệ thống và vùng cần tối ưu.

1. *Page Speed Monitoring:*

* Page speed monitoring là quá trình liên tục theo dõi và đo lường hiệu suất và tốc độ tải trang của một trang web. Nó liên quan đến việc theo dõi các chỉ số hiệu suất khác nhau và thu thập thông tin để tối ưu tốc độ trang web và cải thiện trải nghiệm người dùng. Một công cụ thông dụng được sử dụng để Giám sát Tốc độ Trang là công cụ kiểm thử Load Focus
* Dưới đây là tổng quan về cách Page speed Monitoring hoạt động :
* Performance Metrics: Loadforcus đo đạc một số chỉ số hiệu suất quan trọng của trang web, bao gồm thời gian tải trang, first contentful paint (FCP), largest contentful paint (LCP), thời gian tương tác (TTI) và nhiều hơn nữa. Những chỉ số này cung cấp thông tin về tốc độ tải trang và tương tác của trang đối với người dùng.
* Test Execution: Để giám sát tốc độ trang bằng Loadforcus, bạn có thể chạy kiểm tra trên các trang web cụ thể hoặc thiết lập giám sát tự động để đánh giá hiệu suất theo thời gian. Loadforcus báo cáo chi tiết với điểm hiệu suất và thông tin hành động cho mỗi trang đã được kiểm tra.
* Insights and Recommendations: cung cấp nhiều thông tin và đề xuất dựa trên các kiểm tra đã thực hiện. Những thông tin này giúp xác định các vấn đề về hiệu suất, vấn đề truy cập và các lĩnh vực mà các tiêu chuẩn tốt nhất có thể được thực hiện.
* Historical Performance Tracking: Bằng cách thường xuyên giám sát trang bằng Loadforcus, bạn có thể theo dõi hiệu suất lịch sử của trang web của mình theo thời gian.
* Lợi ích của giám sát tốc độ trang bằng Loadforcus bao gồm:
* Cải thiện Trải nghiệm Người dùng: Bằng cách giám sát tốc độ trang và thực hiện các tối ưu hóa được đề xuất bởi Lighthouse, bạn có thể cung cấp một trải nghiệm nhanh hơn và mượt mà cho người truy cập trang web của bạn. Thời gian tải trang nhanh giúp giảm tỷ lệ thoát, tăng cường tương tác và cải thiện sự hài lòng của người dùng.
* Tối ưu hóa Công cụ tìm kiếm (SEO): Tốc độ trang là một yếu tố quan trọng trong thứ hạng trên công cụ tìm kiếm. Giám sát và cải thiện tốc độ trang bằng Lighthouse giúp đảm bảo trang web của bạn đáp ứng các yêu cầu hiệu suất của các công cụ tìm kiếm, từ đó cải thiện khả năng tìm thấy và thứ hạng trên kết quả tìm kiếm.
* Tối ưu hóa Hiệu suất: Lighthouse cung cấp thông tin chi tiết và đề xuất để tối ưu hiệu suất của trang web của bạn. Bằng cách giải quyết các vấn đề được xác định, chẳng hạn như giảm kích thước tệp, tối ưu hóa việc tải tài nguyên và cải thiện chiến lược bộ nhớ cache, bạn có thể cải thiện đáng kể tốc độ và hiệu suất tổng thể của trang web.

## Cài đặt và sử dụng công cụ LoadForcus

* Công cụ kiểm thử Loadforcus là một nền tảng thử nghiệm đám mây tất cả trong một dành cho các trang web và API. Công cụ được xây dựng trang web có tên là loadforcus.com.
* Công cụ Loadforcus có thể được truy cập với mọi thiết bị với điều kiện thiết bị đó có kết nối mạng.
* Cách sử dụng cũng rất đơn giản: Người dùng chỉ cần truy cập vào trang loadforcus.com sau đó đăng ký 1 tài khoản. Sau đó chọn 1 bài test cho trang web của mình, cập nhật các thông số. Nhập địa chỉ các trang web và bắt đầu. Sau đó Loadforcus sẽ tự động chạy các bài test.

# Giới thiệu phần mềm adcbook.net.vn

## Lý do chọn đề tài

Việc kiểm thử phần mềm là bước không thể thiếu trong quá trình phát triển sản phẩm. Đảm bảo chất lượng và độ tin cậy của phần mềm là một mục tiêu cốt lõi, và việc thực hiện kiểm thử một cách cẩn thận đóng vai trò quan trọng trong việc đạt được điều này Hiện nay, ngành kiểm thử phần mêm đã vô cùng phát triển, đi kèm với đó là hàng loạt các công cụ kiểm thử phần mềm được sinh ra với mục đích đơn giản hoá, cũng như nâng cao chất lượng của phần mềm.

Do đó, chúng em quyết định lựa chọn công cụ LoadForcus để thực hiện đề tài “Kiểm thử phần mềm – website bán sách adcbook.net.vn”. Công cụ kiểm thử Loadfocus được chọn để nghiên cứu và sử dụng trong đề tài này vì tính linh hoạt và hiệu quả của nó. Loadfocus cung cấp các tính năng mạnh mẽ để kiểm thử hiệu suất, kiểm thử tải và kiểm thử chức năng của các ứng dụng web và API.

Qua việc nghiên cứu và sử dụng các kỹ thuật kiểm thử được hỗ trợ bởi công cụ Loadfocus, nhóm em mong muốn nắm vững các phương pháp và tiến trình kiểm thử phần mềm để đảm bảo rằng sản phẩm cuối cùng sẽ đáp ứng được mong đợi của người dùng, mang lại trải nghiệm mua sắm trực tuyến tốt nhất cho người dùng.

## Phân tích thiết kế hệ thống

### Đặc tả hệ thống

### Thiết kế hệ thống

# Kiểm thử phần mềm adcbook.net.vn

## Lập kế hoạch kiểm thử (test plan)

## Thực hiện kiểm thử

### Nguyễn Mỹ Hạnh - <Tên chức năng kiểm thử>

#### *Phân tích thiết kế kiểm thử*

### Nguyễn Hồng Quân -

#### *Phân tích thiết kế kiểm thử*

### Nguyễn Trọng Trường -

#### *Phân tích thiết kế kiểm thử*

### Nguyễn Quang Vũ -

#### *Phân tích thiết kế kiểm thử*

# Phụ lục

## Phụ lục hình ảnh

## Phụ lục bảng biểu

# Tài liệu tham khảo

[1] *Tên tài liệu*  
Tên tác giả - Tên nhà xuất bản, tạp chí, hội nghị, năm xuất bản

[2] *Tên tài liệu*  
Tên tác giả - Tên nhà xuất bản, tạp chí, hội nghị, năm xuất bản