|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**  **TRẦN HOÀNG NAM**  **XÂY DỰNG WEBSITE CUNG CẤP DỊCH VỤ TÌM KIẾM NHÀ Ở,NHÀ TRỌ,BẤT ĐỘNG SẢN PHÒNG TỐT**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **HƯNG YÊN - 2016** |

|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**  **TRẦN HOÀNG NAM**  **XÂY DỰNG WEBSITE CUNG CẤP DỊCH VỤ TÌM KIẾM NHÀ Ở,NHÀ TRỌ,BẤT ĐỘNG SẢN PHÒNG TỐT**  NGÀNH: CÔNG NGHỆ THÔNG TIN  CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGƯỜI HƯỚNG DẪN**  **ĐÀO ANH HIỂN**  **HƯNG YÊN - 2016** |

MỤC LỤC

[MỤC LỤC 3](#_Toc469981004)

[DANH SÁCH HÌNH VẼ 5](#_Toc469981005)

[DANH SÁCH BẢNG BIỂU 6](#_Toc469981006)

[DANH SÁCH TỪ VIẾT TẮT 7](#_Toc469981007)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 8](#_Toc469981008)

[1.1 Lý do chọn đề tài 8](#_Toc469981009)

[1.2 Mục tiêu của đề tài 8](#_Toc469981010)

[1.3 Giới hạn và phạm vi của đề tài 9](#_Toc469981011)

[1.4 Nội dung thực hiện. 9](#_Toc469981012)

[1.5 Phương pháp tiếp cận 9](#_Toc469981013)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 10](#_Toc469981014)

[2.1 Tổng quan về ASP.NET MVC 10](#_Toc469981015)

[2.2 Mô hình MVC là gì? 10](#_Toc469981016)

[2.3 Tổng quan về ASP.NET MVC và Web API 2 12](#_Toc469981017)

[CHƯƠNG 3: NỘI DUNG THỰC HIỆN 14](#_Toc469981018)

[3.1 Phân tích yêu cầu của hệ thống 14](#_Toc469981019)

[3.1.1 Kiến trúc tổng quan của Phòng Tốt 14](#_Toc469981020)

[3.1.2 Đặc tả Use Case 15](#_Toc469981021)

[3.2 Biểu đồ lớp (mức phân tích). 19](#_Toc469981022)

[3.2.1 Biểu đồ lớp 19](#_Toc469981023)

[3.3 Thiết kế cơ sở dữ liệu 21](#_Toc469981024)

[3.3.1 Biểu đồ cơ sở dữ liệu 21](#_Toc469981025)

[3.3.2 Các thuộc tính của các bảng cơ sở dữ liệu 21](#_Toc469981026)

[3.4 Xây dựng kiến trúc ASP.NET Web API 22](#_Toc469981027)

[4.1.1. Mô hình kiến trúc tổng quan về ASP.NET Web API 23](#_Toc469981028)

[3.5 Thiết kế giao diện 25](#_Toc469981029)

[3.5.1 Giao diện chính của website 25](#_Toc469981030)

[3.5.2 Giao diện đăng nhập 26](#_Toc469981031)

[3.5.3 Giao diện đăng ký 26](#_Toc469981032)

[3.5.4 Giao diện thêm tin phòng 27](#_Toc469981033)

[3.5.5 Giao diện kết quả tìm kiếm 28](#_Toc469981034)

[3.5.6 Giao diện chi tiết tin phòng 29](#_Toc469981035)

[3.5.7 Giao diện chi tiết tin tức 31](#_Toc469981036)

[CHƯƠNG 4: KẾT LUẬN 32](#_Toc469981037)

[4.1 Kết quả đạt được của đề tài 32](#_Toc469981038)

[4.2 Hạn chế của đề tài 32](#_Toc469981039)

[4.3 Hướng phát triển của đề tài 32](#_Toc469981040)

[TÀI LIỆU THAM KHẢO 33](#_Toc469981041)

DANH SÁCH HÌNH VẼ

[Hình 3‑1: Mô hình kiến trúc tổng thể hệ thống Phòng Tốt 14](#_Toc469397738)

[Hình 3‑3: Mô hình biểu đồ lớp 19](#_Toc469397739)

DANH SÁCH BẢNG BIỂU

[Bảng 3‑1: Lớp Info 19](#_Toc468544711)

[Bảng 3‑2: Lớp CategoryInfo 19](#_Toc468544712)

[Bảng 3‑2: Lớp User 19](#_Toc468544713)

DANH SÁCH TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Giải thích |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# TỔNG QUAN VỀ ĐỀ TÀI

* 1. Lý do chọn đề tài

Hiện nay Internet là 1 công cụ, môi trường làm việc thuận lợi để liên kết mọi người lại với nhau.Internet rộng khắp và hỗ trợ con người trong nhiều lĩnh vực.

Trong thế giới rộng lớn có hàng tỉ người đang sử dụng internet hằng ngày, phần lớn họ thường đọc các tin rao vặt khi lướt web và đó là cơ hội lớn để các nhà kinh doanh hay cá nhân, tổ chức nào đó có thể quản bá, tiếp thị sản phẩm của mình đến với người dung.

Kinh tế phát triển có rất nhiều các công ty các ngành nghề mới mọc ra. Nhu cầu việc làm của những người dân miền núi và các vùng miền nghèo đang ngày càng cao vì vậy họ sẽ cần phải tìm và thuê các nhà trọ, nhà ở để phục vụ cho công việc của họ. Chính vì vậy nhu cầu tìm kiếm chỗ ở tìm kiếm các ngôi nhà phù hợp với họ ngày càng cao. Hệ thống website rao vặt nhà đất sẽ giúp họ khắc phục những khó khan tìm kiếm nhà ở, tìm kiếm nhanh hơn, dễ dàng hơn.

Qua đây chúng ta cũng thấy được các mô hình tìm kiếm nhà ở rất đa dạng, đáp ứng được phần nào nhu cầu của xã hội. Tuy nhiên vẫn còn khá nhiều điểm hạn chế như:

-Các hình thức quảng cáo tờ rơi không mang lại hiệu quả.

-Người tìm việc phải mất kinh phí để đăng tin tìm kiếm nhà ở cho các trung tâm môi giới.

* 1. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng hệ thống website tìm kiếm, rao vặt nhà ở, đất đai, bất động sản cho người cho thuê hoặc bán có thể dễ dàng tìm kiếm người dung và ngược lại. Mang lại nhiều lựa chọn với những phương thức thanh toán hiệu quả linh động, bản đồ tích hợp sẵn trong việc xác định vị trí và chỉ dẫn đến vị trí người cho thuê là 1 lợi thế

Nắm vững những ngôn ngữ lập trình ASP, HTML, JQERY, BOOSTRAP, ANGULARJS, C#...…

* 1. Giới hạn và phạm vi của đề tài

Đề tài xây dựng dựng website cung cấp dịch vụ tìm kiếm nhà ở, nhà trọ, bất động sản sử dụng các công nghệ mới trong lập trình như ASP.NET, LINQ, SQL Server 2014 và Công cụ lập trình Visual Studio 2015.

* 1. Nội dung thực hiện.

Nội dung thực hiện/nghiên cứu cụ thể như sau:

* Chức năng xem thông tin các bài đăng trên hệ thống.
* Thông tin địa điểm, giá tiền, số điện thoại.......
* Đăng tin rao vặt về nhà ở, nhà trọ, ở ghép.
* Các tin rao vặt theo các loại như Văn phòng, chung cư.
* Các tin tức về bất động sản
* Tìm kiếm các địa điểm trên giúp người dùng định hướng được chỗ mình sẽ ở trong tương lai
* Kiểm thử hệ thống.
* Triển khai thực nghiệm hệ thống trên mạng Internet.
  1. Phương pháp tiếp cận
* Sử dụng các phương pháp nghiên cứu:
  + Phương pháp đọc tài liệu;
  + Phương pháp phân tích mẫu;
  + Phương pháp thực nghiệm.

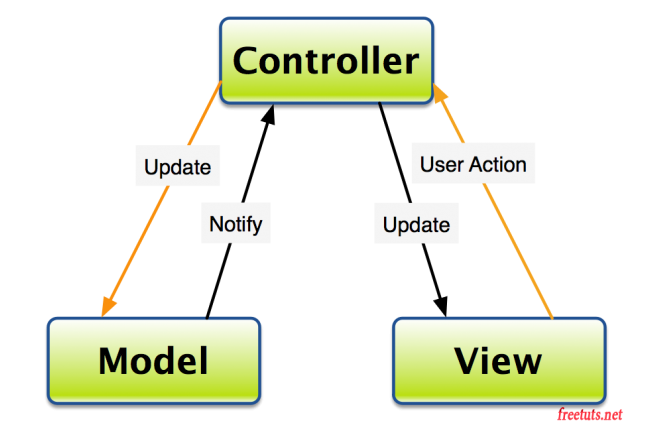
# CƠ SỞ LÝ THUYẾT

* 1. Tổng quan về ASP.NET MVC
  2. Mô hình MVC là gì?

MVC là chữ viết tắt của **M**odel - **V**iew - **C**ontroller, đây là một mô hình kiến phần mềm được tạo ra với mục đích quản lý và xây dựng dự án phần mềm có hệ thống hơn. Mô hình này được dùng khá rộng rãi và đặc biệt là trong các ngôn ngữ lập trình web. Trong PHP hiện tại có khá nhiều Framework và tất cả đều xây dựng từ mô hình MVC. Trong mô hình này thì:

* **Model:** có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu và controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua **View.**
* **View:** có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung sang các đoạn mã HTML, bạn có thể hiểu người ta còn gọi là thành phần giao diện.
* **Controller:** đóng vài trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, loadmodel tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client.

Mô hình thể hiện mối quan hệ trong mô hình MVC



Hình 1: Mô hình thể hiện mối quan hệ trong MVC

* Sự khác biệt giữa mô hình MVC và các mô hình khác
* Hệ thống phân ra từng phần nên dễ dàng phát triển: model, view, controller.
* Hệ thống không sử dụng view state hoặc sever-based form dễ quản lý hết các khía cạnh của ứng dụng.
* Hệ thống chia thành nhiều modun nhỏ nên nhiều người có thể làm dự án chung
* Dễ dàng debug trong quá trình xây dưng dự án.
* Bảo trì tốt và dễ nâng cấp.
* Luồng sử lý trong mô hình MVC’
* Bước 1: Dựa vào yêu cầu, bạn chuyền vào một ID nên controller sẽ gọi tới một hàm lấy dữ liệu theo ID trong model
* Bước 2: Sau khi có dữ liệu controller sẽ gửi qua view, lúc này view có nhiệm vụ sử lý dữ liệu và convert thành những đoạn mã HTML
* Bước 3: Sau khi view kết thúc thì controller sẽ gửi trả nội dung HTML của view về cho client nên bạn thực hiện được yêu cầu
  1. Tổng quan về ASP.NET MVC và Web API 2

Tính năng ASP.NET Web API trước đây được gọi là WCF Web API, nghĩa là WCF Web API đã được tích hợp vào ASP.NET, thay vì được xem là một thành phần của WCF, điều đó ít nhất cũng mang lại cho lập trình viên cảm giác thân thuộc và không phải sợ hãi khi nghe đến danh xưng WCF.

Vậy ASP.NET Web API là gì? Và tại sao dùng ASP.NET Web API?

ASP.NET MVC được thiết kể để phục vụ cho việc giao tiếp giữa người dùng và ứng dụng thông qua các trang web. Tình huống sử dụng chính là sản sinh ra các trang HTML và tương tác người dùng (như nhập dữ liệu, click chuột vào nút nhấn, link…) và rõ ràng là ASP.NET MVC làm rất tốt việc đó.

ASP.NET Web API được xây dựng cho phần còn lại, tức là những thứ không phải do người dùng tương tác. Ví dụ, như mã lệnh jQuery gởi đến một ajax request, hoặc một phần mềm trên smartphone cần lấy thông tin. Trong những trường hợp như vậy, các lời triệu gọi đến từ mã lệnh và yêu cầu dữ liệu có cấu trúc và có thể gởi thêm một số mã HTTP Status Codes.

Nói như vậy không có nghĩa là chúng ta không thể phục vụ các yêu cầu tượng tự bằng ASP.NET MVC, nhưng nó đòi hỏi lập trình viên phải làm nhiều việc hơn để mọi việc diễn ra suôn sẻ. ASP.NET Web API được tích hợp vào ASP.NET MVC đồng nghĩa với việc chúng ta vừa có thể xây dựng ứng dụng theo mô hình MVC vừa có thể cung cấp các HTTP Services thực sự.

ASP.NET Web API hỗ trợ các tính năng dưới đây:

• Mô hình lập trình HTTP hiện đại: Truy xuất trực tiếp và khởi tạo các HTTP request và response bằng Web APIs sử dụng strongly typed HTTP object model. Có thể lập trình theo mô hình này ở phần client, bằng cách sử dụng kiểu HttpClient.

• Hỗ trợ đấy đủ cho việc định tuyến: Web APIs hỗ trợ đầy đủ cho khả năng định tuyến, kể cả route parameters và các constraints. Hơn nữa, Web APIs còn hỡ trợ đấy đủ cho các HTTP Verbs (POST, GET, PUT, DELETE..) nên bạn sẽ không cần phải sử dụng các thuộc tính như là [HttpPost] cho các phương thức của bạn nữa.

• Thương thảo về nội dung trả về: client và server có thể thảo thuận với nhau định dàng cần trả về từ một API. ASP.NET Web APIs hỗ trợ mặc định cho XML, JSON và định dạng Forrm URL-encoded, và bạn có thể mở rộng sự hỗ trợ này bằng cách định nghĩa các fortmatter của riêng bạn.

• Model binding và validation: Model binders cung cấp cách thức giả nén dữ liệu từ các phần khác nhau của HTTP request và chuyển các phần nói trên thành .NET objects một cách dễ dàng bằng cách sử dụng các Web API actions.

• Filters: Web APIs bây giờ đã hỗ trợ các bộ lọc, ví dụ như [Authorize].

• Query composition: Chỉ cần trả về các IQueryable, bạn có thể truy vấn dữ liệu bằng cách sử dụng ODATA URL conventions.

• Tăng cường khả năng kiểm thử các chi tiết HTTP: bạn có thể làm việc với các đối tượng HttpRequestMessage, HttpResponseMessage đễ quy định các chi tiết của HTTP.

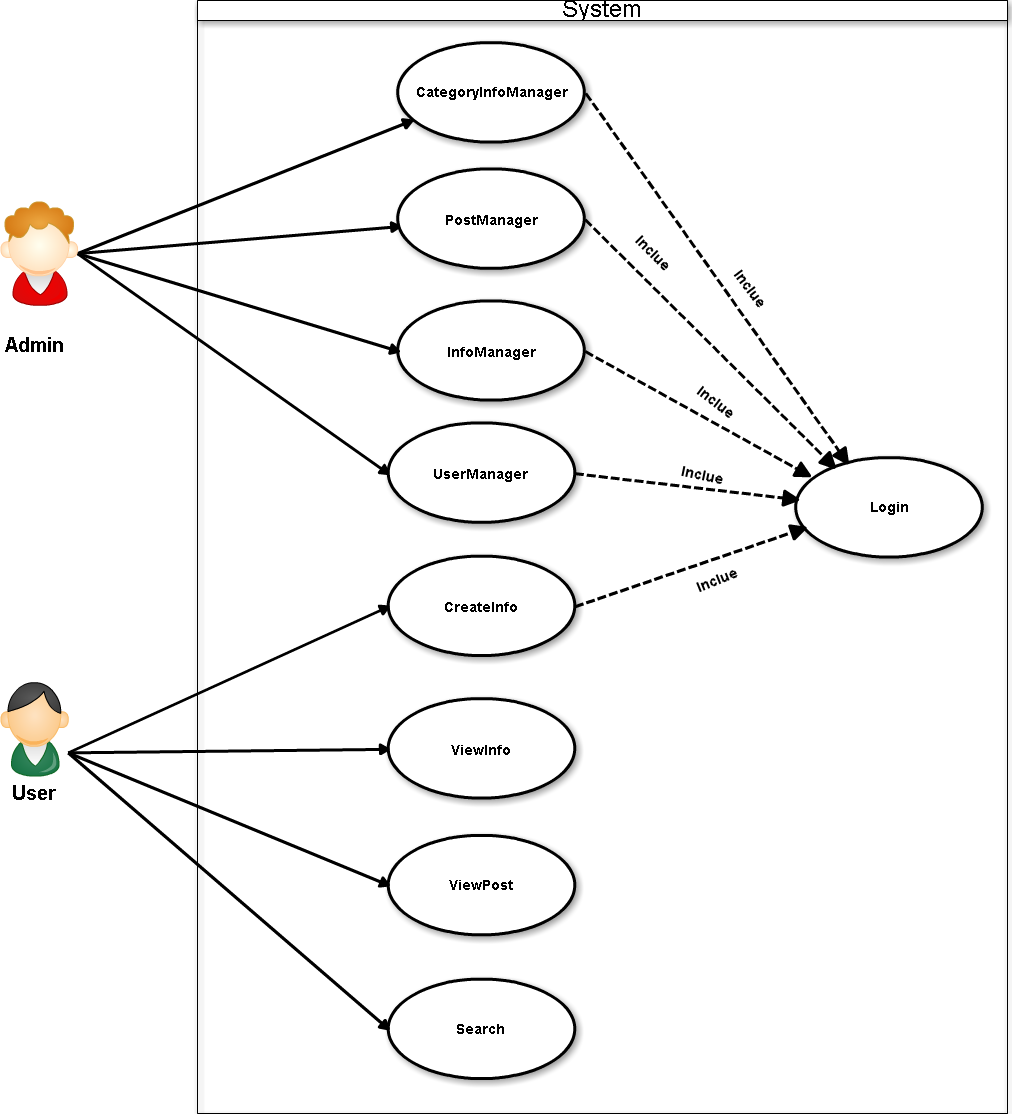
• Cải thiện Inversion of Control (IoC) thông qua Dependency Resolver: Web API hiện nay đã sử dụng mẫu service locator pattern bằng cách sử dụng MVC dependency resolver để triệu gọi các đối tượng từ các lớp khác nhau.

• Cấu hình dạng code-based (Code-based configuration): các thiệt lập cho Web API có thể thực hiện qua mã lệnh, và giúp cho tập tin cấu hình của bạn sạch sẽ hơn.

• Self-host: Web APIs có thể được host trên tiến trình của bạn thay vì chỉ IIS mà vẫn tận dụng được tính năng định tuyến..

# NỘI DUNG THỰC HIỆN

* 1. Phân tích yêu cầu của hệ thống
     1. Kiến trúc tổng quan của Phòng Tốt



Hình 3‑1: Mô hình kiến trúc tổng thể hệ thống Phòng Tốt

Trong mô hình kiến trúc tổng thể trên, ta có thể thấy hệ thống Phòng tốt được xây dựng nên từ việc kết hợp tốt là giải pháp thiết kế các Module, các trang và lưu trữ Database.

* + 1. Đặc tả Use Case
* **Đặc tả Use Case CeateInfo**.
  + Mục đích

Giúp cho người dung có thể tạo thông tin phòng trọ của mình cần cho thuê

* + Tác nhân liên quan

Người dung (User) và hệ thống (system)

* + Điều kiện trước

Người dùng phải đăng nhập vào hệ thống

* + Điều kiện sau

Nếu như thành công thì hệ thống sẽ thông báo là bạn đã cập nhật thành công

Còn nếu như thất bại thì hệ thống sẽ thông báo là bạn cập nhật không thành công

* + Biểu đồ Use Case



* + Dòng Sự kiện chính

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng tạo tin phòng | 1. Hiển thị giao diện phòng mà mình đã chọn. |
| 1. Thực hiện các chức năng them các thông tin | 1. Hiển thị giao diện tương ứng cho chức năng tương ứng mà tác nhân đã chọn để thực hiện công việc. |

* **Đặc tả Use Case Search.**
  + Mục đích

Giúp cho người dung có thể tìm kiếm phòng mà mình đang cần

* + Tác nhân liên quan

Người dung (User) và hệ thống (system)

* + Biểu đồ Use Case



* + Dòng Sự kiện chính

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng tìm kiếm | 1. Hiển thị giao diện tìm kiếm phòng trọ |
| 3. Chọn chức năng cần thao tác trong các chức năng trên | 4. Hiển thị giao diện tương ứng cho các thông tin mà người dung tìm kiếm được |

* **Đặc tả Use Case ViewInfo.**
  + Mục đích

Giúp cho người dung có thể xem được các tin phòng mà mình cần tìm

* + Tác nhân liên quan

Người dung (User) và hệ thống (system)

* + Biểu đồ Use Case



* + Dòng Sự kiện chính

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng xem tin phòng | 1. Hiển thị giao diện thông tin của phòng trọ đó |

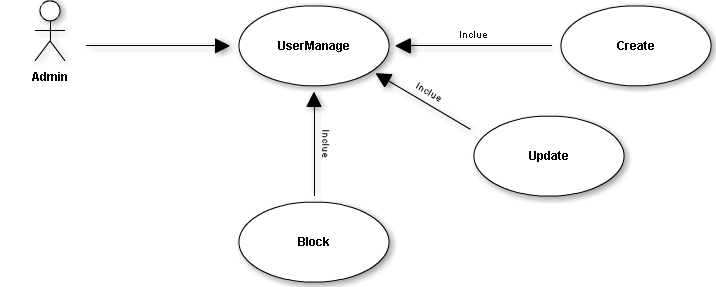
* **Đặc tả Use Case UserManage.**
  + Mục đích

Giúp cho người quản trị có thể quản lý được những người dung trên hệ thống.

* + Tác nhân liên quan

Quản trị (Admin) và hệ thống (system)

* + Biểu đồ Use Case



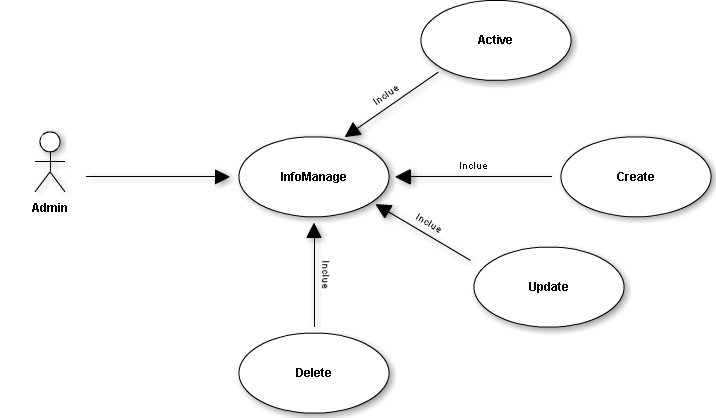
* **Đặc tả Use Case InfoManage.**
  + Mục đích

Giúp cho admin quản lý được các bài đăng tin phòng trên hệ thống

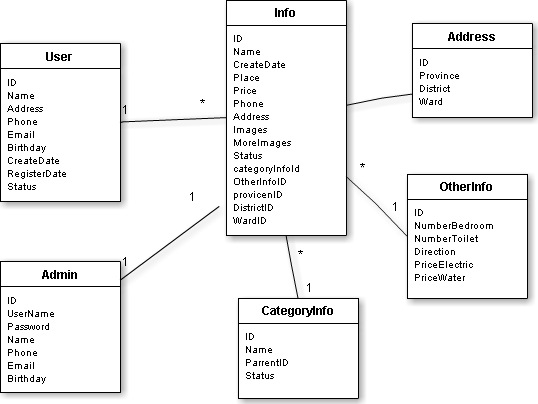
* + Tác nhân liên quan

Quản trị (Admin) và hệ thống (system)

* + Biểu đồ UseCase



* 1. Biểu đồ lớp (mức phân tích).
     1. Biểu đồ lớp



Hình 3‑3: Mô hình biểu đồ lớp

Trong đó các lớp dữ liệu được mô tả như sau:

Bảng 3‑1: Lớp Info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Loại | Ràng buộc | Ý nghĩa/ghi chú |
| 1 | ID | private | Khóa chính | Mã |
| 2 | Name | private | Not null | Tiêu đề tin |
| 3 | Price | private | Not null | Gía tiền |
| 4 | Phone | private | Not null | Số điện thoại |
| 5 | Address | private | Not null | Địa chỉ |
| 5 | Images | private | Not null | Ảnh |
| 6 | MoreImages | private | Not null | Ảnh Khác |
| 7 | Category | private | Khóa phụ | Loại tin |
| 8 | OtherInfoID | private | Khóa phụ | Thông tin thêm |
| 9 | AddressID | private | Khóa phụ | Địa chỉ cụ thể |

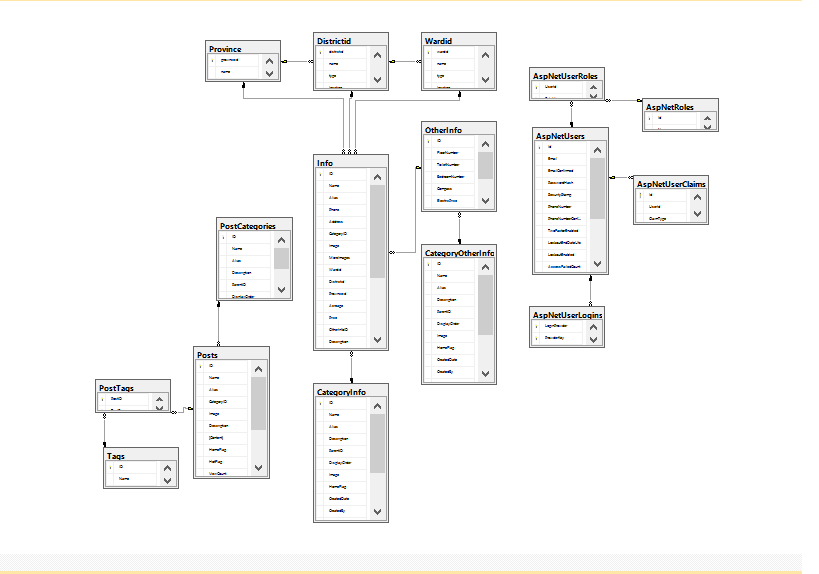
Bảng 3‑2: Lớp CategoryInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Loại | Ràng buộc | Ý nghĩa/ghi chú |
| 2 | ID | private | Khóa chính | Mã |
| 3 | Name | private | Not null | Tên |
| 4 | ParrentID | private | Not null | Cấp |
| 5 | Status | private | Not null | Trạng thái |

Bảng 3‑2: Lớp User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Loại | Ràng buộc | Ý nghĩa/ghi chú |
| 1 | ID | private | Khóa chính | Mã |
| 2 | Name | private | Not null | Tên |
| 3 | Address | private | Not null | Địa chỉ |
| 4 | Email | private | Not null | Email |
| 5 | Birthday | private | Not null | Ngày sinh |
| 6 | Phone | private | Not null | Số điện thoại |

* 1. Thiết kế cơ sở dữ liệu
     1. Biểu đồ cơ sở dữ liệu



Hình 3‑4: Mô hình biểu đồ cơ sở dữ liệu

* + 1. Các thuộc tính của các bảng cơ sở dữ liệu

Bảng 3‑3: Bảng Info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Loại | Ràng buộc | Ý nghĩa/ghi chú |
| 1 | ID | private | Khóa chính | Mã |
| 2 | Name | private | Not null | Tiêu đề tin |
| 3 | Price | private | Not null | Gía tiền |
| 4 | Phone | private | Not null | Số điện thoại |
| 5 | Address | private | Not null | Địa chỉ |
| 5 | Images | private | Not null | Ảnh |
| 6 | MoreImages | private | Not null | Ảnh Khác |
| 7 | Category | private | Khóa phụ | Loại tin |
| 8 | OtherInfoID | private | Khóa phụ | Thông tin thêm |
| 9 | AddressID | private | Khóa phụ | Địa chỉ cụ thể |

Bảng 3‑3: Bảng User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Loại | Ràng buộc | Ý nghĩa/ghi chú |
| 1 | ID | private | Khóa chính | Mã |
| 2 | Name | private | Not null | Tên |
| 3 | Address | private | Not null | Địa chỉ |
| 4 | Email | private | Not null | Email |
| 5 | Birthday | private | Not null | Ngày sinh |
| 6 | Phone | private | Not null | Số điện thoại |

Bảng 3‑3: Bảng CategoryInfo

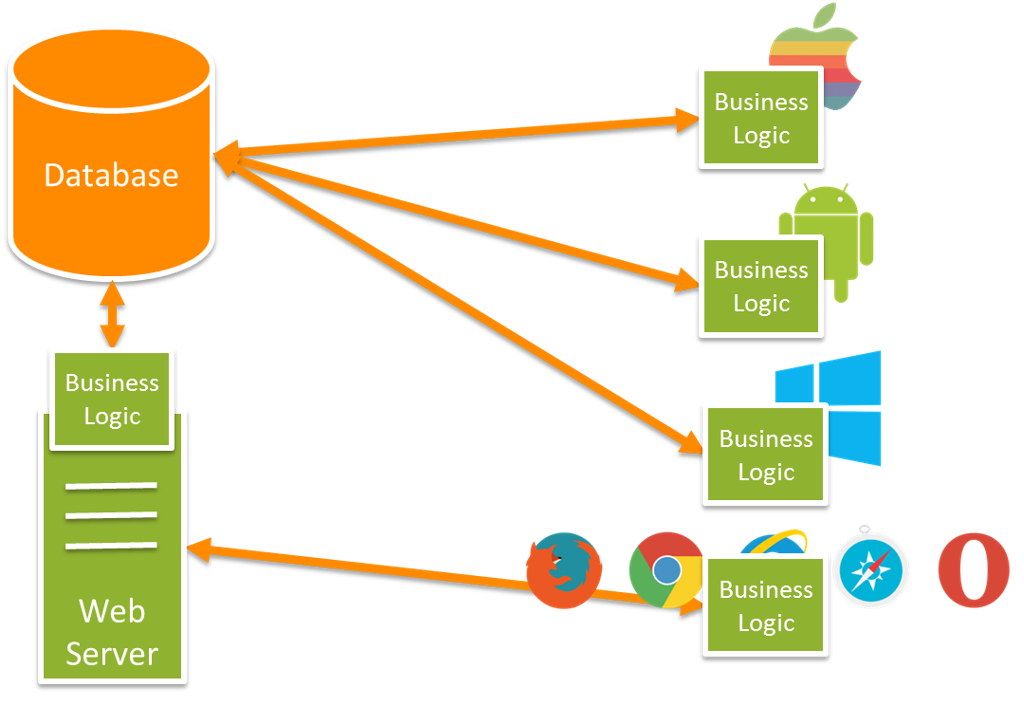
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Loại | Ràng buộc | Ý nghĩa/ghi chú |
| 2 | ID | private | Khóa chính | Mã |
| 3 | Name | private | Not null | Tên |
| 4 | ParrentID | private | Not null | Cấp |
| 5 | Status | private | Not null | Trạng thái |

* 1. Xây dựng kiến trúc ASP.NET Web API

ASP.NET Web API là framework giúp chúng ta tạo ra các Web API – API trên nền web (HTTP)

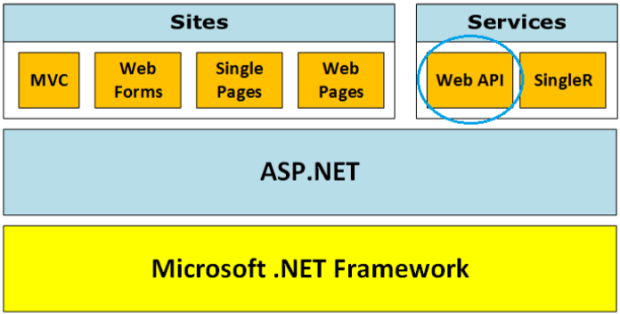
Web API là các dịch vụ Web (Web service) được xây dựng dựa trên HTTP sử dụng mô hình lập trình convention (như ASP.NET MVC).

Web API là 1 dịch vụ có thể phát triển dat nền tảng: Web, WindowPhone, Android…



Hình 4. 1: Mô hình API

### 4.1.1. Mô hình kiến trúc tổng quan về ASP.NET Web API



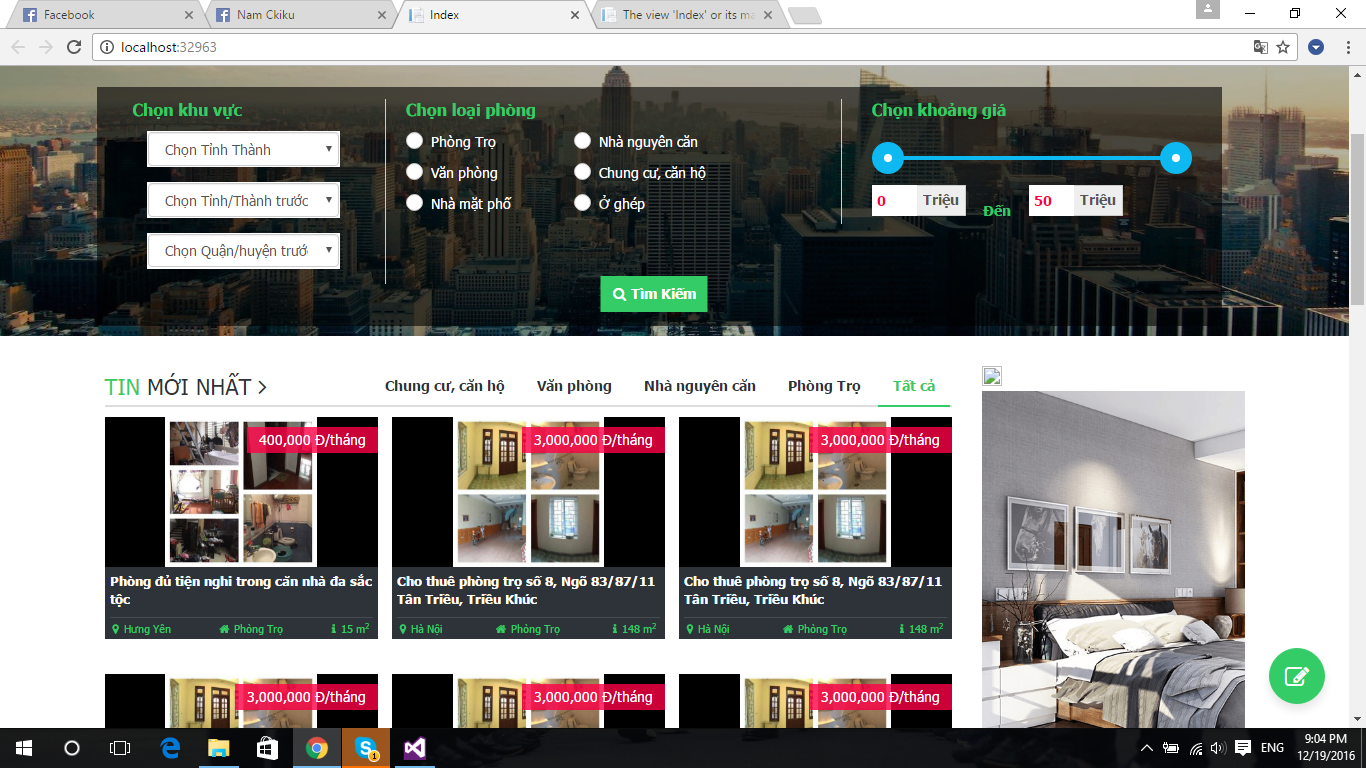
Hình 4. 2: Kiến trúc mô hình API

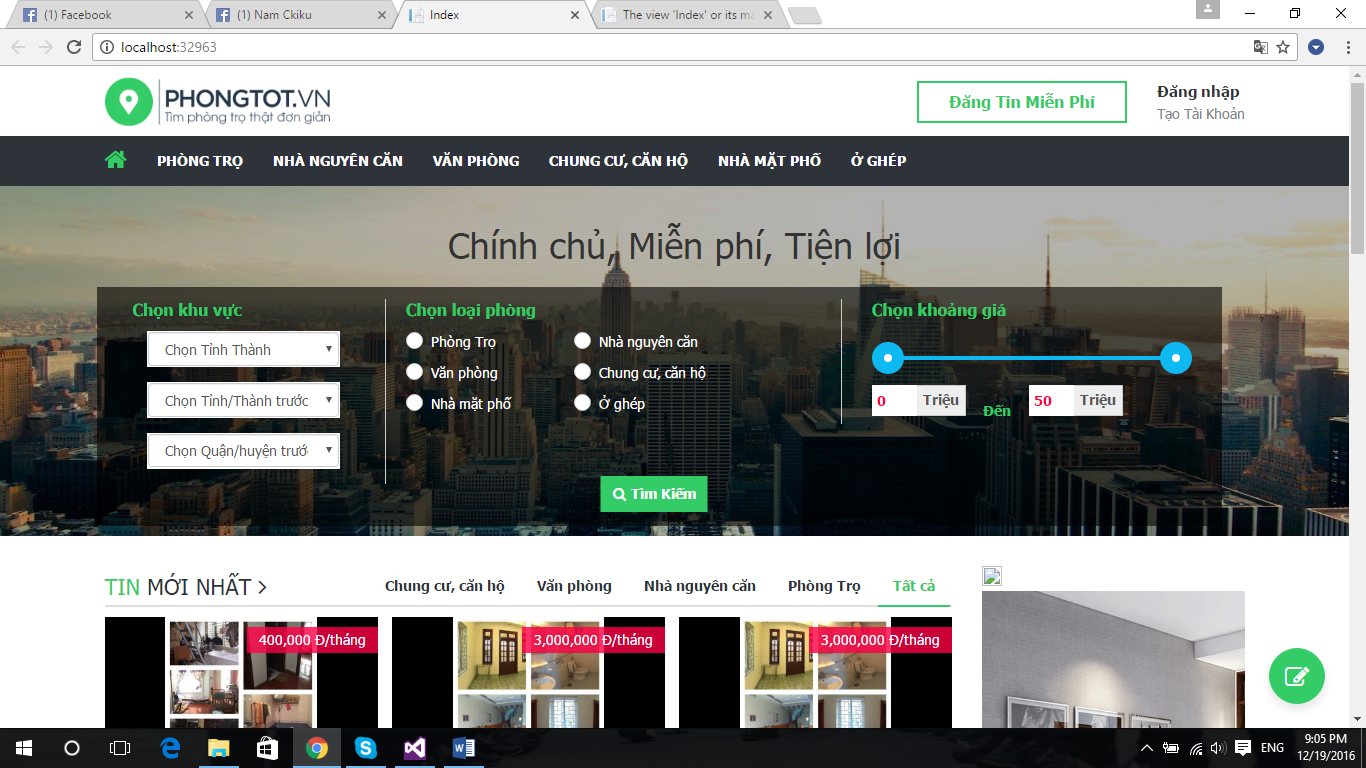
**Đặc** **điểm Web API (.NET 4.0 trở lên)**

* Giúp cho việc xây dựng các HTTP service rất đơn giản, nhanh chóng
* Mã nguồn mở (Open Source) và có thể được sử dụng bởi bất kì client nào hỗ trợ XML, JSON.
* Hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content formats.
* Có thể host trong ứng dụng hoặc trên IIS.
* Kiến trúc lý tưởng cho các thiết bị có băng thông giới hạn như các thiết bị di động.
* Định dạng dữ liệu có thể là JSON, XML hoặc một kiểu dữ liệu bất kỳ.
* Làm mới và hiện đại hóa các mẫu dự án mặc định
* Mẫu dự án trên điện thoại di động
* Nhiều tính năng mới để hỗ trợ các ứng dụng di động
* Tùy chỉnh sinh mã(code).
* Tăng cường hỗ trợ cho các phương pháp bất đồng bộ
* Đọc danh sách đầy đủ tính năng trong các ghi chú phát hành

**Ưu** **điểm** **của Web API**

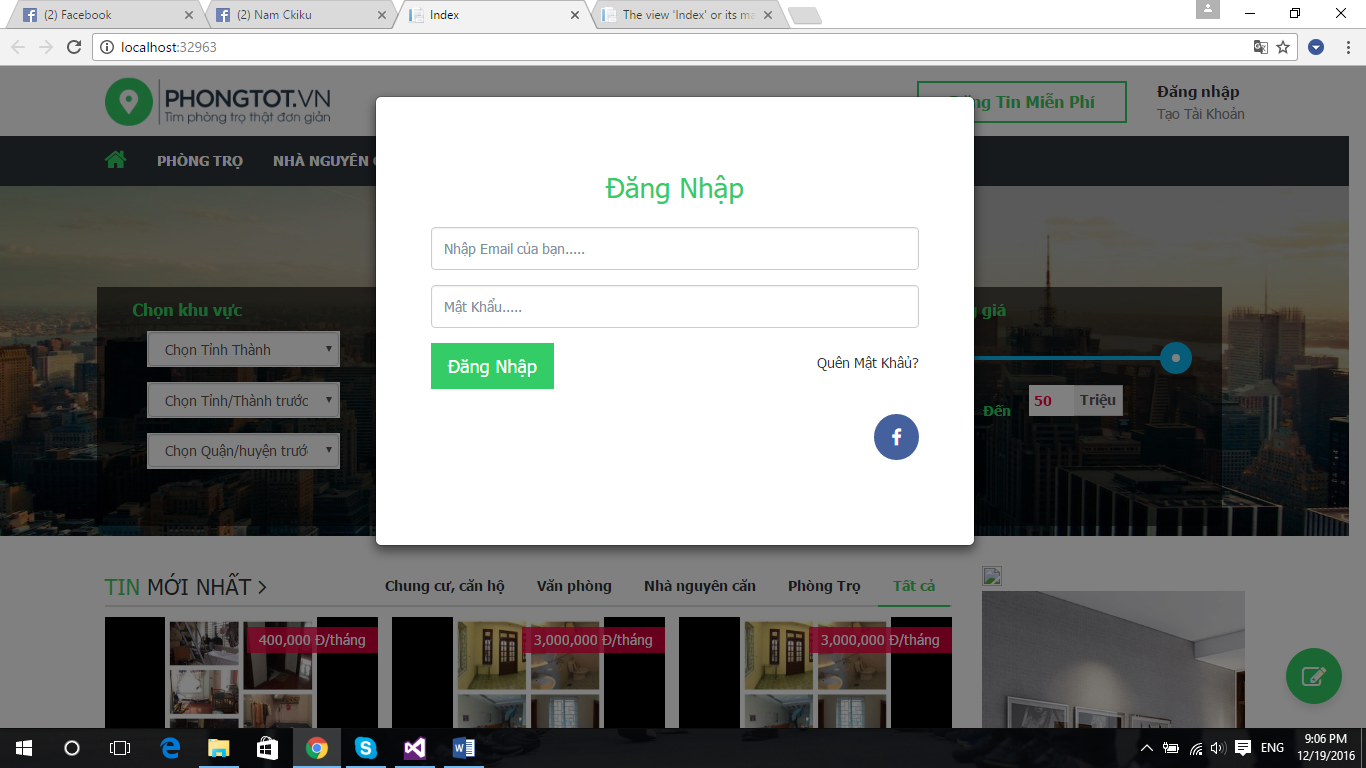
* Cấu hình đơn giản hơn nhiều so với WCF.
* Hiệu suất(performance) cao.
* Hỗ trợ RESTfull đầy đủ.
* Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test, …
* Mã nguồn mở (Open source).
  1. Thiết kế giao diện
     1. Giao diện chính của website





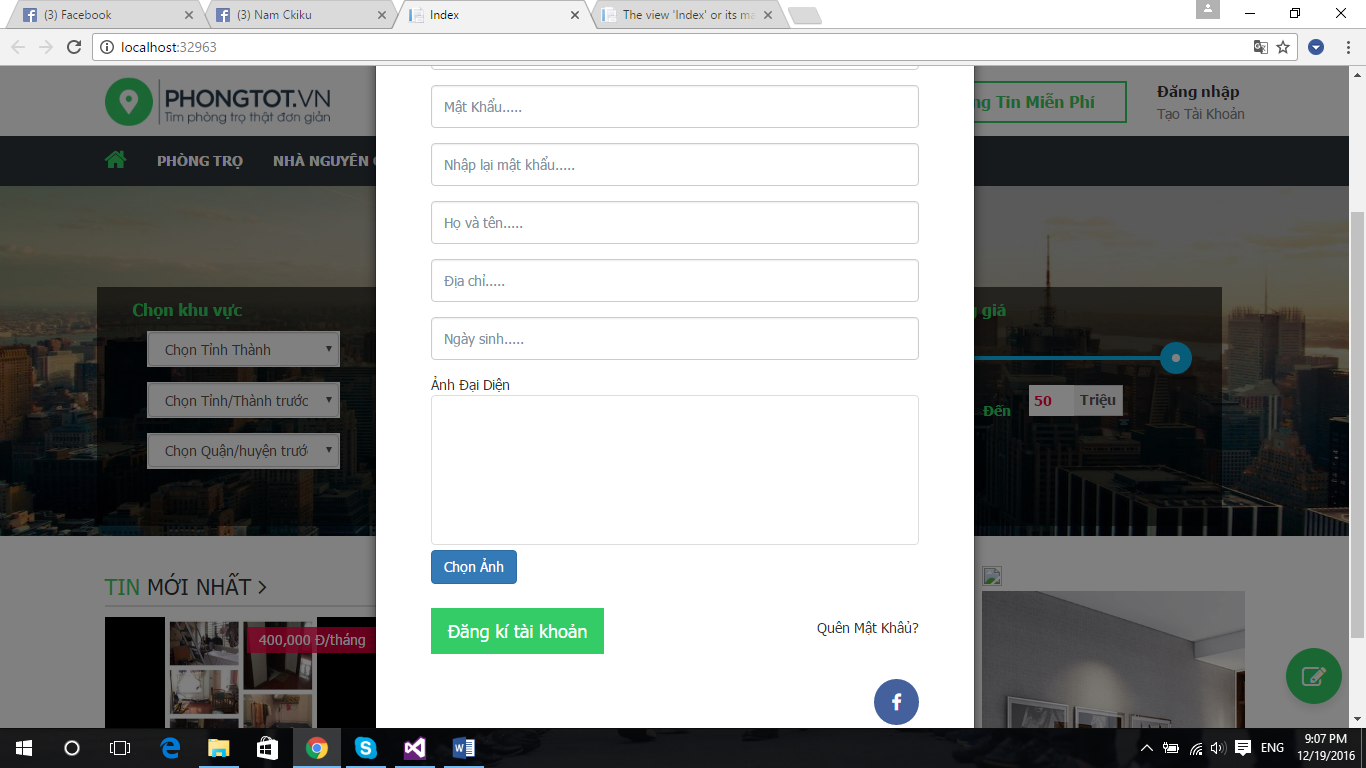
Hình 4. 3: Giao diện chính của website

* + 1. Giao diện đăng nhập

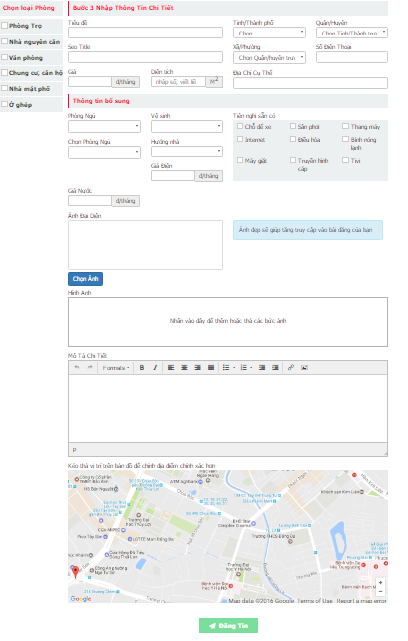


Hình 4. 4: Giao diện đăng nhập

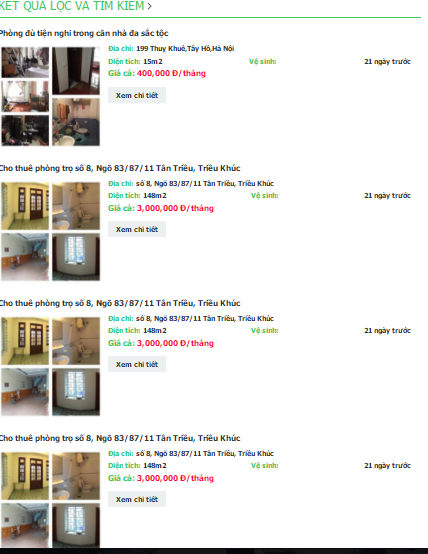
* + 1. Giao diện đăng ký



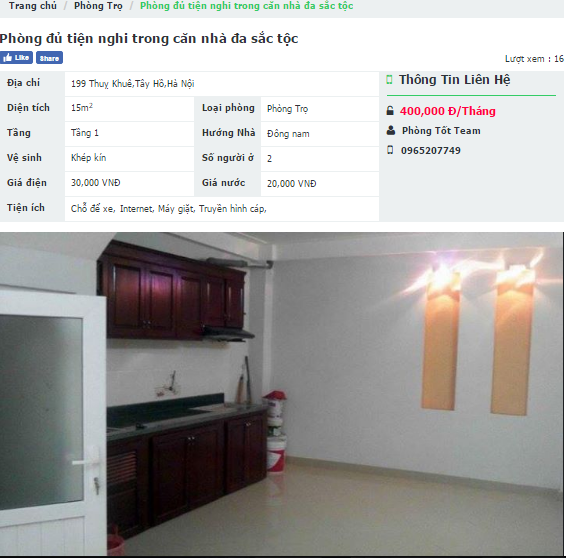
* + 1. Giao diện thêm tin phòng

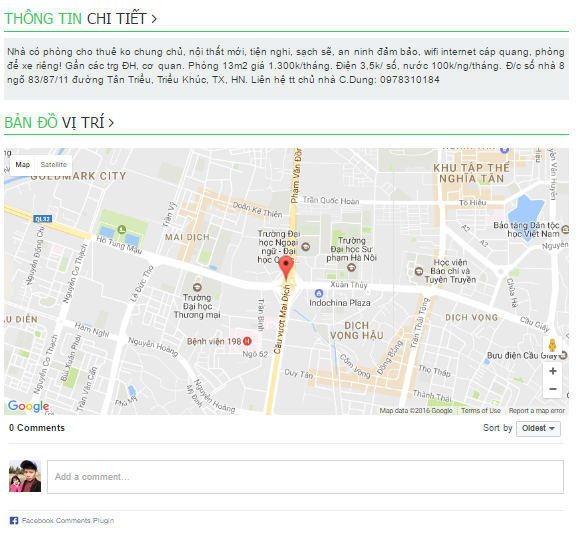


* + 1. Giao diện kết quả tìm kiếm

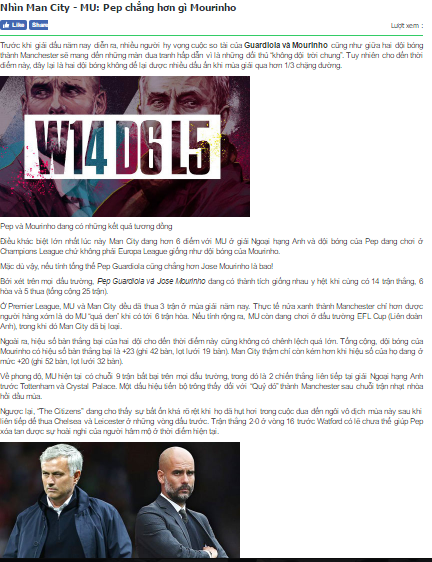


* + 1. Giao diện chi tiết tin phòng





* + 1. Giao diện chi tiết tin tức



# KẾT LUẬN

* 1. Kết quả đạt được của đề tài

Trong quá trình xây dựng website cung cấp dịch vụ học trực truyến em đã đạt được như sau:

- Việc website này giúp cho việc 1 người có thể vừa tìm kiếm vừa đăng thông tin trên hệ thông.

- Các chức năng của website tương đối đầy đủ thông tin.

- Khởi tạo các chức năng in ấn và tìm kiếm giúp cho việc tìm kiếm nhanh hơn và chính xác hơn chỉ bằng các nút lệnh

- Giao diện dễ sử dụng, thân thiện và gần gũi với người dùng.

- Xử lý thông tin khá tốt với chức năng tự động kiểm tra tính đúng đắn của dữ liệu khi nhân viên quản lý nhập các thông tin sai quy định

- Hiển thị thông báo khi xảy ra lỗi trùng lặp dữ liệu trong quá trình cập nhật thông tin.

* 1. Hạn chế của đề tài

Hệ thống chức năng chưa thực sự đầy đủ để giúp cho việc quản lý bài viết, phần mềm còn đơn giản

Hệ thống bao quát được hết các thông tin trong của cửa hàng, một số chức năng còn chưa đầy đủ.

* 1. Hướng phát triển của đề tài

Trong thời gian nghiên cứu và thực hiện đề tài, tác giả đã vạch ra được hướng phát triển tiếp theo của đề tài như sau:

* Hoàn thiện các chức năng còn thiếu sót của website
* Xây dựng chương trình để áp dụng thực tiễn

TÀI LIỆU THAM KHẢO

x

|  |  |
| --- | --- |
| [1] | Phạm Hữu Khang, *Lập trình ASP.NET 2.0*. Hà Nội, Việt Nam: Nhà xuất bản Lao Động Xã Hội, 2005. |

x