

ĐỒ ÁN 2

Trang web bán vé xe

|  |  |
| --- | --- |
| **Nhóm 4** | |
| **Thành viên** | **Đỗ Khải Hưng** (Nhóm trưởng) |
| **Giảng viên** | **Nguyễn Nam Long** |

**Mục lục**

[1. Mở đầu 3](#_Toc113030178)

[2. Giới thiệu 4](#_Toc113030179)

[2.1. Đưa ra vấn đề 4](#_Toc113030180)

[2.2. Hệ thống hiện tại 4](#_Toc113030181)

[2.3. Hệ thống đề nghị 4](#_Toc113030182)

[2.4. Công nghệ 4](#_Toc113030183)

[3. Phân tích yêu cầu người dùng 5](#_Toc113030184)

[3.1. Yêu cầu phi chức năng 5](#_Toc113030185)

[**3.1.1.** **Phía khách hàng** 5](#_Toc113030186)

[3.1.1. Phía quản lí 5](#_Toc113030187)

[3.2. Yêu cầu chức năng 5](#_Toc113030188)

[3.2.1. Nhóm người dùng của hệ thống 5](#_Toc113030189)

[3.2.2. Phân tích chức năng 7](#_Toc113030190)

[3.2.2.1. Phía khách hàng 7](#_Toc113030191)

[3.2.2.2. Phía quản lí 30](#_Toc113030192)

[4. Thiết kế hệ thống 42](#_Toc113030193)

[4.1. Sơ đồ quan hệ thực thể 42](#_Toc113030194)

[4.2. Sơ đồ trang web 49](#_Toc113030195)

[5. Kết luận 51](#_Toc113030196)

[6. Phân công công việc 52](#_Toc113030197)

# Mở đầu

Ngày nay Internet đã trở thành dịch vụ phổ biến và thiết yếu và có ảnh hưởng sâu rộng tới thói quen, sinh hoạt, giải trí của nhiều người. Sự thị hiếu của việc di chuyển bằng xe khách và việc lập các trang web cho các nhà xe để tăng khả năng tiếp cận tới khách hàng ngày càng cần thiết. Trước nhu cầu đó, cùng với yêu cầu môn học, nhóm chúng em quyết định chọn đề tài xây dựng website bán và mua vé xe trực tuyến.

Với đề tài và môn học này, nhóm xin chân thành cảm ơn sự giúp đỡ tận tình của Thầy Nguyễn Nam Long. Song, do còn nhiều hạn chế đề tài nhóm xây dựng không tránh khỏi những thiếu sót. Rất mong được thầy cô và các bạn đóng góp ý kiến để website ngày càng hoàn thiện và được đưa vào sử dụng.

**Trân trọng cảm ơn!**

# Giới thiệu

## Đưa ra vấn đề

Công ty Lokiss là một công ty chuyên vận chuyển nổi tiếng và quen thuộc với rất nhiều khách hàng hiện nay. Với phương châm “Vừa lòng khách đến, vui lòng khách đi”, công ty không ngừng nỗ lực nhằm mang lại sản phẩm cũng như sự phục vụ tốt nhất cho khách hàng. Nhu cầu mở rộng thị trường cũng như quảng bá nhà xe và nâng cao chất lượng, tăng doanh thu là vấn đề đáng quan tâm của công ty. Vì vậy ban quản lí của công ty có mong muốn xây dựng 1 website giới thiệu nhà xe và bán vé trực tuyến. Website gồm 2 phần khách hàng và admin với giao diện bắt mắt đầy đủ các tính năng giúp thuận tiện cho việc mua sắm đối với khách hàng cũng như quản lí đối với nhân viên.

## Hệ thống hiện tại

Trước đây, khi mà mọi thứ được làm bằng thủ công thì với mô hình kinh doanh B2C truyền thống, việc mở rộng quy mô hoàn toàn bắt cách xây dựng cửa hàng mới và quản lí hoàn toàn bằng thủ công trên giấy, Excel, Word. Từ đó dẫn đến việc quản lí lỏng lẻo, xảy ra nhiều trục trặc.

## Hệ thống đề nghị

Hệ thống website tới đây không những là cơ hội để công ty chuyển mình để bắt kịp thời đại mà còn là một bước đột phá trong việc quảng bá đến toàn bộ khác hàng ở khắp mọi nơi. Đồng thời việc quản lí được thực hiện tường minh, minh bạch, tiện lợi, giúp tối đa hóa hiệu suất.

## Công nghệ

1. Laravel 8.83.12(PHP Framework)
2. 5.7.33 - MySQL Community Server.
3. HTML5/CSS3/JavaScript ES6.
4. Boostrap 5.
5. ChartJs, MaatwebsiteExcel, BarryvdhDomPDF

# Phân tích yêu cầu người dùng

## Yêu cầu phi chức năng

1. **Phía khách hàng**

- Giao diện dễ tiếp cận, tông màu sáng, cam xanh trắng đơn giản  
- Không phải tải lại trang nhiều lần  
- Đã responsive tạm ổn   
- Đánh số trang.  
- Slider, hover, chạy chữ, loader.

### Phía quản lí

- Giao diện dễ tiếp cận, tông màu sáng   
- Không phải tải lại trang nhiều lần  
- Đánh số trang

## Yêu cầu chức năng

### Nhóm người dùng của hệ thống

* **Khách hàng chưa có tài khoản:**
  + [Đăng ký](#Signup).
  + [Đăng nhập](#signin).
  + [Xem tất cả chuyến đi.](Báo%20cáo%20đồ%20án%202.docx" \l "signin)
  + [Sắp xếp, lọc sản phẩm](#sort_filter) .
  + [Tìm kiếm sản phẩm](#search).
  + [Thêm sản phẩm vào giỏ hàng.](#add_to_cart)
  + [Xem thêm (Hóa đơn, Collection, Search).](#load_more)
* **Khách hàng có tài khoản:**
  + [Xem tất cả sản phẩm](#display_products).
  + [Sắp xếp, lọc sản phẩm](#sort_filter) .
  + [Tìm kiếm sản phẩm](#search).
  + [Thêm sản phẩm vào giỏ hàng.](#add_to_cart)
  + [Xem thêm (Hóa đơn, Collection, Search).](#load_more)
  + [Đăng nhập](#signin).
  + [Lưu mật khẩu](#savepassword).
  + [Đăng xuất](#signout).
  + [Quên mật khẩu](#forgotpassword).
  + [Xem, chỉnh sửa thông tin cá nhân, đổi mật khẩu](#update_account).
  + [Bình luận, đánh giá sản phẩm](#reviews)
  + [Xem, cập nhật sản phẩm trong giỏ hàng](#view_and_update_cart).
  + [Đặt hàng](#checkout).
    - [Xem đơn hàng đã đặt, hủy đơn](#view_and_delete_bill).
* **Employee:**
  + [Đăng nhập](#signin)
  + Lưu mật khẩu
  + Xem tất cả các sản phẩm
  + Tìm kiếm các đối tượng theo tên
  + Xem cụ thể từng sản phẩm (kèm theo sơ đồ)
  + Sửa thông tin sản phẩm
  + Thêm sản phẩm
  + Xem tất cả các đơn hàng
  + Sửa tình trạng đơn
  + Đổi mật khẩu
  + [Đăng xuất](#signout)
* **Employer:**
  + [Đăng nhập](#signin)
  + Lưu mật khẩu
  + Xem tất cả các sản phẩm
  + Tìm kiếm các đối tượng theo tên
  + Xem cụ thể từng sản phẩm (kèm theo sơ đồ)
  + Sửa thông tin sản phẩm
  + Thêm sản phẩm
  + Xóa sản phẩm
  + Xem tất cả các đơn hàng và tình trang đơn
  + Xem tất cả các nhà sản xuất
  + Xem cụ thể từng nhà sản xuất (kèm theo sơ đồ)
  + Sửa thông tin nhà sản xuất
  + Thêm nhà sản xuất
  + Xem tất cả nhân viên
  + Xem cụ thể từng nhân viên
  + Sửa thông tin nhân viên
  + Thêm nhân viên
  + Xóa nhân viên
  + [Đăng xuất](#signout)
* **Admin**
  + [Đăng xuất.](#signout)
  + [Đăng nhập](#signin).

### Phân tích chức năng

#### Phía khách hàng

* **Đăng nhập**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Tất cả đối tượng |
| ***Mô tả*** | Đăng nhập |
| ***Kích hoạt*** | Người dùng ấn vào nút “Đăng nhập” trên thanh header, chuyển hướng khi ấn vào trang dành cho người dùng có tài khoản. |
| ***Đầu vào*** | Tên đăng nhập  Mật khẩu |
| ***Trình tự xử lý*** | 1. Kiểm tra đã đăng nhập chưa hoặc có Cookie token không:  - Đã đăng nhập:   Chuyển hướng tới trang chủ.   * + Có token và token hợp lệ: * Tạo phiên đăng nhập, lưu mã tài khoản vào phiên. * Chuyển hướng tới trang chủ.   + Ngược lại: chuyển sang form đăng nhập.  1. Dùng tên đăng nhập, mật khẩu 2. Lấy thông tin từ form, kiểm tra không được để trống, nếu trống thông báo lỗi. 3. Kiểm tra tên đăng nhập và mật khẩu có khớp với 1 tài khoản nào không:   - Đúng:   * Tạo phiên đăng nhập, lưu mã tài khoản vào phiên. * Chuyển hướng tới trang chủ.   - Sai: chuyển hướng về trang đăng nhập và thông báo lỗi.   1. Dùng bên thứ ba để đăng nhập 2. Chuyển đến trang của bên thứ 3 để yêu cầu người dùng cấp quyền 3. Kiểm tra các trường nhận được từ bên thứ 3:  * Đúng:   + Cập nhật hoặc tạo tài khoản với các thông tin vừa nhận   + Tạo phiên đăng nhập, lưu thông tin tài khoản vào phiên. * Sai: chuyển hướng về trang đăng nhập và thông báo lỗi.  1. Điều hướng về trang chủ. |
| ***Đầu ra*** | * Đúng: Chuyển hướng trực tiếp tới trang chủ. * Sai: Hiển thị trang đăng nhập và thông báo thất bại. |
| ***Lưu ý*** | * Check vào checkbox lưu mật khẩu để lưu tài khoản cho lần sau. |

* **Đăng ký**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng |
| ***Mô tả*** | Đăng nhập |
| ***Kích hoạt*** | Người dùng ấn vào nút “Đăng ký” trên thanh header, “Create account ” tại trang đăng nhập |
| ***Đầu vào*** | * Tên * Tỉnh/thành phố * Quận/huyện * Email * Mật khẩu |
| ***Trình tự xử lý*** | 1. Chuyển sang form đăng ký 2. Lấy thông tin từ form, validate thông tin ở request:  * Điền thiếu: thông báo lỗi trực tiếp * Mật khẩu không đúng yêu cầu : thông báo lỗi trực tiếp * Tài khoản đã tồn tại: thông báo lỗi. * Đăng ký thành công: Gửi mail đăng ký thành công và thông báo ở trang đăng nhập. |
| ***Đầu ra*** | * Đúng: Gửi mail đăng ký thành công và thông báo ở trang đăng nhập. * Sai: Thông báo lỗi. |
| ***Lưu ý*** |  |

* **Đăng xuất**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Tất cả đối tượng (đã đăng nhập) |
| ***Mô tả*** | Đăng xuất |
| ***Kích hoạt*** | Người dùng ấn vào nút “Đăng xuất” trên thanh header |
| ***Đầu vào*** | * Tài khoản đã đăng nhập |
| ***Trình tự xử lý*** | 1. Chuyển đến trang SignOut 2. Xóa các Session, Cookie liên quan đến tài khoản 3. Chuyển lại về trang chủ. |
| ***Đầu ra*** | Chuyển về trang chủ, người dùng ở trạng thái chưa có tài khoản |
| ***Lưu ý*** |  |

* **Lưu mật khẩu**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng |
| ***Mô tả*** | Lưu mật khẩu |
| ***Kích hoạt*** | Người dùng check vào checkbox “Lưu mật khẩu” trong form đăng nhập |
| ***Đầu vào*** | Tài khoản đã hợp lệ , nếu không thì thông báo lỗi (phần đăng nhập) |
| ***Trình tự xử lý*** | 1. Kiểm tra xem có check vào checkbox hay chưa   - Có: Thực hiện tiếp.  - Không: Bỏ qua.   1. Lưu vào cookie token trên trình duyệt người dùng. 2. Cập nhật thông tin token mới vào database. |
| ***Đầu ra*** | Những lần đăng nhập trên trình duyệt đó sẽ không cần phải điền form ở trang đăng nhập mà khi ấn vào nút “Đăng nhập ” sẽ được cập nhật tài khoản. |
| ***Lưu ý*** | Thời gian duy trì: 30 ngày kể từ ngày đăng nhập có lưu mật khẩu. |

* **Quên mật khẩu**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng (chưa đăng nhập) |
| ***Mô tả*** | Quên mật khẩu |
| ***Kích hoạt*** | Người dùng ấn vào nút quên mật khẩu ở trang đăng nhập |
| ***Đầu vào*** | Tên tài khoản  Mật khẩu mới  Mã code |
| ***Trình tự xử lý*** | 1. Kiểm tra xem tài khoản có tồn tại hay không, nếu không thông báo lỗi. 2. Kiểm tra số lần cập nhật có quá 2 lần/ ngày hay không, nếu có thông báo lỗi 3. Tạo code lưu vào bảng tỏng database và gửi qua gmail thời gian hiệu lực 90s (80s đợi và 10s delay) 4. Nếu code đúng hash password mới và lưu vào database 5. Cập nhật thông tin thời gian vào bảng quên mật khẩu |
| ***Đầu ra*** | Đúng : Mật khẩu được cập nhật, thông báo cho người dùng  Sai: Thông báo lỗi |
| ***Lưu ý*** | Các lỗi bao gồm:   1. Không tồn tại email 2. Một ngày gửi quá 2 lần. 3. Mật khẩu mới không khớp 4. Mã code sai 5. Không đủ thông tin 6. Quá giờ   Messages trả về : Kiểm tra khớp mật khẩu bằng JavaScript còn lại từ API PHP trả về.  Thời gian xử lí hơi lâu nên thêm hiệu ứng loader. |

* **Chỉnh sửa thông tin cá nhân.**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng (đã đăng nhập) |
| ***Mô tả*** | Cập nhật thông tin cá nhân |
| ***Kích hoạt*** | Ấn vào nút “Cập nhật thông tin cá nhân ” ở trang account |
| ***Đầu vào*** | Tên người dùng  Giới tính  Ngày sinh  Số điện thoại  Địa chỉ(địa chỉ cụ thể và tỉnh thành) |
| ***Trình tự xử lý*** | 1. Kiểm tra có xót bất kỳ trường dữ liệu nào không, nếu có thông báo lỗi. 2. Cập nhật thông tin tài khoản vào database. |
| ***Đầu ra*** | Đúng : Thông tin người dùng được cập nhật, thông báo cho người dùng  Sai: Thông báo lỗi |
| ***Lưu ý*** | Messages trả về : Từ API PHP và hiển thị lỗi bằng hàm JavaScript. |

* **Đổi mật khẩu**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng (đã đăng nhập) |
| ***Mô tả*** | Đổi mật khẩu |
| ***Kích hoạt*** | Ấn vào nút “Đổi mật khẩu ” ở trang account.  (Ngay dưới các thông tin tài khoản) |
| ***Đầu vào*** | Mật khẩu cũ  Mật khẩu mới |
| ***Trình tự xử lý*** | 1. Kiểm tra có trống bất kỳ trường dữ liệu nào không, nếu có thông báo lỗi. 2. Kiểm tra hai mật khẩu mới có trùng khớp hay không , nếu không thông báo lỗi. 3. Kiểm tra mật khẩu cũ có đúng hay không, nếu không thông báo lỗi. 4. Đổi mật khẩu được hash mới và lưu vào database đồng thời thông báo thành công |
| ***Đầu ra*** | Đúng : Mật khẩu người dùng được cập nhật, thông báo cho người dùng  Sai: Thông báo lỗi |
| ***Lưu ý*** | Messages trả về : Từ API PHP và hiển thị lỗi bằng hàm JavaScript.  (2 mật khẩu trùng khớp được bắt sự kiện từ JavaScript và PHP) |

* **Xem tất cả sản phẩm**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng |
| ***Mô tả*** | Xem các sản phẩm trưng bày và chi tiết từng sản phẩm |
| ***Kích hoạt*** | Xem tại trang chủ hoặc trang Collection (được dẫn tới qua thanh header ) |
| ***Đầu vào*** | Không có |
| ***Trình tự xử lý*** | * Xem từ trang chủ :   Chia làm 3 phần: Nam, Nữ, Trẻ em. Các bước thực hiện:   1. Lấy dữ liệu được inner join đầy đủ thông tin từ database 2. Tách xuất thành 3 phần giới tính. Mỗi loại không quá 12 sản phẩm lưu vào 3 mảng riêng biệt. 3. Hiển thị sản phẩm và tạo event hover thì sẽ hiện những sản phẩm thuộc giới tính tương ứng.  * Xem từ trang Collection:  1. Lấy ra giới tính của trang là phần cuối trên đường dẫn, nếu không tồn tại thì quay lại trang chủ. 2. Truy vấn tất cả sản phẩm thuộc giới tính ở trang. 3. Truy vấn tất cả thể loại sản phẩm 4. Tạo một mảng 2 chiều là kết quả theo từng thể loại giày từ kết quả truy vấn tất cả sản phẩm (kỹ thuật mảng đánh dấu) và tối đa ban đầu là 8 sản phẩm 1 trang (có nút load-more). 5. Hiện thị các sản phẩm theo thể loại nếu không có sản phẩm hiển thị “không có sản phẩm”. 6. Hiển thị các sản phẩm thuộc thể loại với thể loại khi được ấn (JavaScript). 7. Nếu khi đến trang thông qua header đã chọn thể loại sản phẩm thì filter luôn thể loại đó (PHP in ra JavaScript)  * Trang product:  1. Kết nối database và kiểm tra id sản phẩm có tồn tại hay không, nếu không quay về trang chủ. 2. Lấy các thông tin sản phẩm được truy vấn. 3. Truy vấn đến bảng bình luận để lấy thông tin rating và comment. 4. Hiển thị ra người dùng. |
| ***Đầu ra*** | Đúng : Các sản phẩm được trưng bày ( khi thêm thể loại sẽ không bị lỗi)  Sai: Sai đường dẫn, quay về trang chủ |
| ***Lưu ý*** | * Vì trang Collection lấy đuôi sau .php (sở thích người code :> ) nên phải tạo 2 file header .php để import các đường dẫn tương đối cho phù hợp, dẫn tới việc tạo thêm 2 file nữa khi ở dạng Mobile/Tablet. Đồng thời cũng xuất hiện 2 file header.js và headerCo.js (Vì đường dẫn API là tương đối). * Có thể mở rộng :  1. Ở trang product có thể thêm phần các sản phẩm cùng thể loại. 2. Thêm nhiều ảnh cho 1 sản phẩm.  * Nhận xét: Giao diện trang product xấu. |

* **Sắp xếp, lọc sản phẩm**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng |
| ***Mô tả*** | Sắp xếp, lọc sản phẩm |
| ***Kích hoạt*** | Chức năng ở trang Collection.  (Ngay dưới các thể loại sản phẩm) |
| ***Đầu vào*** | Sắp xếp theo giá, lọc theo nhà sản xuất |
| ***Trình tự xử lý*** | Phía client:   1. Tạo 1 hàm filter bao gồm cả sắp xếp theo giá và lọc theo nhà sản xuất (JavaScript). 2. Call API đến Server với offset sản phẩm = 0 3. Nhận dữ liệu ở dạng json rồi chuyển về object sau đó hiển thị cho người dùng   Phía server:   1. Kiểm tra tính hợp lệ của các trường trong request, nếu có thì kết nối database 2. Tạo câu lệnh truy vấn, bao gồm các thông tin: Thể loại, giới tính, nhà sản xuất, sắp xếp theo giá. 3. Nếu không có sản phẩm nào thì trả về phản hồi 0 ngược lại trả về dữ liệu dưới dạng json. |
| ***Đầu ra*** | Đúng : Hiển thị sản phẩm ra người dùng |
| ***Lưu ý*** | * Luôn có 1 thẻ cha là thẻ chứa các sản phẩm thuộc thể loại được chọn. Do đó phải tạo 1 hàm filter thể loại được chọn trong đó lại gọi đến hàm sắp xếp và lọc. * Sắp xếp và lọc nên gộp với load-more làm 1 file duy nhất. * Có thể mở rộng tính năng: * Thêm biểu tưọng “New” cho những sản phẩm mới. * Sắp xếp theo ngày sản xuất. * Sắp xếp theo số sản phẩm bán chạy. * Vì sau này có sắp xếp và lọc nên trang collection có thể không cần render dữ liệu ra ban đầu |

* **Tìm kiếm sản phẩm.**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng |
| ***Mô tả*** | Tìm kiếm sản phẩm theo tên |
| ***Kích hoạt*** | Biểu tượng kính lúp trên thanh header |
| ***Đầu vào*** | Thông tin được nhập từ bàn phím |
| ***Trình tự xử lý*** | Live search:   1. Nhận dữ liệu thông qua event keyup và call API đến server 2. Phía server kiểm tra tính hợp lệ của request (tránh sql injection). 3. Truy vấn với giới hạn 7 sản phẩm 4. Trả dữ liệu về client và bên khách hàng hiển thị ra bằng JavaScript   Trang search:  Tương tự như live search nhưng hiện thị ra trang riêng (Bao gồm tính năng load-more để xem đầy đủ tất cả sản phẩm). |
| ***Đầu ra*** | Hiển thị sản phẩm ra người dùng, nếu không có sản phẩm nào thì thông báo là không có sản phẩm (trang search) hoặc không gợi ý(live search). |
| ***Lưu ý*** | Dữ liệu ở hiện thị ở live search kèm đường link sản phẩm để người dùng click luôn  Tính năng có thể bổ sung:   * Tạo riêng biệt 2 phần live search là: Từ gợi ý được tìm kiếm nhiều nhất và danh sách sản phẩm khớp. |

* **Thêm sản phẩm vào giỏ hàng**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng |
| ***Mô tả*** | Thêm sản phẩm vào giỏ hàng |
| ***Kích hoạt*** | Người dùng ấn vào nút “Add to cart” cạnh sản phẩm |
| ***Đầu vào*** | Mã sản phẩm |
| ***Trình tự xử lý*** | 1. Mở phiên xử lý 2. Kiểm tra mã sản phẩm truyền lên, nếu không có thì không thêm. 3. Kiểm tra trong phiên giỏ hàng đã tồn tại mã sản phẩm này chưa.  * Không:   + 1. Kết nối CSDL     2. Lấy thông tin sản phẩm theo mã     3. Truyền thông tin sản phẩm vào phiên giỏ hàng theo mã sản phẩm (tên, ảnh, giá, số lượng để mặc định là 1) * Có:  1. Kiểm tra số lượng của sản phẩm này có vượt quá 10 hay chưa, nếu có vượt quá thông báo lỗi ngược lại tiếp tục. 2. Tăng số lượng lên 1 trong phiên giỏ hàng theo mã sản phẩm. 3. Tăng số tượng tất cả sản phẩm trong giỏ lên 1 4. Thông báo thêm sản phẩm thành công. |
| ***Đầu ra*** | * Đúng: Thông bbáo thành công * Sai: Điều hướng quay lại và thông báo thất bại |
| ***Lưu ý*** |  |

* **Xem, cập nhật sản phẩm trong giỏ hàng.**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng (đã đăng nhập) |
| ***Mô tả*** | Xem, thêm, xóa sản phẩm trong giỏ hàng |
| ***Kích hoạt*** | Biểu tượng xe đẩy hàng trên thanh header |
| ***Đầu vào*** | Thông tin giỏ hàng trong phiên. |
| ***Trình tự xử lý*** | 1. Nếu khách hàng chưa đăng nhập thì chuyển sang trang đăng nhập với thông báo. 2. Nếu không có sản phẩm nào trong giỏ hàng thì thông báo không có sản phẩm 3. Nếu có tạo vòng lặp hiển thị ra các thông tin từng sản phẩm 4. Nếu ấn nút thêm sản phẩm hoặc nhập số lượng khác. 5. Gọi API đến server gồm 2 thông tin id và số lượng sản phẩm để cập nhật giỏ hàng, nếu âm thì quy ước là 0, nếu thập phân thì làm tròn dưới. 6. Nếu không có sản phẩm trong giỏ hàng (tức là Call API bậy bạ) thì xóa phiên giỏ hàng. 7. Cập nhật thông tin số lượng sản phẩm trong giỏ hàng, tổng tiền của sản phẩm đó, tổng tiền trong giỏ hàng. 8. Nếu sau khi cập nhật mà số lượng sản phẩm trong giỏ hàng = 0 thì tải lại trang (vì lười code JavaScript :3) 9. Nếu ấn nút xóa sản phẩm: Tương tự cập nhật nhưng với số lượng là 0. 10. Hiển thị giỏ hàng sau khi cập nhật. |
| ***Đầu ra*** | Thông tin giỏ hàng sau khi cập nhật |
| ***Lưu ý*** | * Kiểm tra có phải kiểu số, số âm, số thập phân ở client và server. * Cập nhật thông tin sau khi thêm,sửa, xóa bằng JavaScript không tải lại trang * Số lượng/1 sản phẩm không vượt quá 10. * Format lại giá tiền sau khi cập nhật ( Viết hàm StringtoNumber JavaScript riêng ) |

* **Đặt hàng.**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng (đã đăng nhập) |
| ***Mô tả*** | Đặt hàng |
| ***Kích hoạt*** | Nút thanh toán ngay ở trang giỏ hàng |
| ***Đầu vào*** | Thông tin giỏ hàng trong phiên. |
| ***Trình tự xử lý*** | 1. Hiển thị sản phẩm trong giỏ hàng, form điền thông tin đơn hàng, lấy và hiển thị dữ liệu địa chỉ. 2. Kiểm tra độ hợp lệ của form và request (cả JavaScript và PHP), nếu không hợp lệ thì thông báo lỗi. 3. Lấy mã lớn nhất và chạy vòng lặp truyền thông tin vào bảng hóa đơn cho đến khi không bị lỗi (tránh đặt trùng thời điểm). 4. Truyền thông tin chi tiết vào mã hóa đơn sản phẩm chi tiết. 5. Truyền thông tin vào bảng lịch sử hóa đơn (để bên admin duyệt). 6. Hủy phiên giỏ hàng và phiên số lượng sản phẩm. 7. Thông báo đặt hàng thành công . |
| ***Đầu ra*** | Đúng: Thông báo thành công, đưa ra lựa chọn về trnag chủ hoặc xem thông tin hóa đơn  Sai: Thông báo lỗi. |
| ***Lưu ý*** | * Xử lí địa chỉ bằng JavaScript. |

* **Xem đơn hàng đã đặt, hủy đơn.**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng (đã đăng nhập) |
| ***Mô tả*** | Xem và hủy đơn hàng |
| ***Kích hoạt*** | Nút thông tin đơn hàng ở trang tài khoản |
| ***Đầu vào*** | Dữ liệu khách hàng( id ) trong phiên |
| ***Trình tự xử lý*** | 1. Kết nối database 2. Truy vấn các đơn hàng theo id của khách hàng (Join nhiều lần) 3. Tạo 1 mảng 3 chiều xử lý thông tin trùng lặp của hóa đơn và đưa ra dữ liệu giống nested object. 4. Hiển thị hóa đơn (được sắp xếp theo hóa đơn mới nhất xếp trước ). 5. Các đơn hàng chỉ hiện thị sản phẩm đầu tiên, người dùng ấn nút xem thêm để xem đầy đủ và có thể ẩn bớt sản phẩm.   Khi ấn hủy:   1. Thông báo xác nhận hủy hay không 2. Nếu hủy gửi dữ liệu đến API 3. Truy vấn kiểm tra hóa đơn có tồn tại và ở trạng thái chờ duyệt ứng với id khách hàng hay không, nếu không (call API bậy bạ) thì thoát. 4. Cập nhật tình trạng hóa đơn “Đã hủy” |
| ***Đầu ra*** | Hiển thị thông tin tất cả hóa đơn của người dùng, nếu ấn hủy hóa đơn nào thì cập nhật tình trạng hóa đơn này. |
| ***Lưu ý*** | * Có thể thêm tính năng lọc thành các loại hóa đơn đang chờ duyệt, đã hủy, đã duyệt để người dùng dễ theo dõi. * Cần sắp xếp giao diện đẹp mắt. * Dữ liệu trả về : |

* **Bình luận, đánh giá sản phẩm.**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng (đã đăng nhập) |
| ***Mô tả*** | Bình luận, đánh giá (rating) sản phẩm |
| ***Kích hoạt*** | Nút bình luận ở ngay dưới thông tin rating của sản phẩm, người dùng rating rồi mới được bình luận |
| ***Đầu vào*** | Chỉ số rating và comment. |
| ***Trình tự xử lý*** | * Tạo, cập nhật rating và comment mới:  1. Nếu khách hàng chưa đăng nhập thì thông báo lỗi. 2. Lấy thông tin từ form rồi call API 3. Kiểm tra tính hợp lệ của request 4. Truyền dữ liệu vào database (Insert on duplicate key update) bao gồm cập nhật và tạo cái mới 5. Cập nhật chỉ số rating của sản phẩm, số lượng đánh giá, chỉ số rating của người dùng, comment của người dùng hiện tại (bằng JavaScript). 6. Đóng form đánh giá. |
| ***Đầu ra*** | Trạng thái phần đánh giá của sản phẩm sau khi được cập nhật đánh giá. |
| ***Lưu ý*** | * Phần comment cần sắp xếp phần đánh giá người dùng hiện tại lên trước. * Có thể mở rộng : Dùng web socket để lắng nghe trực tiếp các bình luận mới. |

* **Xem thêm.**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Khách hàng(search, collection), Khách hàng đã đăng nhập (bill) |
| ***Mô tả*** | Thay vì đánh số trang như thông thường, ở đây em dùng xem thêm để load thêm dữ liệu. |
| ***Kích hoạt*** | Nút load-more ở cuối các trang: Search, Collection, Bill |
| ***Đầu vào*** | Số sản phẩm đang hiển thị tại trang (offset), số sản phẩm(đơn hàng) tối đa mỗi lần tải thêm hay số sản phẩm tối đa trên 1 trang lúc đầu (limit) |
| ***Trình tự xử lý*** | * Trang search:  1. Kiểm tra dữ liệu từ request. 2. Truy vấn đến database với offset và limit số sản phẩm 1 trang.  * Trang collection:  1. Lấy giới tính của trang, nhà sản xuất đang lọc, chế độ sắp xếp theo giá, thể loại sản phẩm đang xem call đến API. 2. Tương tự như lúc call API để gọi sản phẩm (kiểm tra dữ liệu từ request rồi truy vấn) với offset và limit. 3. Hiển thị dữ liệu ra cho người dùng, nếu sản phẩm hiển thị nhỏ hơn sản phẩm tối đa mỗi lần tải thì ẩn nút load-more (JavaScript). 4. Tạo hiệu ứng trượt.  * Trang account(bill) (tương tự như trang xem bills lúc đầu):  1. Kết nối database 2. Truy vấn các đơn hàng theo id của khách hàng (Join nhiều lần) với offset và limit. 3. Tạo 1 mảng 3 chiều xử lý thông tin trùng lặp của hóa đơn và đưa ra dữ liệu giống nested object. 4. Hiển thị hóa đơn (Sử dụng JavaScript để chèn thêm dữ liệu) 5. Cập nhật nút load-more(nếu nhỏ hơn số sản phẩm tối đa 1 lần load thì ẩn ), reset lại các events (xem chi tiết đơn hàng, hủy đơn hàng) (JavaScript). |
| ***Đầu ra*** | Tải thêm các sản phẩm(đơn hàng) nếu có. |
| ***Lưu ý*** | * Có thể mở rộng : Tải thêm ở phần bình luận và đánh giá sản phẩm |

**LƯU Ý*:***   
 Các chức năng trong phần người dùng đã đăng nhập (*Chỉnh sửa thông tin cá nhân, đổi mật khẩu,Bình luận, đánh giá sản phẩm, Xem sản phẩm trong giỏ hàng, Đặt hàng, Xem đơn hàng đã đặt, hủy đơn*) đều dùng hàm Authenticate để xác minh liệu API được gọi có phải từ người dùng đã đăng nhập.

#### Phía quản lí

* **Đăng nhập**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Admin, Super Admin |
| ***Mô tả*** | Đăng nhập |
| ***Kích hoạt*** | Trang đầu tiên (Trang chủ) |
| ***Đầu vào*** | Email đăng nhập  Mật khẩu |
| ***Trình tự xử lý*** | 1. Kiểm tra đã đăng nhập chưa hoặc có Cookie token không: - Đã đăng nhập:   Chuyển hướng tới trang hóa đơn.   * + Có token và token hợp lệ: * Tạo phiên đăng nhập, lưu mã tài khoản vào phiên. * Chuyển hướng tới trang xem hóa đơn.   + Ngược lại: chuyển sang form đăng nhập.  1. Lấy thông tin từ form, kiểm tra không được để trống, kiểm tra email, mật khẩu có hợp lệ, nếu sai thì thông báo lỗi. 2. Kiểm tra tên đăng nhập và mật khẩu có khớp với 1 tài khoản nào không:   - Đúng:   * Tạo phiên đăng nhập, lưu mã tài khoản vào phiên. * Chuyển hướng tới trang xem hóa đơn.   - Sai: chuyển hướng về trang đăng nhập và thông báo lỗi. |
| ***Đầu ra*** | * Đúng: Chuyển hướng tới trang xem hóa đơn. * Sai: Hiển thị trang đăng nhập và thông báo thất bại. |
| ***Lưu ý*** | * Kiểm tra ô nhập không được để trống bằng JavaScript. * KIểm tra email, mật khẩu cóhợp lệ không bằng JavaScript. * Thông báo lỗi được hiển thị bằng hàm JavaScript * Check vào checkbox nhớ mật khẩu để lưu tài khoản cho phiên đăng nhập sau. |

* **Lưu mật khẩu**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Admin, Super Admin |
| ***Mô tả*** | Lưu mật khẩu |
| ***Kích hoạt*** | Người dùng check vào checkbox “Nhớ mật khẩu” trong form đăng nhập và đăng nhập vào tài khoản |
| ***Đầu vào*** | Tài khoản đã hợp lệ , nếu không thì thông báo lỗi (phần đăng nhập) |
| ***Trình tự xử lý*** | 1. Kiểm tra xem có check vào checkbox hay chưa   - Có: Thực hiện tiếp.  - Không: Bỏ qua bước lưu cookie.   1. Mã hóa thông tin của người dùng 2. Lưu vào cookie token trên trình duyệt người dùng. 3. Đăng nhập vào tài khoản người dùng |
| ***Đầu ra*** | Những lần đăng nhập trên trình duyệt đó sẽ không cần phải điền form ở trang đăng nhập mà vào thẳng trang xem hóa đơn. |
| ***Lưu ý*** | Thời gian duy trì: 30 ngày kể từ ngày đăng nhập có lưu mật khẩu. |

* **Đăng xuất**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Admin, Super Admin |
| ***Mô tả*** | Đăng xuất |
| ***Kích hoạt*** | Người dùng ấn vào nút “Đăng xuất” trong menu user.  (Bên cạnh nút tìm kiếm) |
| ***Đầu vào*** | * Tài khoản đã đăng nhập |
| ***Trình tự xử lý*** | 1. Chuyển đến trang signout.php 2. Xóa các Session, Cookie liên quan đến tài khoản 3. Chuyển lại về trang chủ. |
| ***Đầu ra*** | Chuyển về trang chủ, người dùng ở trạng thái chưa có tài khoản |
| ***Lưu ý*** |  |

* **Đổi mật khẩu**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Super Admin, Admin |
| ***Mô tả*** | Đổi mật khẩu của người dùng |
| ***Kích hoạt*** | Ấn vào nút “Đổi mật khẩu ” ở menu user.  (Bên cạnh nút tìm kiếm) |
| ***Đầu vào*** | Mật khẩu cũ  Mật khẩu mới |
| ***Trình tự xử lý*** | 1. Kiểm tra có trống bất kỳ trường dữ liệu nào không, nếu có thông báo lỗi. 2. Kiểm tra hai mật khẩu mới có trùng khớp hay không , nếu không thông báo lỗi. 3. Kiểm tra mật khẩu cũ có đúng hay không, nếu không thông báo lỗi. 4. Lưu mật khẩu mới vào database đồng thời thông báo thành công |
| ***Đầu ra*** | Đúng : Mật khẩu người dùng được cập nhật  Sai: Thông báo lỗi |
| ***Lưu ý*** | 2 mật khẩu trùng khớp được bắt sự kiện từ JavaScript và PHP |

* **Xem tất cả sản phẩm**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Super Admin, Admin |
| ***Mô tả*** | Xem các sản phẩm trưng bày và chi tiết từng sản phẩm |
| ***Kích hoạt*** | Nhấn vào tab Sản phẩm |
| ***Đầu vào*** | Không có |
| ***Trình tự xử lý*** | 1. Từ thành phần “Sản phẩm” và tính năng “Danh sách”, lấy dữ liệu được inner join đầy đủ thông tin từ database trong các bảng phù hợp với thành phần và tính năng. 2. Hiển thị ra thông tin: Tên, Giá, Nhà sản xuất và Mô tả của sản phẩm theo từng cột vào khung hiển thị. Hiển thị tối đa 7 đối tượng (7 hàng và 1 hàng tiêu đề) và đánh số trang. |
| ***Đầu ra*** | Đúng : Các sản phẩm được trưng bày  Sai: Không hiển thị sản phẩm hoặc đăng xuất, quay về trang chủ nếu không đủ 1 số điều kiện. |
| ***Lưu ý*** | * Trang web gồm 4 phần chính: 1 cột bên trái để Branding (Logo) và thanh menu dọc. Phần còn lại của trang chia làm 3 phần theo thứ tự: 1 phần cho thanh tìm kiếm và menu thông tin người dùng, 1 phần cho menu ngang và 1 phần cho khung hiển thị * Chia các thành phần: Chia các đường dẫn của các thành phần (Sản phầm, đơn hàng, Nhà sản xuất, Admin) vào thanh menu ngang trên khung hiển thị. Từ các đường dẫn đó mà chúng ta kiểm tra các điều kiện cho bên Admin và Super Admin cho phù hợp, kiểm tra tính năng hiện tại đang ở đường dẫn, qua trình tự xử lý ở trên và hiển thị vào khung hiển thị. * Chia các tính năng: Chia đường dẫn các tính năng của mỗi thành phần (Danh sách, Thêm đối tượng và Xóa đối tượng) vào thanh menu dọc bên trái của khung hiển thị. |

* **Xem cụ thể từng sản phẩm (kèm theo sơ đồ)**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Super Admin, Admin |
| ***Mô tả*** | Xem thông tin chi tiết các sản phẩm và số lượng bán ra trong tuần và trong tháng |
| ***Kích hoạt*** | Nhấn 1 sản phẩm cần xem |
| ***Đầu vào*** | Không có |
| ***Trình tự xử lý*** | 1. Truyền mã sản phẩm thông qua JavaScript cho bên server, lấy dữ liệu được join đầy đủ thông tin từ database trong các bảng phù hợp với thành phần, tính năng và mã sản phẩm. 2. Hiển thị ra thông tin: Ảnh, tất cả các thông tin của sản phẩm vào 1 phần bên trái và hiển thị biểu đồ số lượng sản phẩm bán được trong tuần (7 ngày) và trong tháng (30 ngày) vào phần bên phải vào khung hiển thị. |
| ***Đầu ra*** | Đúng : Đầy đủ thông tin của sản phẩm  Sai: Không hiển thị sản phẩm hoặc đăng xuất, quay về trang chủ nếu không đủ 1 số điều kiện. |
| ***Lưu ý*** | * In đậm tên sản phẩm * Biểu đồ: Trục ngang biểu thị ngày, trục dọc biểu thị số lượng, có tiêu đề biểu đồ. * Bên phần thông tin sản phẩm có thêm nút “Sửa thông tin” để có thể nhảy sang trang sửa thông tin sản phẩm. |

* **Sửa thông tin sản phẩm**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Super Admin, Admin |
| ***Mô tả*** | Sửa thông tin của sản phẩm mình vừa chi tiết |
| ***Kích hoạt*** | Nhấn nút “Sửa thông tin” trong form |
| ***Đầu vào*** | Thông tin sản phẩm cần sửa |
| ***Trình tự xử lý*** | 1. Truyền mã sản phẩm thông qua JavaScript cho bên server, lấy dữ liệu được join đầy đủ thông tin từ database trong các bảng phù hợp với thành phần, tính năng và mã sản phẩm. 2. Hiển thị ra thông tin: Ảnh được đặt trên 1 form chứa thông tin của sản phẩm trong từng khung người dùng có thể nhập được 3. Sửa thông tin: Sau khi người dùng đã nhâp xong thông tin cần sửa thì cần kiểm tra xem thông tin nhập có hợp lệ bằng JavaScript trước khi gửi qua server. 4. Kiểm tra thông tin: Kiểm tra xem còn trống ô nào không? Kiểm tra các ô có nhập hợp lệ không?  * Nếu đúng: thì tiếp tục xử lý * Nếu sai: Hiển thị lỗi  1. Gửi thông tin qua bên server và sửa lại thông tin sản phẩm trong database. |
| ***Đầu ra*** | Đúng : Thông tin của sản phẩm đã được sửa  Sai: Không sửa thông tin sản phẩm. |
| ***Lưu ý*** |  |

* **Thêm sản phẩm**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Super Admin, Admin |
| ***Mô tả*** | Thêm sản phẩm |
| ***Kích hoạt*** | Nhấn nút “Thêm sản phẩm” |
| ***Đầu vào*** | Thông tin sản phẩm cần thêm |
| ***Trình tự xử lý*** | 1. Tạo 1 form chứa các miền cần điền (trong đó có phần lựa chọn cho “Đối tượng”, “Thể loại” và “Nhà sản xuất”) 2. Kiểm tra xem người dùng có nhập hợp lệ bằng JavaScript:   - Nếu đúng: thì tiếp tục xử lý   * Nếu sai: Hiển thị lỗi  1. Gửi thông tin qua bên server, kiểm tra thông tin có hợp lệ bên server và thêm thông tin sản phẩm trong database. |
| ***Đầu ra*** | Đúng : Sản phẩm được thêm vào database.  Sai: Sản phẩm không được thêm vào database. |
| ***Lưu ý*** |  |

**Xóa sản phẩm**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Super Admin |
| ***Mô tả*** | Thêm sản phẩm |
| ***Kích hoạt*** | Nhấn nút “Xóa” cùng dòng với sản phẩm cần xóa |
| ***Đầu vào*** | Không có |
| ***Trình tự xử lý*** | 1. Truyền mã sản phẩm qua bên server 2. Xóa sản phẩm có mã sản phẩm mới truyền qua |
| ***Đầu ra*** | Đúng : Sản phẩm được xóa.  Sai: Sản phẩm không xóa. |
| ***Lưu ý*** |  |

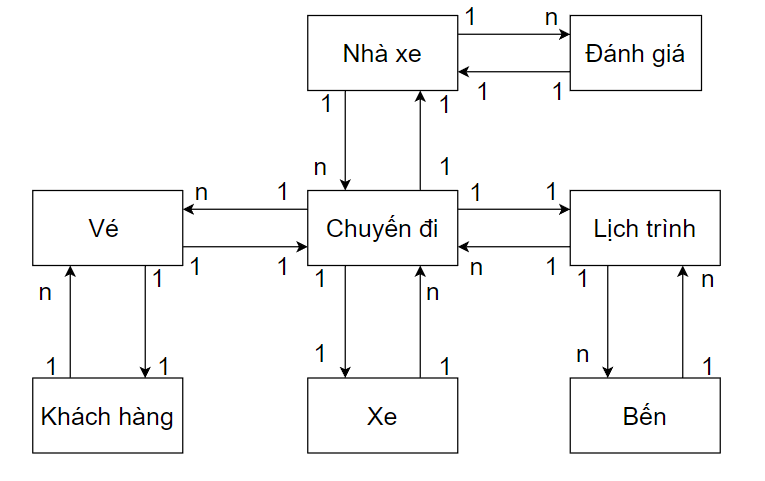
* **Tìm kiếm các đối tượng theo tên**

|  |  |
| --- | --- |
| ***Các tác nhân*** | Admin, Super Admin |
| ***Mô tả*** | Tìm kiếm sản phẩm, nhân viên, nhà sản xuất theo tên |
| ***Kích hoạt*** | Nút tìm kiếm đầu trang |
| ***Đầu vào*** | Thông tin được nhập vào ô tìm kiếm |
| ***Trình tự xử lý*** | 1. Chuyển dữ liệu từ ô tìm kiếm sang server 2. Phía server chuyển ngược lại từ tìm kiếm lên thanh địa chỉ. 3. Truy vấn với giới hạn 7 sản phẩm 4. Trả dữ liệu về client và bên khách hàng hiển thị ra bằng JavaScript |
| ***Đầu ra*** | Hiển thị đối tương ra người dùng, nếu không có đối tượng nào thì không hiển thị đối tượng |
| ***Lưu ý*** |  |

# Thiết kế hệ thống

## Sơ đồ quan hệ thực thể

* Sơ đồ quan hệ thực thể giữa các đối tượng:



**<Các bảng trong cơ sở dữ liệu>**

|  |  |
| --- | --- |
| **users** | |
| ***id*** | bigint(20) primary key auto\_increment |
| name | varchar(255) |
| avatar | varchar(255) NULL |
| address | varchar(255) NULL |
| address2 | varchar(255) NULL |
| remember\_token | varchar(300) NULL |
| email | varchar(255) |
| available\_balance | bigint (20) default 0 |
| password | varchar(255) NULL |
| role | tinyint(4) default 1 |
| phone\_number | varchar(255) NULL |
| created\_at | timestamp default :current\_timestamp |

|  |  |
| --- | --- |
| **trips** | |
| ***id*** | bigint(20) unsigned autoincrement |
| coach\_id | foreign\_key |
| service\_provider\_id | foreign\_key |
| schedule\_id | foreign\_key |
| price | bigint(20) unsigned |
| departure\_date | date |

|  |  |
| --- | --- |
| **tickets** | |
| ***trip\_id*** | foreign\_key primary\_key |
| ***seat\_position*** | tinyint(4) primary\_key |
| user\_id | foreign\_key |
| arrival\_station\_id | foreign\_key |
| departure\_station\_id | foreign\_key |
| created\_at | timestamp default :current\_timestamp |
| deleted\_at | timestamp null |

|  |  |
| --- | --- |
| **stations** | |
| ***id*** | bigint(20) primary key auto\_increment |
| name | varchar(255) |
| address2 | foreign\_key (district) |
| address | foreign\_key (province) |

|  |  |
| --- | --- |
| **service\_providers** | |
| ***id*** | bigint(20) primary key auto\_increment |
| name | varchar(255) |
| phone\_number | varchar(255) |
| address | foreign\_key(province) |
| deleted\_at | timestamp NULL |
| created\_at | timestamp default current\_timestamp |
| employer\_id | foreign\_key(user) |

|  |  |
| --- | --- |
| **schedule\_details** | |
| ***schedule\_id*** | foreign\_key primary key |
| ***station\_id*** | foreign\_key primary key |
| next\_station\_id | foreign\_key |

|  |  |
| --- | --- |
| **schedules** | |
| ***id*** | bigint(20) primary key auto\_increment |
| service\_provider\_id | foreign\_key |
| departure\_province\_code | foreign\_key |
| arrival\_province\_code | foreign\_key |
| departure\_time | time |
| duration | smallint(5) unsigned |

|  |  |
| --- | --- |
| **ratings** | |
| ***trip\_id*** | foreign\_key primary\_key |
| ***user\_id*** | foreign\_key primary\_key |
| rate | tinyint(4) |
| comment | varchar(400) |
| created\_at | timestamp default:current\_timestamp |
| service\_privder\_id | foreign\_key |

|  |  |
| --- | --- |
| **provinces** | |
| ***code*** | varchar(255) primary\_key |
| name | varchar(255) |

|  |  |
| --- | --- |
| **password\_resets** | |
| **email** | foreign\_key (user) |
| **token** | varchar(255) |
| created\_at | timestamp default: current\_timestamp |

|  |  |
| --- | --- |
| **languages** | |
| ***id*** | bigint(20) primary key auto\_increment |
| created\_at | timestamp NULL |
| deleted\_at | timestamp NULL |

|  |  |
| --- | --- |
| **employees\_list** | |
| ***id*** | foreign\_key(user) primary\_key |
| service\_provider\_id | foreign\_key |
| created\_at | timestamp current\_timestamp |

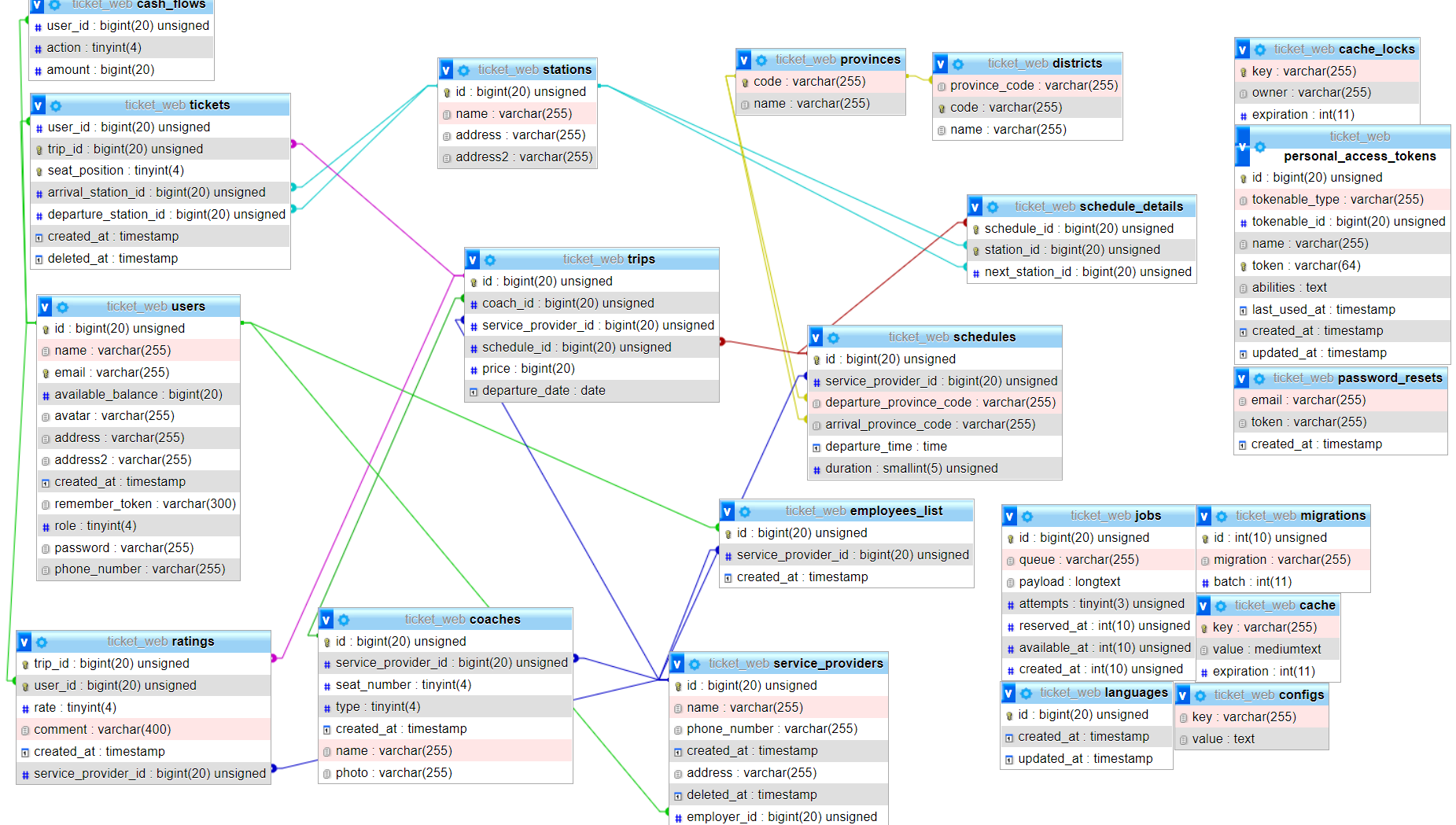
|  |  |
| --- | --- |
| **districts** | |
| ***code*** | varchar(255) primary key |
| province\_code | foreign\_key |
| name | varchar(255) |

|  |  |
| --- | --- |
| **configs** | |
| **key** | varchar(255) |
| value | text |

|  |  |
| --- | --- |
| **coaches** | |
| **id** | bigint(20) unsigned primary\_key increment |
| service\_provider\_id | foreign\_key |
| seat\_number | tinyint(4) |
| type | tinyint(4) |
| created\_at | timestamp current\_timestamp |
| name | varchar(255) |
| photo | varchar(255) |

|  |  |
| --- | --- |
| **cash\_flows** | |
| **user\_id** | foreign\_key |
| action | tinyint(4) |
| amount | bigint(20) |

* Sơ đồ cơ sở dữ liệu giữa các bảng:



## Sơ đồ trang web

* Bên khách hàng:

Diagram

Description automatically generated

* Bên quản lí:

Diagram

Description automatically generated

# Kết luận

Qua lần thực hành đồ án 1 nói riêng và lớp học này của thầy Long nói chung, chúng em có thể khẳng định được rằng chúng em:

* Đã học tập và áp dụng các kỹ năng chuyên sâu về Laravel, MySQL, HTML,CSS và JavaScript.
* Nắm được cơ bản cách thức hoạt động của một trang web bán hàng online, xây dựng được 1 website có các chức năng cần thiết.

Bên cạnh đó, nhóm chúng em vẫn còn những khuyết điểm của những con người mới bước chân vào lĩnh vực này và cũng như những khuyết điểm trong lúc làm việc nhóm như:

* Giao diện chưa được tối ưu bên quản lí.
* Quy mô website còn tương đối nhỏ, khó có thể áp dụng được cho nhiều mô hình, đặc biệt là những mô hình có cơ sở dữ liệu lớn.
* Một vài tính năng phụ chưa hoàn thành
* Giao diện còn tương đối sơ sài, chưa áp dụng được cho nhiều thiết bị

Ngoài ra, chúng em còn mong muốn được làm thêm những tính năng mở rộng khác liên quan đến trang web mà do thời gian có hạn nên chưa thực hiện được:

* Liên kết địa chỉ trên google map.
* Thêm chat box.
* Thêm trang giới thiệu.
* Giao diện bắt mắt hơn và nhiều hiệu ứng.
* Gửi tin nhắn xác minh qua điện thoại và email

# Phân công công việc