BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**QUẢN LÝ KHÁCH SẠN**

**GVHD: Phan Thị Ngọc Mai**

**SINH VIÊN THỰC HIỆN**

1. 2001180233 – Võ Hoàng Bảo Sơn

2. 2001181152 – Huỳnh Quốc Khang

3. 2001181429 – Trần Nhật Hạ Vy

TP. HỒ CHÍ MINH, tháng 05 năm 2022

**LỜI CẢM ƠN**

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép em được bày tỏ lòng biết ơn đến quý thầy cô ở Khoa Công nghệ Thông tin đã tạo môi trường học tập và truyền đạt vốn kiến thức quý báu cho chúng em trong khoảng thời gian theo học tại trường Đại học Công nghiệp Thực Phẩm TP.HCM. Nhờ có những lời hướng dẫn, dạy bảo và sự giúp đỡ của các thầy cô nên nhóm em đã có thể hoàn thành đề tài một cách tốt đẹp.

Em xin chân thành cảm ơn đến cô Phan Thị Ngọc Mai - người đã trực tiếp giúp đỡ, góp ý và hướng dẫn nhóm em hoàn thành đề tài này trong suốt khoảng thời gian qua.

Với khoảng thời gian ngắn ngủi, nhóm em có thể sẽ không tránh khỏi những thiếu sót, nhóm em rất mong sẽ nhận được những ý kiến đóng góp quý báu của cô để nhóm em có thể nắm vững được những kiến thức đồng thời có điều kiện bổ sung nâng cao ý thức của mình.

Nhóm em xin chân thành cảm ơn!

[CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM 3](#_Toc104419208)

[1.1. Các khái niệm cơ bản về kiểm thử phần mềm 3](#_Toc104419209)

[1.1.1. Kiểm thử phần mềm là gì? 3](#_Toc104419210)

[1.1.2. Các phương pháp kiểm thử 3](#_Toc104419211)

[1.1.3. Các chiến lược kiểm thử 4](#_Toc104419212)

[1.1.4. Các cấp độ kiểm thử phần mềm 6](#_Toc104419213)

[1.1.5. Các phương pháp kiểm thử con người 10](#_Toc104419214)

[1.2. Nguyên tắc kiểm thử phần mềm 11](#_Toc104419215)

[CHƯƠNG 2.THIẾT KẾ TEST PLAN 13](#_Toc104419216)

[2.1. Tổng quang. 13](#_Toc104419217)

[2.1.1. Khái niệm Test Plan 13](#_Toc104419218)

[2.1.2. Phân tích sản phẩm 13](#_Toc104419219)

[2.1.3 Các phần mềm sử dụng 13](#_Toc104419220)

[2.2. Xây dựng chiến lược kiểm thử. 13](#_Toc104419221)

[2.2.1. Phạm vi kiểm thử 13](#_Toc104419222)

[2.2.2. Xác định loại kiểm thử. 14](#_Toc104419223)

[2.2.3. Tài liệu rủi ro và vấn đề. 14](#_Toc104419224)

[2.3.4. Test Logistics 14](#_Toc104419225)

[2.3.5. Xác định đối tượng kiểm thử 16](#_Toc104419226)

[2.3.6. Xác định tiêu chí kiểm thử 16](#_Toc104419227)

[CHƯƠNG 3: THIẾT KẾ TEST – CASE 17](#_Toc104419228)

[3.1. Khái niệm 17](#_Toc104419229)

[3.2. Vai trò của thiết kế test – case 17](#_Toc104419230)

[2.2. Quy trình của thiết kế test – case 17](#_Toc104419231)

[3.3.2. Chiến lược 26](#_Toc104419232)

[CHƯƠNG 4: ÁP DỤNG 28](#_Toc104419233)

[4.1 WBS 28](#_Toc104419234)

[4.2 Đặc tả 28](#_Toc104419235)

[4.3 Kiểm tra hộp đen 29](#_Toc104419236)

[4.4 Kiểm thử chức năng 38](#_Toc104419237)

[4.4.1 Chức năng đăng nhập 39](#_Toc104419238)

[4.4.2. Chức năng Quản lý khách hàng 42](#_Toc104419239)

[4.4.3. Chức năng Quản lý nhân viên 54](#_Toc104419240)

[KẾT LUẬN 59](#_Toc104419241)

[TÀI LIỆU THAM KHẢO 60](#_Toc104419242)

# CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

## Các khái niệm cơ bản về kiểm thử phần mềm

### Kiểm thử phần mềm là gì?

Kiểm thử phần mềm là quá trình khảo sát một hệ thống hay thành phần dưới những điều kiện xác định, quan sát và ghi lại các kết quả, và đánh giá một khía cạnh nào đó của hệ thống hay thành phần đó. Là quá trình thực thi một chương trình với mục đích tìm lỗi.

Mục đích của kiểm thử phần mềm là tìm ra các lỗi hay khiếm khuyết phần mềm nhằm đảm bảo hiệu quả hoạt động tối ưu của phần mềm trong nhiều ngành khác nhau. (Theo *Bách khoa toàn thư mở Wikipedia*).

### Các phương pháp kiểm thử

Có 2 phương pháp kiểm thử chính là: Kiểm thử tĩnh và Kiểm thử động.

#### Kiểm thử tĩnh – Static testing

Là phương pháp thử phần mềm đòi hỏi phải duyệt lại các yêu cầu và các đặc tả bằng tay, thông qua việc sử dụng giấy, bút để kiểm tra logic, lần từng chi tiết mà không cần chạy chương trình. Kiểu kiểm thử này thường được sử dụng bởi chuyên viên thiết kế người mà viết mã lệnh một mình.

Kiểm thử tĩnh cũng có thể được tự động hóa. Nó sẽ thực hiện kiểm tra toàn bộ bao gồm các chương trình được phân tích bởi một trình thông dịch hoặc biên dịch mà xác nhận tính hợp lệ về cú pháp của chương trình.

#### Kiểm thử động – Dynamic testing

Là phương pháp thử phần mềm thông qua việc dùng máy chạy chương trình để điều tra trạng thái tác động của chương trình. Kiểm thử động kiểm tra cách thức hoạt động của mã lệnh, tức là kiểm tra sự phản ứng vật lý từ hệ thống tới các biến luôn thay đổi theo thời gian. Trong kiểm thử động, phần mềm phải thực sự được biên dịch và chạy. Kiểm thử động thực sự bao gồm làm việc với phần mềm, nhập các giá trị đầu vào và kiểm tra xem liệu đầu ra có như mong muốn hay không.

### Các chiến lược kiểm thử

Hai trong số những chiến lược kiểm thử thông dụng nhất bao gồm: Kiểm thử hộp đen và Kiểm thử hộp trắng

#### Kiểm thử hộp đen – Black box testing

Một trong những chiến lược kiểm thử quan trọng là kiểm thử hộp đen, hướng dữ liệu, hay hướng vào/ra. Mục đích của bạn là tập trung vào tìm các trường hợp mà chương trình không thực hiện theo các đặc tả của nó.

Theo hướng tiếp cận này, dữ liệu kiểm tra được lấy chỉ từ các đặc tả.

#### Các phương pháp kiểm thử hộp đen:

* Phân lớp tương đương – Equivalence partitioning.
* Phân tích giá trị biên – Boundary value analysis.
* Kiểm thử mọi cặp – All-pairs testing.
* Kiểm thử fuzz – Fuzz testing.
* Kiểm thử dựa trên mô hình – Model-based testing.
* Ma trận dấu vết – Traceability matrix.
* Kiểm thử thăm dò – Exploratory testing.
* Kiểm thử dựa trên đặc tả – Specification-base testing.

Kiểm thử dựa trên đặc tả tập trung vào kiểm tra tính thiết thực của phần mềm theo những yêu cầu thích hợp. Do đó, kiểm thử viên nhập dữ liệu vào, và chỉ thấy dữ liệu ra từ đối tượng kiểm thử. Kiểm thử dựa trên đặc tả là cần thiết, nhưng không đủ để ngăn chặn những rủi ro chắc chắn.

#### Ưu, nhược điểm:

Kiểm thử hộp đen không có mối liên quan nào tới mã lệnh, và kiểm thử viên chỉ rất đơn giản tâm niệm là: một mã lệnh phải có lỗi. Những kiểm thử viên hộp đen tìm ra lỗi mà những lập trình viên đã không tìm ra. Nhưng mặt khác, kiểm thử viên không biết các phần mềm được kiểm tra thực sự được xây dựng như thế nào. Đó là lý do mà có nhiều trường hợp mà một kiểm thử viên hộp đen viết rất nhiều ca kiểm thử để kiểm tra một thứ gì đó mà đáng lẽ có thể chỉ cần kiểm tra bằng 1 ca kiểm thử duy nhất, và/hoặc một số phần của chương trình không được kiểm tra chút nào.

#### Kiểm thử hộp trắng – White box testing

Là một chiến lược kiểm thử trái ngược hoàn toàn với kiểm thử hộp đen, kiểm thử hộp trắng hay kiểm thử hướng logic cho phép bạn khảo sát cấu trúc bên trong của chương trình. Kiểm thử viên sẽ truy cập vào cấu trúc dữ liệu và giải thuật bên trong chương trình (và cả mã lệnh thực hiện chúng).

#### Các phương pháp kiểm thử hộp trắng

* + - * + Kiểm thử giao diện lập trình ứng dụng - API testing (application programming interface): là phương pháp kiểm thử của ứng dụng sử dụng các API công khai và riêng tư.
        + Bao phủ mã lệnh – Code coverage: tạo các kiểm tra để đáp ứng một số tiêu chuẩn về bao phủ mã lệnh.
        + Các phương pháp gán lỗi – Fault injection.
        + Các phương pháp kiểm thử hoán chuyển – Mutation testing methods.
        + Kiểm thử tĩnh – Static testing: kiểm thử hộp trắng bao gồm mọi kiểm thử tĩnh.

Phương pháp kiểm thử hộp trắng cũng có thể được sử dụng để đánh giá sự hoàn thành của một bộ kiểm thử mà được tạo cùng với các phương pháp kiểm thử hộp đen.

### Các cấp độ kiểm thử phần mềm

Kiểm thử phần mềm gồm có các cấp độ: Kiểm thử đơn vị, Kiểm thử tích hợp, Kiểm thử hệ thống và Kiểm thử chấp nhận sản phẩm.

Sơ đồ cấp độ kiểm thử:

Diagram

Description automatically generated

#### Kiểm thử đơn vị - Unit Test

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

Vì Unit được chọn để kiểm tra 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 thể Unit đang kiểm tra.

Unit Test 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. Mục đích của Unit Test là bảo đảm thông tin được xử lý và xuất (khỏi Unit) là chính xác, trong mối tương quan với dữ liệu nhập và chức năng của Unit. Điều này thường đòi hỏi tất cả các nhánh bên trong Unit 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 Unit.

Cùng với các mục kiểm thử khác, Unit Test cũng đòi hỏi phải chuẩn bị trước các ca kiểm thử (Test case) hoặc kịch bản kiểm thử (Test script), trong đó chỉ định rõ dữ liệu đầu vào, các bước thực hiện và dữ liệu đầu ra mong muốn.

#### Kiểm thử tích hợp – Intergration Test

Integration test kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành. Trong khi Unit Test kiểm tra các thành phần và Unit riêng lẻ thì Intgration Test kết hợp chúng lại với nhau và kiểm tra sự giao tiếp giữa chúng.

Hai mục tiêu chính của Integration Test:

* + - * + Phát hiện lỗi giao tiếp xảy ra giữa các Unit.
        + Tích hợp các Unit 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 (System Test).

Trong Unit Test, lập trình viên cố gắng phát hiện lỗi liên quan đến chức năng và cấu trúc nội tại của Unit.

Trừ một số ít ngoại lệ, Integration Test chỉ nên thực hiện trên những Unit đã được kiểm tra cẩn thận trước đó bằng Unit Test, và tất cả các lỗi mức Unit đã được sửa chữa.

Có 4 loại kiểm thử trong Integration Test:

* + - * + **Kiểm thử cấu trúc (Structure Test):** Tương tự White Box Test, kiểm thử cấu trúc nhằm bảo đảm các thành phần bên trong của một chương trình chạy đúng và chú trọng đến hoạt động của các thành phần cấu trúc nội tại của chương trình chẳng hạn các câu lệnh và nhánh bên trong.
        + **Kiểm thử chức năng (Functional Test):** Tương tự Black Box Test, kiểm thử chức năng chỉ chú trọng đến chức năng của chương trình, mà không quan tâm đến cấu trúc bên trong, chỉ khảo sát chức năng của chương trình theo yêu cầu kỹ thuật.
        + **Kiểm thử hiệu năng (Performance Test):** Kiểm thử việc vận hành của hệ thống.
        + **Kiểm thử khả năng chịu tải (Stress Test):** Kiểm thử các giới hạn của hệ thống.

#### Kiểm thử hệ thống – System Test

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

System Test bắt đầu khi tất cả các bộ phận của phần mềm đã được tích hợp thành công. Thông thường loại kiểm thử này tốn rất nhiều công sức và thời gian. Trong nhiều trường hợp, việc kiểm thử đòi hỏi một số thiết bị phụ trợ, phần mềm hoặc phần cứng đặc thù, đặc biệt là các ứng dụng thời gian thực, hệ thống phân bố, hoặc hệ thống nhúng.

Thông thường ta phải thực hiện Unit Test và Integration Test để bảo đảm mọi Unit và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện System Test.

System Test kiểm thử 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.

Bản thân System Test lại gồm nhiều loại kiểm thử khác nhau, phổ biến nhất gồm:

* **Kiểm thử chức năng (Functional Test):** Bảo đảm các hành vi của hệ thống thỏa mãn đúng yêu cầu thiết kế.
* **Kiểm thử hiệu năng (Performance Test):** Bảo đảm tối ưu việc phân bổ tài nguyên hệ thống (ví dụ bộ nhớ) nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng câu truy vấn, ...
* **Kiểm thử khả năng chịu tải (Stress Test hay Load Test):** Bảo đảm hệ thống vận hành đúng dưới áp lực cao (ví dụ nhiều người truy xuất cùng lúc). Stress Test tập trung vào các trạng thái tới hạn, các "điểm chết", các tình huống bất thường như đang giao dịch thì ngắt kết nối.

#### Kiểm thử cấu hình (Configuration Test).

* **Kiểm thử bảo mật (Security Test):** Bảo đảm tính toàn vẹn, bảo mật của dữ liệu và của hệ thống.
* **Kiểm thử khả năng phục hồi (Recovery Test):** Bảo đảm hệ thống có khả năng khôi phục trạng thái ổn định trước đó trong tình huống mất tài nguyên hoặc dữ liệu

Nhìn từ quan điểm người dùng, các cấp độ kiểm thử trên rất quan trọng: Chúng bảo đảm hệ thống đủ khả năng làm việc trong môi trường thực.

#### Kiểm thử chấp nhận sản phẩm – Acceptance Test

Thông thường, sau giai đoạn System Test là Acceptance Test, được khách hàng thực hiện (hoặc ủy quyền cho một nhóm thứ ba thực hiện). Mục đích của Acceptance Test 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).

Đối với những sản phẩm dành bán rộng rãi trên thị trường cho nhiều người sử dụng, thông thường sẽ thông qua hai loại kiểm thử gọi là kiểm thử Alpha – Alpha Test và kiểm thử Beta – Beta Test.

Gắn liền với giai đoạn Acceptance Test thường là một nhóm những dịch vụ và tài liệu đi kèm, phổ biến như hướng dẫn cài đặt, sử dụng v.v... Tất cả tài liệu đi kèm phải được cập nhật và kiểm thử chặt chẽ.

#### Một số cấp độ kiểm thử khác

Ngoài các cấp độ trên, còn một số cấp độ kiểm thử khác như:

#### Kiểm thử hồi quy – Regression Testing:

Kiểm thử hồi quy là “sự kiểm tra lại có lựa chọn của một hệ thống hay thành phần để xác minh là những sự thay đổi không gây ra những hậu quả không mong muốn…”.

#### Kiểm thử tính đúng – Correctness testing:

Tính đúng đắn là yêu cầu tối thiểu của phần mềm, là mục đích chủ yếu của kiểm thử. Kiểm thử tính đúng sẽ cần một kiểu người đáng tin nào đó, để chỉ ra cách hoạt động đúng đắn từ cách hoạt động sai lầm.

### Các phương pháp kiểm thử con người

Hai phương pháp kiểm thử con người chủ yếu là Code Inspections và Walkthroughs. Hai phương pháp này bao gồm một nhóm người đọc và kiểm tra theo mã lệnh của chương trình. Mục tiêu của chúng là để tìm ra lỗi mà không gỡ lỗi.

Một Inspection hay Walkthrough là 1 sự cải tiến của phương pháp kiểm tra mà lập trình viên đọc chương trình của họ trước khi kiểm thử nó

Hiệu quả tìm lỗi sẽ kém đi nếu thiếu đi 1 trong 2 phương pháp. Và đối với việc sửa đổi chương trình cũng nên sử dụng các phương pháp kiểm thử này cũng như các kỹ thuật kiểm thử hồi quy

#### Tổng duyệt – Walkthrough

Walkthrough mã lệnh là 1 tập các thủ tục và các công nghệ dò lỗi cho việc đọc nhóm mã lệnh. Trong một Walkthrough, nhóm các nhà phát triển – với 3 hoặc 4 thành viên là tốt nhất – thực hiện xét duyệt lại.

Một ưu điểm khác của walkthroughs, hiệu quả trong chi phí gỡ lỗi, là 1 thực tế mà khi một lỗi được tìm thấy, nó thường được định vị chính xác trong mã lệnh. Thêm vào đó, phương pháp này thường tìm ra 1 tập các lỗi, cho phép sau đó các lỗi đó được sửa tất cả với nhau. Mặt khác, kiểm thử dựa trên máy tính, chỉ tìm ra triệu chứng của lỗi (chương trình không kết thúc hoặc đưa ra kết quả vô nghĩa), và các lỗi thường được tìm ra và sửa lần lượt từng lỗi một.

#### Thanh tra mã nguồn – Code Inspection

Thanh tra mã nguồn là 1 tập hợp các thủ tục và các kỹ thuật dò lỗi cho việc đọc các nhóm mã lệnh.

## 1.2. Nguyên tắc kiểm thử phần mềm

Để kiểm thử đạt hiệu quả thì khi tiến hành kiểm thử phần mềm cần phải tuân thủ một số quy tắc sau:

**Quy tắc 1:** Một phần quan trọng của 1 ca kiểm thử là định nghĩa của đầu ra hay kết quả mong muốn.

**Quy tắc 2:** Lập trình viên nên tránh tự kiểm tra chương trình của mình.

**Quy tắc 3:** Nhóm lập trình không nên kiểm thử chương trình của chính họ.

**Quy tắc 4:** Kiểm tra thấu đáo mọi kết quả của mỗi kiểm tra.

**Quy tắc 5:** Các ca kiểm thử phải được viết cho các trạng thái đầu vào không hợp lệ và không mong muốn, cũng như cho các đầu vào hợp lệ và mong muốn.

**Quy tắc 6:** Khảo sát 1 chương trình để xem liệu chương trình có thực hiện cái mà nó cần thực hiện chỉ là 1 phần, phần còn lại là xem liệu chương trình có thực hiện cái mà nó không cần phải thực hiện hay không.

**Quy tắc 7:** Tránh các ca kiểm thử bâng quơ trừ khi chương trình thực sự là 1 chương trình bâng quơ.

**Quy tắc 8:** Không dự kiến kết quả của kiểm thử theo giả thiết ngầm là không tìm thấy lỗi.

**Quy tắc 9:** Xác suất tồn tại lỗi trong 1 đoạn chương trình là tương ứng với số lỗi đã tìm thấy trong đoạn đó.

**Quy tắc 10:** Kiểm thử là 1 nhiệm vụ cực kỳ sáng tạo và có tính thử thách trí tuệ.

**CHƯƠNG 2. THIẾT KẾ TEST PLAN**

## 2.1. Tổng quang.

**2.1.1. Khái niệm Test Plan**

Test Plan là một tài liệu chi tiết phác thảo chiến lược kiểm thử, Mục tiêu kiểm thử, tài nguyên (nhân lực, phần mềm, phần cứng) cần thiết để kiểm thử, schedule kiểm thử, Dự toán kiểm thử và deliver. Test Plan đóng vai trò là một kế hoạch chi tiết để tiến hành các hoạt động kiểm thử phần mềm như một quy trình xác định, được giám sát và kiểm soát từng bước bởi Test Manager.

**2.1.2. Phân tích sản phẩm**

Phần mềm Quản lý Khách Sạn tập trung và các nghiệp vụ lưu trữ, quản lý, tính toán những thông tin liên quan tới phòng, khách hàng và các dịch vụ đặt phòng. Khi đó sẽ giúp người quản trị kiểm soát được số lượng cũng như chất lượng hoạt động của khách sạn.

* + 1. **Các phần mềm sử dụng**

Visual Studio 2019, Microsoft SQL 2018, Devexpress v21.2, Github…

## 2.2. Xây dựng chiến lược kiểm thử.

Chiến lược kiểm thử là một bước quan trọng trong việc lập một test plan. Tài liệu chiến lược kiểm thử, là tài liệu cấp cao, thường được phát triển bởi Test Manager. Tài liệu này định nghĩa:

* Mục tiêu kiểm thử của dự án và các phương tiện để đạt được chúng
* Xác định nỗ lực và chi phí kiểm thử
  + 1. **Phạm vi kiểm thử**
* Kiểm thử trong phạm vi: Theo thông số kỹ thuật yêu cầu phần mềm, phần mềm Quản lý Khách Sạn chỉ tập trung vào kiểm thử tất cả các chức và giao diện bên ngoài của phần mềm.
* Kiểm thử ngoài phạm vi: Kiểm thử không chức năng như stress testing, performance testing hoặc logical database testing hiện sẽ không được kiểm thử.
  + 1. **Xác định loại kiểm thử.**

|  |  |
| --- | --- |
| Unit Test | Kiểm thử phần mềm đơn vị (đặt phòng) |

Bảng 2.1 Xác định loại kiểm thử

* + 1. **Tài liệu rủi ro và vấn đề.**

|  |  |
| --- | --- |
| **Rủi ro** | **Giải pháp** |
| Thành viên trong nhóm thiếu các kỹ năng cần thiết để kiểm thử phần mềm | Lập kế hoạch đào tạo để nâng cao trình độ thành viên của nhóm bạn |
| Thành viên trong nhóm thiếu các kỹ năng cần thiết để kiểm thử phần mềm | Đặt ưu tiên kiểm thử cho từng hoạt động kiểm thử |
| Quản lý kiểm thử có kỹ năng quản lý kém | Kế hoạch đào tạo lãnh đạo cho người quản lý |
| Sự thiếu hợp tác ảnh hưởng tiêu cực đến năng suất của nhân viên của bạn | Khuyến khích mỗi thành viên trong nhóm thực hiện nhiệm vụ của mình và truyền cảm hứng cho họ để nỗ lực nhiều hơn |
| Dự toán ngân sách sai và vượt chi phí | Thiết lập phạm vi trước khi bắt đầu công việc, chú ý nhiều đến việc lập kế hoạch dự án và liên tục theo dõi và đo lường tiến độ |

Bảng 2.2. Tài liệu rủi ro và vấn đề

* + 1. **Test Logistics**

|  |  |
| --- | --- |
| **Người thực hiên** | **Trình tự công việc** |
| Trưởng dự án | Lập kế hoạch test |
| Đội dự án | Cài đặt, chuẩn bị test |
| Đội dự án | Unit test (test đơn vị) |
| Đội dự án, Tester | Test module |
| Đội Test | Integeration Test (Test tích hợp) |
| Đội Test | System Test (Test hệ thống) |
| Trưởng dự án | Xem xét, đánh giá kết quả Test |
| Trưởng dự án  Ban giám đốc | Tổng hợp báo cáo |

Bảng 2.3. Test Logistics

* + 1. **Xác định tiêu chí kiểm thử**

|  |  |
| --- | --- |
| **Test Criteria** | **Tiêu chuẩn** |
| Entrance criteria (Tiêu chuẩn bắt đầu) | * 1. Tất cả các mã nguồn đã được thực hiện kiểm tra đơn vị (uint test).   2. Tất cả nhân viên bảo đảm chất lượng có đủ thông tin về chức năng.   3. Hard ware và soft ware đã sẵn sàng.   4. Testplan và testcase đã được renew và duyệt. |
| Exit criteria (Tiêu chuẩn kết thúc) | 1. Không có defect/bug quá thời gian hoặc nổ lực kiểm thử. 2. Kế hoach bàn giao sẵn sàng. 3. Các lỗi nghiêm trọng đã được sửa. |

Bảng 2.4. Tiêu chí kiểm thử

# CHƯƠNG 3: THIẾT KẾ TEST – CASE

## Khái niệm

Thiết kế test – case trong kiểm thử phần mềm là quá trình xây dựng các phương pháp kiểm thử có thể phát hiện lỗi, sai sót, khuyết điểm của phần mềm để xây dựng phần mềm đạt tiêu chuẩn.

## 3.2. Vai trò của thiết kế test – case

* + - Tạo ra các ca kiểm thử tốt nhất có khả năng phát hiện ra lỗi, sai sót của phần mềm một cách nhiều nhất.
    - Tạo ra các ca kiểm thử có chi phí rẻ nhất, đồng thời tốn ít thời gian và công sức nhất.

## Quy trình của thiết kế test – case

Một trong những lý do quan trọng nhất trong kiểm thử chương trình là thiết kế và tạo ra các ca kiểm thử - các Test case có hiệu quả.

Về mặt khả năng tìm ra nhiều lỗi nhất, tập hợp các ca kiểm thử được chọn ngẫu nhiên có rất ít cơ hội là tập hợp tối ưu hay gần tối ưu. Sau đây là một số phương pháp để chọn ra một tập dữ liệu kiểm thử một cách thông minh.

Để kiểm thử hộp đen và kiểm thử hộp trắng một cách thấu đáo là không thể. Do đó, một chiến lược kiểm thử hợp lý là chiến lược có thể kết hợp sức mạnh của cả hai phương pháp trên

Những chiến lược kết hợp đó bao gồm:

|  |  |
| --- | --- |
| **Hộp đen** | **Hộp trắng** |
| * Phân lớp tương đương | * Bao phủ câu lệnh |
| * Phân tích giá trị biên | * Bao phủ quyết định |
| * Đồ thị nguyên nhân kết quả | * Bao phủ điều kiện |
| * Đoán lỗi | * Bao phủ điều kiện - quyết định |
|  | * Bao phủ đa điều kiện |

*Bảng 3. 1 Phương pháp thiết kế ca kiểm thử*

Mỗi phương pháp có những ưu điểm cũng như khuyết điểm riêng, do đó để có được tập các ca kiểm thử tối ưu, chúng ta cần kết hợp hầu hết các phương pháp. Kiểm thử hộp đen

#### Phân lớp tương đương – Equivalence Patitioning

Phân lớp tương đương là một phương pháp kiểm thử hộp đen chia miền đầu vào của một chương trình thành các lớp dữ liệu, từ đó suy dẫn ra các ca kiểm thử. Phương pháp này cố gắng xác định ra một ca kiểm thử mà làm lộ ra một lớp lỗi, do đó làm giảm tổng số các trường hợp kiểm thử phải được xây dựng.

Thiết kế ca kiểm thử cho phân lớp tương đương dựa trên sự đánh giá về các lớp tương đương với một điều kiện vào. Lớp tương đương biểu thị cho tập các trạng thái hợp lệ hay không hợp lệ đối với điều kiện vào.

Một cách xác định tập con này là để nhận ra rằng 1 ca kiểm thử được lựa chọn tốt cũng nên có 2 đặc tính khác:

Giảm thiểu số lượng các ca kiểm thử khác mà phải được phát triển để hoàn thành mục tiêu đã định của kiểm thử “hợp lý”.

Bao phủ một tập rất lớn các ca kiểm thử có thể khác. Tức là, nó nói cho chúng ta một thứ gì đó về sự có mặt hay vắng mặt của những lỗi qua tập giá trị đầu vào cụ thể.

Thiết kế Test-case bằng phân lớp tương đương tiến hành theo 2 bước:

(1). Xác định các lớp tương đương

(2). Xác định các ca kiểm thử.

#### Xác định các lớp tương đương:

Các lớp tương đương được xác định bằng cách lấy mỗi trạng thái đầu vào và phân chia nó thành 2 hay nhiều nhóm.

|  |  |  |
| --- | --- | --- |
| **Điều kiện bên ngoài** | **Các lớp tương đương hợp lệ** | **Các lớp tương đương không hợp lệ** |
|  |  |  |

*Bảng 3. 2 Mẫu liệt kê các lớp tương đương*

Chú ý là hai kiểu lớp tương đương được xác định: lớp tương đương hợp lệ mô tả các đầu vào hợp lệ của chương trình, và lớp tương đương không hợp lệ mô tả tất cả các trạng thái có thể khác của điều kiện. Để xác định các lớp tương đương có thể áp dụng tập các nguyên tắc dưới đây:

* Nếu 1 trạng thái đầu vào định rõ giới hạn của các giá trị, xác định 1 lớp tương đương hợp lệ và 2 lớp tương đương không hợp lệ.
* Nếu 1 trạng thái đầu vào xác định số giá trị, xác định 1 lớp tương đương hợp lệ và 2 lớp tương đương bất hợp lệ.
* Nếu 1 trạng thái đầu vào chỉ định tập các giá trị đầu vào và chương trình sử dụng mỗi giá trị là khác nhau, xác định 1 lớp tương đương hợp lệ cho mỗi loại và 1 lớp tương đương không hợp lệ.
* Nếu 1 trạng thái đầu vào chỉ định một tình huống “chắc chắn – must be”, xác định 1 lớp tương đương hợp lệ và 1 lớp tương đương không hợp lệ.

Nếu có bất kỳ lý do nào để tin rằng chương trình không xử lý các phần tử trong cùng 1 lớp là như nhau, thì hãy chia lớp tương đương đó thành các lớp tương đương nhỏ hơn.

#### Xác định các ca kiểm thử:

Với các lớp tương đương xác định được ở bước trên, bước thứ hai là sử dụng các lớp tương đương đó để xác định các ca kiểm thử. Quá trình này như sau:

* Gán 1 số duy nhất cho mỗi lớp tương đương.
* Cho đến khi tất cả các lớp tương đương hợp lệ được bao phủ bởi (hợp nhất thành) các ca kiểm thử, viết 1 ca kiểm thử mới bao phủ càng nhiều các lớp tương đương đó chưa được bao phủ càng tốt.
* Cho đến khi các ca kiểm thử của bạn đã bao phủ tất cả các lớp tương đương không hợp lệ, viết 1 ca kiểm thử mà bao phủ một và chỉ một trong các lớp tương đương không hợp lệ chưa được bao phủ.
* Lý do mà mỗi ca kiểm thử riêng bao phủ các trường hợp không hợp lệ là vì các kiểm tra đầu vào không đúng nào đó che giấu hoặc thay thế các kiểm tra đầu vào không đúng khác.

#### Phân tích giá trị biên – Boundary Value Analysis

Kinh nghiệm cho thấy các ca kiểm thử mà khảo sát tỷ mỷ các điều kiện biên có tỷ lệ phần trăm cao hơn các ca kiểm thử khác. Các điều kiện biên là những điều kiện mà các tình huống ngay tại, trên và dưới các cạnh của các lớp tương đương đầu vào và các lớp tương đương đầu ra. Phân tích các giá trị biên là phương pháp thiết kế ca kiểm thử bổ sung thêm cho phân lớp tương đương, nhưng khác với phân lớp tương đương ở 2 khía cạnh:

* Phân tích giá trị biên không lựa chọn phần tử bất kỳ nào trong 1 lớp tương đương là điển hình, mà nó yêu cầu là 1 hay nhiều phần tử được lựa chọn như vậy mà mỗi cạnh của lớp tương đương đó chính là đối tượng kiểm tra.
* Ngoài việc chỉ tập trung chú ý vào các trạng thái đầu vào, các ca kiểm thử cũng nhận được bằng việc xem xét không gian kết quả.

Phân tích giá trị biên yêu cầu óc sáng tạo và lượng chuyên môn hóa nhất định và nó là một quá trình mang tính kinh nghiệm rất cao. Tuy nhiên, có một số quy tắc chung như sau:

* Nếu 1 trạng thái đầu vào định rõ giới hạn của các giá trị, hãy viết các ca kiểm thử cho các giá trị cuối của giới hạn, và các ca kiểm thử đầu vào không hợp lệ cho các trường hợp vừa ra ngoài phạm vi.
* Nếu 1 trạng thái đầu vào định rõ số lượng giá trị, hãy viết các ca kiểm thử cho con số lớn nhất và nhỏ nhất của các giá trị và một giá trị trên, một giá trị dưới những giá trị này
* Sử dụng quy tắc 1 cho mỗi trạng thái đầu vào. Việc xem xét giới hạn của không gian kết quả là quan trọng vì không phải lúc nào các biên của miền đầu vào cũng mô tả cùng một tập sự kiện như biên của giới hạn đầu raSử dụng nguyên tắc 2 cho mỗi trạng thái đầu ra.
* Nếu đầu vào hay đầu ra của 1 chương trình là tập được sắp thứ tự, tập trung chú ý vào các phần tử đầu tiên và cuối cùng của tập hợp.
* Sử dụng sự khéo léo của bạn để tìm các điều kiện biên.

#### Sử dụng đồ thị nguyên nhân - kết quả - Cause & Effect Graphing

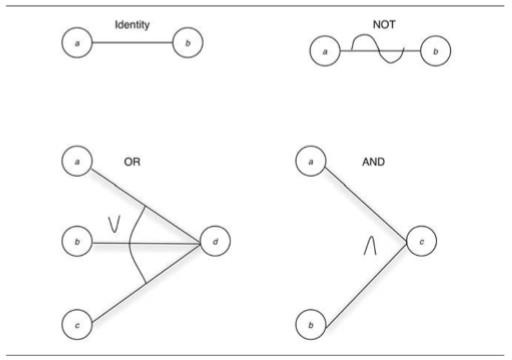
Một yếu điểm của phân tích giá trị biên và phân lớp tương đương là chúng không khảo sát sự kết hợp của các trường hợp đầu vào. Việc kiểm tra sự kết hợp đầu vào không phải là một nhiệm vụ đơn giản bởi vì nếu bạn phân lớp tương đương các trạng thái đầu vào, thì số lượng sự kết hợp thường là rất lớn.

Đồ thị nguyên nhân – kết quả hỗ trợ trong việc lựa chọn một cách có hệ thống tập các ca kiểm thử có hiệu quả cao.

Quá trình dưới đây được sử dụng để xây dựng được các test – case:

* Đặc tả được chia thành các phần có thể thực hiện được. Điều này là cần thiết bởi vì đồ thị nguyên nhân – kết quả trở nên khó sử dụng khi được sử dụng trên những đặc tả lớn.
* Nguyên nhân và kết quả trong các đặc tả được nhận biết. Một nguyên nhân là một trạng thái đầu vào nhất định hay một lớp tương đương của các trạng thái đầu vào. Một kết quả là một trạng thái đầu ra hay 1 sự biến đổi hệ thống.
* Xây dựng đồ thị nguyên nhân – kết quả bằng cách phát triển và biến đổi nội dung ngữ nghĩa của đặc tả thành đồ thị Boolean nối giữa nguyên nhân và kết quả.
* Đồ thị được diễn giải với các ràng buộc mô tả những sự kết hợp của nguyên nhân và/hoặc kết quả là không thể vì các ràng buộc ngữ nghĩa và môi trường.
* Bằng việc dò theo các điều kiện trạng thái trong đồ thị một cách cẩn thận, bạn chuyển đổi đồ thị thành một bảng quyết định mục vào giới hạn. Mỗi cột trong bảng mô tả một ca kiểm thử.

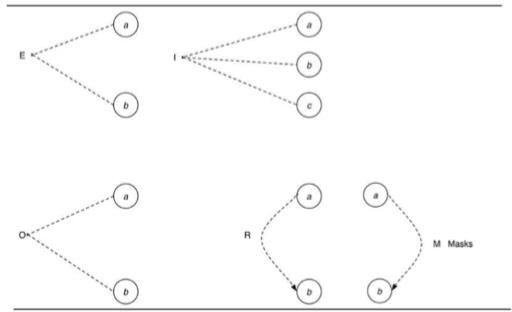
Các cột trong bảng quyết định được chuyển thành các ca kiểm thử.

Ký hiệu cơ bản cho đồ thị được chỉ ra trong hình 2.1. Tưởng tượng mỗi nút có giá trị là 0 hoặc 1; 0 mô tả trạng thái vắng mặt và 1 mô tả trạng thái có mặt.

*Hình 2. 1 Các ký hiệu đồ thị nguyên nhân – kết quả cơ bản*

Trong hầu hết các chương trình, sự kết hợp nào đó của một số nguyên nhân là không thể bởi vì lý do ngữ nghĩa và môi trường (Ví dụ: một ký tự không thể đồng thời vừa là “A” vừa là “B”). Khi đó, ta sử dụng ký hiệu trong Hình 2.2. Ràng buộc E (Exclude

– loại trừ) khẳng định rằng tối đa, chỉ có hoặc a hoặc b có thể là 1 (a và b không thể đồng thời là 1). Ràng buộc I (Include – bao hàm) khẳng định ít nhất một trong a, b hoặc c phải luôn luôn là 1 (a, b hoặc c không thể đồng thời là 0). Ràng buộc O (Only – chỉ một) khẳng định một và chỉ một hoặc a hoặc b phải là 1. Ràng buộc R (Request – yêu cầu) khẳng định rằng khi a là 1, thì b phải là 1 (Ví dụ: không thể có trường hợp a là 1, còn b là 0). Ràng buộc M (Mask – mặt nạ) khẳng định là nếu kết quả a là 1, kết quả b sẽ bắt buộc phải là 0.



*Hình 3. 2 Các ký hiệu ràng buộc*

Bước tiếp theo là tạo bảng quyết định mục vào giới hạn – limited-entry decision table. Tương tự với các bảng quyết định, thì nguyên nhân chính là các điều kiện và kết quả chính là các hành động. Quy trình được sử dụng là như sau:

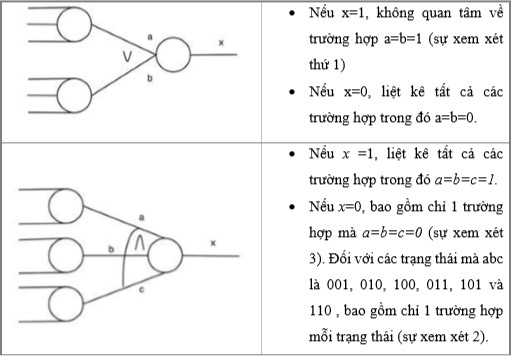
1. Chọn một kết quả để là trạng thái có mặt (1).
2. Lần ngược trở lại đồ thị, tìm tất cả những sự kết hợp của các nguyên nhân (đối tượng cho các rằng buộc) mà sẽ thiết lập kết quả này thành 1.
3. Tạo một cột trong bảng quyết định cho mỗi sự kết hợp nguyên nhân.
4. Với mỗi sự kết hợp, hãy quy định trạng thái của tất cả các kết quả khác và đặt chúng vào mỗi cột.

Trong khi biểu diễn bước 2, cần quan tâm các vấn đề sau:

1. Khi lần ngược trở lại qua một nút or mà đầu ra của nó là 1, không bao giờ thiết lập nhiều hơn 1 đầu vào cho nút or là 1 một cách đồng thời. Điều này được gọi là path sensitizing – làm nhạy đường đi. Mục tiêu của nó là để ngăn chặn dò lỗi thất bại vì một nguyên nhân che đi một nguyên nhân khác.
2. Khi lần ngược trở lại qua một nút and mà đầu ra của nó là 0, dĩ nhiên, phải liệt kê tất cả các sự kết hợp đầu vào dẫn tới đầu ra 0. Tuy nhiên, nếu bạn

đang khảo sát trạng thái mà 1 đầu ra là 0 và một hay nhiều đầu ra khác là 1, thì không nhất thiết phải liệt kê tất cả các điều kiện mà dưới điều kiện đó các đầu vào khác có thể là 1.

1. Khi lần ngược trở lại qua một nút and mà đầu ra của nó là là 0, chỉ cần liệt kê 1 điều kiện trong đó tất cả đầu vào bằng 0. (Nếu nút and ở chính giữa của đồ thị như vậy thì tất cả các đầu vào của nó xuất phát từ các nút trung gian khác, có thể có quá nhiều trạng thái mà trong trạng thái đó tất cả các đầu vào của nó bằng 0.)



*Hình 3. 3 Những xem xét được sử dụng khi dò theo đồ thị*

Những sự xem xét này có thể xuất hiện thất thường, nhưng chúng có một mục đích rất quan trọng: để giảm bớt các kết quả được kết hợp của đồ thị. Chúng liệt kê các trường hợp mà hướng về các ca kiểm thử ít có lợi.

#### NHẬN XÉT

Vẽ đồ thị nguyên nhân – kết quả là phương pháp tạo các ca kiểm thử có hệ thống mô tả sự kết hợp của các điều kiện. Vẽ đồ thị nguyên nhân – kết quả tạo ra tập các ca kiểm thử hữu dụng, nhưng thông thường nó không tạo ra tất cả các ca kiểm thử hữu dụng mà có thể được nhận biết. Tuy nhiên, vấn đề trong việc thực hiện điều này là nó làm cho đồ thị rất phức tạp và dẫn tới số lượng rất lớn các ca kiểm thử. Vì thế, tốt nhất là xét 1 sự phân tích giá trị giới hạn tách rời nhau.

Khía cạnh khó nhất của kỹ thuật này là quá trình chuyển đổi đồ thị thành bảng quyết định. Quá trình này có tính thuật toán, tức là bạn có thể tự động hóa nó bằng việc viết 1 chương trình.

#### Đoán lỗi – Error Guessing

Một kỹ thuật thiết kế test-case khác là error guessing – đoán lỗi. Tester được đưa cho 1 chương trình đặc biệt, họ phỏng đoán, cả bằng trực giác và kinh nghiệm, các loại lỗi có thể và sau đó viết các ca kiểm thử để đưa ra các lỗi đó.

### Chiến lược

Các phương pháp thiết kế test-case đã được thảo luận có thể được kết hợp thành một chiến lược toàn diện. Vì mỗi phương pháp có thể đóng góp 1 tập riêng các ca kiểm thử hữu dụng, nhưng không cái nào trong số chúng tự nó đóng góp một tập trọn vẹn các các ca kiểm thử. Chiến lược hợp lý như sau:

1. Nếu đặc tả có chứa sự kết hợp của các điều kiện đầu vào, hãy bắt đầu với việc vẽ đồ thị nguyên nhân – kết quả.
2. Trong trường hợp bất kỳ, sử dụng phương pháp phân tích giá trị biên. Phương pháp phân tích giá trị biên mang lại một tập các điều kiện kiểm tra bổ sung, và rất nhiều hay toàn bộ các điều kiện này có thể được hợp nhất thành các kiểm thử nguyên nhân – kết quả.
3. Xác định các lớp tương đương hợp lệ và không hợp lệ cho đầu vào và đầu ra, và bổ sung các ca kiểm thử được xác định trên nếu cần thiết.
4. Sử dụng kỹ thuật đoán lỗi để thêm các ca kiểm thử thêm vào.
5. Khảo sát tính logic của chương trình liên quan đến tập các ca kiểm thử. Sử dụng tiêu chuẩn bao phủ quyết định, bao phủ điều kiện, bao phủ quyết định/điều kiện, hay bao phủ đa điều kiện

Tuy việc sử dụng chiến lược này sẽ không đảm bảo rằng tất cả các lỗi sẽ được tìm thấy, nhưng nó đã được xác minh là đại diện cho một sự thỏa thuận hợp lý.



# CHƯƠNG 4: ÁP DỤNG

## 4.1 WBS

## 4.2 Đặc tả

Các chức năng chính của phần mềm:

* Nhân viên:
* Đăng nhập.
* Đặt phòng cho khách.
* Xem thông tin đặt phòng.
* Tra cứu phòng, khách hàng.

Nhân viên được Admin (Quản lí) thêm vào hệ thống thì mới đăng nhập. Thông tin đăng nhập bao gồm:

* Tên đăng nhập.
* Mật khẩu.

Nếu đúng tên đăng nhập và mật khẩu vừa đăng ký sẽ đi đến form trang chủ

Những thông tin đăng nhập phải thỏa mãn:

* Tên đăng nhập, mật khẩu không được để trống.

Khi nhân viên vào phần mềm nếu nhân viên chưa có tài khoản thì có thể yêu cầu quản lý (quản trị viên) để them tài khoản cho mình:

Sau khi được quản lý cung cấp cho tài khoản để đăng nhập hệ thống, nhân viên vào form đăng nhập và điền lại thông tin bao gồm: tên đăng nhập và mật khẩu. Nếu đúng tên đăng nhập và mật khẩu vừa đăng ký sẽ đi đến form trang chủ còn nếu sai vui lòng kiểm tra lại thông tin của mình.

* Quản lý:
* Thêm, xóa, sửa phòng.
* Quản lý nhân viên, khách hàng.
* Phân quyền.
* Thống kê doanh thu tháng, năm.

Quản lý đăng nhập vào trang chủ có quyền thêm xóa sửa sản phẩm, tìm kiếm các loại phòng, quản lý nhân viên và khách hàng bao gồm thông tin cá nhân của nhân viên hoặc của các hàng. Thống kê doanh thu của khách sạn theo tháng, quý.

## 4.3 Kiểm tra hộp đen

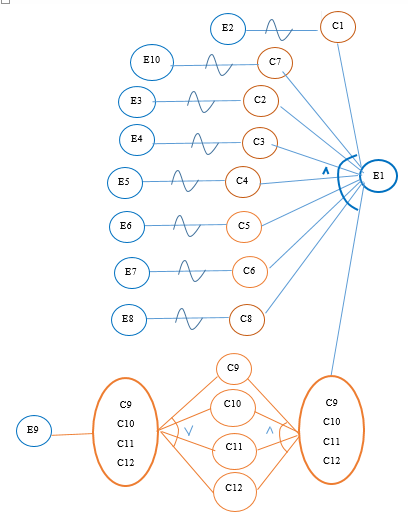
**4.3.1 Vẽ sơ đồ nguyên nhân kết quả**

Nguyên nhân:

* C1: Không để trống tên đăng nhập.
* C2: Không để trống mật khẩu.
* C3: Không được để trống Email.
* C4: Không được để trống họ tên.
* C5: Không được để trống địa chỉ.
* C6: Không được để trống số điện thoại.
* C7: Ngày sinh phải tối thiểu ít nhất 15 tuổi.
* C8: Nhập định dạng email hợp lệ.
* C9: Nhập số điện thoại nhỏ hơn 10 số.
* C10: Nhập số điện thoại lớn hơn 10 số.
* C11: Nhập di động không có số 0 ban đầu.
* C12: Nhập di động khác ký tự số.

Kết quả là:

* E1. Thông báo thêm người dùng thành công.
* E2. Thông báo yêu cầu nhập tên đăng nhập.
* E3. Thông báo yêu cầu nhập mật khẩu
* E4. Thông báo yêu cầu nhập email.
* E5. Thông báo yêu cầu nhập họ tên.
* E6. Thông báo yêu cầu nhập địa chỉ.
* E7. Thông báo yêu cầu nhập số điện thoại
* E8. Thông báo yêu cầu nhập email hợp lệ
* E9. Thông báo yêu cầu nhập số điện thoại hợp lệ
* E10. Thông báo yêu cầu nhập nhập ngày sinh hợp lệ



Hình 4. 2 Đồ thị nguyên nhân kết quả

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Action | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 | T12 | T13 |
| C1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |
| C2 | 1 |  | 1 |  |  |  |  |  |  |  |  |  |  |
| C3 | 1 |  |  | 1 |  |  |  |  |  |  |  |  |  |
| C4 | 1 |  |  |  | 1 |  |  |  |  |  |  |  |  |
| C5 | 1 |  |  |  |  | 1 |  |  |  |  |  |  |  |
| C6 | 1 |  |  |  |  |  | 1 |  |  |  |  |  |  |
| C7 | 1 |  |  |  |  |  |  |  |  |  |  |  | 1 |
| C8 | 1 |  |  |  |  |  |  | 1 |  |  |  |  |  |
| C9 | 0 |  |  |  |  |  |  |  | 1 |  |  |  |  |
| C10 | 0 |  |  |  |  |  |  |  |  | 1 |  |  |  |
| C11 | 0 |  |  |  |  |  |  |  |  |  | 1 |  |  |
| C12 | 0 |  |  |  |  |  |  |  |  |  |  | 1 |  |
| E1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| E2 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |
| E3 |  |  | 1 |  |  |  |  |  |  |  |  |  |  |
| E4 |  |  |  | 1 |  |  |  |  |  |  |  |  |  |
| E5 |  |  |  |  | 1 |  |  |  |  |  |  |  |  |
| E6 |  |  |  |  |  | 1 |  |  |  |  |  |  |  |
| E7 |  |  |  |  |  |  | 1 |  |  |  |  |  |  |
| E8 |  |  |  |  |  |  |  | 1 |  |  |  |  |  |
| E9 |  |  |  |  |  |  |  |  | 1 | 1 | 1 | 1 |  |
| E10 |  |  |  |  |  |  |  |  |  |  |  |  | 1 |

Bảng 4. 3 Bảng quyết định

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên TestCase** | **Bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Tình trạng** |
| 1 | Kiểm tra thông tin tài khoản nhân vien và nhập đầy đủ các thông tin như: tên đăng nhập, mật khẩu | 1. Form đăng nhập xuất hiện. 2. Nhập thông tin nhân viên:   Tên đăng nhập: “0703051386”, mật khẩu: “1” | Đăng nhập thành công | Đăng nhập thành công | PASS |
| 2 | Kiểm tra thông tin đăng nhập không tồn tại | 1.Form đăng nhập xuất hiện.  2.Nhập thông tin nhân viên:  Tên đăng nhập: “07030513”, mật khẩu: “1” | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | PASS |
| 3 | Kiểm tra thông tin đăng nhập sai mật khẩu | 1. Form đăng nhập xuất hiện. 2. Nhập thông tin nhân viên:   Tên đăng nhập: “0703051386”, mật khẩu: “1dsd” | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | PASS |
| 4 | Kiểm tra thông tin đăng nhập không hợp lệ | 1.Form đăng nhập xuất hiện.  2.Nhập thông tin nhân viên:  Tên đăng nhập: “0703051386aaaaaa”, mật khẩu: “1dsdaaaaaaa” | Tên đăng nhập hoặc Mật khẩu không được vượt quá 10 ký tự. Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không được vượt quá 10 ký tự. Xin vui lòng kiển tra lại! | PASS |
| 5 | Kiểm tra thông tin đăng nhập không hợp lệ (Chứa ký tự đặt biệt) | 1.Form đăng nhập xuất hiện.  2.Nhập thông tin nhân viên:  Tên đăng nhập: “52317@`####”, mật khẩu: “12345” | Tên đăng nhập hoặc Mật khẩu chứa ký tự đặt biệt . Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | FAILED |
| 6 | Kiểm tra thông tin đăng nhập không hợp lệ | 1.Form đăng nhập xuất hiện.  2.Nhập thông tin nhân viên:  Tên đăng nhập: “”, mật khẩu: “” | Tên đăng nhập hoặc Mật khẩu không được bỏ trống. Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không được bỏ trống. Xin vui lòng kiển tra lại! | PASS |

#### Phân lớp tương đương

* + - * + ***Xác định các lớp tương đương***

**Nguyên tắc để xác định lớp tương đương:**

**NT1**: Điều kiện đầu vào là một giá trị xác định thì chia vùng tương đương thành 3 tình huống:

Xác định 1 lớp tương đương hợp lệ.

Xác định 2 lớp tương đương không hợp lệ.

**Ví dụ:** Khi nhập thông tin tên đăng nhập phải không được bỏ trống và nhỏ hơn 10 ký tự số vậy sẽ có 3 lớp tương đương l2:

Lớp tương đương hợp lệ: Không bỏ trống tên đăng nhập và không quá 10 ký tự.

Lớp tương đương không hợp lệ: Bỏ trống tên đăng nhập.

Lớp tương đương không hợp lệ: Nhập quá 10 ký tự.

**NT2**: Nếu điều kiện đầu vào xác định là một kiểu đúng sai thì chia vùng tương đương thành 2 tình huống:

* Xác định 1 lớp tương đương hợp lệ
* Xác định 1 lớp tương đương không hợp lệ

`**Ví dụ**: Khi nhập tên đăng nhập không chứa ký tự đặt biệt.

* 1 lớp tương đương hợp lệ: nhập không chứa ký tự đặt biệt.
* 1 lớp tương đương không hợp lệ: nhập chứa ký tự đặc biệt.

#### Phân tích giá trị biên

Phân tích giá trị biên bao gồm:

* + - * + Giá trị nhỏ nhất - 1.
        + Giá trị nhỏ nhất.
        + Giá trị nhỏ nhất + 1.
        + Giá trị bình thường.
        + Giá trị lớn nhất.
        + Giá trị lớn nhất - 1.
        + Giá trị lớn nhất.
        + Giá trị lớn nhất + 1

**Ví dụ:** Khi nhập thông tin tên đăng nhập không quá 10 ký tự.

|  |  |  |
| --- | --- | --- |
| **STT** | **Mô tả kịch bản thử nghiệm** | **Kết quả mong muốn** |
| 1 | Nhập tên đăng nhập không chứa ký tự đặc biệt và không quá 10 ký tự | Hệ thống chấp nhận |
| 2 | Nhập chứa ký tự đặc biệt | Hệ thống không chấp nhận |
| 3 | Nhập 11 ký tự | Hệ thống không chấp nhận |
| 4 | Để trống | Hệ thống không chấp nhận |

***Bảng 4. 2 Test case phân tích giá rị biên kiểm tra tên đăng nhập***

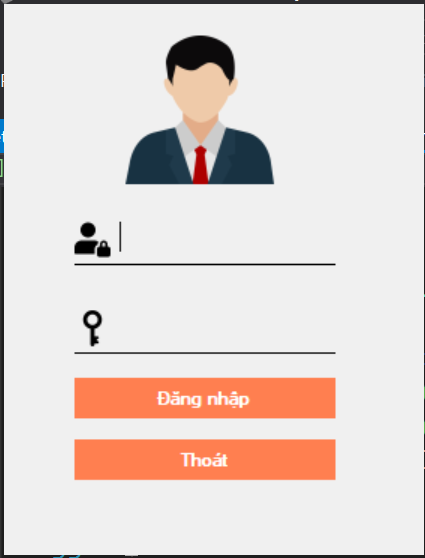
## 4.4 Kiểm thử chức năng

|  |  |
| --- | --- |
| **TEST PLAN** | |
| Tên project: | Quản lý Khách Sạn |
| Ngày bắt đầu: | 01/05/2021 |
| Ngày kết thúc: | 25/05/2021 |
| Danh sách mô đun cần test: | Đăng nhập, Danh sách khách hàng, Thông tin thuê phòng, Thanh toán, Thông tin nhân viên, Chỉnh sửa phòng |
| Danh sách test case | Đăng nhập: Testcase chức năng đăng nhập, testcase kiểm tra thông tin hợp lệ.  Danh sách khách hàng: Testcase chức năng thêm, testcase chức năng xóa, testcase chức năng sửa.  Thông tin nhân viên: Testcase chức năng thêm, testcase chức năng xóa, testcase chức năng sửa.  Thông tin thuê phòng: Testcase hiển thị danh sách, Testcase chức năng thêm, testcase chức năng xóa, testcase chức năng sửa.  Thanh toán: Testcase hiển thị danh sách, Testcase tiềm kiếm, testcase thanh toán.  Chỉnh sửa phòng: Testcase chỉnh sửa thông tin phòng |

***Bảng 4. 3 Test plan***

### 4.4.1 Chức năng đăng nhập

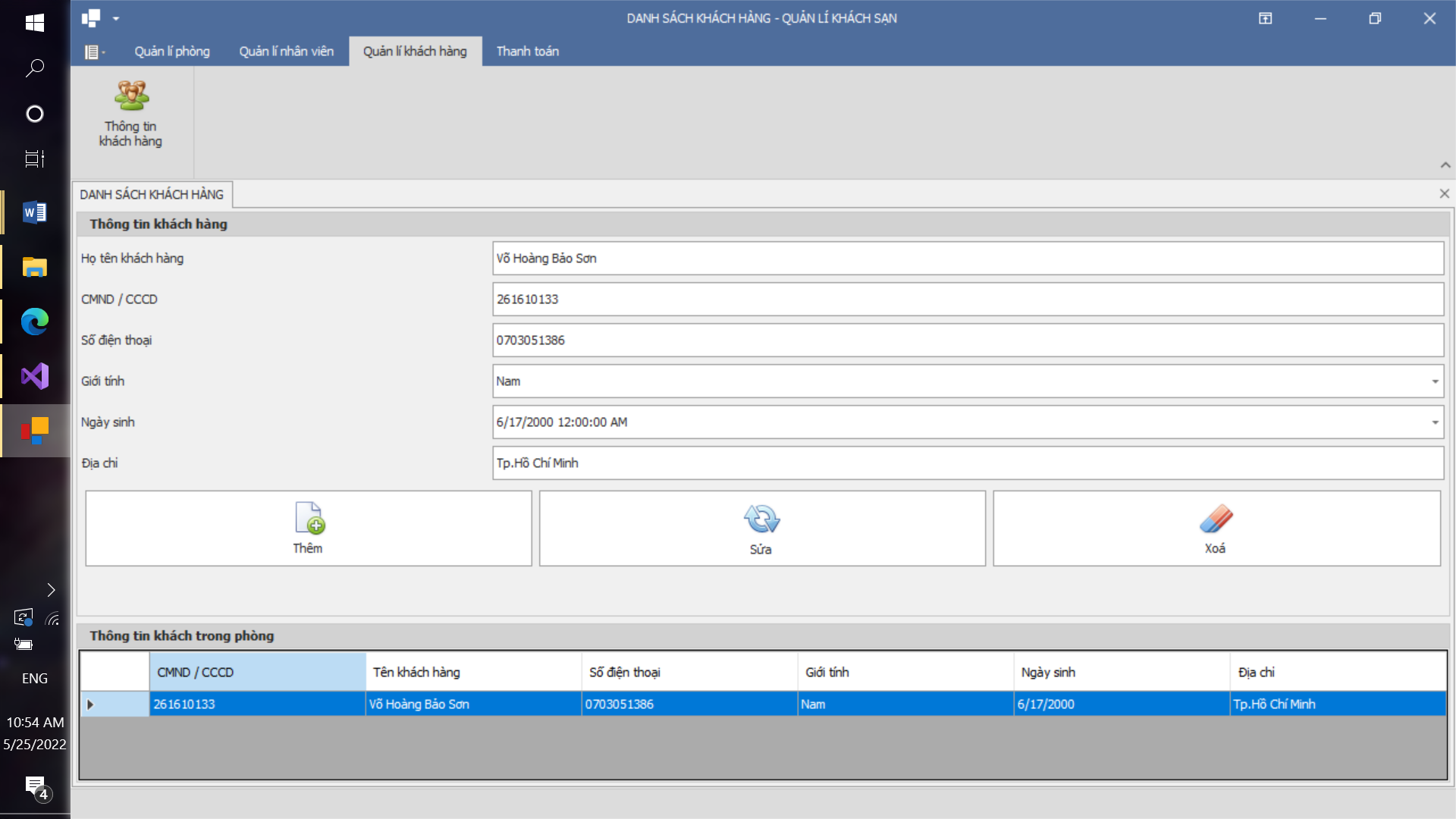
* Kiểm tra màn hình đăng nhập



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **TITLE** | **STEP BY STEP** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| ***TESTCASE CHỨC NĂNG ĐĂNG NHẬP*** | | | | | |
| 1 | Kiểm tra thông tin tài khoản nhân vien và nhập đầy đủ các thông tin như: tên đăng nhập, mật khẩu | 1. Form đăng nhập xuất hiện. 2. Nhập thông tin nhân viên:   Tên đăng nhập: “0703051386”, mật khẩu: “1” | Đăng nhập thành công | Đăng nhập thành công | PASS |
| 2 | Kiểm tra thông tin đăng nhập không tồn tại | 1.Form đăng nhập xuất hiện.  2.Nhập thông tin nhân viên:  Tên đăng nhập: “07030513”, mật khẩu: “1” | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | PASS |
| 3 | Kiểm tra thông tin đăng nhập sai mật khẩu | 1. Form đăng nhập xuất hiện. 2. Nhập thông tin nhân viên:   Tên đăng nhập: “0703051386”, mật khẩu: “1dsd” | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | PASS |
| ***TESTCASE CHỨC NĂNG KIỂM TRA THÔNG TIN HỢP LỆ*** | | | | | |
| 4 | Kiểm tra thông tin đăng nhập không hợp lệ | 1.Form đăng nhập xuất hiện.  2.Nhập thông tin nhân viên:  Tên đăng nhập: “0703051386aaaaaa”, mật khẩu: “1dsdaaaaaaa” | Tên đăng nhập hoặc Mật khẩu không được vượt quá 10 ký tự. Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không được vượt quá 10 ký tự. Xin vui lòng kiển tra lại! | PASS |
| 5 | Kiểm tra thông tin đăng nhập không hợp lệ (Chứa ký tự đặt biệt) | 1.Form đăng nhập xuất hiện.  2.Nhập thông tin nhân viên:  Tên đăng nhập: “52317@`##”, mật khẩu: “12345” | Tên đăng nhập hoặc Mật khẩu chứa ký tự đặt biệt . Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không đúng. Xin vui lòng kiển tra lại! | FAILED |
| 6 | Kiểm tra thông tin đăng nhập không hợp lệ | 1.Form đăng nhập xuất hiện.  2.Nhập thông tin nhân viên:  Tên đăng nhập: “”, mật khẩu: “” | Tên đăng nhập hoặc Mật khẩu không được bỏ trống. Xin vui lòng kiển tra lại! | Tên đăng nhập hoặc Mật khẩu không được bỏ trống. Xin vui lòng kiển tra lại! | PASS |

### 4.4.2. Chức năng Quản lý khách hàng

* Kiểm tra màn hình Khách hàng



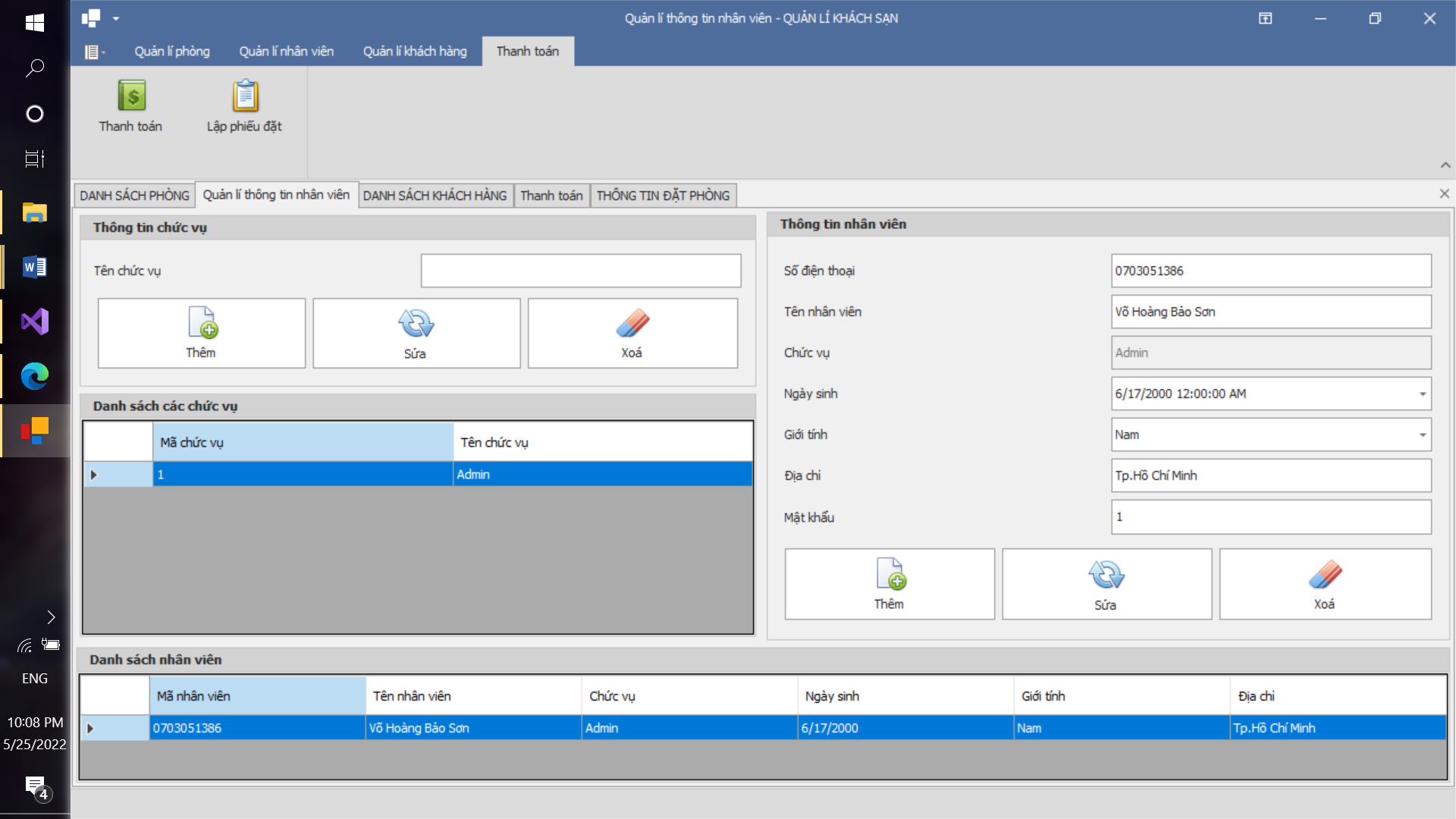
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **TITLE** | **STEP BY STEP** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| ***TESTCASE CHỨC NĂNG THÊM*** | | | | | |
| 1 | Kiểm tra hiển thị chi tiết Khách Hàng | 1. Truy cập vào Tab “Quản lý khách hàng” 2. Nhấn chọn nhân viên trên bảng danh sách Khách Hàng | Các dữ liệu hiển thị trên Textbox, Checkbox, DateTime Picker. | Các dữ liệu hiển thị trên Textbox, Checkbox, DateTime Picker. | PASS |
| 2 | Kiểm tra CMND đã tồn tại | 1. Truy cập vào Tab “Quản lý khách hàng” 2. Nhập CMND “261610133” | Hiển thị thông báo “Mã CMND/CCCD đã được khai báo cho User” | Hiển thị thông báo “Mã CMND/CCCD đã được khai báo cho User” | PASS |
| 3 | Kiểm tra số điện thoại nhập ít hơn 10 ký tự | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Nhập SĐT “09651981” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | PASS |
| 4 | Kiểm tra SĐT để trống | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Để trống textbox SĐT | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | PASS |
| 5 | Kiểm tra ngày sinh để trống | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Để trống textbox ngày sinh, điền đầy đủ các textbox còn lại. | Hiển thị thông báo “Ngày sinh không được bỏ trống. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Ngày sinh không được bỏ trống. Xin vui lòng kiểm tra lại!” | PASS |
| 6 | Kiểm tra khi nhập thông tin người dùng và để trống tên khách hàng | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Để trống textbox họ tên, điền đầy đủ các textbox còn lại. | Hiển thị thông báo “Tên khách hàng không được bỏ trống và không được vượt quá 50 ký tự” | Hiển thị thông báo “Tên khách hàng không được bỏ trống và không được vượt quá 50 ký tự” | PASS |
| 7 | Kiểm tra khi nhập thông tin người dùng và địa chỉ vượt quá 50 ký tự | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Nhập địa chỉ :  “925 Trần Hưng Đạo, khu phố 5, tổ 26, phường Đống Đa, thành phố Quy Nhơn, tỉnh Bình Định.” | Hiển thị thông báo “Địa chỉ khách hàng không được vượt quá 50 ký tự. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Địa chỉ khách hàng không được vượt quá 50 ký tự. Xin vui lòng kiểm tra lại!” | PASS |
| 8 | Kiểm tra khi nhập thông tin người dùng và tên người dùng chứa ký tự đặc biệt | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Nhập tên khách hàng:  “T@ %&\*uan H99\*” | Hiển thị thông báo “Họ tên khách hàng không hợp lệ” | Hiển thị thông báo “Cập nhật dữ liệu khách hàng thành công!” | FAILED |
| 9 | Kiểm tra khi nhập thông tin người dùng đúng | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Nhập tên khách hàng:  “Trần Nhật Hạ Vy”, số điện thoại “0909090909”, CMND “219976456”, ngày sinh “12/10/2000”, giới tính “Nữ”, địa chỉ “Hà Nội” | Hiển thị thông báo “Cập nhật dữ liệu khách hàng thành công!” | Hiển thị thông báo “Cập nhật dữ liệu khách hàng thành công!” | FAILED |
| 10 | Kiểm tra ngày sinh không hợp lệ | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Nhập tên khách hàng: ngày sinh “12/10/2022”, | Hiển thị thông báo “Ngày sinh không hợp lệ” | Hiển thị thông báo “Cập nhật dữ liệu khách hàng thành công!” | FAILED |
| 11 | Kiểm tra CMND nhập chữ | 1. Truy cập vào Tab “Quản lý khách hàng”  2. Nhập CMND  “222**$**kfkaaf” | Hiển thị thông báo “Số CMND không hợp lệ” | Hiển thị thông báo “Cập nhật dữ liệu khách hàng thành công!” | FAILED |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TESTCASE CHỨC NĂNG SỬA*** | | | | | |
| 1 | Kiểm tra hiển thị thông tin khách hàng | 1.Truy cập vào Tab “Quản lý khách hàng”  Nhấn chọn nhân viên trên bảng danh sách Khách Hàng | Hiển thị màn hình sửa với các dữ liệu được load lên các control tương ứng | Hiển thị màn hình sửa với các dữ liệu được load lên các control tương ứng | PASS |
| 2 | Kiểm tra khi thông tin sửa không thay đổi | * Nhấn chọn vào mục hệ thống. * Giữ nguyên dữ liệu ban đầu * Click button Lưu lại | Lưu dữ liệu xuống DB | Lưu dữ liệu xuống DB | PASS |
| 3 | Kiểm tra sửa định dạng SĐT | 1. Chọn vào mục hệ thống 2. Nhấn chọn danh sách khách hàng 3. Chỉnh sửa SĐT: “09786” 4. Nhập đầy đủ các thông tin còn lại hợp lệ 5. Click button “Sửa” | Hiển thị thông báo: “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo: “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | PASS |
| 4 | Kiểm tra sửa không nhập họ tên | 1. Chọn vào mục hệ thống 2. Chọn vào danh sách người dùng 3. Không nhập họ tên 4. Nhập đầy đủ các thông tin còn   lại hợp lệ  4.Click button Lưu | Hiển thị thông báo “Tên khách hàng không được bỏ trống và không được vượt quá 50 ký tự” | Hiển thị thông báo “Tên khách hàng không được bỏ trống và không được vượt quá 50 ký tự” | PASS |
| 5 | Kiểm tra sửa nhập họ tên quá 50 ký tự | 1. Chọn vào mục hệ thống. 2. Chọn vào danh sách người dùng 3. Nhập họ tên 51 ký tự: “Trần Quốc Toanrnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn” 4. Nhập đầy đủ các thông tin còn lại hợp lệ 5. Click button Lưu | Hiển thị thông báo “Tên khách hàng không được bỏ trống và không được vượt quá 50 ký tự” | Hiển thị thông báo “Tên khách hàng không được bỏ trống và không được vượt quá 50 ký tự” | PASS |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | Kiểm tra sửa không nhập địa chỉ | 1. Chọn vào mục hệ thống 2. Chọn vào danh sách người dùng 3. Không nhập họ tên 4. Nhập đầy đủ các thông tin còn   lại hợp lệ   1. Click button Lưu | Hiển thị thông báo “Địa chỉ khách hàng không được vượt quá 50 ký tự. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Địa chỉ khách hàng không được vượt quá 50 ký tự. Xin vui lòng kiểm tra lại!” | PASS |
| 7 | Kiểm tra sửa nhập địa chỉ quá 50 ký tự | 1. Chọn vào mục hệ thống. 2. Chọn vào danh sách người dùng 3. Nhập địa chỉ 51 ký tự: “Lê Trọng Tanrnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn” 4. Nhập đầy đủ các thông tin còn lại hợp lệ 5. Click button Lưu | Hiển thị thông báo “Địa chỉ khách hàng không được vượt quá 50 ký tự. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Địa chỉ khách hàng không được vượt quá 50 ký tự. Xin vui lòng kiểm tra lại!” | PASS |
| 8 | Kiểm tra sửa không nhập số điện thoại | 1. Chọn vào mục hệ thống 2. Chọn vào danh sách người dùng 3. Không nhập số điện thoại 4. Nhập đầy đủ các thông tin còn   lại hợp lệ   1. Click button Lưu | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | PASS |
| 9 | Kiểm tra sửa khi nhập số điện thoại 9 số | * Nhấn chọn vào mục hệ thống. * Chọn mục danh sách người dùng * Nhập thông tin người dùng: số điện thoai chứa 9 ký tự số là: “098971127” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | PASS |
| 10 | Kiểm tra sửa khi nhập số điện thoại 11 số | * Nhấn chọn vào mục hệ thống. * Chọn mục danh sách người dùng * Nhập thông tin người dùng: số điện thoai chứa 11 ký tự số là: “09897112789” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | PASS |
| 11 | Kiểm tra sửa khi nhập số điện thoại là chữ | * Nhấn chọn vào mục hệ thống. * Chọn mục danh sách người dùng * Nhập thông tin người dùng:   số điện thoai là: “abc” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Không cho phép nhập chữ | FAILED |
| 12 | Kiểm tra button Lưu lại | * Nhấn chọn vào mục hệ thống. * Chọn mục danh sách người dùng * Nhập thông tin người dùng: họ tên là: “Trần Quốc Toản”, ngày sinh là: “25/02/2000”, giới tính: “Nam”, địa chỉ: “Bình Tân” số điện thoai: “0989711273” | Hiển thị thông báo” Cập nhật dữ liệu khách hàng thành công!” | Hiển thị thông báo” Cập nhật dữ liệu khách hàng thành công!” | PASS |

### 4.4.3. Chức năng Quản lý nhân viên

* Kiểm tra màn hình Nhân viên



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **TITLE** | **STEP BY STEP** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| ***TESTCASE CHỨC NĂNG THÊM (NHÂN VIÊN)*** | | | | | |
| 1 | Kiểm tra hiển thị chi tiết Nhân viên | 1. Truy cập vào Tab “Thông tin nhân viên” 2. Nhấn chọn nhân viên trên bảng danh sách Nhân viên | Các dữ liệu hiển thị trên Textbox, Checkbox, DateTime Picker. | Các dữ liệu hiển thị trên Textbox, Checkbox, DateTime Picker. | PASS |
| 2 | Kiểm tra chỉ hiển thị mật khẩu của User đang đăng nhập | 1. Truy cập vào Tab “Thông tin nhân viên” 2. Nhấn chọn nhân viên trên bảng danh sách Nhân viên | Các dữ liệu hiển thị trên Textbox, Checkbox, DateTime Picker. Phần mật khẩu của những User khác bị ẩn đi | Các dữ liệu hiển thị trên Textbox, Checkbox, DateTime Picker. Phần mật khẩu của những User khác bị ẩn đi | PASS |
| 3 | Kiểm tra số điện thoại nhập ít hơn 10 ký tự | 1. Truy cập vào Tab “Thông tin nhân viên”  2. Nhập SĐT “09651981” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | PASS |
| 4 | Kiểm tra SĐT để trống | 1. Truy cập vào Tab “Thông tin nhân viên”  2. Để trống textbox SĐT | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Số điện thoại không được bỏ trống và không được vượt quá 10 số. Xin vui lòng kiểm tra lại!” | PASS |
| 5 | Kiểm tra SĐT chứa ký tự đặc biệt | 1. Truy cập vào Tab “Thông tin nhân viên”  2. Nhập vào textbox số điện thoại “0998765hhh”  3. Nhấn button “Thêm” | Hiển thị thông báo “Số điện thoại không hợp lệ” | Hiển thị thông báo “Cập nhật dữ liệu cho nhân viên thành công!” | FAILED |
| 6 | Kiểm tra SĐT chứa 11 ký tự | 1. Truy cập vào Tab “Thông tin nhân viên”  2. Nhập vào textbox số điện thoại “09987659999”  3. Nhấn button “Thêm” | Hiển thị thông báo “Số điện thoại nhân viên không được vượt quá 10 ký tự. Xin vui lòng thử lại!” | Hiển thị thông báo “Số điện thoại nhân viên không được vượt quá 50 ký tự. Xin vui lòng thử lại!” | FAILED |
| 7 | Kiểm tra khi nhập thông tin nhân viên, tên chứa ký tự đặc biệt | 1. Truy cập vào Tab “Thông tin nhân viên”  2. Nhập vào textbox Họ Tên nhân viên “Nguyễn Văn A457”  3. Nhấn button “Thêm” | Hiển thị thông báo “Tên nhân viên không hợp lệ” | Hiển thị thông báo “Cập nhật dữ liệu cho nhân viên thành công!” | FAILED |
| ***TESTCASE CHỨC NĂNG THÊM (CHỨC VỤ)*** | | | | | |
| 1 | Kiểm tra khi không nhập tên chức vụ | 1. Truy cập vào Tab “Thông tin nhân viên”  2. Không nhập tên chức vụ | Hiển thị thông báo “Tên chức vụ không được bỏ trống. Xin vui lòng kiểm tra lại!” | Hiển thị thông báo “Tên chức vụ không được bỏ trống. Xin vui lòng kiểm tra lại!” | PASS |

**KẾT LUẬN**

Kiểm thử phần mềm hiện nay vẫn là vấn đề hết sức quan trọng với các tổ chức phát triển phần mềm. Trong khuôn khổ báo cáo đồ án của mình do thời gian và kinh nghiệm còn hạn chế nên có những phần của báo cáo chưa được đào sâu nghiên cứu.

Sau một thời gian thực hiện báo cáo đồ án dưới sự hướng dẫn của cô Phan Thị Ngọc Mai báo cáo của nhóm em đã thực hiện tốt được các mục tiêu đề ra và đạt được những kết quả như sau:

*- Kết quả đạt được:*

Trình bày đầy đủ và chính xác các vấn đề tổng quan về phần mềm, công nghệ phần mềm, lỗi phần mềm, và các vấn đề liên quan đến kiểm thử phần mềm.

* Giới thiệu quy trình kiểm thử phần mềm, làm quen khái niệm TestPlan, TestCase, TestData.
* Thiết kế phần mềm Winform Quản Lý Khách Sạn từ phần mềm Visual Studio 2019.
* Áp dụng các kiến thức đã nghiên cứu thực hiện kiểm thử các chức năng của ứng dụng
* Báo cáo đồ án kết thúc môn là một tài liệu xúc tích tổng hợp được các vấn đề chính của kiểm thử phần mềm, và có thể được coi là một tài liệu hướng dẫn sử dụng ứng dụng ngắn gọn rõ ràng bằng tiếng Việt để tham khảo.

*- Hạn chế:*

Mặc dù đã cố gắng hết sức trong thời gian thực hiện đề tài nhưng với kinh nghiệm còn hạn chế nên báo cáo thực tập tốt nghiệp không tránh khỏi những thiếu sót:

* Chưa rà soát được các TestCase
* Chưa nghiên cứu phần lập trình nâng cao.

*-Hướng phát triển đề tài:*

* Trong thời gian tới em sẽ tiếp tục nghiên cứu sâu hơn về các vấn đề của kiểm thử phần mềm để có thể vận dụng vào kiểm thử các ứng dụng lớn hơn trong thực tế công việc trong tương lai nhằm góp một phần nhỏ bé vào công cuộc chuyên nghiệp hóa kiểm thử phần mềm ở Việt Nam.

**TÀI LIỆU THAM KHẢO**

1. <http://www.testingvn.com/viewforum.php?f=33>

2. <http://en.wikipedia.org/wiki/Software_testing>

3. Một số trang web về kiểm thử phần mềm khác.