|  |
| --- |
| **ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  **TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  **KHOA: CÔNG NGHỆ THÔNG TIN**  **A blue and white logo  Description automatically generated**  **BÁO CÁO ĐỒ ÁN SEMINAR**  **Bảng kế hoạch**  **Đề tài: Automation testing for mobile apps**  **MÔN: KIỂM THỬ PHẦN MỀM**    **NHÓM THỰC HIỆN – NHÓM F2:**  **MSSV: 20120049 – HỌ TÊN: Nguyễn Hải Đăng**  **MSSV: 20120084 – HỌ TÊN: Nguyễn Văn Hiếu**  **MSSV: 20120458 – HỌ TÊN: Hồ Sĩ Đức**  **MSSV: 20120467 – HỌ TÊN: Nguyễn Phước Hải**    **Giảng viên hướng dẫn: ThS. Trần Duy Hoàng**  **Giảng viên thực hành: ThS. Trương Phước Lộc**  **Lớp lý thuyết: 20\_1**  **Học kỳ - Niên khoá: HK1 - 2023-2024** |

MỤC LỤC

[I. Danh sách thành viên – Phân công bài tập 3](#_Toc148801265)

[II. Lý thuyết 3](#_Toc148801266)

[1. What? 3](#_Toc148801267)

[A. Tìm hiểu về mobile testing và automation mobile testing 3](#_Toc148801268)

[B. Các bước thực hiện automation mobile tesing 4](#_Toc148801269)

[C. Một số tool được sử dụng cho automatio mobile testing hiện nay 5](#_Toc148801270)

[D. Ví dụ về automation mobile testing trong thực tế 5](#_Toc148801271)

[2. Why? 5](#_Toc148801272)

[A. Lợi ích của Automation Testing for mobile apps 6](#_Toc148801273)

[B. So sánh giữa Automation Testing và Manual Testing 7](#_Toc148801275)

[3. Who? 8](#_Toc148801276)

[1) Automation Testers 8](#_Toc148801277)

[2) QA Engineers 9](#_Toc148801278)

[3) Developers 9](#_Toc148801279)

[4) Testing Service Providers 9](#_Toc148801280)

[4. When? 9](#_Toc148801281)

[5. Difficulty? 10](#_Toc148801282)

[III. Thực hành – Demo 12](#_Toc148801283)

[1. Công cụ kiểm thử: Katalon 12](#_Toc148801284)

[2. SUT 12](#_Toc148801285)

[3. Kịch bản demo 13](#_Toc148801286)

# Danh sách thành viên – Phân công bài tập

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Mã số sinh viên** | **Phân công** |
| Nguyễn Hải Đăng | 20120049 | Tìm hiểu công cụ kiểm thử Katalon, xác định SUT, viết kịch bản demo. |
| Nguyễn Văn Hiếu | 20120084 | Khi nào cần thực hiện Automation Testing trên Mobile, những khó khăn trong quá trình Automation testing. So sánh sự khác nhưng của quá trình Automation Testing trên hai nền tảng Android và iOS |
| Hồ Sĩ Đức | 20120458 | Giới thiệu sơ lược về mobile testing và automation mobile testing, các bước thực hiện automation mobile testing và một số tools được sử dụng hiện nay. |
| Nguyễn Phước Hải | 20120467 | Tại sao cần automation testing for mobile và Ai là người thực hiện automation testing for mobile |

# Lý thuyết

## What?

### Tìm hiểu về mobile testing và automation mobile testing

Mobile testing là quá trình kiểm tra chức năng, hiệu suất, khả năng sử dụng và tính nhất quán trên các thiết bị di động như điện thoại di động, máy tính bảng. Mục tiêu chính của mobile testing là đảm bảo rằng ứng dụng hoạt động đúng và tương thích trên các thiết bị khác, nền tảng, hệ điều hành khác nhau và trong điều kiện sử dụng khác nhau.

Mobile testing được chia làm 2 loại là:

* Kiểm thử thủ công (manual testing): là quá trình kiểm thử hoàn toàn bằng con người, tester sẻ sử dụng các chiến lược, giải phép để tìm ra và ghi nhận lại vấn đề mà ứng dụng di động gặp phải. Tất cả quá trình trên đều được thực hiện thủ công.
* Kiểm thử tự động (automation testing): là quá trình kiểm thử bỏ qua một số nhiệm vụ nhất định khỏi con người bằng cách sử dụng các công cụ kiểm thử (tools) và kịch bản tự động, giải phóng tester tập trung vào các nhiệm vụ kiểm thử khác.

### Các bước thực hiện automation mobile tesing

1. Chuẩn bị môi trường thực thi
2. Thiết kế test case
3. Thiết kế test script
4. Thực hiện kiểm thử tự động
5. Phân tích kết quả kiểm thử

### Một số tool được sử dụng cho automatio mobile testing hiện nay

Hiện nay có rất nhiều tool hỗ trợ thực hiện automation mobile testing mà tiêu biểu trong số đó là: Appium, Katalon, Google Espresso, XCTest & XCUITest, Quantum.

Trong chủ để của seminar lần này nhóm sẽ tập trung giới thiệu 2 tool đó là: Appium và Katalon.

## Why?

### Lợi ích của Automation Testing for mobile apps

* Hiệu suất , tốc độ và tiết kiệm chi phí
* Khả năng tái sử dụng
* Tính nhất quán
* Phạm vi kiểm
* Độ chính xác
* Tối ưu hóa
* Khả năng mở rộng
* Cải thiện báo cáo về kiểm thử



### So sánh giữa Automation Testing và Manual Testing

|  |  |  |
| --- | --- | --- |
|  | Automation Testing | Manual Testing |
| Tốc độ và Hiệu Quả | Chạy nhanh hơn | Chạy chậm hơn |
| Phạm Vi Kiểm Thử | Rộng và toàn diện. | Không hiệu quả cho việc kiểm thử toàn diện và lặp lại. |
| Độ Chính Xác | Giảm thiểu sai sót, cung cấp kết quả chính xác và nhất quán. | Có khả năng sai sót do yếu tố con người. |
| Tính Nhất Quán | Thực hiện bước kiểm thử giống nhau mỗi lần, không bị ảnh hưởng bởi yếu tố con người. | Có khả năng thiếu nhất quán khi thực hiện kiểm thử nhiều lần. |
| Thích Ứng với Thay Đổi | Tiết kiệm thời gian và công sức với kiểm thử lặp lại | Tốn nhiều thời gian và công sức đối với kiểm thử lặp lại. |
| Báo Cáo Kiểm Thử | Báo cáo tự động cung cấp thông tin chi tiết về tỷ lệ bao phủ kiểm thử và thời gian thực hiện. | Tốn thời gian và công sức để thực hiện báo cáo , có khả năng xảy ra sai sót |

## Who?

### Automation Testers

* Vai trò: Người kiểm thử tự động chuyên nghiệp là chuyên gia trong việc tạo và thực hiện các kịch bản kiểm thử tự động.

### QA Engineers

* Vai trò: Kỹ sư Kiểm định Chất lượng (QA) có kỹ năng tự động hóa đóng góp vào cả kiểm thử thủ công và tự động.

### Developers

* Vai trò: Trong một số trường hợp, developer có thể đóng góp vào kiểm thử tự động, đặc biệt là trong kiểm thử đơn vị và tích hợp.

### Testing Service Providers

* Vai trò: Một số tổ chức có thể chọn gửi nhu cầu kiểm thử của họ tới các nhà cung cấp dịch vụ kiểm thử chuyên nghiệp.

## When?

Khi nào thực hiện Automation Testing cho Mobile?

1. **Kiểm thử tự động cho các kịch bản kiểm thử lặp đi lặp lại**: Khi có các kịch bản kiểm thử mà cần được thực hiện nhiều lần, automation testing có thể giúp tiết kiệm thời gian và tăng tính chính xác.
2. **Kiểm thử tích hợp**: Khi cần kiểm tra sự tương tác giữa ứng dụng di động và các thành phần hệ thống bên ngoài, automation testing có thể giúp đảm bảo rằng tích hợp diễn ra một cách đúng đắn.
3. **Kiểm thử đa nền tảng và đa thiết bị:** Khi ứng dụng cần hoạt động trên nhiều loại thiết bị và nền tảng khác nhau, automation testing có thể giúp đảm bảo tính nhất quán và tính đa dạng của ứng dụng trên các nền tảng này.
4. **Kiểm thử hiệu suất và độ ổn định:** Khi cần đánh giá hiệu suất và độ ổn định của ứng dụng dưới tải, automation testing có thể tạo ra tải tự động và đo lường hiệu suất theo cách thức được thiết lập trước đó.
5. **Kiểm thử đa ngôn ngữ:** Khi ứng dụng hỗ trợ nhiều ngôn ngữ, automation testing có thể giúp đảm bảo rằng tất cả các phiên bản ngôn ngữ của ứng dụng hoạt động đúng như mong đợi trên mọi thiết bị và hệ điều hành.
6. **Kiểm thử tương thích thiết bị và hệ điều hành:** Automation testing có thể giúp kiểm tra tính tương thích của ứng dụng trên nhiều loại thiết bị di động và phiên bản hệ điều hành khác nhau.

## Difficulty?

Những khó khăn trong việc thực hiện Automation Testing for Mobile:

1. **Tính đa dạng của thiết bị:** Với sự đa dạng về kích cỡ màn hình, độ phân giải, hệ điều hành và cấu hình phần cứng của các thiết bị di động, việc xây dựng kịch bản kiểm thử tự động có thể trở nên phức tạp để đảm bảo rằng ứng dụng hoạt động đúng trên tất cả các nền tảng này.
2. **Thay đổi thường xuyên trong ứng dụng:** Khi ứng dụng thường xuyên được cập nhật với các tính năng mới, việc duy trì và cập nhật các kịch bản kiểm thử tự động trở nên phức tạp và tốn kém.
3. **Kỹ năng kỹ thuật yêu cầu cao:** Việc xây dựng và triển khai automation testing đòi hỏi kiến thức vững về lập trình, kiến thức về kiểm thử và hiểu biết sâu rộng về cách thức hoạt động của ứng dụng và thiết bị di động.
4. **Không thể thay thế hoàn toàn kiểm thử thủ công:** Mặc dù automation testing có thể tăng tính hiệu quả và độ chính xác của quá trình kiểm thử, nhưng nó không thể thay thế hoàn toàn kiểm thử thủ công, đặc biệt là trong việc đánh giá trải nghiệm người dùng và các tình huống không dự đoán được.
5. **Khó khăn trong việc kiểm tra giao diện người dùng:** Mặc dù có các công cụ để kiểm tra giao diện người dùng, việc xác minh tính tương thích và tính thẩm mỹ của giao diện người dùng trên các thiết bị và nền tảng khác nhau vẫn là một thách thức.

So sánh Automation Testing giữa hai hệ điều hành phổ biến: Android và iOS

1. **Ngôn ngữ lập trình:** Android hỗ trợ việc sử dụng Java, Kotlin và một số ngôn ngữ khác, trong khi iOS tập trung chủ yếu vào Swift và Objective-C. Việc lựa chọn ngôn ngữ lập trình cũng ảnh hưởng đến việc lựa chọn các công cụ kiểm thử tự động.
2. **Đa dạng thiết bị:** Android có một loạt lớn các thiết bị với nhiều kích cỡ màn hình, độ phân giải và cấu hình phần cứng khác nhau, trong khi iOS có số lượng thiết bị ít hơn và tập trung hơn về đồng nhất về phần cứng.
3. **Sự phát triển của công cụ kiểm thử:** Cả hai hệ điều hành đều có các công cụ kiểm thử mạnh mẽ và phổ biến như Appium, Espresso, XCTest, và XCUITest. Tuy nhiên, cộng đồng Android thường có nhiều sự lựa chọn hơn và mở rộng hơn so với cộng đồng iOS.
4. **Quy trình kiểm thử và phê duyệt ứng dụng:** Quy trình phê duyệt ứng dụng trên App Store của Apple thường khắt khe hơn so với Google Play Store. Điều này có thể ảnh hưởng đến quá trình kiểm thử và phát triển ứng dụng trên hai nền tảng này.
5. **Hiệu suất ứng dụng:** Android có đa dạng thiết bị hơn, dẫn đến việc kiểm thử hiệu suất có thể phức tạp hơn so với iOS. Điều này đặc biệt đúng khi cần kiểm tra hiệu suất trên nhiều loại thiết bị có cấu hình phần cứng khác nhau.
6. **Khả năng tương tác với phần cứng và hệ thống nền tảng:** Android thường cung cấp nhiều khả năng tương tác với phần cứng và hệ thống nền tảng hơn so với iOS. Điều này có thể ảnh hưởng đến việc xây dựng các kịch bản kiểm thử tự động.
7. **Cơ chế bảo mật:** iOS có cơ chế bảo mật nghiêm ngặt hơn so với Android, điều này có thể ảnh hưởng đến quá trình kiểm thử tự động, đặc biệt là khi thực hiện kiểm thử bảo mật.
8. **Quy trình phát triển và triển khai:** Quy trình phát triển và triển khai ứng dụng trên hai nền tảng cũng có những khác biệt đáng kể, và điều này có thể ảnh hưởng đến cách tiếp cận kiểm thử và triển khai các kịch bản kiểm thử tự động.

# Thực hành – Demo

## Công cụ kiểm thử: Katalon

Katalon Studio là một công cụ kiểm thử tự động và tự động hóa thử nghiệm phần mềm được phát triển bởi Katalon, Inc. Nó giúp các nhà phát triển và nhóm kiểm thử thực hiện kiểm tra tự động cho ứng dụng web, ứng dụng di động và dịch vụ web. Katalon Studio cung cấp một môi trường tích hợp để tạo, quản lý và chạy các kịch bản kiểm tra tự động một cách dễ dàng mà không cần biết nhiều về lập trình.

Các tính năng chính của Katalon Studio bao gồm:

* Giao diện dựa trên GUI: Katalon Studio cung cấp một giao diện đồ họa dễ sử dụng để tạo và quản lý các kịch bản kiểm tra, cho phép người dùng không cần có kiến thức chuyên sâu về lập trình.
* Hỗ trợ nhiều nền tảng: Nó hỗ trợ kiểm tra ứng dụng web, ứng dụng di động trên nhiều hệ điều hành như iOS và Android, và dịch vụ web.
* Báo cáo và lưu trữ kết quả: Katalon cung cấp báo cáo chi tiết về kết quả kiểm tra và lưu trữ lịch sử kiểm tra.

## SUT

* SUT là viết tắt của "System Under Test," tạm dịch là "Hệ thống đang được kiểm tra." Trong kiểm thử phần mềm, SUT là thành phần hoặc hệ thống cần được kiểm tra hoặc kiểm tra để đảm bảo tính đúng đắn, hiệu quả và đáng tin cậy của nó.
* Ứng dụng di động: Nếu chúng ta đang kiểm tra một ứng dụng di động như một ứng dụng cho điện thoại di động, thì ứng dụng đó chính là SUT.
* Trong đề tài này, **nhóm sẽ sử dụng ứng dụng Lazada** trên điện thoại di động làm SUT để kiểm thử tự động một số chức năng của phần mềm.

## Kịch bản demo

Trong khoảng thời gian được phép trình bày, nhóm sẽ tiến hành kiểm thử tự động từ 2 đến 3 chức năng trong số những chức năng sau đây:

* 1. Chức năng đăng nhập bằng Google (khi vào app lần đầu)
* Mục tiêu: Kiểm tra chức năng đăng nhập bằng Google (khi vào app lần đầu)
* Tiền điều kiện: App mới vô lần đầu tiên (đã xóa dữ liệu trước trên điện thoại).
* Bước 1: Nhấn vào tab “Tài khoản”.
* Bước 2: Chọn Button “Đăng nhập”.
* Bước 3: Chọn chức năng “Đăng nhập bằng Google”.
* Bước 4: Nhấn “Đồng ý” khi popup hiện lên.
* Bước 5: Chọn một account đã đăng ký từ trước.
* Bước 6: Bấm Back để hủy kích hoạt vân tay.
* Bước 7:
  + 7.1: Nếu thành công => Vô màn hình chính
  + 7.2: Nếu thất bại => Không vô màn hình chính.
* Kết quả mong đợi: Vô màn hình chính khi đăng nhập thành công.
  1. Chức năng thêm món hàng vào giỏ hàng
* Mục tiêu: Kiểm tra khả năng thêm sản phẩm vào giỏ hàng của ứng dụng
* Bước thực hiện:
  + Mở ứng dụng Lazada.
  + Nhấn vào sản phẩm muốn thêm vào giỏ hàng.
  + Nhấn vào nút "Thêm vào giỏ hàng".
  + Kiểm tra xem sản phẩm đã được thêm vào giỏ hàng hay chưa.
* Kết quả mong đợi: Sản phẩm đã được thêm vào giỏ hàng.
  1. Chức năng tìm kiếm món hàng
* Mục tiêu: Kiểm tra tính năng chat và hỗ trợ khách có bị lỗi
* Bước thực hiện:
  + Nhấn vào nút "Tìm kiếm".
  + Nhập từ khóa tìm kiếm.
  + Nhấn vào nút "Tìm kiếm".
  + Kiểm tra xem danh sách sản phẩm có phù hợp với từ khóa tìm kiếm hay không.
  + Kiểm tra xem người dùng có thể lọc sản phẩm theo giá, thương hiệu, v.v. hay không.
* Kết quả mong đợi: Tìm kiếm thành công.
  1. Chức năng mua hàng (thanh toán bằng tiền mặt)
* Mục đích: Kiểm tra tính năng mua hàng trên app Lazada có thể thực hiện thành công hay không (thanh toán bằng tiền mặt)
* Dữ liệu đầu ra: Đơn hàng được tạo thành công.
  + Bước 1: Mở app Lazada và truy cập vào trang chủ.
  + Bước 2: Thêm sản phẩm vào giỏ hàng.
  + Bước 3: Chọn thông tin người mua hàng, thông tin vận chuyển và phương thức thanh toán.
  + Bước 4: Chọn hình thức thanh toán bằng tiền mặt.
  + Bước 5: Nhấn nút Đặt hàng.
  + Bước 6: Kiểm tra đơn hàng đã được tạo thành công hay không.