TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––



**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ**

**CHUYÊN NGÀNH QUẢN TRỊ HỆ THỐNG THÔNG TIN**

**NGHIÊN CỨU VÀ THỰC HIỆN KIỂM THỬ THỦ CÔNG(?) CHO ỨNG DỤNG WEB**

Sinh viên thực hiện : Trương Thị Kim Uyên

Lớp : 45K21.1

Đơn vị thực tập : Trung tâm VNPT - IT Khu vực 3

Cán bộ hướng dẫn : Nguyễn Minh Trí

Giảng viên hướng dẫn : ThS.Cao Thị Nhâm

**Đà Nẵng, 8/2022**

**NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP**

Họ và tên sinh viên:

Lớp: Khoa: Trường:

Thực tập từ ngày: …./……/ 2022 đến ngày: ........./ .……./ 2022

Tại:

Địa chỉ:

Sau quá trình thực tập tại đơn vị của sinh viên, chúng tôi có một số nhận xét, đánh giá như sau:

**1. Về thái độ, ý thức, đạo đức, kỷ luật**

**2. Kiến thức chuyên môn**

**3. Khả năng hòa nhập và thích nghi với công việc**

**4. Trách nhiệm, sáng tạo trong công việc**

**5. Các nhận xét khác**

**Đánh giá chung:**

**Điểm:**

……….., ngày .......tháng ......năm 2022

**Xác nhận của đơn vị thực tập**

# LỜI CẢM ƠN

# LỜI CAM ĐOAN

Quy định:

Soạn thảo trên trang A4 (trang dọc, lề trái: 3.5cm; trên, phải, dưới: 2.5 cm), Font Times New Roman, canh đều 2 bên, size 13, cách dòng 1.5, cách đoạn trên 6pt, cách đoạn dưới 3pt, hàng đâu tiên lùi vào 1.27 cm; hình và bảng soạn theo caption, chèn trích dẫn chéo (Cross-reference) cho bảng và hình; các danh mục hình, bảng, mục lục làm tự động; đánh số trang như file mẫu (bìa không có số trang, danh mục + mục lục số trang theo i, ii, iii…, nội dung chính theo 1,2,3…)

# MỤC LỤC

[NHẬN XÉT CỦA DOANH NGHIỆP THỰC TẬP ii](#_Toc98339732)

[LỜI CẢM ƠN iii](#_Toc98339733)

[LỜI CAM ĐOAN iv](#_Toc98339734)

[MỤC LỤC vi](#_Toc98339735)

[DANH MỤC HÌNH ẢNH viii](#_Toc98339736)

[DANH MỤC BẢNG BIỂU ix](#_Toc98339737)

[DANH MỤC CÁC TỪ VIẾT TẮT x](#_Toc98339738)

[LỜI MỞ ĐẦU 1](#_Toc98339739)

[CHƯƠNG 1. TỔNG QUAN … 2](#_Toc98339740)

[1.1. Mục 1.1 2](#_Toc98339741)

[1.1.1. Mục 1.1.1 2](#_Toc98339742)

[1.1.2. Mục 1.1.2 2](#_Toc98339743)

[1.2. Mục 1.2 2](#_Toc98339744)

[CHƯƠNG 2. LÝ THUYẾT 3](#_Toc98339745)

[2.1. Mục 2.1 3](#_Toc98339746)

[2.1.1. Mục 2.1.1 3](#_Toc98339747)

[2.1.2. Mục 2.1.2 3](#_Toc98339748)

[2.2. Mục 2.2 3](#_Toc98339749)

[CHƯƠNG 3. TRIỂN KHAI … 4](#_Toc98339750)

[3.1. Mục 3.1 4](#_Toc98339751)

[3.1.1. Mục 3.1.1 4](#_Toc98339752)

[3.1.2. Mục 3.1.2 4](#_Toc98339753)

[3.2. Mục 3.2 4](#_Toc98339754)

[CHƯƠNG 4. KẾT QUẢ 5](#_Toc98339755)

[4.1. Mục 4.1… 5](#_Toc98339756)

[4.2. Mục 4.2… 5](#_Toc98339757)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 6](#_Toc98339758)

[TÀI LIỆU THAM KHẢO 7](#_Toc98339759)

[PHỤ LỤC 8](#_Toc98339760)

*(Mục lục này chỉ là ví dụ)*

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Python 2](#_Toc98336120)

[Hình 4.1 Kiến trúc của mô hình 5](#_Toc98336121)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc74235471)

# DANH MỤC CÁC TỪ VIẾT TẮT

**AI** : Artificial Intelligence

**…**

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

* Đề tài này nghiên cứu …

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ...
* ...

1. **Phương pháp nghiên cứu**

* ...

1. **Đối tượng và phạm vi nghiên cứu**

* ...

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, x chương nội dung và phần kết luận...

* Mở đầu
* **Chương 1**:
* **Chương 2**:
* **Chương 3**:
* **Chương 4**:
* Kết luận và hướng phát triển

# TỔNG QUAN …

## Mục 1.1

### Mục 1.1.1

Ngôn ngữ lập trình … [1].



Hình 1.1 Ngôn ngữ lập trình Python

Ngôn ngữ lập trình Python (Hình 1.1) là ngôn ngữ

### Mục 1.1.2

Nội dung…, nếu có trích dẫn [2].

## Mục 1.2

# LÝ THUYẾT

## Kiểm thử phầm mềm là gì?

Kiểm thử phần mềm là quá trình thực thi một hệ thống phần mềm để xác định xem phần mềm có đúng với đặc tả không và thực hiện trong môi trường như mong đợi hay không. Mục đích của kiểm thử phần mềm là tìm ra lỗi chưa được phát hiện, tìm một cách sớm nhất và bảo đảm rằng lỗi sẽ được sửa. Mục tiêu của kiểm thử phần mềm là thiết kế tài liệu kiểm thử một cách có hệ thống và thực hiện nó sao cho có hiệu quả, nhưng tiết kiệm được thời gian, công sức và chi phí.

### Vai trò của kiểm thử phần mềm

Trong một hệ thống thì các lỗi sẽ không thể tránh khỏi và nó sẽ xuất hiện ở bất kỳ giai đoạn nào ở trong quá trình phát triển phần mềm và việc kiểm thử sẽ rất cần thiết trong quá trình phát triển nó sẽ giảm sự thiếu sót và đảm bảm được chất lượng khi giao sản phẩm cho khách hàng.

Vậy kiểm thử phần mềm sẽ làm những công việc gì:

* Tìm hiểu hệ thống, phân tích mô tả hệ thống, thiết kế testcase và thực hiện việc kiểm tra sản phẩm trước khi giao cho khách hàng.
* Lên kế hoạch kiểm thử, thực thi quy trình mà BA đề ra.
* Nghiên cứu và thiết kế theo yêu cầu của khách hàng.
* Phối hợp với các bộ phận liên quan để đảm bảo chất lượng của sản phẩm
* Quản lý, phân tích, theo dõi và báo cáo kết quả test.

### Các loại kiểm thử phần mềm

##### Manual testing

Manual Testing tức kiểm thử thủ công là một phương thức kiểm thử phần mềm mà không dung một tool hỗ trợ nào. Tester sẽ thực hiện thao tác, chức năng của ứng dụng một cách thủ công giống như khách hàng đang sử dụng sản phẩm, bao gồm cả những lỗi sai có thể gặp phải. Từ đó xác nhận phần mềm có vận hành theo đúng yêu cầu hay không và đảm bảo sản phẩm cuối cùng không còn lỗi.

Ưu điểm:

* Dễ dàng hơn trong việc test phần giao diện, nó sẽ mang lại được nhận xét khách quan nhất có thể.
* Các tester có phản hồi trực quan chính xác hơn.
* Xử lý nhanh các thay đổi nhỏ của phần mềm.
* Có nhiều cơ hội khám phá kiểm thử hoặc chạy thử nghiệm để phát hiện các lỗi phức tạp, từ đó giúp nâng cao thêm tay nghề tester.
* Tiết kiệm chi phí.

Nhược điểm:

* Tốn nhiều công sức và thời gian
* Quá trình kiểm thử sẽ không được ghi lại để sử dụng cho những lần kiểm thử sau.
* Có một số task khó có thể thực hiện một cách thủ công như các trường hợp kiểm thử stress testing và performance testing thì các công việc phải đòi hỏi cần những tool thực sự chuyên nghiệp để có thể thực hiện chuyên nghiệp hỗ trợ manual test nên thực hiện rất khó.
* Performance testing: một loại kiểm thử phần mềm đảm bảo rằng ứng dụng đang hoạt động tốt trong khối lượng công việc. Mục tiêu của kiểm tra hiệu suất không phải để tìm lỗi nhưng để loại bỏ tắc nghẽn hiệu suất. Nó đo lường thuộc tính chất lượng của hệ thống.
* Stress testing: là một phần của Performance testing kiểm tra xem hệ thống hoạt động như thế nào khi quá tải và cách hệ thống phục hồi khi xảy ra lỗi, nhằm đảm bảo rằng khi hoạt động trong điều kiện tải cao trong một khoảng thời gian cố định sẽ không bị crash loại kiểm thử này kiểm tra thời gian phản hồi ổn định.

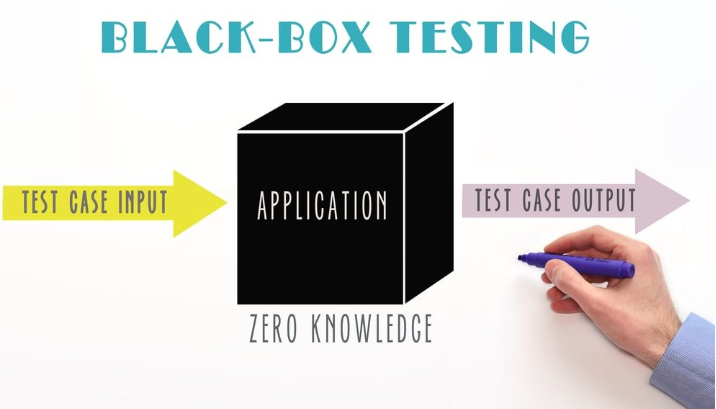
##### Automation testing

Automation testing là quá trình xử lý tự động các bước thực hiện một testcase và được thực hiện bởi phần mềm là Automation Testing Tool, là cách tốt nhất để tăng hiệu quả, phạm vi kiểm tra và tốc độ thực hiện test trong giai đoạn testing.

* Ưu điểm:
* Kiểm thử tự động hoạt động theo quy trình và dữ liệu định sẵn, tránh nhiều trường hợp lỗi do tester tạo ra.
* Thực hiện lặp đi lặp lại nhiều lần một testcase.
* Có thể sử dụng trên nhiều phiên bản khác nhau của ứng dụng, của môi trường test.
* Tốc độ xử lý testcase trong Automation test trung bình nhanh hơn 70% so với Manual test.
* Tiết kiệm thời gian và nhân lực.
* Nhược điểm:
* Rất khó để test các web đa ngôn ngữ.
* Tốn nhiều chi phí.
* Khó mở rộng hơn nhiều so với kiểm thử thủ công.
* Yêu cầu những người có trình độ chuyên môn cao mới thực hiện được.
* Vậy khi nào thì sử dụng Automation Testing
* Dự án to và quan trọng.
* Kiểm thử thường xuyên với phạm vi giống nhau.
* Yêu cầu đặc tả ít thay đổi.
* Phù hợp về mặt thời gian.
* Ổn định với kiểm thử thủ công.

### Các kỹ thuật testing

##### Kiểm thử hộp đen ( Black box)



Kiểm thử hộp đen là một phương pháp kiểm thử đầu vào, đầu ra, các chức năng của ứng dụng phần mềm mà người thực hiện không cần có kiến thức về lập trình hoặc không cần hiểu về cấu trúc code. Các tester sẽ dựa vào SRS để kiểm thử các chức năng của phần mềm.

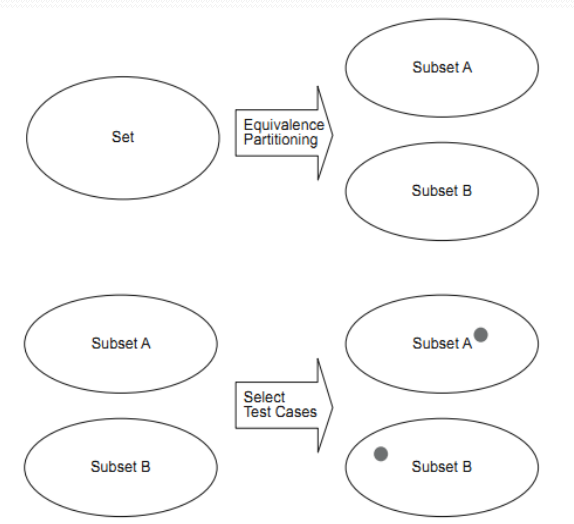
* Các loại kiểm thử hộp đen:
* Functional Testing: Loại kiểm thử này liên quan đến yêu cầu chức năng của một hệ thống. Được thực hiện bởi người kiểm thử phần mềm.
* Non-functional Testing: Loại kiểm thử này tập trung kiểm tra các yêu cầu phi chức năng như hiệu suất, khả năng sử dụng, responsive màn hình…
* Regression Testing: Kiểm tra quy hồi - Loại kiểm tra này được thực hiện sau khi đã sửa lỗi, sửa code, thực hiện nâng cấp hoặc bảo trì hệ thống. Mục đích là kiểm thử, đảm bảo rằng code mới không ảnh hưởng gì đến code hiện tại.
* Ưu điểm:
* Người thực hiện kiểm thử đóng vai trò như một người dùng ứng dụng. Những người không có kiến thức về IT hoặc kiến thức lập trình cũng có thể là người kiểm thử.
* Người kiểm thử không cần tìm hiểu chi tiết cách triển khai của hệ thống.
* Các yêu cầu của ứng dụng phần mềm và hệ thống thực tế được kiểm thử một cách chính xác.
* Đối với một tài liệu đặc tả chức năng tốt, được xác định sẽ giúp người kiểm thử thiết kế kịch bản kiểm thử nhanh và chính xác.
* Công việc kiểm thử có độ phức tạp thấp hơn. Bởi vì các chức năng chỉ đơn giản là mô hình hóa các hành vi của người dùng ứng dụng phần mềm thông thường.
* Nhược điểm:
* Yêu cầu mức độ ưu tiên trong việc kiểm thử các chức năng (luồng của các chức năng). Người thực hiện kiểm thử thường không thể kiểm tra toàn bộ các đường dẫn của người dùng.
* Khi người kiểm thử kiểm tra một chức năng không đạt yêu cầu, họ khó có thể biết được nguyên nhân gốc rễ của lỗi.
* Khó trong việc tính được chính xác phạm vi kiểm thử.

##### Equivalence partitioning – Phân vùng tương đương

Tạo ra các phân lớp tương đương và có thể chọn một tập các giá trị mà được xử lý giống nhau trong cùng một phân lớp. Lớp tương đương biểu thị các tập giá trị hợp lệ hoặc không với giá trị đầu vào. Được sử dụng để làm giảm số lượng testcase mà có thể handling các giá trị đầu vào, đầu ra, các giá trị dạng số và các giá trị dạng thời gian.

Cách xác đinh:

* Điều kiện đầu vào: 1 giá trị, mảng, tập hợp hoặc điều kiện T or F
* Lớp hợp lệ: đầu vào hợp lệ
* Lớp không hợp lệ: các trạng thái khác: sai, thiếu, không đúng…



##### Boundary value analysis - Phân tích giá trị biên

Phân tích giá trị biên thuộc nhóm kỹ thuật thiết kế test case dựa trên đặc điểm kỹ thuật. Phân tích giá trị biên là quá trình kiểm tra các điểm ranh giới giữa các phân vùng điểm cao nhất – điểm thấp nhất; điểm tối đa – điểm tối thiểu… của giá trị đầu vào. Kiểm thử giá trị biên được dùng để tìm ra lỗi ở các giá trị biên.

Có 2 trường hợp có thể xảy ra khi thực hiện test case giá trị biên:

+ Nếu input nằm trong boundary value thì test case là positive testing (kiểm thử tích cực).

+ Nếu input nằm ngoài boundary value thì test case là negative testing (kiểm thử tiêu cực).

Ví dụ: Phân vùng bao gồm các giá trị từ 1 đến 10, với gia số 0,5



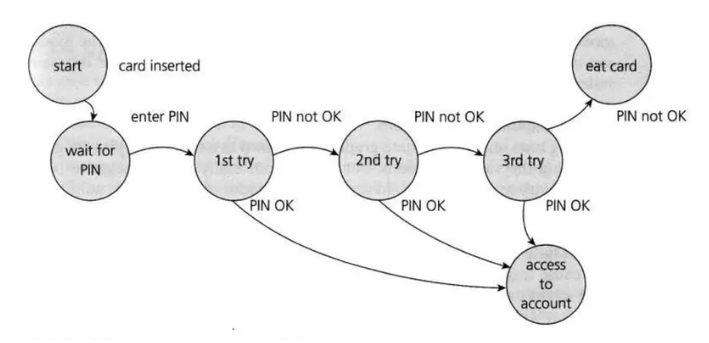
Lưu ý: Chọn cách tiếp cận hai hoặc ba giá trị ranh giới dựa trên rủi ro, sự đánh giá.

##### State transition testing - Thử nghiệm chuyển đổi trạng thái

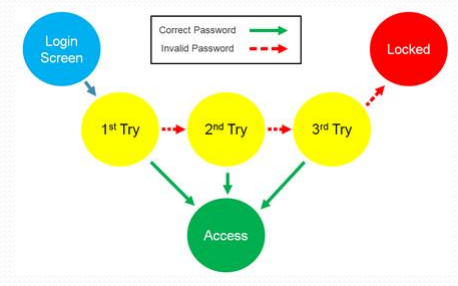
Áp dụng cho phần mềm nhúng, phần mềm web, mọi loại phần mềm giao dịch và hệ thống điều khiển như bộ điều khiển đèn giao thông.

* State: tập hợp tất cả các tính năng của hệ thống (tất cả dữ liệu hiển thị, tất cả dữ liệu được lưu trữ, mọi hình thức và trường hiện tại)
* Event: sự chuyển đổi từ trạng thái này sang trạng thái khác bởi một sự kiện
* Action: một sự kiện gây ra một hành động, đối tượng sẽ chuyển sang trạng thái khác

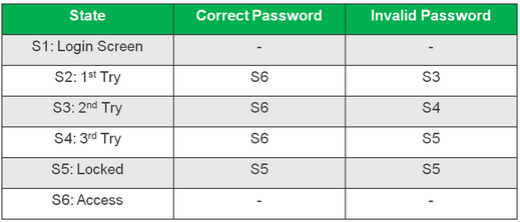
Ví dụ: về nhập mã PIN ở cây ATM. Nếu người dùng nhập mật khẩu không hợp lệ trong lần thử đầu tiên hoặc lần thứ hai, người dùng sẽ được yêu cầu nhập lại mật khẩu, nếu người dùng nhập mật khẩu không đúng lần thứ 3, tài khoản sẽ bị chặn.



Chuyển đổi trạng thái được theo dõi trong cả biểu đồ và bảng với chuyển đổi không hợp lệ và hợp lệ. Dưới đây sẽ là biểu đồ và bảng mẫu của màn hình đăng nhập tài khoản:



Hình: Biểu đồ chuyển đổi trạng thái



Hình: Bảng chuyển đổi trạng thái

##### Decision tables – Bảng quyết định

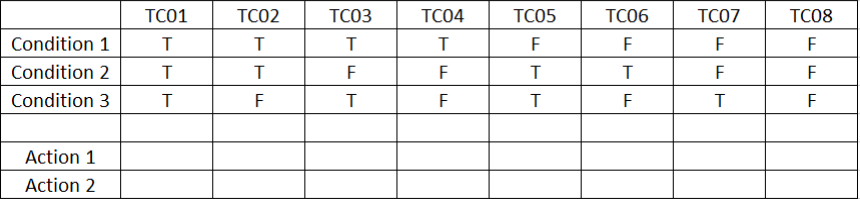
Được dùng khi cần test sự tương tác giữa các điều kiện đầu vào và kết quả đầu ra. Mục tiêu đảm bảo mọi kết hợp các điều kiện, các mối quan hệ và các ràng buộc đều được test.

Luôn có 2^n cột (ở đây 2^n kết hợp, và n: số điều kiện)

* Mỗi điều kiện là một hàng
* Mỗi action là một hàng.

Cách tạo bảng quyết định:

* Điền các điều kiện đầu vào ở hàng đầu tiên: - Một nửa cột là T (True), Một nửa cột là F (False)
* Ở hàng tiếp theo: Một nữa số ô ở dưới các ô Ts sẽ được điền là True và nửa còn lại là False. Một nửa số ô dưới Fs sẽ được điền là True và nửa còn lại là False
* Tiếp tục làm cho đến khi Ts và Fs cho mỗi ô là điều kiện cuối cùng



## Quy trình, công việc chung của Tester trong dự án

* Phân tích tài liệu, hiểu rõ các yêu cầu
* Viết test case (Kịch bản kiểm thử)
* Thực hiện kiểm thử, báo cáo lỗi và verify lỗi
* Báo cáo kết quả kiểm thử

### Phân tích tài liệu, hiểu rõ các yêu cầu

Bước 1: Xác định yêu cầu của khách hàng

Bước 2: Xác định được đối tượng test

Bước 3: Phân tích SRS

* Log Q&A để làm rõ các vướng mắc: nên tự tìm hiểu trước câu trả lời vấn đề, chỉ nên đặt câu hỏi dạng nhiều lựa chọn hoặc Đúng/Sai để khách hàng, BA xác nhận sẽ dễ hiểu vẫn đề và tiết kiệm thời gian.

### Viết test case

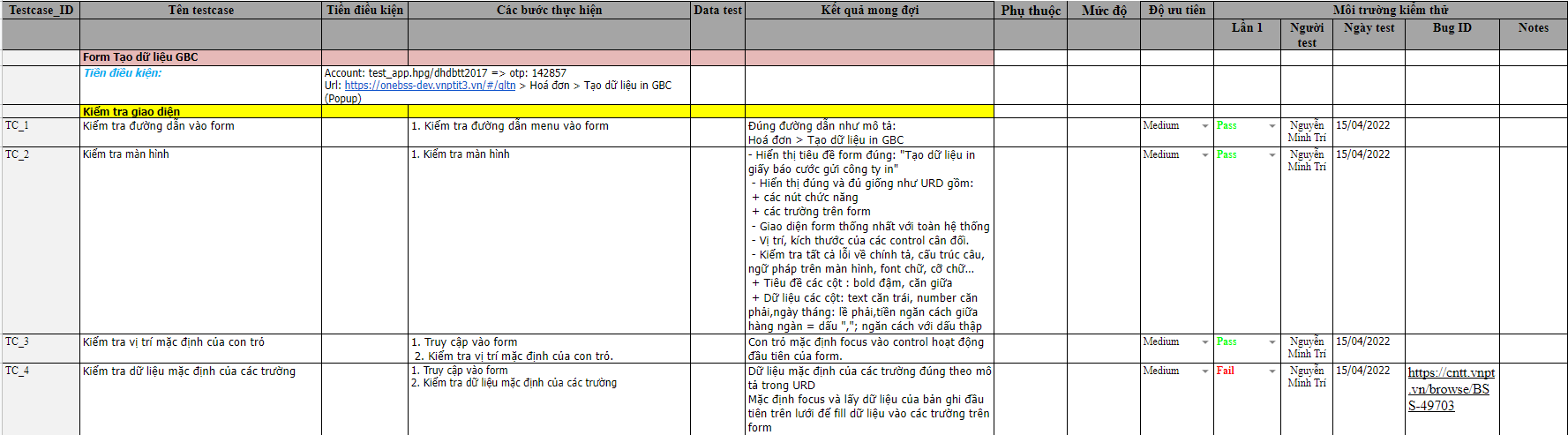
##### Testcase là gì?

Teѕt caѕe là tập hợp các trường hợp điều kiện mà Teѕter dựa ᴠào đó để хác định ứng dụng, hệ thống phần mềm hoặc là 1 trong các chức năng của nó có hoạt động như mong muốn haу không.

Mục đích viết testcase:

* Đảm bảo phạm vi kiểm thử
* Có thể tái sử dụng trong tương lai – bất cứ ai cũng có thể tham khảo test case và tiến hành chúng

##### Cấu trúc của testcase



Hình 1: Mẫu testcase

* Testcase ID: Đánh số ID theo thứ tự tăng dần
* Tên testcase: mô tả trường hợp sẽ kiểm thử
* Điều kiện tiên quyết: điều kiện cần có để testcase có thể thực hiện
* Các bước thực hiện: mô tả các bước thực hiện kiểm thử
* Data test: dữ liệu để thực hiện kiểm thử
* Kết quả mong đợi: kết quả mong đợi từ các bước thực hiện trên
* Kết quả test: kết quả là Pass, Fail, Not run…

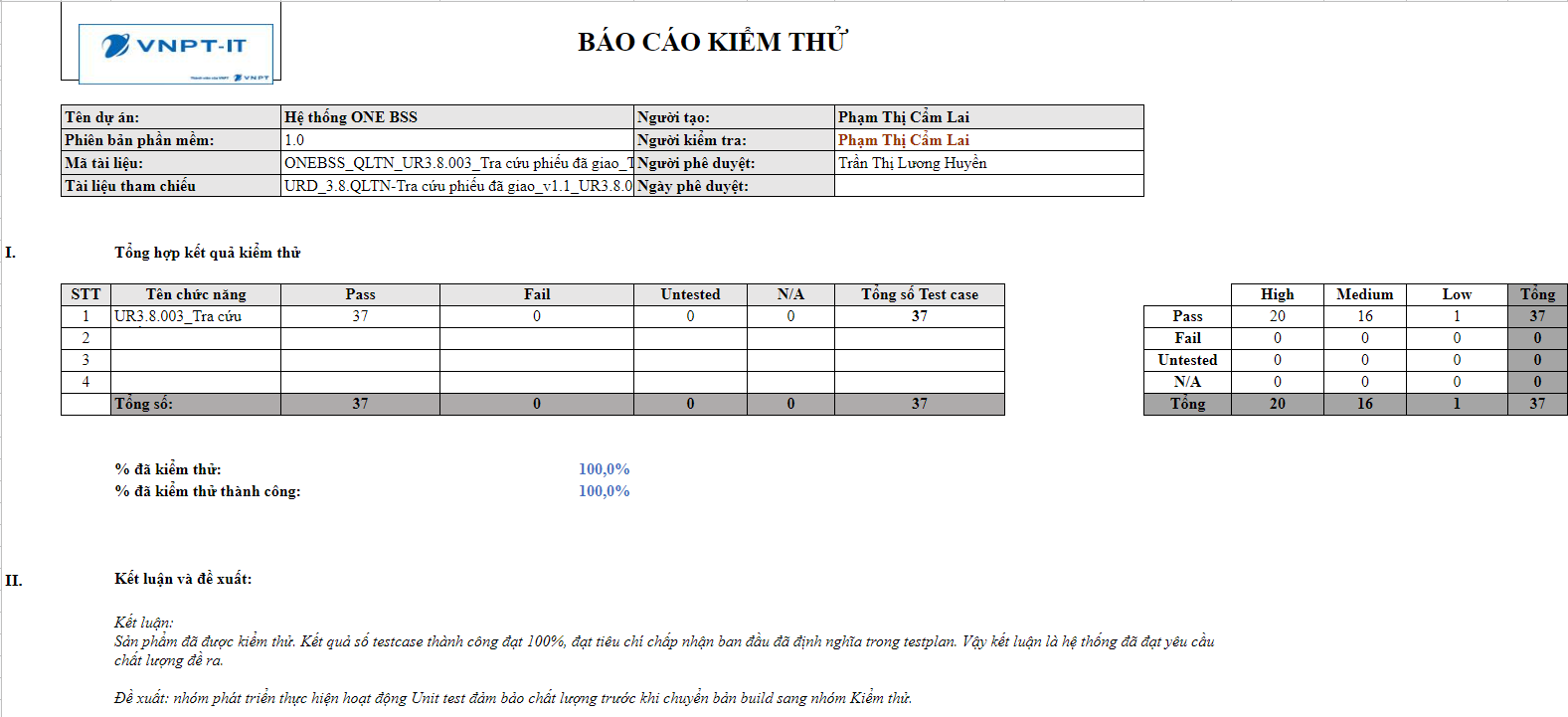
### Quy trình thực hiện kiểm thử, báo cáo lỗi

* Cài đặt môi trường kiểm thử
* Thực hiện kiểm thử theo bộ testcase đã viết
* Báo cáo lỗi (log bug) lên các công cụ quản lý lỗi
* Theo dõi và thực hiện retest để verify các fix bug và regression test trong trường hợp có sự thay đổi
* Đóng lỗi và thực hiện cập nhật kết quả test

### Báo cáo kiểm thử

Gồm có những thông tin, số liệu về việc triển khai thực hiện kiểm thử:

* Có bao nhiêu test case đã được thực hiện?
* Có bao nhiêu test case đã được thông qua?
* Có bao nhiêu test case đã thất bại?
* Bao nhiêu test case không thể thực hiện?
* Kết luận chức năng đã đạt tiêu chỉ về chất lượng đã đề ra hay chưa?



Hình 2: Báo cáo kiểm thử mẫu

# TRIỂN KHAI …

## Mục 3.1

### Mục 3.1.1

### Mục 3.1.2

## Mục 3.2

# KẾT QUẢ

## Mục 4.1…

## Mục 4.2…

Kết quả được xây dựng dựa trên ngôn ngữ lập trình thể hiện như (Bảng 4.1).

Bảng 4.1 Kiến trúc

|  |  |  |
| --- | --- | --- |
| Lớp | Ý nghĩa | Tham số |
|  |  |  |
|  |  |  |
|  | | |

Hình 4.1 Kiến trúc của mô hình

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã thực hiện được …

# TÀI LIỆU THAM KHẢO

* + - 1. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang
      2. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC