**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Thiết kế hệ thống bán sách trực tuyến**

**NGUYỄN TRƯỜNG GIANG**

giang.nt166022@sis.hust.edu.vn

**Ngành Công nghệ thông tin**

Chữ ký của GVHD

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | PGS. TS. Nguyễn Nhất Hải |
| **Bộ môn:** | Công nghệ phần mềm |
| **Viện:** | Công nghệ thông tin và truyền thông |
| **HÀ NỘI, 6/2020** | |

**ĐỀ TÀI TỐT NGHIỆP**

Biểu mẫu của Đề tài/khóa luận tốt nghiệp theo qui định của viện, tuy nhiên cần đảm bảo giáo viên giao đề tài ký và ghi rõ họ và tên.

Trường hợp có 2 giáo viên hướng dẫn thì sẽ cùng ký tên.

Giáo viên hướng dẫn

Ký và ghi rõ họ tên

**Lời cảm ơn**

Đầu tiên cho em xin gửi lời cảm ơn đến nhà trường và toàn bộ các giảng viên đã tận tâm giảng dạy chúng em trong suốt quá trình chúng em học tại trường.

Em xin chân thành cảm ơn thầy Nguyễn Nhất Hải đã tận tình chỉ dạy, hướng dẫn em trong suốt quá trình thực hiện đồ án.

Cuối cùng, em xin phép gửi lời cảm ơn tới cha mẹ, gia đình và bạn bè, những người luôn bên cạnh, ủng hộ em, là nguồn động lực, chỗ dựa tinh thần của em trong suốt thời gian học tập và thực hiện đồ án.

**Tóm tắt nội dung đồ án**

1. Vấn đề cần thực hiện:

* Do xu hướng đọc sách giấy đang nổi lên rất mạnh mẽ, hơn nữa trong thời kì công nghệ thông tin bùng nổ như hiện nay, việc mua bán trực tuyến đã trở nên phổ biến.
* Việc quản lý một cửa hàng sách bằng giấy tờ tốn rất nhiều chi phí thời gian và nhân công lại rất dễ nhầm lẫn, ngoài ra việc quảng cáo và tiếp cận độc giả cũng sẽ gặp rất nhiều khó khan nếu không áp dụng công nghệ thông tin vào quá trình marketing, quản lý, mua bán, …
* Từ đó em xin đề xuất làm một hệ thống bán sách trực tuyến hỗ trợ các cửa hàng sách vừa và nhỏ trong việc quản lý danh mục đơn hàng, thanh toán, …

1. Phương pháp thực hiện:
2. Công cụ sử dụng:

* Visual Studio Code
* PgAdmin 4
* Google Chrome
* Docker, Docker-compose

1. Kết quả đồ án
2. Định hướng phát triển
3. Kiến thức, kỹ năng đã đạt được:

* Nắm rõ hơn về nodejs, reactjs, docker-compose, docker và cách triển khai ứng dụng web container hóa.

Sinh viên thực hiện

Ký và ghi rõ họ tên

# GIỚI THIỆU ĐỀ TÀI

## Đặt vấn đề:

Với sự phát triển của khoa học công nghệ, hội nhập kinh tế, cách thức lựa chọn và mua hàng của người tiêu dùng đang có những thay đổi mạnh mẽ từ cách lựa chọn và mua trực tiếp chuyển sang trực tuyến. Đi cùng với đó, đọc sách đang trở thành một thói quen, sở thích của rất nhiều người.

Do nhu cầu của xã hội ngày càng tăng, các cửa hàng/chuỗi cửa hàng sách cũng phải thay đổi để đáp ứng các nhu cầu đó. Việc quản lý và tiếp thị sách một cách thủ công đã không còn là phương pháp tối ưu nếu các cửa hàng muốn cạnh tranh và mở rộng số lượng khách hàng cũng như doanh thu. Để có thể quản lý dễ dàng và nhanh chóng thông tin của nhân viên, khách hàng, sản phẩm, đơn hàng,… và tối ưu hóa việc lựa chọn, tìm kiếm, mua sách của khách hàng, cần có một hệ thống hỗ trợ. Và đây cũng là lý do em chọn đề tài “Xây dựng hệ thống bán sách trực tuyến”.

## Mục tiêu và phạm vi đề tài

Một website bán sách online cần đảm bảo được các chức năng cơ bản cho người dung:

* Cho phép người dùng dễ dàng lựa chọn và tìm kiếm sách cũng như xem thông tin chi tiết sách
* Đánh giá sách
* Thêm sách vào giỏ hàng
* Đặt hàng và thanh toán trực tuyến
* Quản lý thông tin bản thân như địa chỉ giao hàng, email, số điện thoại, đơn hàng.

Với người quản trị hệ thống:

* Một tài khoản admin có đầy đủ quyền trong trang admin. Tài khoản có thể thực hiện tất các các chức năng trong trang quản trị. Khi có một nhân viên mới, admin kĩ thuật sẽ cung cấp tài khoản mới cho nhân viên đó, xác định rõ quyền hạn của tài khoản để nhân viên chỉ thao tác đúng chức năng được phân quyền hạn. Nếu tài khoản xảy ra vấn đề gì thì nhân viên báo với admin kĩ thuật quản lý lại tài khoản đó.

Với nhân viên:

* Được phép sử dụng trang admin với các quyền hạn được người quản trị hệ thống cấp

## Định hướng giải pháp:

Với các chức năng đã nêu trong phần “Mục tiêu và phạm vi đề tài”, hệ thống sẽ có 2 trang riêng biệt:

* Trang đầu tiên là trang người dung, tại đây khách hàng có thể đăng ký/đăng nhập và tìm kiếm, lựa chọn sách, mua sách và quản lý thông tin cá nhân của mình.
* Trang thứ 2 là trang admin, chỉ có nhân viên và người quản trị hệ thống mới đăng nhập vào được. Trang này có nhiệm vụ quản lý Danh mục sách, thể loại, tác giả, khuyến mại, …và cung cấp các biểu đồ cũng như báo cáo về đơn hàng, doanh thu.

Để thực hiện được các yêu cầu trên thì thực tế có rất nhiều công cụ có thể sử dụng xây dựng nên hệ thống. Trong đồ án này em chọn làm Ứng dụng Web trên nền tảng Nodejs và Reactjs theo mô hình MVC với phần view được tách riêng viết bằng Reactjs – đây là một thư viện Javascript được phát triển bởi Facebook cực kì mạnh mẽ cho việc tạo các ứng dụng SPA trên nền tảng web và được các tên tuổi hàng đầu lựa chọn như Facebook, Netflix, Microsoft (Outlook), Hubspot, Twilio,… Về phần back-end, em lựa chọn Nodejs – một môi trường chạy Javascript trên server với ưu điểm tốc độ xử lý I/O cao,cơ chế xử lý bất đồng bộ mạnh mẽ, dễ học, dễ sử dụng, được cộng đồng hỗ trợ thường xuyên với rất nhiều thư viện làm tăng tốc quá trình phát triển ứng dụng. Bên cạnh đó, việc lựa chọn mô hình MVC giúp việc mô hình hóa các đối tượng vào bài toán trở nên rõ rang và dễ hiểu hơn.

Để lưu trữ và quản lý dữ liệu, trong đồ án này em sử dụng hệ quản trị cơ sở dữ liệu quan hệ MySQL. MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở được phát triển bởi Tập đoàn Oracle. MySQL được sử dụng trong rất nhiều hệ thống thương mại điện tử, các kho dữ liệu với ưu điểm ràng buộc quan hệ chặt chẽ, hỗ trợ truy vấn Transaction, hỗ trợ atomic query tránh việc sai sót dữ liệu.

## Bố cục đồ án

* Chương 2 trình bày phân tích và khảo sát các hệ thống bán sách online đang có trên thị trường và rút ra các chức năng cần có cho một hệ thống bán sách trực tuyến
* Chương 3 phân tích thiết kế hệ thống và lựa chọn các công nghệ, công cụ phù hợp để thiết kế hệ thống phù hợp với tình hình thực tế cũng như phù hợp với khả năng của bản thân
* Chương 4 đi vào chi tiết cụ thể về thiết kế hệ thống và kiểm thử
* Chương 5 nói về những khó khăn nổi bật và phương pháp giải quyết trong quá trình làm đồ án
* Chương 6 kết luận lại tất cả kết quả đạt được trong quá trình làm đồ án, kinh nghiệm rút ra, các kĩ năng được cải thiện/trau dồi và định hướng phát triển tương lai cho đồ án

# KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Khảo sát và khởi tạo dự án:

Sau khi khảo sát các hệ thống bán sách trực tuyến khác cũng như phân tích các trang thương mại điện tử, em đã rút ra các điểm sau:

### Đối tượng người dùng:

* Hệ thống bán sách trực tuyến hướng đến mục đích hỗ trợ tốt hơn cho khách hàng cũng như người quản lý/nhân viên cửa hàng trong việc lựa chọn, mua bán và quản lý cửa hàng sách.

### Yêu cầu phi chức năng:

#### Yêu cầu vận hành:

* Hệ thống phải tương thích với các trình duyệt phổ biến như Chrome, Firefox, …

#### Yêu cầu hiệu năng:

* Những tương tác của người dùng đối với hệ thống (tìm kiếm, thêm sửa, xóa…) không vượt quá 4 giây.
* Hệ thống quản lý liên tục, có thể hoạt động 24 giờ trong ngày và 365 ngày trong một năm.

#### Yêu cầu bảo mật:

* Chỉ có người quản trị hệ thống mới có quyền xem và chỉnh sửa thông tin cá nhân của người dùng và nhân viên.
* Chỉ có người quản trị hệ thống hoặc nhân viên mới có thể quản lý sách, thể loại,…
* Khách hàng chỉ có thể xem thông tin của chính mình

### Yêu cầu chức năng:

#### Về phía người dùng (khách hàng):

* Chọn sách, tìm kiếm sách, lọc sách theo thể loại, giá tiền, tác giả,…
* Xem chi tiết sách
* Đánh giá sách
* Quản lý giỏ hàng
* Đặt mua/thanh toán
* Quản lý thông tin cá nhân (địa chỉ giao hàng, số điện thoại, Đơn hàng, các đánh giá đã tạo)

#### Về phía nhân viên:

* Quản lý sách:
  + Xem danh sách tất cả sách (có phân trang)
  + Tìm kiếm sách
  + Sửa thông tin sách
  + Xóa sách
  + Thêm sách mới
  + Xem chi tiết thông tin sách
* Quản lý thể loại sách:
  + Xem danh sách tất cả thể loại sách (có phân trang)
  + Tìm kiếm thể lọai sách
  + Sửa thông tin thể loại sách
  + Xóa thể loại sách
  + Thêm thể loại sách mới
  + Xem chi tiết thông tin thể loại sách
  + Tự động tính toán sách bán chạy dựa theo số lượng bán được trong những khoảng thời gian nhất đinh.
* Quản lý đánh giá:
  + Xem danh sách tất cả đánh giá (có phân trang)
  + Tìm kiếm đánh giá
  + Sửa đánh giá
  + Xóa đánh giá
  + Duyệt đánh giá
  + Xem chi tiết đánh giá
  + Trả lời đánh giá
* Quản lý tác giả:
  + Xem danh sách tất cả tác giả (có phân trang)
  + Tìm kiếm tác giả
  + Sửa thông tin tác giả
  + Xóa tác giả
  + Thêm tác giả mới
  + Xem chi tiết thông tin tác giả
* Quản lý tuyển tập sách:
  + Xem danh sách tất cả tuyển tập sách (có phân trang)
  + Tìm kiếm tuyển tập sách
  + Sửa thông tin tuyển tập sách
  + Xóa tuyển tập sách
  + Thêm tuyển tập sách mới
  + Xem chi tiết thông tin tuyển tập sách
  + Thêm sách vào tuyển tập
  + Xóa sách khỏi tuyển tập
* Quản lý đơn hàng:
  + Xem danh sách tất cả đơn hàng (có phân trang)
  + Tìm kiếm đơn hàng
  + Sửa thông tin đơn hàng
  + Xóa đơn hàng
  + Xem chi tiết thông tin đơn hàng
* Quản lý nhà xuất bản:
  + Xem danh sách tất cả nhà xuất bản(có phân trang)
  + Tìm kiếm nhà xuất bản
  + Sửa thông tin nhà xuất bản
  + Xóa nhà xuất bản
  + Xem chi tiết thông tin nhà xuất bản
* Quản lý giảm giá:
  + Xem danh sách tất cả giảm giá (có phân trang)
  + Tìm kiếm giảm giá
  + Sửa thông tin giảm giá
  + Xóa giảm giá
  + Thêm giảm giá
  + Xem chi tiết thông tin giảm giá
  + Thêm sách vào giảm giá
  + Xóa sách khỏi giảm giá
* Quản lý nội dung trang web:
  + Cập nhật các banner thông tin khuyến mại
  + Cập nhật thông tin liên hệ

#### Người quản trị hệ thống:

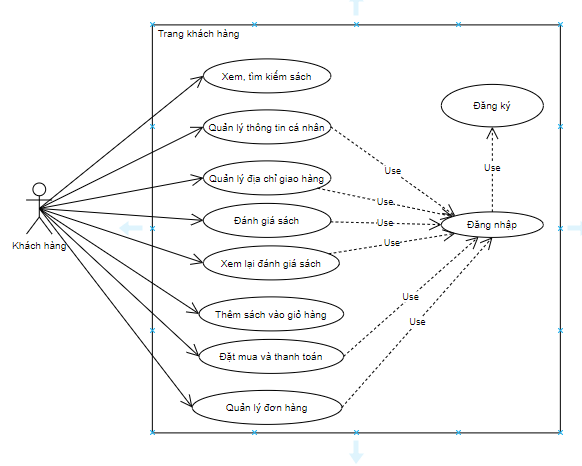
* Ngoài tất cả chức năng của nhân viên, còn thêm chức năng quản lý người dùng:
  + Thêm người dùng
  + Xem danh sách người dùng
  + Xóa người dùng
  + Vô hiệu hóa/Kích hoạt người dùng
  + Sửa thông tin người dùng
  + Tìm kiếm người dùng

## Phân tích và thiết kế hệ thống:

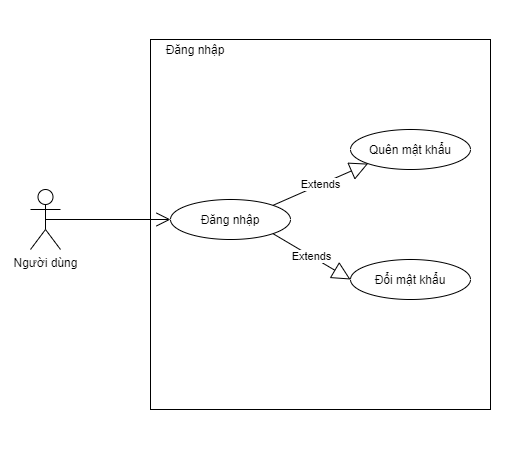
### Mô hình hóa chức năng:

#### Biểu đồ usecase và đặc tả ca sử dụng:

* + 1. Phía khách hàng:
* Biểu đồ usecase tổng quan trang khách hàng:



* Ca sử dụng đăng nhập:
  + Biểu đồ usecase:



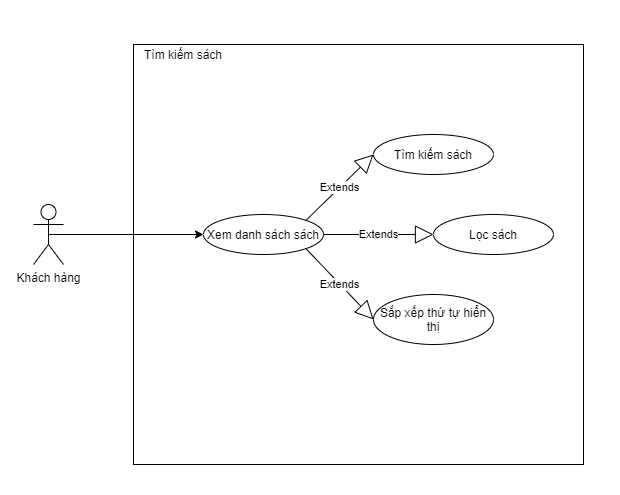
* + Đặc tả ca sử dụng đăng nhập:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Người dùng đăng nhập | ID: 1 | | Mức quan trọng: Cao |
| Tác nhân chính: Người dùng (khách hàng) | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Người dùng - Đăng nhập vào hệ thống. | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Người dùng đăng nhập vào hệ thống | | | |
| Kích hoạt: Người dùng vào trang web và thực hiện đăng nhập  Kiều: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Người dùng * Bao gốm: * Mở rộng: Đổi mật khẩu, quên mật khẩu * Khái quát hóa: | | | |
| Luồng sự kiện chính:   1. Người dùng chọn mục đăng nhập. 2. Hệ thống đưa Người dùng tới giao diện đăng nhập 3. Người dùng nhập thông tin đăng nhập (ID, Password,…) 4. Hệ thống tiến hành kiểm tra thông tin đăng nhập 5. Nếu thông tin đăng nhập hợp lệ hệ thống cho phép Người dùng đăng nhập vào hệ thống 6. Nếu thông tin đăng nhập không hợp lệ hệ thống thông báo lỗi đăng nhập và yêu cầu Người dùng nhập lại thông tin đăng nhập 7. Người dùng lặp lại các bước 3 đến 6 cho đến khi đăng nhập thành công | | | |
| Luồng sự kiện con: | | | |
| Luồng sự kiện tương đương/ngoại lệ:  3a. Người dùng gọi chức năng đổi mật khẩu  6a. Hệ thông hỏi Người dùng có muốn thực thi chức năng quên mật khẩu hay không  6a-1. Nếu có chuyển Người dùng đến trang quên mật khẩu  6a-2. Nếu không tiếp tục luồng sự kiện chính.  7a. Người dùng thoát khỏi trang web | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Người dùng đổi mật khẩu | ID: 2 | | Mức quan trọng: trung bình |
| Tác nhân chính: Người dùng | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Người dùng - Đổi mật khẩu. | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Người dùng đổi mật khẩu. | | | |
| Kích hoạt: Người dùng chọn tùy chọn đổi mật khẩu.  Kiểu: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Người dùng * Bao gồm: * Mở rộng * Khái quát hóa: | | | |
| Luồng sự kiện chính:   1. Người dùng chọn tùy chọn đổi mật khẩu. 2. Hệ thống đưa Người dùng đến trang đổi mật khẩu 3. Người dùng nhập mật khẩu hiện tại và mật khẩu mới. 4. Người dùng xác nhận mật khẩu mới 5. Hệ thống kiểm tra mật khẩu hiện tại và mật khẩu mới   Nếu mật khẩu hiện tại không trùng hoặc xác nhận mật khẩu không trùng trở lại bước 3.  Nếu trùng đi đến bước 6   1. Tiến hành đổi mật khẩu | | | |
| Luồng sự kiện con: | | | |
| Luồng sự kiện tương đương/ngoại lệ:  6a. Người dùng trở lại trang đăng nhập hoặc thoát khỏi trang web. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Người dùng quên mật khẩu | ID: 3 | | Mức quan trọng: Trung bình |
| Tác nhân chính: Người quản trị | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Người dùng – quên mật khẩu | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Người dùng đặt lại mật khẩu trong trường hợp quên mật khẩu | | | |
| Kích hoạt: Người dùng chọn tùy chọn quên mật khẩu  Kiểu: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Người dùng * Bao gồm: * Mở rộng: * Khái quát hóa | | | |
| Các luồng sự kiện chính:   1. Người dùng chọn tùy chọn quên mật khẩu 2. Hệ thống đưa Người dùng tới trang quên mật khẩu 3. Người dùng nhập username và email vào form. 4. Hệ thống gửi mail cho Người dùng đường dẫn đến trang đặt lại mật khẩu mới 5. Người dùng nhập mật khẩu mới và xác nhận mật khẩu mới 6. Tiến hành đổi mật khẩu | | | |
| Các luồng sự kiện con: | | | |
| Các luồng sự kiện ngoại lệ/tương đương:  5a. Xác nhận mật khẩu không đúng  6a. Đang tiến hành đổi mật khẩu thì Người dùng thoát khỏi trang. | | | |

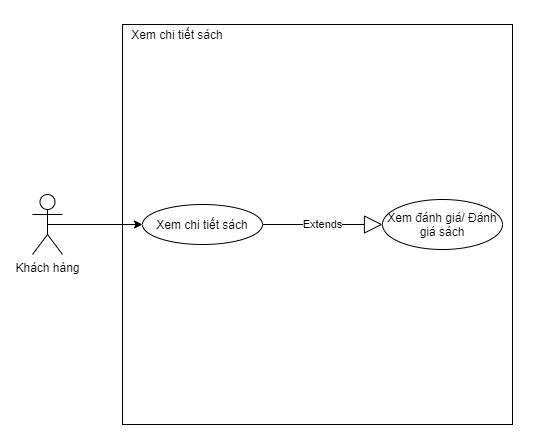
* Biểu đồ usecase ca sử dụng chọn và tìm kiếm sách:
  + Biểu đồ usecase:



* + Đặc tả ca sử dụng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Người dùng xem danh sách sách | ID: 4 | | Mức quan trọng: Cao |
| Tác nhân chính: Người dùng | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Người dùng – Xem danh sách sách | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Người dùng xem danh sách sách. | | | |
| Kích hoạt: Khi người dùng chọn xem danh sách sách  Kiểu: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Người dùng * Bao gồm: * Mở rộng: Tìm kiếm sách, Lọc sách, Sắp xếp lại thứ tự hiển thị * Khái quát hóa | | | |
| Các luồng sự kiện chính:   1. Người dùng chọn vào giao diện cửa hàng sách trong trang khách hàng 2. Hệ thống đưa Người dùng tới trang cửa hàng sách 3. Người dùng xem danh sách sách có trong giao diện:  * Nếu người dùng muốn xem một trang khác, luồng sự kiện con S-1 được thực thi * Nếu người dùng muốn tạo các bộ lọc để tùy chọn danh sách có thể xem, luồng sự kiện con S-2 được thực thi * Nếu người dùng muốn sắp xếp lại thứ tự hiển thị, luồng sự kiện con S3 được thực thi | | | |
| Các luồng sự kiện con:  S-1. Phân trang:   * Người dùng cuộn xuống cuối trang mà vẫn chưa tìm được sách mong muốn * Người dùng sẽ nhấn vào các nút tiến/lùi/lên đầu/tới cuối hoặc bấm vào số trang muốn tới   S-2. Lọc sách:   * Người dùng chọn các bộ lọc sách khác nhau (theo giá, tác giả, thể loại) trên giao diện để tùy chọn danh sách có thể xem   S-3. Sắp xếp lại thứ tự hiển thị:   * Người dùng chọn một trong cách tiêu chí trên để sắp xếp lại thứ tự hiển thị của danh sách | | | |
| Các luồng sự kiện ngoại lệ/tương đương:  3a. Người dùng tìm kiếm sách  3a1. Người dùng ấn click vào biểu tượng tìm kiếm trên giao diện  3a2. Hệ thống hiển thị giao diện tìm kiếm  3a3. Người dùng nhập tên sách/tác giả/NXB để tìm kiếm sách  3b. Người dùng click vào một quyển sách:  3b1. Người dùng click vào một quyển sách  3b2. Hệ thống đưa người dùng tới trang chi tiết sách | | | |

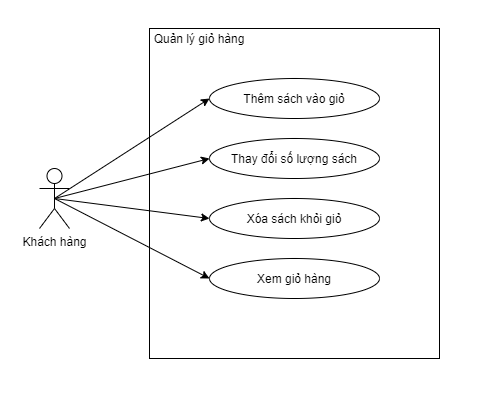
* Ca sử dụng xem chi tiết sách:



* + Đặc tả ca sử dụng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Người dùng xem chi tiết sách | ID: 5 | | Mức quan trọng: Cao |
| Tác nhân chính: Người dùng | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Người dùng – Xem chi tiết sách | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Người dùng xem chi tiết sách. | | | |
| Kích hoạt: Khi người dùng chọn xem chi tiết sách  Kiểu: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Người dùng * Bao gồm: * Mở rộng: Xem đánh giá , Đánh giá sách * Khái quát hóa | | | |
| Các luồng sự kiện chính:   * + - 1. Người dùng click chọn vào một cuốn sách trên giao diện       2. Hệ thống đưa Người dùng tới trang chi tiết sách       3. Người dùng xem các thông tin chi tiết về cuốn sách hiện tại * Nếu người dùng muốn xem đánh giá sách, luống S-1 được thực thi. * Nếu người dùng muốn để lại đánh giá sách, luống S-2 được thực thi. | | | |
| Các luồng sự kiện con:  S-1. Xem đánh giá:   * Người dùng đi tới khu vực đánh giá và xem đánh giá của người dùng khác.   S-2. Đánh giá sách:   * Người dùng để lại bình luận và điểm đánh giá của mình. | | | |
| Các luồng sự kiện ngoại lệ/tương đương:  3a. Người dùng thêm sách đó vào giỏ hàng | | | |

* Ca sử dụng quản lý giỏ hàng:
  + Biểu đồ usecase:



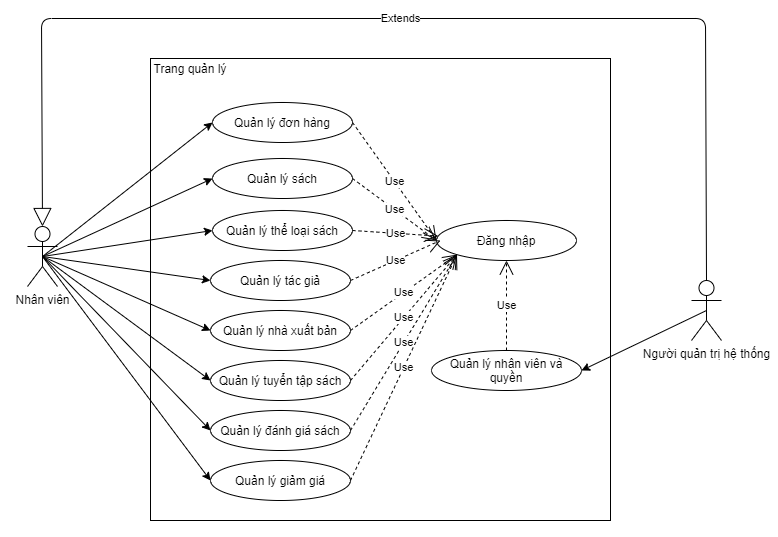
* + Đặc tả ca sử dụng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Quản lý giỏ hàng | ID: 6 | | Mức quan trọng: Cao |
| Tác nhân chính: Người dùng | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Người dùng – Quản lý giỏ hàng | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Người dùng xem chi tiết sách. | | | |
| Kích hoạt: Khi người dùng chọn xem chi tiết sách  Kiểu: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Người dùng * Bao gồm: Thêm sách vào giỏ, Thay đổi số lượng sách, Xóa sách khỏi giỏ hàng, Xem giỏ hàng * Mở rộng * Khái quát hóa | | | |
| Các luồng sự kiện chính:   * + - 1. Người dùng thêm sách vào giỏ hàng.       2. Hệ thống đưa sách khách đã chọn vào giỏ hàng       3. Sau đó tùy vào ý muốn của khách hàng, một trong các luồng sự kiện con sau sẽ được thực thi: * Nếu người dùng muốn xem lại giỏ hàng luồng sự kiện S-1 được thực thi. * Nếu người dùng muôn thay đổi số lượng sách trong giỏ hàng, luồng sự kiện S-2 được thực thi. * Nếu người dùng muốn thay xóa sách khỏi giỏ hàng, luồng sự kiện S-3 được thực thi. | | | |
| Các luồng sự kiện con:  S-1. Xem giỏ hàng:   * Người dùng click vào icon giỏ hàng trên header   S-2. Thay đổi số lượng sách trong giỏ:   * Người dùng click vào icon giỏ hàng trên header * Thay đổi số lượng sách theo mong muốn.   S-3. Xóa sách khỏi giỏ hàng.   * Người dùng click vào biểu tượng xóa tương ứng. | | | |
| Các luồng sự kiện ngoại lệ/tương đương:  3a. Người dùng chuyển đến trang đặt mua và thanh toán | | | |

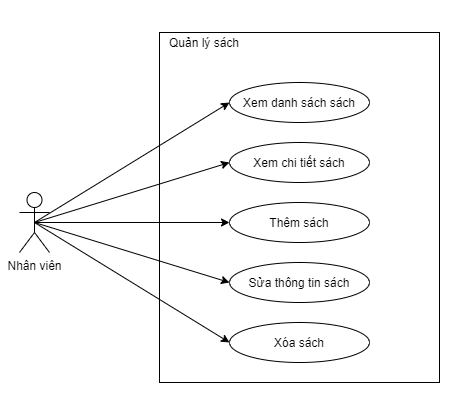
* Ca sử dụng Đặt mua và thanh toán:
  + Đặc tả ca sử dụng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Đặt mua và thanh toán | ID: 7 | | Mức quan trọng: Cao |
| Tác nhân chính: Người dùng | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Người dùng – Đặt mua và thanh toán | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Người dùng đặt mua và thanh toán | | | |
| Kích hoạt: Khi người dùng click nút Đặt mua trên giao diện giỏ hàng  Kiểu: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Người dùng * Bao gồm: * Mở rộng * Khái quát hóa | | | |
| Các luồng sự kiện chính:   1. Người dùng click vào nút Đặt mua trên giao diện giỏ hàng 2. Hệ thống đưa người dùng đến trang Đặt mua và thanh toán 3. Người dùng thực hiện lần lượt các bước   B1. Đăng nhập (nếu người dùng đã đăng nhập thì bước này được tự động bỏ qua)  B2. Chọn địa chỉ giao hàng  B3. Chọn hình thức thanh toán  B4. Click nút thanh toán.   1. Hệ thống tiến hành xử lý đơn hàng 2. Hệ thống thông báo thành công hoặc thất bại  * Nếu xử lý thành công, hệ thống sẽ reset giỏ hàng, hiển thị thành công cho người dùng. * Nếu xử lý thất bại, hệ thống sẽ thông báo thất bại cho người dùng. | | | |
| Các luồng sự kiện con: | | | |
| Các luồng sự kiện ngoại lệ/tương đương:  3a. Người dùng chưa có địa chỉ giao hàng hoặc muốn thêm địa chỉ giao hàng mới.   * Người dùng click thêm địa chỉ giao hàng, hệ thống sẽ cho phép người dùng tạo địa chỉ giao hàng mới | | | |

* + 1. Phía nhân viên và người quản trị hệ thống:
* Biểu đồ usecase tổng quan:



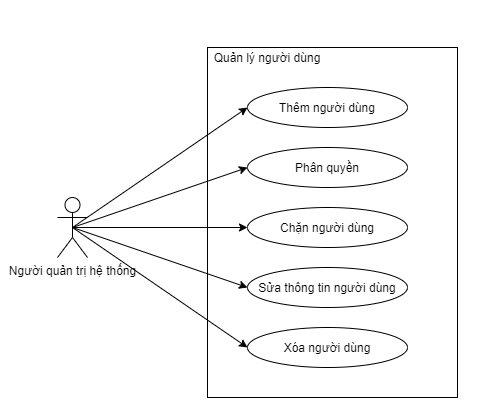
* Ca sử dụng Quản lý sách:
  + Biểu đồ usecase:



* + Đặc tả ca sử dụng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Quản lý sách | ID: 8 | | Mức quan trọng: Cao |
| Tác nhân chính: Nhân viên | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Nhân viên – Quản lý sách | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Nhân viên quản lý sách | | | |
| Kích hoạt: Khi nhân viên chọn vào giao diện quản lý sách  Kiểu: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Nhân viên * Bao gồm: Xem danh sách sách, xóa sách, sửa thông tin sách, thêm sách * Mở rộng * Khái quát hóa | | | |
| Các luồng sự kiện chính:   * + - 1. Nhân viên vào giao diện quản lý sách       2. Nhân viên chọn tính năng muốn sử dụng: * Nếu nhân viên muốn xem danh sách sách, thì trên giao diện quản lý sách đã có sẵn. * Nếu nhân viên muốn xem chi tiết sách, click vào tên sách muốn xem * Nếu nhân viên muốn sửa thông tin sách, luồng sự kiện con S-1 được thực thi * Nếu nhân viên muốn xóa sách, luồng sự kiện con S-2 được thực thi | | | |
| Các luồng sự kiện con:  S-1. Nhân viên sửa thông tin sách:   * Nhân viên click vào tên sách muốn sửa trong màn hình danh sách sách. * Hệ thống đưa chuyển hướng nhân viên tới trang chi tiết sách. * Nhân viên sửa thông tin sách theo nhu cầu.   S-2. Nhân viên xóa sách:   * Nhân viên click chọn sách sử dụng các checkbox ở đầu mỗi dòng trên bảng danh sách sách. * Click nút xóa. * Hệ thống tiến hành xóa sách. | | | |
| Các luồng sự kiện ngoại lệ/tương đương:  2a. Nếu có lỗi xảy ra khi sửa hoặc xóa sách, hệ thống báo lỗi và nhân viên thực hiện lại thao tác. | | | |

* Ca sử dụng quản lý người dùng:
  + Biểu đồ usecase:



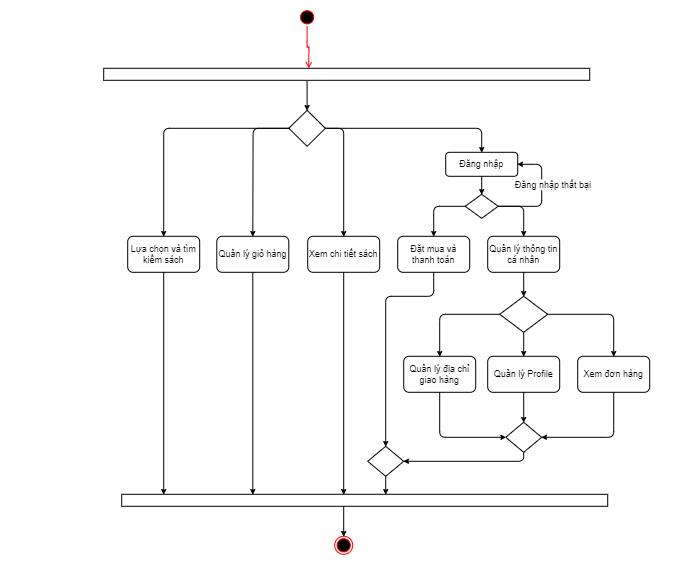
* + Đặc tả ca sử dụng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên ca sử dụng: Quản lý người dùng | ID: 9 | | Mức quan trọng: Cao |
| Tác nhân chính: Người quản trị hệ thống | | Kiểu ca sử dụng: Chi tiết, thiết yếu | |
| Các nhân tố và mối quan tâm: Người quản trị hệ thống – Quản lý người dùng | | | |
| Mô tả ngắn gọn: Ca sử dụng này mô tả cách Người quản trị quan lý người dùng | | | |
| Kích hoạt: Khi nhân viên chọn vào giao diện quản lý sách  Kiểu: Ngoại | | | |
| Các mối quan hệ:   * Liên quan: Người quản trị hệ thống * Bao gồm: Xem danh sách người dùng, xóa người dùng, sửa thông tin người dùng, thêm người dùng * Mở rộng * Khái quát hóa | | | |
| Các luồng sự kiện chính:   1. Người quản trị hệ thống vào giao diện quản lý người dùng 2. Người quản trị hệ thống chọn tính năng muốn sử dụng:  * Nếu Người quản trị hệ thống muốn xem danh sách người dùng, thì trên giao diện quản lý người dùng đã có sẵn. * Nếu Người quản trị hệ thống muốn xem chi tiết người dùng, click vào tên người dùng muốn xem * Nếu Người quản trị hệ thống muốn sửa thông tin người dùng, luồng sự kiện con S-1 được thực thi * Nếu Người quản trị hệ thống muốn xóa người dùng, luồng sự kiện con S-2 được thực thi | | | |
| Các luồng sự kiện con:  S-1. Người quản trị hệ thống sửa thông tin người dùng:   * Người quản trị hệ thống click vào tên người dùng muốn sửa trong màn hình danh sách người dùng. * Hệ thống đưa chuyển hướng Người quản trị hệ thống tới trang chi tiết người dùng. * Người quản trị hệ thống sửa thông tin người dùng theo nhu cầu.   S-2. Người quản trị hệ thống xóa người dùng:   * Người quản trị hệ thống click chọn người dùng sử dụng các checkbox ở đầu mỗi dòng trên bảng danh sách người dùng. * Click nút xóa. * Hệ thống tiến hành xóa người dùng. | | | |
| Các luồng sự kiện ngoại lệ/tương đương:  2a. Nếu có lỗi xảy ra khi sửa hoặc xóa người dùng, hệ thống báo lỗi và Người quản trị hệ thống thực hiện lại thao tác. | | | |

* Các ca sử dụng quản lý khác cũng tương tự.

#### Biểu đồ hoạt động:

* + 1. Về phía trang khách hàng:
* Biểu đồ hoạt động tổng quan trang khách hàng:



# CÔNG NGHỆ SỬ DỤNG

## Giới thiệu các công nghệ đã sử dụng:

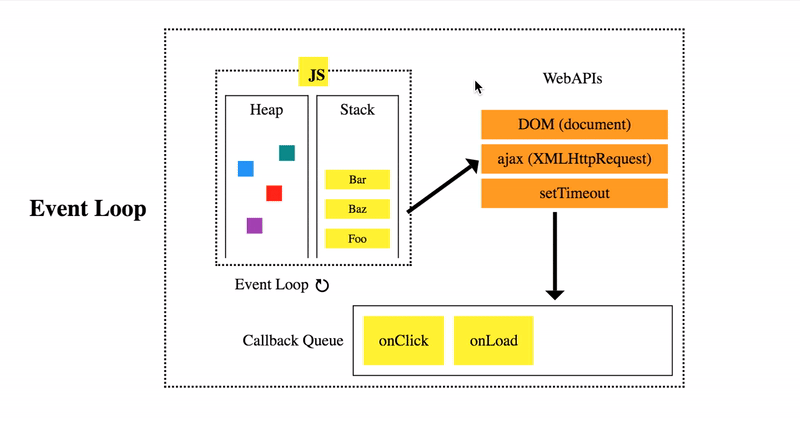
### Node.js – môi trường chạy JavaScript server-side:

1. Tổng quan:

* Node.js là một môi trường chạy Javascript server-side đa nền tảng, mã nguồn mở, được xây dựng trên engine Javascript V8 của Chrome. Node.js được viết bởi Ryan Dahl vào năm 2009.
* Node.js đi cùng với NPM (Node Package Manager) – trình quản lý gói cho Javascript với một GUI client và một database để lưu trữ các gói gọi là npm registry.
* Node.js có các đặc điểm:
  + Đơn luồng (Single threaded): Tại một thời điểm chỉ có một luồng xử lý cho một ứng dụng Node.js
  + Non-blocking(Không khóa luồng) và bất đồng bộ(Asynchronous): Ngôn ngữ chính được dùng để xây dựng nên Node.js là Javascript, có cơ chế xử lý bất đồng bộ rất mạnh, những tác vụ bất đồng bộ như đọc ghi file/database không khóa luồng xử lý.
  + Hướng sự kiện (Event-driven): Dùng để chỉ việc ta thiết lập các hành động để thực thi sau khi một hành động bất đồng bộ được xử lý xong.

1. Cách thức hoạt động:

* Như đã đề cập ở trên, Node.js không phải là một ngôn ngữ lập trình, nó là một môi trường chạy Javascript server-side side được xây dựng trên engine Javascript V8 của Chrome. Khi thực thi code Javascript, Node.js sẽ chuyển toàn bộ mã nguồn Javascript cho V8 để thực thi và nhận lại kết quả, nó cũng cung cấp các tính năng mở rộng được viết chủ yếu bằng C/C++ cho Javascript như đọc ghi file/database,…
* Trong javascript, khi một hàm được thực thi, nó sẽ được đưa vào một vùng đặc biệt trong bộ nhớ gọi là CallStack để xử lý, khi xử lý xong, nó sẽ được xóa khỏi CallStack. Khi hàm đó là một hàm bất đồng bộ, nó sẽ được đưa vào CallStack, sau đó sẽ được đưa vào Node.js APIs (viết chủ yêu bằng C/C++) để xử lý (những tác vụ như đọc ghi file), lúc này, Node.js coi như hàm đó đã được xử lý xong và đưa ra khỏi CallStack để thực thi các câu lệnh tiếp theo. Khi hàm đó được xử lý xong trong Node.js APIs, Callback của nó sẽ được đưa vào Callback Queue để chờ xử lý. Lúc này, Event Loop – một cấu trúc đặc biệt trong Nodejs sẽ liên tục kiểm tra Callback Queue và CallStack, chỉ khi CallStack rỗng thì Callback mới được lấy ra để đưa vào CallStack thực thi.



1. Ưu điểm:

* Cơ chế bất đồng bộ rất mạnh nên dù chỉ có 1 luồng xử lý chính nhưng tốc độ xử lý với các tác vụ I/O bound cực kì cao.
* Đơn giản, dễ học.
* Xây dựng ứng dụng Web bằng Node.js cực kì nhanh chóng, chỉ cần chưa đến 10 dòng code là đã có một Web server chạy được.
* Kho thư viện khổng lồ cùng với cộng đồng người dùng lớn.
* Được các ông lớn trong giới công nghệ tin dùng như Netflix, Trello, Paypal,…
* Có khả năng xử lý các tác vụ realtime như streaming với tốc độ cao.
* Cấu trúc thư mục đơn giản

1. Nhược điểm:

* Vì Node.js có tính đơn luồng nên khả năng xử lý các tác vụ nặng về CPU (CPU bound) không cao.
* Khó tiếp cận với những người không quen với lập trình bất đồng bộ hướng sự kiện.
* Khi ứng dụng lớn dần sẽ rất khó quản lý nếu tổ chức thư mục không tốt.

### Reactjs – Thư viện Javascript để xây dựng giao diện người dùng.

* 1. Tổng quan:
* Reactjs là một thư viện Javascript mã nguồn mở để xây dựng giao diện người dùng trên nền tảng web.
* Được phát triển và duy trì bởi Facebook và cộng đồng lập trình viên trên toàn thế giới
* Reactjs có các đặc điểm:
  + Declarative
  + Component-based: Một ứng dụng Reactjs được chia thành các thành phần (Component) và có thể tái sử dụng để xây dựng nên các thành phần phức tạp hơn.
  + Stateful: Mỗi component lưu trữ và quản lý state của riêng nó, có thể hiểu state như là một trạng thái mà các component dùng để tính toán việc hiển thị UI, do đó, khi state thay đổi thì UI cũng thay đổi.
* Chạy trên môi trường Node.js
  1. Cách thức hoạt động của Reactjs:
* Ứng dụng được viết bằng Reactjs được chia thành các thành phần nhỏ hơn, mỗi thành phần quản lý state của riêng nó.
* Các thành phần có các chu trình sống riêng, ta có thể override lại chúng để thực hiện các tác vụ mong muốn.
* Mỗi thành phần có một hàm render() lấy các giá trị đầu vào để trả ra UI cần hiển thị.
* Dựa trên nguyên lý Virtual DOM (DOM ảo): Trong React, với mỗi đối tượng DOM, sẽ có một đối tượng DOM ảo (được tạo ra từ hàm render()), DOM ảo này lưu trữ toàn bộ thông tin cần thiết để tạo ra DOM thật và được lưu trữ dưới dạng cây như DOM thật. Khi có state thay đổi, React sẽ cập nhật lại toàn bộ cây DOM ảo rồi so sánh với cây DOM ảo được tạo từ lần update trước đó, qua đó, React biết được phần nào có thay đổi và sẽ chỉ cập nhật phần thay đổi tương ứng đó lên DOM thật từ đó chỉ cập nhật lại phần thay đổi tương ứng lên UI chứ không dựng lại toàn bộ cậy DOM thật do đó hiệu năng của cả chương trình được tăng lên.
* Các thành phần cha có thể truyền state của mình xuống cho thành phần con dưới dạng các props, các props này không thể bị thay đổi trực tiếp bởi các thành phần con.
  1. Ưu điểm:
* Tốc độ rất cao do được thiết kế dựa trên nguyên lý Virtual DOM.
* Được hỗ trợ và cập nhật thường xuyên từ Facebook và cộng đồng lập trình viên
* Được các gã khổng lồ công nghệ sử dụng như Facebook, Netflix, Slack, Microsoft, Trello,…
* Hệ thống thư viện rất lớn
  1. Nhược điểm:
* Không có sẵn cơ chế routing.
* Khi ứng dụng lớn việc quản lý state sẽ khó khăn.

### GraphQL – Ngôn ngữ truy vấn và thao tác dữ liệu cho API.

1. Tổng quan:

* GraphQL là một ngôn ngữ truy vấn và trên Web API được phát triển bởi Facebook vào từ năm 2012 và được mã nguồn mở vào năm 2015. Nó cũng là một môi trường chạy để thực thi các truy vấn.
* GraphQL cung cấp khả năng cho người dùng API (API consumer) khả năng lấy những dữ liệu mà họ cần – nghĩa là người dùng API được phép quyết định xem họ sẽ lấy những gì chứ không phải như Web API truyền thống, nơi mà dữ liệu trả về được quyết định bởi phía cấp API, từ đó tăng hiệu quả và giảm lưu lượng mạng.
* Các API của GraphQL được định nghĩa trong một file schema dưới dạng các field chứ không phải endpoint. Các ứng dụng dùng GraphQL API chỉ có một endpoint duy nhất, do đó giảm được độ phức tạp khi thiết kế API.

1. Ưu điểm:

* Linh hoạt và hiệu quả do phía dùng API có thể quyết định được họ sẽ lấy những data gì, tất nhiên là vẫn dưới sự kiểm soát của người cấp API.
* Có khả năng lấy được cả dữ liệu quan hệ trong 1 query, từ đó tăng tốc độ truy vấn và giảm lưu lượng mạng.
* Hỗ trợ hành động Subscription, khi bên dùng API đăng ký vào một API, khi có thay đổi, dữ liệu sẽ tự được đẩy xuống.
* Có nhiều thư viện hỗ trợ.

1. Nhược điểm:

* Phức tạp và khó sử dụng với người mới bắt đầu.

### MySQL:

1. Tổng quan:

* MySQL là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u" \o "Hệ quản trị cơ sở dữ liệu) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F" \o "Phần mềm nguồn mở) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X), [Unix](https://vi.wikipedia.org/wiki/Unix), [FreeBSD](https://vi.wikipedia.org/wiki/FreeBSD), [NetBSD](https://vi.wikipedia.org/w/index.php?title=NetBSD&action=edit&redlink=1), [Novell NetWare](https://vi.wikipedia.org/wiki/Novell_NetWare), [SGI Irix](https://vi.wikipedia.org/w/index.php?title=SGI_Irix&action=edit&redlink=1), [Solaris](https://vi.wikipedia.org/wiki/Solaris), [SunOS](https://vi.wikipedia.org/w/index.php?title=SunOS&action=edit&redlink=1),..
* MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).
* MySQL được sử dụng cho việc bổ trợ NodeJs, [PHP](https://vi.wikipedia.org/wiki/PHP), [Perl](https://vi.wikipedia.org/wiki/Perl), và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

1. Phương thức hoạt động của MySQL

* Câu lệnh SQL (Transact-SQL) là ngôn ngữ MySQL dùng để trao đổi dữ liệu giữa máy Client và máy cài MySQL Server. Có 4 loại ngôn ngữ MySQL sử dụng:
  + Ngôn ngữ định nghĩa dữ liệu (Data Definition Language)
  + Ngôn ngữ truy vấn dữ liệu (Structured Query Language)
  + Ngôn ngữ thao tác dữ liệu (Data Manipulation Language)
  + Ngôn ngữ điều khiển dữ liệu (Data Control Language)
* MySQL lưu trữ database trong các bảng tập hợp những dữ liệu liên quan đến nhau. Một bảng gồm nhiều **hàng** (bản ghi của bảng) và **cột** (trường dữ liệu).
* MySQL cung cấp các câu lệnh phong phú để kết nối, yêu cầu và lấy dữ liệu phục vụ cho trang web như:
  + Chỉnh sửa cơ bản các hàng trong 1 quan hệ: chèn, xóa, cập nhật
  + Thêm, xóa và sửa đổi các đối tượng trong của cơ sở dữ liệu.
  + Điều khiển việc truy cập tới cơ sở dữ liệu và các đối tượng của cơ sở dữ liệu nhằm đảm bảo tính bảo mật, tính nhất quán và sự ràng buộc của cơ sở dữ liệu.

1. Cơ chế quản lý sử dụng riêng của MySQL

* Mỗi người dùng đều có 1 **username** (tên truy cập) và **password** (mật khẩu) để đăng nhập và đăng xuất đến cơ sở dữ liệu. Người dùng có thể mỗi người quản lý 1 hoặc nhiều cơ sở dữ liệu khác nhau.

1. Ưu điểm:

* **Dễ sử dụng**: **MySQL** là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.
* **Độ** **bảo mật cao**:  **MySQL** rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều nhiều tính năng bảo mật thậm chí là ở cấp cao.
* **Đa tính năng**: **MySQL** hỗ trợ rất nhiều **chức năng SQL** được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.
* **Khả năng mở rộng và mạnh mẽ**: **MySQL** có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.
* **Nhanh chóng**: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

1. Nhược điểm:

* **Giới hạn**: Theo thiết kế, **MySQL** không có ý định làm tất cả và nó đi kèm với các hạn chế về chức năng mà một vào ứng dụng có thể cần.
* **Độ tin cậy**: Cách các chức năng cụ thể được xử lý với **MySQL** (ví dụ tài liệu tham khảo, các giao dịch, kiểm toán,…) làm cho nó kém tin cậy hơn so với một số hệ quản trị cơ sở dữ liệu quan hệ khác.
* **Dung lượng hạn chế**: Nếu số bản ghi lớn dần lên thì việc truy xuất dữ liệu của bạn là khá khó khăn, khi đó chúng ta sẽ phải áp dụng nhiều biện pháp để tăng tốc độ truy xuất dữ liệu như là chia tải database này ra nhiều server, hoặc **tạo cache MySQL**

### UML – Ngôn ngữ mô hình hóa thống nhất:

* 1. Tổng quan:
* UML (Unified Modeling Language) là ngôn ngữ dành cho việc đặc tả, hình dung, xây dựng và làm tài liệu của các hệ thống phần mềm.
* UML tạo cơ hội để viết thiết kế  hệ thống, bao gồm những khái niệm như tiến trình nghiệp vụ và các chức năng của hệ thống.
* Cụ thể, nó hữu dụng cho những ngôn ngữ khai báo, giản đồ cơ sở dữ liệu, thành phần phần mềm có khả năng tái sử dụng.
* UML được phát triển bởi Rational Rose và một số nhóm cộng tác, nó nhanh chóng trở thành một trong những ngôn ngữ chuẩn để xây dựng hệ thống phần mềm hướng đối tượng (Object-Oriented).
  1. Mục tiêu của UML:
* UML cung cấp cho người dùng một ngôn ngữ mô hình hoá trực quan sẵn sàng để dùng và có ý nghĩa:
  + Cho phép phát triển và trao đổi những mô hình mang nhiều ý nghĩa.
* Cung cấp khả năng mở rộng và chuyên môn hoá để mở rộng những khái niệm cốt lõi.
* Độc lập với ngôn ngữ lập trình chuyên biệt và các tiến trình phát triển.
* Cung cấp nền tảng về sự hiểu biết ngôn ngữ mô hình hoá.
* Khuyến khích và hỗ trợ sự phát triển của các công cụ hướng đối tượng.
* Hỗ trợ những khái niệm phát triển cấp độ cao như collaboration, framework, pattern and component.
* Tích hợp một cách tốt nhất với thực tiễn.
  1. Các loại biểu đồ trong UML:
* **Biểu Đồ Lớp (Class Diagram)**
  + Class diagram là xương sống của hầu như tất cả các phương pháp hướng đối tượng, bao gồm cả UML. Chúng mô tả các cấu trúc tĩnh của hệ thống.
* **Biểu Đồ Gói (Package Diagram)**
  + Package Diagram là tập hợp các class diagram. Các package diagram thiết lập mối quan hệ giữa các pakage, trong đó pakage là những nhóm phần tử của hệ thống có mối quan hệ liên quan đến nhau.
* **Biểu Đồ Chức Năng**
  + Uses case diagram đưa ra cách nhìn bao quát (từ trên xuống) cách sử dụng của hệ thống cũng như cách nhìn hệ thống từ bên ngoài.
  + Biểu đồ này hiển thị những chức năng của hệ thống hoặc các lớp và tương tác của hệ thống với thế giới bên ngoài như thế nào.
  + Uses case diagram được dùng trong quá trình phân tích hệ thống để nắm bắt được yêu cầu của hệ thống và hiểu được sự hoạt động của hệ  thống.
* **Biểu Đồ Tương Tác**
  + Collaboration Diagram cung cấp về cách nhìn sự tương tác hoặc mối quan hệ có cấu trúc giữa các đối tượng trong mô hình hiện thời. Collaboration Diagram bao gồm các đối tượng, liên kết và thông báo.
  + Sử dụng mô hình như là một phương tiện chính để mô tả những tương tác và cách giải quyết của các hành vi trong hệ thống.
* **Biểu Đồ Tiến Trình**
  + Sequence diagram mô tả sự tương tác của các lớp trong trình tự về thời gian. Những mô hình này được liên kết với phương pháp case (tình huống).
  + Sequence diagram hiển thị cho bạn từng bước những sự kiện xảy ra trong phương pháp case.
* **Biểu Đồ Trạng Thái (Statechart Diagram)**
  + Statechart Diagram mô tả những hành động của các lớp và đối tượng riêng lẻ,  mô tả trình tự những trạng thái mà các đối tượng sẽ đi qua.
* **Biểu Đồ Hoạt Động (Activity Diagram)**
  + Activity Diagram mô tả tiến trình xử lý và trình tự những hành động trong tiến trình xử lý, Trông nó giống như biểu đồ tiến trình (flowchart) bởi vì nó mô tả dòng làm việc từ hoạt động sang hoạt động và từ hoạt động sang trạng thái.
  + Khi xây dựng activity diagram nó giúp bạn có thể hiểu được toàn bộ tiến trình hoạt động. Nó rất hữu dụng khi bạn mô tả những tiến trình song song hoặc mô tả  một vài tương tác trong use case.
* **Biểu Đồ Thành Phần (Component Diagram)**
  + Component Diagram cho chúng ta cách nhìn vật lý của mô hình thực tế. Nó thể hiện rõ cho chúng ta thấy sự cấu tạo và sự phụ thuộc giữa các thành phần của phần mềm bao gồm mã nguồn, mã nhị phân (binary code) và những thành phần có khả năng thực thi.
* **Biểu Đồ Triển Khai (Deployment Diagram)**
  + Deployment Diagrams mô tả các tài nguyên vật lý trong hệ thống, bao gồm các nút (node), thành phần và kết nối. Mỗi mô hình chỉ bao gồm một deployment diagram hiển thị ánh xạ giữa những tiến trình xử lý tới thiết bị phần cứng.