**FUNCTIONAL TESTING**

**AND NON-FUNCTIONAL TESTING**

1. **Function**
2. **Giới thiệu Functional Testing**

* *Functional testing là loại kiểm thử* ***kiểm tra chức năng*** *của một thành phần hoặc toàn bộ hệ thống để đảm bảo rằng nó* ***hoạt động đúng như yêu cầu và mong đợi****.*

1. **Mục đích Functional testing**

* Kiểm tra các chức năng của phần mềm có hoạt động chính xác và đúng như mong đợi hay không.

1. **Phạm vi áp dụng Functional testing**

* Functional testing nên được áp dụng ở tất cả test levels (Unit Testing – Integration Testing – System Testing – Acceptance Testing).

1. **Test coverage của Functional testing**

* Mức độ bao phủ của Functional testing là phạm vi của các chức năng đã được kiểm thử và được thể hiện bằng phầm trăm số lượng chức năng đã được kiểm tra trên tổng số chức năng trong phạm vi test.
* Ví dụ: Khi ta đã hoàn thành kiểm thử 95 chức năng trong tổng số 100 chức năng có thể test trong hệ thống, ta nói rằng test coverage của hệ thống là 95%.

1. **Quy trình thực hiện function testing**

* Bước 1: Xác định yêu cầu và mục tiêu.
  + Đọc hiểu tài liệu…
  + phân tích yêu cầu, hiểu rõ các chức năng…
  + giới hạn input/output….
* Bước 2: Lập kế hoạch: Xác định phạm vi, viết testcase/check list.
* Bước 3: Chuẩn bị môi trường: Cài đặt và cấu hình các công cụ, thư viện, dữ liệu
* Bước 4: Thực hiện kiểm thử : tiến hành chạy các bộ test cases và ghi lại các kết quả thực tế để so sánh với kết quả mong đợi.
* Bước 5: Báo cáo lỗi: Viết báo cáo về quá trình kiểm thử.

Bước 6: Theo dõi và retest :

1. **Test types phổ biến**: Black box, Smoke testing, Integration testing, Interface testing, Kiểm thử hồi quy, …

* Smoke testing là một loại kiểm thử phần mềm được thực hiện sau khi có một bản build mới, để đảm bảo rằng các chức năng chính, quan trọng của phần mềm vẫn hoạt động bình thường.
* Sanity testing: là một loại kiểm thử phần mềm được thực hiện sau khi nhận được bản build, ở bản build này có một số chức năng của phần mềm được chỉnh sửa, cập nhật.Việc này giúp kiểm tra các thành phần chức năng khác có bị ảnh hưởng hay không.

1. **Phân tích kỹ thuật test phổ biến nhất**

Kỹ thuật phổ biến nhất của function testing là: Black box

Black box bao gồm các kỹ thuật viết test case sau:

* Kỹ thuật phân vùng tương đương
* Kỹ thuật phân tích giá trị biên
* Kỹ thuật bảng quyết định
* Kỹ thuật kiểm thử chuyển trạng thái
* Kỹ thuật kiểm thử theo Use Case

\***Kỹ thuật phân vùng tương đương**: sử dụng khi phân vùng dữ liệu hợp lệ và không hợp lệ.

(Vùng tương đương là vùng mà tập hợp các input cho cùng kết quả output hoặc các output có cùng input

Phân vùng chứa các giá trị hợp lệ gọi là phân vùng tương tương hợp lệ

Phân vùng chứa các giá trị không hợp lệ gọi là phân vùng tương tương không hợp lệ)

- Cách sử dụng: Chọn một giá trị đại diện cho phân vùng để thực hiện kiểm tra.

Ví dụ 1: Một form tạo mật khẩu mới của ZALO như sau, yêu cầu field mật khẩu mới phải từ 8-32 ký tự.(Viết trường hợp test)



|  |  |
| --- | --- |
| Mô tả | Kết quả dự kiến |
| Nhập 0-7 kí tự | Hệ thống không chấp nhận:  Không nhập -> nút Cập nhật bị disable.  Nhập nhỏ hơn <= 7 kí tự : |
| Nhập 8-32 kí tự | Hệ thống chấp nhận |
| Nhập 33-50 kí tự | Hệ thống không chấp nhận |

Ví dụ 2: Một sản phẩm Thước Nhựa, số lượng sản phẩm đang có sẵn chỉ còn 5 chiếc thước

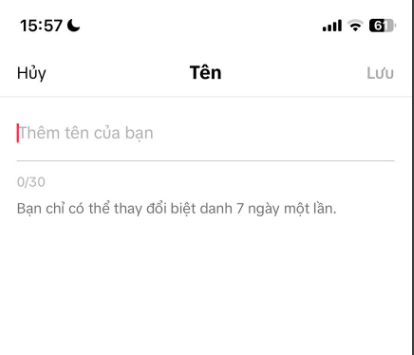


|  |  |
| --- | --- |
| Mô tả | Kết quả dự kiến |
| Nhập số lượng bất kì từ 0 đến âm vô cùng.  Cụ thể: -1 | Hệ thống không cho nhập |
| Nhập số lượng từ 1-5  Cụ thể: 2 | Hệ thống chấp nhận |
| Nhập từ 6 đến dương vô cùng.  Cụ thể: 7 | Hệ thống không cho nhập. |

\***Kỹ thuật phân tích giá trị biên**: sử dụng khi kiểm tra các giá trị tại vị trí biên, vị trí cận biên.

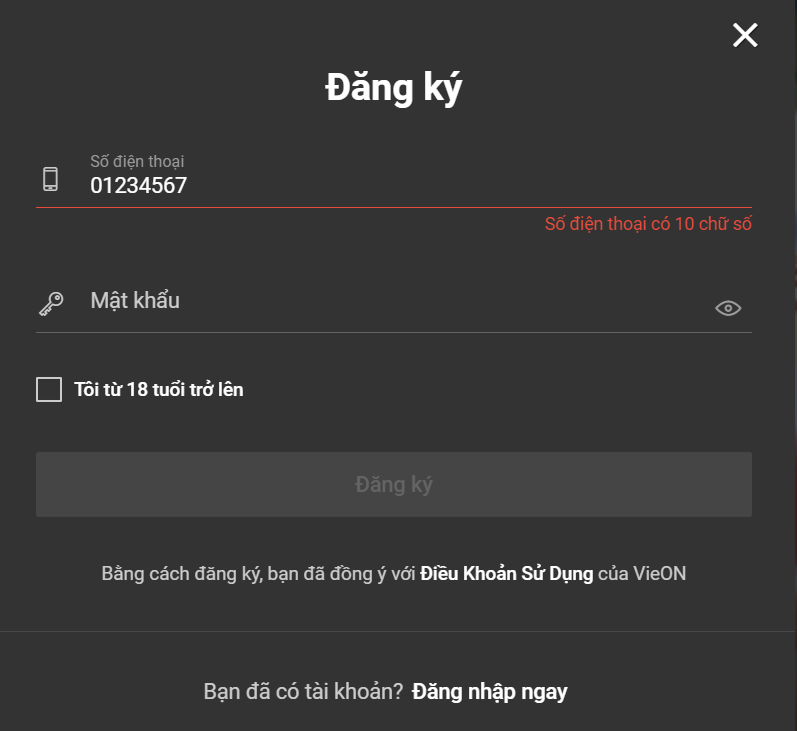
Cách sử dụng: Lấy giá trị tại vị trí min, max, và cận biên như min-1,max+1 để kiểm thử giá trị có hợp lệ hay không.

Ví dụ 1: Form đổi tên User của TikTok như sau, tối đa chỉ nhập 30 kí tự



|  |  |
| --- | --- |
| Giá trị | Kết quả mong đợi |
| Nhập 29 kí tự | Hệ thống chấp nhận |
| Nhập 30 kí tự | Hệ thống chấp nhận |
| Nhập 31 kí tự | Hệ thống không chấp nhận, message error: “Đã vượt quá giới hạn kí tự” |
| Empty | Hệ thống không chấp nhận, nút Lưu bị disable |

Ví dụ 2: Form đăng ký tài khoản VieOn như sau, yêu cầu số điện thoại phải có 10 chữ số và không được để trống



|  |  |
| --- | --- |
| Giá trị | Kết quả mong đợi |
| Nhập 10 kí tự số | Hệ thống chấp nhận |
| Nhập 9 kí tự số | Hệ thống không chấp nhận |
| Nhập 11 kí tự số | Hệ thống không chấp nhận |
| Không nhập | Hệ thống không chấp nhận |

\***Kỹ thuật dựa trên Bảng quyết định** : Sử dụng khi có nhiều điền kiện cần xử lí, các điều kiện đó có thể kết hợp được với nhau.

- Cách sử dụng: Liệt kê các điều kiện,trường hợp có sự liên quan với nhau , khi kết hợp các điều kiện lại sẽ dẫn đến một output.

\***Kiểm thử Chuyển đổi trạng thái :** Test ứng dụng cho một bộ giá trị input hữu hạn, Không thực hiện đối với các input tuần tự.

**-Cách sử dụng:** thay đổi input dẫn đến sự thay đổi trạng thái của phần mềm hoặc giá trị output

Ví dụ 2: Mua hàng của Shoppe



\***Kỹ thuật kiểm thử theo mô hình Use Case** : Dùng khi muốn mô tả flow mà user sẽ gặp khi sử dụng hệ thống,test cases mô tả từ đầu đến cuối hành vi của hệ thống.

-Cách sử dụng: Lập kịch bảng và thực hiện chuỗi các thao tác mà user sẽ thực hiện khi sử dụng phần mềm, để kiểm tra sự logic.

Ví dụ : Mô tả luồng hoạt động của Login của ngân hàng BIDV, người dùng nhập vào username và password:

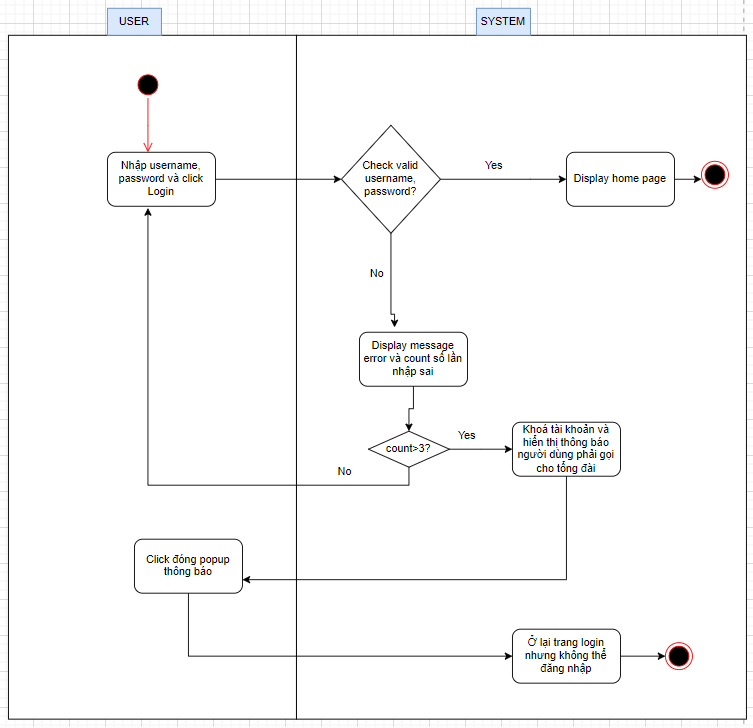
+Nhập đúng username và password , hệ thống chuyển hướng đến page home

+ Nhập username không tồn tại, hệ thống bắt nhập lại

+Nhập sai password 3 lần,hệ thống bắt nhập lại

+Nhập sai password 3 lần, nhưng lần thứ 4 nhập đúng -> hệ thống chuyển hướng đến page home

+Nhập sai password 4 lần, hiển thị thông báo “Tài khoản của bạn đã bị khoá”



1. **NON-FUNCTION**
2. **Giới thiệu non-functional**

* Là loại kiểm thử tập trung vào các thuộc tính phi chức năng của phần mềm như: hiệu năng, độ bảo mật, độ tin cậy, khả năng phục hồi và tương tác người dùng…

1. **Mục đích Non-functional**

* Tăng độ tin cậy, độ bảo mật, sự trải nghiệm và tương tác của người dùng đối với phần mềm

1. **Phạm vi sử dụng Non-functional**

* Non-functional testing nên được áp dụng ở tất cả test levels (Unit Testing – Integration Testing – System Testing – Acceptance Testing).

1. **Test types phổ biến**: Kiểm thử hiệu năng(Performance testing), kiểm thử độ tải(Load testing), kiểm thử vượt tải(Stress testing), kiểm thử tính khả dụng(Usability testing), kiểm thử độ tin cậy(Reliability testing), kiểm thử tính tương thích(Portability testing),…
2. **Phân tích các kỹ thuật phổ biến**
   1. **Kiểm thử hiệu năng** **(Performance testing):** đảm bảo ứng dụng hoạt động tốt với một số lượng công việc nhất định, tập trung vào kiểm *tra tốc độ phản hồi, khả năng mở rộng, sự ổn định.*
   2. **Kiểm thử tính khả dụng(Usability testing):** tập trung vào *sự dễ hiểu, dễ dùng, hấp dẫn* được người khác sử dụng.
   3. **Kiểm thử tính tương thích**: kiểm thử trên trình duyệt hoặc thiết bị khác để kiểm chứng *khả năng thay thế, thích nghi* của phần mềm trên môi trường khác.
   4. **Kiểm thử độ tin cậy(Reliability testing):** kiểm tra để đảm bảo phần mềm có thể hoạt động xuyên suốt thời gian mà không phát sinh lỗi, trong trường hợp có sự cố phần mềm phải nhanh chóng được khôi phục.
3. **So sánh functional testing và non-function testing**

|  |  |
| --- | --- |
| Functional testing | Non-functional testing |
| Tập trung kiểm thử các tính năng và chức năng | Tập trung kiểm thử phi chức năng: hiệu suất, độ tin cậy, khả năng bảo mật,… |
| Dễ thực hiện thủ công | Khó thực hiện thủ công |
| Giúp phát hiện lỗi, rủi ro trong các chức năng | Cải thiện hiệu suất phần mềm |
| Thực hiện trước kiểm thử phi chức năng | Thực hiện sau khi kiểm thử chức năng |
| Kiểm tra tính đúng đắn, chính xác của các chức năng | Kiểm tra tốc độ và thời gian phản hồi, phục hồi, sự dễ dùng,độ bảo mật,… |