TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GIAO THÔNG VẬN TẢI

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----o0o-----**

A logo of a university of transport technology

Description automatically generated

**MÔN HỌC**

**KIỂM THỬ PHẦN MỀM**

**Tên đề tài :**

**KIỂM THỬ TỰ ĐỘNG APPIUM VÀ ỨNG DỤNG VÀO ANDROID**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:** | **LÊ THỊ CHI** |
| **NHÓM SINH VIÊN SỐ 2:** | **NGUYỄN VĂN KHÁNH**  **NGUYỄN TIẾN THÀNH**  **NGUYỄN VIỆT ANH**  **LÊ NGÔ QUANG** |
| **LỚP:** | **72DCHT21** |

**HÀ NỘI 12-2023**

**MỤC LỤC**

[**MỞ ĐẦU** 3](#_Toc156028407)

[**CHƯƠNG 1: TỔNG QUAN LÝ THUYẾT KIỂM THỬ PHẦN MỀM** 4](#_Toc156028408)

[**1.** **Phần mềm** 5](#_Toc156028409)

[**2.** **Kiểm thử phần mềm và một số khái niệm niên quan** 5](#_Toc156028410)

[**2.1.** **Kiểm thử phần mềm** 5](#_Toc156028411)

[**2.2.** **Một số khái niệm liên quan** 6](#_Toc156028412)

[**3.** **Quy trình kiểm thử phần mềm** 7](#_Toc156028413)

[**4.** **Các cấp độ kiểm thử** 9](#_Toc156028414)

[**4.1.** **Kiểm thử mức đơn vị** 9](#_Toc156028415)

[**4.2.** **Kiểm thử tích hợp** 11](#_Toc156028416)

[**4.3.** **Kiểm thử hồi quy** 11](#_Toc156028417)

[**4.4.** **Kiểm thử chấp nhận sản phẩm** 11](#_Toc156028418)

[**4.5.** **Kiểm thử mức hệ thống** 12](#_Toc156028419)

[**5.** **Các kỹ thuật kiểm thử phần mềm** 12](#_Toc156028420)

[**5.1.** **Nguyên tắc cơ bản của kiểm thử phần mềm** 12](#_Toc156028421)

[**5.2.** **Kỹ thuật kiểm thử hộp trắng (While-Box Testing)** 14](#_Toc156028422)

[**5.3.** **Kỹ thuật kiểm thử hộp đen (Black-Box Testing)** 15](#_Toc156028423)

[**6.** **Ứng dụng của kiểm thử phần mềm trong việc kiểm tra chất lượng sản phẩm.** 16](#_Toc156028424)

[**CHƯƠNG 2 LẬP KẾ HOẠCH TEST** 17](#_Toc156028425)

[**2.** **Giới thiệu tổng quan** 17](#_Toc156028426)

[**2.1.** **App ban hang** 17](#_Toc156028427)

[**2.2.** **Phạm vi** 18](#_Toc156028428)

[**2.3.** **Nhân sự** 18](#_Toc156028429)

[**2.4.** **Môi trường test (Công Cụ)** 18](#_Toc156028430)

[**CHƯƠNG 3: GIỚI THIỆU VÀ CÀI ĐẶT VỀ CÔNG CỤ KIỂM THỬ APPIUM STUDIO** 18](#_Toc156028431)

[**3.1.** **Giới thiệu về kiểm thử tự động Appium Studio** 18](#_Toc156028432)

[**3.1.1.** **Công cụ appium** 18](#_Toc156028433)

[**3.1.2.** **Phần mềm Appium Studio** 19](#_Toc156028434)

[**3.1.3.** **Appium Studio tích hợp trong Eclipse** 20](#_Toc156028435)

[**3.2.** **Hướng dẫn cài đặt đầy đủ Appium Studio** 21](#_Toc156028436)

[**CHƯƠNG 4: GIỚI THIỆU APP BÁN HÀNG** 27](#_Toc156028437)

[**4.1.** **Mô tả chung về phần mềm** 27](#_Toc156028438)

[**4.2.** **Mô tả chức năng** 27](#_Toc156028439)

[**CHƯƠNG 5: BÁO CÁO KẾT QUẢ** 38](#_Toc156028440)

[**5.1.** **Thiết kế test case** 38](#_Toc156028441)

[**5.2.** **Kết quả test** 47](#_Toc156028442)

[**LỜI KẾT** 49](#_Toc156028443)

[**TÀI LIỆU THAM KHẢO** 50](#_Toc156028444)

# **MỞ ĐẦU**

Với sự phát triển như vũ bão của công nghệ thông tin nói chung và công nghệ phần mềm nói riêng, việc phát triển phần mềm ngày càng được hỗ trợ bởi nhiều công cụ tiên tiến, giúp cho việc xây dựng phần mềm đỡ mệt nhọc và hiệu quả hơn. Tuy nhiên, vì độ phức tạp của phần mềm và những giới hạn về thời gian và chi phí, cho dù các hoạt động đảm bảo chất lượng phần mềm nói chung và kiểm thử nói riêng ngày càng chặt chẽ và khoa học, vẫn không đảm bảo được rằng các sản phẩm phần mềm đang được ứng dụng không có lỗi. Lỗi vẫn luôn tiềm ẩn trong mọi sản phẩm phần mềm và cũng có thể gây những thiệt hại khôn lường.

Kiểm thử phần mềm là một quá trình liên tục, xuyên suốt mọi giai đoạn phát triển phần mềm để đảm bảo rằng phần mềm thoả mãn các yêu cầu thiết kế và các yêu cầu đó đáp ứng các nhu cầu của người dùng. Các kỹ thuật kiểm thử phần mềm đã và đang được nghiên cứu, và việc kiểm thử phần mềm đã trở thành quy trình bắt buộc trong các dự án phát triển phần mềm trên thế giới. Kiểm thử phần mềm là một hoạt động rất tốn kém, mất thời gian, và khó phát hiện được hết lỗi. Vì vậy, việc kiểm thử phần mềm đòi hỏi phải có chiến lược phù hợp, một kế hoạch hợp lý và việc thực hiện được quản lí chặt chẽ.

Và với việc những chiếc điện thoại thông minh đang ngày càng được sử dụng nhiều hơn nhằm đáp ứng nhu cầu giải trí đa dạng của người dùng. Từ một chiếc điện thoại thông thường chỉ được cài đặt sẵn vài ba ứng dụng của nhà sản xuất thì nay với các thiết bị chạy các hệ điều hành nhúng (Android, iOS, v.v.) ta có thể dễ dàng đáp ứng được các nhu cầu của người dùng bằng cách cài thêm các phần mềm bên thứ ba mà không gây ra trở ngại nào. Từ đây lại đặt ra một vấn đề hiển nhiên là kiểm thử các phần mềm chạy trên di động này để xem chúng có đáp ứng được các yêu cầu đề ra ban đầu hay không trước khi phát hành sản phẩm tới tay người tiêu dùng.

Đó là lý em chọn đề tài “Kiểm thử tự động Appium và ứng dụng vào andoird”.

Mục đích của đề tài:

Đề tài tìm hiểu cơ sở lý thuyết về kiểm thử nói chung và kiểm thử trên di động nói riêng cũng như cách triển khai công cụ kiểm thử phần mềm tự động để giảm nhân lực kiểm thử và đảm bảo chất lượng phần mềm hơn với công việc kiểm thử bằng tay. Mục tiêu chính của đề tài là nghiên cứu về kiểm thử trên thiết bị di động.

# **CHƯƠNG 1: TỔNG QUAN LÝ THUYẾT KIỂM THỬ PHẦN MỀM**

Kiểm thử nhằm đánh giá chất lượng hoặc tính chấp nhận được của sản phẩm. Ngoài ra, kiểm thử còn giúp phát hiện lỗi hoặc bất cứ vấn đề gì về sản phẩm. Chúng ta cần kiểm thử vì biết rằng con người luôn có thể mắc sai lầm. Điều này đặc biệt đúng trong lĩnh vực phát triển phần mềm và các hệ thống điều khiển bởi phần mềm. Chương này sẽ giới thiệu các khái niệm trong lĩnh vực kiểm thử phần mềm.

1. **Phần mềm**

Phần mềm thường được mô tả bởi ba thành phần cấu thành:

* Tập các lệnh (chương trình máy tính) trên máy tính khi thực hiện sẽ tạo ra các dịch vụ và đem lại những kết quả mong muốn cho người dùng.
* Các cấu trúc dữ liệu (lưu giữ trên các bộ nhớ) làm cho chương trình thao tác hiệu quả với các thông tin thích hợp và nội dung thông tin được số hóa. –
* Các tài liệu để mô tả thao tác, cách sử dụng và bảo trì phần mềm (hướng dẫn sử dụng, tài liệu kỹ thuật, tài liệu phân tích, thiết kế, kiểm thử, v.v.).

1. **Kiểm thử phần mềm và một số khái niệm niên quan** 
   1. **Kiểm thử phần mềm**

Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Kiểm thử có thể cung cấp cho doanh nghiệp một quan điểm, một cách nhìn độc lập về phần mềm để từ đó cho phép đánh giá và thấu hiểu được những rủi ro trong quá trình triển khai phần mềm.

Trong kỹ thuật kiểm thử không chỉ giới hạn ở việc thực hiện một chương trình hoặc ứng dụng với mục đích đi tìm các lỗi phần mềm (bao gồm các lỗi và Đồ án tốt nghiệp Kiểm thử phần mềm trên thiết bị di động các thiếu sót) mà còn là một quá trình phê chuẩn và xác minh một chương trình máy tính / ứng dụng / sản phẩm nhằm:

Đáp ứng được mọi yêu cầu hướng dẫn khi thiết kế và phát triển phần mềm. Thực hiện công việc đúng như kỳ vọng. Có thể triển khai được với những đặc tính tương tự. Và đáp ứng được mọi nhu cầu của các bên liên quan.

Tùy thuộc vào từng phương pháp, việc kiểm thử có thể được thực hiện bất cứ lúc nào trong quá trình phát triển phần mềm. Theo truyền thống thì các nỗ lực kiểm thử được tiến hành sau khi các yêu cầu được xác định và việc lập trình được hoàn tất nhưng trong Agile (là một tập hợp các phương pháp phát triển phần mềm linh hoạt dựa trên việc lặp đi lặp lại và gia tăng giá trị) thì việc kiểm thử được tiến hành liên tục trong suốt quá trình xây dựng phần mềm. Như vậy, mỗi một phương pháp kiểm thử bị chi phối theo một quy trình phát triển phần mềm nhất định.

* 1. **Một số khái niệm liên quan**

Chất lượng phần mềm (Software quality): là mức độ mà một hệ thống, thành phần hay quy trình đáp ứng các yêu cầu của đặc tả phần mềm, các nhu cầu mong đợi của khách hàng hoặc người sử dụng.

Đảm bảo chất lượng phần mềm (Software quality assurance): là một quy trình có kế hoạch và hệ thống của tất cả các hành động cần thiết để cung cấp các thông tin đầy đủ để đảm bảo các sản phẩm có phù hợp với các yêu cầu về kỹ thuật hay không. Mục đích cuối cùng là để đánh giá quy trình sản xuất sản phẩm phần mềm.

Xác nhận (Validation): là quá trình đánh giá một hệ thống hay cấu phần trong hay cuối của quá trình phát triển để xác định xem nó đáp ứng yêu cầu quy định.

Xác minh, kiểm chứng (Verification): là quá trình đánh giá một hệ thống hay thành phần để xác định xem các sản phẩm của một giai đoạn phát triển nhất định đáp ứng các điều kiện áp đặt tại lúc bắt đầu của giai đoạn đó. Xác minh thường là hoạt động có tính kỹ thuật cao hơn, sử dụng những tri thức về các yêu cầu, đặc tả phần mềm. Xác nhận thường phụ thuộc vào tri thức về lĩnh vực tương ứng. Cụ thể là, tri thức về ứng dụng của phần mềm được viết. Ví dụ, xác nhận của phần mềm về máy bay yêu cầu tri thức từ kỹ sư hàng không và phi công.

Lỗi (Error): Lỗi là những vấn đề mà con người mắc phải trong quá trình phát triển các sản phẩm phần mềm.

Sai (Fault): Sai là kết quả của lỗi, hay nói khác đi, lỗi sẽ dẫn đến sai.

Thất bại (Failure): Thất bại xuất hiện khi một lỗi được thực thi.

Sự cố (Incident): Khi thất bại xuất hiện, nó có thể hiển thị hoặc không, tức là rõ ràng hoặc không rõ ràng đối với người dùng hoặc người kiểm thử. Sự cố là triệu chứng liên kết với một thất bại và thể hiện cho người dùng hoặc người kiểm thử về sự xuất hiện của thất bại này.

Ca kiểm thử (Test case): Ca kiểm thử gồm một tập các dữ liệu đầu vào và một xâu các giá trị đầu ra mong đợi đối với phần mềm, mục đích là dựa vào đó để kiểm tra xem phần mềm có thỏa các yêu cầu đặt ra hay không.

A computer screen shot of a computer code

Description automatically generatedKịch bản kiểm thử (Test script): Một kịch bản kiểm thử là một nhóm mã lệnh dạng đặc tả kịch bản dùng để tự động hóa một quy trình hay một ca kiểm tra, giúp cho việc kiểm tra nhanh hơn, hoặc cho những trường hợp mà kiểm tra bằng tay sẽ rất khó khăn hoặc không khả thi.

Hình 1-1: Ví dụ về 1 kịch bản kiểm thử

1. **Quy trình kiểm thử phần mềm**

A diagram of a flowchart

Description automatically generatedMục đích của kiểm thử là thiết kế một chuỗi các trường hợp kiểm thử mà có khả năng phát hiện lỗi cao. Để cho việc kiểm thử đạt được kết quả tốt cần có sự chuẩn bị về kế hoạch kiểm thử, thiết kế các trường hợp kiểm thử và các dữ liệu kiểm thử cho các trường hợp. Đây chính là đầu vào cho giai đoạn kiểm thử. Và sản phẩm công việc của giai đoạn kiểm thử chính là “báo cáo kiểm thử” mà tài liệu hóa tất cả các trường hợp kiểm thử đã chạy, dữ liệu đầu vào, đầu ra mong đợi, đầu ra thực tế và mục đích của kiểm thử.

Hình 1-2: Giai đoạn kiểm thửu trong xử lý phần mềm

Quy trình kiểm thử bao gồm một số giai đoạn:

* Lập kế hoạch kiểm thử: Bước đầu tiên là lập kế hoạch cho tất cả các hoạt động sẽ được thực hiện và các phương pháp được sử dụng. Các chuẩn IEEE bao gồm các thông tin về tác giả chuẩn bị kế hoạch, danh sách liệt kê của kế hoạch kiểm thử. Vấn đề quan trọng nhất đối với kế hoạch kiểm thử:
* Mục đích: Quy định về phạm vi, phương pháp, tài nguyên và lịch biểu của các hoạt động kiểm thử.
* Các tài liệu tham khảo.
* Các định nghĩa.
* Khái quát về xác minh và thẩm định (V&V): tổ chức, tài nguyên, trách nhiệm, các công cụ, kỹ thuật và các phương pháp luận.
* Vòng đời của V&V: các nhiệm vụ, các dữ liệu vào và các kết quả ra trên một giai đoạn vòng đời.
* Báo cáo xác minh và thẩm định(V&V) phần mềm: mô tả nội dung, định dạng và thời gian cho tất cả các báo cáo V&V.
* Các thủ tục quản lý V&V bao gồm các chính sách, thủ tục, các chuẩn, thực nghiệm và các quy ước.
* Giai đoạn bố trí nhân viên kiểm thử: Việc kiểm thử thường phải tiến hành một cách độc lập và các nhóm độc lập có trách nhiệm tiến hành các họat động kiểm thử, gọi là các nhóm kiểm thử.
* Thiết kế các trường hợp kiểm thử: Các trường hợp kiểm thử là các đặc tả đầu vào cho kiểm thử và đầu ra mong đợi của hệ thống cùng với các câu lệnh được kiểm thử.
* Các kỹ thuật kiểm thử hộp đen để kiểm thử dựa trên chức năng.
* Các kỹ thuật kiểm thử hộp trắng để kiểm thử dựa vào cấu trúc bên trong.
* Xử lý đo lường kiểm thử bằng cách thu thập dữ liệu.
* Đánh giá sản phẩm phần mềm để xác nhận sản phẩm có thể sẵn sàng phát hành được chưa?

1. **Các cấp độ kiểm thử**

Các mức kiểm thử phần mềm thông thường:

* Unit Test – Kiểm thử mức đơn vị
* Integration Test – Kiểm thử tích hợp
* System Test - Kiểm thử mức hệ thống
* Acceptance Test - Kiểm thử chấp nhận sản phẩm
* Regression Test - Kiểm thử hồi quy
  1. **Kiểm thử mức đơn vị**

Một đơn vị kiểm thử là một thành phần phần mềm nhỏ nhất mà ta có thể kiểm thử được. Theo định nghĩa này, các hàm (Function), thủ tục (Procedure), lớp (Class), hoặc các phương thức (Method) đều có thể được xem là đơn vị kiểm thử.

Vì đơn vị kiểm thử được chọn để kiểm thử thường có kích thước nhỏ và chức năng hoạt động đơn giản, chúng ta không khó khăn gì trong việc tổ chức, kiểm thử, ghi nhận và phân tích kết quả kiểm thử. Nếu phát hiện lỗi, việc xác định nguyên nhân và khắc phục cũng tương đối dễ dàng vì chỉ khoanh vùng trong một đơn vị đang kiểm thử. Một nguyên lý đúc kết từ thực tiễn: thời gian tốn cho Kiểm thử đơn vị sẽ được đền bù bằng việc tiết kiệm rất nhiều thời gian và chi phí cho việc kiểm thử và sửa lỗi ở các mức kiểm thử sau đó.

Kiểm thử đơn vị thường do lập trình viên thực hiện. Công đoạn này cần được thực hiện càng sớm càng tốt trong giai đoạn viết code và xuyên suốt chu kỳ phát triển phần mềm. Thông thường, Kiểm thử đơn vị đòi hỏi kiểm thử viên có kiến thức về thiết kế và mã nguồn của chương trình. Mục đích của Kiểm thử đơn vị là bảo đảm thông tin được xử lý và xuất ra là chính xác, trong mối tương quan với dữ liệu nhập và chức năng của đơn vị kiểm thử. Điều này thường đòi hỏi tất cả các nhánh bên trong đơn vị kiểm thử đều phải được kiểm tra để phát hiện nhánh phát sinh lỗi. Một nhánh thường là một chuỗi các lệnh được thực thi trong một đơn vị kiểm thử, ví dụ: chuỗi các lệnh sau điều kiện If và nằm giữa then … else là một nhánh. Thực tế việc chọn lựa các nhánh để đơn giản hóa việc kiểm thử và quét hết các đơn vị kiểm thử đòi hỏi phải có kỹ thuật, đôi khi phải dùng thuật toán để chọn lựa.

Cũng như các mức kiểm thử khác, Kiểm thử đơn vị cũng đòi hỏi phải chuẩn bị trước các ca kiểm thử hoặc kịch bản kiểm thử, trong đó chỉ định rõ dữ liệu vào, các bước thực hiện và dữ liệu mong chờ sẽ xuất ra. Các ca kiểm thử và kịch bản này nên được giữ lại để tái sử dụng.

Kiểm thử đơn vị thường sử dụng các Unit Test Framework, đó là các khung chương trình được viết sẵn để hộ trợ cho việc test các mô đun, các đơn vị phần mềm.

* 1. **Kiểm thử tích hợp**

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 thử các thành phần và đơn vị phần mềm riêng lẻ thì kiểm thử tích hợp kết hợp chúng lại với nhau và kiểm thử sự giao tiếp giữa chúng. Kiểm thử tích hợp có 2 mục tiêu chính:

* Phát hiện lỗi giao tiếp xảy ra giữa các đơn vị kiểm thử.
* Tích hợp các đơn vị kiểm thử đơn lẻ thành các hệ thống nhỏ (subsystem) và cuối cùng là nguyên hệ thống hoàn chỉnh (system) chuẩn bị cho kiểm thử ở mức hệ thống.
  1. **Kiểm thử hồi quy**

Kiểm thử hồi quy không phải là một mức kiểm thử, như các mức khác đã nói ở trên. Nó đơn thuần kiểm tra lại phần mềm sau khi có một sự thay đổi xảy ra, để bảo đảm phiên bản phần mềm mới thực hiện tốt các chức năng như phiên bản cũ và sự thay đổi không gây ra lỗi mới trên những chức năng vốn đã làm việc tốt. Kiểm thử hồi quy có thể thực hiện tại mọi mức kiểm thử. Ví dụ: một phần mềm đang phát triển khi kiểm tra cho thấy nó chạy tốt các chức năng A, B và C. Khi có thay đổi code của chức năng C, nếu chỉ kiểm tra chức năng C thì chưa đủ, cần phải kiểm tra lại tất cả các chức năng khác liên quan đến chức năng C, trong ví dụ này là A và B. Lý do là khi C thay đổi, nó có thể sẽ làm A và B không còn làm việc đúng nữa.

* 1. **Kiểm thử chấp nhận sản phẩm**

Thông thường, sau giai đoạn kiểm thử hệ thống là kiểm thử chấp nhận, được khách hàng thực hiện (hoặc ủy quyền cho một nhóm thứ ba thực hiện). Mục đích của kiểm thử chấp nhận là để chứng minh phần mềm thỏa mãn tất cả yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm (và trả tiền thanh toán hợp đồng). Kiểm thử chấp nhận có ý nghĩa hết sức quan trọng, mặc dù trong hầu hết mọi trường hợp, các phép kiểm thử của kiểm thử hệ thống và kiểm thử chấp nhận gần như tương tự, nhưng bản chất và cách thức thực hiện lại rất khác biệt.

* 1. **Kiểm thử mức hệ thống**

Mục đích Kiểm thử mức hệ thống là kiểm tra thiết kế và toàn bộ hệ thống (sau khi tích hợp) có thỏa mãn yêu cầu đặt ra hay không. 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 vị 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 đơn vị phần mềm và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện kiểm thử hệ thống. 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. Mức kiểm thử này đặc biệt thích hợp cho việc phát hiện lỗi giao tiếp với phần mềm hoặc phần cứng bên ngoài, chẳng hạn các lỗi “bế tắc” (deadlock) hoặc chiếm dụng bộ nhớ. Sau giai đoạn kiểm thử hệ thống, phần mềm thường đã sẵn sàng cho khách hàng hoặc người dùng cuối cùng kiểm thử để chấp nhận hoặc dùng thử (Alpha/Beta Test).

1. **Các kỹ thuật kiểm thử phần mềm**

Có thể chia các kỹ thuật kiểm thử phần mềm thành hai loại: các kỹ thuật kiểm thử hộp đen (black-box testing) và kỹ thuật kiểm thử hộp trắng (whitebox testing). Các kiểm thử hộp đen tìm các lỗi như thiếu các chức năng, khả năng sử dụng và các yêu cầu phi chức năng. Trong khi các kỹ thuật kiểm thử hộp trắng yêu cầu hiểu biết về cấu trúc chương trình bên trong và các kiểm thử nhận được từ đặc tả thiết kế bên trong hoặc từ mã.

* 1. **Nguyên tắc cơ bản của kiểm thử phần mềm**

Trong lúc kiểm thử, công nghệ phần mềm phát sinh một chuỗi các trường hợp kiểm thử được sử dụng để “tách từng phần” phần mềm. Kiểm thử là một bước trong quy trình phần mềm mà có thể được xem xét bởi đội ngũ phát triển bằng cách phá vỡ thay vì xây dựng. Các kỹ sư phần mềm chính là những người xây dựng và kiểm thử yêu cầu họ vượt qua các khái niệm cho trước về độ chính xác và giải quyết mâu thuẫn khi các lỗi được xác định.

* + 1. **Mục tiêu của kiểm thử**

Các nguyên tắc được xem như mục tiêu kiểm thử là:

* + Kiểm thử là một quá trình thực thi chương trình với mục đích tìm lỗi.
  + Một trường hợp kiểm thử tốt là trường hợp kiểm thử mà có khả năng cao việc tìm thấy các lỗi chưa từng được phát hiện.
  + Một kiểm thử thành công là kiểm thử mà phát hiện lỗi chưa từng được phát hiện.
    1. **Luông thông tin kiểm thử**

Luồng thông tin cho kiểm thử được biểu diễn bởi mô hình trong Hình 1.4. Hai kiểu của đầu vào được truyền cho quá trình kiểm thử:

* + Cấu hình phần mềm: gồm các đặc tả yêu cầu, đặc tả thiết kế, và mã nguồn.
  + A diagram of a diagram

    Description automatically generatedCấu hình kiểm thử: gồm có kế hoạch kiểm thử, các thủ tục, trường hợp kiểm thử, và các công cụ kiểm thử.

Hình 1-3: Luồng thông tin kiểm thử

* + 1. **Thiết kế trưởng hợp kiểm thử**

Thiết kế kiểm thử phần mềm có thể là một quá trình thu thập, phân tích và thực hiện yêu cầu. 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 nhất với thời gian và công sức tối thiểu. Như vậy, vấn đề quan trọng nhất trong kiểm thử phần mềm là thiết kế và tạo ra các trường hợp kiểm thử có hiệu quả. Lý do về tầm quan trọng của việc thiết kế các trường hợp kiểm thử xuất phát từ thực tế: Kiểm thử “vét cạn” là điều không thể, và như vậy, kiểm thử một chương trình phải luôn xác định là không thể vét cạn. Vấn đề quan trọng là cố gắng làm giảm sự “không thể vét cạn” nhiều nhất có thể.

Kiểm thử phần mềm còn có các ràng buộc về thời gian, chi phí, v.v. Chìa khoá của kiểm thử là trả lời của câu hỏi: “Tập con của tất cả các trường hợp kiểm thử có thể có xác suất phát hiện lỗi cao nhất là gì?”. Việc nghiên cứu các phương pháp thiết kế trường hợp kiểm thử sẽ cung cấp câu trả lời cho câu hỏi này.

Bất kỳ sản phẩm công nghệ nào có thể được kiểm thử trong hai cách:

* Biết về các chức năng cụ thể mà sản phẩm đã được thiết kế để thực hiện.
* Biết cách hoạt động bên trong của sản phẩm, kiểm thử có thể được thực hiện để đảm bảo rằng “tất cả các thành phần ăn khớp nhau”.

Cách tiếp cận kiểm thử đầu tiên được gọi là kiểm thử hộp đen và cách thứ hai là kiểm thử hộp trắng.

* 1. **Kỹ thuật kiểm thử hộp trắng (While-Box Testing)**

Kiểm thử hộp trắng: Là kỹ thuật kiểm thử dựa trên đặc tả bên trong của chương trình, dựa vào mã nguồn, cấu trúc chương trình. Kiểm thử hộp trắng thường phát hiện các lỗi lập trình. Loại kiểm thử này khá khó thực hiện và chi phí cao.

Với các module quan trọng, thực thi việc tính toán chính của hệ thống, phương pháp này là cần thiết.

* 1. **Kỹ thuật kiểm thử hộp đen (Black-Box Testing)**

Kiểm thử hộp đen: là một phương pháp kiểm thử phần mềm được thực hiện mà không biết được cấu tạo bên trong của phần mềm, là cách mà các tester kiểm tra xem hệ thống như một chiếc hộp đen, không có cách nào nhìn thấy bên trong của cái hộp.

Phương pháp này được đặt tên như vậy bởi vì các chương trình phần mềm, trong con mắt của các tester, giống như một hộp đen; bên trong mà người ta không thể nhìn thấy. Phương pháp này cố gắng tìm ra các lỗi trong các loại sau:

- Chức năng không chính xác hoặc thiếu.

- Lỗi giao diện.

- Lỗi trong cấu trúc dữ liệu hoặc truy cập cơ sở dữ liệu bên ngoài.

- Hành vi hoặc hiệu suất lỗi.

- Khởi tạo và chấm dứt các lỗi.

A diagram of a black box

Description automatically generatedƯu điểm:

- Kỹ sư kiểm thử có thể không phải IT chuyên nghiệp.

- Hệ thống thật sự với toàn bộ yêu cầu của nó được kiểm thử chính xác.

- Thiết kế kịch bản kiểm thử khá nhanh, ngay khi mà các yêu cầu chức năng được xác định.

Nhược điểm:

- Dữ liệu đầu vào yêu cầu một khối lượng mẫu (sample) khá lớn.

- Khó viết kịch bản kiểm thử do cần xác định tất cả các yếu tố đầu vào, và thiếu cả thời gian cho việc tập hợp này.

- Khả năng để bản thân kỹ sư lạc lối trong khi kiểm thử là khá cao

Mọi kỹ thuật nào cũng có ưu điểm và nhược điểm của nó. Các hệ thống thường phải được sử dụng nhiều phương pháp kiểm thử khác nhau để đảm bảo được chất lượng của hệ thống khi đến tay người dùng.

1. **Ứng dụng của kiểm thử phần mềm trong việc kiểm tra chất lượng sản phẩm.**

Đảm bảo tính đúng đắn: Kiểm thử phần mềm giúp xác định xem phần mềm hoạt động theo các yêu cầu chức năng đã đặt ra hay không. Nó đảm bảo rằng phần mềm cung cấp đúng các tính năng và chức năng được yêu cầu.

Phát hiện lỗi và thiếu sót: Kiểm thử phần mềm giúp phát hiện các lỗi, vấn đề vàthiếu sót trong phần mềm. Bằng cách kiểm tra và phân tích kết quả kiểm thử, các lỗi và vấn đề được ghi nhận và sau đó có thể được sửa chữa để cải thiện chất lượng sản phẩm.

Đảm bảo tính ổn định và tin cậy: Kiểm thử phần mềm giúp đảm bảo tính ổn định và tin cậy của phần mềm trong các tình huống khác nhau. Bằng cách thực hiện các ca kiểm thử và kiểm tra hiệu năng, phần mềm có thể được đánh giá về khả năng chịu tải và đáp ứng trong các điều kiện khác nhau.

Đảm bảo tương thích: Kiểm thử phần mềm đảm bảo tính tương thích của phần mềm trên các nền tảng, hệ điều hành và môi trường khác nhau. Điều này đảm bảo rằng phần mềm hoạt động một cách nhất quán và đáp ứng được yêu cầu của người dùng trên các môi trường khác nhau.

Đảm bảo hiệu suất và tối ưu hóa: Kiểm thử phần mềm giúp đánh giá hiệu suất của phần mềm và tìm ra các vấn đề về tốc độ, thời gian phản hồi và sử dụng tài nguyên. Điều này giúp nhà phát triển tối ưu hóa hiệu suất và sửa các vấn đề liên quan đến hiệu năng của phần mềm.

Xác nhận tính bảo mật: Kiểm thử phần mềm giúp xác nhận tính bảo mật của phần mềm. Bằng cách tìm kiếm và phát hiện các lỗ hổng bảo mật, kiểm thử phần mềm đóng góp vào việc đảm bảo an toàn và bảo mật cho dữ liệu và hệ thống.

Qua đó, kiểm thử phần mềm đóng góp quan trọng vào việc đảm bảo chất lượng sản phẩm. Nó giúp phát hiện và sửa các lỗi, thiếu sót, tăng tính ổn định và tin cậy của phần mềm, đảm bảo tính tương thích và hiệu suất, và xác nhận tính bảo mật.

# **CHƯƠNG 2 LẬP KẾ HOẠCH TEST**

1. **Giới thiệu tổng quan** 
   1. **App ban hang**

Một app bán hàng nhỏ chúng em tạo ra

* 1. **Phạm vi**

Các chức năng test:

* Chức năng tìm kiếm sản phẩm
* Chức năng đăng nhập
* Chức năng đăng kí
* Chức năng thêm sản phẩm vào giỏ hàng
* Chức năng xóa sản phẩm trong giỏ hàng
  1. **Nhân sự**

1. Nguyễn Văn Khánh
2. Nguyễn Tiến Thành
3. Nguyễn Việt Anh
4. Lê Ngô Quang
   1. **Môi trường test (Công Cụ)**

Appium Studio, Android

# **CHƯƠNG 3: GIỚI THIỆU VÀ CÀI ĐẶT VỀ CÔNG CỤ KIỂM THỬ APPIUM STUDIO**

* 1. **Giới thiệu về kiểm thử tự động Appium Studio**
     1. **Công cụ appium**

Appium là một công cụ mã nguồn mở được sử dụng để kiểm thử tự động (test automation) các native app, mobile web app, và hybrid app trên nền tảng iOS và Android.

Appium được phát triển từ nền tảng hệ thống Selenium (kế thừa các đối tượng, cấu trúc và cú pháp) nên Appium có khả năng làm việc với nhiều nền tảng khác nhau, iOS và Android, trên cùng một mã kiểm thử (cross-platforms).

Khả năng cross-platforms của Appium đến từ việc sử dụng các thư viện của Appium, thông qua một chương trình server, chuyển các câu lệnh sử dụng trong mã kiểm thử thành các lệnh UIAutomation (với iOS) hay UIAutomator (với Android) để tương tác với thiết bị.

Đặc biệt là Appium hỗ trợ “đa nền tảng” (cross-platform) cho phép bạn sử dụng API giống nhau để viết test cho các nền tảng khác nhau (iOS và Android). Điều này khá là tiện lợi khi bạn muốn sử dụng lại các bộ ca kiểm thử của mình. Bên cạnh đó, Appium hỗ trợ viết test cho rất nhiều ngôn ngữ, từ Java cho đến Ruby, Python, JavaScript, v.v.

Các thành phần của Appium

- Chương trình máy chủ Appium – Appium Server: là một chương trình tạo lập một máy chủ Java, dùng để chuyển các lệnh trong mã kiểm thử thành các lệnh có thể tương tác với thiết bị: UIAutomation với iOS hay UIAutomator với Android.

- Hệ thống thư viện Appium: Cũng như Selenium, Appium có một hệ thống thư viện dùng để nhận diện và tương tác với các đối tượng UI trên ứng dụng di động. Hệ thống thư viện Appium được cung cấp cho nhiều ngôn ngữ lập trình khác nhau như C#, Java, Python, v.v. để kỹ sư kiểm thử tự động có thể chọn ngôn ngữ lập trình quen thuộc cho việc tự động hóa kiểm thử.

* + 1. **Phần mềm Appium Studio**

Appium Studio là một IDE được thiết kế cho việc phát triển và thực thi kiểm thử tự động trên di động bằng cách sử dụng Appium/Selenium WebDriver API.

Phần mềm này loại bỏ phần lớn sự phụ thuộc cứng nhắc và các điều kiện tiên quyết của Appium chẳng hạn như yêu cầu phát triển của iOS trên các máy MAC OSX hoặc không có khả năng chạy các thử nghiệm song song trên các thiết bị iOS thực / mô phỏng. Giao diện của Appium Studio cung cấp một bộ công cụ để đơn giản hóa quá trình kiểm thử.

Những ưu điểm của Appium Studio:

* Có thể chạy kiểm thử trên hệ điều hành Windows.
* Đơn giản hoá việc thiết lập môi trường (không cần Xcode).
* Không bị giới hạn để kiểm tra ứng dụng của mình.
* Đẩy nhanh và đơn giản hóa quá trình kiểm thử.

Một số nhược điểm:

* Độ tùy biến không cao.
* Có phiên bản miễn phí nhưng không được hỗ trợ truy cập trên đám mây.
  + 1. **Appium Studio tích hợp trong Eclipse**

Appium Studio dành cho Eclipse là một giải pháp dành cho các nhà phát triển Java muốn tự động kiểm thử ứng dụng trên thiết bị di động trên cả 2 nền tảng Android và iOS.

Appium Studio là một thành phần mở rộng để tự động hóa quá trình kiểm thử và có thể tích hợp vào JUnit, TestNG.

Các ưu điểm của Appium Studio dành cho Eclipse:

* Kết nối thiết bị đám mây và phản chiếu thiết bị: hỗ trợ truy cập từ xa vào thiết bị.
* Quản lý ứng dụng: Dễ dàng quản lý các ứng dụng di động của bạn
* Có thể xác định đối tượng dễ dàng và cung cấp chế độ offline (phát triển các bài kiểm tra di động mà không internet hay thiết bị).
* Không cần kiểm thử viên phải hiểu được mã nguồn của chương trình.
* Kiểm thử một lúc được đa nền tảng và nhiều ca kiểm thử.

Một số nhược điểm:

* Không hỗ trợ kiểm thử trên thiết bị thật
* Độ trễ của các thiết bị đám mây sẽ kéo dài thời gian kiểm thử.
* Đôi lúc hơi khó khăn để xác định đối tượng với những chương trình không rõ ràng.
* Bị giới hạn khả năng truy cập trên đám mây với bản miễn phí.
* Không thể kiểm thử được chính xác hiệu năng của phần mềm.
* Không thể kiểm thử khả năng gián đoạn khi có cuộc gọi, tin nhắn đến.
  1. **Hướng dẫn cài đặt đầy đủ Appium Studio**
     1. **Tải java và cài đặt java\_home trong các biến môi trường**

Truy cập link sau: <https://www.java.com/en/download/>

Xong chúng ta vào tìm thư mục bin có đường dẫn này của mình là thư mục này :

C:\Program Files\Java\jdk-20\bin

A screenshot of a computer

Description automatically generated

Rồi cài đặt môi trường trong phần Environment Variables ở trong phần PathA screenshot of a computer

Description automatically generated

* + 1. **Rồi ta tải android studio trên trang android studio**

Sau đó ta kiểm tra C\User\System\Name\AppData. Nhấn Enter>AppData có thể ở chế độ ẩn. rồi bấm vào thư mục cục bộ. Bấm vào thưu mực Android. Bạn sẽ thấy thưu mục sdk.

Sau đó ta vào SDK folder>tools>bin>sao chép đường dẫn bin path>quay lại environment variable>chọn path>past the bin path>; tiếp tục copy và dán tools path C:\Users\Sarif\AppData\Local\Android\Sdk\tools > tiếp tục copy đường dẫn và path tiếp> ‘platform-tools’.

Path:(C:\Users\Sarif\AppData\Local\Android\Sdk\platform-tools)

A screenshot of a computer

Description automatically generated

* + 1. **Rồi ta mở Android studio và định cấu hình thiết bị mô phỏng ảo**

A screenshot of a computer program

Description automatically generated

* + 1. **Tiếp đến ta tải Node.js** ở link sau: <https://nodejs.org/en/download/> rồi ta tiếp tục cài đường dẫn C:\Program Files\nodejs\node\_modules\npm\ bin vào phần Environment Variables
    2. **A computer screen with white text

       Description automatically generatedTải xuống Appiumn từ nút npm** ta mở cmd gõ “npm install -g appium”, rồi nhấn Enter > giải nén và hoàn tất tất cả quá trình cài đặt ở cuối sẽ thông báo các gói đã thêm sau khi cài đặt xong ta gõ appium > enter nó sẽ cho bạn biết số phiên bản trình giao diện như hình ở bên dưới.

Lưu ý: Appium có 2 phần là Client và Server

**Client**: chúng tôi có thể viết mã Appium bằng nhiều ngôn ngữ như Java, JavaScript, C#, Python, Rube. Vì vậy, chúng tôi cần các jar/gói/thư viện cụ thể theo ngôn ngữ mà chúng tôi sẽ sử dụng. Ví dụ: Appium java client jar.

**Server**: Chúng tôi cũng cần Máy chủ Appium để nghe mã của chúng tôi từ máy khách và thực thi theo yêu cầu trong Máy chủ và trình kích hoạt trên thiết bị di động (ảo).

**npm**: npm là công cụ dòng lệnh hoặc trình cài đặt để cài đặt Appium. Chúng tôi chỉ có thể tải xuống bất kỳ mô-đun nút nào (Ví dụ: Máy chủ Appium) bằng cách sử dụng npm.

Để gỡ cài đặt server appium: “npm unistall -g appium”

Nếu không dùng cách trên ta có thể cài đặt appium ở đường dẫn sau: <https://github.com/appium/appium-desktop/releases/tag/v1.13.0> Phiên bản có thể cũ các bạn có thể tải phiên bản mới hơn.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* + 1. Sau đó ta cài đặt và mở Eclipse và tạo dư án Maven (mẫu QuickStart) với các phần của client appium.

A screenshot of a computer

Description automatically generatedNếu các bạn ko hiểu mình có thể tìm hiểu qua link youtube ở phần tài liệu tham khảo

# **CHƯƠNG 4: GIỚI THIỆU APP BÁN HÀNG**

* 1. **Mô tả chung về phần mềm**

Một app bán hàng do chúng em tạo ra

* 1. **Mô tả chức năng**
     1. **Chức năng tìm kiếm sản phẩm**

A screenshot of a phone

Description automatically generatedTìm kiếm Đặc tả chức năng tìm kiếm

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Chức năng | Tìm kiếm |
| Mô tả | Cho phép khách hàng tìm kiếm sản phẩm trên app |
| Tác nhân | Khách hàng |
| Tiền điều kiện | Khách hàng đã đăng nhập vào hệ thống |
| Dòng sự kiện chính: Tìm kiếm thành công   1. Khách hàng chọn 2. thanh “tìm kiếm”. 3. Khách hàng nhập tên sản phẩm cần tìm. 4. Hệ thống tìm kiếm sản phẩm trong csdl 5. Hệ thống tìm thấy thông tin sản phẩm 6. Hiện ra màm hình thông tin sản phẩm | |
| Dòng sự kiện khác: không tìm thấy sản phầm   1. Hệ thống không tìm thấy thông tin sản phẩm. 2. Hiện ra thông báo “không tìm thấy thông tin sản phẩm cần tìm”. | |

* + 1. **Chức năng đổi mật khẩu**

A screenshot of a phone

Description automatically generatedĐổi mật khẩu

Đặc tả chức năng đổi mật khẩu

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Chức năng | Đổi mật khẩu |
| Mô tả | Cho phép khách hàng đổi mật khẩu trên trang web |
| Tác nhân | Khách hàng |
| Tiền điều kiện | Khách hàng đã đăng nhập vào hệ thống |
| Dòng sự kiện chính: đổi mật khẩu thành công   1. Khách hàng chọn “ Quên mật khẩu”. 2. Khách hàng nhập “Email”. 3. Khách hàng nhập mật khẩu cũ. 4. Khách hàng nhập mật khẩu mới. 5. Khách hàng nhấn “Cập nhập”. 6. Hệ thống kiểm tra mật khẩu cũ trong csdl. 7. Hê thống kiểm tra mật khẩu mới . 8. Hệ thống kiểm tra xác nhận mật khẩu. 9. Hệ thống thông báo “Đổi mật khẩu thành công”. | |
| Dòng sự kiện khác: không tìm thấy sản phầm   * 1. Hệ thống thông báo “Mật khẩu cũ không chính xác”.   7.1. Hệ thống thông báo“Mật khẩu mới phải chứ ít nhất 6 kí tự”.  8.1. Hệ thống thông báo “xác nhận nhập mật khẩu không đúng”. | |

* + 1. **Chức năng đăng nhập**

Đăng nhập

A screenshot of a cell phone

Description automatically generated

Đặc tả chức năng tra cứu đơn hàng

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Chức năng | Đăng nhập |
| Mô tả | Cho phép quản lý và khánh hàng đăng nhập để sử dụng hệ thống |
| Tác nhân | Khách hàng,quản lý |
| Tiền điều kiện | Khách hàng,quản lý đã đăng nhập vào hệ thống |
| Dòng sự kiện chính:   1. Nhập thông tin đăng nhập 2. Kiểm tra đầu vào 3. Gửi yêu cầu đăng nhập 4. Xác thực thông tin 5. Xử lý kết quả đăng nhập | |
| Dòng sự kiện khác:   1. Người dùng nhấn quên mật khẩu 2. Người dùng chuyển đến trang đăng kí | |

* + 1. **Chức năng thêm sản phẩm vào giỏ hàng**

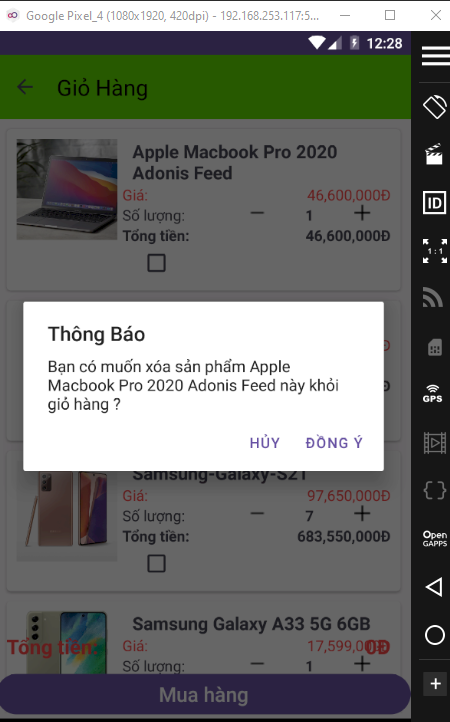
A screenshot of a phone

Description automatically generatedThêm sản phẩm vào giỏ hàng

Đặc tả chức năng thêm sản phẩm vào giỏ hàng

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Chức năng | Thêm giỏ hàng |
| Mô tả | Cho phép khách hàng thêm các sản phẩm mình muốn mua vào giỏ hàng |
| Tác nhân | Khách hàng |
| Tiền điều kiện | không |
| Dòng sự kiện chính: Thêm vào giỏ hàng thành công   1. Khách hàng chọn nút “thêm vào giỏ hàng” ở sản phẩm muốn mua 2. Hệ thống thêm sản phẩm vào giỏ hàng thông báo “Thêm thành công”. | |
| Dòng sự kiện khác: không | |

* + 1. **Chức năng xóa sản phẩm trong giỏ hàng**

Xóa sản phẩm trong giỏ hàng

Đặc tả chức năng xóa sản phẩm

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Chức năng | Xóa giỏ hàng |
| Mô tả | Cho phép khách hàng xóa sản phẩm đang định mua trong giỏ hàng |
| Tác nhân | Khách hàng |
| Tiền điều kiện | Khách hàng vào chứ năng “Giỏ hàng”. Có sản phẩm trong giỏ hàng |
| Dòng sự kiện chính: Xóa giỏ hàng thành công   1. Khách hàng vào giỏ hàng và có thể chọn nút “Xóa” ở sản phẩm cần xóa 2. Hệ thống sẽ thực hiện lệch xóa của khách hàng và thông báo “Xóa thành công”. 3. Hệ thống cập nhập lại tổng số tiền trong giỏ hàng | |
| Dòng sự kiện khác: không | |

# **CHƯƠNG 5: BÁO CÁO KẾT QUẢ**

* 1. **Thiết kế test case**
     1. **Chức năng tìm kiếm**

1. **Kiểm thử lớp tương đương**

Xác định các lớp hợp lệ, lớp không hợp lệ:

Lớp hợp lệ:

* Kiểu text độ dài >0

Lớp không hợp lê:

* Các kí tự đặc biệt (%,\*,&,…)
* Chỉ có ký tự trống(“ ”)
* Ø

Bảng kiểm thử lớp tương đương chức năng tìm kiếm sản phẩm

|  |  |  |
| --- | --- | --- |
| Dữu liệu vào | Hợp lệ | Không hợp lệ |
| Kiểu text | Kiểu text độ dài >0. | - Có ký tự đặc biệt (%, &,\*,…)  - Chỉ có kí tự trống (“ ”) |
| Không phải kiểu text | Ø | Ø |

Kịch bản test và kết quả thực thi của chức năng tìm kiếm sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\_ID | Tiêu đề | Kịch bản | EO | RO | Kết Luận |
| TC1 | Nhập đúng tên sản phẩm | 1. Nhấn biểu tượng tìm kiếm 2. Nhập “iphone” | Hiển thị sản phẩm thành công | Hiển thị sản phẩm thành công | pass |
| TC2 | Nhập tên sản phầm kèm kí tự đặc biệt | 1. Nhấn biểu tượng tìm kiếm 2. Nhập “%iphone” | Hệ thống thông báo lỗi  “lỗi trong trình sử lý Ngoại lệ” | Hệ thống không hiển thị sản phẩm và thông báo lỗi | pass |
| TC3 | Nhập “khoảng trắng” | 1. Nhấn biểu tượng tìm kiếm 2. Nhập “ ” | Hệ thống không hiển thị sản phẩm | Hệ thống không hiển thị sản phẩm | Pass |
| TC4 | Không nhập tên sản phẩm | 1. Nhấn biểu tượng tìm kiếm 2. Để trống | Hệ thống không hiển thị các sản phẩm | Hệ thống không hiển thị các sản phẩm | pass |

Bảng test report chức năng tìm kiếm

|  |  |  |  |
| --- | --- | --- | --- |
| Số lượng test case | Số lượng passed | Số lượng fail | Số lượng test case không chạy |
| 4 | 4 | 0 | 0 |

1. **Kiểm thử bảng quyết định**

Xác định điều kiện, giá trị điều kiện, hành động của hệ thống

Xác định điều kiện và giá trị

T: Nhập đúng

F: Nhập sai

B: Bỏ trống

Bảng quyết định chức năng tìm kiếm sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| Điều kiện | TH1 | TH2 | TH3 |
| Nhập tên sản phẩm | T | F | F |
| Hành động hệ thống | | | |
| Tìm kiếm sản phẩm | T | F | F |

Kịch bản test và kết quả thực thi của chức năng tìm kiếm sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\_ID | Tiêu đề | Kịch bản | EO | RO | Kết luận |
| TC1 | Nhập đúng tên sản phẩm | 1. Chọn biểu tượng tìm kiếm 2. Nhập “iphone” | Hiển thị sản phẩm thành công | Hiển thị sản phẩm thành công | Pass |
| TC2 | Nhập sai tên sản phẩm | 1. Chọn biểu tượng tìm kiếm 2. Nhập “ipone1” | Không hiển thị sản phẩm | Không hiển thị sản phẩm | pass |
| TC3 | Không nhập tên sản phẩm | 1. Chọn biểu tượng tìm kiếm 2. Để trống | Hệ thống không hiển thị các sản phẩm | Hệ thống không hiển thị các sản phẩm | Pass |

Bảng test report chức năng tìm kiếm

|  |  |  |  |
| --- | --- | --- | --- |
| Số lượng test case | Số lượng passed | Số lượng fail | Số lượng test không chạy |
| 3 | 3 | 0 | 0 |

* + 1. **Chức năng thêm và xóa sản phẩm ở giỏ hàng**

1. **Chức năng thêm sản phẩm vào giỏ hàng tại trang xem chi tiết sản phẩm**

Xác định điều kiện, giá trị điều kiện, hành động của hệ thống

Xác định điều kiện và giá trị

T: chọn

F: không chọn

Chọn số lượng: T, F

Bảng quyết định

|  |  |  |
| --- | --- | --- |
| Điều kiện | TH1 | TH2 |
| Chọn sản phẩm | T | F |
| Hành động của hệ thống |  |  |
| Thêm sản phẩm thành công hay không | T | F |

Kịch bản test và kết quả thực thi chức năng thêm sản phẩm vào giỏ hàng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\_ID | Tiêu đề | Kịch bản | EO | RO | Kết luận |
| TC1 | Thêm sản phẩm thành công | 1. Bấm vào sản phẩm 2. Thêm sản phẩm vào giỏ hàng | Hệ thống thông báo thêm “sản phẩm thành công” | Thêm sản phẩm thành công | Pass |
| TC2 | Thêm sản phẩm thất bại | 1. Không chọn sản phẩm | Không có gì xảy ra | Không có gì xảy ra | pass |

Bảng test report chức năng thêm sản phẩm vào giỏ hàng

|  |  |  |  |
| --- | --- | --- | --- |
| Số lượng test case | Số lượng passed | Số lượng fail | Số lượng test case không chạy |
| 2 | 2 | 0 | 0 |

1. **Chức năng xóa sản phẩm trong giỏ hàng**

Xác định điều kiện, giá trị điều kiện, hành động hệ thống

Xấc định điều kiện, giá trị điều kiện, hành động của hệ thống

Xác định điều kiện và giá trị

T: chọn

F: không chọn

Chọn số lượng: T, F

Bảng quyết định

|  |  |  |
| --- | --- | --- |
| Điều kiện | TH1 | TH2 |
| Chọn sản phẩm để xóa | T | F |
| Hành động của hệ thống | | |
| Xóa sản phẩm thành công hay không | T | F |

Kịch bản test và kết quả thực thi chức năng xóa sản phẩm trong giỏ hàng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\_ID | Tiêu đều | Kịch bản | EO | RO | Kết luận |
| TC1 | Xóa sản phẩm thành công | 1. Chọn sản phẩm 2. Bấm xóa | Sản phẩm bị xóa khỏi giỏ hàng | Sản phẩm bị xóa khỏi giỏ hàng | pass |
| TC2 | Xóa sản phẩm thất bạn | 1. Không chọn sản phẩm 2. Bấm xóa | Hệ thống thông báo “xóa sản phẩm thất bại” | Không có gì xảy ra | Fail |

Bảng test report chức năng xóa sản phẩm trong giỏ hàng

|  |  |  |  |
| --- | --- | --- | --- |
| Số lượng test case | Số lượng passed | Số lượng fail | Số lượng test case không chạy |
| 2 | 1 | 1 | 0 |

* + 1. **Chức năng đổi mật khẩu**

Xác định điều kiện, giá trị điều kiện, hành động của hệ thống

Xác định điều kiện và giá trị (T hợp lệ, F không hợp lệ, B để trống)

Mật khẩu cũ: T, F, B

Mật khẩu mới: T, F, B

Xác nhận mật khẩu: T, F, B

Bảng quyết định chức năng đổi mật khẩu

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Điều kiện | TH1 | TH2 | TH3 | TH4 | TH5 | TH6 | TH7 |
| Mật khẩu cũ | T | T | T | T | T | F | B |
| Mật khẩu mới | T | T | T | F | B | - | - |
| Xác nhận mật khẩu | T | F | B | - | - | - | - |
| Hành động hệ thống | | | | | | | |
| Đổi mật khẩu Thành công hoặc thất bại | T | F | F | F | F | F | F |

Bảng kết quả test chức năng đổi mật khẩu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID-TC | Tiêu đề | Kịch bản | EQ | RO | Kết luận |
| TC1 | Đổi mật khẩu thành công | - Mật khẩu cũ:  giang123  - Mật khẩu mới:  123456789  - Xác nhận mật khẩu:  123456789  - Nhấn nút lưu | Hệ thống hiển thị thông báo: “đổi mật khẩu thành công” | Hệ thống hiển thị thông báo: “đổi mật khẩu thành công” | pass |
| TC2 | Đổi mật khẩu thất bại | -Mật khẩu cũ: 123456789  -Mật khẩu mới: giang 123  - xác nhận mật khẩu: kiemthu123  - nhấn nút lưu mật khẩu | Hệ thống hiển thị thông báo  “xác nhận mật khẩu chưa chính xác” | Hệ thống hiển thị thông báo  “Thông tin xác nhận mật khẩu không đúng” | pass |
| TC3 | Đổi mật khẩu thất bại | -Mật khẩu cũ: 123456789  -Mật khẩu mới: giang 123  - xác nhận mật khẩu: “bỏ trống”  - nhấn nút lưu mật khẩu | Hệ thống hiển thị thông báo “Phải nhập xác nhận mật khẩu” | Hệ thống hiển thị thông báo “vui lòng nhập lại mật khẩu mới” | pass |

Bảng test case report chức năng đổi mật khẩu

|  |  |  |  |
| --- | --- | --- | --- |
| Số lượng test case | Số lượng passed | Số lượng fail | Số lượng test case không chạy |
| 3 | 3 | 0 | 0 |

* + 1. **Chức năng đăng nhập**

Xác định điều kiện, giá trị điều kiện, hành động của hệ thống

Xác định điều kiện và giá trị (T hợp lệ, F không hợp lệ, B để trống)

|  |  |  |  |
| --- | --- | --- | --- |
|  | A1 | A2 | A3 |
| Tập A (email) | T | F | B |
| Tập B (password) | T | F | B |

+ Trường hợp kiểm thử tương đương yếu: 3 test case A1B1, A2B2, A3B3

+Trường hợp kiểm thử tương đương mạnh: 3x3 = 9 Test Case.

Bảng test kết quả chức năng đăng nhập

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\_ID | Tiêu đề | Kịch bản | EO | RO | Kết luận |
| TC1 | Nhập đúng Email, đúng Mật khẩu | 1.nhấn vào nhập Email  2.Nhập “hihi1104@gmail.com”  3.Nhấn vào Password  4.Nhập số “123456”  5.Nhấn vào đăng nhập | Hiển thị đăng nhập thành công | Hiển thị đăng nhập thành công | Pass |
| TC2 | Nhập sai Emai, sai Mật khẩu | 1.nhấn vào nhập Email  2.Nhập “hihi1103@gmail.com”  3.Nhấn vào Password  4.Nhập số “123455”  5.Nhấn vào đăng nhập | Hiển thị đăng nhập thất bại | Hiển thị đăng nhập thất bại | Fail |
| TC3 | Bỏ trống Email và Mật khẩu | 1.nhấn vào nhập Email  2.Nhập “”  3.Nhấn vào Password  4.Nhập số “”  5.Nhấn vào đăng nhập | Hệ thống hiển thị bạn nhập email,password | Hệ thống hiển thị bạn nhập email,password | pass |
| TC4 | Bỏ trống Email, nhập sai mật khẩu | 1.nhấn vào nhập Email  2.Nhập “”  3.Nhấn vào Password  4.Nhập số “123455”  5.Nhấn vào đăng nhập | Hiển thị đăng nhập thất bại | Hiển thị đăng nhập thất bại | Pass |
| TC5 | Bỏ trống Email, nhập đúng Mật khẩu | 1.nhấn vào nhập Email  2.Nhập “”  3.Nhấn vào Password  4.Nhập số “123  456”  5.Nhấn vào đăng nhập | Hiển thị đăng nhập thất bại | Hiển thị đăng nhập thất bại | Pass |
| TC6 | Nhập đúng Email, sai Mật khẩu | 1.nhấn vào nhập Email  2.Nhập “hihi1104@gmail.com”  3.Nhấn vào Password  4.Nhập số “123  455”  5.Nhấn vào đăng nhập | Hiển thị đăng nhập thất bại | Hiển thị đăng nhập thất bại | Pass |
| TC7 | Nhập sai Email, đúng Mật khẩu | 1.nhấn vào nhập Email  2.Nhập “hihi1103@gmail.com”  3.Nhấn vào Password  4.Nhập số “123  456”  5.Nhấn vào đăng nhập | Hiển thị đăng nhập thất bại | Hiển thị đăng nhập thất bại | Pass |
| TC8 | Nhập đúng Email, bỏ trống Mật khẩu | 1.nhấn vào nhập Email  2.Nhập “hihi1104@gmail.com”  3.Nhấn vào Password  4.Nhập số “”  5.Nhấn vào đăng nhập | Hiển thị đăng nhập thất bại | Hiển thị đăng nhập thất bại | Pass |
| TC-9 | Nhập sai Email, bỏ trống Mật khẩu | 1.nhấn vào nhập Email  2.Nhập “hihi1103@gmail.com”  3.Nhấn vào Password  4.Nhập số “”  5.Nhấn vào đăng nhập | Hiển thị đăng nhập thất bại | Hiển thị đăng nhập thất bại | Pass |

Bảng test report chức năng đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| Số lượng test case | Số lượng passed | Số lượng fail | Số lượng test không chạy |
| 9 | 8 | 1 | 0 |

* 1. **Kết quả test**

A screenshot of a computer

Description automatically generatedTìm kiếm sản phẩm

# Chức năng đăng nhập

A screenshot of a computer

Description automatically generated

# **LỜI KẾT**

Trong thời đại ngày nay, sự phổ biến của ứng dụng di động đã đặt ra những thách thức mới đối với quá trình phát triển và đảm bảo chất lượng phần mềm. Đề tài "Kiểm thử tự động Appium và ứng dụng vào Android" không chỉ là một nỗ lực nghiên cứu mà còn là một bước tiến quan trọng để giải quyết những thách thức này.

Chúng ta đã đi sâu vào việc tìm hiểu về cơ sở lý thuyết của kiểm thử phần mềm, đồng thời áp dụng công cụ Appium để thực hiện kiểm thử tự động trên nền tảng Android. Những kết quả và thảo luận từ nghiên cứu này đã làm rõ về sức mạnh và tính linh hoạt của Appium trong quá trình đảm bảo chất lượng ứng dụng di động.

Mục tiêu chính của đề tài là không chỉ đề xuất một phương pháp hiệu quả để thực hiện kiểm thử tự động mà còn khám phá các cơ hội và thách thức trong quá trình triển khai. Chúng ta hy vọng rằng kết quả từ nghiên cứu này sẽ cung cấp một cái nhìn sâu sắc về tầm quan trọng của kiểm thử tự động, đặc biệt là khi áp dụng vào môi trường Android ngày nay.

Cùng với sự tiến bộ của công nghệ, đề tài này là một bước nhảy vọt trong việc tối ưu hóa quy trình kiểm thử và đồng thời giúp những người phát triển và chuyên gia kiểm thử có cái nhìn rõ ràng về việc tích hợp Appium vào quá trình phát triển ứng dụng di động.

Cuối cùng, chúng ta nhận ra rằng hành trình này chỉ là một phần nhỏ trong sự phát triển không ngừng của lĩnh vực kiểm thử phần mềm và hi vọng rằng nghiên cứu này sẽ làm phong phú thêm bảng kiến thức và thảo luận trong cộng đồng chuyên gia phần mềm.

# 

# **TÀI LIỆU THAM KHẢO**

1. Giáo trình kiểm thử phần mềm
2. https://viblo.asia, QA
3. <https://www.lambdatest.com/appium>
4. <https://bizflycloud.vn/tin-tuc/appium-la-gi-20220701152542932.htm>
5. <https://en.wikipedia.org/wiki/Appium>