BÁO CÁO MÔN KIỂM THỬ PHẦN MỀM

NHÓM:

**NHẬT KÍ THỰC HIỆN ĐỒ ÁN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ tên sinh viên** | | | **Ký tên** |
| Họ tên sinh viên 1: Phạm Nguyễn Hoàng Vĩnh Phúc | | |  |
| Họ tên sinh viên 2: Lý Thanh Hùng | | |  |
| Họ tên sinh viên 3: Trần Hoàng | | |  |
| **Ngày** | **Phiên bản** | **Mô tả công việc** | ***Sinh viên thực hiện*** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

MỤC LỤC

**CHƯƠNG 1. GIỚI THIỆU**

* 1. **Mục tiêu đồ án**
* Có cái nhìn đúng đắn và sâu sắc hơn về các vấn đề cơ bản của công nghệ phần mềm, lỗi phần mềm và kiểm thử phần mềm.
* Hiểu rõ về các thành phần của bộ công cụ xxxxx.
* Nắm được cách thức sử dụng của bộ công cụ xxxxx.
* Ứng dụng được các kiến thức kiểm thử phần mềm, kỹ thuật kiểm thử hộp đen, kỹ thuật kiểm thử hộp trắng đã được học vào đồ án để viết kịch bản kiểm thử cho website bán đồng hồ ZiWatch.
  1. **Các chức năng / module thực hiện kiểm thử**
* Trong đồ án môn học này, chúng em sẽ thực hiện kiểm thử các chức năng sau của website bán đồng hồ ZiWatch:
* Chức năng đăng ký tài khoản
* Chức năng đăng nhập
* Chức năng xxx
* Chức năng xxx
* Chức năng xxx
  1. **Môi trường, công cụ và các giả định**
* Đồ án này chúng em tiến hành kiểm thử trên môi trường như sau:
* Hệ điều hành:
* Trình duyệt:
* WebServer:
* Database Server:
* Ngôn ngữ lập trình:
* …..
  1. **Nhân sự thực hiện**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Họ và tên** | **MSSV** | **Vai trò** |
| 1 | Phạm Nguyễn Hoàng Vĩnh Phúc |  | Nhóm trưởng |
| 2 | Lý Thanh Hùng |  | Thành viên |
| 3 | Trần Hoàng | 1511061103 | Thành viên |

* 1. **Kế hoạch thực hiện và phân công việc**

|  |  |
| --- | --- |
| **Ngày** | **Nội dung công việc** |
| 24/06/2020 | Họp nhóm và phân công nhiệm vụ cho từng thành viên |
| 25/06/2020 | Mỗi thành viên tiến hành nhiệm vụ được phân công |
| 26/06/2020 | Mỗi thành viên tiến hành nhiệm vụ được phân công |
| 27/06/2020 | Họp nhóm và phân công tiếp nhiệm vụ cho từng thành viên |
| 28/06/2020 | Mỗi thành viên tiến hành nhiệm vụ được phân công |
| 29/06/2020 | Mỗi thành viên tiến hành nhiệm vụ được phân công |
| 30/06/2020 | Tiến hành chạy deadline cho đồ án |
| 01/07/2020 | Tiến hành chạy deadline cho đồ án |
| 02/07/2020 | Họp nhóm và chạy demo, tổng kết hoàn thành đồ án |
| 03/07/2020 | Báo cáo kết thúc môn học |

**CHƯƠNG 2. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM**

## 2.1. Các cấp độ kiểm thử phần mềm

### 2.1.1. Kiểm thử đơn vị (Unit Test)

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

Kiểm thử đơn vị thường do lập trình viên thực hiện. Công đoạn này cần được thực hiện càng sớm càng tốt trong giai đoạn viết code và xuyên suốt chu kỳ phát triển phần mềm.

Mục đích của kiểm thử đơn vị là bảo đảm thông tin được xử lý và kết xuất (khỏi Unit) là chính xác, trong mối tương quan với dữ liệu nhập và chức năng xử lý của Unit. Điều này thường đòi hỏi tất cả các nhánh bên trong Unit đều phải được kiểm tra để phát hiện nhánh phát sinh lỗi.

Cũng như các mức kiểm thử khác, kiểm thử đơn vị cũng đòi hỏi phải chuẩn bị trước các ca kiểm thử (hay trường hợp kiểm thử) (test case) hoặc kịch bản (test script), trong đó chỉ định rõ dữ liệu vào, các bước thực hiện và dữ liệu mong muốn sẽ xuất ra. Các test case và test script được giữ lại để sử dụng sau này.

### 2.1.2. Kiểm thử tích hợp (Integration Test)

Kiểm thử tích hợp kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành. Trong khi kiểm thử đơn vị kiểm tra các thành phần và Unit riêng lẻ thì kiểm thử tích hợp kết hợp chúng lại với nhau và kiểm tra sự giao tiếp giữa chúng.

Kiểm thử tích hợp có hai mục tiêu chính là:

- Phát hiện lỗi giao tiếp xảy ra giữa các Unit

- Tích hợp các Unit đơn lẻ thành các hệ thống con (gọi là subsystem) và cuối cùng là nguyên hệ thống hoàn chỉnh chuẩn bị cho kiểm thử ở mức hệ thống (system test).

Có 4 loại kiểm thử trong kiểm thử tích hợp như sau:

- Kiểm thử cấu trúc (Structure test): Kiểm thử nhằm bảo đảm các thành phần bên trong của một chương trình chạy đúng, chú trọng đến hoạt động của các thành phần cấu trúc nội tại của chương trình, chẳng hạn các lệnh và nhánh bên trong.

- Kiểm thử chức năng (Functional test): Kiểm thử chỉ chú trọng đến chức năng của chương trình, không quan tâm đến cấu trúc bên trong, chỉ khảo sát chức năng của chương trình theo yêu cầu kỹ thuật.

- Kiểm thử hiệu năng (Performance test): Kiểm thử việc vận hành của hệ thống.

- Kiểm thử khả năng chịu tải (Stress test): Kiểm thử các giới hạn của hệ thống.

### 2.1.3. Kiểm thử hệ thống (System Test)

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

Kiểm thử hệ thống kiểm tra cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật.

Kiểm thử hệ thống bắt đầu khi tất cả các bộ phận của phần mềm đã được tích hợp thành công. Thông thường loại kiểm thử này tốn rất nhiều công sức và thời gian. Trong nhiều trường hợp, việc kiểm thử đòi hỏi một số thiết bị phụ trợ, phần mềm hoặc phần cứng đặc thù, đặc biệt là các ứng dụng thời gian thực, hệ thống phân bố, hoặc hệ thống nhúng. Ở mức độ hệ thống, người kiểm thử cũng tìm kiếm các lỗi, nhưng trọng tâm là đánh giá về hoạt động, thao tác, sự tin cậy và các yêu cầu khác liên quan đến chất lượng của toàn hệ thống.

Điểm khác nhau then chốt giữa kiểm thử tích hợp và kiểm thử hệ thống là kiểm thử hệ thống chú trọng các hành vi và lỗi trên toàn hệ thống, còn kiểm thử tích hợp chú trọng sự giao tiếp giữa các đơn thể hoặc đối tượng khi chúng làm việc cùng nhau. Thông thường ta phải thực hiện kiểm thử đơn vị và kiểm thử tích hợp để bảo đảm mọi Unit và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện kiểm thử hệ thống.

Sau khi hoàn thành kiểm thử tích hợp, một hệ thống phần mềm đã được hình thành cùng với các thành phần đã được kiểm tra đầy đủ. Tại thời điểm này, lập trình viên hoặc kiểm thử viên (Tester) bắt đầu kiểm thử phần mềm như một hệ thống hoàn chỉnh. Việc lập kế hoạch cho kiểm thử hệ thống nên bắt đầu từ giai đoạn hình thành và phân tích các yêu cầu.

Đòi hỏi nhiều công sức, thời gian và tính chính xác, khách quan, kiểm thử hệ thống được thực hiện bởi một nhóm kiểm tra viên hoàn toàn độc lập với nhóm phát triển dự án để đảm bảo tính chính xác và khách quan.

Kiểm thử hệ thống thường có các loại kiểm thử sau:

Kiểm thử chức năng (Functional test): Bảo đảm các hành vi của hệ thống thỏa mãn đúng yêu cầu thiết kế.

- Kiểm thử khả năng vận hành (Performance test): Bảo đảm tối ưu việc phân bổ tài nguyên hệ thống (ví dụ bộ nhớ) nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng câu truy vấn,....

- Kiểm thử khả năng chịu tải (Stress test hay Load test): Bảo đảm hệ thống vận hành đúng dưới áp lực cao (ví dụ nhiều người truy xuất cùng lúc). Stress test tập trung vào các trạng thái tới hạn, các "điểm chết", các tình huống bất thường như đang giao dịch thì ngắt kết nối (xuất hiện nhiều trong test thiết bị như POS, ATM),....

- Kiểm thử cấu hình (Configuration test): Đảm bảo hệ thống hoạt động tương thích với các loại phần cứng khác nhau.

- Kiểm thử khả năng bảo mật (Security test): Bảo đảm tính toàn vẹn, bảo mật của dữ liệu và của hệ thống.

- Kiểm thử khả năng phục hồi (Recovery test): Bảo đảm hệ thống có khả năng khôi phục trạng thái ổn định trước đó trong tình huống mất tài nguyên hoặc dữ liệu; đặc biệt quan trọng đối với các hệ thống giao dịch như ngân hàng trực tuyến.

### 2.1.4. Kiểm thử chấp nhận sản phẩm (Acceptance Test)

Mục đích của kiểm thử chấp nhận là kiểm thử khả năng chấp nhận cuối cùng để chắc chắn rằng sản phẩm là phù hợp và thỏa mãn các yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm.

Trong giai đoạn kiểm thử chấp nhận thì người kiểm tra là khách hàng. Khách hàng sẽ đánh giá phần mềm với mong đợi theo những thao tác sử dụng quen thuộc của họ. Việc kiểm tra ở giai đoạn này có ý nghĩa hết sức quan trọng tránh cho việc hiểu sai yêu cầu cũng như sự mong đợi của khách hàng.

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

## 2.2. Kỹ thuật kiểm thử phần mềm

Mục tiêu của kiểm thử là phải thiết kế các trường hợp kiểm thử có khả năng cao nhất trong việc phát hiện nhiều lỗi với thời gian và công sức tối thiểu. Do đó có thể chia các kỹ thuật kiểm thử thành hai loại:

- Kỹ thuật kiểm thử hộp đen (Black – box Testing) hay còn gọi là kỹ thuật kiểm thử chức năng (Functional Testing).

- Kỹ thuật kiểm thử hộp trắng (White – box Testing) hay còn gọi là kỹ thuật kiểm thử cấu trúc (Structural Testing).

### 2.2.1. Kỹ thuật kiểm thử hộp đen (Black – box Testing)

Kiểm thử hộp đen còn được gọi là kiểm thử hướng dữ liệu (data - driven) hay là kiểm thử hướng vào/ra (input/output driven).

Trong kỹ thuật này, người kiểm thử xem phần mềm như là một hộp đen. Người kiểm thử hoàn toàn không quan tâm đến cấu trúc và hành vi bên trong của chương trình. Người kiểm thử chỉ cần quan tâm đến việc tìm các hiện tượng mà phần mềm không hành xử theo đúng đặc tả của nó. Do đó, dữ liệu kiểm thử sẽ xuất phát từ đặc tả.

Như vậy, cách tiếp cận kiểm thử hộp đen tập trung vào các yêu cầu chức năng của phần mềm. Kiểm thử hộp đen cho phép người kiểm thử xây dựng các nhóm giá trị đầu vào sẽ thực thi đầy đủ tất cả các yêu cầu chức năng của chương trình. Kiểm thử hộp đen không thay thế kỹ thuật kiểm thử hộp trắng, nhưng nó bổ sung khả năng phát hiện các lớp lỗi khác với các phương pháp hộp trắng.

Kiểm thử hộp đen cố gắng tìm các loại lỗi sau:

- Các chức năng thiếu hoặc không đúng.

- Các lỗi giao diện.

- Các lỗi cấu trúc dữ liệu trong truy cập cơ sở dữ liệu bên ngoài.

- Các lỗi thực hiện.

- Các lỗi khởi tạo hoặc kết thúc.

- Và các lỗi khác ...

Không giống với kiểm thử hộp trắng được thực hiện sớm trong quá trình kiểm thử, kiểm thử hộp đen được áp dụng trong các giai đoạn sau của kiểm thử. Vì kiểm thử hộp đen không để ý có chủ đích cấu trúc điều khiển, sự quan tâm tập trung trên miền thông tin. Nếu người kiểm thử muốn sử dụng phương phá

chương trình thì điều kiện bắt buộc là phải kiểm thử tất cả các đầu vào, tức là mỗi một điều kiện đầu vào có thể có là một trường hợp kiểm thử. Bởi vì nếu chỉ kiểm thử một số điều kiện đầu vào thì không đảm bảo được chương trình đã hết lỗi. Vì thế, để đạt được mục tiêu kiểm thử, người ta đã áp dụng một số phương pháp kiểm thử hộp đen như: phân hoạch tương đương, phân tích giá trị biên.

### 2.2.2. Kỹ thuật kiểm thử hộp trắng (White – box Testing)

Kiểm thử hộp trắng hay còn gọi là kiểm thử hướng logic, cho phép kiểm tra cấu trúc bên trong của phần mềm với mục đích bảo đảm rằng tất cả các câu lệnh và điều kiện sẽ được thực hiện ít nhất một lần. Người kiểm thử truy nhập vào mã nguồn chương trình và có thể kiểm tra nó, lấy đó làm cơ sở để hỗ trợ việc kiểm thử.

Các loại kiểm thử:

* Kiểm thử đường dẫn cơ sở.
* Kiểm thử cấu trúc điều khiển:

+ Kiểm thử bao phủ

+ Kiểm thử luồng dữ liệu.

+ Kiểm thử vòng lặp.

# CHƯƠNG 2. CHIẾN LƯỢC KIỂM THỬ

## 2.1. Phân tích yêu cầu và đọc code của chức năng cần kiểm thử

- Nhóm chúng em tiến hành đọc đặc tả, phân tích yêu cầu của từng chức năng cần kiểm thử, sau đó áp dụng các phương pháp kiểm thử hộp đen để tìm ra số testcase cần thiết của từng chức năng.

- Sau đó nhóm chúng em tiến hành đọc code, phân tích code và áp dụng các phương pháp kiểm thử hộp trắng để tìm ra số testcase cần thiết của từng chức năng.

## 2.2. Thiết kế các testcase để kiểm thử

- Sau khi tìm ra số các testcase cần thiết của từng chức năng, nhóm chúng em tiến hành phân loại mức độ nghiêm trọng của các testcase từ cao đến thấp, sau đó tiến hành thiết kế các testcase có mức độ nghiêm trọng cao trước rồi đến thiết kế các testcase có mức độ nghiêm trọng thấp sau.

- Các testcase sẽ được lưu vào một file excel để dùng cho bước sau.

## 2.3. Thực hiện kiểm thử

- Sau khi đã có các testcase nhóm chúng em tiến hành dùng công cụ kiểm thử tự động xxxxx để test.

- Nhóm chúng em thực hiện quay Record các thao tác

## 2.4. Ghi nhận kết quả kiểm thử, quản lý testcase và bug

- Nhóm chúng em thực hiện quay Record các thao tác test trên website bằng công cụ xxxxx. Sau đó từ Record vừa quay xong chúng em tiến hành đổi giá trị các tham số nhập vào để tạo thành các testcase khác.

- Nhóm chúng em quản lý các testcase bằng phần mềm Microsoft Excel 2016.

- Sau khi thực hiện test các testcase, nhóm chúng em tiến hành ghi nhận các kết quả nhận được vào file excel ở bước trước.

- Sau khi thực hiện test xong các testcase, nếu phát hiện có bug chúng em sẽ ghi nhận bug vào công cụ quản lý bug xxxxx.

## 2.5. Phương thức đánh giá kết quả

- Sau khi ghi nhận các kết quả nhận được vào file excel nhóm chúng em tiến hành bàn bạc và đánh giá kết quả nhận được dựa theo tiêu chí kết quả nhận được có giống với kết quả mong đợi hay không.

- Nếu kết quả nhận được đúng với kết quả mong đợi thì nhóm chúng em cập nhật trạng thái “OK” vào testcase trong file excel (“OK” là chương trình đúng không cần sửa gì thêm).

- Nếu kết quả nhận được không đúng với mong đợi thì nhóm chúng em cập nhật trạng thái “NG” vào testcase trong file excel (“NG” là chuong trình chạy không đúng phải chỉnh sửa lại cho đúng). Sau đó nhóm chúng em sẽ đánh giá mức độ nghiêm trọng của lỗi trên thang điểm 5 và mô tả nội dung lỗi và quyết định xem có đề nghị sửa lỗi này hay không vào bảng “Danh sách lỗi tìm được”.

**CHƯƠNG 3. THỰC HIỆN KIỂM THỬ**

* 1. **Kiểm thử đơn vị**
     1. **Test case 1**

|  |  |
| --- | --- |
| **Test Case ID: DV01** | |
| **Tên của Test case: Kiểm tra đăng nhập rỗng (LOGIN null)** | |
| **Mục đích kiểm tra** | Đảm bảo trang đăng nhập không có lỗi xét duyệt |
| **Chức năng kiểm tra** | Đăng nhập |
| **Giả định** | Trang web sẽ không cho login nếu người dùng không nhập thông tin đăng nhập |
| **Dữ liệu dùng để kiểm tra** | 1. Email = null 2. password = null |
| **Các bước thực hiện** | 1. Nhấp vào mục login bên phải header 2. Để trống mục nhập cho trước 3. Nhấp nút Sign in |
| **Kết quả mong đợi** | Trang đăng nhập không cho chuyển trang và cho nhập lại |
| **Kết quả thực hiện** | Trang đăng nhập không cho chuyển trang và thông báo cho nhập lại |
| **Đề nghị, ghi chú thêm** | Không |

* + 1. **Test case 2**

|  |  |
| --- | --- |
| **Test Case ID: DV02** | |
| **Tên của Test case: Kiểm tra đăng nhập rỗng (LOGIN null)** | |
| **Mục đích kiểm tra** | Đảm bảo trang đăng nhập không có lỗi xét duyệt |
| **Chức năng kiểm tra** | Đăng nhập |
| **Giả định** | Trang web sẽ không cho login nếu người dùng không nhập thông tin Email đăng nhập |
| **Dữ liệu dùng để kiểm tra** | 1. Email = null 2. Password = true |
| **Các bước thực hiện** | 1. Nhấp vào mục login bên phải header 2. Để trống mục nhập cho trước 3. Nhấp nút Sign in |
| **Kết quả mong đợi** | Trang đăng nhập không cho chuyển trang và cho nhập lại |
| **Kết quả thực hiện** | Trang đăng nhập không cho chuyển trang và thông báo cho nhập lại |
| **Đề nghị, ghi chú thêm** | Không |

* + 1. **Test case 3**

|  |  |
| --- | --- |
| **Test Case ID: DV03** | |
| **Tên của Test case: Kiểm tra đăng nhập rỗng (LOGIN null)** | |
| **Mục đích kiểm tra** | Đảm bảo trang đăng nhập không có lỗi xét duyệt |
| **Chức năng kiểm tra** | Đăng nhập |
| **Giả định** | Trang web sẽ không cho login nếu người dùng không nhập thông tin password đăng nhập |
| **Dữ liệu dùng để kiểm tra** | 1. Email = true 2. Password = null |
| **Các bước thực hiện** | 1. Nhấp vào mục login bên phải header 2. Để trống mục nhập cho trước 3. Nhấp nút Sign in |
| **Kết quả mong đợi** | Trang đăng nhập không cho chuyển trang và cho nhập lại |
| **Kết quả thực hiện** | Trang đăng nhập không cho chuyển trang và thông báo cho nhập lại |
| **Đề nghị, ghi chú thêm** | Không |

* + 1. **Test case 4**

|  |  |
| --- | --- |
| **Test Case ID: DV04** | |
| **Tên của Test case: Kiểm tra đăng nhập sai thông tin (LOGIN wrong)** | |
| **Mục đích kiểm tra** | Đảm bảo trang đăng nhập không có lỗi xét duyệt |
| **Chức năng kiểm tra** | Đăng nhập |
| **Giả định** | Trang đăng nhập sẽ không cho login và thông báo nhập lại thông tin đăng nhập nếu nhập sai Email, Password |
| **Dữ liệu dùng để kiểm tra** | 1. Email = wrong 2. password = wrong |
| **Các bước thực hiện** | 1. Nhấp vào mục login bên phải header 2. Điền các mục nhập cho trước 3. Nhấp nút Sign in |
| **Kết quả mong đợi** | Sai thông tin đăng nhập |
| **Kết quả thực hiện** | Đăng nhập không thành công |
| **Đề nghị, ghi chú thêm** | Không |

* + 1. **Test case 5**

|  |  |
| --- | --- |
| **Test Case ID: DV05** | |
| **Tên của Test case: Kiểm tra đăng nhập sai thông tin (LOGIN wrong)** | |
| **Mục đích kiểm tra** | Đảm bảo trang đăng nhập không có lỗi xét duyệt |
| **Chức năng kiểm tra** | Đăng nhập |
| **Giả định** | Trang đăng nhập sẽ không cho login và thông báo nhập lại thông tin đăng nhập nếu nhập sai email |
| **Dữ liệu dùng để kiểm tra** | 1. Email = wrong 2. password = true |
| **Các bước thực hiện** | 1. Nhấp vào mục login bên phải header 2. Điền các mục nhập cho trước 3. Nhấp nút Sign in |
| **Kết quả mong đợi** | Sai thông tin đăng nhập |
| **Kết quả thực hiện** | Đăng nhập không thành công |
| **Đề nghị, ghi chú thêm** | Không |

* + 1. **Test case 6**

|  |  |
| --- | --- |
| **Test Case ID: DV06** | |
| **Tên của Test case: Kiểm tra đăng nhập sai thông tin (LOGIN wrong)** | |
| **Mục đích kiểm tra** | Đảm bảo trang đăng nhập không có lỗi xét duyệt |
| **Chức năng kiểm tra** | Đăng nhập |
| **Giả định** | Trang đăng nhập sẽ không cho login và thông báo nhập lại thông tin đăng nhập nếu nhập sai password |
| **Dữ liệu dùng để kiểm tra** | 1. Email = true 2. password = wrong |
| **Các bước thực hiện** | 1. Nhấp vào mục login bên phải header 2. Điền các mục nhập cho trước 3. Nhấp nút Sign in |
| **Kết quả mong đợi** | Sai thông tin đăng nhập |
| **Kết quả thực hiện** | Đăng nhập không thành công |
| **Đề nghị, ghi chú thêm** | Không |

* + 1. **Test case 7**

|  |  |
| --- | --- |
| **Test Case ID: DV07** | |
| **Tên của Test case: Kiểm tra đăng nhập sai thông tin (LOGIN wrong)** | |
| **Mục đích kiểm tra** | Đảm bảo trang đăng nhập không có lỗi xét duyệt |
| **Chức năng kiểm tra** | Đăng nhập |
| **Giả định** | Trang login sẽ cho đăng nhập và chuyển sdang trang admin nếu nhập đúng thông tin login của admin |
| **Dữ liệu dùng để kiểm tra** | 1. Email = true-admin 2. password = true-admin |
| **Các bước thực hiện** | 1. Nhấp vào mục login bên phải header 2. Điền các mục nhập cho trước 3. Nhấp nút Sign in |
| **Kết quả mong đợi** | 1. Đăng nhập thành công 2. Chuyển sang trang dành cho admin |
| **Kết quả thực hiện** | 1. Đăng nhập thành công 2. Chuyển sang trang dành cho admin |
| **Đề nghị, ghi chú thêm** | Không |

* + 1. **Test case 9**

|  |  |
| --- | --- |
| **Test Case ID: DV09** | |
| **Tên của Test case: Kiểm tra đăng kí (SIGNUP)** | |
| **Mục đích kiểm tra** |  |
| **Chức năng kiểm tra** |  |
| **Giả định** |  |
| **Dữ liệu dùng để kiểm tra** |  |
| **Các bước thực hiện** |  |
| **Kết quả mong đợi** |  |
| **Kết quả thực hiện** |  |
| **Đề nghị, ghi chú thêm** |  |

* + 1. Test case 4

|  |  |
| --- | --- |
| Test Case ID: DV04 | |
| Tên của Test case: Kiểm tra thông tin giỏ hàng (My cart) | |
| Mục đích kiểm tra |  |
| Chức năng kiểm tra |  |
| Giả định |  |
| Dữ liệu dùng để kiểm tra |  |
| Các bước thực hiện |  |
| Kết quả mong đợi |  |
| Kết quả thực hiện |  |
| Đề nghị, ghi chú thêm |  |

* 1. Kiểm thử cơ sở dữ liệu
     1. Test case 4

|  |  |
| --- | --- |
| Test Case ID: = | |
| Tên của Test case: | |
| Mục đích kiểm tra |  |
| Chức năng kiểm tra |  |
| Giả định |  |
| Dữ liệu dùng để kiểm tra |  |
| Các bước thực hiện |  |
| Kết quả mong đợi |  |
| Kết quả thực hiện |  |
| Đề nghị, ghi chú thêm |  |

* + 1. Test case 6

|  |  |
| --- | --- |
| Test Case ID: | |
| Tên của Test case: | |
| Mục đích kiểm tra |  |
| Chức năng kiểm tra |  |
| Giả định |  |
| Dữ liệu dùng để kiểm tra |  |
| Các bước thực hiện |  |
| Kết quả mong đợi |  |
| Kết quả thực hiện |  |
| Đề nghị, ghi chú thêm |  |

* 1. Kiểm thử giao diện
     1. Test case 7

|  |  |
| --- | --- |
| Test Case ID: | |
| Tên của Test case: | |
| Mục đích kiểm tra |  |
| Chức năng kiểm tra |  |
| Giả định |  |
| Dữ liệu dùng để kiểm tra |  |
| Các bước thực hiện |  |
| Kết quả mong đợi |  |
| Kết quả thực hiện |  |
| Đề nghị, ghi chú thêm |  |

* + 1. Test case 8

|  |  |
| --- | --- |
| Test Case ID: | |
| Tên của Test case: | |
| Mục đích kiểm tra |  |
| Chức năng kiểm tra |  |
| Giả định |  |
| Dữ liệu dùng để kiểm tra |  |
| Các bước thực hiện |  |
| Kết quả mong đợi |  |
| Kết quả thực hiện |  |
| Đề nghị, ghi chú thêm |  |

* 1. Kiểm thử tính tiện dụng
     1. Test case 9

|  |  |
| --- | --- |
| Test Case ID: | |
| Tên của Test case: | |
| Mục đích kiểm tra |  |
| Chức năng kiểm tra |  |
| Giả định |  |
| Dữ liệu dùng để kiểm tra |  |
| Các bước thực hiện |  |
| Kết quả mong đợi |  |
| Kết quả thực hiện |  |
| Đề nghị, ghi chú thêm |  |

* + 1. Test case 10

|  |  |
| --- | --- |
| Test Case ID: | |
| Tên của Test case: | |
| Mục đích kiểm tra |  |
| Chức năng kiểm tra |  |
| Giả định |  |
| Dữ liệu dùng để kiểm tra |  |
| Các bước thực hiện |  |
| Kết quả mong đợi |  |
| Kết quả thực hiện |  |
| Đề nghị, ghi chú thêm |  |

* 1. Các loại kiểm thử khác

# CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

## Kết quả kiểm thử

## Danh sách lỗi tìm được

# CHƯƠNG 5. NHẬN XÉT, ĐÁNH GIÁ VÀ ĐỀ XUẤT

## 5.1. Nhận xét, đánh giá

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

- Sau một thời gian thực hiện đồ án dưới sự hướng dẫn của thầy Dương Thành Phết, đồ án của chúng em đã thực hiện được khá tốt các mục tiêu đề ra và đạt được những kết quả khả quan, mặc dù đã cố gắng hết sức trong thời gian thực hiện đề tài nhưng với thời gian có hạn và kinh nghiệm còn ít nên đồ án không tránh khỏi những thiếu sót. Nhóm chúng em rất mong nhận được những góp ý của thầy để đồ án của chúng em được hoàn thiện hơn.

## 5.2. 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 bộ công cụ kiểm thử ứng dụng watir để có thể vận dụng vào kiểm thử các ứng dụng lớn hơn trong tương lai nhằm góp một phần nhỏ bé vào công cuộc chuyên nghiệp hóa kiểm thử phần mềm ở Việt Nam.