**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**BÁO CÁO CÔNG NGHỆ PHẦN MỀM**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG WEBSITE CHO THUÊ PHÒNG TRỌ**

**THEO MÔ HÌNH MVC**

****

Giảng viên hướng dẫn: **ThS.** **Nguyễn Thị Thảo Nguyên**

Sinh viên thực hiện: **Cao Xuân Mỹ**

Lớp**: 03-CNTT02**

Khoá**: 2014-2018**

***TP. Hồ Chí Minh, tháng 6 năm 2018***

**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG WEBSITE CHO THUÊ PHÒNG TRỌ**

**THEO MÔ HÌNH MVC**

****

Giảng viên hướng dẫn: **ThS.** **Nguyễn Thị Thảo Nguyên**

Sinh viên thực hiện:  **Cao Xuân Mỹ**

Lớp**: 03-CNTT02**

Khoá**: 2014-2018**

***TP. Hồ Chí Minh, tháng 6 năm 2018***

**MỞ ĐẦU**

Như chúng ta đã biết ngày nay là thời kì của sự phát triển ngành công nghệ thông tin đặc biệt là Internet. Đi theo đó là những nhu cầu tìm kiếm, giải trí, đặc biệt là mở mang kiến thức qua internet rất phổ biến. Với một chiếc máy tính, điện thoại có kết nối với Internet thì chỉ cần ở nhà duyệt Web thì gần như có thể biết được mọi diễn biến của cả nước cũng như thế giới như thế nào. Hiện này, có trên hàng triệu trang web phục vụ cho mục đích sống của chúng ta từ ẩm thực, giải trí, mua sắm, kinh doanh cho đến học online…vv. Từ đó, cho thấy rằng nhu cầu sử dụng Web càng ngày càng tăng cao để đáp ứng nhu cầu xã hội như ngày nay.

Bài báo cáo trình bày về quá trình thực tập tại công ty ARIS Việt Nam và những kiến thức đã học được cũng như là công việc đã làm tại công ty.

**LỜI CẢM ƠN**

Để hoàn thành tốt quá trình thưc tập tại công ty em xin tỏ lòng biết ơn đến anh Vũ Hoàng Quân (hướng dẫn thực tập chuyên ngành PHP) và anh Nguyễn Văn Khá (hướng dẫn thực tập chính của công ty) cũng như toàn bộ thành viên công ty ARIS Việt Nam và cô Nguyễn Thị Thảo Nguyên đã tạo điều kiện và tận tình hướng dẫn em trong suốt quá trình thực tập.

Với những gì mà em học được trong suốt thời gian thực tập tại công ty không chỉ là nền tảng để em hoàn thành tốt việc thực tập tại công ty, mà còn là hành trang quý báu để em bước vào đời một cách vững chắc và tự tin.

Tuy vậy, do thời gian có hạn, cũng như kiến thức, kinh nghiệm còn hạn chế nên khó tránh khỏi những thiếu sót, hạn chế nhất định. Vì vậy em mong nhận được sự chỉ bảo, đóng góp của quý thầy cô để chúng em có điều kiện bổ sung nâng cao kiến thức, phục vụ tốt hơn trong công việc sau này.

Cuối cùng chúng em xin gửi đến toàn thể công ty ARIS Việt Nam và quý thầy cô giáo lời chúc sức khỏe, và thành công.

Em xin chân thành cảm ơn!

**NHẬN XÉT CÁN BỘ HƯỚNG DẤN TẠI ĐƠN VỊ THỰC TẬP**

………….., ngày….tháng….năm……

CÁN BỘ HƯỚNG DẪN

*(ký tên)*

XÁC NHẬN CỦA CƠ QUAN

THỦ TRƯỞNG

*(ký tên, đóng dấu)*

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

………….., ngày….tháng….năm……

CÁN BỘ HƯỚNG DẪN

*(ký tên)*

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

………..,ngày….tháng….năm……

NGƯỜI NHẬN XÉT

*(ký tên)*

**Mục lục**

[**CHƯƠNG 1: TỔNG QUAN** 1](#_Toc516128842)

[**1.1** **Giới thiệu về công ty ARIS Việt Nam.** 1](#_Toc516128843)

[1.1.1 Tổng quan về công ty. 1](#_Toc516128844)

[1.1.2 Dịch vụ của công ty. 2](#_Toc516128845)

[1.1.3 Công việc được giao thực tập tại công ty 6](#_Toc516128846)

[1.1.4 Phạm vi đề tài: 10](#_Toc516128847)

[**CHƯƠNG 2: CỞ SỞ LÝ LUẬN** 12](#_Toc516128848)

[**2.1** **Lý thuyết** 12](#_Toc516128849)

[2.1.1 Giới thiệu về Agile 12](#_Toc516128850)

[2.1.2 Giới thiệu về Scrum 13](#_Toc516128851)

[**2.2** **Kỹ thuật** 15](#_Toc516128852)

[2.2.1 Giới thiệu về PHP 15](#_Toc516128853)

[2.2.2 Lịch sử phát triển của PHP. 15](#_Toc516128854)

[2.2.3 Giới thiệu về Git và Github. 16](#_Toc516128855)

[2.2.4 Cách tạo một repository. 17](#_Toc516128856)

[2.2.5 Giới thiệu về Source Tree 18](#_Toc516128857)

[2.2.6 Kết nối Github với Source Tree 18](#_Toc516128858)

[2.2.7 Giới thiệu về Laravel. 21](#_Toc516128859)

[2.2.8 Cài đặt Laravel. 22](#_Toc516128860)

[2.2.9 Mô hình MVC trong Laravel 23](#_Toc516128861)

[**CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM** 27](#_Toc516128862)

[**3.1** **Jquery Validation.** 27](#_Toc516128863)

[**3.2** **Datatable JQuery.** 31](#_Toc516128864)

[**3.3** **Laravel excel.** 33](#_Toc516128865)

[**CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 40](#_Toc516128866)

[**4.1** **Đánh giá kết quả đạt được.** 40](#_Toc516128867)

[**4.2** **Kiến thức lý thuyết**. 40](#_Toc516128868)

[**4.3** **Kỹ năng thực hành.** 40](#_Toc516128869)

[**4.4** **Hạn chế** 41](#_Toc516128870)

**MỤC LỤC HÌNH ẢNH**

[**Hình 2.1:** Mô hình hoạt động của DVCS 16](#_Toc516055329)

[**Hình 2.2:** Tạo 1 repository trên web 17](#_Toc516055330)

[**Hình 2.3:** Đặt tên cho repository 17](#_Toc516055331)

[**Hình 2.4:** Đường link repository 18](#_Toc516055332)

[**Hình 2.6:** Clone github từ source Tree 19](#_Toc516055333)

[**Hình 2.7:** Commit file 19](#_Toc516055334)

[**Hình 2.8:** Push file lên repository 20](#_Toc516055335)

[**Hình 2.9:** Sau khi push thì repository sẽ tồn tại file đó 20](#_Toc516055336)

[**Hình 2.10:** Fetch để hiện số file code thay đổi 21](#_Toc516055337)

[**Hình 2.11:** Pull từ repository về máy local 21](#_Toc516055338)

[**Hình 2.12:** Cài đặt laravel trên cmd 22](#_Toc516055339)

[**Hình 2.13:** Laravel đã cài thành công 22](#_Toc516055340)

[**Hình 2.14:** Kết nối Laravel với cơ sở dữ liệu MySQL 23](#_Toc516055341)

[**Hình 2.15:** Sử dụng schema để tạo bảng 23](#_Toc516055342)

[**Hình 2.16:** Bảng users được tạo 24](#_Toc516055343)

[**Hình 2.18:** Route trong laravel 25](#_Toc516055344)

[**Hình 2.19:** Một file controller xử lý hiện thị dữ liệu từ model ra view 26](#_Toc516055345)

[**Hình 3.1:** Thêm thư viện Jquery validation vào website 27](#_Toc516055346)

[**Hình 3.2:** Tạo form để ràng buộc validate 27](#_Toc516055347)

[**Hình 3.3:** Rules cho các name của form 28](#_Toc516055348)

[**Hình 3.4:** Message cho các rules từng name trong form 29](#_Toc516055349)

[**Hình 3.5:** Hiển thị message trên form 30](#_Toc516055350)

[**Hình 3.6:** Nhúng file dataTables.js vào website 31](#_Toc516055351)

[**Hình 3.7:** Chạy lệnh DataTable trong thẻ JS 31](#_Toc516055352)

[**Hình 3.8:** Giao diện tổng quan về Datatalbe 31](#_Toc516055353)

[**Hình 3.9:** Chức năng tìm kiếm và số tin hiển thị 32](#_Toc516055354)

[**Hình 3.10:** Cấu hình trong file config/app.php 33](#_Toc516055355)

[**Hình 3.11:** Code controller để xuất dữ liệu ra excel 33](#_Toc516055356)

[**Hình 3.12:** Nội dung từ database cần được xuất ra excel 34](#_Toc516055357)

[**Hình 3.13:** Nội dung đã được xuất ra file excel thành công 34](#_Toc516055358)

[**Hình 3.14:** Giao diện trang chủ của FindHomeD 35](#_Toc516055359)

[**Hình 3.15:** Trang thông tin chi tiết của bài viết 36](#_Toc516055360)

[**Hình 3.16:** Giao diện đăng tin 37](#_Toc516055361)

[**Hình 3.17:** Chức năng đăng nhập 38](#_Toc516055362)

[**Hình 3.18:** Quản lý bài đăng 38](#_Toc516055363)

[**Hình 3.19:** Chức năng lọc tin 39](#_Toc516055364)

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

**CHƯƠNG 2: MÔ TẢ NGHIỆP VỤ, YÊU CẦU**

* 1. **Bài toán nghiệp vụ.**
     1. **Bài toán đặt ra.**

Một công ty muốn xây dựng một trang web để quản lý các nhà trọ đang tìm người thuê và cho phép người cần tìm nhà trọ tìm kiếm phòng trọ hoặc liên quan đến nơi ở.

Đây là một hệ thống chứa một lượng lớn thông tin về các nhà, phòng, chung cư… hiện tại chủ nhà không có nhu cầu sử dụng và muốn cho thuê. Người truy cập vào trang web có thể sử dụng chức năng tìm kiếm theo khu vực ở thành phố, quận huyện… Hệ thống sẽ cung cấp cho người truy cập thông tin chi tiết để có thể liên hệ và thuê theo ý muốn.

* + 1. **Đối tượng và phạm vi.**

Tất cả mọi người trong lãnh thổ Việt Nam có nhu cầu thuê, đặc biệt là sinh viên đại học và phạm vi trên toàn quốc đặc biệt là các tỉnh có khu dân cư cao như Hồ Chí Minh, Hà Nội, Đà Nẵng…

* + 1. **Mô tả bài toán.**

Xây dựng hệ thống cho phép tương tác giữa người tìm và chủ cần cho thuê.

* Đối với chủ cho thuê:

Hệ thống sẽ cho phép chủ thuê đăng thông tin lên với nội dung cần có là: địa chỉ, đối tượng, giá tiền, diện tích, loại mục cho thuê, mô tả, số điện thoại liên hệ…

* Đối với người thuê:

Hệ thống cho phép tìm kiếm các mục thuê theo địa chỉ, giá tiền, diện tích và loại mục cho thuê.

* 1. **Giải quyết yêu cầu.**
     1. **Sơ lược phân tích hệ thống.**

Dựa vào yêu cầu hệ thống và bài toán nghiệp vụ được đặt ra thì sơ lược hệ thống sẽ cần có:

* Phần dữ liệu bao gồm: thông tin nhà cho thuê, các danh mục cho thuê, thông tin về địa điểm, diện tích, quyền truy cập….
* Phần xử lý: có thể thêm xóa sửa trong quản lý danh mục, thêm xóa sửa thông tin của khách hàng, có thể xóa trong quản lý bài đăng, có thể in, xuất excel các nhà cho thuê.
  + 1. **Biểu đồ phân rã chức năng.**

# **CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM**

* 1. **Phân tích hệ thống.**
     1. **Xác định thực thể và các thuộc tính.**

Sau khi nghiên cứu bài toán và phân tích hệ thống sẽ có những thực thể và các thuộc tính đi kèm như sau:

**Nhà:** Danh mục loại cho thuê, giá cho thuê, địa chỉ, diện tích, hình ảnh, vị trí trên google map, số điện thoại, đối tượng cho thuê, ngày đăng lên.

**Loại hình cho thuê:** Bao gồm danh mục cho thuê như phòng trọ, mặt bằng, chung cư…

**Tỉnh thành:** Chứa các tỉnh của nước Việt Nam.

**Thành phố, huyện:** Bao gồm các thành phố, huyện nơi các tỉnh của nước Việt Nam

**Người đăng:** Chứa tài khoản người dùng bao gồm tên tài khoản, mật khẩu và số điện thoại.

**Vai trò:** Là nơi chứa vai trò của tài khoản sở hữu quyền truy cập web.

* + 1. **Mô hình hóa dữ liệu.**
  1. **Thiết kế hệ thống.**
     1. **Thiết kế cơ sở dữ liệu.**

Ta chọn hệ quản trị cơ sở MySQL để tổ chức lưu trữ thông tin, ta thiết kế được các bảng tương ứng với các thực thể như sau:

1. **Table Home.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Định dạng** | **Ràng buộc** |
| Home\_id | Int | 10 | Số | Tự tăng, khóa chính |
| Type\_id | Int | 10 | Số | Khóa ngoại của table hometype |
| User\_id | Int | 10 | Số | Khóa ngoại của table Users |
| Title | Varchar | 191 | Kí tự |  |
| Desc | Varchar | 10000 | Kí tự |  |
| Price | Varchar | 191 | Kí tự |  |
| Area | Varchar | 191 | Kí tự |  |
| Street | Varchar | 191 | Kí tự |  |
| District | Int | 5 | Số | Khóa ngoại của table District |
| City | Int | 5 | Số | Khóa ngoại của table Province |
| Image | Varchar | 1000 | Kí tự |  |
| View | Int | 11 | Số |  |
| Hienthi | Int | 2 | Số |  |
| Doituong | Int | 2 | Số |  |
| Phone\_home | Varchar | 191 | Kí tự |  |
| Location\_map | Varchar | 100 | Kí tự |  |
| Created\_at | Timestamp |  | Thời gian |  |

1. **Table Hometype.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Định dạng** | **Ràng buộc** |
| id | Int | 10 | Số | Khóa chính |
| Nametype | Varchar | 191 | Kí tự |  |
| Nametypelink | Varchar | 191 | Kí tự |  |
| Status | Int | 2 | Số |  |
| Created\_at | Timestamp |  | Thời gian |  |

1. **Table Province.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Định dạng** | **Ràng buộc** |
| Provinceid | Int | 5 | Số | Khóa chính |
| Name | Varchar | 100 | Kí tự |  |
| Type | Varchar | 30 | Kí tự |  |

1. **Table District.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Định dạng** | **Ràng buộc** |
| Districtid | Int | 5 | Số | Khóa chính |
| Name | Varchar | 100 | Kí tự |  |
| Type | Var char | 30 | Kí tự |  |
| Provinceid | Int | 5 | Số | Khóa ngoại của table province |

1. **Table Roles.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Định dạng** | **Ràng buộc** |
| Id | Int | 10 | Số | Tự tăng, khóa chinh |
| Name | Varchar | 191 | Kí tự |  |
| Display\_name | Varchar | 191 | Kí tự |  |
| Description | Varchar | 191 | Kí tự |  |
| Created\_at | Timestamp |  | Thời gian |  |

1. **Table Role\_User.**

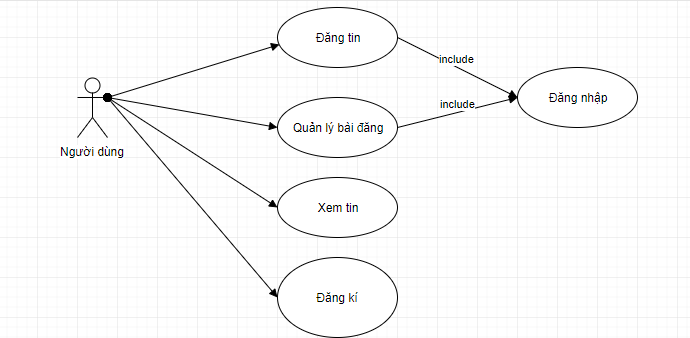
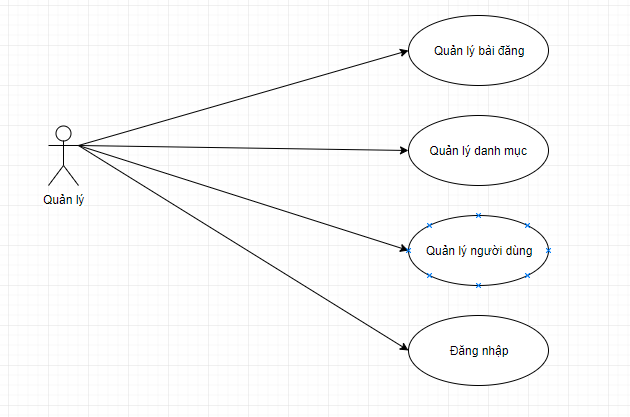
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Định dạng** | **Ràng buộc** |
| User\_id | Int | 10 | Số | khóa chính, khóa ngoại của table users |
| Role\_id | Int | 10 | Số | Khóa chính, khóa ngoại của table Roles |

1. **Table Users.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Định dạng** | **Ràng buộc** |
| id | Int | 10 | Số | Tự tăng, khóa chính |
| Username | Varchar | 191 | Kí tự |  |
| Password | Varchar | 191 | Kí tự |  |
| Status | Int | 11 | Số |  |
| Phone | Varchar | 191 | Kí tự |  |
| Created\_at | Timestamp |  | Thời gian |  |

* + 1. **Lược đồ Usecase.**

1. **Usecase tổng:**



1. **Đặc tả Usecase Đăng nhập:**

* Tóm tắt: Use case này mô tả cách đăng nhập vào hệ thống
* Dòng sự kiện:

Dòng sự kiện chính: use case này bắt đầu khi một người dùng muốn đăng nhập vào hệ thống.

* Người dùng nhập tên và mật khẩu.

Hệ thống sẽ kiểm tra thong tin của người dùng nhập vào và cho phép người dùng đăng nhập vào hệ thống.

* Dòng sự kiện khác:

Nếu trong dòng sự kiện chính các người dùng nhập thông tin sai thì hệ thống sẽ báo lỗi. Người dùng có thể quay về và nhập thông tin lại hoặc hủy bỏ việc đăng nhập 🡪 use case kết thúc.

1. **Đặc tả Usecase quản lý tài người dùng:**

* Tóm tắt: Use case này mô tả cách quản lý các tài khoản
* Dòng sự kiện:

Dòng sự kiện chính: use case này bắt đầu khi một người dùng có tài khoản là quyển Admin hoặc Owner muốn quản lý tài khoản.

Hệ thống cho phép người dùng có thể xem, thêm, xóa tài khoản.

* Dòng sự kiện khác:

Nếu là Owner đăng nhập thì sẽ không thấy chức năng xóa những tài khoản admin và tài khoản chính minh, chỉ có tài khoản admin (chủ) mới được quyền tạo tài khoản cho Owner.

1. **Đặc tả Usecase Xem tin:**

* Tóm tắt: Use case này mô tả cách người dùng có thể xem tin tức bài đăng
* Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn xem chi tiết tin của bài đăng khi click. Hệ thống sẽ cho phép người dùng xem chi tiết tin.
* Dòng sự kiện khác: Người dùng có thể thực hiện chức năng lưu tin theo phiên đang được thực hiện, nghĩa là nếu tắt trình duyệt tin lưu sẽ bị mất đi.

1. **Đặc tả Usecase Quản lý bài đăng:**

* Tóm tắt: Use case này mô tả cách người dùng quản lý bài đăng của mình nếu là User, hoặc quản lý bài đăng của tất cả nếu là Owner, Admin.
* Dòng sự kiện chinh:

+ Đối với User: Use case bắt đầu khi người dùng đăng nhập với quyền là User vào hệ thống để quản lý bài đăng của mình. Hệ thống sẽ cho phép người dùng có thể xem và xóa bài đăng.

+ Đối với Admin và Owner: Use case bắt đầu khi người dùng đăng nhập với quyền Admin hoặc Owner vào quản lý bài đăng. Hệ thống sẽ cho phép người dùng có thể cho phép duyệt bài và hiện thị cũng như xóa bài đăng.

1. **Đặc tả Usecase đăng kí:**

* Tóm tắt: Use case này mổ tả cách người dùng có thể đăng kí tài khoản để đăng tin.
* Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn đăng kí tài khoản. Hệ thống sẽ cho phép người dùng đắng kí theo đúng form được đặt ra.
* Dòng sự kiện phụ:

**+** Nếu người dùng nhập sai thông tin thì hệ thống sẽ thông báo lỗi và từ chối đăng kí của người dùng.

**+** Nếu người dùng đăng kí thành công thì hệ thống sẽ thông báo và đăng nhập trực tiếp từ tài khoản đã đăng kí.

1. **Đặc tả Usecase Đăng tin:**

* Tóm tắt: Use case này mô tả người dùng có thể đăng tin.
* Dòng sự kiện chính: Use case này bắt đầu khi người dùng đăng nhập và đăng tin. Hệ thống cho phép người dùng đăng tin nếu tài khoản có hiệu lực và đúng form của hệ thống.
* Dòng sự kiện phụ: Hệ thống sau khi nhập đăng tin sẽ thông báo và đợi người xét duyệt bài đăng thì mới hiện thị trên trang.

1. **Đặc tả Usecase Quản lý danh mục:**

* Tóm tắt: Use case này mô tả cách người dùng có thể quản lý danh mục đăng.
* Dòng sự kiện chính: Use case này bắt đầu khi người dùng có tài khoản là quyền Admin hoặc Owner vào quản lý danh mục. Hệ thống cho phép người dùng có thể thêm, xóa, sửa danh mục.
* Dòng sự kiện phụ: Hệ thống sẽ báo lỗi nếu như người dùng xóa đi danh mục đang được sử dụng và từ chối xóa danh mục đó.
  + 1. **Lược đồ Activity.**

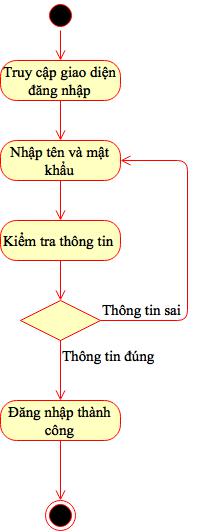
1. **Lược đồ Activity cho Usecase Đăng nhập:**

**Bước 1:**

* Truy cập giao diện đăng nhập.
* Nhập thông tin tài khoản vào.
* Hệ thống sẽ kiểm tra tài khoản và mật khẩu:
  + Nếu thông tin đúng thì đăng nhập thành công vào hệ thống.
  + Nếu thông tin vào không hợp lệ thì quay lại giao diện đăng nhập và thông báo lỗi cho người dùng.

**Bước 2:**

* Truy cập giao diện hệ thống (b).
* Nhập tài khoản (b).
* Hệ thống kiểm tra (b).
* Đăng nhập thành công (b).

**Bước 3: Vẽ lược đồ Acitvity**

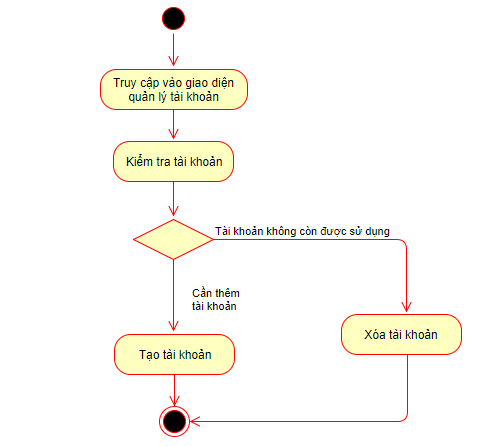
1. **Lược đồ Activity cho Usecase Quản lý tài khoản người dùng:**

**Bước 1:**

* Truy cập giao diện quản lý tài khoản.
* Hành động quản lý:
  + Thêm tài khoản.
  + Sửa tài khoản.
  + Xóa tài khoản.

**Bước 2:**

* Xóa tài khoản (a).
* Thêm tài khoản (a).
* Sửa tài khoản (a).

**Bước 3: Vẽ lược đồ Acitvity**

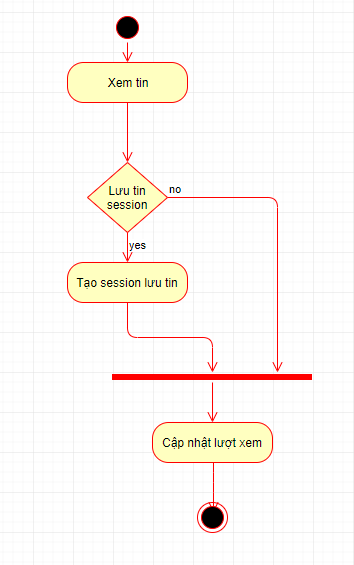
1. **Lược đồ Activity cho Usecase Xem tin:**

**Bước 1:**

* Truy cập vào bài tin cần xem.
* Lưu tin.
* Xem tin.
* Cập nhật lượt xem.

**Bước 2:**

* Lưu tin (b).
* Xem tin (b).
* Cập nhật lượt xem (a).

**Bước 3: Vẽ lược đồ Activity**

1. **Lược đồ Activity cho Usecase Quản lý bài đăng:**

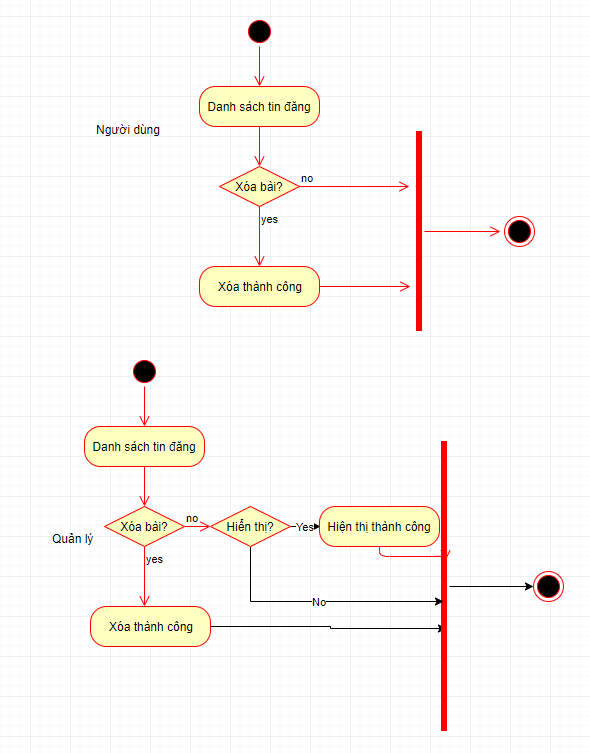
**Bước 1:**

* Truy cập giao diện quản lý bài đăng.
* Hành động:
  + Xem bài đăng.
  + Xóa bài đăng.
  + Duyệt hiển thi bài đăng.

**Bước 2:**

* Xem bài đăng (b).
* Duyệt hiển thị bài đăng (a).
* Xóa bài đăng (a).

**Bước 3: Vẽ lược đồ Activity**

****

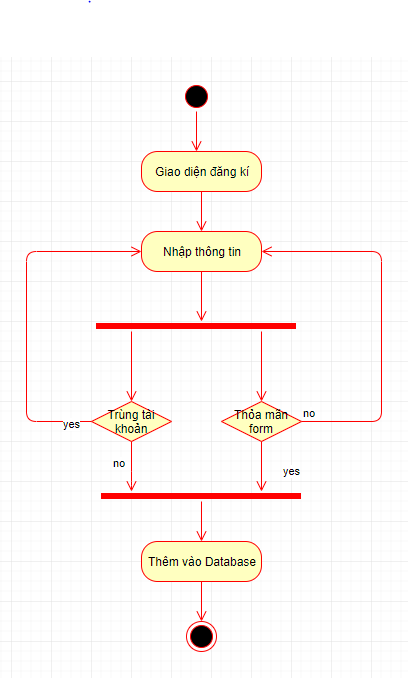
1. **Lược đồ Activity cho Usecase Đăng kí:**

**Bước 1:**

* Truy cập giao diện đăng kí.
* Nhập thông tin vào form.
* Kiểm tra trung tên hay không nếu trùng thì thông báo ra màn hình.
* Xác thực đăng kí thành công.

**Bước 2:**

* Nhập thông tin đăng kí (b).
* Xác thức trùng tài khoản (b).
* Thêm tài khoản người dùng (a).

**Bước 3: Vẽ lược đồ Activity**

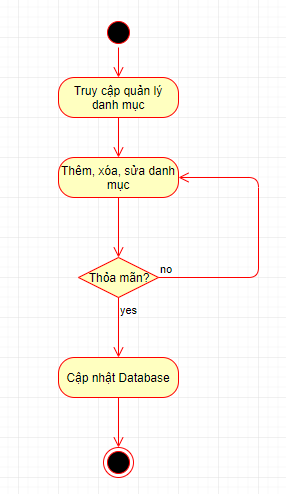
1. **Lược đồ Activity cho Usecase Quản lý danh mục:**

**Bước 1:**

* Truy cập giao diện quản lý danh mục.
* Hành động quản lý:
  + Thêm danh mục.
  + Xóa danh mục.
  + Sửa danh mục.

**Bước 2:**

* Thêm danh mục (a).
* Xóa danh mục (a).
* Sửa danh mục (a).

**Bước 3: Vẽ lược đồ Activity**

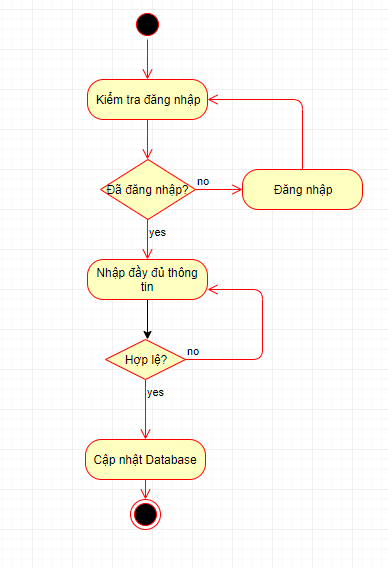
1. **Lược đồ Activity cho Usecase Đăng tin:**

**Bước 1:**

* Truy cập giao diện đăng tin.
* Kiểm tra đăng nhập hay chưa nếu chưa thì bắt phải đăng nhập
* Kiểm tra trường form có hợp lệ.

**Bước 2:**

* Kiểm tra đăng nhập (b).
* Đăng tin thành công (a).

**Bước 3: Vẽ lược đồ Activity**

* + 1. **Lược đồ Sequence**.
  1. **Lập trình và kiểm thử.**
     1. **Lập trình.**

1. **Kỹ thuật**

**Giới thiệu về PHP**

* **PHP** viết tắt là Hypertext Preprocessor là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó thích hợp với web và có thể dễ dàng nhúng vào HTML. Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

**Lịch sử phát triển của PHP.**

1. **PHP/FI.**

**-** PHP/FI viết tắt của Personal Home Page/Forms Interperter do Rasmus Lerdorf tạo ra năm 1994. Nó có kiểu biến như Perl, thông dịch tự động các biến của form và cứ pháp HTML nhúng.

1. **PHP 3.**

* PHP 3.0 la phiên bản đầu tiên cho chúng ta thấy một hình ảnh gần gủi với các phiên bản mà chúng ta biết hiện nay. Nó được tạo ra năm 1997 do Andi Gutmans và Zeev Suraski. PHP 3.0 có tính năng mở rộng mạnh mẻ, cung cấp một hạ tầng chặt chẽ cho nhiều cơ sở dữ liệu. Hoàn toàn có thể kết luận rằng đây là điểm mấu chốt dẫn đến sự thành công vang dội của PHP 3.0.

1. **PHP** **4.**

* Sau khi PHP 3 chính thức công bố thì Andi và Zeev Suraski bắt tay vào viết phần lõi của PHP. Mục đích nhằm cải tiến tốc độ các ứng dụng phức tạp và cải tiến mô đun cơ sở mã nguồn PHP. Thế là PHP 4 ra đời vào mùa đông năm 1998.

1. **PHP 5.**

* Sự thành công to lớn của PHP 4.0 đã không làm cho nhóm phát triển tự mãn. Khi nhận ra sự yếu kém còn sót lại như khả năng hỗ trợ lập trình hướng đối tượng (OOP) kém cộng với không hỗ trợ giao thức máy khách mới của MySQL thì năm 2005 PHP 5 đã được công bố đánh dấu sự chính muồi của PHP với sự có mặt của PDO.

1. **PHP 6.**

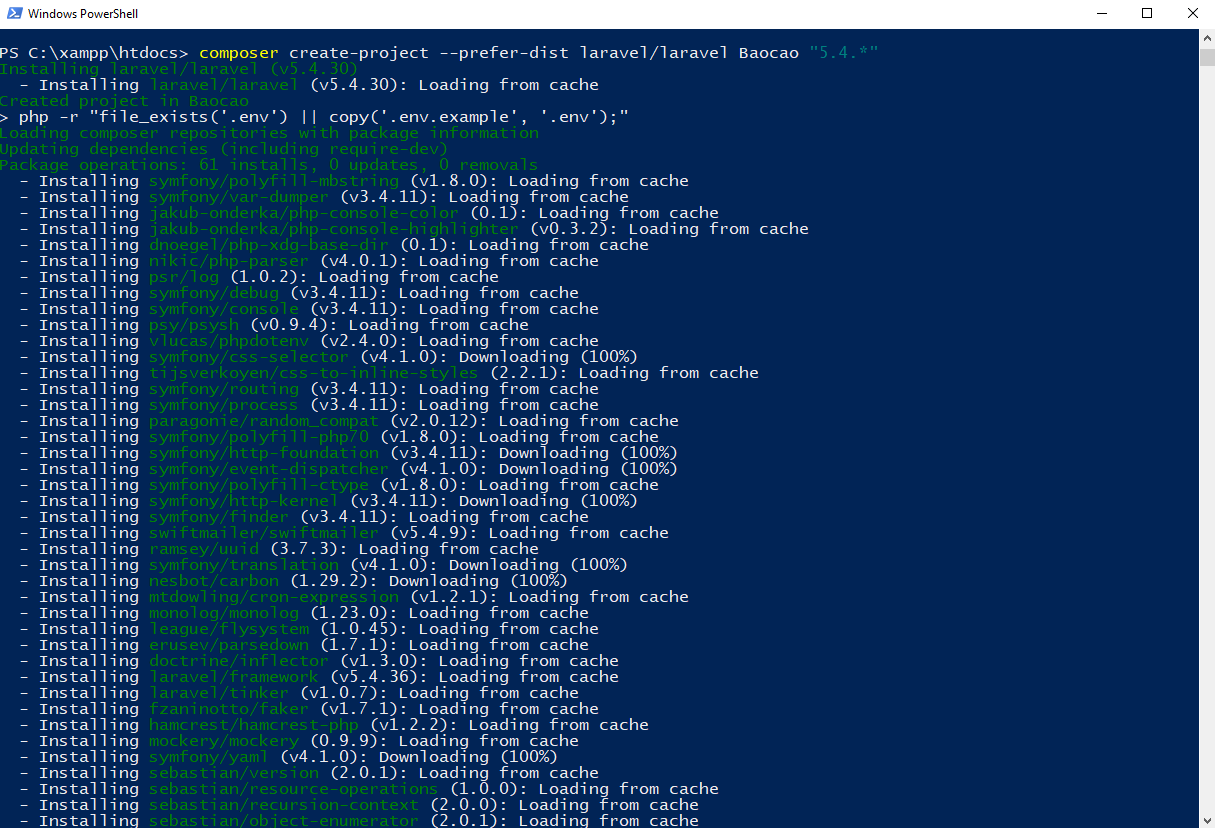
* Hiện nay phiên bản tiếp theo của PHP đang được phát triển, PHP 6 kì vọng sẽ lấp đầy những khuyết điểm ở các phiên bản trước.

**Giới thiệu về Laravel.**

* Laravel là một PHP Framework mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell nhằm mục tiêu hỗ trợ phát triển các ứng dựng Web theo kiến trúc MVC (Model-View-Controller).
* Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu – rõ ràng, một hệ thống đóng gói module và quản lý gọi phụ thuộc, nhiều cách khách để truy cập vào cơ sở dữ liệu quan hệ, nhiều tiện ích (package) khác nhau hỗ trợ triển khai và bảo trì ứng dụng.

**Cài đặt Laravel.**

* Để cài được laravel chúng ta cần có composer và xampp. Chúng ta có thể reseach trên web cách cài composer và xampp.
* Mở cửa sổ command line và di chuyển tới mục htdocs rồi gõ dòng lệnh

composer create-project --prefer-dist laravel/larvel Baocao

**Hình 2.12:** Cài đặt laravel trên cmd

* Nó sẽ tạo cho chúng ta một project Laravel mang tên Baocao ở bên trong thư mục htdocs
* Để kiểm tra laravel đã được cài đặt thành công hay chưa chỉ cần mở xampp và chạy đường link file public.

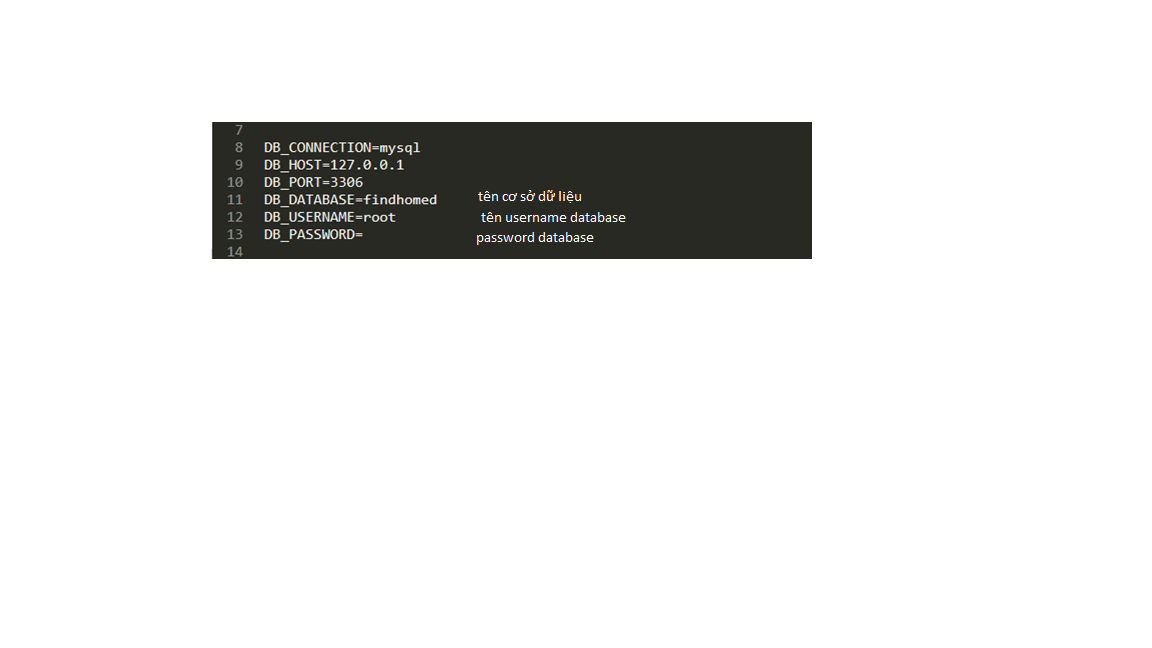


**Hình 2.13:** Laravel đã cài thành công

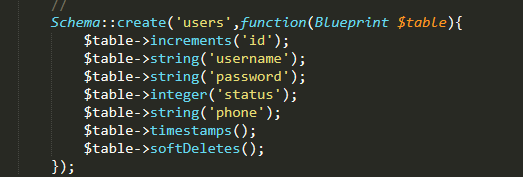
**Mô hình MVC trong Laravel**

* + 1. **Model:**
* Trong mô hình MVC thì Model đóng vai trò giao tiếp với database. Trong laravel để tạo model chúng ta dùng câu lệnh *php artisan make:model {tên model}* để tạo model. Và trực quan hơn để tạo hay quản lý table trong database thì laravel hỗ trợ có **migration.**

**\* Migration**

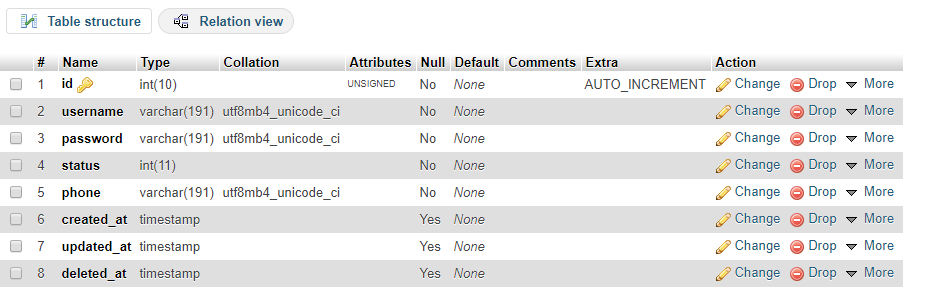
* Muốn sử dụng migration thì chúng ta phải kết nối database qua file .env trong laravel.

**Hình 2.14:** Kết nối Laravel với cơ sở dữ liệu MySQL

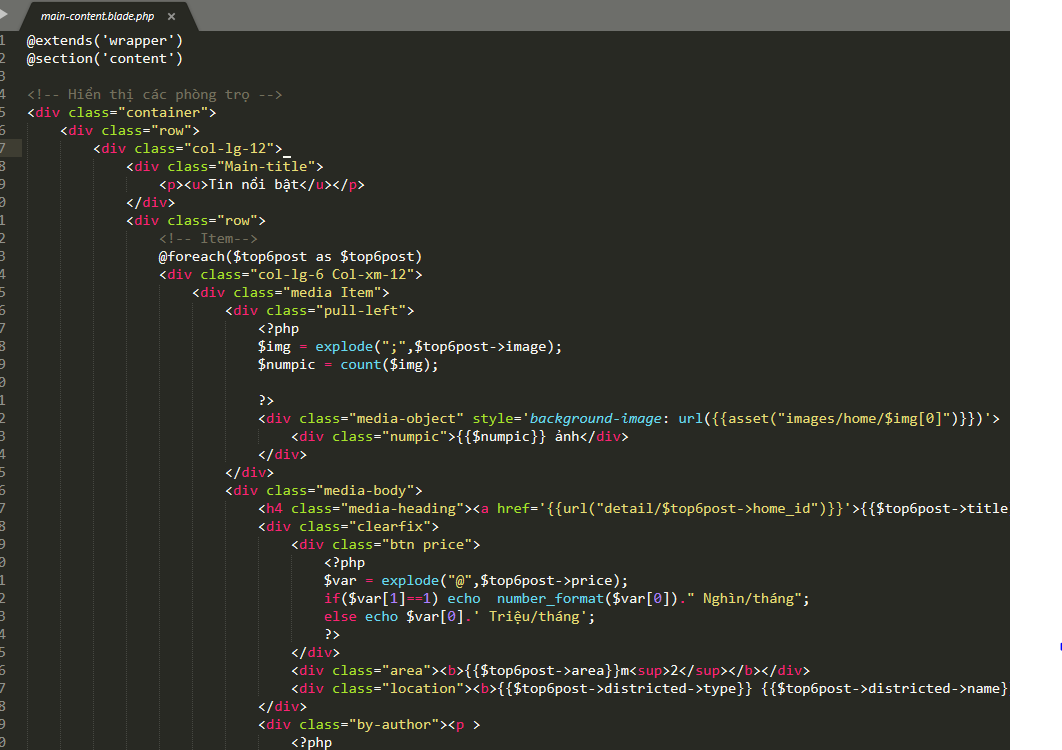
* Trong thư mục database/migration là nơi lưu trữ quản lý các table mà project chúng ta cần làm. Để tạo một table mới chúng ta dùng câu lệnh: *php artisan make:migrate {tên migration}*
* Chúng ta sử dụng lớp Schema trong laravel để tạo bảng cũng như rằng buộc, khóa chính, khóa ngoại, null … cho bảng cần tạo.

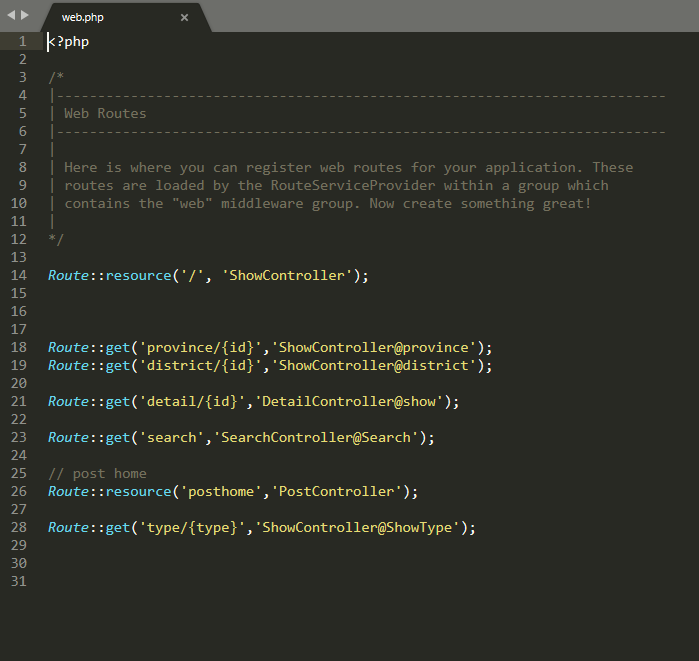
**Hình 2.15:** Sử dụng schema để tạo bảng

* Để chạy migration bảng vừa tạo ta dùng câu lệnh: *php artisan migrate* thì trên MySQL sẽ tạo một bảng như ta đã tạo ở migration.

**Hình 2.16:** Bảng users được tạo

* + 1. **View:**
* View là nơi giao tiếp với người dùng, hiển thị các thông tin, hình ảnh các nút xử lý mà người dùng muốn làm.
* Trong Laravel sử dụng Blade Template là một mẫu bố cục chung cho tất cả các trang có sử dụng nhưng thành phần giống nhau mà không phải viết lại toàn bộ, từ đó trên mỗi trang, chỉ cần thay đổi ở một số nơi được chỉ định trên trang từ template. Nó có thể code được PHP, HTML, CSS, JAVASCRIPT ở trên đó. Các file view dùng cho blade có phân đuôi tên là blade.php và được lưu trong resources/views.
* Để gọi các view Blade thì phải thông qua Route để định dạng yêu cầu (request) từ url định sẵn.

**Hình 2.17:** Một blade template trong laravel

****

**Hình 2.18:** Route trong laravel

* + 1. **Controller**
* Sau khi route nhận được request thì sẽ chuyển hướng đến cho controller và giao cho controller này xử lý. Controller sẽ gọi tới Model để tương tác với cơ sở dữ liệu và đưa dữ liệu tới view và view nhận dữ liệu chuyển thành những đoạn mã HTML rồi gửi về cho client.
* Tất cả controller trong laravel đều được chứa ở app/Http/Controllers và để tạo một controller chúng ta gõ lệnh: *php artisan make:controller {tên controller}*

**Hình 2.19:** Một file controller xử lý hiện thị dữ liệu từ model ra view

* + 1. **Kiểm thử.**
  1. **Thực nghiệm.**

# **CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* 1. **Đánh giá kết quả đạt được.**
* Với sự hướng dẫn nhiệt tình của anh Vũ Hoàng Quân cũng như anh Nguyễn Văn Khá và cô Nguyễn Thị Thảo Nguyên cùng toàn thể gia đình ARIS Việt Nam thì quá trình tham gia thực tập đã được hoàn thành.
* Sau khi được training về các package của Laravel cũng như là các plug-in của Jquery thì em nắm được khá chắc về khả năng rằng buộc giá trị validation của front-end. Tiếp đó, khả năng trình bày cũng như thể hiện giao diện front-end càng ngày càng gần gũi với người sử dụng, tạo nên sự tư tin khi được giao công việc và khả năng thiết kế một website hoàn chỉnh.
* Về back-end em nắm được khá vững hơn về các ràng buộc, xử lý trên server, response những thứ cần thiết để trả về client. Đổ dữ liệu từ database, thêm dữ liệu hay xuất ra file exel, pdf... cũng thành thục hơn.
* Trong quá trình thực tập có những thuận lợi cũng như khó khăn thì em không thể tránh khỏi những sai sót. Tuy nhiên, em đã nổ lực và hoàn thành nhiệm vụ được giao. Từ đó em rút ra được nhiều bài học quý giá về kỹ luật công ty, thái độ làm việc cũng như là khả năng giao tiếp, chuyên môn làm việc tại công ty là bàn đạp, hành trang quý giá cho em sau này.
* Một lần nữa em xin chân thành cảm ơn gia đình ARIS, và cô Nguyễn Thị Thảo Nguyên đã tạo điều kiện, hướng dẫn em trong suốt thời gian thực tập tại công ty.
  1. **Kiến thức lý thuyết**.
* Ôn lại những kiến thức cơ bản về lập trình Web từ Font-end đến Back-end gồm các ngôn ngữ HTML, CSS, JAVASCRIPT và PHP.
* Ôn lại quá trình quản lý dự án và quản lý kiểm thử, củng cố kiến thức về quá trình Agile và Scrum cũng như là xây dựng một bộ óc phân tích phần mềm.
  1. **Kỹ năng thực hành.**
* Thành thạo các quy trình xử lý font-end hay back-end. Cắt layout trong laravel chuẩn và ràng buộc trong form cũng như trên server.
* Thực hành đọc tài liệu Document cho từng ngôn ngữ mới hay một chức năng mới mà không cần phải hỏi hay chờ đợi người khác.
  1. **Hạn chế**
* Do thời gian quá ngắn, với kiến thức chưa đủ sâu nên còn chậm trong các quá trình xử lý code.
* Nắm bắt các chức năng mới còn chậm do chưa thành thạo đọc document.
* Phân tích nghiệp vụ chuyên sâu còn hổng.

**DANH MỤC TÀI LIỆU THAM KHẢO**

**[1] 12 Manifesto Agile:**

<https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>

**[2] About Aris:**

<https://aris-vn.com/>

**[3] Datatable Manual:**

<https://datatables.net/manual/>

**[4] Laravel Document:**

<http://laravel.com/docs/5.4>

**[5] Maatwebsite excel:**

<https://www.laragist.org/packages/maatwebsite/excel>

**[6] JQuery Validation Document:**

<https://jqueryvalidation.org/documentation/>

**[7] Git – Documentation:**

<https://git-scm.com/doc>

---Hết---