# LỜI CẢM ƠN

Để hoàn thành tốt bài báo cáo Đồ án tốt nghiệp này, lời đầu tiên em xin gửi lời cảm ơn chân thành tới các quý thầy cô Trường Đại học công nghệ thông tin và truyền thông Thái Nguyên nói chung, quý thầy cô Khoa công nghệ thông tin nói riêng đã dùng tâm huyết và tri thức của mình để dìu dắt, truyền đạt vốn kiến thức quý báu cho em trong suốt thời gian học tập tại trường.

Em xin chân thành thành cảm ơn cô *Th.S Nguyễn Thu Phương* đã tận tình hướng dẫn em trong quá trình thực hiện đồ án này. Nếu không có những lời hướng dẫn, chỉ bảo của cô thì em sẽ không thể hoàn thành đồ án này. Một lần nữa em xin chân thành cảm ơn cô.

Em xin gửi lời cảm ơn chân thành đến gia đình và bạn bè vì đã luôn là nguồn động viên to lớn, giúp đỡ em vượt qua những khó khăn trong suốt quá trình học tập.

Tuy nhiên, dù đã rất cố gắng hoàn thiện nhưng do thời gian cũng như vốn kiến thức có hạn của mình nên bài báo cáo này khó tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm cũng như những đóng góp quý báu của quý thầy cô và các bạn để em có thêm kinh nghiệm và tiếp tục hoàn thiện mình hơn nữa.

Một lần nữa em xin chân thành cảm ơn!

*Thái Nguyên, tháng 5 năm 2019*

Sinh viên

**Lê Công Nam**

# LỜI CAM ĐOAN

Tôi xin cam đoan đây là phần nghiên cứu và kết quả nghiên cứu của đồ án tốt nghiệp này là của riêng tôi, trung thực, không sao chép các đồ án khác. Nếu sai, tôi xin hoàn toàn chịu trách nhiệm và chịu mọi kỷ luật của khoa và nhà trường đề ra.

*Thái Nguyên, tháng 5 năm 2019*

Sinh viên

**Lê Công Nam**

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc9962869)

[LỜI CAM ĐOAN 2](#_Toc9962870)

[LỜI MỞ ĐẦU 5](#_Toc9962871)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 6](#_Toc9962872)

[1.1 Tổng quan HTML (Hypertext Markup Language) 6](#_Toc9962873)

[1.1.1 Giới thiệu HTML 6](#_Toc9962874)

[1.1.2 Cấu trúc chung của một trang HTML 6](#_Toc9962875)

[1.1.3 Các thẻ HTML cơ bản 7](#_Toc9962876)

[1.2 Tổng quan CSS (Cascading Style Sheets) 9](#_Toc9962877)

[1.2.1 Giới thiệu về CSS 9](#_Toc9962878)

[1.2.2 Quy tắc viết CSS 9](#_Toc9962879)

[1.2.3 Nhúng CSS vào HTML 9](#_Toc9962880)

[1.3. Giới thiệu PHP 11](#_Toc9962881)

[1.3.1 Khái niệm 11](#_Toc9962882)

[1.3.2 Lịch sự phát triển của PHP 11](#_Toc9962883)

[1.3.3 Tại sao nên dùng PHP 15](#_Toc9962884)

[1.3.4 Hoạt động của PHP 16](#_Toc9962885)

[1.3.5 Các loại thẻ trong PHP 17](#_Toc9962886)

[1.3.6 Biến, giá trị 17](#_Toc9962887)

[1.3.7. Các giá trị bên ngoài phạm vi PHP 18](#_Toc9962888)

[1.3.8 Biểu thức 19](#_Toc9962889)

[1.3.9 Các cấu trúc lệnh 19](#_Toc9962890)

[1.3.10 Các toán tử 22](#_Toc9962891)

[1.3.11 Lớp và đối tượng 24](#_Toc9962892)

[1.4 MySQL và PHP 24](#_Toc9962893)

[1.4.1 Giới thiệu cơ sở dữ liệu 25](#_Toc9962894)

[1.4.2 Mục đích sử dụng cơ sở dữ liệu 26](#_Toc9962895)

[1.4.3 Các thao tác cập nhật dữ liệu 27](#_Toc9962896)

[1.4.4 Các hàm thông dụng trong MySQL 27](#_Toc9962897)

[1.5 Sơ lược về Framework Laravel 28](#_Toc9962898)

[1.5.1 Tổng quan về Laravel 28](#_Toc9962899)

[1.5.2 Cấu trúc thư mục ứng dụng Laravel 29](#_Toc9962900)

[CHƯƠNG 2: KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 32](#_Toc9962901)

[2.1. Phát biểu bài toán 32](#_Toc9962902)

[2.2. Khảo sát một số website 32](#_Toc9962903)

[2.3. Phân tích yêu cầu hệ thống 35](#_Toc9962904)

[2.3.1. Các đối tượng phục vụ của hệ thống 35](#_Toc9962905)

[2.3.2. Các yêu cầu chức năng 35](#_Toc9962906)

[2.4. Các sơ đồ Use Case: 37](#_Toc9962907)

[2.5. Đặc tả chi tiết các use-case ( UC ) 38](#_Toc9962908)

[2.6. Biểu đồ trình tự 44](#_Toc9962909)

[2.7. Biểu đồ Hoạt động 49](#_Toc9962910)

[2.8. Biểu đồ lớp 53](#_Toc9962911)

[Chương 3: Xây dựng website 54](#_Toc9962912)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 56](#_Toc9962913)

[TÀI LIỆU THAM KHẢO 57](#_Toc9962914)

# LỜI MỞ ĐẦU

Xu hướng du lịch để trải nghiệm đang phát triển nhanh chóng thay cho việc du lịch để nghe nhìn như trước đây. Nắm bắt xu hướng đó của giới trẻ, các homestay ra đời đáp ứng đầy đủ nhu cầu của người dùng như là được sống, được trải nghiệm văn hóa, đời sống của người bản địa với giá cả phải chăng.

Đặc thù của việc kinh doanh homestay chính là đối tượng khách hàng và các “ông chủ homestay” hầu như đều là người trẻ. Vì vậy, các hoạt động marketing tiếp cận khách hàng không chỉ giới hạn ở truyền miệng mà phần lớn là online. Một trong những công cụ marketing được yêu thích nhất và mang lại hiệu quả tối ưu chính là **website homestay**.

Với lý do trên, em đã quyết định chọn đề tài:”**Xây dựng website đặt phòng Homestay**”. Báo cáo được chia làm 3 phần chính bao gồm:

**Chương 1**: **Cơ sở lý thuyết**

*Chương này tìm hiểu kiến thức cơ bản tạo lên một website.*

**Chương 2: Khảo sát, phân tích & thiết kế hệ thống website**

*Chương này khảo sát thực tế, phân tích bài toán, đưa ra các chức năng của website đặt phòng Homestay*

**Chương 3**: **Xây dựng website**

*Chương này giới thiệu một số chức năng chính của website đặt phòng Homestay*

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1 Tổng quan HTML (Hypertext Markup Language)

### 1.1.1 Giới thiệu HTML

**HTML** (Hypertext Markup Language – Ngôn ngữ đánh dấu siêu văn bản) là một sự định dạng để báo cho trình duyệt Web (Web browser) biết cách để hiển thị một trang Web. Các trang Web thực ra không có gì khác ngoài văn bản cùng với các thẻ (tag) HTML được sắp xếp đúng cách hoặc các đoạn mã để trình duyệt Web biết cách để thông dịch và hiển thị chúng lên trên màn hình.

Tên gọi ngôn ngữ đánh dấu siêu văn bản có ý nghĩa như sau:

* Đánh dấu (Markup): HTML là một ngôn ngữ của các thẻ đánh dấu – Tag. Các thẻ này xác định cách thức trình bày đoạn văn bản tương ứng trên màn hình.
* Ngôn ngữ (Language): HTML là một ngôn ngữ tương tự như các ngôn ngữ lập trình, tuy nhiên đơn giản hơn. Nó có cú pháp chặt chẽ để viết các lệnh thực hiên việc trình diên văn bản. Các từ khóa có ý nghĩa xác định được công đồng Internet thừa nhận và sử dụng.
* Văn bản (Text): HTML đầu tiên và trước hết là để trình bày văn bản và dựa trên nền tảng là một văn bản. Các thành phần khác như hình ảnh, âm thanh, hoạt hình đều phải “cắm neo” vào một đoạn văn bản nào đó.
* Siêu văn bản (HyperText): HTML cho phép liên kết nhiều trang văn bản rải rác khắp nơi trên Internet. Nó có tác dụng che dấu sự phức tạp của Internet đối với người sử dụng. Người dùng Internet có thể đọc văn bản mà không cần biết đến văn bản đó nằm ỏ đâu, hệ thống được xây dựng phức tạp như thế nào. HTML thực sự đã vượt ra ngoài khuôn khổ khái niệm văn bản cổ điển.

### 1.1.2 Cấu trúc chung của một trang HTML

<!DOCTYPE html>

<html>

<head>

<title>Tiêu đề trang web</title>

</head>

<body>

Nội dung của trang web

</body>

</html>

Một tài liệu HTML gồm 3 phần cơ bản:

* Phần html: Mọi tài liệu HTML phải bắt đầu bằng thẻ mở HTML <html> và kết thúc bằng thẻ đóng HTML </html>
* Thẻ html báo cho trình duyệt biết nội dung giữa hai thẻ này là một tài liệu html
* Phần tiêu đề: Phần tiêu đề bắt đầu bằng thẻ <head> và kết thúc bởi thẻ </head>. Phần này chứa tiêu đề mà được hiển thị trên thanh điều hướng của trang Web. Tiêu đề nằm trong thẻ title, bắt đầu bằng thẻ <title> và kết thúc là thẻ </title>.

Tiêu đề là phần khá quan trọng. Khi người dùng tìm kiếm thông tin, tiêu đề của trang Web cung cấp từ khóa chính yếu cho việc tìm kiếm.

* Phần thân: phần này nằm sau phần tiêu đề. Phầ̀n thân bao gồm văn bản, hình ảnh và các liên kết mà bạn muốn hiển thị trên trang web của mình. Phần thân bắt đầu bằng thẻ <body> và kết thúc bằng thẻ </body>

### 1.1.3 Các thẻ HTML cơ bản

* **Thẻ <head>.... </head>:** Tạo đầu mục trang
* **Thẻ <title>... </title>:** Tạo tiêu đề trang trên thanh tiêu đề, đây là thẻ bắt buộc. Thẻ **title** cho phép bạn trình bày chuỗi trên thanh tựa đề của trang Web mỗi khi trang Web đó được duyệt trên trình duyệt Web.
* **Thẻ <body>... </body>:** Tất cả các thông tin khai báo trong thẻ **<body>** đều có thể xuất hiện trên trang Web. Những thông tin này có thể nhìn thấy trên trang Web.
* **Thẻ <p>… </p>:**Tạo một đoạn mới.
* **Thẻ định dạng bảng <table>… </table>:** Đây là thẻ định dạng bảng trên trang Web. Sau khi khai báo thẻ này, bạn phải khai báo các thẻ hàng <tr> và thẻ cột <td> cùng với các thuộc tính của nó.
* **Thẻ hình ảnh <img>:** Cho phép bạn chèn hình ảnh vào trang Web. Thẻ này thuộc loại thẻ không có thẻ đóng.
* **Thẻ liên kết <a>... </a>:** Là loại thẻ dùng để liên kết giữa các trang Web hoặc liên kết đến địa chỉ Internet, Mail hay Intranet(URL) và địa chỉ trong tập tin trong mạng cục bộ (UNC).
* **Các thẻ Input:** Thẻ Input cho phép người dùng nhập dữ liệu hay chỉ thị thực thi một hành động nào đó, thẻ Input bao gồm các loại thẻ như: text, password, submit, button, reset, checkbox, radio, image.
* **Thẻ Textarea < Textarea>.... < \Textarea>:** Thẻ Textarea cho phép người dùng nhập liệu với rất nhiều dòng. Với thẻ này bạn không thể giới hạn chiều dài lớn nhất trên trang Web.
* **Thẻ Select:**

Thẻ Select cho phép người dùng chọn phần tử trong tập phương thức đã được định nghĩa trước.

Nếu thẻ Select cho phép người dùng chọn một phần tử trong danh sách phần tử thì thẻ Select sẽ giống như combobox.

Nếu thẻ Select cho phép người dùng chọn nhiều phần tử cùng một lần trong danh sách phần tử, thẻ Select đó là dạng listbox.

* **Thẻ Form:**

Khi bạn muốn submit dữ liệu người dùng nhập từ trang Web phía Client lên phía Server, bạn có hai cách để làm điều này ứng với hai phương thức POST và GET trong thẻ form.

Trong một trang Web có thể có nhiều thẻ Form khác nhau, nhưng các thẻ Form này không được lồng nhau, mỗi thẻ form sẽ được khai báo hành động (action) chỉ đến một trang khác.

## 1.2 Tổng quan CSS (**C**ascading **S**tyle **S**heets)

### 1.2.1 Giới thiệu về CSS

CSS – được dùng để miêu tả cách trình bày các tài liệu viết bằng HTML. CSS được hiểu một cách đơn giản đó là cách mà chúng ta thêm các kiểu hiển thị cho một tài liệu Web.

Các thuộc tính cơ bản trong CSS: Color, Background, Line-height, Font-size, Border, Height, Width, …

### 1.2.2 Quy tắc viết CSS

* Cú pháp: Selector {property: value;}

+ Bộ chọn (selector) là yếu tố HTML mà quy tắc CSS áp dụng tới.

+ Property: Xác định thuộc tính mà bạn muốn thay đổi tương ứng tới bộ chọn.

+ Value: Giá trị mà thuộc tính có thể nhận.

+ Thuộc tính và giá trị được phân biệt nhau bởi dấu hai chấm (:).

+ Mỗi khai báo được phân biệt nhau bởi dấu chấm phảy (;).

### 1.2.3 Nhúng CSS vào HTML

Có ba cách khác nhau để nhúng CSS vào trong một tài liệu HTML:

* *Cách thứ 1*: Nội tuyến (kiểu thuộc tính)

Đây là một phương pháp nguyên thủy nhất để nhúng CSS vào một tài liệu HTML bằng cách nhúng vào từng thẻ HTML muốn áp dụng. Và dĩ nhiên trong trường hợp này chúng ta sẽ không cần selector trong cú pháp.

Nếu bạn muốn áp dụng nhiều thuộc tính cho nhiều thẻ HTML khác nhau thì không nên dùng cách này.

<html>

<head>

<title>Ví dụ</title>

</head>

<body style="background-color=#FFF;">

<p style="color:green">Hello World</p>

</body>

</html>

* *Cách thứ 2:* Bên trong (thẻ style)

Thật ra nếu nhìn kỹ chúng ta cũng nhận ra đây chỉ là một phương cách thay thế cách thứ nhất bằng cách rút tất cả các thuộc tính CSS vào trong thẻ style (để tiện cho công tác bảo trì, sửa chữa ấy mà).

<html>

<head>

<title>Ví dụ</title>

<style type="text/css">

body { background-color:#FFF }

p { color:#00FF00 }

</style>

</head>

<body>

<p Hello World </p>

</body>

</html>

* Cách 3: Bên ngoài (liên kết với một file CSS bên ngoài)

Tương tự như cách 2 nhưng thay vì đặt tất cả các mã CSS trong thẻ style chúng ta sẽ đưa chúng vào trong một file CSS (có phần mở rộng .css) bên ngoài và liên kết nó vào trang web bằng thuộc tính href trong thẻ link.

Đây là cách làm được khuyến cáo, nó đặc biệt hữu ích cho việc đồng bộ hay bảo trì một website lớn sử dụng cùng một kiểu mẫu.

Sự ưu tiên:

Trước khi thực thi CSS cho một trang web. Trình duyệt sẽ đọc toàn bộ CSS mà trang web có thể được áp dụng, bao gồm: CSS mặc định của trình duyệt, file CSS bên ngoài liên kết vào trang web, CSS nhúng trong thẻ <style> và các CSS nội tuyến. Sau đó, trình duyệt sẽ tổng hợp toàn bộ CSS này vào một CSS ảo, và nếu có các thuộc tính CSS giống nhau thì thuộc tính CSS nào nằm sau sẽ được ưu tiên sử dụng (cái này cũng giống như chương trình “Ai Là Triệu Phú” trên truyền hình vậy, chỉ câu trả lời sau cùng mới được chấp nhận (smile)). Theo nguyên tắc đó trình duyệt của bạn sẽ ưu tiên cho các CSS nội tuyến > CSS bên trong > CSS bên ngoài > CSS mặc định của trình duyệt.

## 1.3. Giới thiệu PHP

### 1.3.1 Khái niệm

PHP là chữ viết tắt của “Personal Home Page” do Rasmus Lerdorf tạo ra năm 1994. Vì tính hữu dụng của nó và khả năng phát triển, PHP bắt đầu được sử dụng trong môi trường chuyên nghiệp và nó trở thành “PHP: Hypertext Preprocessor”.

PHP là một ngôn ngữ lập trình được kết nối chặt chẽ với máy chủ, là một công nghệ phía máy chủ (Server-Side) và không phụ thuộc vào môi trường (cross-platform). Đây là hai yếu tố rất quan trọng, thứ nhất khi nói công nghệ phía máy chủ tức là nói đến mọi thứ trong PHP đều xảy ra trên máy chủ, thứ hai, chính vì tính chất không phụ thuộc môi trường cho phép PHP chạy trên hầu hết trên các hệ điều hành như Windows, Unix và nhiều biến thể của nó... Đặc biệt các mã kịch bản PHP viết trên máy chủ này sẽ làm việc bình thường trên máy chủ khác mà không cần phải chỉnh sửa hoặc chỉnh sửa rất ít.

### 1.3.2 Lịch sự phát triển của PHP

**PHP/FI**:

PHP được phát triển từ một sản phẩm có tên là PHP/FI. PHP/FI do Rasmus Lerdorf tạo ra năm 1994, ban đầu được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho bộ mã kịch bản này là 'Personal Home Page Tools'. Khi cần đến các chức năng rộng hơn, Rasmus đã viết ra một bộ thực thi bằng C lớn hơn để có thể truy vấn tới các cơ sở dữ liệu và giúp cho người sử dụng phát triển các ứng dụng web đơn giản. Rasmus đã quyết định công bố mã nguồn của PHP/FI cho mọi người xem, sử dụng cũng như sửa các lỗi có trong nó đồng thời cải tiến mã nguồn.

PHP/FI, viết tắt từ "Personal Home Page/Forms Interpreter", bao gồm một số các chức năng cơ bản cho PHP như ta đã biết đến chúng ngày nay. Nó có các biến kiểu như Perl, thông dịch tự động các biến của form và cú pháp HTML nhúng. Cú pháp này giống như của Perl, mặc dù hạn chế hơn nhiều, đơn giản và có phần thiếu nhất quán.

Vào năm 1997, PHP/FI 2.0, lần viết lại thứ hai của phiên bản C, đã thu hút được hàng ngàn người sử dụng trên toàn thế giới với xấp xỉ 50.000 tên miền đã được ghi nhận là có cài đặt nó, chiếm khoảng 1% số tên miền có trên mạng Internet. Tuy đã có tới hàng nghìn người tham gia đóng góp vào việc tu chỉnh mã nguồn của dự án này thì vào thời đó nó vẫn chủ yếu chỉ là dự án của một người.

PHP/FI 2.0 được chính thức công bố vào tháng 11 năm 1997, sau một thời gian khá dài chỉ được công bố dưới dạng các bản beta. Nhưng không lâu sau đó, nó đã được thay thế bởi các bản alpha đầu tiên của PHP 3.0.

**PHP 3**

PHP 3.0 là phiên bản đầu tiên cho chúng ta thấy một hình ảnh gần gũi với các phiên bản PHP mà chúng ta được biết ngày nay. Nó đã được Andi Gutmans và Zeev Suraski tạo ra năm 1997 sau khi viết lại hoàn toàn bộ mã nguồn trước đó. Lý do chính mà họ đã tạo ra phiên bản này là do họ nhận thấy PHP/FI 2.0 hết sức yếu kém trong việc phát triển các ứng dụng thương mại điện tử mà họ đang xúc tiến trong một dự án của trường đại học. Trong một nỗ lực hợp tác và bắt đầu xây dựng dựa trên cơ sở người dùng đã có của PHP/FI, Andi, Rasmus và Zeev đã quyết định hợp tác và công bố PHP 3.0 như là phiên bản thế hệ kế tiếp của PHP/FI 2.0, và chấm dứt phát triển PHP/FI 2.0.

Một trong những sức mạnh lớn nhất của PHP 3.0 là các tính năng mở rộng mạnh mẽ của nó. Ngoài khả năng cung cấp cho người dùng cuối một cơ sở hạ tầng chặt chẽ dùng cho nhiều cơ sở dữ liệu, giao thức và API khác nhau, các tính năng mở rộng của PHP 3.0 đã thu hút rất nhiều nhà phát triển tham gia và đề xuất các mô đun mở rộng mới. Hoàn toàn có thể kết luận được rằng đây chính là điểm mấu chốt dẫn đến thành công vang dội của PHP 3.0. Các tính năng khác được giới thiệu trong PHP 3.0 gồm có hỗ trợ cú pháp hướng đối tượng và nhiều cú pháp ngôn ngữ nhất quán khác.

Ngôn ngữ hoàn toàn mới đã được công bố dưới một cái tên mới, xóa bỏ mối liên hệ với việc sử dụng vào mục đích cá nhân hạn hẹp mà cái tên PHP/FI 2.0 gợi nhắc. Nó đã được đặt tên ngắn gọn là 'PHP', một kiểu viết tắt hồi quy của "PHP: Hypertext Preprocessor".

Vào cuối năm 1998, PHP đã phát triển được con số cài đặt lên tới hàng chục ngàn người sử dụng và hàng chục ngàn Web site báo cáo là đã cài nó. Vào thời kì đỉnh cao, PHP 3.0 đã được cài đặt cho xấp xỉ 10% số máy chủ Web có trên mạng Internet.

PHP 3.0 đã chính thức được công bố vào tháng 6 năm 1998, sau thời gian 9 tháng được cộng đồng kiểm nghiệm.

**PHP 4**

Vào mùa đông năm 1998, ngay sau khi PHP 3.0 chính thức được công bố, Andi Gutmans và Zeev Suraski đã bắt đầu bắt tay vào việc viết lại phần lõi của PHP. Mục đích thiết kế là nhằm cải tiến tốc độ xử lý các ứng dụng phức tạp, và cải tiến tính mô đun của cơ sở mã PHP. Những ứng dụng như vậy đã chạy được trên PHP 3.0 dựa trên các tính năng mới và sự hỗ trợ khá nhiều các cơ sở dữ liệu và API của bên thứ ba, nhưng PHP 3.0 đã không được thiết kế để xử lý các ứng dụng phức tạp như thế này một cách có hiệu quả.

Một động cơ mới, có tên 'Zend Engine' (ghép từ các chữ đầu trong tên của Zeev và Andi), đã đáp ứng được các nhu cầu thiết kế này một cách thành công, và lần đầu tiên được giới thiệu vào giữa năm 1999. PHP 4.0, dựa trên động cơ này, và đi kèm với hàng loạt các tính năng mới bổ sung, đã chính thức được công bố vào tháng 5 năm 2000, gần 2 năm sau khi bản PHP 3.0 ra đời. Ngoài tốc độ xử lý được cải thiện rất nhiều, PHP 4.0 đem đến các tính năng chủ yếu khác gồm có sự hỗ trợ nhiều máy chủ Web hơn, hỗ trợ phiên làm việc HTTP, tạo bộ đệm thông tin đầu ra, nhiều cách xử lý thông tin người sử dụng nhập vào bảo mật hơn và cung cấp một vài các cấu trúc ngôn ngữ mới.

Với PHP 4, số nhà phát triển dùng PHP đã lên đến hàng trăm nghìn và hàng triệu site đã công bố cài đặt PHP, chiếm khoảng 20% số tên miền trên mạng Internet.

Nhóm phát triển PHP cũng đã lên tới con số hàng nghìn người và nhiều nghìn người khác tham gia vào các dự án có liên quan đến PHP như  PEAR, PECL và tài liệu kĩ thuật cho PHP.

**PHP 5**

Sự thành công hết sức to lớn của PHP 4.0 đã không làm cho nhóm phát triển PHP tự mãn. Cộng đồng PHP đã nhanh chóng giúp họ nhận ra những yếu kém của PHP 4 đặc biệt với khả năng hỗ trợ lập trình hướng đối tượng (OOP), xử lý XML, không hỗ trợ giao thức máy khách mới của MySQL 4.1 và 5.0, hỗ trợ dịch vụ web yếu. Những điểm này chính là mục đích để Zeev và Andi viết Zend Engine 2.0, lõi của PHP 5.0. Một thảo luận trên Slashdot đã cho thấy việc phát triển PHP 5.0 có thể đã bắt đầu vào thời điểm tháng 12 năm 2002 nhưng những bài phỏng vấn Zeev liên quan đến phiên bản này thì đã có mặt trên mạng Internet vào khoảng tháng 7 năm 2002. Ngày 29 tháng 6 năm 2003, PHP 5 Beta 1 đã chính thức được công bố để cộng đồng kiểm nghiệm. Đó cũng là phiên bản đầu tiên của Zend Engine 2.0. Phiên bản Beta 2 sau đó đã ra mắt vào tháng 10 năm 2003 với sự xuất hiện của hai tính năng rất được chờ đợi: Iterators, Reflection nhưng namespaces một tính năng gây tranh cãi khác đã bị loại khỏi mã nguồn. Ngày 21 tháng 12 năm 2003: PHP 5 Beta 3 đã được công bố để kiểm tra với việc phân phối kèm với Tidy, bỏ hỗ trợ Windows 95, khả năng gọi các hàm PHP bên trong XSLT, sửa chữa nhiều lỗi và thêm khá nhiều hàm mới. PHP năm bản chính thức đã ra mắt ngày 13 tháng 7 năm 2004 sau một chuỗi khá dài các bản kiểm tra thử bao gồm Beta 4, RC 1, RC2, RC3. Mặc dù coi đây là phiên bản sản xuất đầu tiên nhưng PHP 5.0 vẫn còn một số lỗi trong đó đáng kể là lỗi xác thực HTTP.

Ngày 14 tháng 7 năm 2005, PHP 5.1 Beta 3 được PHP Team công bố đánh dấu sự chín muồi mới của PHP với sự có mặt của PDO, một nỗ lực trong việc tạo ra một hệ thống API nhất quán trong việc truy cập cơ sở dữ liệu và thực hiện các câu truy vấn. Ngoài ra, trong PHP 5.1, các nhà phát triển PHP tiếp tục có những cải tiến trong nhân Zend Engine 2, nâng cấp mô đun PCRE lên bản PCRE 5.0 cùng những tính năng và cải tiến mới trong SOAP, streams và SPL.

**PHP 7**

Trong năm 2014 và 2015, một phiên bản PHP chính mới đã được phát triển, được đánh số là PHP 7. Việc đánh số phiên bản này liên quan đến một số cuộc tranh luận. Trong khi thử nghiệm Unicode 6 PHP chưa bao giờ được phát hành, một số bài viết và tiêu đề sách tham chiếu tên PHP 6, điều này có thể gây nhầm lẫn nếu một bản phát hành mới được sử dụng lại tên. Sau khi bỏ phiếu, tên PHP 7 đã được chọn.

Nền tảng của PHP 7 là một nhánh PHP ban đầu được gọi là PHP thế hệ tiếp theo (phpng). Nó được viết bởi Dmitry Stogov, Xinchen Hui và Nikita Popov và nhằm mục đích tối ưu hóa hiệu năng PHP bằng cách tái cấu trúc Công cụ Zend trong khi duy trì khả năng tương thích ngôn ngữ gần hoàn toàn. Kể từ ngày 14 tháng 7 năm 2014, các tiêu chuẩn dựa trên WordPress, được dùng làm bộ tiêu chuẩn chính cho dự án phpng, cho thấy hiệu suất tăng gần 100%. Các thay đổi từ phpng cũng được kỳ vọng sẽ giúp cải thiện hiệu suất trong tương lai, vì các cấu trúc dữ liệu nhỏ gọn hơn và các thay đổi khác được xem là phù hợp hơn cho việc chuyển đổi thành công tới trình biên dịch JIT. Do những thay đổi quan trọng, Zend Engine được làm lại được gọi là Zend Engine 3, thành công Zend Engine 2 được sử dụng trong PHP 5.

### 1.3.3 Tại sao nên dùng PHP

Để thiết kế Web động có rất nhiều ngôn ngữ lập trình khác nhau để lựa chọn, mặc dù cấu hình và tính năng khác nhau nhưng chúng vẵn đưa ra những kết quả giống nhau. Chúng ta có thể lựa chọn cho mình một ngôn ngữ: ASP, PHP, Java, Perl... và một số loại khác nữa. Vậy tại sao chúng ta lại nên chọn PHP. Rất đơn giản, có những lí do sau mà khi lập trình Web chúng ta không nên bỏ qua sự lựa chọn tuyệt vời này.

PHP được sử dụng làm Web động vì nó nhanh, dễ dàng, tốt hơn so với các giải pháp khác.

PHP có khả năng thực hiện và tích hợp chặt chẽ với hầu hết các cơ sở dữ liệu có sẵn, tính linh động, bền vững và khả năng phát triển không giới hạn.

Đặc biệt PHP là mã nguồn mở do đó tất cả các đặc tính trên đều miễn phí, và chính vì mã nguồn mở sẵn có nên cộng đồng các nhà phát triển Web luôn có ý thức cải tiến nó, nâng cao để khắc phục các lỗi trong các chương trình này

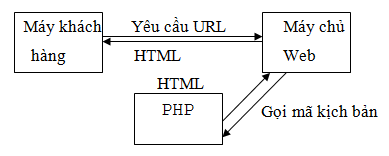
PHP vừa dễ với người mới sử dụng vừa có thể đáp ứng mọi yêu cầu của các lập trình viên chuyên nghiệp, mọi ý tuởng của các bạn PHP có thể đáp ứng một cách xuất sắc.

Cách đây không lâu ASP vốn được xem là ngôn ngữ kịch bản phổ biến nhất, vậy mà bây giờ PHP đã bắt kịp ASP, bằng chứng là nó đã có mặt trên 12 triệu Website.

### 1.3.4 Hoạt động của PHP

Vì PHP là ngôn ngữ của máy chủ nên mã lệnh của PHP sẽ tập trung trên máy chủ để phục vụ các trang Web theo yêu cầu của người dùng thông qua trình duyệt.

Sơ đồ hoạt động:



Hình 1. 1 Sơ đồ hoạt động của PHP

Khi người dùng truy cập Website viết bằng PHP, máy chủ đọc mã lệnh PHP và xử lí chúng theo các hướng dẫn được mã hóa. Mã lệnh PHP yêu cầu máy chủ gửi một dữ liệu thích hợp (mã lệnh HTML) đến trình duyệt Web. Trình duyệt xem nó như là một trang HTML têu chuẩn. Như ta đã nói, PHP cũng chính là một trang HTML nhưng có nhúng mã PHP và có phần mở rộng là HTML. Phần mở của PHP được đặt trong thẻ mở <?php và thẻ đóng ?> . Khi trình duyệt truy cập vào một trang PHP, Server sẽ đọc nội dung file PHP lên và lọc ra các đoạn mã PHP và thực thi các đoạn mã đó, lấy kết quả nhận được của đoạn mã PHP thay thế vào chỗ ban đầu của chúng trong file PHP, cuối cùng Server trả về kết quả cuối cùng là một trang nội dung HTML về cho trình duyệt.

### 1.3.5 Các loại thẻ trong PHP

Có 4 loại thẻ khác nhau mà bạn có thể sử dụng khi thiết kế trang PHP:

* Kiểu **Short**:Thẻ mặc định mà các nhà lập trình PHP thường sử dụng.

Ví dụ:

<? Echo “Well come to PHP. ”; ?>

* Kiều đinh dạng XML: Thẻ này có thể sử dụng với văn bản đinh dạng XML

Ví dụ:

<? Phpecho“Well come to PHP with XML”; >?

* Kiểu **Script**: Trong trường hợp bạn sử dụng PHP như một script tương tự khai báo JavaScipt hay VBScript:

Ví dụ:

**<**scriptlanguage**= “**php**”>**

echo “Php Script”;

**</**script**>**

* Kiểu **ASP**: Trong trường hợp bạn khai báo thẻ PHP như một phần trong trang ASP.

Ví dụ:

**<%** echo **“**PHP – ASP**”; %>**

PHP và HTML là các ngôn ngữ không “nhạy cảm” với khoảng trắng, khoảng trắng có thể được đặt xung quanh để các mã lệnh để rõ ràng hơn. Chỉ có khoảng trắng đơn có ảnh hưởng đến sự thể hiện của trang Web (nhiều khoảng trắng liên tiếp sẽ chỉ thể hiện dưới dạng một khoảng trắng đơn).

### 1.3.6 Biến, giá trị

PHP quy định một biến được biểu diễn bắt đầu bằng dấu $, sau đó là một chữ cái hoặc dấu gạch dưới.

**1.3.6.1** **Một số biến đã được khai báo sẵn**

HTTP\_GET\_VARS: Mảng các giá trị nguyên truyền tới **script** thông qua phương thức HTTP GET. Chỉ có tác dụng nếu “**track\_vars**”. Trong cấu hình được đặt hoặc chỉ dẫn **<? Php\_track\_vars?>.**

HTTP\_POST\_VARS: Mảng các giá trị nguyên truyền tới script thông qua phương thức HTTP POST.

HTTP\_COOKIE\_VARS: Một mảng các giá trị được truyền tới script hiện tại bằng HTTP cookie. Chỉ có tác dụng nếu “**track\_vars**” trong cấu hình được đặt hoặc chỉ dẫn **<?php\_track\_vars?>...**

**1.3.6.2** **Phạm vi giá trị**

PHP coi một biến có một giới hạn. Để xác định một biến toàn cục (global) có tác dụng trong một hàm ta cần khai báo lại. Nếu không có giá trị của biến sẽ được coi như là cục bộ trong hàm.

**1.3.6.3** **Tên biến**

Một biến có thể gắn với một cái tên

Ví dụ:

$a= “chao”;

$$a= “cacban”;

* $Chao= “cacban”

Và echo”$a${chao}”;

Kết quả sẽ là “chaocacban”

### 1.3.7. Các giá trị bên ngoài phạm vi PHP

HTML Form: Khi 1 giá trị gắn với 1 file php qua phương thức POST

Ví dụ:

<form action = “top. php” method= “post”>

Name: < input type = “text” name = “name” ><BR>

<input type = “Submit” value= “Submit”>

</form>

PHP sẽ tạo 1 biến $name bao gồm mọi giá trị trong trường Name của Form

PHP có thể hiểu được một mảng một chiều gồm các giá trị trong một Form

Vì vậy, bạn có thể nhóm những giá trị liên quan lại với nhau hoặc sử dụng đặc tính này để nhận các giá trị từ 1 đầu vào tuỳ chọn.

Khi track\_vars được đặt trong cấu hình hoặc có chỉ dẫn <?php\_track\_vars?>. Các giá trị được submit sẽ lấy ra qua phương thức GET và POST có thể lấy ra từ 2 mảng toàn cục $HTTP\_POST\_VARS và $HTTP\_GET\_VARS

### 1.3.8 Biểu thức

Biểu thức là một thành phần quan trọng trong PHP. Một dạng cơ bản nhất của biểu thức bao gồm các biến và hằng số. PHP hỗ trợ 3 kiểu giá trị cơ bản nhất: Số nguyên, số thực và xâu. Ngoài ra còn có mảng và đối tượng. Mỗi kiểu giá trị này có thể gán cho các biến hay làm giá trị ra của các hàm.

### 1.3.9 Các cấu trúc lệnh

**1.3.9.1** **Các lệnh điều kiện và toán tử**

Mỗi câu lệnh điều kiện bao gồm một mệnh đề **if:**

**If(điều kiện){**

**//thực hiện một điều gì đó**

**}**

Điều kiện này có thể được mở rộng thành:

**If(điều kiện){**

**//thực hiện một điều gì đó**

**}**

**else {**

**//thực hiện một điều khác**

**}**

Và:

**If(điều kiện 1){**

**//thực hiện một điều gì đó**

**}**

**elseif(điều kiện 2) {**

**//thực hiện một điều khác**

**}else {**

**//thực hiện một điều khác nữa**

**}**

Các toán tử thường được sử dụng với câu lệnh điều kiện

|  |  |  |  |
| --- | --- | --- | --- |
| **Kí hiệu** | **Ý nghĩa** | **Dạng** | **Ví dụ** |
| = | Được gán giá trị của | Gán | $n=1 |
| == | Bằng | So sánh | $x==$y |
| != | Không bằng | So sánh | $x!=$y |
| < | Nhỏ hơn | So sánh | $x<$y |
| > | Lớn hơn | So sánh | $x>$y |
| <= | Nhỏ hơn hoặc bằng | So sánh | $x<=$y |
| >= | Lớn hơn hoặc bằng | So sánh | $x>=$y |
| ! | Phủ định | Logic | !$x |
| && | AND | Logic | $x && $y |
| || | OR | Logic | $x || $y |

PHP còn có một dạng câu lệnh điều kiện nữa là **Swich**. Câu lệnh này rất phù hợp cho việc thay thế nếu câu lệnh **if-elseif-else** quá dài. Cú pháp của câu lệnh **switch** như sau:

**Switch($variable){**

**Case ‘giá trị 1’:**

**//thực hiện lệnh**

**Break;**

**Case ‘giá trị 2’:**

**//thực hiện lệnh**

**Break;**

**Default:**

**//thực hiện lệnh**

**}**

#### **1.3.9.2 Phát biểu vòng lặp While**

Phát biểu đơn giản nhất trong PHP là vòng lặp While, cho phép thực thi khối lệnh trong While cho đến điều kiện của While là True như cú pháp:

**While (condition){**

**Expression;**

**}**

Trong đó:

**Condition:** Biểu thức điều kiện, biến,...

**Expression:** Khối lệnh trong vòng lặp while

**1.3.9.3 Vòng lặp For**

**For(expression1;condition;expression2**)

{

**expression3**;

}

Trong đó:

**expression1:** là một câu lệnh gán giá trị ban đầu cho biến điều khiển trước khi thực hiên vòng lặp, hoặc là một biến có giá trị sẵn mà ta đã truyền vào cho nó trước khi tạo vòng lặp này, lệnh này được thực hiện duy nhất một lần.

**condition**: là một biểu thức quan hệ xác định điều kiện thoát khỏi vòng lặp.

**expression2:** Xác định biến điều khiển sẽ bị thay đổi như thế nào sau mỗi lần lặp được lặp lại (thường là tăng hoặc giảm giá trị của biến điều khiển).

Ba biểu thức trên được cách nhau bởi dấu chấm phẩy, vòng lặp sẽ lặp khi biểu thức điều kiện đúng, khi biểu thức điều kiện sai thì vòng lặp sẽ dừng và thoát, và ta sử dụng các toán tử quan hệ và toán tử logic trong các biểu thức điều kiện để điều khiển vòng lặp.

* + - 1. **Vòng lặp do... while**

**Do{**

**expression**

**}**

**while(condition)**

Trong đó:

**Expression**: khối lệnh bên trong vòng lặp **do... while**

**Condition**: biểu thức điều kiện, biến...

Vòng lặp while sẽ kiểm tra điều kiện trước rồi thực hiện câu lệnh bên trong vòng lặp, còn vòng lặp **do while**thì ngược lại sẽ thực hiện câu lệnh bên trong vòng lặp trước rồi mới kiểm tra điều kiện. Nếu điều kiện đúng thì sẽ thực hiện tiếp vòng lặp kế tiếp, nếu điều kiện sai thì sẽ dừng vòng lặp. **Vòng lặp do while trong php**luôn luôn thực hiện ít nhất một lần lặp vì nó thực hiện trước rồi mới kiểm tra điều kiện.

### 1.3.10 Các toán tử

**Các phép số học:** +, -, \*, /%

**Các toán tử logic:** And, or, xor: &&, ||, !

**Toán tử thao tác với bít:** &, |, ^, ~, <<, >>

**Toán tử so sánh:** ==, !=, <, >, <=, >=, ===

**Toán tử điều khiển lỗi:** @

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Mô tả** | **Ví dụ** |
| + | Cộng 2 số | A + B = 30 |
| - | Trừ 2 số | A - B = -10 |
| \* | Nhân 2 số | A \* B = 200 |
| / | Chia 2 số | B / A = 2 |
| % | Toán tử Modulo - lấy số dư còn lại sau khi chia 2 số | B % A = 0 |
| ++ | Cộng thêm 1 | A++ = 11 |
| -- | Trừ đi 1 | A-- = 9 |
| == | Kiểm tra giá trị 2 toán hạng có bằng nhau hay không, nếu bẳng trả về true, ngược lại là false. | (A == B) is not true. |
| != | Kiểm tra giá trị 2 toán hạng có không bằng nhau hay không, nếu không bẳng trả về true, ngược lại là false. | (A != B) is true. |
| > | Kiểm tra toán hạng bên trái có lớn hơn bên phải hay không, nếu đúng trả về true. | (A > B) is not true. |
| < | Kiểm tra toán hạng bên trái có nhỏ hơn bên phải hay không, nếu đúng trả về true. | (A < B) is true. |
| >= | Kiểm tra toán hạng bên trái có lớn hơn hoặc bằng bên phải hay không, nếu đúng trả về true. | (A >= B) is not true. |
| <= | Kiểm tra toán hạng bên trái có nhỏ hơn hoặc bằng bên phải hay không, nếu đúng trả về true. | (A <= B) is true. |
| = | Gán giá trị toán hạng bên phải sang toán hạng bên trái | C = A + B sẽ gán giá trị của A + B cho C |
| += | Cộng toán hạng bên trái với toán hạng bên phải và gán kết quả cho toán hạng bên trái | C += A tương đương với C = C + A |
| -= | Trừ toán hạng bên trái với toán hạng bên phải và gán kết quả cho toán hạng bên trái | C -= A tương đương với C = C - A |
| \*= | Nhân toán hạng bên trái với toán hạng bên phải và gán kết quả cho toán hạng bên trái | C \*= A tương đương với C = C \* A |
| /= | Chia toán hạng bên trái với toán hạng bên phải và gán kết quả cho toán hạng bên trái | C /= A tương đương với C = C / A |
| %= | Modulo toán hạng bên trái với toán hạng bên phải và gán kết quả cho toán hạng bên trái | C %= A tương đương với C = C % A |
| ? : | Biểu thức điều kiện | Nếu điều kiện là true ? Gán giá trị là X : Ngược lại là Y |

### 1.3.11 Lớp và đối tượng

Lớp nghĩa là một khung kịch bản của một đối tượng, hoặc bạn có thể hiểu đối tượng được tạo ra từ một lớp. Trong lớp nó sẽ có các biến mà biến này ta gọi là các thuộc tính (properties), và lớp nó có thể chứa các hàm mà các hàm này chúng ta gọi nó là phương thức (method).

**Class**: là tập hợp các biến và hàm làm việc với các biến này. Một lớp có định dạng như sau:

<?php

Class Cart{

Var $items;

Function add\_item($a)

Lớp Cart ở đây là một kiểu dữ liệu

## 1.4 MySQL và PHP

Để connect tới 1 CSDL trên MySQL Server rất đơn giản. Chỉ cần sử dụng hàm:mysql\_connect (host, user, password) để mở 1 kết nối tới MSQL Server với kết quả là giá trị trả về của hàm (Giả sử là biến $dbname). Sau đó, ta sẽ dùng hàm mysql\_select\_db(dbname, $link)

Để thực hiện một câu lệnh truy vấn, dùng hàm mysql\_query(query, link). Giá trị trả lại của hàm là kết quả của câu truy vấn. Nếu bỏ qua link thì kết nối cuối cùng tới MySQL server sẽ được thực hiện.

Để lấy kết quả thực hiện, ta có thể dùng hàm mysql\_result(), mysql\_fetch\_ row(), mysql\_fetch\_array(), mysql\_fetch\_object().

**mysql\_fetch\_row():** Trả về một mảng các dữ liệu lấy từ 1 dòng trong CSDL. Nếu đã ở cuối CSDL, giá trị trả về là false. Bạn phải dùng chỉ số của các trường trong CSDL nếu muốn lấy dữ liệu.

**mysql\_fetch\_array(int result[, int result\_type]):** Trả về một mảng chứa dữ liệu lấy từ một dòng trong CSDL. Tương tự như mysql\_fetch\_row() nhưng bạn có thể lấy thêm trường để lấy dữ liệu.

**mysql\_fetch\_object:** Chuyển kết quả thành một đối tượng với các trường là các trường trong cơ sở dữ liệu.

### 1.4.1 Giới thiệu cơ sở dữ liệu

MySQL là ứng dụng cơ sở dữ liệu mã nguồn mở phổ biến nhất hiện nay (theo www.mysql.com) và được sử dụng phối hợp với PHP. Trước khi làm việc với MySQL cần xác định các nhu cầu cho ứng dụng.

MySQL là cơ sở dữ liệu có trình giao diện trên Windows hay Linux, cho phép người sử dụng có thể thao tác các hành động liên quan đến nó**.** Việc tìm hiểu từng công nghệ trước khi bắt tay vào việc viết mã kịch bản PHP, việc tích hợp hai công nghệ PHP và MySQL là một công việc cần thiết và rất quan trọng.

**Ưu điểm:**

* Giảm sự trùng lặp thông tin xuống mức thấp nhất. Do đó đảm bảo thông tin có tính nhất quán và toàn vẹn dữ liệu.
* Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau.
* Nhiều người có thể sử dụng một cơ sở dữ liệu.

**Nhược điểm:**

*Tính chủ quyền của dữ liệu:*

* Thể hiện ở phương diện an toàn dữ liệu.
* Khả năng biểu diễn mỗi liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu.
* Người khai thác cơ sở dữ liệu phải cập nhật cho CSDL những thông tin mới nhất.

*Tính bảo mật và quyền khai thác thông tin của người sử dụng:*

* Do ưu điểm CSDL có thể cho nhiều người khai thác đồng thời. nên cần phải có một cơ chế bảo mật phân quyền khai thác CSDL.
* Các hệ điều hành nhiều người sử dụng hay cục bộ đều cung cấp cơ chế này.

*Tranh chấp dữ liệu:*

* Khi nhiều người cùng truy nhập CSDL với các mục đích khác nhau. Rất có thể sẽ xảy ra hiện tượng tranh chấp dữ liệu.
* Cần có cơ chế ưu tiên khi truy cập CSDL. Ví dụ: Admin luôn có thể truy cập cơ sở dữ liệu.
* Cấp quyền ưu tiên cho từng người khai thác.

*Cần đảm bảo an toàn dữ liệu khi có sự cố:*

* Khi CSDL nhiều và được quản lý tập trung. Khả năng rủi ro mất dữ liệu rất cao. Các nguyên nhân chính là mất điện đột ngột hoặc hỏng thiết bị lưu trữ.
* Hiện tại có một số hệ điều hành đã có cơ chế tự động sao lưu ổ cứng và fix lỗi khi có sự cố xảy ra.
* Tuy nhiên: cẩn tắc vô áy náy. Chúng ta nên sao lưu dự phòng cho dữ liệu đề phòng trường hợp xấu xảy ra.

### 1.4.2 Mục đích sử dụng cơ sở dữ liệu

Mục đích sử dụng cơ sở dữ liệu bao gồm các chức năng như: lưu trữ (storage), truy cập (accessibility), tổ chức (organization) và xử lí (manipulation).

Lưu trữ: Lưu trữ trên đĩa và có thể chuyển đổi dữ liệu từ cơ sở dữ liệu này sang cơ sở dữ liệu khác, nếu bạn sử dụng cho quy mô nhỏ, bạn có thể chọn cơ sở dữ liệu nhỏ như: Microsoft Exel, Microsoft Access, MySQL, Microsoft Visual FoxPro… Nếu ứng dụng có quy mô lớn, bạn có thể chọn cơ sở dữ liệu có quy mô lớn như: Oracle, SQL Server…

Truy cập: Truy cập dữ liệu phụ thuộc vào mục đích và yêu cầu của người sử dụng, ở mức độ mang tính cục bộ, truy cập cơ sỏ dữ liệu ngay trong cơ sở dữ liệu với nhau, nhằm trao đổi hay xử lí dữ liệu ngay bên trong chính nó, nhưng do mục đích và yêu cầu người dùng vượt ra ngoài cơ sở dữ liệu, nên bạn cần có các phương thức truy cập dữ liệu giữa các cơ sở dử liệu với nhau như: Microsoft Access với SQL Server, hay SQL Server và cơ sở dữ liệu Oracle....

Tổ chức: Tổ chức cơ sở dữ liệu phụ thuộc vào mô hình cơ sở dữ liệu, phân tích và thiết kế cơ sở dữ liệu tức là tổ chức cơ sở dữ liệu phụ thuộc vào đặc điểm riêng của từng ứng dụng. Tuy nhiên khi tổ chức cơ sở dữ liệu cần phải tuân theo một số tiêu chuẩn của hệ thống cơ sở dữ liệu nnhằm tăng tính tối ưu khi truy cập và xử lí.

Xử lí: Tùy vào nhu cầu tính toán và truy vấn cơ sở dữ liệu với các mục đích khác nhau, cần phải sử dụng các phát biểu truy vấn cùng các phép toán, phát biểu của cơ sở dữ liệu để xuất ra kết quả như yêu cầu. Để thao tác hay xử lí dữ liệu bên trong chính cơ sở dữ liệu ta sử dụng các ngôn ngữ lập trình như: PHP, C++, Java, Visual Basic…

### 1.4.3 Các thao tác cập nhật dữ liệu

SELECT (Truy vấn mẫu tin): Select dùng để truy vấn từ một hay nhiều bảng khác nhau, kết quả trả về là một tập mẫu tin thỏa mãn các điều kiện cho trước nếu có, cú pháp của phát biểu SQL dạng SELECT như sau:

SELECT<danh sách các cột>

[FROM<danh sách bảng>]

[WHERE<các điều kiện ràng buộc>]

[GROUP BY<tên cột/biểu thức trong SELECT>]

[HAVING<điều kiện bắt buộc của GROUP BY>]

[ORDER BY<danh sách các cột>]

[LIMIT FromNumber áToNumber]

INSERT(Thêm mẫu tin):

Cú pháp: INSERT INTO Tên\_bảng VALUES(Bộ\_giá\_trị)

UPDATE(Cập nhật dữ liệu):

Cú pháp: UPDATE TABLE Tên\_bảng

SET Tên\_cột=Biểu\_thức,...

[WHERE Điều\_kiện]

DELETE(Xóa mẫu tin):

Cú pháp: DELETE FROM Tên\_bảng

[WHERE Điều\_kiện]

### 1.4.4 Các hàm thông dụng trong MySQL

#### **1.4.4.1 Các hàm trong phát biểu GROUP BY**

* Hàm AVG: Hàm trả về giá trị bình quân của cột hay trường trong câu truy vấn
* Hàm MIN: Hàm trả về giá trị nhỏ nhất của cột hay trường trong câu truy vấn
* Hàm MAX: Hàm trả về giá trị lớn nhất của cột hay trường trong câu truy vấn
* Hàm Count: Hàm trả về số lượng mẩu tin trong câu truy vấn
* Hàm Sum: Hàm trả về tổng các giá trị của trường, cột trong câu truy vấn.

#### **1.4.4.2 Các hàm xử lí chuỗi**

* Hàm ASCII: Hàm trả về giá trị mã ASCII của kí tự bên trái của chuỗi.
* Hàm Char: Hàm này chuyển đổi kkiểu mã ASCII từ số nguyên sang dạng chuỗi.
* Hàm UPPER: Hàm này chuyển đổi chuỗi sang kiểu chữ hoa.
* Hàm LOWER: Hàm này chuyển đổi chuỗi sang kiểu chữ thường.
* Hàm Len: Hàm này trả về chiều dài của chuỗi.
* Thủ tục LTRIM: Thủ tục loại bỏ khỏang trắng bên trái của chuỗi.
* Thủ tục RTRIM: Thủ tục loại bỏ khỏang trắng bên phải của chuỗi.
* Hàm Left(str, n): Hàm trả về chuỗi bên trái tính từ đầu cho đến vị trí n.
* Hàm Right(str, n): Hàm trả về chuỗi bên phải tính từ đầu cho đến vị trí n.

- Hàm Instrt: Hàm trả về chuỗi vị trí bắt đầu của chuỗi con trong chuỗi xét.

#### **1.4.4.3 Các hàm xử lí về thời gian**

* Hàm CurDate(): Hàm trả về ngày, tháng và năm hiện hành của hệ thống.
* Hàm CurTime(): Hàm trả về giờ, phút và giây hiện hành của hệ thống.
* Hàm Period\_Diff: Hàm trả về số ngày trong khoảng thời gian giữa 2 ngày.
* Hàm dayofmonth: Hàm trả về ngày thứ mấy trong tháng.

#### **1.4.4.4 Các hàm về toán học**

* Hàm sqrt: Hàm trả về là căn bậc hai của một biểu thức.
* Hàm CurDate(): Hàm trả về ngày, tháng và năm hiện hành của hệ thống.

## 1.5 Sơ lược về Framework Laravel

### 1.5.1 Tổng quan về Laravel

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

Laravel phiên bản đầu tiên được phát hành vào tháng 6 năm 2011 bởi Taylor Otwell.

Vào khoảng Tháng 3 năm 2015, các lập trình viên đã có một cuộc bình chọn PHP framework phổ biến nhất, Laravel đã giành vị trí quán quân cho PHP framework phổ biến nhất năm 2015, theo sau lần lượt là Symfony2, Nette, CodeIgniter, Yii2 vào một số khác. Trước đó, Tháng 8 2014, Laravel đã trở thành project PHP phổ biến nhất và được theo dõi nhiều nhất trên Github.

Laravel được phát hành theo giấy phép MIT, với source code được lưu trữ tại Github.

### 1.5.2 Cấu trúc thư mục ứng dụng Laravel

|  |  |
| --- | --- |
| /app | Chứa controller, model, views và assets của ứng dụng,  là nơi chứa code chính của bạn. |
| /public | Lưu trữ file css, javascrip, images, và những file. Đồng  thời nó chứa file khởi động index.php. |
| /vendor | Là nơi chứa toàn bộ code của bên thứ ba. Chứ plugin  chúng ta cài thêm cho ứng dụng. |

|  |  |
| --- | --- |
| /app/config/ | Nơi chứa các file cấu hình khi chạy ứng dụng, csdl,  session và nhiều thứ khác. |
| /app/config/app.php | Cấu hình mức cài đặt cảu ứng dụng ở khía cạnh khác  như timezone, locale, mode debug và khóa mã hóa duy nhất. |
| /app/config/auth.php | Drive xác thực. |
| /app/config/cache.php | Nếu ứng dụng có lưu cache thì thời gian hồi đáp ứng  dụng nhanh hơn. |
| /app/database/migrations/ | Thư mục migration chứa những lớp PHP mà cho phép Laravel cập nhật Schema cơ sở dữ liệu mà vẫn giữ được toàn bộ versions cơ sở dữ liệu trong khi đồng bộ.  Những file migration được tạo bởi tool Artisan. |
| /app/database/seeds/ | Chứa những file PHP cho phép Artisan đưa vào bảng  csdl với những dữ liệu được đề xuất. |
| /app/models/ | Chứa những file models của ứng dụng |
| /app/views/ | Thư mục chứa những file HTML được sử dụng bởi  controller hay route. |
| /app/lang/ | Mặc định thư mục này chứa những dòng ngôn ngữ cho  việc phân trang và chứng thực form người dùng với ngôn ngữ là tiếng anh. |
| /app/start/ | Chứa những thiết lập tùy chỉnh liên quan đến tool  Artisan cũng như context local và global. |
| /app/storage | Thư mục storage được sử dụng để lưu trữ file tạm thời cho những dịch vụ Laravel khác nhau như session, cache, biên dịch template views. Thư mục này có thể ghi lại bởi web server. Thư mục này được duy trì bởi  Laravel mà mình không cần can thiệp vào. |
| /app/routes.php | Đây là file route trong ứng dụng của bạn, nó lưu trứ  toàn bộ những route để thông báo với Laravel cách như thế nào để kết nối khi có yêu cầu, theo hàm khai sẵn, |
|  | controller và action. Đây là file cũng chứa cách khai báo cho một vài sự kiện bao gồm trang lỗi, và nó được  sử dụng để định nghĩa views composers. |
| /app/filter.php | File này dùng để giới hạn một số khu vực không được  truy cập của trang web. |

# CHƯƠNG 2: KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

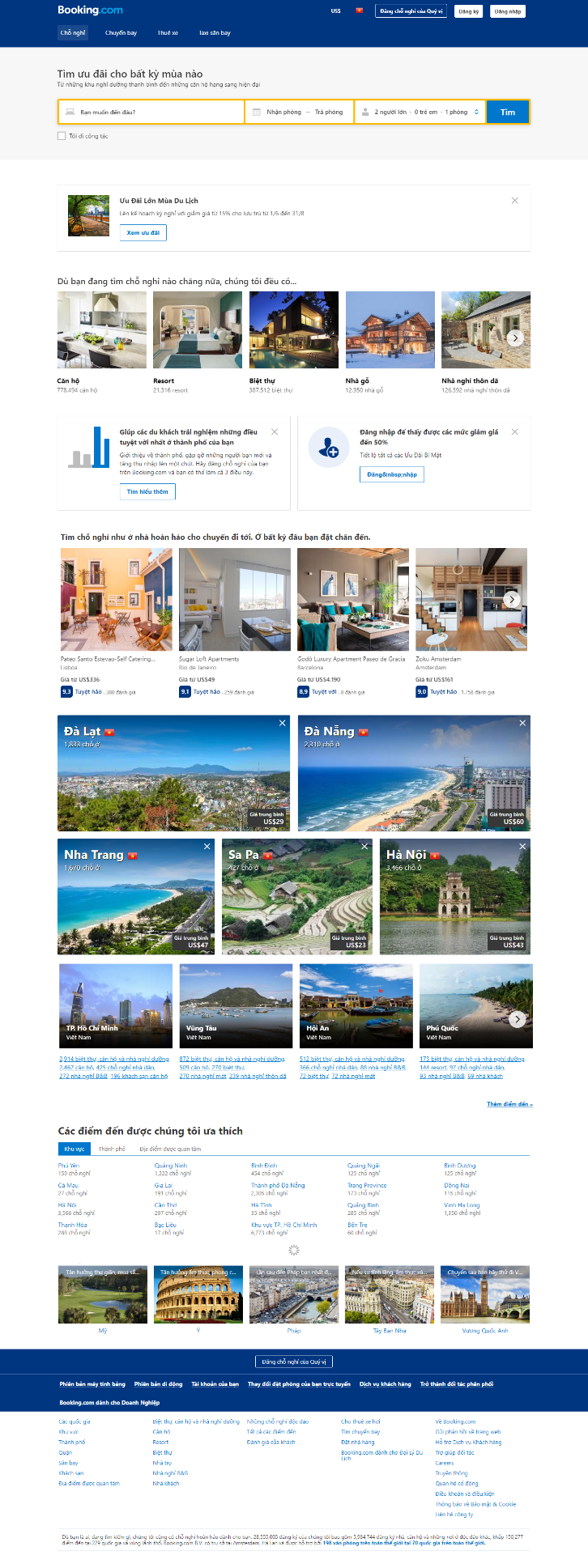
## 2.1. Phát biểu bài toán

Homestay là một loại hình “du lịch xanh” lý tưởng đối với các bạn trẻ yêu thích khám phá văn hóa tại các vùng đất mới. Khi đi du lịch homestay, thay vì ở khách sạn hoặc nhà nghỉ các bạn sẽ ở ngay tại nhà của dân địa phương để có thể có một góc nhìn gần gũi và thực tế hơn với cách sống và nền văn hóa của chính họ. Khách được xem như một thành viên của gia đình và tham gia vào các sinh hoạt đời thường như ăn cơm chung mâm và trò chuyện trao đổi với các thành viên. Khách cũng được yêu cầu phải “nhập gia tùy tục” và biết cách tôn trọng các quy tắc và sự riêng tư nhất định của gia chủ.

Hiện nay kinh doanh Homestay đã trở thành 1 xu thế cà trong nước và nước ngoài và đang trên đà phát triển mạnh. Những người đi du lịch thường có thói quen tìm kiếm thông tin trên mạng internet. Khi có ý tưởng về việc đi du lịch, họ sẽ lên google và tìm kiếm địa điểm, nơi ăn ở, review, book phòng… Trong hoàn cảnh đó, các website homestay nghiễm nhiên trở thành “mỏ vàng” cần được khai thác. Bài toán trên đặt ra là làm sao có thể xây dựng một website cho phép khách hàng (kể cả trong và ngoài nước) có thể tìm kiếm thông tin của các homestay trong nước với các nội dung cụ thể như: Tên homestay, địa chỉ, số phòng, giá phòng… Ngoài ra, nếu tìm được homestay mong muốn họ có thể đặt phòng trực tiếp qua mạng (thanh toán bằng các hình thức trả trước và trả sau).

## 2.2. Khảo sát một số website

Trong quá trình thu thập thông tin, em đã khảo sát một số website đặt phòng online khác như: Booking.com, Luxstay.com để có thể học hỏi kinh nghiệm tạo ra website của riêng mình.



Giao diện trang chủ của Booking.com 1



Giao diện trang chủ của Luxstay.com 1

Qua quá trình khảo sát em rút ra được 1 số đặc điểm của 2 website trên là:  
- Website có tốc độ tải trang nhanh, tương thích với mọi trình duyệt và mọi thiết bị.

* Website được tối ưu, Seo onpage tốt.
* Website có giao diện đơn giản mà đầy đủ, đẹp mà không quá cầu kỳ.
* Website có bố cục phần header là chức năng đăng ký, đăng nhập. Tiếp đến ngay sau đó là phần Tìm Kiếm cho khách hàng tìm phòng, rất phù hợp với nhu cầu của người dùng.

## 2.3. Phân tích yêu cầu hệ thống

### 2.3.1. Các đối tượng phục vụ của hệ thống

* Các homestay

Các homestay sẽ đăng ký với website để trở thành một nhà cung cấp dịch vụ cho thuê phòng và các dịch vụ khác của homestay. Họ có nhiệm vụ cung cấp các thông tin về homestay, về loại phòng, tiện nghi phòng, tổng số phòng, cung cách phục vụ, giá cả, nội quy, các dịch vụ bổ sung… Những thông tin này sẽ được cập nhật vào những thời điểm thích hợp để đảm bảo thông tin chính xác.

* Khách hàng, những người có nhu cầu tìm homestay

Đối tượng này có thể: đăng ký với website để trở thành thành viên (thông tin được lưu trữ và họ có thể đăng nhập), tìm kiếm các homestay thích hợp, đặt phòng trực tuyến qua mạng.

* Người quản trị của website

Nhà quản trị có thể đăng nhập để quản trị website ở bất cứ nơi đâu. Họ có quyền kiểm soát tất cả các hoạt động của website, cụ thể như: xem các danh sách đăng ký (nhà cung cấp dịch vụ, thành viên), thêm, xóa, cập nhật thông tin…

### 2.3.2. Các yêu cầu chức năng

2.3.2.1. Yêu cầu chức năng nghiệp vụ

* + - * + Cho phép người dùng đăng ký với hệ thống với vai trò homestay, khách hàng.
        + Lưu trữ thông tin của tất cả các đối tượng.
        + Lưu trữ thông tin đặt phòng của khách hàng.
        + Cho phép khách hàng đặt phòng homestay qua web.
        + Cho phép khách hàng tìm kiếm thông tin homestay theo các tiêu chí khác nhau.
        + Cho phép các homestay nhập thông tin mô tả về dịch vụ của mình và có thể hiệu chỉnh bất cứ lúc nào.
        + Cho phép các loại người dùng khác nhau hiệu chỉnh thông tin cá nhân khi cần thiết.
        + Hỗ trợ cho người quản trị trong việc quản lý hệ thống bao gồm: quản lý thông tin của tất cả các đối tượng (khách hàng, homestay, người dùng quyền quản trị), được quyền thêm và xóa các đối tượng này; quản lý các thông tin đặt phòng, và các thông tin liên quan.
        + Hỗ trợ việc trao đổi thông tin lẫn nhau của các đối tượng.

2.3.2.2. Yêu cầu chức năng hệ thống:

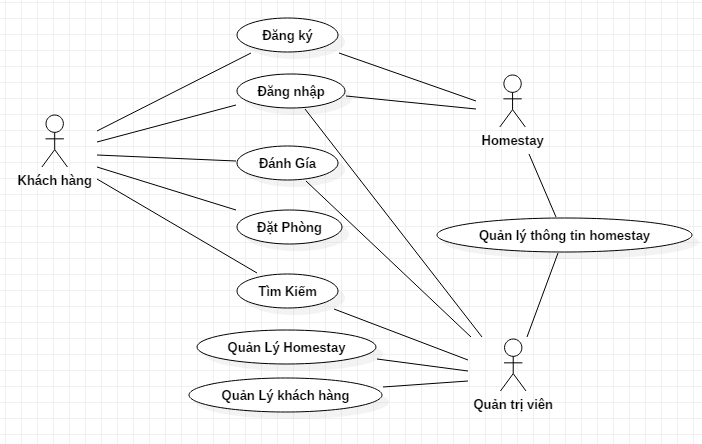
* + - * + Cho phép người dùng đăng nhập vào hệ thống với các vai trò khác nhau ứng với từng đối tượng.
        + Đảm bảo tính bảo mật của các thông tin đăng nhập của người dùng ở xa.
        + Đảm bảo tính bảo mật của các thông tin đăng ký dịch vụ của khách hàng (như các mã số của thẻ tín dụng…).
        + Phân cấp quyền hạn của từng đối tượng người dùng trong việc truy cập vào các tài nguyên của hệ thống.

2.3.3. Các yêu cầu phi chức năng

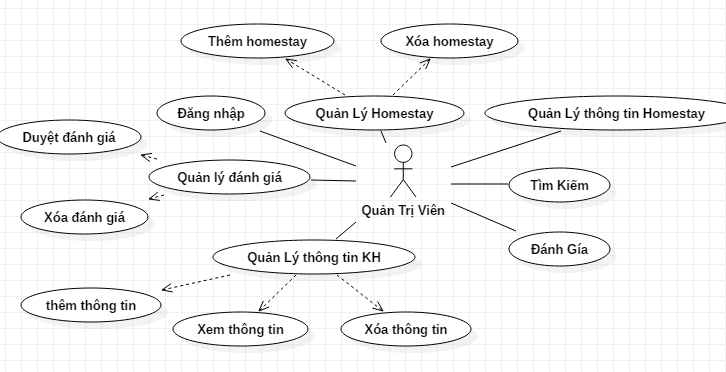
2.3.3.1. Đối với các đối tượng người dùng

* + - * + Tính tiện dụng: Giao diện thân thiện, dễ sử dụng.
        + Tính hiệu quả: Đảm bảo việc truy xuất nhanh đến các trang và khả năng kiểm soát lỗi tốt.
        + Tính tương thích: hỗ trợ và sử dụng tốt trên các trình duyệt thông dụng hiện nay như Internet Explorer, Google Chrome, Mozila Firefox…
        + Tính tiến hóa.

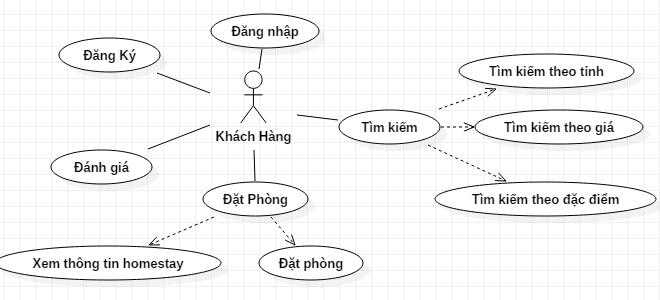
## 2.4. Các sơ đồ Use Case:



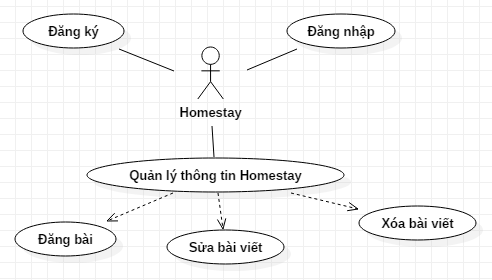
Biểu đồ usecase tổng quát



Biểu đồ usecase tác nhân Quản trị viên



Biểu đồ usecase tác nhân Khách Hàng



Biểu đồ usecase tác nhân Homestay

## 2.5. Đặc tả chi tiết các use-case ( UC )

* UC Đăng nhập
* Mục đích: Đảm bảo xác thực thông tin người sử dụng và an toàn bảo mật hệ thống.
* Tác nhân: Khách hàng, Quản trị viên, homestay.
* Mô tả chung: Use case này mô tả cách người dùng đăng nhập vào hệ thống với các vai trò khác nhau (Người quản trị website, Khách sạn, Hãng hàng không, Khách viếng).
* Luồng sự kiện chíh:

|  |  |  |
| --- | --- | --- |
| **STT** | **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1 | Chọn chức năng đăng nhập |  |
| 2 |  | Hiển thị form đăng nhập |
| 3 | Nhập thông tin đăng nhập |  |
| 4 | Gửi thông tin đã đăng nhập đến hệ thống. |  |
| 5 |  | Kiểm tra thông tin đăng nhập, nếu đúng thì cho phép truy cập hệ thống, nếu sai thì thông báo yêu cầu nhập lại. |

Luồng thay thế: Nếu người dùng nhập sai thông tin thì yêu cầu nhập lại .

- Yêu cầu cụ thể: Tên đăng nhập và mật khẩu phải khớp với thông tin đã đăng ký.

- Điều kiện trước: Người dùng đã đăng ký thành viên trong hệ thống.

- Điều kiện sau: Thông báo đăng nhập thành công/thất bại.

- Thông tin đầu vào: tên truy cập và mật khẩu

- Thông tin đầu ra: Những thông tin mà người sử dụng yêu cầu.

* UC Quản lý khách hàng
* Tác nhân: Quản trị viên
* Mô tả: Use case này cho phép người quản trị website có thể vào xem danh sách của các khách viếng, xem thông tin chi tiết của một khách viếng nào đó và xóa một khách viếng khỏi cơ sở dữ liệu.
* Luồng sự kiện chính:

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Đăng nhập vào hệ thống. |  |
| 2 | Chọn chức năng Quản lý khách hàng | Trả về danh sách các khách hàng |
| 3 | Chọn 1 khách hàng |  |
| 4 | Chọn chức năng “xem thông tin” hoặc “ xóa” | Truy xuất cơ sở dữ liệu hoặc xóa thông tin của KH |
| 5 |  |  |

- Điều kiện tiên quyết: Quản trị viên phải đăng nhập thành công vào hệ thống.

-Luồng sự kiện khác: Nếu usecase thành công, hệ thống sẽ trả về kết quả ở một cửa sổ mới.

* UseCase: Quản lý thông tin Homestay
* Tác nhân: Quản trị viên, Homestay
* Mô tả: Use case này cho phép những người dùng là phía Homestay (hoặc Quản trị viên) có thể quản lý các thông tin chung của mình như: xem thông tin chung, cập nhật các thông tin chung…
* Luồng sự kiện chính:

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Đăng nhập vào hệ thống với vai trò Homestay. |  |
| 2 | Chọn chức năng Quản Lý Khách Sạn | Truy xuất và hiển thị dữ liệu |
| 3 | Chọn chức năng Cập Nhật | Hiển thị ô nhập dữ liệu cho người dùng. |
| 4 | Người dùng điền thông tin và chọn Cập nhật | Kiểm tra tính hợp lệ của thông tin và thực hiện cập nhật |

* Luồng sự kiện khác: Nếu trong quá trình cập nhật thông tin, người dùng quyết định hủy bỏ quá trình cập nhật thì hệ thống sẽ trở lại trạng thái ban đầu và hiển thị lại trang thông tin khách sạn.
* Điều kiện tiên quyết: Người dùng phải đăng nhập vào hệ thống với vai trò Khách sạn để có thể thực hiện được các chức năng trong use case này.
* Nếu usecase thành công thì cơ sở dữ liệu của hệ thống sẽ được cập nhật, ngược lại thì sẽ không có gì thay đổi.
* UseCase: Đăng ký
* Tác nhân: Khách hàng, homestay
* Use case này cho phép người sử dụng đăng ký với trang web để trở thành thành viên với vai trò Khách hàng. Với tài khoản này, sau đó người dùng có thể đăng nhập vào hệ thống (với vai trò khách hàng) và có thể đặt phòng homestay.
* Luồng sự kiện chính

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Chọn chức năng Đăng ký | Hiển thị form đăng ký |
| 2 | Điền thông tin vào form và chọn Đăng ký | Kiểm tra tính hợp lệ của thông tin |
| 3 |  | Lưu vào CSDL. |

* Luồng sự kiện khác: Nếu trong quá trình đăng ký người dùng quyết định hủy bỏ các thông tin đăng ký thì use case sẽ kết thúc và trạng thái của hệ thống không thay đổi.
* Nếu quá trình kiểm tra thông tin nhập phát hiện không hợp lệ thì hệ thống sẽ trở về trạng thái trước đó (mở trang nhập liệu với các thông tin nhập trước đó vẫn còn để người dùng có thể chỉnh sửa).
* Nếu use case này thành công thì thông tin đăng ký của một khách viếng sẽ được lưu vào cơ sở dữ liệu của hệ thống. Ngược lại, trạng thái của hệ thống không thay đổi.
* Usecase Đặt phòng
* Tác nhân: Khách hàng
* Mô tả: Use case này cho phép người dùng có thể đăng ký phòng khách sạn trực tiếp trên mạng.
* Luồng sự kiện chính:

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Chọn chức năng xem thông tin phòng | Hiển thị cửa sổ mới về thông tin homestay |
| 2 | Chọn chức năng Đặt phòng | Hiển thị webform cho người dùng điền thông tin đặt phòng |
| 3 | Điền các thông tin vào form | Xác thực tính hợp lệ của thông tin |
| 4 |  | Lưu trữ thông tin vào CSDL |
|  |  | Hiển thị thời gian chờ thông báo kết quả cho KH. |
|  |  | Nếu quá trình xác nhận thông tin với Homestay thành công thì hệ thống sẽ hiển thị thông báo và đề nghị thanh toán. |
|  |  | Nếu quá trình xác nhận thông tin với Homestay thành công thì hệ thống sẽ hiển thị thông báo và đề nghị thực hiện lại. |

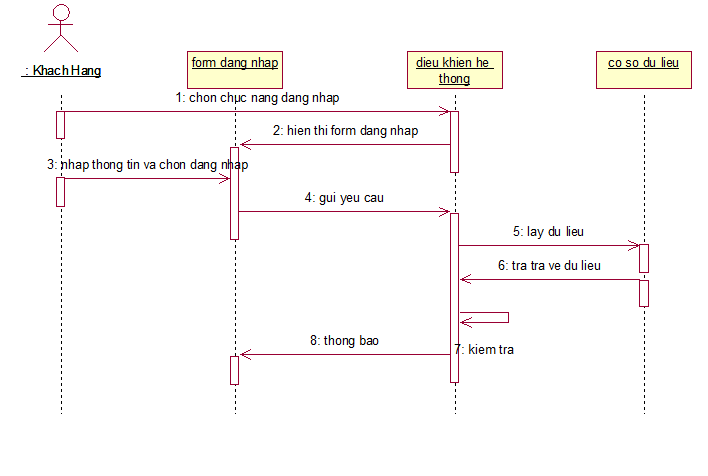
* *Luồng sự kiện khác:* Người dùng hủy bỏ quá trình đăng ký: Trạng thái của hệ thống không thay đổi và use case kết thúc.
* *Yêu cầu đặc biệt:* Người dùng phải hoặc đã có đăng ký thành viên với trang web với vai trò Khách Hàng, hoặc phải thực hiện việc đăng ký thành viên (để nhập các thông tin cá nhân) trước khi có thể thực hiện use case này.
* *Điều kiện tiên quyết: Người dùng phải đăng nhập với vai trò Khách hàng để thực hiện UseCase này.*
* *Usecase Tìm Kiếm*
* *Tác nhân: Khách hàng, quản trị viên*
* Mô tả: Use case này cho phép người dùng hoặc quản trị viên tìm kiếm các homestay thỏa mãn một nhu cầu nào đó.
* Luồng sự kiện chính:

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Chọn chức năng tìm kiếm | Hiển thị cho form cho phép người dùng điền thông tin |
| 2 | Nhập vào từ khóa tìm kiếm | Tìm kiểm trong CSDL  Hiển thị kết quả tìm kiếm |
| 3 | Người dùng có thể chọn xem thông tin chi tiết của Homestay | Hiển thị tùy vào lựa chọn của Khách hàng |

* *Luồng sự kiện khác:* Quá trình tìm kiếm thất bại: hệ thống yêu cầu người dùng nhập lại từ khóa hay hủy bỏ thao tác tìm kiếm.

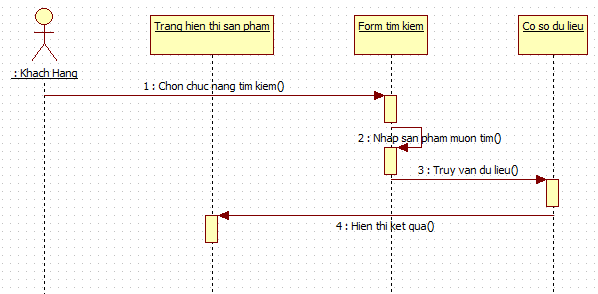
## 2.6. Biểu đồ trình tự

* *UC đăng nhập*



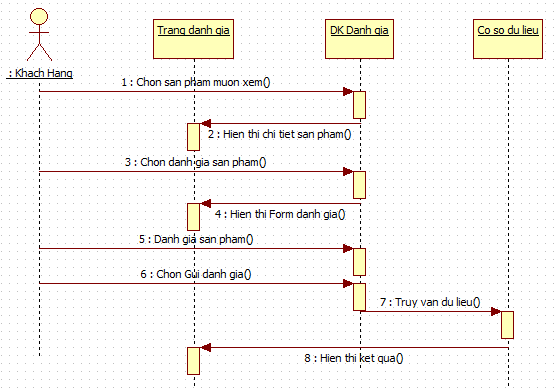
Biểu đồ trình tự UC đăng nhập

* UC tìm kiếm

**

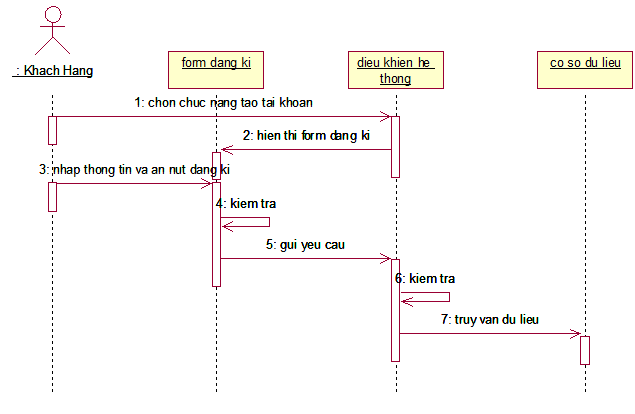
Biểu đồ trình tự UC tìm kiếm

* UC đánh giá sản phẩm

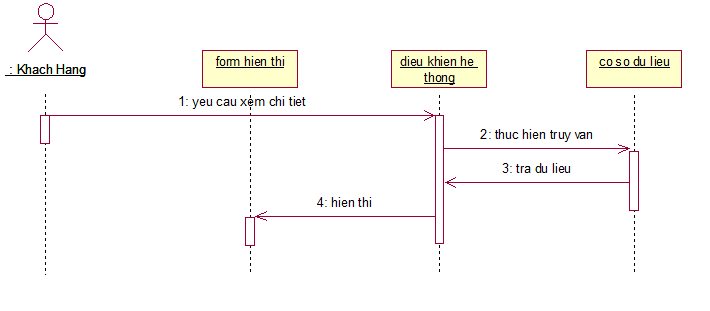
**

Biểu đồ trình tự UC đánh giá sản phẩm

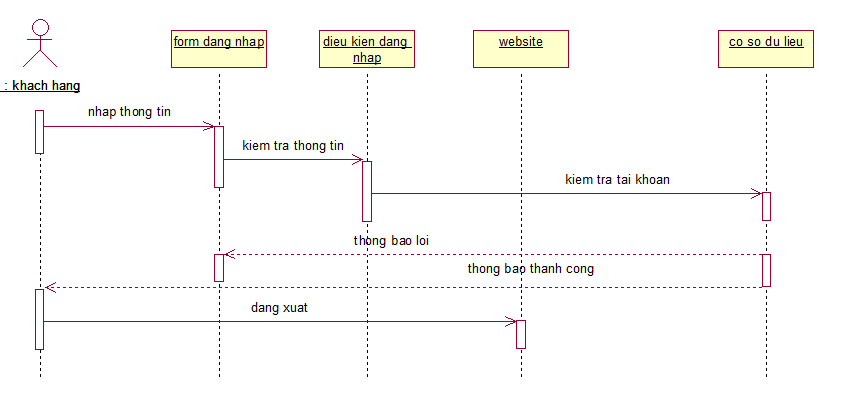
* *UC đăng ký*



Biểu đồ trình tự cho UC đăng ký 1



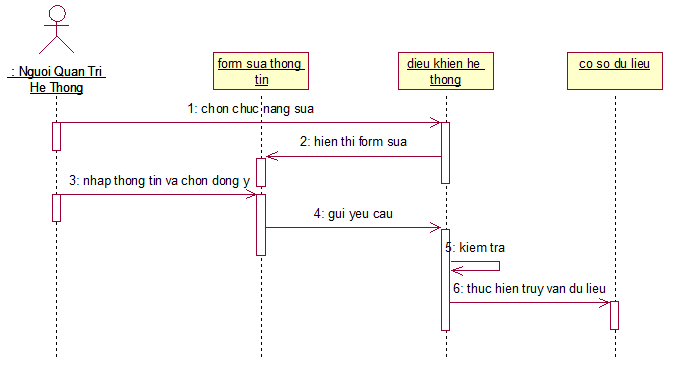
Biểu đồ trình tự cho chức năng xem phòng



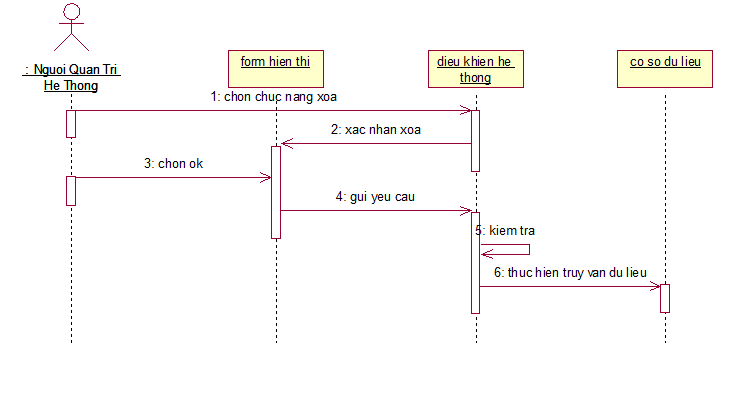
Biểu đồ trình tự UC đăng xuất



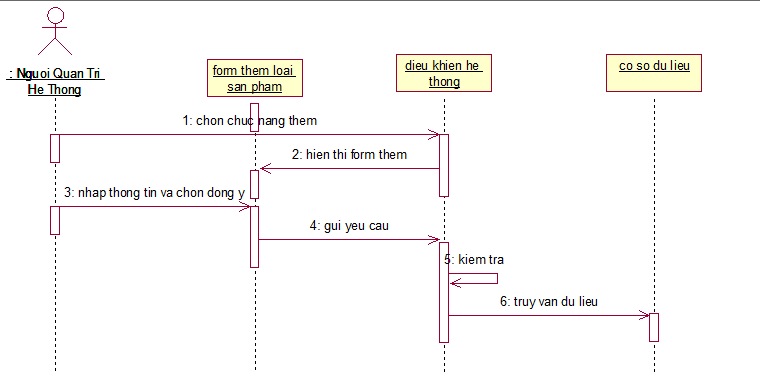
Biểu đồ trình tự UC quản lý khách hàng 1



Biểu đồ trình tự cho chức năng Sửa

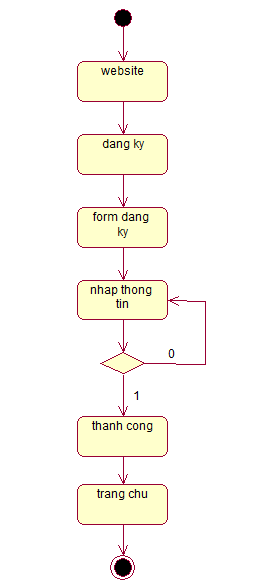


Biểu đồ trình tự cho chức năng Xóa



Biểu đồ trình tự cho chức năng thêm

## 2.7. Biểu đồ Hoạt động



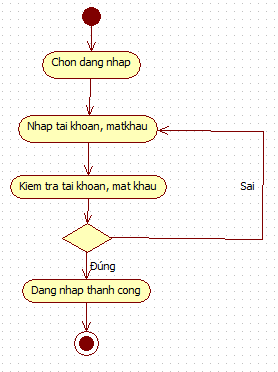
Biểu đồ hoạt động Đăng ký



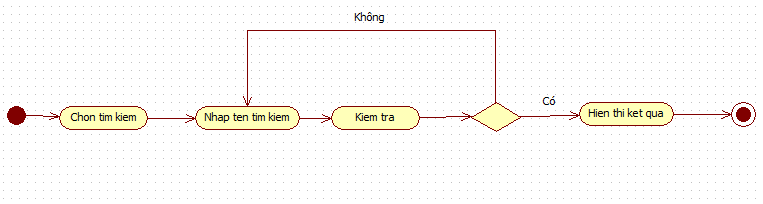
Biểu đồ hoạt động UC Đánh giá



Biểu đồ hoạt động UC quản lý đặt phòng

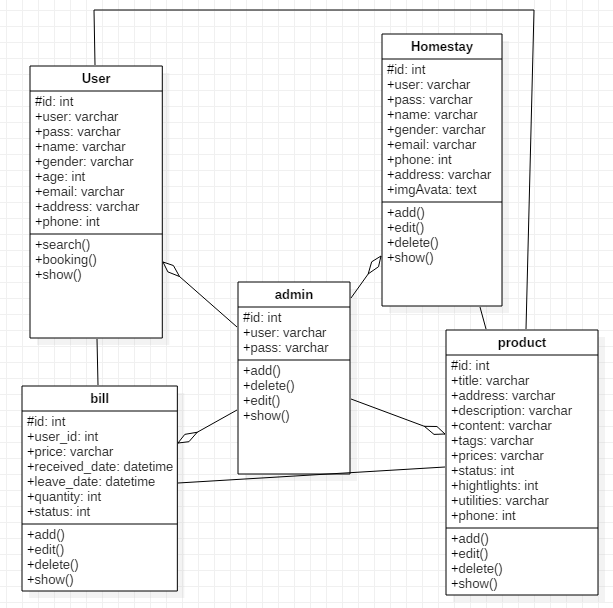
**

Biểu đồ hoạt động UC đăng nhập

**

Biểu đồ hoạt động UC tìm kiếm

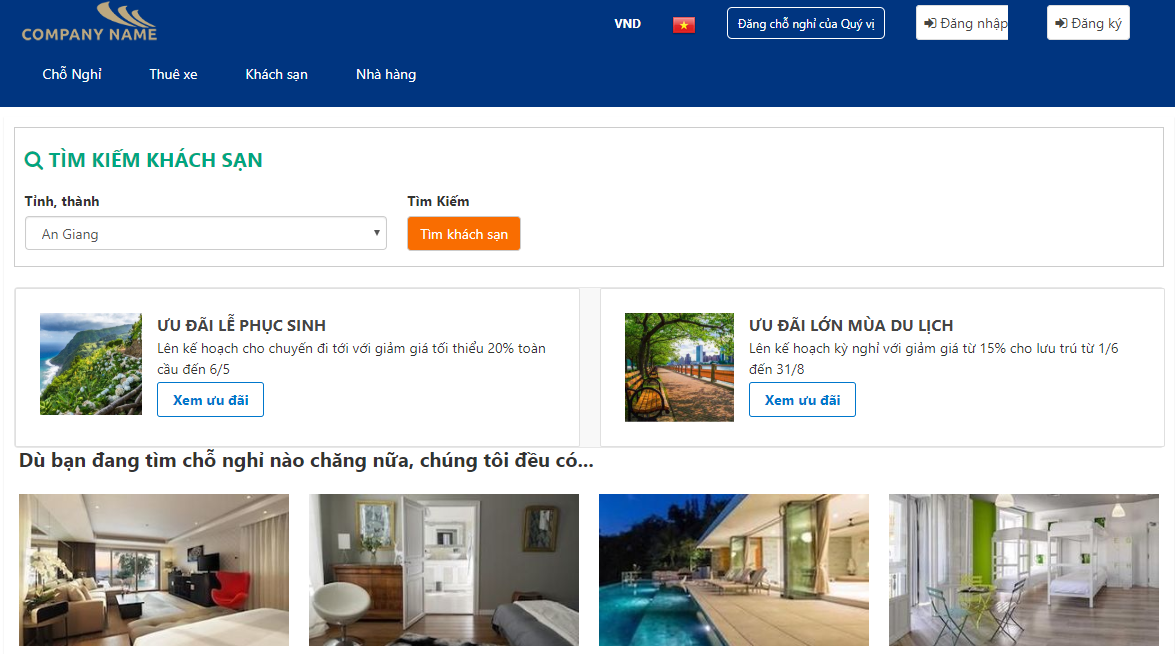
## 2.8. Biểu đồ lớp



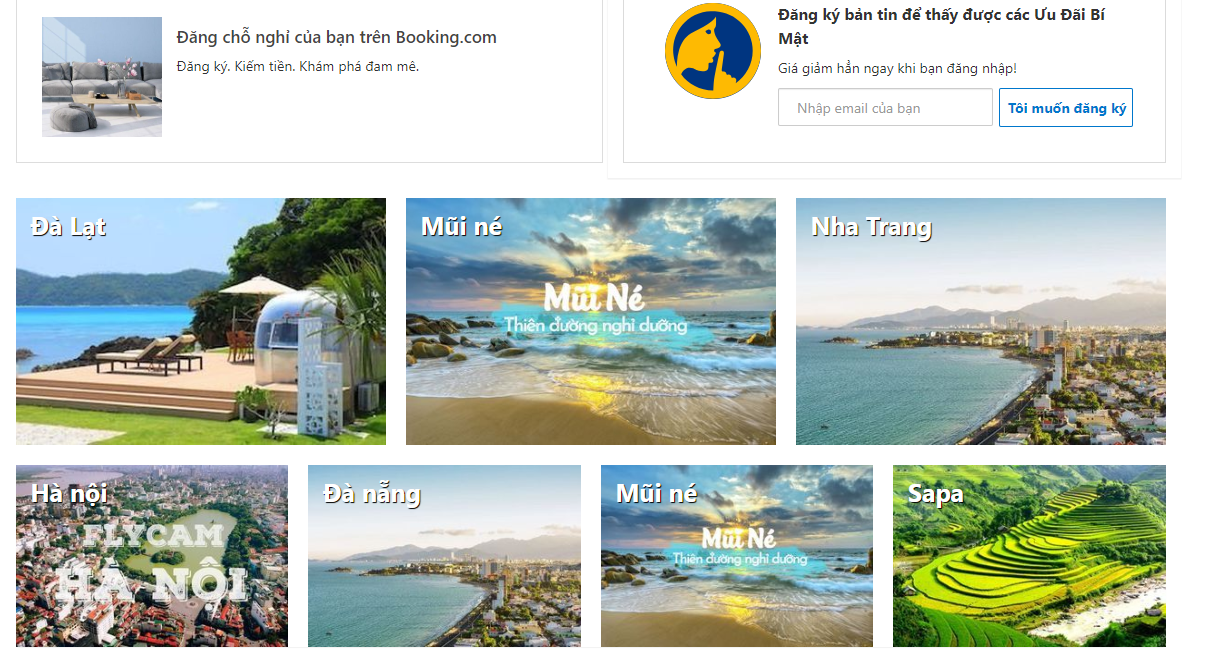
Biểu đồ lớp

# Chương 3: Xây dựng website

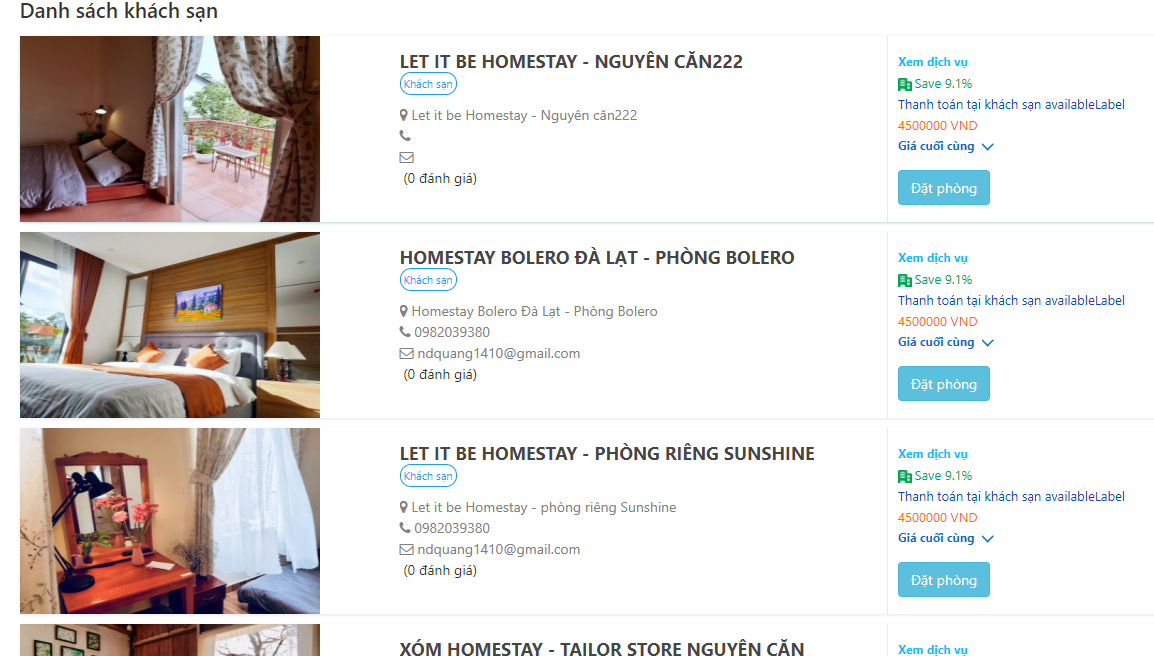
Giao diện trang chủ của website



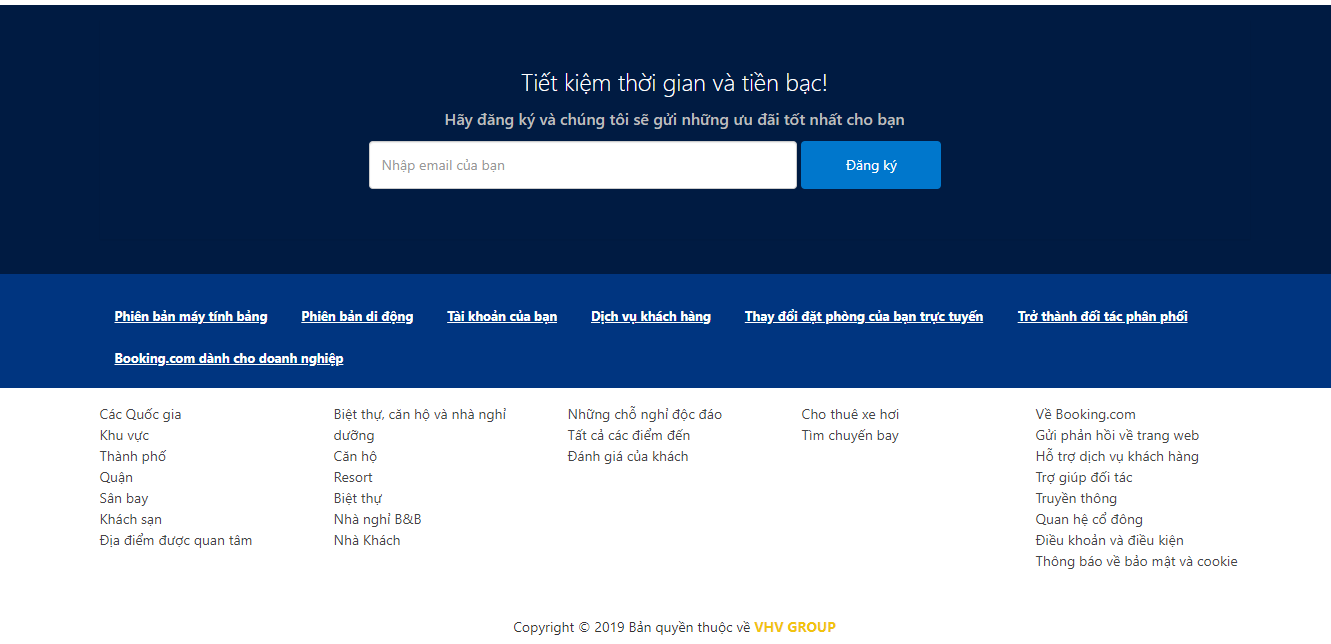
Giao diện trang chủ 1 1



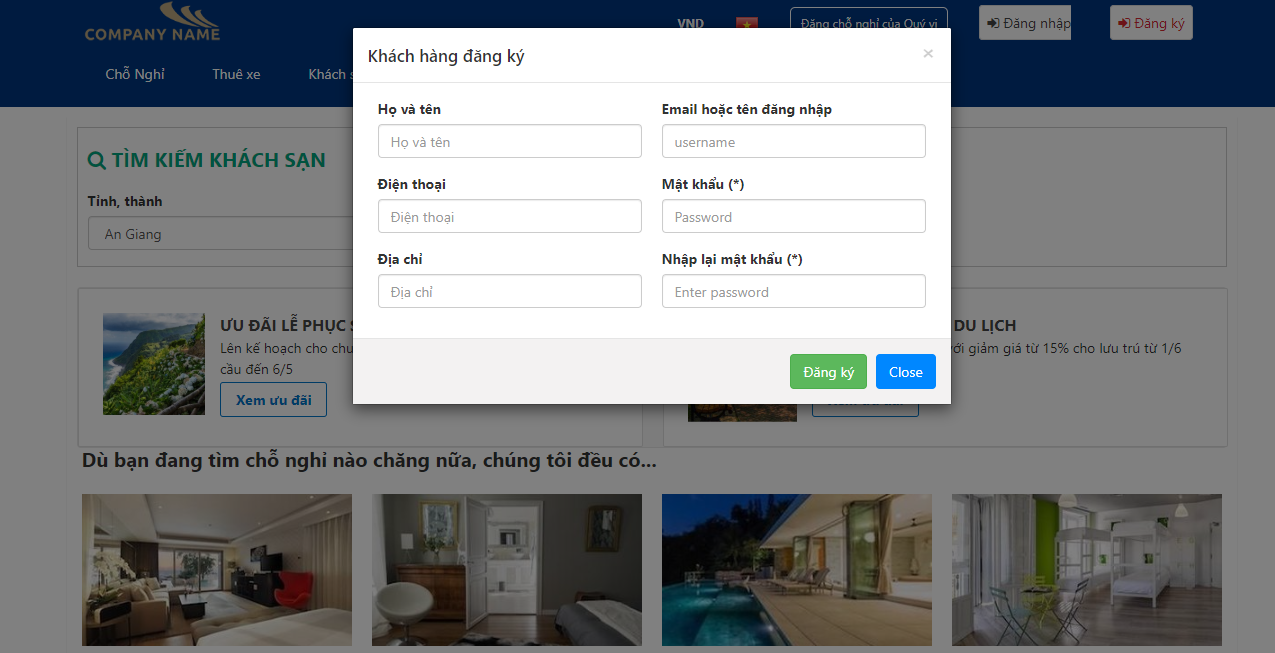
Giao diện trang chủ 1



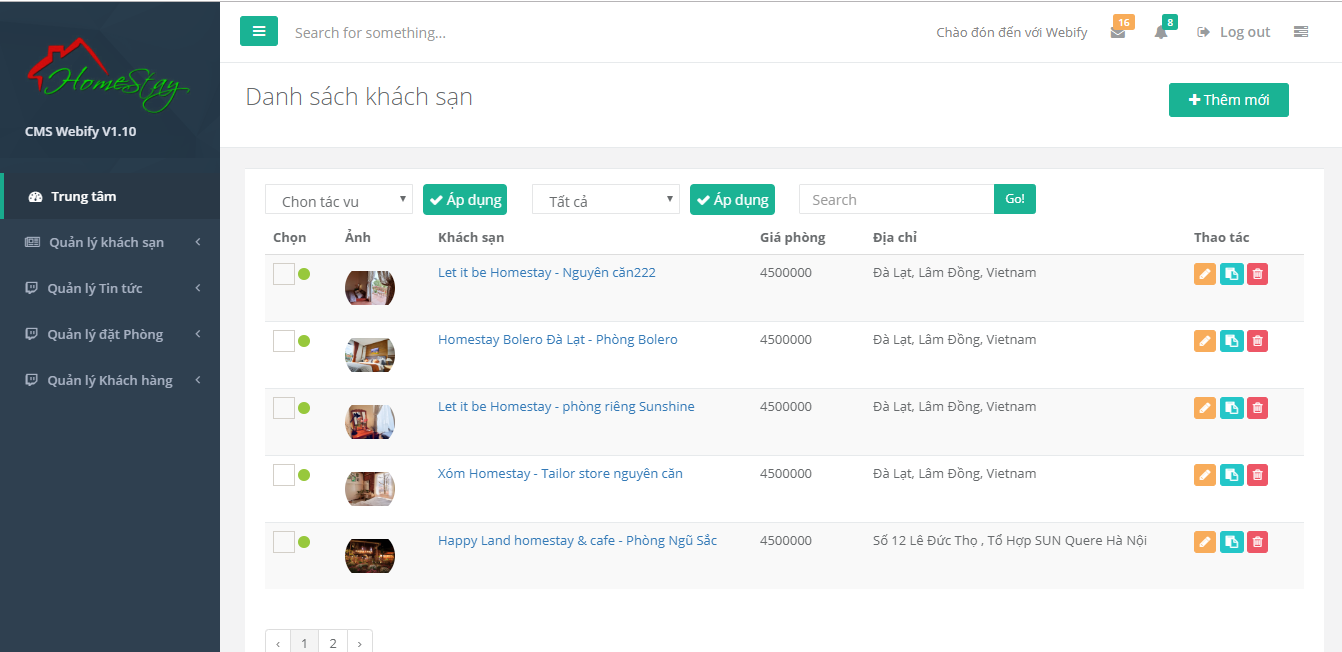
Giao diện trang chủ 3 1



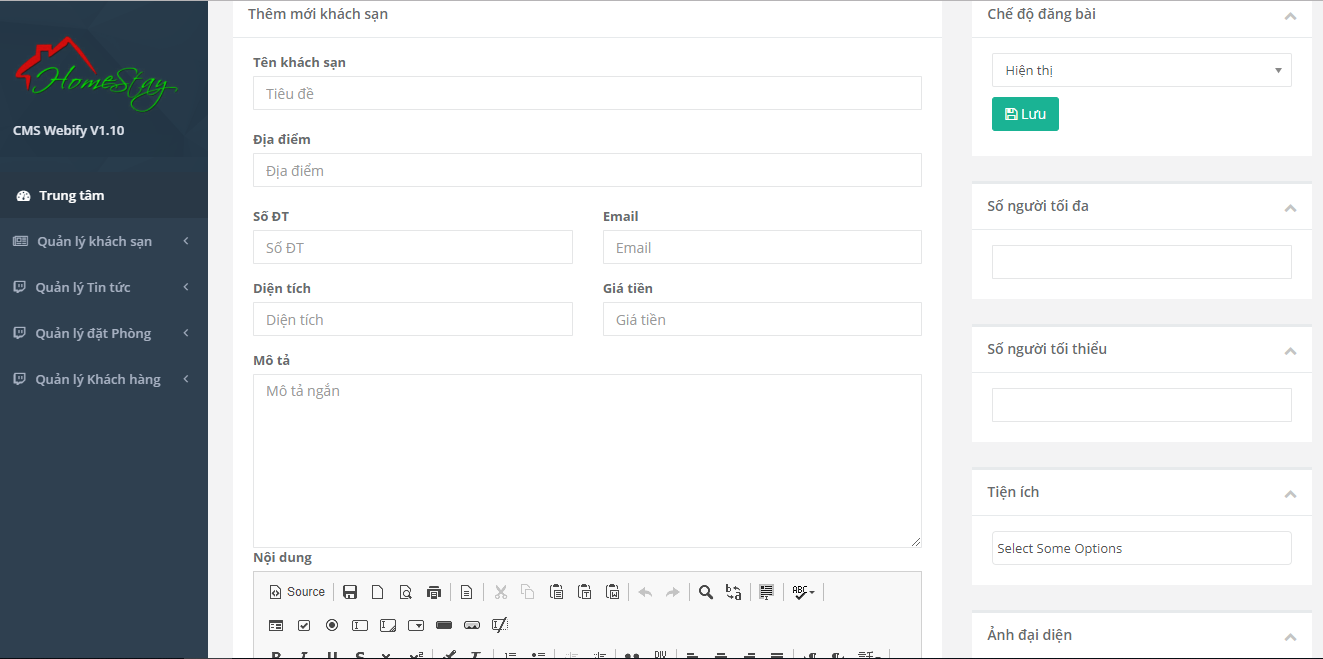
Giao diện Footer



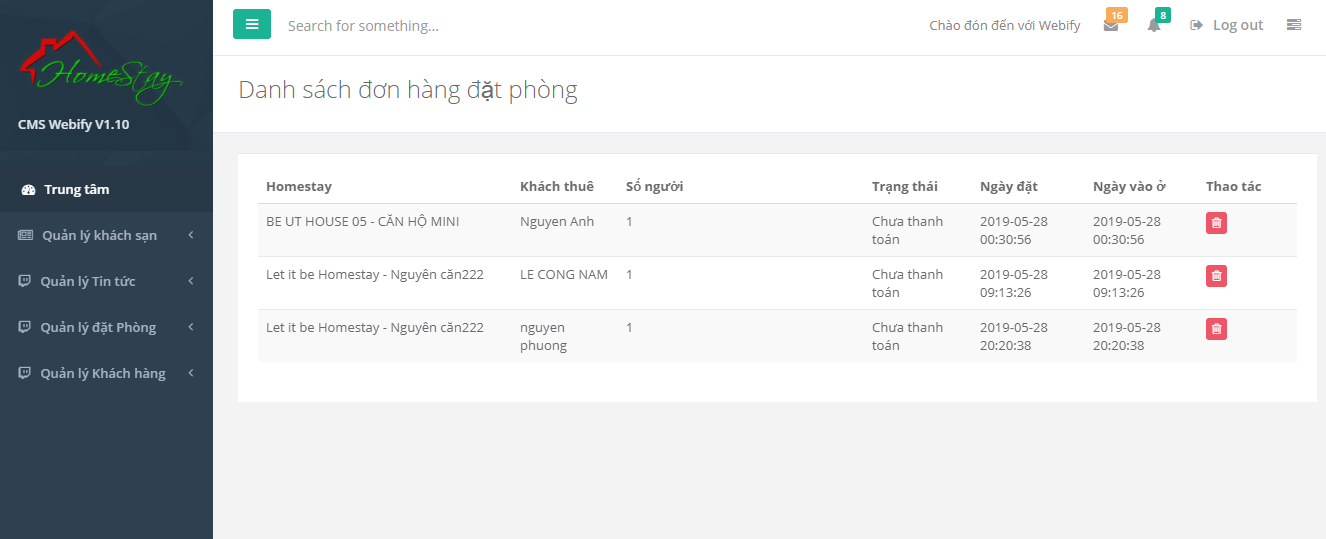
Giao diện chức năng đăng ký



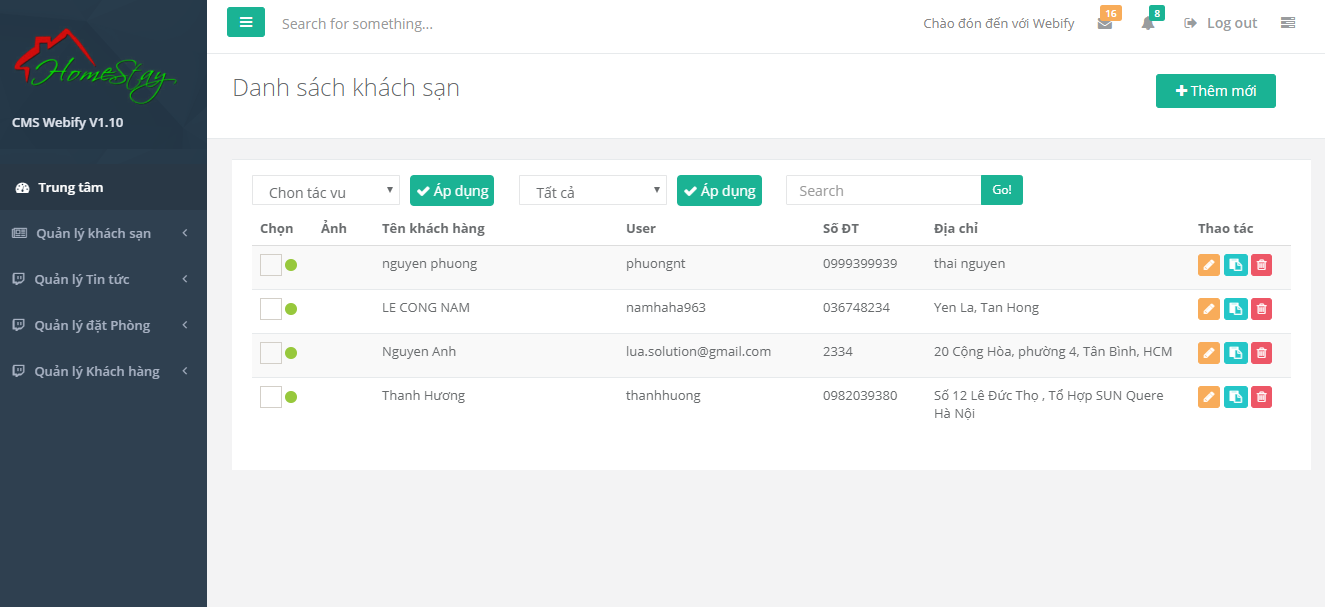
Giao diện trang admin



Giao diện trang thêm thông tin Homestay



Giao diện quảng lý đơn hàng



Giao diện quản lý khách hàng

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* Kết quả đạt được
* Nắm được cơ sở lý thuyết.
* Nắm được quy trình thiết kế và xây dựng một website đặt phòng homestay.
* Nghiên cứu, tìm hiểu về phân tích thiết kế hướng đối tượng, famework Laravel là công cụ khá mạnh mẽ hiện nay trong lĩnh vực lập trình Web đặc biệt là các website lớn.
* Xây dựng thành công website đặt phòng Homestay với đầy đủ các chức năng cơ bản.
* Một số hạn chế
* Chưa thể test hết được các lỗi phát sinh mới để khắc phục.
* Các chức năng cần cải tiến hơn nữa để có thể phù hợp với yêu cầu ngày càng cao của người dùng.
* Một số loại sản phẩm thông tin còn chưa đầy đủ cần cập nhật thêm.
* Hướng phát triển
* Tối ưu hóa các chức năng hiện có
* Nâng cấp thêm một số chức năng như: thanh toán đơn hàng, giao tiếp với admin,..
* Tối ưu hóa an toàn bảo mật thông tin cho website

# TÀI LIỆU THAM KHẢO

[1] PGS.TS. Đặng Văn Đức, *Phân tích thiết kế hướng đối tượng*, Viện Khoa học và Công nghệ Việt Nam, 2002.

[2] Phạm Hữu Khang, *Xây dựng ứng dụng Web bằng PHP & MySQl*, Nhà xuất bản Mũi Cà Mau, 2003.

[3] Nguyễn Trường Sinh, *Sử dụng PHP & MySQL thiết kế Web động*, Nhà xuất bản Thống Kê, 2005.

[4] Khuất Thùy Phương, *Lập trình web với PHP và Mysql*, Nhà xuất bản Đại học Quốc gia TP.Hồ Chí Minh, 2009.

[5] Website: <http://php.net/>

[6] Website: <http://www.w3school.com>

[7] Website: <http://hocphp.info>

[8] Website: <http://www.mysql.com>

[9] Website: <https://jquery.com/>

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*Thái Nguyên, ngày … tháng …. năm 2019*

**Giảng viên hướng dẫn**