

**BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  
KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: KIỂM THỬ TỰ ĐỘNG ỨNG DỤNG WEB VỚI CÔNG CỤ  
QUICK TEST PROFESSIONAL**

*Sinh viên thực hiện* : Đặng Thị Nhỏ

*Lớp* : HTTT2 – K8

*Mã sinh viên* : 0841260153

*Giảng viên hướng dẫn* : Tiến sĩ Nguyễn Hoàng Tú

**Hà Nội, tháng 05 năm 2017**

## NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của GV hướng dẫn)

This image shows a full page of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Điểm..... (Bằng Chữ ..... )

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp.

Hà Nội, Ngày ..... Tháng ..... Năm 2017

[illegible]

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp.

3

## LỜI NÓI ĐẦU

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ từ quý thầy cô, gia đình và bạn bè.

Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý thầy cô ở khoa Công nghệ thông tin – Trường Đại học Công nghiệp Hà Nội, những người đã cùng với tri thức và tâm huyết của mình truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Với vốn kiến thức được tiếp thu trong quá trình học tập đó, nó không chỉ là nền tảng cho quá trình báo cáo mà còn là hành trang quý báu để em bước vào đời một cách vững chắc và tự tin hơn.

Em xin chân thành cảm ơn thầy Nguyễn Hoàng Tú đã tận tâm hướng dẫn em trong suốt quá trình thực tập tốt nghiệp. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì thiết nghĩ bài báo cáo này của em rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn thầy!

*Em xin chân thành cảm ơn!*

## TÓM TẮT ĐỒ ÁN

### **Mục đích:**

- Tìm hiểu tổng quan về kiểm thử phần mềm, mô hình ứng dụng web và kiểm thử các ứng dụng web.
- Tìm hiểu về kiểm thử tự động và các công cụ kiểm thử tự động
- Tìm hiểu, nghiên cứu công cụ Quick test professional
- Lập kế hoạch kiểm thử, phân tích thiết kế kiểm thử, sử dụng Quick test professional để thực hiện kiểm thử.

### **Kết luận:**

- Tìm hiểu về kiểm thử phần mềm; Các phương pháp kiểm thử; Các chiến lược kiểm thử; Các cấp độ kiểm thử trong kiểm thử phần mềm.
- Giới thiệu công cụ kiểm thử phần mềm Quick Test Professional, các thao tác cơ bản để sử dụng công cụ Quick Test Professional.
- Áp dụng các kiến thức đã nghiên cứu thực hiện kiểm thử chức năng đăng nhập và chức năng tìm kiếm cho website bán điện thoại di động.

## MỤC LỤC

<b>MỞ ĐẦU .....</b>	<b>10</b>
<b>1. Tên đề tài.....</b>	<b>10</b>
<b>2. Lý do chọn đề tài.....</b>	<b>10</b>
<b>3. Mục đích nghiên cứu .....</b>	<b>10</b>
<b>4. Bố cục đề tài.....</b>	<b>10</b>
<b>CHƯƠNG 1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM.....</b>	<b>12</b>
<b>1.1.Định nghĩa.....</b>	<b>12</b>
<b>1.2.1. Căn cứ vào phương pháp kiểm thử.....</b>	<b>12</b>
<b>1.2.2.Căn cứ vào giai đoạn kiểm thử .....</b>	<b>13</b>
<b>1.2.3.Căn cứ vào kiểu kiểm thử.....</b>	<b>14</b>
<b>1.3.Mục đích của kiểm thử .....</b>	<b>15</b>
<b>1.4.Kỹ thuật kiểm thử.....</b>	<b>15</b>
<b>1.4.1. Phương pháp kiểm thử là gì?.....</b>	<b>15</b>
<b>1.4.2.Kỹ thuật kiểm thử .....</b>	<b>15</b>
<b>1.5.Các thuật ngữ trong kiểm thử phần mềm .....</b>	<b>18</b>
<b>1.6.Các nguyên nhân gây ra lỗi.....</b>	<b>19</b>
<b>1.7.Chi phí sửa lỗi phần mềm .....</b>	<b>21</b>
<b>1.8.Tổng kết .....</b>	<b>22</b>
<b>CHƯƠNG 2. TÌM HIỂU VỀ KIỂM THỬ TỰ ĐỘNG.....</b>	<b>23</b>
<b>2.1.Tổng quan về kiểm thử tự động.....</b>	<b>23</b>
<b>2.2.Kiến trúc của một bộ kiểm thử tự động.....</b>	<b>24</b>
<b>2.3.Uưu và nhược điểm của kiểm thử tự động .....</b>	<b>24</b>
<b>2.4.Một số công cụ của kiểm thử tự động .....</b>	<b>25</b>
<b>2.5.Tổng kết kiểm thử tự động.....</b>	<b>26</b>
<b>CHƯƠNG 3. TỔNG QUAN VỀ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG</b>	
<b>QUICKTEST PROFESSIONAL .....</b>	<b>27</b>
<b>3.1.Giới thiệu về công cụ Quicktest Professional .....</b>	<b>27</b>
<b>3.2.Loại phần mềm hỗ trợ .....</b>	<b>27</b>
<b>3.3.Đặc Điểm.....</b>	<b>28</b>
<b>3.4.Các thành phần quan trọng trong QuickTest Professional .....</b>	<b>30</b>
<b>3.4.1. Action .....</b>	<b>30</b>

3.4.2. Data Table.....	30
3.4.3. Object Repository (OR).....	30
3.4.4. Checkpoint.....	30
3.5.Ngôn ngữ sử dụng viết Script.....	30
3.6.Cách sử dụng Quick test professional.....	31
3.7.Tổng kết về công cụ Quick test professional .....	34
<b>CHƯƠNG 4. ỨNG DỤNG.....</b>	<b>35</b>
4.1.Giới thiệu ứng dụng web .....	35
4.2.Phân tích thiết kế hệ thống.....	35
4.2.1.Đặc tả hệ thống.....	35
4.2.2.Thiết kế hệ thống.....	36
4.3.Kế hoạch kiểm thử .....	38
4.3.1.Mục đích.....	38
4.3.2.Phạm vi kiểm thử .....	38
4.3.3.Công cụ, môi trường test .....	38
4.3.4.Kế hoạch test .....	39
4.3.5.Nhân sự và trách nhiệm test .....	40
4.3.6.Điều kiện bắt đầu test, dừng test và kết thúc test. ....	40
4.4.Quy trình kiểm thử .....	42
4.4.1.Kiểm thử đơn vị.....	42
4.4.2.Kiểm thử hệ thống.....	48
4.4.3.Thực hiện kiểm thử.....	51
4.4.4.Kết quả kiểm thử.....	55
4.4.5.Các lỗi tồn tại trong hệ thống.....	58
<b>KẾT LUẬN .....</b>	<b>60</b>

## **DANH SÁCH HÌNH VẼ**

Hình 1.1. Sơ đồ: Chi phí cho việc sửa lỗi phần mềm

Hình 2.1. Kiến trúc chung của một bộ kiểm thử tự động

Hình 2.2. Hàm IsTrangle làm đầu vào cho công cụ CFT4CUnit

Hình 3.1. Biểu tượng Quicktest professional trên màn hình

Hình 3.2. Tạo mới một test script

Hình 3.3. Chọn loại kiểm thử

Hình 3.4. Tạo Record

Hình 3.5. Tùy chọn ghi và chạy test script

Hình 3.6. Click nút record để thực hiện ghi

Hình 4.1. Giao diện Website

Hình 4.2. Biểu đồ usecase tổng quát

Hình 4.3. Giao diện đăng nhập

Hình 4.4. Giao diện chức năng tìm kiếm

Hình 4.5. Kịch bản kiểm thử chức năng đăng nhập

Hình 4.6. Kịch bản kiểm thử chức năng tìm kiếm

Hình 4.7. Kết quả của chức năng đăng nhập

Hình 4.8. Kết quả của chức năng tìm kiếm



## **DANH MỤC BẢNG BIỂU**

Bảng 3.1. Các phần mềm cần cài đặt bổ sung

Bảng 4.1. Danh sách các Usecase

Bảng 4.2. Kế hoạch test

Bảng 4.3. Nhân sự và trách nhiệm test

Bảng 4.4. Các trường hợp kiểm thử chức năng đăng nhập

Bảng 4.5. Các trường hợp kiểm thử chức năng tìm kiếm

## MỞ ĐẦU

### 1. Tên đề tài

***Kiểm thử tự động với công cụ Quick Test Professional.***

### 2. Lý do chọn đề tài

Ngày nay, tự động hóa đang được nghiên cứu và ứng dụng trong nhiều lĩnh vực trong đó công nghệ phần mềm nói chung và kiểm thử phần mềm nói riêng cũng không ngoại lệ. Khi mà kiểm thử phần mềm vẫn tiêu tốn một lượng lớn thời gian, kinh phí và nhân lực trong một dự án phần mềm thì song song với kiểm thử truyền thống thủ công, sự ra đời của các công cụ hỗ trợ kiểm thử tự động như Selenium, Quick Test Professional, Nunit, Junit, Load Runner (thường dùng trong kiểm thử hiệu năng) là tất yếu. Trong số đó, Quick Test Professional của hãng Mercury khá tốt và mạnh, bao gồm nhiều chức năng điển hình của một công cụ kiểm tra tự động. Đây là công cụ áp dụng phương pháp Keyword-Driven, một kỹ thuật scripting (lập trình trong KTTĐ) hiện đại, cho phép KTV bổ sung test case bằng cách tạo file mô tả cho nó mà không phải chỉnh sửa hay bổ sung bất cứ script nào cả. Nó cũng phù hợp trong tình huống chuyển giao công việc mà người mới tiếp nhận chưa có thời gian hoặc chưa hiểu script vẫn có thể thực hiện kiểm tra phần mềm theo đúng yêu cầu. Tuy chưa được ứng dụng nhiều trong các tổ chức ở Việt Nam, song với những ưu điểm trên, Quick Test Professional hứa hẹn sẽ ngày càng phát triển và trở lên thông dụng hơn trong các tổ chức phát triển phần mềm ở nước ta.

Và để làm tiền đề cho định hướng tương lai khi tốt nghiệp đại học sẽ trở thành một kỹ sư kiểm thử phần mềm, em lựa chọn đề tài “***Kiểm thử tự động với công cụ Quick Test Professional***” làm đề tài tốt nghiệp của mình.

### 3. Mục đích nghiên cứu

Đề tài tìm hiểu cơ sở lý thuyết về kiểm thử và cách sử dụng công cụ Quick Test Professional để đảm bảo chất lượng phần mềm.

### 4. Bố cục đề tài

Những nội dung và kết quả tìm hiểu của đề tài được trình bày trong 3 chương như sau:

- Chương 1: Tổng quan về kiểm thử phần mềm: Chương này trình bày được những định nghĩa và những vấn đề cơ bản xung quanh kiểm thử phần mềm, lỗi phần mềm.

- Chương 2: Kiểm thử tự động và công cụ hỗ trợ: Chương này trình bày các vấn đề cơ bản của kiểm thử tự động và một số công cụ hỗ trợ kiểm thử phần mềm.
- Chương 3: Công cụ kiểm thử tự động Quick Test Professional: Chương này trình bày các thành phần của công cụ Quick Test Professional, nắm được cách sử dụng và vận hành của nó trên một ứng dụng cụ thể.
- Chương 4: Ứng dụng.

## **5. Phương pháp nghiên cứu**

Phương pháp nghiên cứu lý thuyết:

- Tiến hành thu thập và nghiên cứu các tài liệu có liên quan đến đề tài.
- Tổng hợp các tài liệu.
- Chọn lọc các tài liệu nghiên cứu để báo cáo đề tài thực tập.

Phương pháp nghiên cứu thực nghiệm:

- Tìm hiểu cách thức, tiêu chí kiểm thử chức năng.
- Ứng dụng công cụ Quick Test Professional vào kiểm thử phần mềm.
- Kiểm tra và đánh giá kết quả đạt được.

## CHƯƠNG 1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Kiểm thử phần mềm cũng nhằm phát hiện lỗi hoặc bất cứ vấn đề gì về sản phẩm. Chúng ta cần kiểm thử vì biết rằng con người luôn có thể mắc sai lầm. Điều này đặc biệt đúng trong lĩnh vực phát triển phần mềm và các hệ thống phát triển phần mềm.

### 1.1. Định nghĩa

Có nhiều định nghĩa khác nhau về kiểm thử phần mềm, đề xuất bởi nhiều tổ chức hay nhiều cá nhân khác nhau. Sau đây là một số định nghĩa nổi bật:

- *Định nghĩa của Myer (1979)*: “Kiểm thử là quá trình thực thi một chương trình với mục đích tìm ra lỗi”. Theo như định nghĩa này, Quá trình kiểm thử bao gồm tất cả các hoạt động từ kiểm tra mã nguồn được thực hiện bởi trưởng nhóm phát triển, đến việc chạy thử chương trình được tiến hành bởi các đồng nghiệp khác. Tất cả các hoạt động trên đều được coi là các hoạt động kiểm thử.
- *Hai định nghĩa của IEEE (1990)*:
  - Định nghĩa 1: Kiểm thử phần mềm là quá trình vận hành một hệ thống hoặc một thành phần của hệ thống với các điều kiện xác định, nhận xét và ghi lại các kết quả, tạo ra đánh giá về những khía cạnh của hệ thống hay thành phần của hệ thống.
  - Định nghĩa 2: Kiểm thử phần mềm là quá trình phân tích các yếu tố phần mềm để phát hiện những khác biệt giữa chương trình với các điều kiện yêu cầu và đánh giá các đặc điểm của các yếu tố phần mềm.
- *Định nghĩa của Daniel Galin*: “Kiểm thử phần mềm là một quá trình được tiến hành bởi một nhóm chuyên viên kiểm thử, trong đó một đơn vị phần mềm, một nhóm các đơn vị được tích hợp, hoặc cả gói phần mềm được kiểm tra bằng cách chạy các chương trình trên máy tính. Tất cả các bước kiểm tra được tiến hành theo các thủ tục kiểm thử và các trường hợp kiểm thử đã được thông qua”.

### 1.2. Phân loại kiểm thử

#### 1.2.1. Căn cứ vào 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 kiểm 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 kiểm thử phần mềm thông qua việc dùng máy chạy chương trình để điều tra trạng thái tác động của chương trình. Đó là kiểm thử dựa trên các ca kiểm thử xác định bằng sự thực hiện của đối tượng kiểm thử hay chạy các chương trình. Kiểm thử động kiểm tra cách thức hoạt động 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 phương pháp kiểm thử động gồm có:

- Kiểm thử Unit – *Unit Tests*
- Kiểm thử tích hợp – *Integration Tests*
- Kiểm thử hệ thống – *System Tests*
- Kiểm thử chấp nhận sản phẩm – *Acceptance Tests*. [1]

**1.2.2. Căn cứ vào giai đoạn kiểm thử**

- **Unit test (kiểm thử đơn vị):**

Thường được thực hiện bởi developer. Chia nhỏ thành các đơn vị để test.

Unit test để đảm bảo rằng mỗi method và function đều hoạt động đúng. Các test phải độc lập với nhau

Còn functional test là để đảm bảo rằng kết quả trả về của ứng dụng đúng như mong đợi.

- **Integration Testing (kiểm thử tích hợp):**

Kiểm tra luồng dữ liệu, giao tiếp giữa các thành phần/module bằng cách tích hợp các module với nhau.

- **System Testing (kiểm thử hệ thống):**

Được thực hiện bởi một team chuyên trách riêng biệt, Kiểm tra toàn bộ hệ thống để xác định hệ thống đáp ứng được cả về mặt chức năng, nghiệp vụ và kiến trúc ứng dụng, thông số kỹ thuật. Hệ thống phải được dựng trên môi trường sát với thực tế nhất.

- **Acceptance Testing (kiểm thử chấp nhận):**

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**.

- Alpha Test: người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm, lập trình viên sẽ ghi nhận các lỗi hoặc phản hồi, và lên kế hoạch sửa chữa.
- Beta Test: phần mềm sẽ được gửi tới cho người dùng để kiểm thử ngay trong môi trường thực, lỗi hoặc phản hồi cũng sẽ gửi ngược lại cho lập trình viên để sửa chữa.

Thực tế cho thấy, nếu khách hàng không quan tâm và không tham gia vào quá trình phát triển phần mềm thì kết quả Acceptance Test sẽ sai lệch rất lớn, mặc dù phần mềm đã trải qua tất cả các kiểm thử trước đó. Sự sai lệch này liên quan đến việc hiểu sai yêu cầu cũng như sự mong chờ của khách hàng. Ví dụ đôi khi một phần mềm xuất sắc vượt qua các phép kiểm thử về chức năng thực hiện bởi nhóm thực hiện dự án, nhưng khách hàng khi kiểm thử sau cùng vẫn thất vọng vì bố cục màn hình nghèo nàn, thao tác không tự nhiên, không theo tập quán sử dụng của khách hàng ... 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 ... Tất cả tài liệu đi kèm phải được cập nhật và kiểm thử chặt chẽ.

- **Regression Test – Kiểm thử hồi quy:**

Bên cạnh các loại kiểm thử trên, còn có kiểm thử hồi quy.

Trước tiên cần khẳng định Regression Test không phải là một mức kiểm tra, như các mức khác đã nói ở trên. Nó đơn thuần kiểm tra lại phần mềm sau khi có một sự thay đổi xảy ra, để bảo đảm phiên bản phần mềm mới thực hiện tốt các chức năng như phiên bản cũ và sự thay đổi không gây ra lỗi mới trên những chức năng vốn đã làm việc tốt. Regression test có thể thực hiện tại mọi mức kiểm tra.[5]

### 1.2.3. Căn cứ vào kiểu kiểm thử

- **Manual Testing (test thủ công):** Không sử dụng tool hay script mà tester đóng vai trò là người dùng cuối để kiểm tra.

**Quy trình kiểm thử thủ công:**

- Lập kế hoạch và kiểm soát
- Phân tích yêu cầu và thiết kế testcase
- Thực hiện test theo testcase
- Xác định tiêu chí hoàn thành và làm báo cáo
- Hoạt động đóng dự án.

- **AutomationTesting (test tự động):** Kiểm thử tự động là quá trình xử lý một cách tự động các bước thực hiện các test case. Kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử.

#### **Quy trình kiểm thử tự động**

- Bước 1: Viết kịch bản kiểm thử, dùng công cụ kiểm thử để ghi lại các thao tác lên phần mềm cần kiểm tra và tự động sinh ra test script.
- Bước 2: Chỉnh sửa để kịch bản kiểm thử thực hiện kiểm tra theo đúng yêu cầu đặt ra, làm theo trường hợp kiểm thử cần thực hiện.
- Bước 3: Chạy kịch bản kiểm thử, giám sát hoạt động kiểm tra phần mềm của kịch bản kiểm thử
- Bước 4: Kiểm tra kết quả thông báo sau khi thực hiện kiểm thử tự động. Sau đó bổ sung, chỉnh sửa những sai sót.

### **1.3. Mục đích của kiểm thử**

Kiểm thử phần mềm là một cuộc kiểm tra nhằm cung cấp chi các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ đang kiểm thử. Kiểm thử phần mềm cũng có thể xem như là quá trình thẩm định và thẩm tra phần mềm/ chương trình, ứng dụng/ sản phẩm để:

- Đáp ứng được các yêu cầu công việc và kỹ thuật đã được quy định trong thiết kế và trong lúc phát triển.
- Làm việc được như mong đợi.
- Có thể thực thi với các đặc tính giống nhau.
- Đáp ứng được nhu cầu của các bên liên quan.

### **1.4. Kỹ thuật kiểm thử**

#### **1.4.1. Phương pháp kiểm thử là gì?**

Là hoạt động kiểm tra xem phần mềm có chạy chính xác hay không (Verification) và có thỏa mãn yêu cầu của khách hàng hay không (Validation) nhằm hướng tới mục tiêu Chất lượng cho phần mềm. Các hoạt động Verification chiếm 80%, Validation chiếm 20% khối lượng công việc kiểm thử. Tuy nhiên, hiệu quả của Validation có thể tác động lên 80% hiệu quả chung của dự án/sản phẩm. Verification là điều kiện Cần, còn Validation là điều kiện Đủ.

Chúng ta không thể kiểm tra tất cả các trường hợp vì những yếu tố như: Tiến độ, Chi phí, tài nguyên.... Chúng ta cần có quá trình lựa chọn các trường hợp để test sao cho hiệu quả.

⇒ Kỹ thuật kiểm thử sẽ giúp chúng ta làm việc đó

#### **1.4.2. Kỹ thuật kiểm thử**

- Là phương thức để lựa chọn hoặc thiết kế testcase
- Dựa trên cấu trúc, chức năng của phần mềm
- Tìm kiếm được nhiều lỗi tốn ít nỗ lực
- Sinh ra những trường hợp kiểm thử tốt
- Đánh giá khách quan hiệu quả kiểm thử

- Tránh việc trùng lặp testcase
- Tập trung vào những lỗi đặc thù.

Có 2 kiểu kiểm thử thông dụng là: Kiểm thử hộp trắng và kiểm thử hộp đen[2]

❖ **White box testing:** Kiểm thử các code của hệ thống, kỹ thuật này có các ưu điểm:

- + Phát hiện lỗ hổng bảo mật
- + Phát hiện những dòng code không tối ưu, code thừa
- + Kiểm tra đầu vào một cách cụ thể
- + Test chi tiết đến từng dòng code.

**Các kỹ thuật kiểm thử hộp trắng – White box testing:**

**Bao phủ câu lệnh – Statement Coverage (SC):** Lựa chọn số testcase tối thiểu để đi qua tất cả các dòng code ít nhất 1 lần

**Bao phủ rẽ nhánh – Bao phủ quyết định Branch Coverage (or DecisionCoverage) (BC):** Số testcase tối thiểu đi qua các nhánh (các quyết định Yes/ No) ít nhất 1 lần (các câu lệnh rẽ nhánh như switch, do-while, và if-else)

Công thức tính độ bao phủ dòng lệnh/ rẽ nhánh (của cả dự án\_ phục vụ phân định giá\_Metric)

$\text{Statement Coverage} = (\text{Number of statements exercised} / \text{Total number of statements}) * 100 \Rightarrow (\text{Số câu lệnh đã được kiểm tra} / \text{tổng số câu lệnh}) * 100\%$   
100% độ bao phủ không có nghĩa là đã được test 100%.

**Các loại kiểm thử của kỹ thuật test White box testing:**

- **Unit Testing:**

Là kiểu test đầu tiên được áp dụng trong ứng dụng.

Được thực hiện trên từng đơn vị được thực hiện bởi dev.

Phát hiện lỗi sớm và chi phí sửa lỗi rẻ, dễ dàng.

- **Testing for Memory Leaks (Test tràn bộ nhớ)**

Memory leaks Là nguyên nhân hàng đầu dẫn tới việc ứng dụng chạy chậm

Có nhiều tool hỗ trợ việc test Memory leaks, ví dụ: Rational Purify for windows application.

- **Penetration Testing (Test thâm nhập hệ thống)**

Mục đích là Tấn công bằng nhiều cách, góc cạnh để phát hiện lỗi bảo mật.

Có đầy đủ thông tin về source code, thông tin mạng, IP, thông tin server của ứng dụng.



- **Mutation Testing (Test đột biến)**

+ Mục đích: đánh giá hiệu quả của test case, không phải đánh giá chương trình đang test. (Loại test này sẽ chỉ đánh giá hiệu quả của test case (ở đây là các case unit test mà dev đã tạo, thường dùng tool để test)

+ Khi đã test theo test case OK rồi mà vẫn thấy lẫn tăn nghi ngờ về việc còn sót lỗi thì làm tiếp mutation test. Cách làm này mình sẽ sửa code cho sai đi và vẫn dùng test case đó test. Nếu không case nào bị lỗi chứng tỏ bị thiếu case. Cần check lại và bổ sung case rồi test lại. Mỗi lần sửa code lưu riêng ra 1 bản và check lại. Cái này cũng có nhiều hạn chế do phải phán đoán để sửa code hoặc phải sửa toàn bộ code mất thời gian. Nhưng chuyên gia Microsoft đánh giá là hiệu quả với các hệ thống tầm trung.

❖ **Kiểm thử hộp đen - Black box testing:**

Là một kỹ thuật kiểm thử phần mềm, trong đó việc kiểm tra các chức năng mà không cần quan tâm đến việc thực thi các câu lệnh hoặc kiến trúc bên trong. Việc kiểm thử hoàn toàn dựa vào đặc tả và yêu cầu của khách hàng. Trong kỹ thuật test hộp đen chúng ta tập trung quan tâm đến việc kiểm tra đầu vào, đầu ra mà không cần biết kiến thức về lập trình.

**Các kiểu test của kỹ thuật test hộp đen:**

- **Functional Testing:**

Dựa trên đặc tả của phần mềm và đánh giá việc đáp ứng được các yêu cầu trong đặc tả. Được Tester thực hiện trong giai đoạn kiểm thử.

- **Non-Functional Testing:**

Không phải là test 1 chức năng cụ thể nhưng yêu cầu phi chức năng ví dụ như là: Yêu cầu về hiệu năng, Khả năng mở rộng (tính khả dụng), khả năng sử dụng (thân thiện hay ko thân thiện.. )

- **Regression Testing:**

Test hồi quy được thực hiện khi có bất cứ thay đổi nào về phần mềm. Để kiểm tra việc sửa code, nâng cấp, bảo trì nhưng không dẫn đến kết quả sai khác về chức năng hoặc nghiệp vụ.

**Chiến lược của Black box testing:**

- Phân lớp tương đương: Phân chia tập hợp các điều kiện kiểm thử thành các vùng/ lớp tương tự nhau.

- Kiểm tra giá trị biên: Tập trung vào kiểm tra các giá trị tại biên, xung quanh biên và đưa ra quyết định chọn giá trị đầu vào để kiểm tra. Có tác dụng làm giảm số lượng các trường hợp cần kiểm thử. Phù hợp với các hệ thống mà đầu vào để kiểm tra. Có tác dụng làm giảm số lượng các trường hợp cần kiểm thử. Phù hợp với các hệ thống mà đầu vào có giới hạn rõ ràng (có phạm vi nhất định)

### Ưu điểm và 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. Sử dụng nguyên tắc “hãy đòi hỏi và bạn sẽ được nhận”, những kiểm thử viên tìm ra lỗi mà những lập trình viên đã không tìm ra. Nhưng mặt khác, người ta cũng nói kiểm thử hộp đen “Giống như là đi trong bóng tối mà không có đèn”, bởi vì 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 ra rất nhiều ca kiểm thử để kiểm tra mà đáng lẽ có thể chỉ cần kiểm tra bằng 1 ca kiểm thử và có thể 1 số phần của chương trình lại không được kiểm tra đến.

Vì vậy, kiểm thử hộp đen có ưu điểm là “một sự đánh giá khách quan”, mặt khác nó lại có nhược điểm là “thăm dò mù”

### ❖ Kiểm thử hộp xám

Kiểm thử hộp xám đòi hỏi phải có sự truy cập tới cấu trúc dữ liệu và giải thuật bên trong cho những mục đích thiết kế ca kiểm thử, nhưng là kiểm thử ở mức người sử dụng hay mức hộp đen. Việc thao tác tới dữ liệu đầu vào và định dạng dữ liệu đầu ra là không rõ ràng, giống như một chiếc “*hộp xám*”, bởi vì đầu vào và đầu ra rõ ràng là ở bên ngoài “*hộp đen*” mà chúng ta vẫn gọi về hệ thống được kiểm tra. Sự khác biệt này đặc biệt quan trọng khi quản lý kiểm thử tích hợp – *Intergartion testing* giữa 2 modun mã lệnh được viết bởi hai chuyên viên thiết kế khác nhau, trong đó chỉ giao diện là được đưa ra để kiểm thử.

Kiểm thử hộp xám có thể cũng bao gồm cả thiết kế đối chiếu để quyết định, ví dụ, giá trị biên hay thông báo lỗi

### 1.5. Các thuật ngữ trong kiểm thử phần mềm

**Lỗi (Error):** Lỗi là những vấn đề mà con người mắc phải trong quá trình phát triển các sản phẩm phần mềm. Trong thực tế, con người luôn có thể phạm lỗi. Khi lập trình viên phạm lỗi trong lập trình, ta gọi các lỗi đó là bug (con bọ). Lỗi có thể phát tán. Chẳng hạn, một lỗi về xác định yêu cầu có thể dẫn đến sai lầm về thiết kế và càng sai khi lập trình theo thiết kế này. Lỗi là nguyên nhân dẫn đến sai.

**Sai (Fault):** Sai là kết quả của lỗi, hay nói khác đi, lỗi sẽ dẫn đến sai. Cũng có thể nói sai là một biểu diễn của lỗi dưới dạng một biểu thức, chẳng hạn chương trình, văn bản, sơ đồ dòng dữ liệu, biểu đồ lớp,.... Sai lầm có thể khó bị phát hiện. Khi nhà thiết kế mắc lỗi bỏ sót trong quá trình thiết kế, sai kết quả từ lỗi đó là thiếu mất cái gì đó mà lẽ ra cần phải có. Sai về nhiệm vụ xuất hiện khi vào sai thông tin, còn sai về bỏ quên xuất hiện khi không vào đủ thông tin. Loại sai thứ hai khó phát hiện và khó sửa hơn loại sai thứ nhất.

**Thất bại (Failure):** Thất bại xuất hiện khi một lỗi được thực thi. Có hai điều cần lưu ý ở đây. Một là thất bại chỉ xuất hiện dưới dạng có thể chạy được mà

thông thường là mã nguồn. Hai là các thất bại chỉ liên kết với các lỗi về nhiệm vụ. Còn các thất bại tương ứng với các lỗi về bỏ quên thì xử lý thế nào? Những cái lỗi không bao giờ được tiến hành, hoặc không được tiến hành trong khoảng thời gian dài cần được xử lý thế nào? Virus Michaelangelo là một ví dụ về lỗi loại này. Nó chỉ được tiến hành vào ngày sinh của Michaelangelo, tức ngày 6/3 mà thôi. Việc khảo sát có thể ngăn chặn nhiều thất bại bằng cách tìm ra các lỗi thuộc cả hai loại.

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

### 1.6. Các nguyên nhân gây ra lỗi

Lỗi phần mềm có thể đến từ nhiều nguyên nhân khác nhau, trong đó có cả các nguyên nhân chủ quan và các nguyên nhân khách quan. Dưới đây là chín nguyên nhân chủ yếu gây ra lỗi phần mềm:

- **Định nghĩa các yêu cầu bị lỗi:** Những lỗi trong việc xác định yêu cầu thường nằm ở phía khách hàng. Một số lỗi thường gặp là: định nghĩa sai yêu cầu, lỗi không hoàn chỉnh, thiếu các yêu cầu quan trọng hoặc là quá chú trọng các yêu cầu không thật sự cần thiết.

- **Các lỗi trong giao tiếp giữa khách hàng và nhà phát triển:** Hiểu lầm trong giao tiếp giữa khách hàng và nhà phát triển cũng là nguyên nhân gây lỗi. Những lỗi này thường xuất hiện trong giai đoạn đầu của dự án. Một số lỗi hay gặp phải: hiểu sai chỉ dẫn trong tài liệu yêu cầu, hiểu sai thay đổi khi khách hàng trình bày bằng lời nói và tài liệu, hiểu sai về phản hồi và thiếu quan tâm đến những đề cập của khách hàng.

Giải pháp khắc phục: Cần có ủy ban liên kết giữa khách hàng và nhà cung cấp để tránh lỗi trong giao tiếp. Ủy ban do quản trị dự án đứng đầu và khách hàng phải giới thiệu những người hiểu về mặt nghiệp vụ vào ủy ban đó.

- **Sai lệch có chủ ý với các yêu cầu phần mềm:** Trong một số trường hợp các nhà phát triển cố tình làm sai lệnh các yêu cầu trong tài liệu đặc tả. Nguyên nhân của việc này đến từ các áp lực thời gian, ngân sách, hay cố tình sử dụng lại các mô-đun từ các dự án trước mà chưa phân tích đầy đủ những thay đổi để thích nghi với các yêu cầu mới.

Giải pháp khắc phục: Dựa trên những thống kê để quyết định xem giải pháp như thế nào, sắp xếp ưu tiên xem bỏ được yêu cầu nào hay sử dụng lại được mô-đun nào.

- **Các lỗi thiết kế logic:** Lỗi phần mềm xảy ra trong quá trình các chuyên gia thiết kế hệ thống, các kiến trúc sư hệ thống, kỹ sư phần mềm, các nhà phân tích xây dựng phần mềm theo yêu cầu. Các lỗi điển hình bao gồm:

- Định nghĩa các yêu cầu phần mềm bằng các thuật toán sai
- Quy trình định nghĩa có chứa trình tự lỗi
- Sai sót trong các định nghĩa biên như  $> 3$  hay  $\geq 3$
- Thiếu sót các trạng thái hệ thống phần mềm được yêu cầu

- **Các lỗi lập trình:** Có rất nhiều lý do dẫn đến việc các lập trình viên gây ra các lỗi lập trình. Những lý do này bao gồm: sự hiểu sai các tài liệu thiết kế, ngôn ngữ; sai sót trong ngôn ngữ lập trình; sai sót trong việc áp dụng các công cụ phát triển; sai sót trong lựa chọn dữ liệu...

- **Không tuân thủ theo các tài liệu hướng dẫn và tiêu chuẩn lập trình:** Các lỗi phần mềm có thể đến từ việc không tuân thủ các tài liệu và tiêu chuẩn lập trình của các tổ chức phát triển phần mềm.

- **Thiếu sót trong quá trình kiểm thử:** Lỗi phần mềm có thể đến từ chính quá trình kiểm thử khi mà người kiểm thử để lọt lỗi.

Những lỗi này đến từ các nguyên nhân sau đây:

- Kế hoạch kiểm thử chưa hoàn chỉnh, để sót yêu cầu cần kiểm thử.
- Lỗi trong tài liệu và báo cáo kiểm thử.
- Việc sửa chữa các lỗi được phát hiện không hoàn chỉnh do áp lực thời gian hay do thiếu cẩn thận.

Giải pháp: Lên kế hoạch kiểm thử cụ thể tại giai đoạn đầu của dự án.

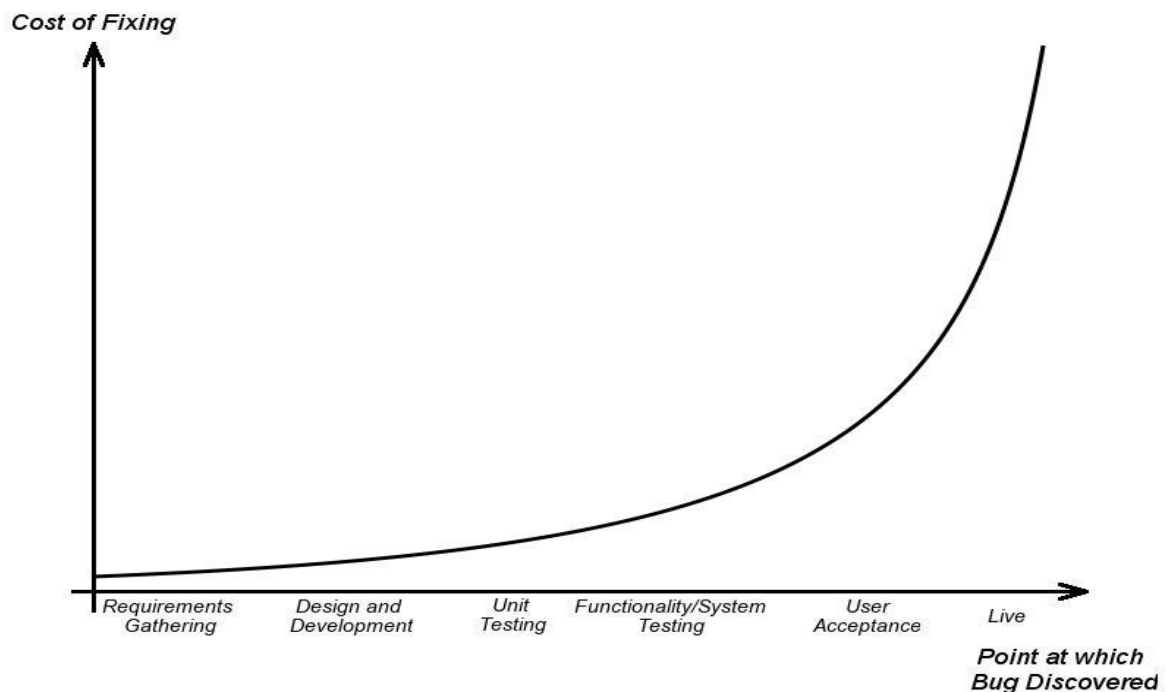
- **Các lỗi thủ tục:** Các thủ tục hướng dẫn cho người sử dụng tại từng bước của tiến trình. Chúng có tầm quan trọng đặc biệt trong các hệ thống phần mềm phức tạp mà các tiến trình được thực bằng nhiều bước, mỗi bước có thể có nhiều kiểu dữ liệu và cho phép kiểm tra các kết quả trung gian. Các lỗi có thể đến từ việc viết các thủ tục.

- **Các lỗi về tài liệu:** Các lỗi về tài liệu là vấn đề của các đội phát triển và bảo trì khi có những sai sót trong các tài liệu liên quan. Những lỗi này có thể là nguyên nhân gây ra lỗi trong giai đoạn phát triển kế tiếp và giai đoạn bảo trì.

### 1.7. Chi phí sửa lỗi phần mềm

Việc kiểm thử và sửa lỗi phần mềm có thể thực hiện trong bất cứ giai đoạn nào của vòng đời phần mềm. Tuy nhiên công việc này nên được thực hiện càng sớm càng tốt vì càng về giai đoạn sau của vòng đời phần mềm, chi phí cho việc tìm và sửa lỗi càng tăng, đặc biệt là đến giai đoạn đã triển khai phần mềm thì chi phí cho sửa lỗi sẽ trở nên rất lớn và ảnh hưởng trực tiếp đến uy tín của tổ chức phát triển phần mềm.

Theo tài liệu của Boehm, chi phí cho việc tìm và sửa lỗi phần mềm sẽ tăng theo hàm mũ trong biểu đồ sau:



Hình 1.1. Sơ đồ: Chi phí cho việc sửa lỗi phần mềm

## 1.8. Tổng kết

Trong chương 1 đã nêu tổng quan về kiểm thử phần mềm. Các kỹ thuật kiểm thử. Trong đó, Kiểm thử hộp trắng xem xét chương trình ở mức độ chi tiết và phù hợp khi kiểm tra các mô đun nhỏ. Tuy nhiên, kiểm thử hộp trắng có thể không đầy đủ vì kiểm thử hết các lệnh không chứng tỏ là chúng ta đã kiểm thử hết các trường hợp có thể. Ngoài ra chúng ta không thể kiểm thử hết các đường đi đối với các vòng lặp lớn.

Kiểm thử hộp đen chú trọng vào việc kiểm tra các quan hệ vào ra và những chức năng giao diện bên ngoài, nó thích hợp hơn cho các hệ thống phần mềm lớn hay các thành phần quan trọng của chúng. Nhưng chỉ sử dụng kiểm thử hộp đen là chưa đủ. Bởi vì, kiểm thử chức năng chỉ dựa trên đặc tả của mô đun nên không thể kiểm thử được các trường hợp không được khai báo trong đặc tả. Ngoài ra, do không phân tích mã nguồn nên không thể biết được mô đun nào của chương trình đã hay chưa được kiểm thử, khi đó phải kiểm thử lại hay bỏ qua những lỗi tiềm ẩn trong gói phần mềm.

Phương pháp kiểm thử hộp trắng và kiểm thử hộp đen là hai phương pháp cơ bản có vai trò rất quan trọng trong quá trình phát triển phần mềm, nếu chúng ta biết kết hợp chúng để bổ sung khiếm khuyết lẫn nhau.

## CHƯƠNG 2. TÌM HIỂU VỀ KIỂM THỬ TỰ ĐỘNG

### 2.1. Tổng quan về kiểm thử tự động

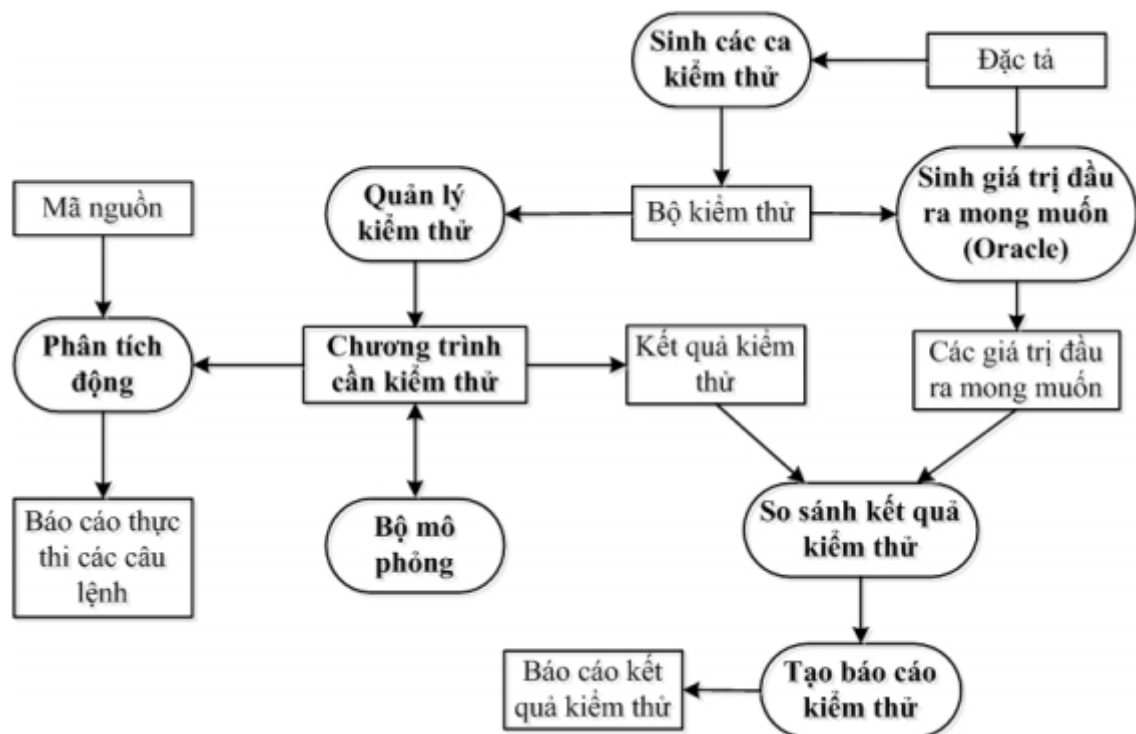
Một thực tế đáng buồn hiện nay là chất lượng của hầu hết các sản phẩm phần mềm rất thấp. Hàng năm, chúng ta phải chịu thiệt hại nhiều tỷ đô la do các lỗi phần mềm gây ra [oST02a, G.95]. Theo thống kê của NIST công bố năm 2002 [oST02b], chất lượng phần mềm thấp đã gây thiệt hại cho kinh tế Mỹ 60 tỷ đô la mỗi năm và tiêu tốn 22 tỷ đô la cho việc phát triển các công cụ nhằm phát hiện các lỗi và kiểm thử tự động. Có hai lý do chính dẫn đến tình trạng này. Thứ nhất, hầu hết các công cụ hiện nay đều tập trung vào việc thực thi tự động các ca kiểm thử (auto-test execution) trong khi vấn đề cốt lõi của kiểm thử là các phương pháp và kỹ thuật sinh các ca kiểm thử vẫn còn thiếu. Thứ hai, các công cụ hiện nay chưa hỗ trợ một cách hiệu quả cho kiểm thử hồi quy (regression testing). Một khi phần mềm bị tiến hóa/thay đổi, chúng ta cần kiểm thử lại sản phẩm. Làm thế nào để sử dụng lại các ca kiểm thử đã có và sinh ra các ca kiểm thử mới một cách hiệu quả đang là một vấn đề mở và chưa có giải pháp thỏa đáng. Hơn nữa, một trong những vấn đề khó nhất của kiểm thử tự động đó là việc sinh các giá trị đầu ra mong đợi tương ứng với các đầu vào của các ca kiểm thử. Đây là một bài toán khó và chưa có giải pháp hiệu quả nhằm giải quyết vấn đề này.

Giải pháp chủ yếu để giải quyết các vấn đề trên là đề xuất các phương pháp và công cụ hỗ trợ tối đa các hoạt động trong quy trình kiểm thử phần mềm. Trong quy trình kiểm thử, chúng ta cần một số công cụ ứng với các pha và các mục tiêu kiểm thử khác nhau. Ví dụ, để tự động hóa chiến lược kiểm thử hộp đen, phương pháp kiểm thử dựa trên mô hình (modelbased testing) đang được biết đến như là một giải pháp tin cậy và hiệu quả [MRA04, KJ02, BFM04]. Với kiểm thử hộp trắng, ứng với mỗi phương pháp khác nhau chúng ta cũng có rất nhiều công cụ hỗ trợ. Ngoài ra, một số công cụ hỗ trợ kiểm thử các tính chất phi chức năng như độ an toàn, bảo mật, hiệu năng và khả năng chịu tải, ... cũng đã được phát triển và sử dụng rộng rãi.

Kiểm thử tự động là quá trình thực hiện một cách tự động các bước trong một kịch bản kiểm thử. Kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử. Mục đích của kiểm thử tự động là giảm thiểu thời gian, công sức và kinh phí, tăng độ tin cậy, tăng tính hiệu quả và giảm sự nhầm lẫn cho người kiểm thử trong quá trình kiểm thử sản phẩm phần mềm.

Kiểm thử tự động sẽ được sử dụng khi dự án không đủ tài nguyên (thời gian, nhân lực và chi phí), phải thực hiện kiểm thử hồi quy khi sản phẩm được sửa đổi hoặc nâng cấp và cần kiểm thử lại các tính năng đã thực hiện tốt trước đó, kiểm tra khả năng vận hành của sản phẩm trong các môi trường đặc biệt (do tốc độ xử lý trung bình ứng với mỗi yêu cầu, xác định khả năng chịu tải tối đa, xác định cấu hình tối thiểu để thực thi hệ thống, kiểm tra các cơ chế an ninh và an toàn, ...).

## 2.2. Kiến trúc của một bộ kiểm thử tự động



Hình 2.1. Kiến trúc chung của một bộ kiểm thử tự động.

## 2.3. Ưu và nhược điểm của kiểm thử tự động

- Các ưu điểm có thể kể đến của kiểm thử tự động là:
  - Kiểm thử chính xác và có thể bao quát thông tin.
  - Theo dõi được chính xác kết quả từng giai đoạn và các báo cáo tổng hợp.
  - Cần ít nhân lực trong quá trình kiểm thử.
  - Hiệu năng của kiểm thử các lớp vượt xa tầm với của kiểm thử thủ công.
- Tuy nhiên không thể không kể đến các nhược điểm của kiểm thử tự động:
  - Tốn chi phí đầu tư lớn cho việc phát triển công cụ kiểm thử tự động.
  - Chi phí cao cho việc chuyển giao công nghệ và đào tạo nhân viên.
  - Tốn chi phí và thời gian cho việc tạo các kịch bản kiểm thử và bảo trì các kịch bản kiểm thử.
  - Giai đoạn chuẩn bị kiểm thử yêu cầu nhiều nhân lực.

Khu vực kiểm thử tự động có thể không bao quát đầy đủ, không áp dụng được trong việc tìm lỗi mới của phần mềm.



## 2.4. Một số công cụ của kiểm thử tự động

### 2.4.1. CFT4Cunit

Nhằm cung cấp một công cụ kiểm thử tự động các đơn vị chương trình (các hàm) viết bằng ngôn ngữ C phục vụ các sinh viên trong việc nghiên cứu và học tập, chúng tôi đã phát triển một công cụ có tên CFT4CUnit (Control Flow Testing for C Unit). Công cụ này tự động hóa các bước trong quy trình kiểm thử dòng điều khiển như đã giới thiệu trong chương 6. Đầu vào của công cụ này là các hàm/đơn vị chương trình viết bằng ngôn ngữ C và độ đo cần kiểm thử. Công cụ sẽ xây dựng đồ thị dòng dữ liệu ứng với độ đo này, hiển thị đồ thị luồng điều khiển một cách trực quan và sinh ra các ca kiểm thử tương ứng. Các ca kiểm thử được sinh ra sẽ được xuất ra một tệp nhằm giúp cho kiểm thử viên thêm giá trị đầu ra mong muốn vào mỗi ca kiểm thử. Khi kiểm thử viên làm việc với mỗi ca kiểm thử, công cụ cho phép làm nổi bật dòng điều khiển của đơn vị chương trình ứng với ca kiểm thử này nhằm trợ giúp trong việc sinh giá trị đầu ra mong muốn một cách chính xác. Cuối cùng, công cụ cho phép thực hiện các ca kiểm thử và tạo ra báo cáo kiểm thử.

```
//tra lai 1 neu a, b, c la do dai 3 canh
//cua mot tam giac. Tra lai -1 neu nguoc lai
int IsTrangle(float a, float b, float c){
    if(a <= 0 || b <= 0 || c <= 0)
        return -1;
    if(a + b <= c || b + c <= a || a + c <= b)
        return -1;
    return 1;
}
```

Hình 2.2. Hàm *IsTrangle* làm đầu vào cho công cụ *CFT4CUnit*.

### 2.4.2. Junit

Công cụ kiểm thử cho các đơn vị chương trình viết bằng Java, Junit, cung cấp một cơ sở hạ tầng chuẩn cho việc thiết lập các bộ kiểm thử. Một khi bộ kiểm thử được thiết lập, nó có thể tự động chạy mỗi khi mã thay đổi. Junit khuyến khích các nhà phát triển viết các kịch bản kiểm thử, chèn các mã kiểm thử vào mã nguồn Java và thực hiện chúng để phát hiện các lỗi bên trong đơn vị chương trình. Khác với các công cụ khác, JUnit không hỗ trợ cơ chế sinh các ca kiểm thử. Hiện nay, JUnit đã được tích hợp trong Eclipse và hỗ trợ rất đặc lực cho quá trình kiểm thử.

### 2.4.3. Quick Test Professional

Quick Test Professional là phần mềm kiểm soát việc kiểm thử tự động các chức năng của các sản phẩm phần mềm cần kiểm thử. Sản phẩm này bao gồm một tập các mô-đun có thể tương tác với nhau nhằm quản lý toàn bộ quy trình kiểm thử phần mềm. Quick Test Professional là một công cụ hỗ trợ kiểm thử hàm (kiểm thử chức năng) và cho phép tiến hành kiểm thử hồi quy một cách tự động.

#### 2.4.4. Load Runner

Load Runner giả lập một môi trường ảo gồm nhiều người dùng thực hiện các giao dịch cùng một lúc nhằm giám sát các thông số xử lý của phần mềm cần kiểm thử. Kết quả thống kê sẽ được lưu lại và cho phép kiểm thử viên thực hiện phân tích nhằm kiểm thử khả năng chịu tải và các yêu cầu phíchức năng khác của sản phẩm. Trong quá trình kiểm thử, Load Runner tự động tạo ra các kịch bản kiểm thử để lưu lại các thao tác người dùng tương tác lên phần mềm. Mỗi kịch bản này còn được xem là hoạt động của một người dùng ảo mà Load Runner giả lập. Ngoài ra, công cụ này còn cho phép tổ chức, điều chỉnh, quản lý và giám sát hoạt động kiểm tra khả năng chịu tải.

#### 2.5. Tổng kết kiểm thử tự động

Kiểm thử tự động đang được quan tâm như là một giải pháp hiệu quả và duy nhất nhằm cải thiện tính chính xác và hiệu quả cũng như giảm kinh phí và rút ngắn thời gian trong quá trình kiểm thử các sản phẩm phần mềm. Đã có nhiều công cụ được phát triển hỗ trợ các mục đích trên. Tùy thuộc vào yêu cầu kiểm thử của từng sản phẩm, các công ty sẽ lựa chọn các công cụ phù hợp. Tuy nhiên, rất khó để tìm được một bộ công cụ đáp ứng tất cả các yêu cầu kiểm thử. Trong nhiều trường hợp, các công ty cần chủ động mở rộng và phát triển thêm các công cụ phục vụ các mục đích cụ thể.

## **CHƯƠNG 3. TỔNG QUAN VỀ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG QUICKTEST PROFESSIONAL**

### **3.1. Giới thiệu về công cụ Quicktest Professional**

Phần mềm HP Quick Test Professional (QTP) là phần mềm kiểm soát việc test tự động những chức năng của một sản phẩm phần mềm khác. Phần mềm QTP là một bộ phận (mô-đun) của hệ thống Mercury Quality Center bao gồm nhiều mô-đun phần mềm phối hợp với nhau để quản lý toàn bộ quy trình đảm bảo chất lượng sản phẩm phần mềm. Trước đây do hãng Mercury phát hành. Sau này, tập đoàn HP đã mua lại toàn bộ hãng Mercury, nên tên gọi của nó là HP Quick Test Professional.

Quick Test Professional là một công cụ kiểm thử (Test Tool) dùng để kiểm tra chức năng (functional test) và cho phép thực hiện kiểm tra hồi quy (regression test) một cách tự động. Đây cũng là công cụ áp dụng phương pháp Keyword – Driven, một kỹ thuật scripting (lập trình trong kiểm tra tự động) hiện đại, cho phép kiểm tra viên bổ sung Test case bằng cách tạo file mô tả cho nó mà không cần phải chỉnh sửa hay bổ sung bất cứ script nào cả. Nó cũng phù hợp trong tình huống chuyển giao công việc mà người mới tiếp nhận chưa có thời gian hoặc không hiểu script vẫn có thể thực hiện kiểm tra phần mềm theo đúng yêu cầu.[9]

### **3.2. Loại phần mềm hỗ trợ**

QTP giúp chúng ta kiểm thử phần mềm theo hướng chức năng trên rất nhiều loại chương trình phần mềm khác nhau. Tuy nhiên Mercury chỉ hỗ trợ sẵn một số loại chương trình thông dụng như:

- Ứng dụng Windows chuẩn/Win32.
- Ứng dụng web theo chuẩn HTML, XML chạy trong trình duyệt Internet Explorer, Netscape hoặc AOL.
- Visual Basic.
- ActiveX.
- QTP hỗ trợ Unicode (UTF-8, UTF-16).

Một số loại chương trình khác đòi hỏi chúng ta phải cài đặt thêm thành phần bổ sung của QTP thì mới thực hiện kiểm tra được. Các loại chương trình đó là:

<b>.NET</b>	<ul style="list-style-type: none"> <li>• NET Framework 1.0, 1.1, 2.0 beta</li> <li>• Các đối tượng chuẩn của .NET và các đối tượng khác thừa kế từ các đối tượng chuẩn.</li> </ul>
<b>Java</b>	<ul style="list-style-type: none"> <li>• Sun JDK 1.1 – 1.4.2</li> <li>• IBM JDK 1.2 – 1.4</li> </ul>
<b>Oracle</b>	<ul style="list-style-type: none"> <li>• Oracle Applications 11.5.7, 11.5.8, 11.5.9</li> </ul>
<b>People Soft</b>	<ul style="list-style-type: none"> <li>• PeopleSoft Enterprise 8.0 – 8.8</li> </ul>
<b>SAP</b>	<ul style="list-style-type: none"> <li>• SAP GUI HMTL 4.6D, 6.10, 6.20</li> <li>• SAP Workplace 2.11</li> <li>• SAP Enterprise Portal 5.0</li> </ul>
<b>Siebel</b>	<ul style="list-style-type: none"> <li>• Siebel 7.0, 7.5, 7.7</li> </ul>
<b>Terminal Emulators</b>	<ul style="list-style-type: none"> <li>• Attachmate EXTRA! 6.7, 7.1</li> <li>• Attachmate EXTRA! Terminal Viewer 3.1</li> <li>Java sessions</li> <li>• IBM Personal Communications</li> <li>• ...</li> </ul>

*Bảng 3.1. Các phần mềm cần cài đặt bổ sung*

### 3.3. Đặc Điểm

- Dễ sử dụng, bảo trì, tạo test script nhanh. Cung cấp dữ liệu kiểm tra rõ ràng và dễ hiểu.
- Kiểm tra phiên bản mới của ứng dụng với rất ít sự thay đổi. Ví dụ khi ứng dụng thay đổi nút tên “Login” thành “Đăng nhập” thì chỉ cần cập nhật lại Object Repository để QTP nhận ra sự thay đổi đó mà không cần thay đổi bất cứ test script nào.
- Hỗ trợ làm việc theo nhóm thông qua sự chia sẻ thư viện, thống nhất quản lý Object Repository.
- Thực tế cho thấy, QTP thực hiện KTTĐ trên nhiều trình duyệt cùng lúc tốt hơn những TT khác.

- Với chức năng Recovery Scenarios, QTP cho phép xử lý những sự kiện hoặc lỗi không thể đoán trước có thể làm script bị dừng trong khi đang chạy.
- QTP có khả năng hiểu test script của Mercury Winrunner (một công cụ kiểm tra khác của Mercury).
- Ưu điểm Lưu trữ độc lập của quá trình kiểm tra:

Lưu trữ các thành phần test thành từng phần độc lập. Với số lượng tập tin không nhiều, chúng ta có thể hoàn toàn lưu trữ một cách dễ dàng. Nhìn chung, chúng ta có 4 loại tập tin cần phải lưu:

- Test script: lưu trữ những đoạn script dưới dạng \*.vbs
- Repository: lưu trữ những đối tượng test của ứng dụng dưới dạng \*.tsr
- Paramater: lưu trữ những thông số test dưới dạng \*.xls.
- Settings: lưu trữ cấu hình cho QTP hoặc những cấu hình đặc thù dưới dạng \*.vbs.

### **3.4. Các thành phần quan trọng trong QuickTest Professional**

#### **3.4.1. Action**

Giống như thủ tục hay hàm trong các ngôn ngữ lập trình khác, Action ghi lại các bước thực hiện KTTĐ và nó có thể được sử dụng lại nhiều lần. Trong một test script có thể có nhiều Action.

#### **3.4.2. Data Table**

Nơi lưu dữ liệu phục vụ cho KTTĐ. Một test script sẽ có một Data Table được dùng chung cho tất cả các Action. Bên cạnh đó mỗi Action cũng có một Data Table cho riêng mình.

#### **3.4.3. Object Repository (OR)**

Cấu trúc theo dạng cây, mô tả các đối tượng trong phần mềm được kiểm tra. Đây được xem là cầu nối để Test script tương tác với phần mềm được kiểm tra.

Khi ra lệnh cho QTP ghi lại thao tác người dùng lên PM thì trong OR sẽ tự động phát sinh thành phần đại diện cho những đối tượng trên PM vừa được thao tác. OR có thể tổ chức thành 2 loại, một loại dùng chung trong nhiều test script, loại khác dùng theo từng Action.

Để xem OR, chọn Menu Tools -> Object Repository.

#### **3.4.4. Checkpoint**

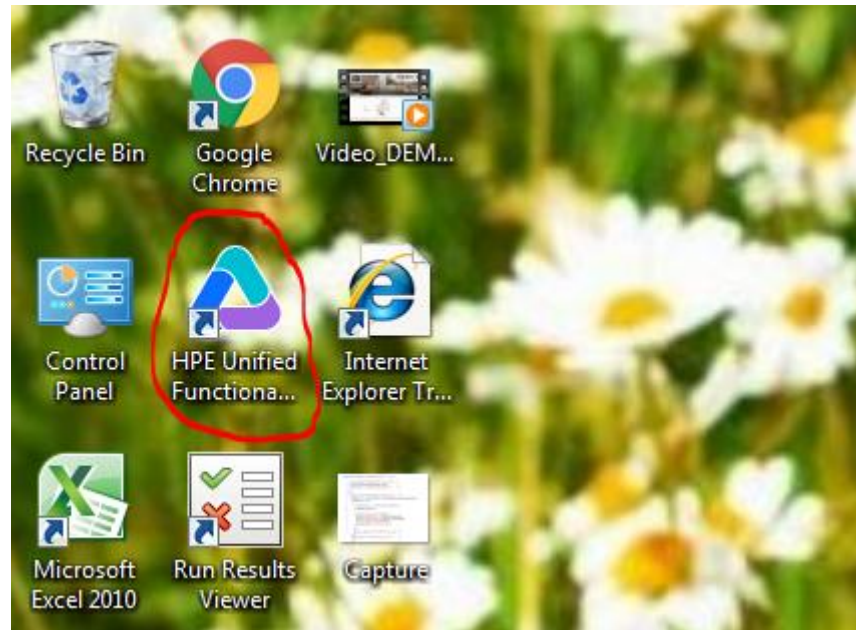
Có thể hiểu là nơi kiểm tra trong test script, khi chạy nó sẽ thực hiện so sánh kết quả thực tế khi kiểm tra phần mềm với kết quả mong đợi. Sau khi tiến hành so sánh QTP sẽ tự động ghi lại kết quả vào Test Result (nơi lưu kết quả khi chạy test script).

### **3.5. Ngôn ngữ sử dụng viết Script**

QTP sử dụng ngôn ngữ VBScript để viết test script. Đây là ngôn ngữ dễ học, rất giống ngôn ngữ VBA. Chế độ Expert View của QTP là chế độ soạn thảo dành cho VBScript. Ngoài việc dùng VBScript để tương tác với phần mềm được kiểm tra, QTP còn có khả năng cấu hình hệ thống bằng ngôn ngữ Windows Script.[3]

### 3.6. Cách sử dụng Quick test professional

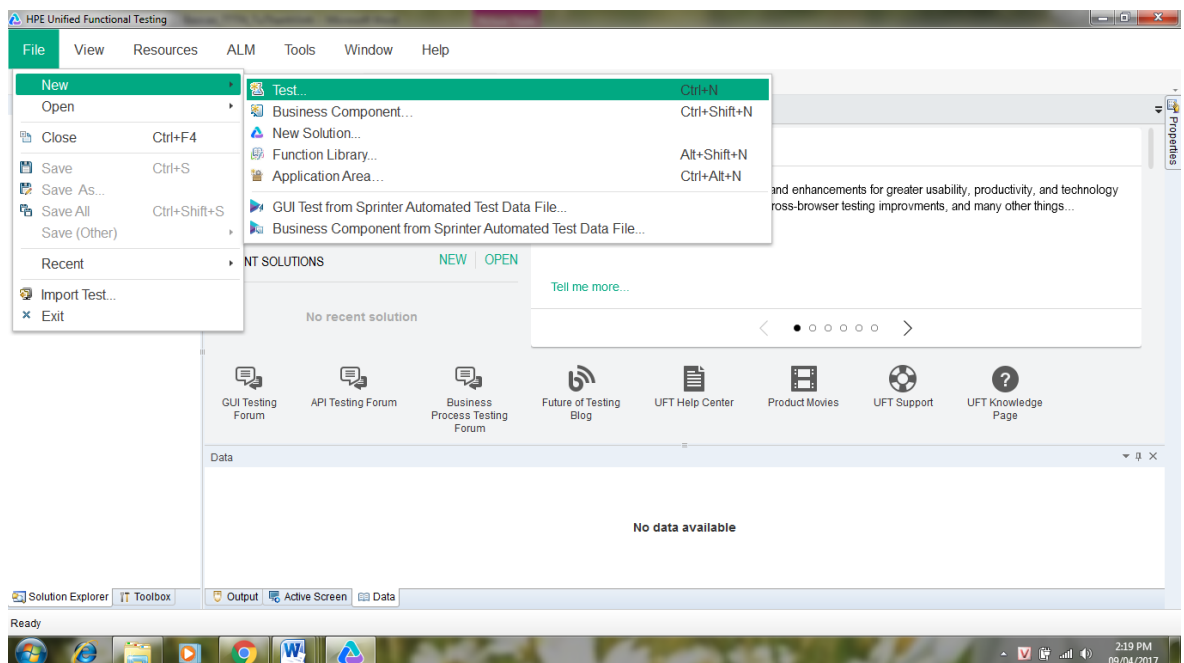
a. Mở chương trình QTP: click đúp vào biểu tượng QTP trên màn hình.



Hình 3.1. Biểu tượng QTP trên màn hình.

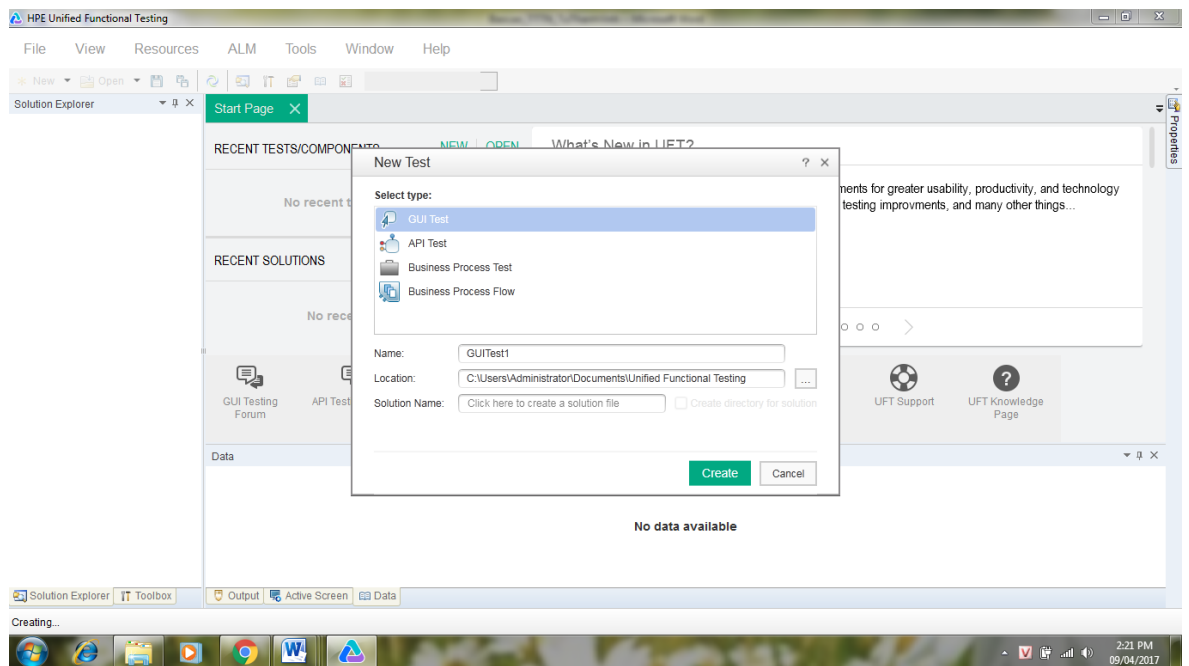
b. Ghi và phát lại (Record and Run)

Bước 1: Click File → New → Test



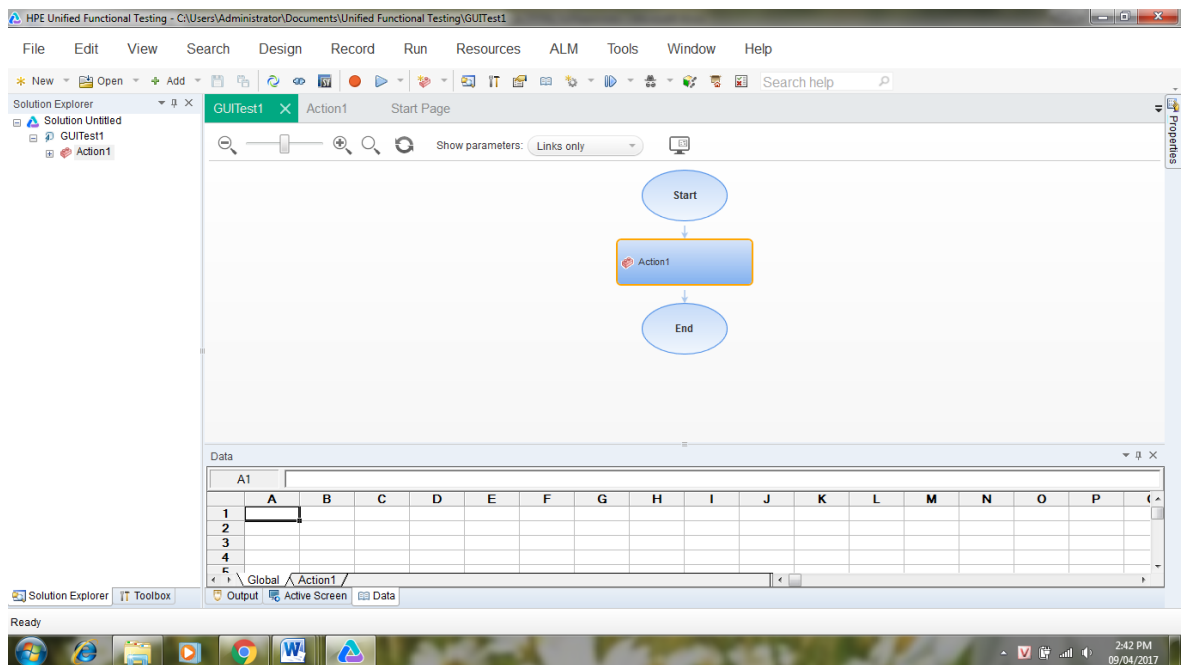
Hình 3.2. Tạo mới một test script.

*Bước 2: Click GUI Testing→Creat để chọn kiểu test*



*Hình 3.3. Chọn loại kiểm thử.*

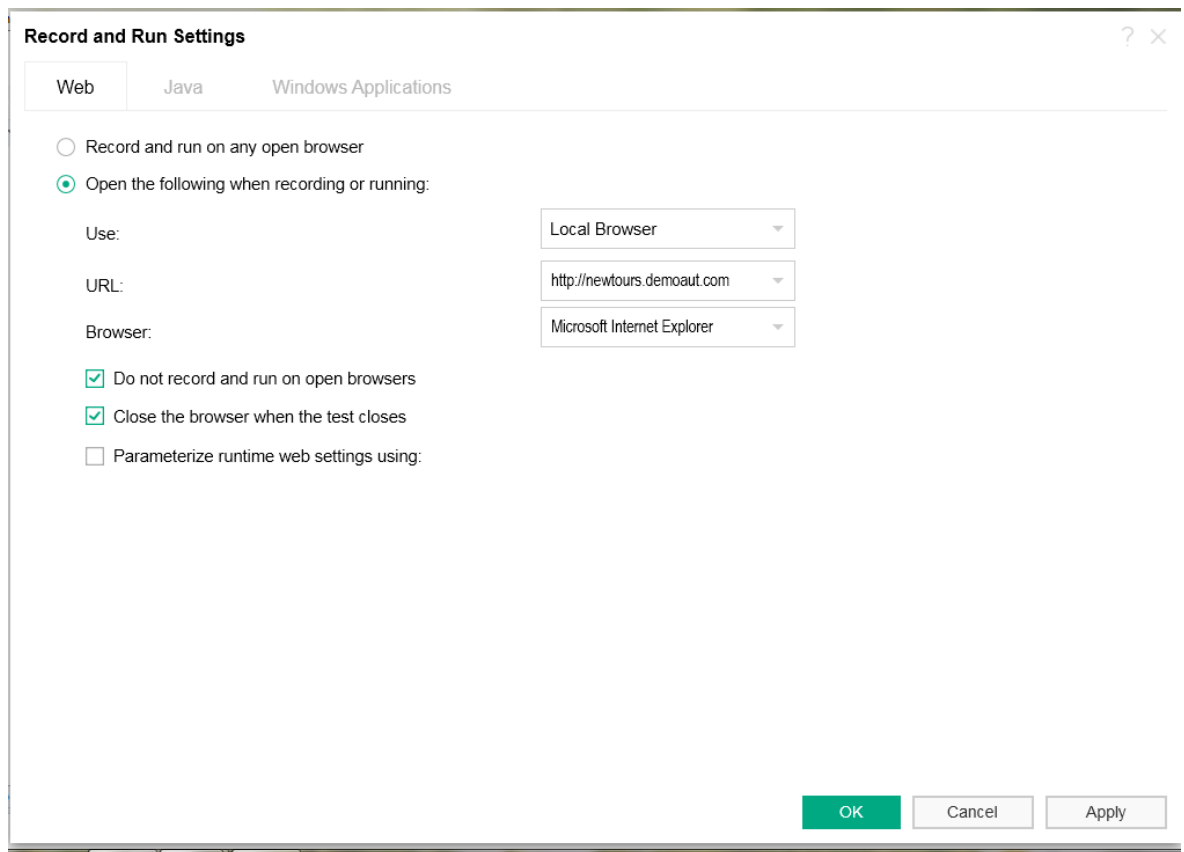
*Bước 3:Hiện thị màn hình GUItest.*



*Hình 3.4.. Tạo Record.*

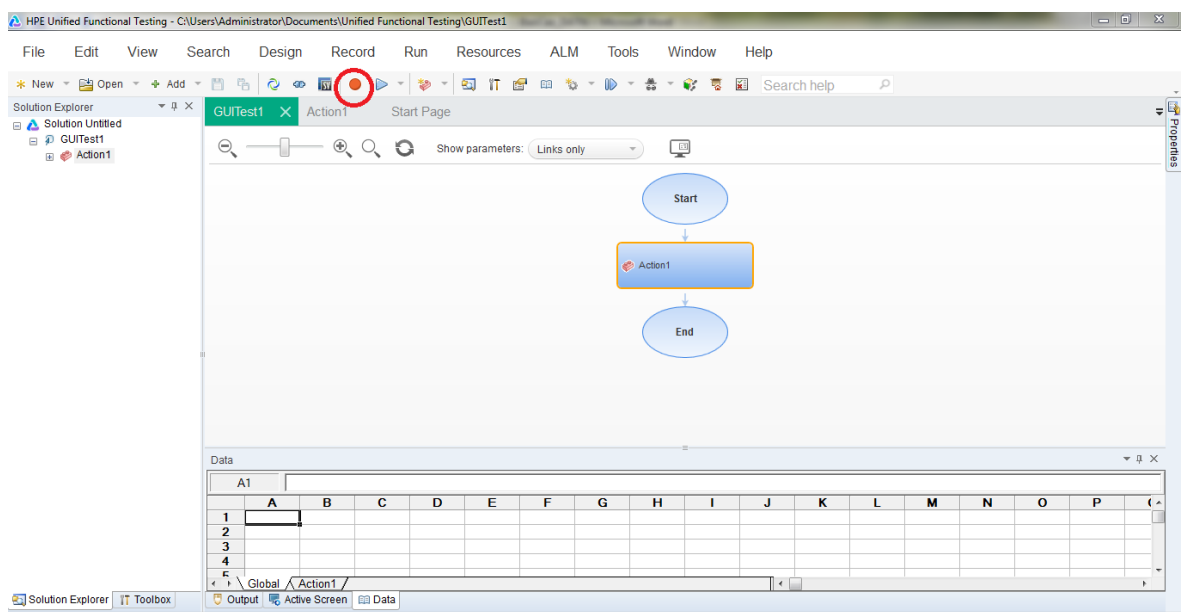


### Bước 5: Hiển thị cửa sổ tùy chọn



Hình 3.5. Tùy chọn ghi và chạy test script

### Bước 6: Click nút Record để thực hiện ghi



Hình 3.6. Click nút record để thực hiện ghi

Bước 7: Sau khi đã thực hiện record thành công, ta click Run để thực hiện việc chạy lại và cho ra kết quả.

### **3.7. Tổng kết về công cụ Quick test professional**

- Trong chương này tập trung vào tìm hiểu chi tiết công cụ kiểm thử chức năng QuickTest Professional, gồm: Loại phần mềm hỗ trợ, đặc điểm về công cụ, Các thành phần quan trọng và ngôn ngữ sử dụng để viết script.
- Tải và cài đặt công cụ. [4]
- Cách sử dụng công cụ Quicktest professional.

## CHƯƠNG 4. ỨNG DỤNG

### 4.1. Giới thiệu ứng dụng web



Hình 4.1. Giao diện Website

Website Quản lý bán hàng là website được viết trên nền tảng ngôn ngữ chuẩn C#. Là Website để bán hàng và giới thiệu sản phẩm điện thoại di động, Cho phép khách hàng tìm kiếm và mua sản phẩm.

### 4.2. Phân tích thiết kế hệ thống

#### 4.2.1. Đặc tả hệ thống

Qua việc phân tích và thiết kế hệ thống website, ta có thể đưa ra được bài toán thử nghiệm như sau:

Đây là một website bán và giới thiệu các sản phẩm về điện thoại di động đến khách hàng với các hình ảnh chi tiết và giá cả chính xác. Vì thế, website sẽ có những chức năng cơ bản sau:

- **Dành cho khách hàng:**
  - Hiển thị danh sách các mặt hàng của shop để xem và lựa chọn
  - Tìm kiếm các sản phẩm cần tìm
  - Sau khi khách hàng chọn và đặt hàng hàng trực tiếp thì phải hiện lên đơn hàng để khách hàng có thể nhập thông tin mua hàng và xem hoá đơn mua hàng.
- **Dành cho người quản trị**
  - Quản lý sản phẩm: thêm, sửa, xóa các sản phẩm, loại sản phẩm, tin tức, giá cả, ...
  - Quản lý việc nhập xuất hóa đơn khi có đơn hàng, thống kê và báo cáo.

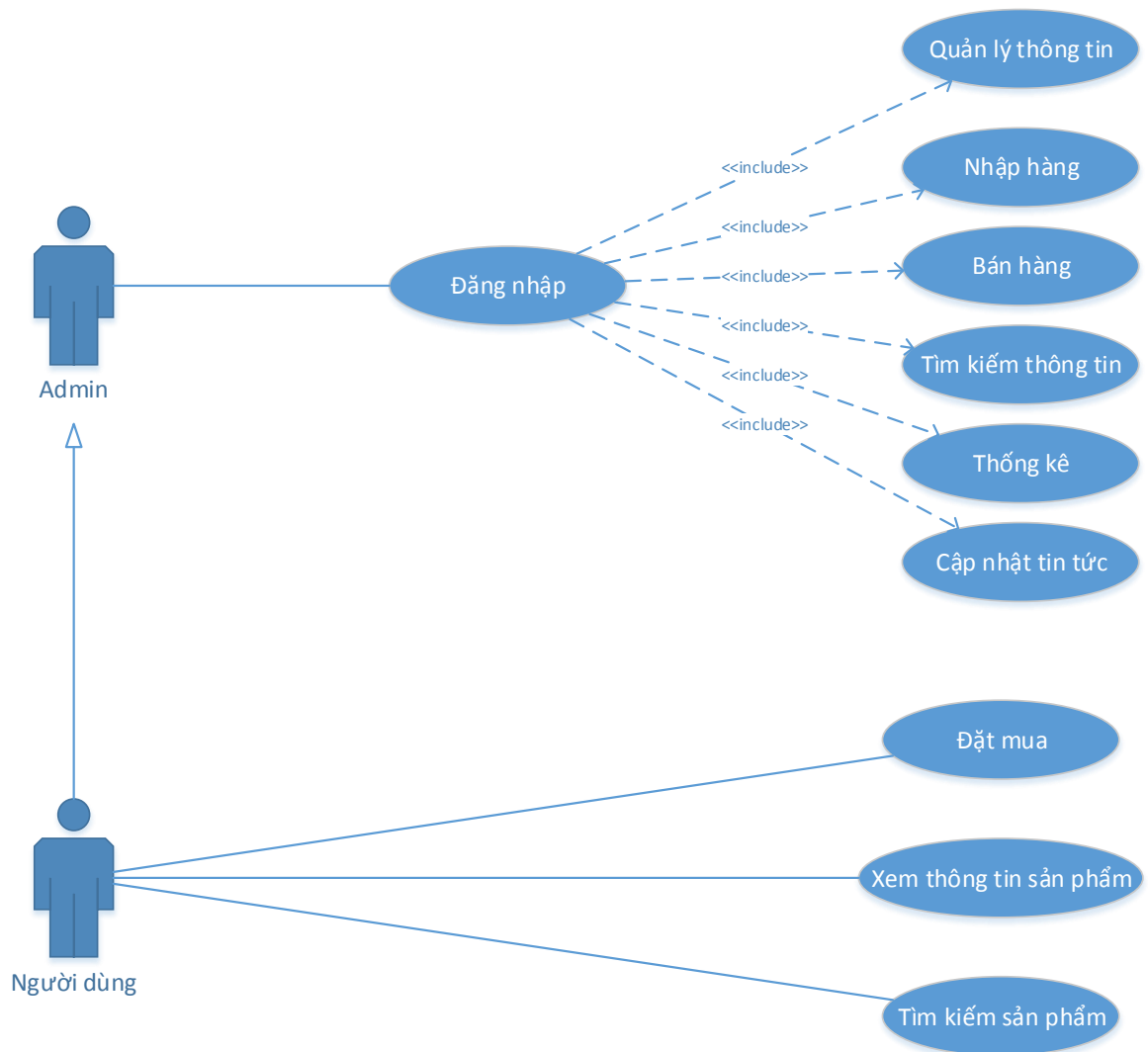
#### 4.2.2. Thiết kế hệ thống

##### a. Danh sách các Usecase

STT	Tên Usecase	Ý nghĩa
1	Đặt mua	UC này mô tả chức năng đặt mua sản phẩm tại website của người dùng
2	Xem thông tin sản phẩm	UC này mô tả chức năng xem thông tin sản phẩm có trên website của người dùng
3	Tìm kiếm sản phẩm	UC này mô tả chức năng tìm kiếm các sản phẩm trên website của người dùng
4	Quản lý thông tin	UC này mô tả chức năng quản lý các thông tin như quản lý người dùng, sản phẩm, nhà cung cấp của Admin
5	Nhập hàng	UC này mô tả chức năng nhập hàng của Admin
6	Bán hàng	UC này mô tả chức năng bán hàng trực tuyến của Admin
7	Tìm kiếm thông tin	UC này mô tả chức năng tìm kiếm các thông tin như: sản phẩm, khách hàng, nhà cung cấp, hãng sản xuất của Admin
8	Thống kê	UC này mô tả chức năng thống kê theo các tiêu chí khác nhau của Admin
9	Cập nhật tin tức	UC này mô tả chức năng admin cập nhật những tin tức mới nhất theo xu hướng thị trường nhằm đáp ứng nhu cầu thiết yếu của khách hàng.
10	Đăng nhập	UC này mô tả chức năng admin đăng nhập vào trang chủ admin.

*Bảng 4.1. Danh sách các Usecase*

b. Biểu đồ Usecase tổng quát



Hình 4.2. Biểu đồ usecase tổng quát

### **4.3. Kế hoạch kiểm thử**

#### **4.3.1. Mục đích**

- Cung cấp thông tin về quy trình kiểm thử sẽ xảy ra trong website.
- Liệt kê những yêu cầu kiểm thử.
- Xác định những phương pháp, chiến lược kiểm thử nên sử dụng.
- Giúp cho mỗi thành viên trong đội kiểm thử có hướng đi đúng.
- Gán các trách nhiệm rõ ràng cụ thể cho mỗi thành viên đội kiểm thử.
- Xác định lịch biểu làm việc rõ ràng cho quá trình kiểm thử.
- Giúp cho tổ dự án có phương pháp phòng tránh các rủi ro có thể xảy ra trong quá trình kiểm thử.
- Chuẩn bị được các công cụ cần thiết cho quá trình kiểm thử.

#### **4.3.2. Phạm vi kiểm thử**

- Các giai đoạn test: Kiểm thử đơn vị, kiểm thử hệ thống.
- Loại test: Kiểm thử chức năng, kiểm thử giao diện.
- Tính năng sẽ test: Kiểm thử chức năng Đăng Nhập trong website.

#### **4.3.3. Công cụ, môi trường test**

- Công cụ thực hiện kiểm thử: Quick test professional
- Phần mềm:
  - Microsoft Windows 7 hoặc cao hơn.
  - Xampp
  - Trình duyệt web Internet Explorer.

#### 4.3.4. Kế hoạch test

<b>Công việc</b>	<b>Số ngày làm</b>	<b>Ngày bắt đầu</b>	<b>Ngày kết thúc</b>
<b>Đọc tài liệu đặc tả</b>	3 days?	14/03/2017	16/03/2017
<b>Lập kế hoạch kiểm thử</b>	3 days?	17/3/2017	19/03/2017
<b>Kiểm thử đơn vị</b>	14 days	21/03/2017	5/4/2017
<b>Phân tích, thiết kế kiểm thử</b>	8 days?	21/03/2017	29/03/2017
<b>Thực hiện kiểm thử đơn vị</b>	6 days?	30/03/2017	5/4/2017
<b>Kiểm thử hệ thống</b>	14 days?	6/4/2017	22/04/2017
<b>Phân tích, thiết kế các ca kiểm thử hệ thống</b>	8 days?	6/4/2017	14/04/2017
<b>Thực hiện kiểm thử hệ thống</b>	6 days?	15/04/2017	22/04/2017
<b>Ghi nhận và đánh giá kết quả kiểm thử</b>	3 days?	23/04/2017	26/04/2017
<b>Lập báo cáo kiểm thử</b>	3 days?	27/04/2017	29/04/2017

*Bảng 4.2. Kế hoạch test*

#### 4.3.5. Nhân sự và trách nhiệm test

<b>Công việc</b>	<b>Số ngày làm</b>	<b>Ngày bắt đầu</b>	<b>Ngày kết thúc</b>	<b>Người thực hiện</b>
<b>Đọc tài liệu đặc tả</b>	3 days?	14/03/2017	16/03/2017	Đặng Thị Nhỏ
<b>Lập kế hoạch kiểm thử</b>	3 days?	17/3/2017	19/03/2017	Đặng Thị Nhỏ
<b>Kiểm thử đơn vị</b>	14 days	21/03/2017	5/4/2017	Đặng Thị Nhỏ
<b>Phân tích, thiết kế kiểm thử</b>	8 days?	21/03/2017	29/03/2017	Đặng Thị Nhỏ
<b>Thực hiện kiểm thử đơn vị</b>	6 days?	30/03/2017	5/4/2017	Đặng Thị Nhỏ
<b>Kiểm thử hệ thống</b>	14 days?	6/4/2017	22/04/2017	Đặng Thị Nhỏ
<b>Phân tích, thiết kế các ca kiểm thử hệ thống</b>	8 days?	6/4/2017	14/04/2017	Đặng Thị Nhỏ
<b>Thực hiện kiểm thử hệ thống</b>	6 days?	15/04/2017	22/04/2017	Đặng Thị Nhỏ
<b>Ghi nhận và đánh giá kết quả kiểm thử</b>	3 days?	23/04/2017	26/04/2017	Đặng Thị Nhỏ
<b>Lập báo cáo kiểm thử</b>	3 days?	27/04/2017	29/04/2017	Đặng Thị Nhỏ

*Bảng 4.3. Nhân sự và trách nhiệm test*

#### 4.3.6. Điều kiện bắt đầu test, dừng test và kết thúc test.

##### a. Điều kiện bắt đầu test

- Phần mềm cần kiểm thử phải có đúng thời hạn
- Phần mềm không có quá nhiều lỗi
- Đầy đủ công cụ để kiểm thử
- Có bản test case để thực hiện

##### b. Điều kiện dừng test

- Trong quá trình test, phát hiện có quá nhiều test case sai, hoặc test case không phù hợp với những mong muốn test của tester.
- Các test case không phủ hết tất cả các trường hợp xảy ra của phần mềm, tạo nên những lỗi tiềm ẩn.
- Khi trong test case có nhiều lỗi, không đảm bảo được việc test chương trình cho ra kết quả chính xác.



- Nguồn nhân lực không đảm bảo để thực hiện kiểm thử
  - Thiếu chi phí cho quá trình kiểm thử.
- c. Điều kiện kết thúc test
- Các yêu cầu phần mềm được kiểm thử dưới môi trường xác định.
  - Đã sửa các lỗi tìm ra được trong quá trình kiểm thử
  - Các test case chưa thực hiện trong tổng các test case  $< 5\%$
  - Phần mềm được chuyển giao cho khách hàng sử dụng.

#### 4.4. Quy trình kiểm thử

##### 4.4.1. Kiểm thử đơn vị

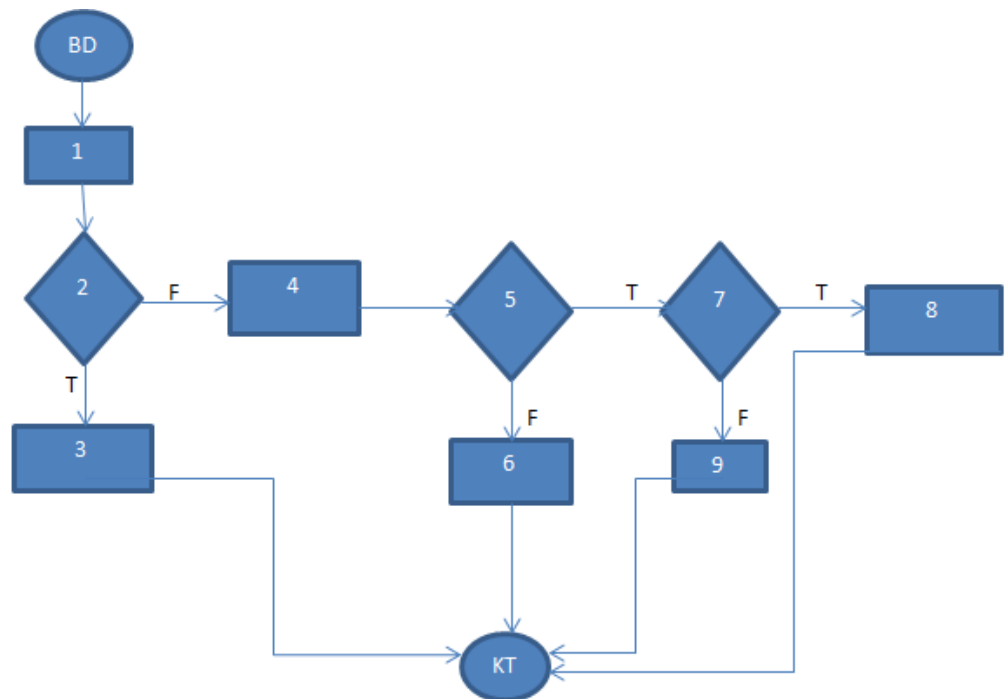
###### 4.4.1.1. Chức năng đăng nhập

- a. Phân tích thiết kế kiểm thử

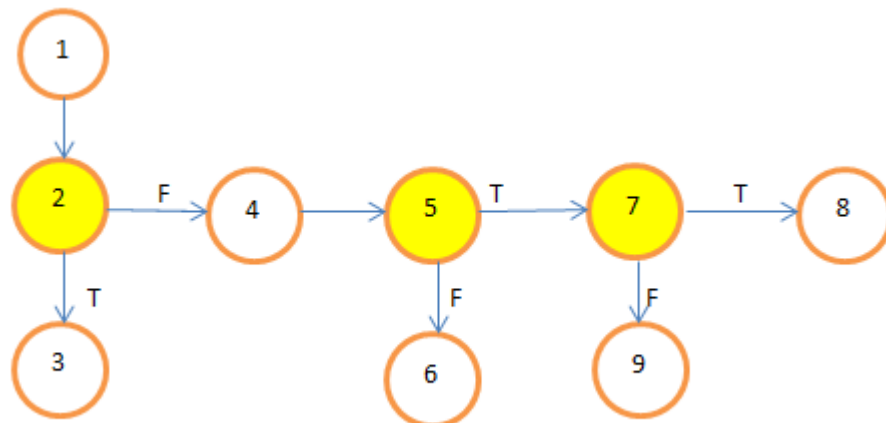
```
protected void btnLogin_Click(object sender, EventArgs e)
{
    string userName = Request.Form["txtUsername"]; } 1

    if (nV.CheckTaiKhoan(userName) == false) } 2
    {txtErrorr.Text = "Tên đăng nhập không tồn tại";} } 3
    else
    {string password = Request.Form["txtPassword"];
    Entities.NhanVienEn nhanVien = nV.GetTaiKhoan(userName,password); } 4
    if (nhanVien != null) } 5
    {
        int maNguoiDung = nhanVien.MaLoaiNguoiDung;
        if (maNguoiDung == 1) } 7
        {
            Session["User_Admin"] = nhanVien;
            Session["MaNhanVien"] = nhanVien.MaNhanVien;
            Session["TenDangNhap"] = nhanVien.TenDangNhap;
            Session["HoTen"] = nhanVien.HoTen;
            Response.Redirect("IndexAdmin.aspx"); } 8
        }else
        {Response.Redirect("NVGIAOHANG.aspx"); } 9
        }
    else
    {txtErrorr.Text = "Mật khẩu không chính xác"; } 6
    }
}
```

b. Lưu đồ thuật giải.



c. Đồ thị dòng

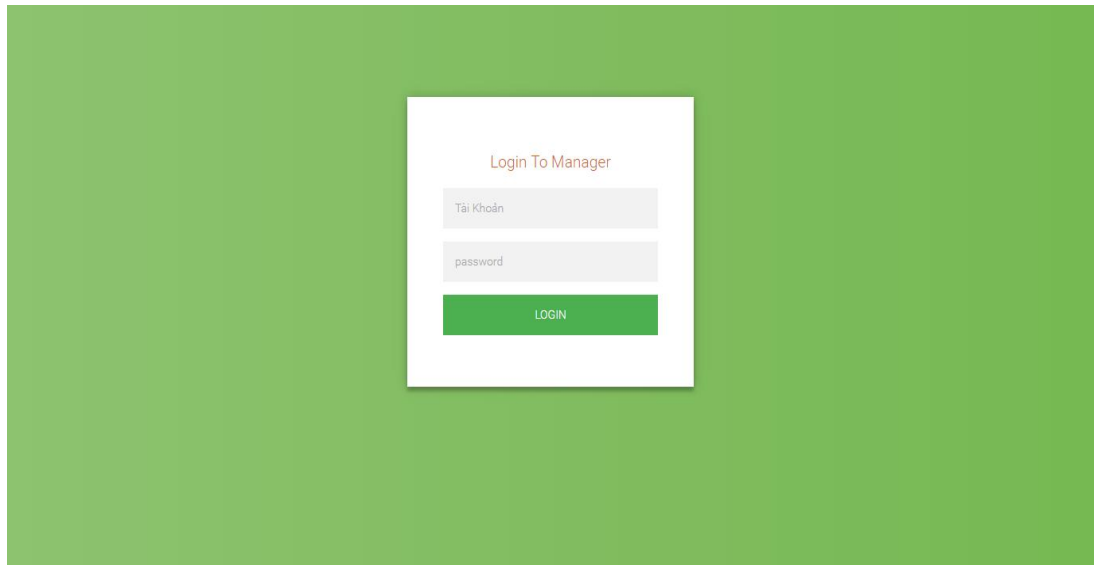


d. Độ phức tạp của chu trình

$C=4$

- 1,2,3
- 1,2,4,5,6
- 1,2,4,5,7,9
- 1,2,3,4,7,8

- e. Ứng dụng kiểm thử chức năng đăng nhập hệ thống
- Màn hình đăng nhập gồm các thành phần:
- Textbox tài khoản và password cho phép admin nhập dữ liệu
  - Button LOGIN cho phép admin click chọn.



*Hình 4.3. Giao diện đăng nhập*

- f. Xây dựng các testcase kiểm thử chức năng đăng nhập
- Mô tả điều kiện.
    - Tất cả các trường đều là bắt buộc.
    - Tên đăng nhập phải lớn hơn hoặc bằng 6 ký tự.
    - Password phải lớn hơn hoặc bằng 5 ký tự.
  - Xây dựng testcase sử dụng bảng quyết định (Decision Tables)[6]
- Đúng: D
- Sai : S
- Không nhập: 0

		Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8	Rule 9
Điều kiện	Tên đăng nhập	D	S	S	D	D	S	0	0	0
	Mật khẩu	S	D	S	D	0	0	D	S	0
Kết quả	Thành công				X					
	Không thành công	X	X	X		X	X	X	X	X

g. Các trường hợp kiểm thử.

Dựa vào kỹ thuật kiểm thử hộp đen, sử dụng bảng quyết định và đồ thị dòng ta có các trường hợp kiểm thử.

Đầu vào	Đầu ra mong đợi
Tài khoản: “phonghd” Mật khẩu: “12346”	Thông báo”Mật khẩu không chính xác”.
Tài khoản: “dangnho” Mật khẩu: “123456”	Thông báo”Tên đăng nhập không tồn tại”.
Tài khoản: “dangnho” Mật khẩu: “12346”	Thông báo”Tên đăng nhập và mật khẩu không đúng, vui lòng nhập lại”.
Tài khoản: “phonghd” Mật khẩu: “123456”	Hiện thị trang web của admin
Tài khoản: “phonghd” Và để trống mật khẩu	Thông báo “Bạn chưa nhập mật khẩu”.
Tài khoản: “dangnho” Và để trống mật khẩu	Thông báo “Tên đăng nhập không tồn tại”
Để trống Tài khoản và Mật khẩu: “123456”	Thông báo “Bạn chưa nhập tên đăng nhập”.
Để trống Tài khoản và Mật khẩu: “12346”	Thông báo “Bạn chưa nhập tên đăng nhập”.
Để trống Tài khoản và mật khẩu	Thông báo “Bạn chưa nhập tên đăng nhập và mật khẩu”.

*Bảng 4.4. Các trường hợp kiểm thử chức năng đăng nhập*

#### 4.4.1.2. Chức năng tìm kiếm

##### a. Ứng dụng kiểm thử chức năng tìm kiếm của hệ thống

Mã SP	Tên Sản Phẩm	Giá	Số Lượng	Tình Trạng	Màn hình	Ram	Bộ Nhớ Trong	Hệ Điều Hành	Chip Xử Lý	Xem	Sửa	Xóa
8	Apple iPhone 7 128GB	19000000	1000	Mới	Cảm ứng điện dung LED-backlit IPS LCD, 16 triệu màu	2 GB RAM	128 GB	iOS	Apple A10 Fusion APL1W24	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>
11	iPhone 6s Plus 64GB	19990000	200	Mới	LED-backlit IPS LCD, 5.5", Retina HD	2 GB	64 GB	iOS 9	Apple A9 2 nhân 64-bit	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>
12	iPhone SE 64GB	11990000	300	Mới	IPS LCD, 4", Retina	2 GB	64 GB	iOS 9	Apple A9 2 nhân 64-bit	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>
13	Microsoft Lumia 640 XL	2490000	500	Mới	5.7 inch (720 x 1280 pixels)	1GB	8 GB	Windows Phone 8.1	Qualcomm® Snapdragon™ 400-Quad-core 1.2 GHz	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>
14	Nokia Lumia 730	1990000	2000	Mới	4.7 inch (720 x 1280 pixels)	1 GB	8 GB	Windows Phone 8.1	Qualcomm Snapdragon 400+ Quad-core 1.2 GHz	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>
15	Microsoft Lumia 950XL	6990000	200	Mới	5.7 inch (1440 x 2560 pixels)	3 GB	32 GB	Windows 10	Qualcomm Snapdragon 810+Octa-core	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>
16	Galaxy S7 Edge Blue	18490000	300	Mới	5.5 inch (1440 x 2560 pixels)	4 GB	32 GB	Android 6.0 (Marshmallow)	Octa-core-Mali-T880 MP12	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>
17	iPhone 5s 16GB	6999000	100	Mới	4 inch (640 x 1136 pixels)	1GB	16GB	iOS 9	Apple A7-Dual-core 1.3Ghz	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>
18	Samsung Galaxy Note 5	12490000	100	Mới	Super AMOLED, 5.7", Quad HD	4 GB	32 GB	Android 6.0 (Marshmallow)	Exynos 7420 8 nhân 64-bit	<a href="#">Xem</a>	<a href="#">Sửa</a>	<a href="#">Xóa</a>

Hình 4.4. Giao diện chức năng tìm kiếm

##### b. Xây dựng các testcase kiểm thử chức năng tìm kiếm

###### ➤ Mô tả điều kiện.

- Khi nhập vào từ khóa đúng hiển thị sản phẩm cần tìm.
- Khi nhập vào từ khóa sai, hiển thị thông báo “Sản phẩm bạn cần tìm không có”
- Khi không nhập từ khóa tìm kiếm, hiển thị tất cả sản phẩm có trong trang web.

###### ➤ Xây dựng testcase sử dụng bảng quyết định (Decision Tables)

- Từ khóa đúng: D
- Từ khóa sai: S
- Không nhập từ khóa: 0

		Rule1	Rule 2	Rule 3
Điều kiện	Từ khóa	D	S	0
Kết quả	Hiển thị Danh sách sản phẩm cần tìm	x		
	Hiển thị thông báo lỗi		x	
	Hiển thị tất cả sản phẩm có trong trang web			x

c. Các trường hợp kiểm thử

Sử dụng kiểm thử hộp đen, sử dụng bảng quyết định ta có các trường hợp kiểm thử.

Đầu vào	Đầu ra mong đợi
Từ khóa: “iphone”	Hiện thị danh sách sản phẩm có chứa từ khóa “iphone”
Từ khóa: “abcdefg”	Thông báo:”Sản phẩm bạn cần tìm không có”
Không nhập từ khóa	Hiện thị tất cả sản phẩm có trong tranh web.

*Bảng 4.5. Các trường hợp kiểm thử chức năng tìm kiếm*

#### 4.4.2. Kiểm thử hệ thống

##### 4.4.2.1. Phân tích thiết kế các ca kiểm thử

###### a. Chức năng đăng nhập

Mã ca KT	Tiêu đề ca KT	Các bước thực hiện	Kết quả mong muốn	Người viết
DN_001	Kiểm tra trường hợp nhập đúng tên tài khoản và nhập sai mật khẩu.	1.Nhập thông tin: Tên đăng nhập: phonghd Mật khẩu: 12345 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Mật khẩu không chính xác".	Đặng Thị Nhỏ
DN_002	Kiểm tra trường hợp nhập sai tên tài khoản và nhập đúng mật khẩu.	1.Nhập thông tin: Tên đăng nhập: dangnho Mật khẩu: 123456 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Đặng Thị Nhỏ
DN_003	Kiểm tra trường hợp nhập sai tài khoản và sai mật khẩu	1.Nhập thông tin: Tên đăng nhập: dangnho Mật khẩu: 12345 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Đặng Thị Nhỏ
DN_004	Kiểm tra trường hợp nhập đúng tài khoản và mật khẩu	1.Nhập thông tin: Tên đăng nhập: phonghd Mật khẩu: 123456 2.Click button: Login	Đăng nhập thành công	Đặng Thị Nhỏ
DN_005	Kiểm tra trường hợp nhập đúng tài khoản và bỏ trống mật khẩu	1.Nhập thông tin: Tên đăng nhập: phonghd Mật khẩu: bỏ trống 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Bạn chưa nhập mật khẩu".	Đặng Thị Nhỏ
DN_006	Kiểm tra trường hợp nhập sai tài khoản và bỏ trống mật khẩu	1.Nhập 3 lần liên tiếp thông tin: Tên đăng nhập: dangnho Mật khẩu: bỏ trống 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Đặng Thị Nhỏ
DN_007	Kiểm tra trường hợp bỏ trống tên tài khoản và mật khẩu đúng	1.Nhập thông tin: Tên đăng nhập: bỏ trống Mật khẩu: 123456 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Bạn chưa nhập tài khoản".	Đặng Thị Nhỏ
DN_008	Kiểm tra trường hợp bỏ trống tên tài khoản và mật khẩu sai	1.Nhập thông tin: Tên đăng nhập: bỏ trống Mật khẩu: 12345 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Bạn chưa nhập tài khoản".	Đặng Thị Nhỏ



DN_009	Đăng nhập không thành công khi bỏ trống tài khoản và mật khẩu	1.Nhập thông tin: Tên đăng nhập: Bỏ trống Mật khẩu: Bỏ trống 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Bạn chưa nhập tài khoản và mật khẩu".	Đặng Thị Nhỏ
DN_010	Kiểm tra trường hợp nhập tài khoản có khoảng trắng 2 đầu.	1.Nhập thông tin: Tên đăng nhập: PhongHd Mật khẩu: 123456 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Đặng Thị Nhỏ
DN_011	Kiểm tra trường hợp nhập tên tài khoản có dấu gạch ngang, dấu nháy đơn, dấu nháy kép.	1.Nhập thông tin: Tên đăng nhập: phongh-d Mật khẩu: 123456 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Đặng Thị Nhỏ
DN_012	Kiểm tra trường hợp nhập mật khẩu nhỏ hơn 5 kí tự	1.Nhập thông tin: Tên đăng nhập: phonghd Mật khẩu: 1234 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Mật khẩu không chính xác".	Đặng Thị Nhỏ

#### 4.4.2.2. Chức năng tìm kiếm

Mã ca KT	Tiêu đề ca KT	Các bước thực hiện	Kết quả mong muốn	Người viết
TC_001	Tìm kiếm sản phẩm thành công bằng cách ấn vào button tìm kiếm	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Nhập vào từ tìm kiếm: iphone. 4. Click vào biểu tượng tìm kiếm	1. Hiện trang chủ 2. Hiện từ tìm kiếm:iphone 3. Hiện danh sách các sản phẩm có tên có từ giống với từ nhập vào.	Đặng Thị Nhỏ
TC_002	Tìm kiếm sản phẩm thành công bằng cách ấn phím enter trên bàn phím	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Nhập vào từ tìm kiếm: iphone. 4. Click vào nút enter trên bàn phím	1. Hiện trang chủ 2. Hiện từ tìm kiếm:iphone 3. Hiện danh sách các sản phẩm có tên có từ giống với từ nhập vào.	Đặng Thị Nhỏ
TC_003	Tìm kiếm sản phẩm không thành công do không có sản phẩm nào có tên giống từ tìm kiếm, ấn vào biểu tượng tìm kiếm	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Nhập vào từ tìm kiếm: abcdefg 4. Click vào biểu tượng tìm kiếm	1. Hiện trang chủ 2. Hiện từ tìm kiếm: abcdefg 3. Hiện thị thông báo:”Sản phẩm bạn cần tìm không có”	Đặng Thị Nhỏ
TC_004	Tìm kiếm sản phẩm không thành công do không có sản phẩm nào có tên giống từ tìm kiếm,ấn phím enter trên bàn phím	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Nhập vào từ tìm kiếm: abcdefg 4. Click vào nút enter trên bàn phím	1. Hiện trang chủ 2. Hiện từ tìm kiếm: abcdefg 3. Hiện thị thông báo:”Sản phẩm bạn cần tìm không có”	Đặng Thị Nhỏ
TC_005	Tìm kiếm sản phẩm khi không nhập gì vào ô tìm kiếm và nhấn enter	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Không nhập gì vào ô tìm kiếm 4. Click vào nút enter trên bàn phím	1. Hiện trang chủ 3. Hiện thị tất cả các sản phẩm có trong CSDL ra màn hình	Đặng Thị Nhỏ

TC_006	Tìm kiếm sản phẩm khi không nhập gì vào ô tìm kiếm và click biểu tượng tìm kiếm	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Không nhập gì vào ô tìm kiếm 4. Click vào nút enter trên bàn phím	1. Hiện trang chủ 3. Hiện thị tất cả các sản phẩm có trong CSDL ra màn hình	Đặng Thị Nhỏ
--------	---	---	--	--------------

#### **4.4.3. Thực hiện kiểm thử**

##### *4.4.3.1. Ghi lại các bước kiểm thử*

B1. Tạo mới 1 file kịch bản: Chọn file\new\test

B2. Chọn Record → Record and run settings.

B3. Chọn Open the following address when recording or running. Trong ô chọn bên dưới nhập vào địa chỉ: <http://localhost:59442/Admin/LoginAdmin.aspx>

B4. Thực hiện thao tác các bước thực hiện test

B5. Sau khi thực hiện xong các bước test, bấm Stop để dừng việc ghi.

#### 4.4.3.2. Tạo file dữ liệu test.

##### a. Chức năng đăng nhập

phonghd	12345	Mật khẩu không chính xác
phong	123456	Tên đăng nhập không tồn tại
phong	12345	Tên đăng nhập không tồn tại
phonghd	123456	
phonghd		Bạn chưa nhập mật khẩu
phong		Tên đăng nhập không tồn tại
	123456	Bạn chưa nhập tên đăng nhập
	12345	Bạn chưa nhập tên đăng nhập
		Bạn chưa nhập tên đăng nhập và mật khẩu
phonghd	123456	Tên đăng nhập không tồn tại
phongh-d	123456	Tên đăng nhập không tồn tại
phonghd	1234	Mật khẩu không chính xác

##### b. Chức năng tìm kiếm

Từ khóa tìm kiếm	Thông báo
iphone	
abcdefg	sản phẩm bạn cần tìm không có

#### 4.4.3.3. Kết quả sau khi ghi lại các bước test và dữ liệu test.

DangNhap Action1 X Start Page

Main

```

1 wait 10
2 Browser("Browser").Page("Page").WebEdit("txtUsername").Set DataTable("Tài_khoản", dtGlobalSheet)
3 Browser("Browser").Page("Page").WebEdit("txtPassword").SetSecure DataTable("Mật_khẩu", dtGlobalSheet)
4 Browser("Browser").Page("Page").WebButton("Login").Click
5 If Browser("Browser").Page("Page").WebButton("Login").Exist Then
6 Browser("Browser").Page("Page").Check CheckPoint("Page_4")
7 Else
8 Browser("Browser").Page("Trang Chủ Quản Lý").Check CheckPoint("Trang Chủ Quản Lý_4")
9 End If
10 Browser("Browser").Page("Trang Chủ Quản Lý").Sync
11 Browser("Browser").CloseAllTabs

```

Data

C9		Bạn chưa nhập tên đăng nhập và mật khẩu						
	Tài_khoản	Mật_khẩu	Thông_báo	D	E	F	G	H
1	Phong	12345	Tên đăng nhập không tồn tại					
2	Phong	123456	Tên đăng nhập không tồn tại					
3	PhongHd	12345	Mật khẩu không chính xác					
4	PhongHd	123456						
5	PhongHd		Bạn chưa nhập mật khẩu					
6	Phong		Tên đăng nhập không tồn tại					
7		123456	Bạn chưa nhập tên đăng nhập					
8		12345	Bạn chưa nhập tên đăng nhập					

Hình 4.5. Kịch bản kiểm thử chức năng đăng nhập

timkiem Action1 X Start Page

Main

```

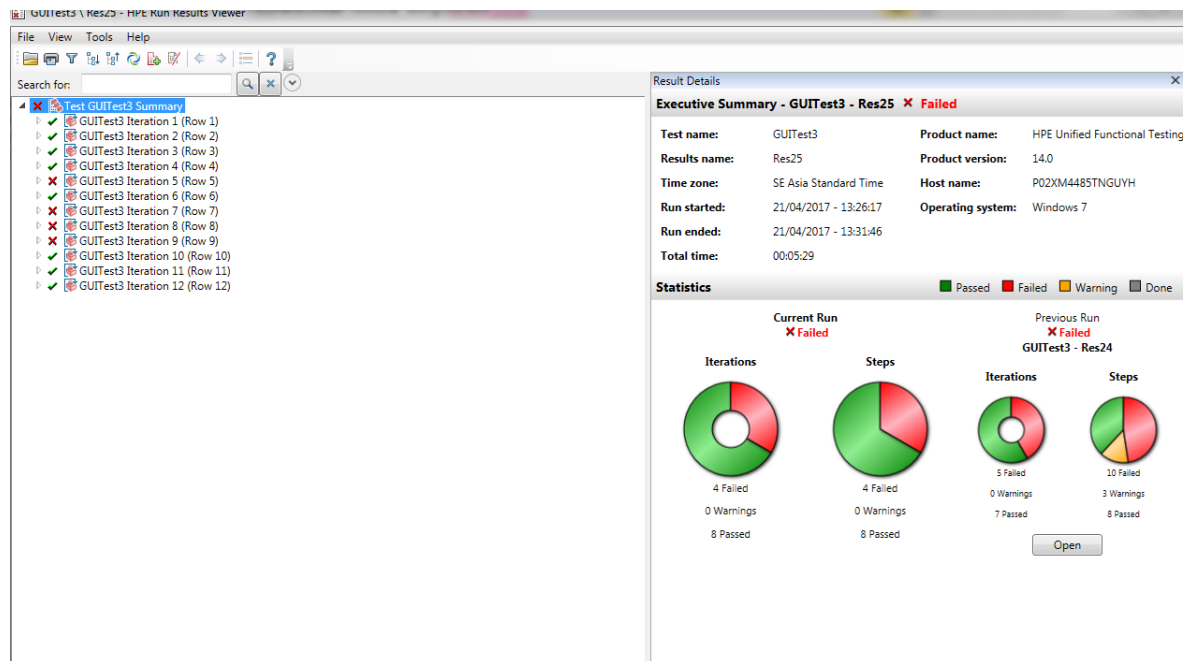
1 Browser("Browser").Page("Page").WebEdit("txtUsername").Set "PhongHd"
2 Browser("Browser").Page("Page").WebEdit("txtPassword").SetSecure "58fd706c131e7905285da38a62bffb65692"
3 Browser("Browser").Page("Page").WebButton("Login").Click
4 Browser("Browser").Page("Trang Chủ Quản Lý").Link("Sản Phẩm").Click
5 Browser("Browser").Page("Trang Chủ Quản Lý_2").WebEdit("ctl00$ContentPlaceholder2$txt1").Set DataTable("Từ_khóa_tìm_kiểm", dtGlobalSheet)
6 Browser("Browser").Page("Trang Chủ Quản Lý_2").WebButton("Tìm kiếm").Click
7 If Browser("Browser").Page("Trang Chủ Quản Lý_2").WebButton("Tìm kiếm").Exist then
8 Browser("Browser").Page("Trang Chủ Quản Lý_2").Check CheckPoint("Trang Chủ Quản Lý")
9 else
10 Browser("Browser").Page("Trang Chủ Quản Lý_2").Check CheckPoint("Trang Chủ Quản Lý_2")
11 End If
12 Browser("Browser").Page("Trang Chủ Quản Lý_2").Check CheckPoint("Trang Chủ Quản Lý_3")
13 Browser("Browser").Page("Trang Chủ Quản Lý_2").Sync
14 Browser("Browser").CloseAllTabs

```

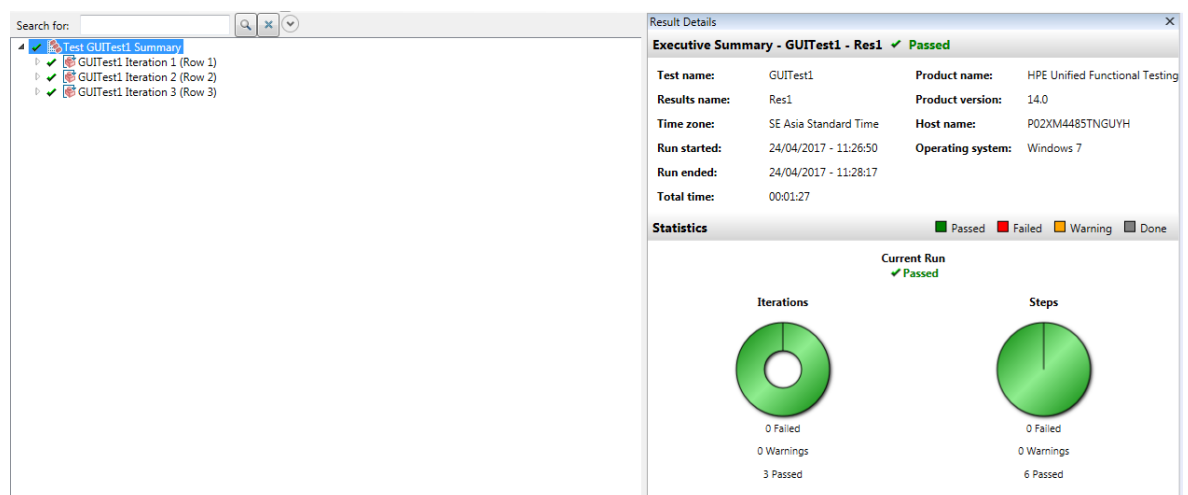
Data

D1		Apple				
	Từ_khóa_tìm_kiểm	Thông_báo	Trang_Chủ_Quản_LýChecked_Text	Trang_Chủ_Quản_Lý_3Checked_Text	E	F
1	iphone		iPhone	Apple		
2						
3	abcdefg	sản phẩm bạn cần tìm không có				

Hình 4.6. Kịch bản kiểm thử chức năng tìm kiếm



Hình 4.7. Kết quả của chức năng đăng nhập



Hình 4.8. Kết quả của chức năng tìm kiếm

#### 4.4.4. Kết quả kiểm thử

##### 4.4.4.1. Chức năng đăng nhập

Mã ca KT	Tiêu đề ca KT	Các bước thực hiện	Kết quả mong muốn	Kết quả thực tế
DN_001	Kiểm tra trường hợp nhập đúng tên tài khoản và nhập sai mật khẩu.	1.Nhập thông tin: Tên đăng nhập: PhongHd Mật khẩu: 12345 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Mật khẩu không chính xác".	Passed
DN_002	Kiểm tra trường hợp nhập sai tên tài khoản và nhập đúng mật khẩu.	1.Nhập thông tin: Tên đăng nhập: phong Mật khẩu: 123456 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Passed
DN_003	Kiểm tra trường hợp nhập sai tài khoản và sai mật khẩu	1.Nhập thông tin: Tên đăng nhập: phong Mật khẩu: 12345 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Passed
DN_004	Kiểm tra trường hợp nhập đúng tài khoản và mật khẩu	1.Nhập thông tin: Tên đăng nhập: PhongHd Mật khẩu: 123456 2.Click button: Login	Đăng nhập thành công	Passed
DN_005	Kiểm tra trường hợp nhập đúng tài khoản và bỏ trống mật khẩu	1.Nhập thông tin: Tên đăng nhập: PhongHd Mật khẩu: bỏ trống 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Bạn chưa nhập mật khẩu".	Failed
DN_006	Kiểm tra trường hợp nhập sai tài khoản và bỏ trống mật khẩu	1. Nhập thông tin: Tên đăng nhập: phong Mật khẩu: bỏ trống 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Passed
DN_007	Kiểm tra trường hợp bỏ trống tên tài khoản và mật khẩu đúng	1.Nhập thông tin: Tên đăng nhập: bỏ trống Mật khẩu: 123456 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Bạn chưa nhập tài khoản".	Failed
DN_008	Kiểm tra trường hợp bỏ trống tên tài khoản và mật khẩu sai	1.Nhập thông tin: Tên đăng nhập: bỏ trống Mật khẩu: 12345 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Bạn chưa nhập tài khoản".	Failed

DN_009	Đăng nhập không thành công khi bỏ trống tài khoản và mật khẩu	1.Nhập thông tin: Tên đăng nhập: Bỏ trống Mật khẩu: Bỏ trống 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Bạn chưa nhập tài khoản và mật khẩu".	Failed
DN_010	Kiểm tra trường hợp nhập tài khoản có khoảng trắng 2 đầu.	1.Nhập thông tin: Tên đăng nhập: PhongHd Mật khẩu: 123456 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Passed
DN_011	Kiểm tra trường hợp nhập tên tài khoản có dấu gạch ngang, dấu nháy đơn, dấu nháy kép.	1.Nhập thông tin: Tên đăng nhập:PhongH-d Mật khẩu: 123456 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Tên đăng nhập không tồn tại".	Passed
DN_012	Kiểm tra trường hợp nhập mật khẩu nhỏ hơn 5 kí tự	1.Nhập thông tin: Tên đăng nhập:PhongHd Mật khẩu: 1234 2.Click button: Login	Đăng nhập không thành công. Hiển thị thông báo: "Mật khẩu không chính xác".	Passed



#### 4.4.4.2. Chức năng tìm kiếm

Mã ca KT	Tiêu đề ca KT	Các bước thực hiện	Kết quả mong muốn	Kết quả thực tế
TC_001	Tìm kiếm sản phẩm thành công bằng cách ấn vào button tìm kiếm	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Nhập vào từ tìm kiếm: iphone. 4. Click vào biểu tượng tìm kiếm	1. Hiện trang chủ 2. Hiện từ tìm kiếm:iphone 3. Hiện danh sách các sản phẩm có tên có từ giống với từ nhập vào.	Passed
TC_002	Tìm kiếm sản phẩm thành công bằng cách ấn phím enter trên bàn phím	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Nhập vào từ tìm kiếm: iphone. 4. Click vào nút enter trên bàn phím	1. Hiện trang chủ 2. Hiện từ tìm kiếm:iphone 3. Hiện danh sách các sản phẩm có tên có từ giống với từ nhập vào.	Passed
TC_003	Tìm kiếm sản phẩm không thành công do không có sản phẩm nào có tên giống từ tìm kiếm, ấn vào biểu tượng tìm kiếm	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Nhập vào từ tìm kiếm: abcdefg 4. Click vào biểu tượng tìm kiếm	1. Hiện trang chủ 2. Hiện từ tìm kiếm: abcdefg 3. Hiện thị thông báo:”Sản phẩm bạn cần tìm không có”	Passed
TC_004	Tìm kiếm sản phẩm không thành công do không có sản phẩm nào có tên giống từ tìm kiếm,ấn phím enter trên bàn phím	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Nhập vào từ tìm kiếm: abcdefg 4. Click vào nút enter trên bàn phím	1. Hiện trang chủ 2. Hiện từ tìm kiếm: abcdefg 3. Hiện thị thông báo:”Sản phẩm bạn cần tìm không có”	Passed
TC_005	Tìm kiếm sản phẩm khi không nhập gì vào ô tìm kiếm và nhấn enter	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Không nhập gì vào ô tìm kiếm 4. Click vào nút enter trên bàn phím	1. Hiện trang chủ 3. Hiện thị tất cả các sản phẩm có trong CSDL ra màn hình	Passed

TC_006	Tìm kiếm sản phẩm khi không nhập gì vào ô tìm kiếm và click biểu tượng tìm kiếm	1. Đăng nhập thành công Admin 2. Click vào sản phẩm 3. Không nhập gì vào ô tìm kiếm 4. Click vào nút enter trên bàn phím	1. Hiện trang chủ 3. Hiện thị tất cả các sản phẩm có trong CSDL ra màn hình	Passed
--------	---	---	--	--------

#### 4.4.5. Các lỗi tồn tại trong hệ thống

ID bug	Tên bug/ Bug name	Nội dung bug/ Bug Description	Assignee	Version phát sinh bug	Trạng thái bug
Bug001	[Đăng nhập]Đăng nhập không thành công với trường hợp nhập đúng tài khoản và bỏ trống mật khẩu	Hiện tượng :Hiện thị thông báo mật khẩu không chính xác  Mong muốn: Hiện thị thông báo "Bạn chưa nhập mật khẩu"  Các bước thực hiện 1.Nhập thông tin: Tên đăng nhập: PhongHd Mật khẩu: bỏ trống 2.Click button: Login	Đặng Thị Nhỏ	V.0.1	Open
Bug002	[Đăng nhập]Đăng nhập không thành công với trường hợp bỏ trống tên tài khoản và mật khẩu đúng	Hiện tượng :Hiện thị thông báo Tên đăng nhập không tồn tại  Mong muốn: Hiện thị thông báo "Bạn chưa nhập tên tài khoản"  Các bước thực hiện 1.Nhập thông tin: Tên đăng nhập: Bỏ trống Mật khẩu: 123456 2.Click button: Login	Đặng Thị Nhỏ	V.0.1	Open

<b>Bug003</b>	[Đăng nhập]Đăng nhập không thành công với trường hợp bỏ trống tên tài khoản và nhập mật khẩu sai	<p>Hiện tượng :Hiện thị thông báo Tên đăng nhập không tồn tại</p> <p>Mong muốn: Hiện thị thông báo "Bạn chưa nhập tên tài khoản"</p> <p>Các bước thực hiện 1.Nhập thông tin: Tên đăng nhập: Bỏ trống Mật khẩu: 12345 2.Click button: Login</p>	Đặng Thị Nhỏ	V.0.1	Open
<b>Bug004</b>	[Đăng nhập]Đăng nhập không thành công với trường hợp bỏ trống tên tài khoản và mật khẩu	<p>Hiện tượng :Hiện thị thông báo Tên đăng nhập không tồn tại</p> <p>Mong muốn: Hiện thị thông báo "Bạn chưa nhập tên tài khoản và mật khẩu"</p> <p>Các bước thực hiện 1.Nhập thông tin: Tên đăng nhập: Bỏ trống Mật khẩu: bỏ trống 2.Click button: Login</p>	Đặng Thị Nhỏ	V.0.1	Open

## KẾT LUẬN

### 1. Kết quả đạt được:

Trình bày được các vấn đề:

- Khái niệm về kiểm thử phần mềm; Các phương pháp kiểm thử; Các chiến lược kiểm thử; Các cấp độ kiểm thử trong kiểm thử phần mềm.
- Giới thiệu công cụ kiểm thử phần mềm Quick Test Professional
- Giới thiệu các thao tác cơ bản để sử dụng công cụ Quick Test Professional.
- Áp dụng các kiến thức đã nghiên cứu thực hiện kiểm thử chức năng đăng nhập và chức năng tìm kiếm cho website bán điện thoại di động.

### 2. 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, và đặc biệt là công cụ kiểm thử Quick Test Professional, 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://istqbexamcertification.com/what-is-a-software-testing/> [online]
- [2] [https://vi.wikipedia.org/wiki/Ki%E1%BB%83m\\_th%E1%BB%AD\\_ph%E1%BA%A7n\\_m%E1%BB%81m](https://vi.wikipedia.org/wiki/Ki%E1%BB%83m_th%E1%BB%AD_ph%E1%BA%A7n_m%E1%BB%81m) [Online].
- [3] <http://learntesting123.blogspot.com/2015/04/bat-au-voi-cong-cu-quick-test-pro.html> [online]
- [4] <https://home.saas.hpe.com/myaccount/#/myProducts> [online]
- [5] <https://www.ntu.edu.sg/home/ehchua/programming/java/JavaUnitTesting.html> [online]
- [6] <http://labs.septeni-technology.jp/sqa/mot-so-ky-thuat-thiet-ke-test-case/> [online]
- [7] <http://www.testingvn.com/viewforum.php?f=11> [online]
- [8] <https://www.youtube.com/watch?v=fK1yp0cjPcM> [online]
- [9] [https://en.wikipedia.org/wiki/HP\\_QuickTest\\_Professional](https://en.wikipedia.org/wiki/HP_QuickTest_Professional) [online]