**BÁO CÁO THỰC HÀNH TUẦN 6**

MÔN HỌC: CÔNG NGHỆ MỚI TRONG PTUD LỚP: DHHTTT14

Họ và tên sinh viên 1: Nguyễn Hoàng Huy Mã sinh viên:17081321

Họ và tên sinh viên 2: Nguyễn Minh Hiếu Mã sinh viên:17055021

**Tên đề tài:** Tìm hiểu công nghệ QRcode và áp dụng để xây dựng website bán, quản lý sách trực tuyến

**YÊU CẦU**

* Hiện thực chương trình (1/4 mô hình use case)
* Tìm hiểu công nghệ mới

**NỘI DUNG**

1. **Testcase Đăng nhập**

**-Quy trình nghiệp vụ:**

+ Sau khi người dùng truy cập vào hệ thống, và chọn chức năng đăng nhập, hệ thống sẽ hiển thị trang đăng nhập và yêu cầu nhập vào tên email và password.

**-Luồng cơ sở đăng kí:**

+Người dùng truy cập vào hệ thống và chọn chức năng “đăng nhập”.

+Hệ thống hiển thị form đăng nhập.

+Người dùng nhập thông tin đăng nhập.

+Hệ thống kiểm tra thông tin đầy đủ và hợp lệ.

+Hệ thống thông báo đăng nhập thành công.

-Luồng ngoài cơ sở:

+A1-Người dùng không nhập email : Hệ thống nhắc nhỡ người dùng nhập vào email.

+A2-Người dùng không nhập mật khẩu : Hệ thống nhắc nhỡ người dùng nhập vào mật khẩu.

+A3-Cơ sở dữ liệu trục trặc: Nếu cơ sở dữ liệu trục trặc hệ thống hiển thị lỗi và kết thúc usecase.

+A4-Người dùng thoát trong quá trình đăng nhập: Quá trình đăng nhập bị hủy bỏ, usecase đăng nhập kết thúc.

|  |  |  |
| --- | --- | --- |
| TÊN KỊCH BẢN | BẮT ĐẦU LUỒNG | NGOÀI LUỒNG |
| Đăng nhập thành công | Luồng cơ sở |  |
| Không nhập email | Luồng cơ sở | A1 |
| Không nhập mật khẩu | Luồng cơ sở | A2 |
| Cơ sở dữ liệu trục trặc | Luồng cơ sở | A3 |
| Thoát | Luồng cơ sở | A4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TC** | **Các bước** | **Dữ liệu vào** | **Kết quả mong đợi** | **Kết quả thực tế** | **P/F** |
|  | Từ trang chủ hệ thống chọn chức năng ”đăng nhập”  Nhập đầy đủ thông tin : email, mật khẩu nhấn nút ”đăng nhập” | Email:  “hieus@gmail.com”  Mật khẩu :”hieu123” | Phải hiển thị thông báo ”Đăng nhập thành công” | Thông báo ”Đăng nhập thành công” | **P** |
| **1** | Từ trang chủ hệ thống chọn chức năng ”đăng nhập”  Không nhập thông tin email vào ô điền email | Email : “ ” | Phải hiển thị thông báo ”Không được bỏ trống email ” | Thông báo ”Không được bỏ trống email ” | **F** |
| **2** | Từ trang chủ hệ thống chọn chức năng ”đăng nhập”  Không nhập thông tin mật khẩu vào ô điền mật khẩu | Mật khẩu :” ” | Phải hiện thị thông báo “Không được bỏ trống password” | Thông báo “Không được bỏ trống password” | **F** |
| **3** | Từ trang chủ hệ thống chọn chức năng ”đăng nhập” |  | Cơ sở dữ liệu bị trục trặc. Hệ thống thông báo “lỗi hệ thống” |  | **F** |
| **4** | Thoát |  | Thoát khỏi hệ thống |  |  |

1. **Testcase Tìm kiếm**

**-Quy trình nghiệp vụ:**

+ Sau khi người dùng truy cập vào hệ thống, người dùng nhấp chuột vào ô tìm kiếm và nhập từ khóa, sau đó nhấn enter, hệ thống sẽ hiển thị các sản phẩm có liên quan đến từ khóa

**-Luồng cơ sở tìm kiếm:**

+Người dùng truy cập vào hệ thống và nhấp chuột vào ô tìm kiếm

+Người dùng nhập từ khóa vào ô tìm kiếm

+Hệ thống kiểm tra thông tin hợp lệ.

+Hệ thống hiển thị các sản phẩm có liên quan đến từ khóa.

-Luồng ngoài cơ sở:

+A1-Người dùng không nhập từ khóa : Hệ thống nhắc nhỡ người dùng nhập vào từ khóa

+A2-Người dùng nhập từ khóa không hợp lệ : Hệ thống thông báo từ khóa không hợp lệ

+A3-Cơ sở dữ liệu trục trặc: Nếu cơ sở dữ liệu trục trặc hệ thống hiển thị lỗi và kết thúc usecase.

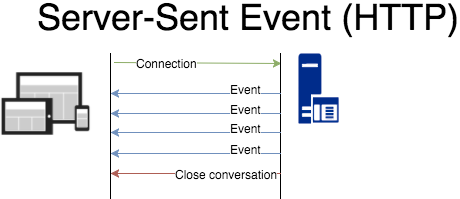
+A4-Người dùng thoát trong quá trình tìm kiếm: Quá trình tìm kiếm bị hủy bỏ, usecase tìm kiếm kết thúc.

|  |  |  |
| --- | --- | --- |
| TÊN KỊCH BẢN | BẮT ĐẦU LUỒNG | NGOÀI LUỒNG |
| Tìm kiếm thành công | Luồng cơ sở |  |
| Không nhập từ khóa | Luồng cơ sở | A1 |
| Từ khóa không hợp lệ | Luồng cơ sở | A2 |
| Cơ sở dữ liệu trục trặc | Luồng cơ sở | A3 |
| Thoát | Luồng cơ sở | A4 |

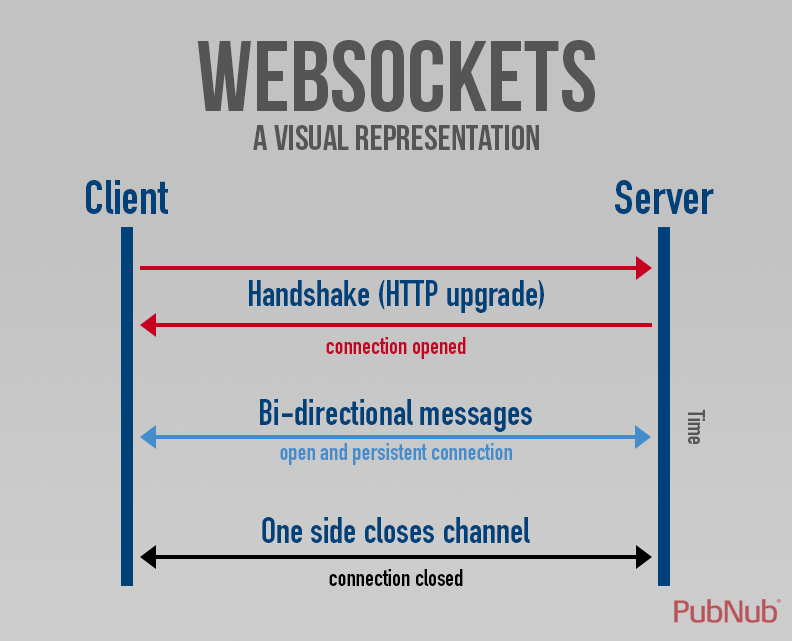
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TC** | **Các bước** | **Dữ liệu vào** | **Kết quả mong đợi** | **Kết quả thực tế** | **P/F** |
|  | Từ trang chủ hệ thống, nhấp vào ô tìm kiếm  Người dùng nhập đúng từ khóa sản phẩm | Từ khóa : Doremon | Hiển thị các sản phẩm có liên quan đến từ khóa | Hiển thị các sản phẩm có liên quan đến từ khóa | **P** |
| **1** | Từ trang chủ hệ thống, nhấp vào ô tìm kiếm  Người dùng không nhập từ khóa sản phẩm | Từ khóa : “ “ | Phải hiển thị thông báo ”Không được bỏ trống” | Phải hiển thị thông báo ”Không được bỏ trống” | **F** |
| **2** | Từ trang chủ hệ thống, nhấp vào ô tìm kiếm  Người dùng nhập từ khóa sản phẩm không hợp lệ | Từ khóa : “ !@#“ | Phải hiển thị thông báo ”Từ khóa không hợp lệ” | Phải hiển thị thông báo ”Từ khóa không hợp lệ” | **F** |
| **3** | Từ trang chủ hệ thống, nhấp vào ô tìm kiếm |  | Cơ sở dữ liệu bị trục trặc. Hệ thống thông báo “lỗi hệ thống” |  | **F** |
| **4** | Thoát |  | Thoát khỏi hệ thống |  |  |

**Tìm hiểu công nghệ mới : Áp dụng real-time**

Sự bùng nổ của dữ liệu thời gian thực (real-time data) đã làm cho nhiều người đặt ra câu hỏi là với những công nghệ đang có, công nghệ nào là sự lựa chọn tốt nhất cho ứng dụng của họ và các nền tảng phân phối dữ liệu thời gian thực (real-time data delivery platform) có thể giúp họ như thế nào. Trong bài viết này, chúng ta sẽ thảo luận sự khác nhau giữa Server-Sent Events và WebSocket.



Với nhiều người Server-Sent Events (SSEs) có thể khá xa lạ. Trong vài năm qua, nó đã bị lu mờ bởi giao thức kết nối với nhiều tính năng hơn - WebSockets. Tuy nhiên, ý tưởng đằng sau SSEs khá quen thuộc: cho phép một ứng dụng web (Web Application) "đăng ký" (subscribe) để theo dõi sự thay đổi của dữ liệu gốc được gửi bởi server, và được thông báo (notified) thông qua các event.



WebSockets cung cấp một giao thức với nhiều tính năng để thực hiện các kết nối hai chiều (bi-directional, full-duplex). Việc có một kênh kết nối 2 chiều khá phổ biến với những thứ như game, các ứng dụng tin nhắn, các công cụ cộng tác, trải nghiệm tương tác... và các trường hợp bạn cần cập nhật dữ liệu theo thời gian thực ở cả hai phía.

Nhưng để thực sự hiểu các công nghệ thời gian thực (real-time technology) này, chúng ta cần biết giới hạn của những người tiền nhiệm, bao gồm:

* **Polling**đã được sử dụng bởi phần lớn ứng dụng ngày nay. Trong mô hình này, ứng dụng client lặp đi lặp lại việc lấy dữ liệu (poll)  từ server. Hoàn toàn dựa trên các nền tảng của web, các ứng dụng này tận dụng giao thức HTTP, cái cho phép lấy dữ liệu trong một mẫu (pattern) request/response. Ứng dụng yêu cầu (request) dữ liệu từ server dựa trên một vài tham số truy vấn (query parametters), và chờ phản hồi (response), cái thường không biết kích thước. Nếu không có dữ liệu, một phản hồi rỗng (empty response) sẽ được trả lại. Vấn đề ở đây, là có một lượng lớn các phản hồi rỗng, hoặc nhiều phản hồi chứa dữ liệu trùng lặp sẽ gây quá tải.
* **Long polling** là một biến thể của polling được thiết kế cho các kịch bản mà server không có dữ liệu mới. Khi nhận một yêu cầu (request HTTP), server sẽ giữ kết nối (hang on) cho đến khi có dữ liệu mới, server sẽ đáp trả  (respond), đóng kết nối và quá trình được lặp lại. Điều này hiệu quả hơn vì dựa trên event, nhưng thực sự vẫn khá tốn kém.

**WebSockets vs Server-Sent Events**

Một kết nối Websocket có thể vừa gửi dữ liệu tới ứng dụng, vừa nhận dữ liệu từ ứng dụng. Một trải nghiệm tương tác thực sự, dựa trên trao đổi dữ liệu 2 chiều trong thời gian thực. Các kết nối Server-Sent Event là một trường hợp khác, nó chỉ có thể đẩy dữ liệu tới ứng dụng và không cung cấp kết nối theo hướng ngược lại.

Trong thực tế, mọi thứ bạn có thể làm với SSE cũng có thể làm với Websockets, vì thế Websockets được hỗ trợ và sử dụng nhiều hơn - số lượng trình duyệt và các thiết bị hỗ trợ Websockets cũng nhiều hơn nhiều so với SSEs.