**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A blue button with white text

Description automatically generated**

**Học phần: Kiểm thử phần mềm**

**Lớp: DCT122C4**

|  |  |
| --- | --- |
| **Thành viên nhóm:** | **3122411049 - Lê Gia Hân** |
|  | **3122411079 - Nguyễn Thị Thu Hường** |
|  | **3122411256 - Nguyễn Ngọc Thúy Vy** |
|  | **3122411141 - Phan Thị Hồng Nhiên** |

**Giảng viên: Đỗ Như Tài**

|  |  |  |
| --- | --- | --- |
| **Mã số sinh viên** | **Tên thành viên** | **Email** |
| 3122411049 | Lê Gia Hân | hanlegia2302@gmail.com |
| 3122411079 | Nguyễn Thị Thu Hường | huonghanna10052004@gmail.com |
| 3122411256 | Nguyễn Ngọc Thúy Vy | vydethuong212@gmail.com |
| 3122411141 | Phan Thị Hồng Nhiên | phanthnhien27@gmail.com |

**MỤC LỤC**

[Câu a. 1](#_Toc213110067)

[Sơ đồ trạng thái của giỏ hàng 2](#_Toc213110068)

[Câu b. 3](#_Toc213110069)

[Câu c. 5](#_Toc213110070)

Giỏ hàng trên một trang mua bán trực tuyến được bắt đầu với trạng thái là rỗng (không có món hàng nào). Khi bạn chọn một sản phẩm thì nó sẽ được đưa vào giỏ hàng. Bạn cũng có thể bỏ chọn các món hàng trong giỏ hàng. Khi bạn quyết định mua hàng, thì sẽ xuất hiện màn hình tổng hợp các món hàng đang có trong giỏ cùng với thông tin về giá tiền, số lượng và tổng tiền của giỏ hàng, để cho bạn xác nhận xem đúng hay chưa. Nếu bạn thấy số lượng hàng và giá tiền OK thì bạn sẽ được chuyển sang trang thanh toán. Ngược lại bạn sẽ quay lại trang mua hàng (lúc này bạn có thể bỏ chọn các món hàng bạn muốn bỏ bớt).  
  
**Yêu cầu:**

1. Đưa ra sơ đồ trạng thái - state diagram – cho thấy các trạng thái/states và sự chuyển tiếp/transition khác. Xác định test case – một loạt các trạng thái – bao phủ toàn bộ các chuyển tiếp.
2. Đưa ra một bảng trạng thái. Cho một ví dụ kiểm thử trường hợp chuyển tiếp không hợp lệ
3. Dùng pairwise để đưa ra những kết hợp để test trên những môi trường:Client: Win XP, Win 7. Server: 2003, 2008. Browser: IE8, IE9, IE10, Firefox, Google Chrome và Safari

**Bài làm**

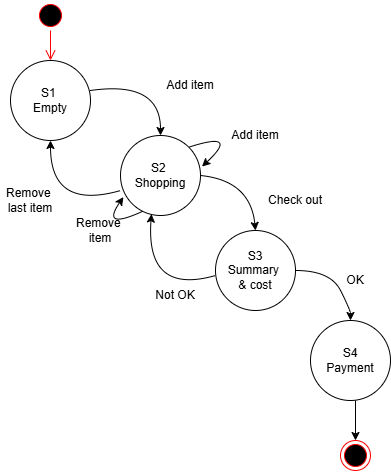
# Câu a.

* Trạng thái (state) khởi tạo là S1 khi giỏ hàng chưa có hàng (empty - đang rỗng).
* Khi thêm một món hàng vào giỏ (add item) thì sẽ chuyển sang (transition) trạng thái có hàng S2. Khi thêm một món hàng khác nữa vào giỏ thì không làm thay đổi trạng thái này (tức này chỉ làm thay đổi số lượng hàng trong giỏ). Các món hàng trong giỏ có thể bỏ bớt (remove item), khi bỏ hết 1 món hàng, mà trong giỏ còn hàng thì không làm thay đổi trạng thái, nhưng khi bỏ hết món cuối cùng (remove last item) thì sẽ bị chuyển sang trạng thái S1, là trạng thái rỗng.
* Khi khách hàng muốn kết thúc (checkout) thì chuyển sang trạng thái S3, là tóm tắt tiến trình (Summary & cost) để xem tính toán (Checkout) chi tiết (nếu mọi thứ đều ổn - giỏ hàng minh thuyết được - OK), thì chuyển sang mua.
* Nếu danh sách hàng này hợp lệ (có hàng và tổng hợp hợp lý không có vấn đề mua chẳng hạn,...) thì trạng thái S4 là thành toán; Nếu kiểm tra thấy có lỗi thì khách hàng rời quay lại trạng thái S4.

Tổng cộng có 4 trạng thái (rỗng, có hàng, tóm tắt, kết thúc) và 7 sự kiện (các đường nối trên).

Chú ý: Trong ví dụ này, S1 là Trạng Thái Bắt Đầu (Start State) và S4 là Trạng Thái Kết Thúc (End State), nghĩa là chỉ có S4 là không đi đâu được nữa, còn ở các trạng thái khác thì chúng ta đều có thể đi đến 1 trạng thái bất kỳ nào đó khác.

## Sơ đồ trạng thái của giỏ hàng



Dưới đây là một trường hợp – test case – bao phủ toàn bộ các trạng thái. Lưu ý rằng trạng thái kết thúc từ một bước (step) hoặc sự kiện (event) là trạng thái cho sự kiện tiếp theo, vì vậy các bước này phải được đi qua theo thứ tự này:

|  |  |  |
| --- | --- | --- |
| Trạng thái | Sự kiện (Event-Action) | Số lượng |
| S1 | Add item | 1 |
| S1 | Remove last item | 0 |
| S1 | Add item | 1 |
| S1 | Add item | 2 |
| S2 | Add item | 3 |
| S2 | Remove item | 2 |
| S2 | Check out | 2 |
| S3 | Not OK | 2 |
| S2 | Remove item | 1 |
| S2 | Check out | 1 |
| S4 |  | 1 |

Mục đích trong ví dụ này chúng ta không quan tâm đến việc gì sẽ xảy ra tại Trạng thái 4, nhưng có thể có vài sự kiện hoặc hành động nào đó xảy ra, ví dụ như click Cancel (không mua hàng nữa) hoặc click nút Payment để tiến hành thanh toán, và có thể là sau khi thực hiện thanh toán nó sẽ hiển thị màn hình thông báo hoặc chuyển sang màn hình khác hoặc bắt đầu một sơ đồ trạng thái khác (state diagram) như kiểm tra tính hợp lệ của thẻ credit card, số dư, email,...

# Câu b.

Dưới đây là một bảng mô tả trạng thái tương ứng với sơ đồ trạng thái trên:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Trạng thái/Sự kiện | Add item | Remove item | Remove last item | Check out | Not OK | OK |
| S1-Empty | S2 | - | - | - | - | - |
| S2-Shopping | S2 | S2 | S1 | S3 | - | - |
| S3-Summary | - | - | - | - | S2 | S4 |
| S4-Payment | - | - | - | - | - | - |

Trong bảng trên tất cả các ô chứa dấu “–” là trạng thái không hợp lệ (invalid transition). Sau đây là một vài ví dụ các trường hợp kiểm thử không hợp lệ có thể xảy ra:

* Thử thêm một món hàng ở trạng thái S3 – tổng kết và tính giá tiền.
* Thử bớt một món hàng khi giỏ hàng đang rỗng (empty) – không có món hàng nào – ở trạng thái S1.
* Thử bấm OK khi đang shopping – ở trạng thái S2 – vì ở màn hình shopping thì không có nút OK mà chỉ có nút Check out.

Từ bảng trên chúng ta sẽ viết được các test case sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase ID | Preconditions | Action | Expected | Ghi chú |
| TC1 | Giỏ hàng đang rỗng | Click nút Add item để thêm 1 món hàng | Giỏ hàng chứa món hàng đã chọn | S1 => S2 |
| TC2 | Giỏ hàng đang có ít nhất 1 món hàng | Click nút Add item để thêm 1 món hàng | Giỏ hàng chứa món hàng đã chọn | S2 => S2 |
| TC3 | Giỏ hàng đang có ít nhất 2 món hàng | Click nút Remove item để bớt 1 món hàng | Giỏ hàng bớt đi item vừa chọn | S2 => S1 |
| TC4 | Giỏ hàng đang có ít nhất 1 món hàng | Nhấn nút Checkout để tính tiền | Hiển thị danh sách hàng đang chọn trong giỏ và tính tổng | S2 => S3 |
| TC5 | Đang ở màn hình kiểm tra hàng và tiền | Click nút Back (thấy Not OK) để quay về màn hình mua hàng | Quay về màn hình mua hàng | S3 => S2 |
| TC6 | Đang ở màn hình kiểm tra hàng và tiền | Click nút Payment (thấy OK) để mua hàng | Hiển thị màn hình thanh toán | S3 => S4 |
| TC7 | Giỏ hàng đang rỗng | Click nút Checkout | Hiển thị thông báo lỗi hoặc không có nút Checkout | S1 => S3 |
| TC8 | Giỏ hàng đang rỗng | Click nút Remove item | Hiển thị thông báo lỗi hoặc không có nút Remove Item hoặc nút Remove Item bị disabled | S1 => S1 |
| TC9 | Giỏ hàng đang có ít nhất 1 món hàng | Click nút Payment | Hiển thị thông báo lỗi hoặc không có nút Payment | S2 => S4 |
| TC10 | Đang ở màn hình kiểm tra hàng và tiền | Click nút Add Item | Hiển thị thông báo lỗi hoặc không có nút Add Item | S3 => S1 |
| RC11 | Đang ở màn hình kiểm tra hàng và tiền | Click nút Remove item | Hiển thị thông báo lỗi hoặc không có nút Remove item | S3=>S2 |
| TC12 | Đang ở màn hình thanh toán | Click nút Add Item | Hiển thị thông báo lỗi hoặc không có nút Add item | S4=>S1 |
| TC13 | Đang ở màn hình thanh toán | Click nút Remove item | Hiển thị thông báo lỗi hoặc không có nút Remove item | S4=>S2 |

Vì trạng thái đầu tiên của chúng ta trong bài tập này là ở màn hình mua hàng và giỏ hàng đang rỗng ⇒ vì vậy ở đây không thêm case đang ở màn hình chính, click nút mua hàng để vào màn hình mua hàng; hoặc đang ở màn hình login, rồi vào màn hình mua hàng (shopping).

Và cũng không có case mô tả ở màn hình thanh toán sẽ đi đâu nữa, khi click Cancel thì quay về màn hình mua hàng hoặc khi click nút Thanh toán thì hiển thị màn hình in biên lai…

Các test case TC1 đến TC6 là test case cho các trường hợp hợp lệ (valid – Positive test case); các test case TC7 đến TC10 là các trường hợp không hợp lệ (invalid – Negative test case).

Trong trường hợp không hợp lệ là có một số case trên là không thực hiện được, nghĩa là không thể test được, như TC7, tức người dùng không có nút Checkout để bấm; hoặc nút bị ẩn, nên test case này sẽ được ghi chú lại.

# Câu c.

Dùng Pairwise để tạo các tổ hợp kiểm thử môi trường

Mục tiêu:

Giảm số lượng test case cần chạy trên các tổ hợp Client–Server–Browser, nhưng vẫn đảm bảo mỗi cặp giá trị của 2 biến bất kỳ đều được kiểm thử ít nhất một lần.

Các biến môi trường:

|  |  |
| --- | --- |
| Thành phần | Giá trị |
| Client | 2 (Win XP, Win 7) |
| Server | 2 (2003, 2008) |
| Browser | 6 (IE8, IE9, IE10, Firefox, Google Chrome, Safari) |

Tổng số tổ hợp nếu kiểm thử toàn bộ: Số lựa chọn của Client × Số lựa chọn của Server × Số lựa chọn của Browser: 2 x 2 x 6 = 24 test case

Nghĩa là, nếu kiểm thử đầy đủ, ta phải chạy **24 lần kiểm thử khác nhau**, ví dụ:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Client | Server | Browser |
| 1 | Win XP | 2003 | IE8 |
| 2 | Win XP | 2003 | IE9 |
| 3 | Win XP | 2003 | IE10 |
| 4 | Win XP | 2003 | Firefox |
| 5 | Win XP | 2003 | Chrome |
| 6 | Win XP | 2003 | Safari |
| 7 | Win XP | 2008 | IE8 |
| … | … | … | … |
| 24 | Win 7 | 2008 | Safari |

Tuy nhiên, kiểm thử toàn bộ như vậy rất tốn thời gian.

Pairwise testing là kỹ thuật chọn ra một tập hợp nhỏ các trường hợp kiểm thử, sao cho mọi cặp giá trị giữa các yếu tố đều xuất hiện ít nhất một lần.

Tức là:

* Mọi cặp Client–Server đều có trong ít nhất một test case.
* Mọi cặp Client–Browser đều có trong ít nhất một test case.
* Mọi cặp Server–Browser cũng đều được bao phủ.

Tổng số cặp giá trị cần bao phủ (Pairwise):

(Client×Server)+(Client×Browser)+(Server×Browser): (2×2)+(2×6)+(2×6)=4+12+12=28

Có 28 cặp giá trị khác nhau giữa ba tham số.

Pairwise testing đảm bảo mỗi cặp này xuất hiện ít nhất một lần.

Nhờ đó, ta giảm số lượng test case mà vẫn đảm bảo độ bao phủ cao, ta chỉ cần 10 test case mà vẫn bao phủ mọi cặp Client–Server–Browser.

Bảng kết hợp Pairwise Test:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | Client | Server | Browser |
| TC1 | Win XP | 2003 | IE8 |
| TC2 | Win XP | 2008 | Firefox |
| TC3 | Win XP | 2003 | Google Chrome |
| TC4 | Win 7 | 2008 | IE9 |
| TC5 | Win 7 | 2003 | IE10 |
| TC6 | Win 7 | 2008 | Safari |
| TC7 | Win XP | 2008 | IE9 |
| TC8 | Win XP | 2003 | Safari |
| TC9 | Win 7 | 2003 | Google Chrome |
| TC10 | Win 7 | 2008 | IE8 |