|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  <HỌ VÀ TÊN TÁC GIẢ> <TÊN ĐỀ TÀI> 2014  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**  **TRẦN HOÀNG NAM**  **XÂY DỰNG WEBSITE CUNG CẤP DỊCH VỤ TÌM KIẾM THÔNG TIN PHÒNG TRỌ**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **HƯNG YÊN - 2014** |

|  |
| --- |
| **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 THÔNG TIN PHÒNG TRỌ**  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 - 2014** |

MỤC LỤC

[MỤC LỤC 3](#_Toc483475493)

[DANH SÁCH HÌNH VẼ 5](#_Toc483475494)

[DANH SÁCH BẢNG BIỂU 6](#_Toc483475495)

[DANH SÁCH TỪ VIẾT TẮT 7](#_Toc483475496)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 8](#_Toc483475497)

[1.1 Lý do chọn đề tài 8](#_Toc483475498)

[1.2 Mục tiêu của đề tài 9](#_Toc483475499)

[1.3 Giới hạn và phạm vi của đề tài 9](#_Toc483475500)

[1.4 Nội dung thực hiện 9](#_Toc483475501)

[1.5 Phương pháp tiếp cận 10](#_Toc483475502)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 11](#_Toc483475503)

[2.1 Tổng quan về ASP.NET MVC và Web API 2 11](#_Toc483475504)

[2.1.1 Mô hình MVC là gì? 11](#_Toc483475505)

[2.1.2 Tại sao cần sử dụng ASP.NET MVC ? 12](#_Toc483475506)

[2.1.3 Web API 2 là gì? 13](#_Toc483475507)

[2.2 Nguyên tắc thiết kế Website cung cấp dịch vụ tìm kiếm nhà trọ 15](#_Toc483475508)

[2.2.1 Nguyên tắc 1: Tổ chức website chặt chẽ và dễ sử dụng 15](#_Toc483475509)

[2.2.2 Nguyên tắc 2: Đảm bảo các liên kết được khám phá dễ dàng 15](#_Toc483475510)

[2.3 Tổng quan về Angularjs 16](#_Toc483475511)

[2.3.1 Angluarjs là gì? Tại sao phải sử dụng nó? 16](#_Toc483475512)

[CHƯƠNG 3: NỘI DUNG THỰC HIỆN 19](#_Toc483475513)

[3.1 Phát biểu bài toán 19](#_Toc483475514)

[3.1.1 Tổng quan website cung cấp dịch vụ tìm kiếm nhà trọ 19](#_Toc483475515)

[3.1.2 Kiến trúc tổng quan của hệ thống 19](#_Toc483475516)

[3.2 Phân Tích Hệ Thống 20](#_Toc483475517)

[3.2.1 Biểu Đồ UseCase 20](#_Toc483475518)

[3.2.2 Biểu Đồ Lớp 25](#_Toc483475519)

[3.3 Thiết Kế Hệ Thống 29](#_Toc483475520)

[3.3.1 Thiết Kế Cơ Sở Dữ Liệu 29](#_Toc483475521)

[3.3.2 Thiết kế giao diện 33](#_Toc483475522)

[3.4 Cài Đặt Và Triển Khai Hệ Thống 38](#_Toc483475523)

[CHƯƠNG 4: KẾT LUẬN 39](#_Toc483475524)

[4.1 Kết quả đạt được của đề tài 39](#_Toc483475525)

[4.2 Hạn chế của đề tài 39](#_Toc483475526)

[4.3 Hướng phát triển của đề tài 39](#_Toc483475527)

[TÀI LIỆU THAM KHẢO 40](#_Toc483475528)

DANH SÁCH HÌNH VẼ

[Hình 3‑1: Mô hình kiến trúc tổng thể hệ thống 19](#_Toc483475360)

[Hình 3‑3: Mô hình lưu trữ cấu trúc website trên Database 29](#_Toc483475361)

DANH SÁCH BẢNG BIỂU

[Bảng 3‑1: Class Room 26](#_Toc483475359)

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 tìm kiếm nhà trọ 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.

Chúng ta đang sống trong thời đại của công nghệ thông tin,nhu cầu tìm kiếm nhà ở,nhà trọ trên các thành phố lớn,các khu công nghiệp ở các tỉnh thành rất cao.Đặc biệt là nhu cầu tìm kiếm nhà trọ cho các học sinh,sinh viên.Chính vì vậy rất cần 1 website cung cấp dịch vụ tìm kiếm phòng trọ để giúp cho người dùng có thể tìm kiếm một cách dễ dàng hơn cũng như giúp cho các chủ nhà trọ có thể tìm được những người ở,quảng bá phòng trọ của mình dễ dàng hơn

* 1. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng hệ thống website tìm kiếm nhà trọ,nhà ở ,văn phòng cho người cho thuê hoặc bán có thể dễ dàng tìm kiếm người dùng 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#...…

- Xây dựng ứng dụng tìm kiếm phòng trọ cho người dùng có nhu cầu

- Ứng dụng cung cấp các thông tin các phòng trọ

- Cho phép người dùng đăng tin phòng trọ của mình cần cho thuê

- Cung cấp thông tin về hình ảnh, giá và các đồ dùng sinh hoạt

- Đưa các tin về bất động sản trong và ngoài nước

* 1. Giới hạn và phạm vi của đề tài

Đề tài xây dựng dựng website cung cấp dịch vụ tìm kiếm phòng trọ, nhà ở,văn phòng sử dụng các công nghệ mới trong lập trình như ASP.NET, LINQ, SQL,JQUERY,GOOGLE MAP API, Server 2014 và Công cụ lập trình Visual Studio 2015……

*-* Tìm hiểu hiểu tổng quan về WEBAPI, ASP.NET MVC

- Khảo sát nhu cầu tìm kiếm phòng trọ của người dùng.

- Triển khai website trên nền web.

* 1. Nội dung thực hiện

Hiện một số giải pháp thiết kế web trên nền tảng công nghệ .NET nhưng còn.

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

* Chức năng xem thông tin các bài đăng trên hệ thống.
* Tra cứu thông tin địa điểm, giá tiền, số điện thoại.......
* Đăng thông tin về nhà trọ,nhà ở,văn phòng lên hệ thống.
* Các bài viết thông tin dưa lên theo các loại như phòng trọ,nhà ở,văn phòng, chung cư.
* Các tin tức,bài viết,thông tin về thị trường bất động sản,nhà đất,tin tức trong và ngoài nước…
* Tìm kiếm các địa điểm trên giúp người dùng định hướng được chỗ mình sẽ ở trong tương lai
* Tra cứu thông tin theo bản đồ giúp người dùng có thể dễ dàng tìm kiếm địa điểm theo 1 cách trực quan nhất
* Kiểm thử hệ thống.
* Triển khai thực nghiệm hệ thống trên mạng Internet.
  1. Phương pháp tiếp cận
* Cách tiếp cận : Nghiên cứu các Framework như ASP.NET MVC,ANGULARJS,ASP.NET API 2,SQL SEVER 2014,JQUEY,HTML,CSS
* Sử dụng các phương pháp nghiên cứu:
  + Phương pháp đọc tài liệu;
  + Phương pháp phân tích mẫu;
  + Phương pháp thực nghiệm.

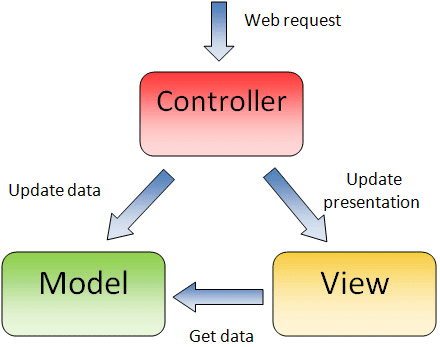
# CƠ SỞ LÝ THUYẾT

* 1. Tổng quan về ASP.NET MVC và Web API 2
     1. 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 : 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. Tại sao cần sử dụng ASP.NET MVC ?

Bạn nên sử dụng ASP.NET MVC cho việc tạo ra các ứng dụng web vì những lý do liệt kê sau đây:

- Việc chia ứng dụng thành ba thành phần model, view, controller sẽ giúp web dễ dàng quản lý

- Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)

- Những ứng dụng được phát triển bởi nhiều lập trình viên thì ASP.NET MVC sẽ hỗ trợ tốt cho các ứng dụng đó để quản lý được tính năng của ứng dụng.

Với mức lương cao, hấp dẫn cùng nhu cầu việc làm luôn ở mức cao, Lập trình với ASP.NET MVC đang là lựa chọn của rất nhiều bạn trẻ đam mê lập trình đặc biệt là lập trình web

* + 1. Web API 2 là gì?

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 hỗ 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.

* 1. Nguyên tắc thiết kế Website cung cấp dịch vụ tìm kiếm nhà trọ
     1. Nguyên tắc 1: Tổ chức website chặt chẽ và dễ sử dụng

Cách tổ chức và phân chia nội dung cho website là một trong những vấn đề rất quan trọng đối với người dùng truy cập web. Ngoài ra, bạn còn cần phải quan tâm đến bố cục của website sao cho hợp lý để người dùng dễ dàng tiếp cận các thông tin cần thiết khi truy cập. Bởi website tổ chức không tốt thì người dùng sẽ khó tiếp cận được những nội dung theo đúng ý muốn của đơn vị. Từ đó, người truy cập sẽ nhanh chóng thoát ra khỏi web và cũng sẽ không muốn quay lại khi có nhu cầu tìm kiếm các thông tin liên quan khác. [[1](#Phạ05)] [[2](#Quy10)]

Về cách phân chia và tổ chức nội dung, trước khi website được thiết kế, nhóm đề tài đã định hình một số khung giao diện thường gặp và thiết kế chúng trở thành Skin. Sau đó phân chia các Control theo các Module và vị trí. Qua việc phân tích nhu cầu tìm đọc của người dùng, nhóm đề tài đã lựa chọn xây dựng các module chính, module tiện ích giúp người dùng có thể tìm đến bản tin một cách nhanh nhất.

* + 1. Nguyên tắc 2: Đảm bảo các liên kết được khám phá dễ dàng

Các đường liên kết là một điểm mạnh của website so với những tập thông tin bằng giấy. Do đó, nhóm đề tài đã tận dụng tối đa ưu điểm này để phân chia nội dung nhằm giúp người tiếp cận thông tin của bạn một cách dễ dàng nhất.

Một vấn đề đặt ra, tại sao trên các mạng xã hội, thông tin được truyền tải đi rất nhanh và rộng rãi. Bởi các nhà thiết kế đã tạo ra rất nhiều liên kết toàn bộ trên các thông tin của người dùng với bạn bè của họ. Tương tự như vậy, muốn người dùng khám phá nhiều thông tin trên website, nhóm đề tài đã phân chia nội dung và bố cục có những đường liên kết đến ngay các vị trí thích hợp cho người dùng.

….

* 1. Tổng quan về Angularjs
     1. Angluarjs là gì? Tại sao phải sử dụng nó?

Angular là một bộ Javascript Framework rất mạnh và thường được sử dụng để xây dựng project **Single Page Application** (SPA). Nó hoạt động dựa trên các thuộc tính mở rộng HTML (các atributes theo quy tắc của Angular). Đây là một Framework mã nguồn mở hoàn toàn miễn phí và được hàng ngàn các lập trình viên trên thế giới ưa chuộng và sử dụng. Framework này được thế hệ Web 2.0 phát triển khá mạnh ở nước ngoài, tuy nhiên ở Việt Nam thì vẫn chưa thông dụng lắm

* AngularJS là một Framework phát triển mạnh mẽ dựa trên JavaScript để tạo các ứng dụng RICH Internet Application (RIA).
* AngularJS cung cấp cho lập trình viên những tùy chọn để viết các ứng dụng client-side trong mô hình MVC (Model View Controller) một cách rõ ràng.
* Các ứng dụng được viết bởi AngularJS tương thích với nhiều phiên bản trình duyệt web. AngularJS tự động xử lý mã JavaScript để phù hợp với mỗi trình duyệt.
* AngularJS có mã nguồn mở, miễn phí hoàn toàn, được sử dụng bởi hàng ngàn lập trình viên trên thế giới. Nó hoạt động dưới giấy phép Apache License version 2.0

Nhìn chung, AngularJS là một framework để tạo các ứng dụng lớn, các hiệu năng cao trong khi giữ cho chúng có thể dễ dàng duy trì.

Dưới đây là các tính năng cốt lõi của AngularJS:

* **Data-binding:** Nó tự động đồng bộ hóa dữ liệu giữa thành phần *model* và *view*.
* **Scope:** Là những đối tượng hướng đến *model*, nó hoạt động như là cầu nối giữa *controller* và *view*.
* **Controller:** Đây là những tính năng của AngularJS mà được giới hạn tới một *scope* cụ thể.
* **Service:** AngularJS hoạt động với một vài dịch vụ (service) có sẵn , ví dụ *$http* để tạo *XMLHttpRequests*. Nó là các **singleton object** mà được khởi tạo duy nhất một lần trong ứng dụng.
* **Filter:** Nó lựa chọn (hay là lọc) các tập con từ tập item trong các mảng và trả về các mảng mới.
* **Directive:** Directive là các *marker* trong các phần tử DOM (như các phần tử, thuộc tính, css và nhiều hơn thế). Nó có thể dùng để tạo các thẻ HTML riêng phục vụ những mục đích riêng. AngularJS có những directive có sẵn như ngBind,ngModel…
* **Template:**Là các *rendered view* với các thông tin từ *controller* và *model*. Nó có thể được sử dụng trong các file riêng rẽ (ví dụ như index.jsp) hoặc nhiều view với một trang sử dụng "*partials*".
* **Routing:** Là khái niệm của sự chuyển dịch qua lại các *view*.
* **Model View Whatever:** MVC là một mô hình thiết kế để phân chia các ứng dụng thành nhiều phần khác nhau (gọi là Model, View và Controller), một phần sử dụng với một nhiệm vụ nhất định. AngularJS không triển khai MVC theo cách truyền thống, mà gắn liền hơn với *Model-View-ViewModel*. Nhóm phát triển AngularJS đã đặt tên vui cho mô hình này là *Model View Whatever*.
* **Deep Linking:** Cho phép bạn mã hóa trạng thái các ứng dụng trên địa chỉ URL để nó có thể được *bookmark*. Các ứng dụng có thể được phục hồi lại từ các địa chỉ URL với cùng một trạng thái.
* **Dependency Injection:** AngularJS có sẵn một hệ thống con *dependency injection* để giúp các lập trình viên tạo ra các ứng dụng dễ phát triển, dễ hiểu và kiểm tra.



Mặc dù AngularJS có thể kể đến rất nhiều các ưu điểm, nhưng đến thời điểm này, nó vẫn có một số điểm yếu sau:

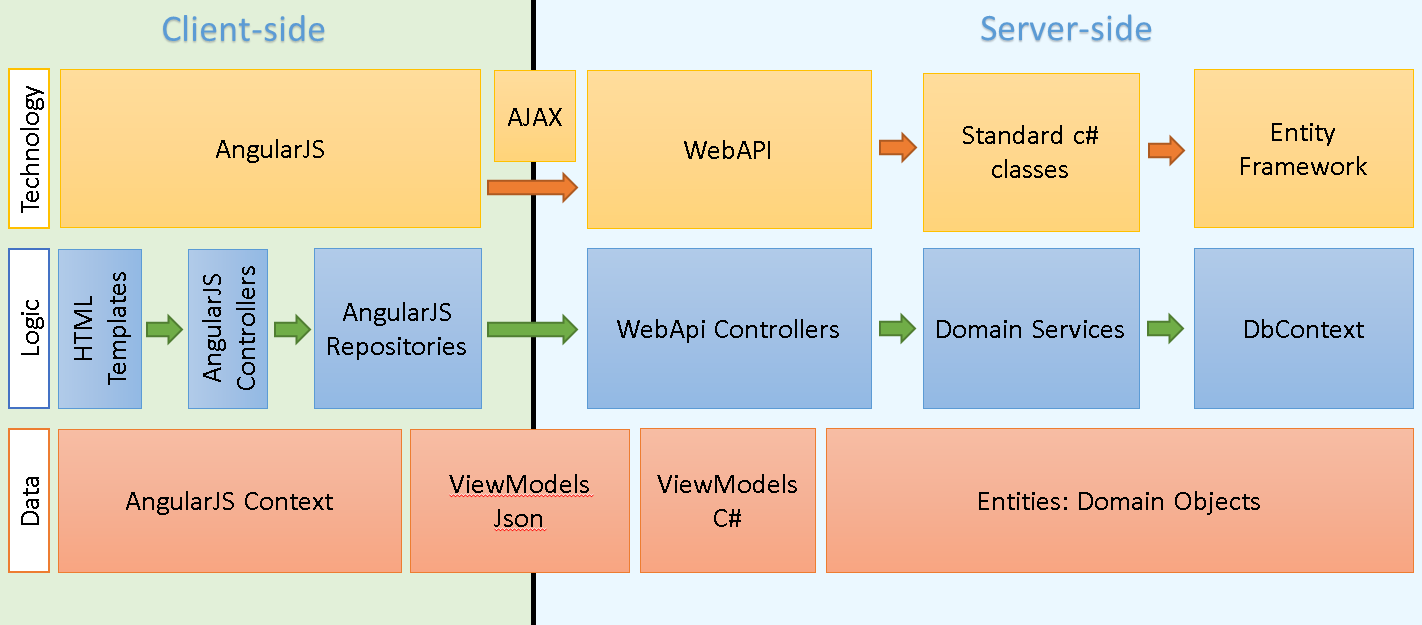
* **Không an toàn** : Là một JavaScript framework, ứng dụng được viết bởi AngularJS không an toàn. Phải có các tính năng bảo mật và xác thực phía server sẽ giúp ứng dụng trở nên an toàn hơn.
* Nếu người sử dụng ứng dụng của bạn vô hiệu hóa JavaScript thì họ chỉ nhìn được trang cơ bản, không thấy gì thêm.

# NỘI DUNG THỰC HIỆN

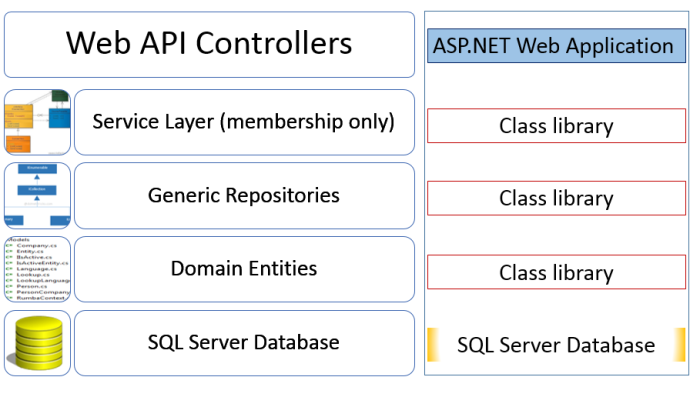
* 1. Phát biểu bài toán
     1. Tổng quan website cung cấp dịch vụ tìm kiếm nhà trọ

Website là một hệ thống được viết bằng ngôn ngữ C# trên nên ASP.NET của của Microsoft do tác giả đề xuất. Bởi vậy trong kiến trúc của Website có chứa một số thành phần APIs, công nghệ của Microsoft. Trong hai mô hình kiến trúc tổng thể và chi tiết dưới đây sẽ chỉ rõ cho chúng ta về các thành phần cấu thành nên hệ thống

* + 1. Kiến trúc tổng quan của hệ thống



Hình ‑: Mô hình kiến trúc tổng thể hệ thống



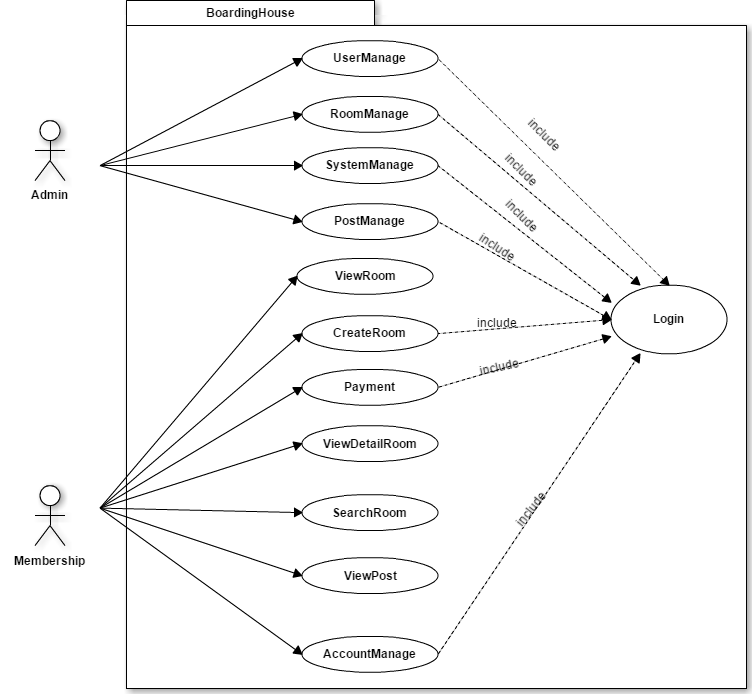
Hình ‑2: Mô hình kiến trúc của dự án phần Server side

Hệ thống trên ta có thể nhìn rõ cấu trúc rõ ràng rành mạnh được chia thành 3 tầng chính là Data,Logic,Technology.

Mỗi tầng của dự án sẽ có chức năng riêng biệt độc lập nên dễ dàng bảo trì và phát triển hệ thống

* 1. Phân Tích Hệ Thống
     1. Biểu Đồ UseCase

Biểu Đồ UseCase Tổng Quát



Hình ‑3: Mô hình UseCase tổng quát của hệ thống

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 website

Danh Sách Các Tác Nhân

|  |  |  |
| --- | --- | --- |
| Tác nhân | Mô tả tác nhân | Ghi chú |
| **Admin** | Admin là người quản trị hệ thống website như duyệt vài ,viết tin tức, quản lý các Membership. | Phải đăng nhập |
| **Membership** | Là người dùng vào hệ thống website.Người dùng có thể sử dụng các chức năng trên hệ thống mà website cung cấp |  |

Đặc tả UseCase

* **Đặc tả UseCase CreateRoom**
* Mục đích

Giúp người dùng có thể tạo thông tin nhà trọ,nhà ở,văn phòng của mình cần cho thuê hoặc bán

* Tác nhân liên quan

Người dùng(Membership) và hệ thống (Admin)

* Điều kiện trước

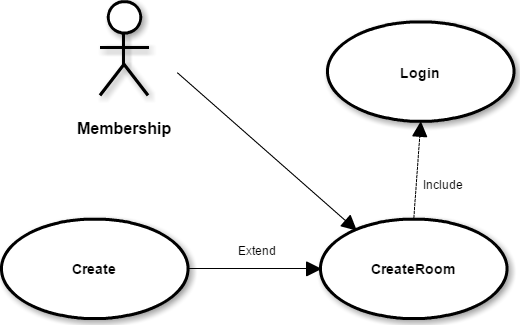
Người dùng phải đăng nhập

* Điều kiện sau

Nếu thành công thì hệ thống sẽ thông báo Bạn đã đăng tin thành công

Còn thất bại thì hệ thống sẽ thông báo ”Bạn đã đăng tin không thành công”

* Biểu đồ UseCase

****

* Dòng sự kiện chính

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng tạo thông tin nhà | 1. Nếu chưa đăng nhập thì hệ thống sẽ hiển thị giao diện đăng nhập.Ngược lại thì hệ thống sẽ hiển thị giao diện nhập thông tin nhà |
| 1. Thực hiện các chức năng thêm 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. |
| 1. Chọn nút Thêm thông tin | 1. Nếu thành công thì xuất thông báo “Bạn đã đăng tin thành công”.Ngược lại xuất thông báo “Bạn đăng tin không thành công” |

* **Đặc tả UseCase SearchRoom**
* Mục đích

Giúp người dùng có thể tìm kiếm thông tin mà mình mong muốn trên hệ thống của website

* Tác nhân liên quan

Người dùng(Membership)

* Điều kiện sau

Nếu thành công thì hệ thống đưa ra các kết quả tìm kiếm của bạn

Còn thất bại thì hệ thống sẽ thông báo ”Không có thông tin nào được tìm thấy”

* Biểu đồ UseCase

****

* Dòng sự kiện chính

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng tìm kiếm thông tin | 2. Chọn các option trong giao diện bộ lọc tìm kiếm |
| 3. Nhấn nút tìm kiếm | 4 . Hiển thị giao diện kết quả tìm kiếm được ra màn hình |

* **Đặc tả UseCase ViewRoom**
* Mục đích

Giúp người dùng có thể xem thông tin mà mình chọn trên hệ thống của website

* Tác nhân liên quan

Người dùng(Membership)

* Điều kiện sau
* Biểu đồ UseCase

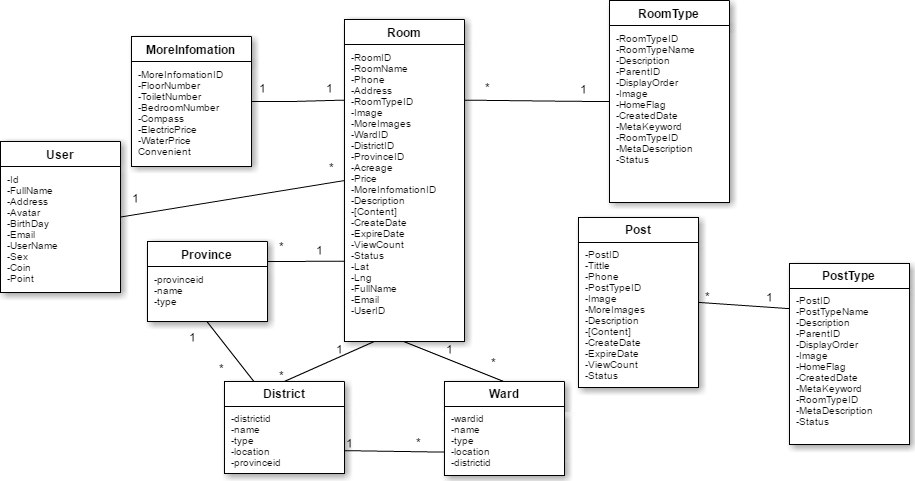
****

* Dòng sự kiện chính

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn một thông tin mà mình mong muốn | 2. Hiển thị giao diện thông tin đó trên màn mình |
| 3. Xem thông tin bà nhấn nút hiển thị số điện thoại | 4 . Hiển thị số điện thoại của người đăng tin |

* + 1. Biểu Đồ Lớp

Biểu Đồ Lớp Tổng quát



Hình ‑2: Mô hình biểu đồ lớp

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

Bảng ‑: Class Room

| # | Mã trường | Kiểu trường | Độ dài | Null | Unique | PK/ FK | Mặc định | Mô tả |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RoomID | Int |  |  |  | PK |  | Mã Phòng |
|  | RoomName | Nvarchar | 250 |  |  |  |  | Tên của phòng |
|  | Phone | Nvarchar | 20 |  |  |  |  | Số điện thoại |
|  | Address | Nvarchar | 250 |  |  |  |  | Địa chỉ |
|  | RoomTypeID | Int |  |  |  | FK |  | Ngày tạo |
|  | Image | Nvarchar | 250 |  |  |  |  | Ảnh |
|  | MoreImages | XML |  |  |  |  |  | Nhiều ảnh |
|  | WardID | Int |  |  |  | FK |  | Mã xã |
|  | DistrictID | Int |  |  |  | FK |  | Mã Huyện |
|  | ProvinceID | Int |  |  |  | FK |  | Mã Tỉnh |
|  | Acreage | Float |  |  |  |  |  | Diện tích |
|  | Price | Float |  |  |  |  |  | Gía tiền |
|  | MoreInfomationID | Int |  |  |  | FK |  | Mã thông tin thêm |
|  | Description | Nvarchar | 500 |  |  |  |  | Ghi chú |
|  | [Content] | Ntext |  |  |  |  |  | Mô tả |
|  | Lat | Float |  |  |  |  |  | Kinh độ |
|  | Lng | Float |  |  |  |  |  | Vĩ độ |
|  | FullName | Nvarchar | 250 |  |  |  |  | Họ và tên |
|  | Email | Nvarchar | 250 |  |  |  |  | Email |
|  | UserID | Nvarchar | 250 |  |  | FK |  | Mã người dùng |

Bảng ‑2: Class RoomType

| # | Mã trường | Kiểu trường | Độ dài | Null | Unique | PK/ FK | Mặc định | Mô tả |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RoomTypeID | Int |  |  |  | PK |  | Mã loại Phòng |
|  | RoomTypeName | Nvarchar | 250 |  |  |  |  | Tên loại phòng |
|  | Description | Nvarchar | 20 |  |  |  |  | Ghi chú |
|  | ParentID | Int |  |  |  |  |  | Mã cha |
|  | DisplayOrder | Int |  |  |  |  |  | Thứ tự |
|  | Image | Nvarchar | 250 |  |  |  |  | Ảnh |
|  | HomeFlag | Bit |  |  |  |  |  | Hiển thị trang chủ |
|  | CreatedDate | Datetime |  |  |  |  |  | Ngày tạo |
|  | MetaKeyword | Nvarchar |  |  |  |  |  | Thẻ từ khóa |
|  | MetaDescription | Nvarchar |  |  |  |  |  | Nội dung từ khóa |
|  | Status | Bit |  |  |  |  |  | Trạng thái |

Bảng ‑3: Class MoreInfomation

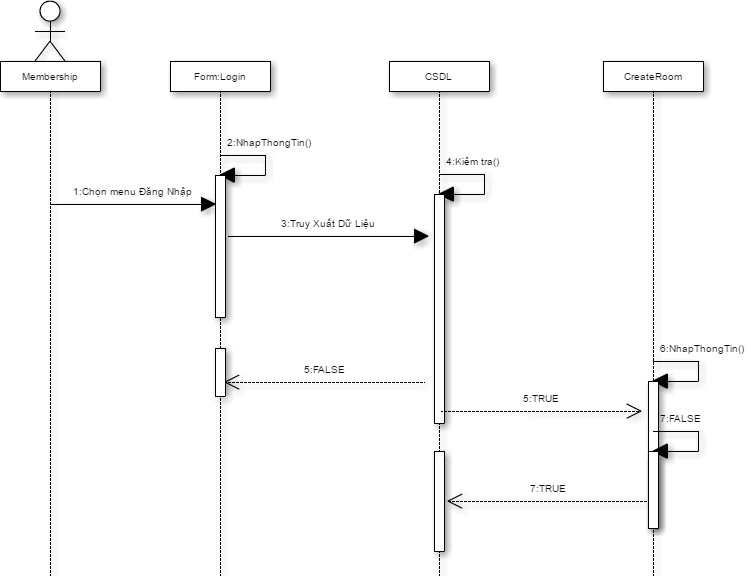
| # | Mã trường | Kiểu trường | Độ dài | Null | Unique | PK/ FK | Mặc định | Mô tả |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | MoreInfomationID | Int |  |  |  | PK |  | Mã thông tin thêm |
|  | FloorNumber | Nvarchar | 250 |  |  |  |  | Số tầng |
|  | ToiletNumber | Nvarchar | 250 |  |  |  |  | Số nhà vệ sinh |
|  | BedroomNumber | Nvarchar | 250 |  |  |  |  | Số phòng ngủ |
|  | Compass | Nvarchar | 250 |  |  |  |  | Hướng nhà |
|  | ElectricPrice | int |  |  |  |  |  | Gía điện |
|  | WaterPrice | int |  |  |  |  |  | Gía nước |
|  | Convenient | Xml |  |  |  |  |  | Tiện ích |

Bảng ‑4: Class User

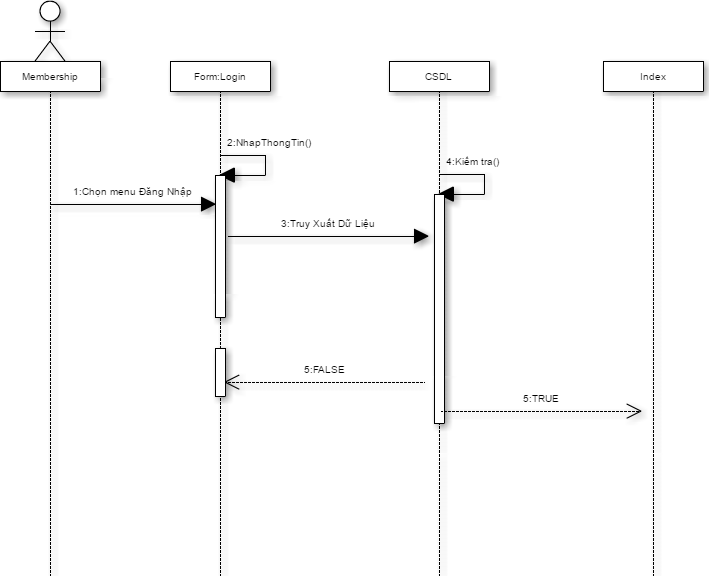
| # | Mã trường | Kiểu trường | Độ dài | Null | Unique | PK/ FK | Mặc định | Mô tả |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Id | Nvarchar | 250 |  |  | PK |  | Mã người dùng |
|  | FullName | Nvarchar | 250 |  |  |  |  | Họ và tên |
|  | Address | Nvarchar | 250 |  |  |  |  | Địa chỉ |
|  | Avatar | Nvarchar | 250 |  |  |  |  | Anh đại diện |
|  | BirthDay | Nvarchar | 250 |  |  |  |  | Sinh nhật |
|  | Email | Nvarchar | 250 |  |  |  |  | Email |
|  | UserName | Nvarchar | 250 |  |  |  |  | Tên tài khoản |
|  | Sex | Nvarchar | 10 |  |  |  |  | Giới tính |
|  | Coin | int |  |  |  |  |  | Xu |
|  | Point | int |  |  |  |  |  | Điểm |

* + 1. Biểu Đồ Tuần Tự

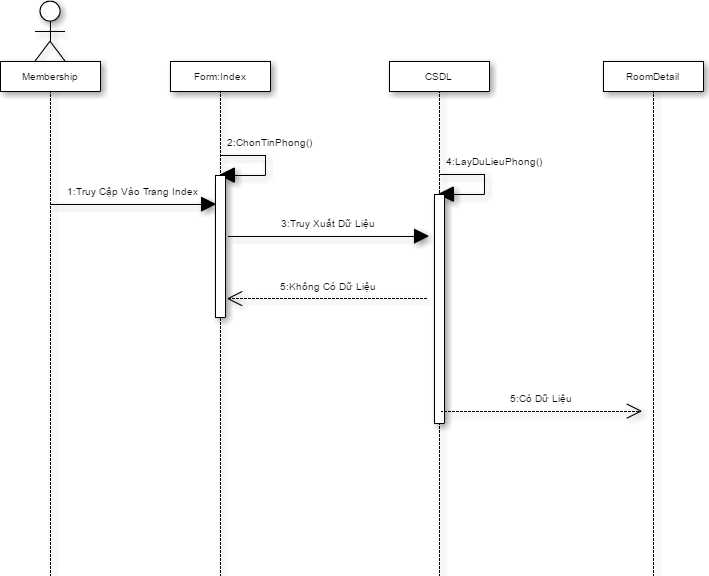
Biểu Đồ Tuần Tự CreateRoom



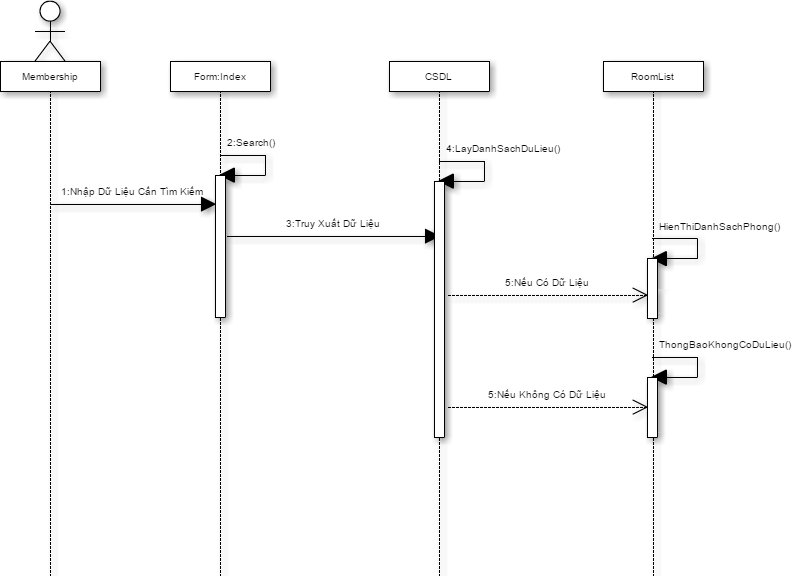
Biểu Đồ Tuần Tự Login



Biểu Đồ Tuần Tự ViewRoom

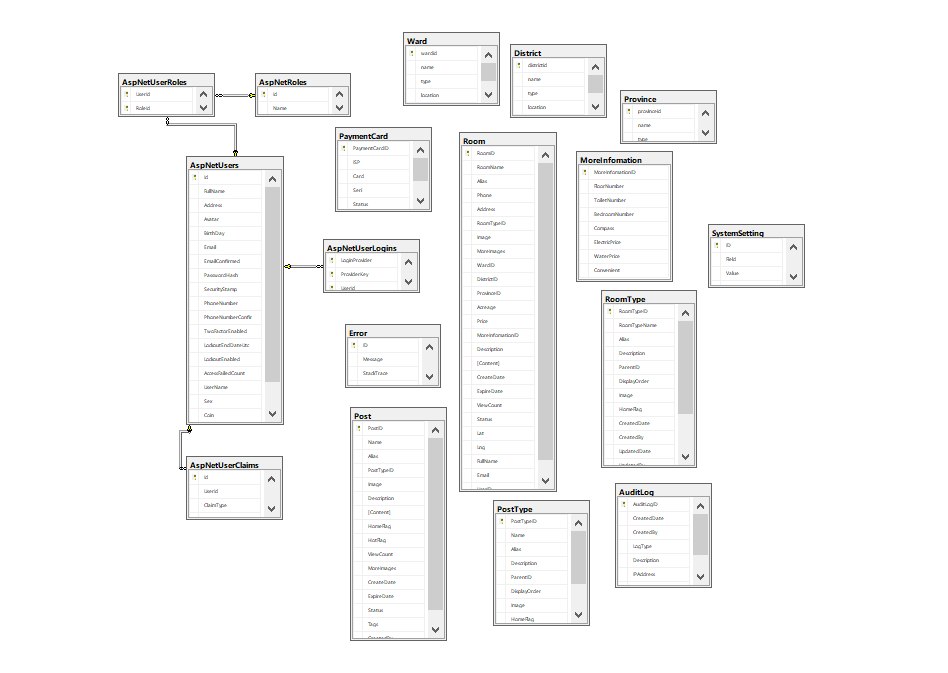


Biểu Đồ Tuần Tự Search



* 1. Thiết Kế Hệ Thống
     1. Thiết Kế Cơ Sở Dữ Liệu

Biểu đồ cơ sở dữ liệu của hệ thống



Hình ‑: Mô hình lưu trữ cấu trúc website trên Database

Các thuộc tính của các bảng cơ sở dữ liệu

Bảng ‑5: Bảng Room

| # | Mã trường | Kiểu trường | Độ dài | Null | Unique | PK/ FK | Mặc định | Mô tả |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RoomID | Int |  |  |  | PK |  | Mã Phòng |
|  | RoomName | Nvarchar | 250 |  |  |  |  | Tên của phòng |
|  | Phone | Nvarchar | 20 |  |  |  |  | Số điện thoại |
|  | Address | Nvarchar | 250 |  |  |  |  | Địa chỉ |
|  | RoomTypeID | Int |  |  |  | FK |  | Ngày tạo |
|  | Image | Nvarchar | 250 |  |  |  |  | Ảnh |
|  | MoreImages | XML |  |  |  |  |  | Nhiều ảnh |
|  | WardID | Int |  |  |  | FK |  | Mã xã |
|  | DistrictID | Int |  |  |  | FK |  | Mã Huyện |
|  | ProvinceID | Int |  |  |  | FK |  | Mã Tỉnh |
|  | Acreage | Float |  |  |  |  |  | Diện tích |
|  | Price | Float |  |  |  |  |  | Gía tiền |
|  | MoreInfomationID | Int |  |  |  | FK |  | Mã thông tin thêm |
|  | Description | Nvarchar | 500 |  |  |  |  | Ghi chú |
|  | [Content] | Ntext |  |  |  |  |  | Mô tả |
|  | Lat | Float |  |  |  |  |  | Kinh độ |
|  | Lng | Float |  |  |  |  |  | Vĩ độ |
|  | FullName | Nvarchar | 250 |  |  |  |  | Họ và tên |
|  | Email | Nvarchar | 250 |  |  |  |  | Email |
|  | UserID | Nvarchar | 250 |  |  | FK |  | Mã người dùng |

Bảng ‑6: Bảng RoomType

| # | Mã trường | Kiểu trường | Độ dài | Null | Unique | PK/ FK | Mặc định | Mô tả |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RoomTypeID | Int |  |  |  | PK |  | Mã loại Phòng |
|  | RoomTypeName | Nvarchar | 250 |  |  |  |  | Tên loại phòng |
|  | Description | Nvarchar | 20 |  |  |  |  | Ghi chú |
|  | ParentID | Int |  |  |  |  |  | Mã cha |
|  | DisplayOrder | Int |  |  |  |  |  | Thứ tự |
|  | Image | Nvarchar | 250 |  |  |  |  | Ảnh |
|  | HomeFlag | Bit |  |  |  |  |  | Hiển thị trang chủ |
|  | CreatedDate | Datetime |  |  |  |  |  | Ngày tạo |
|  | MetaKeyword | Nvarchar |  |  |  |  |  | Thẻ từ khóa |
|  | MetaDescription | Nvarchar |  |  |  |  |  | Nội dung từ khóa |
|  | Status | Bit |  |  |  |  |  | Trạng thái |

Bảng ‑7: Bảng MoreInfomation

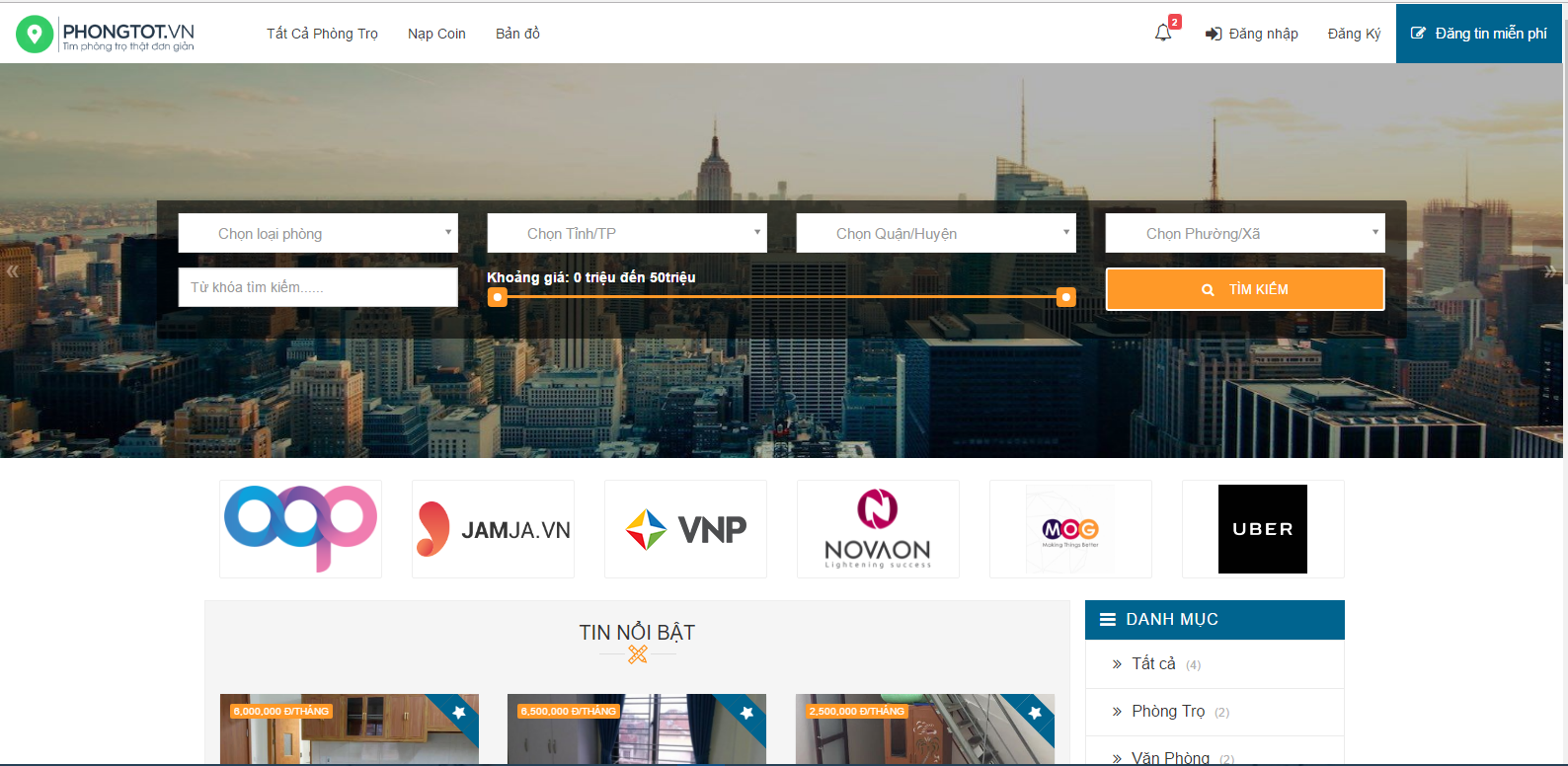
| # | Mã trường | Kiểu trường | Độ dài | Null | Unique | PK/ FK | Mặc định | Mô tả |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | MoreInfomationID | Int |  |  |  | PK |  | Mã thông tin thêm |
|  | FloorNumber | Nvarchar | 250 |  |  |  |  | Số tầng |
|  | ToiletNumber | Nvarchar | 250 |  |  |  |  | Số nhà vệ sinh |
|  | BedroomNumber | Nvarchar | 250 |  |  |  |  | Số phòng ngủ |
|  | Compass | Nvarchar | 250 |  |  |  |  | Hướng nhà |
|  | ElectricPrice | int |  |  |  |  |  | Gía điện |
|  | WaterPrice | int |  |  |  |  |  | Gía nước |
|  | Convenient | Xml |  |  |  |  |  | Tiện ích |

Bảng ‑8: Bảng User

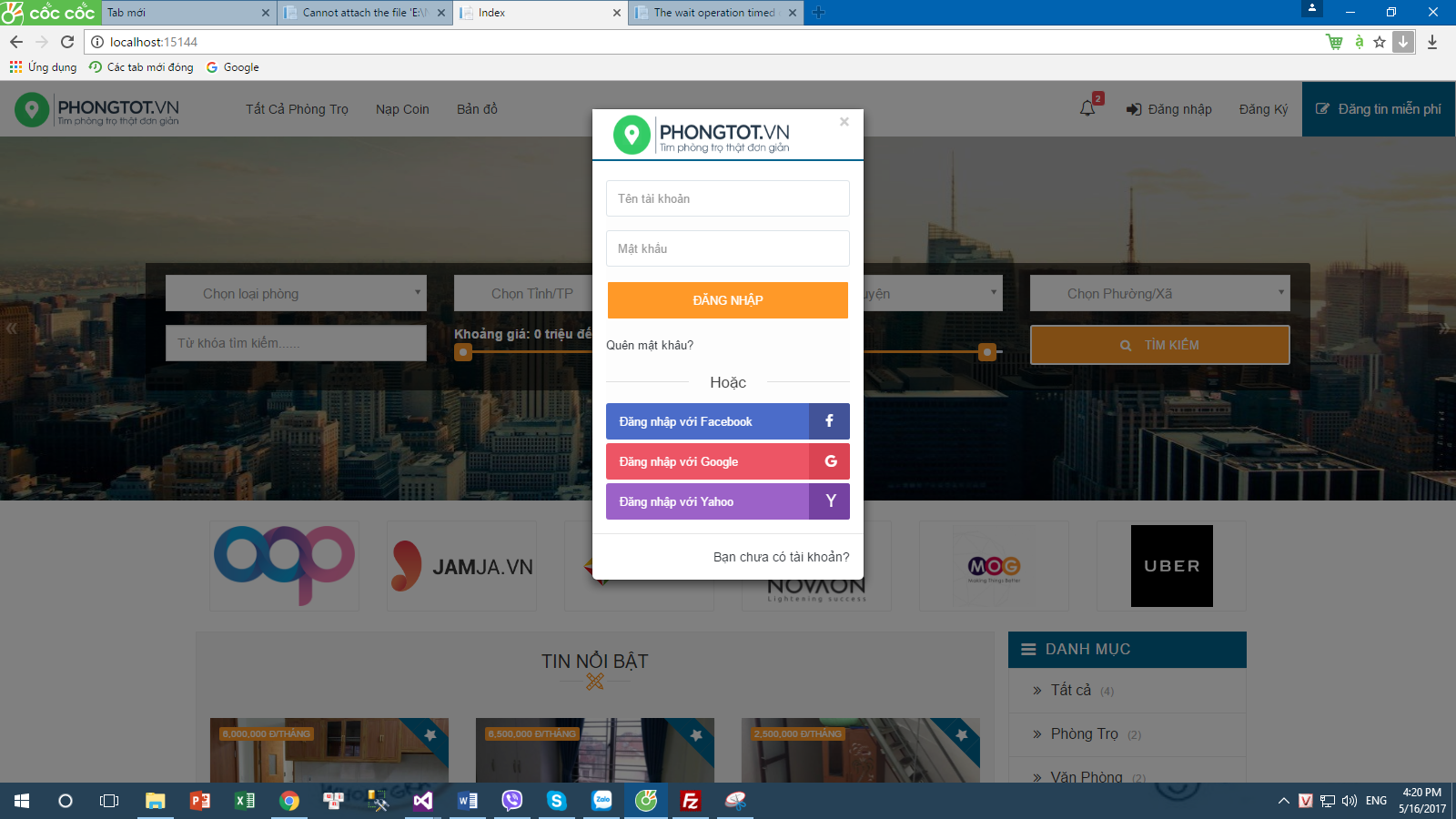
| # | Mã trường | Kiểu trường | Độ dài | Null | Unique | PK/ FK | Mặc định | Mô tả |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Id | Nvarchar | 250 |  |  | PK |  | Mã người dùng |
|  | FullName | Nvarchar | 250 |  |  |  |  | Họ và tên |
|  | Address | Nvarchar | 250 |  |  |  |  | Địa chỉ |
|  | Avatar | Nvarchar | 250 |  |  |  |  | Anh đại diện |
|  | BirthDay | Nvarchar | 250 |  |  |  |  | Sinh nhật |
|  | Email | Nvarchar | 250 |  |  |  |  | Email |
|  | UserName | Nvarchar | 250 |  |  |  |  | Tên tài khoản |
|  | Sex | Nvarchar | 10 |  |  |  |  | Giới tính |
|  | Coin | int |  |  |  |  |  | Xu |
|  | Point | int |  |  |  |  |  | Điểm |

* + 1. Thiết kế giao diện

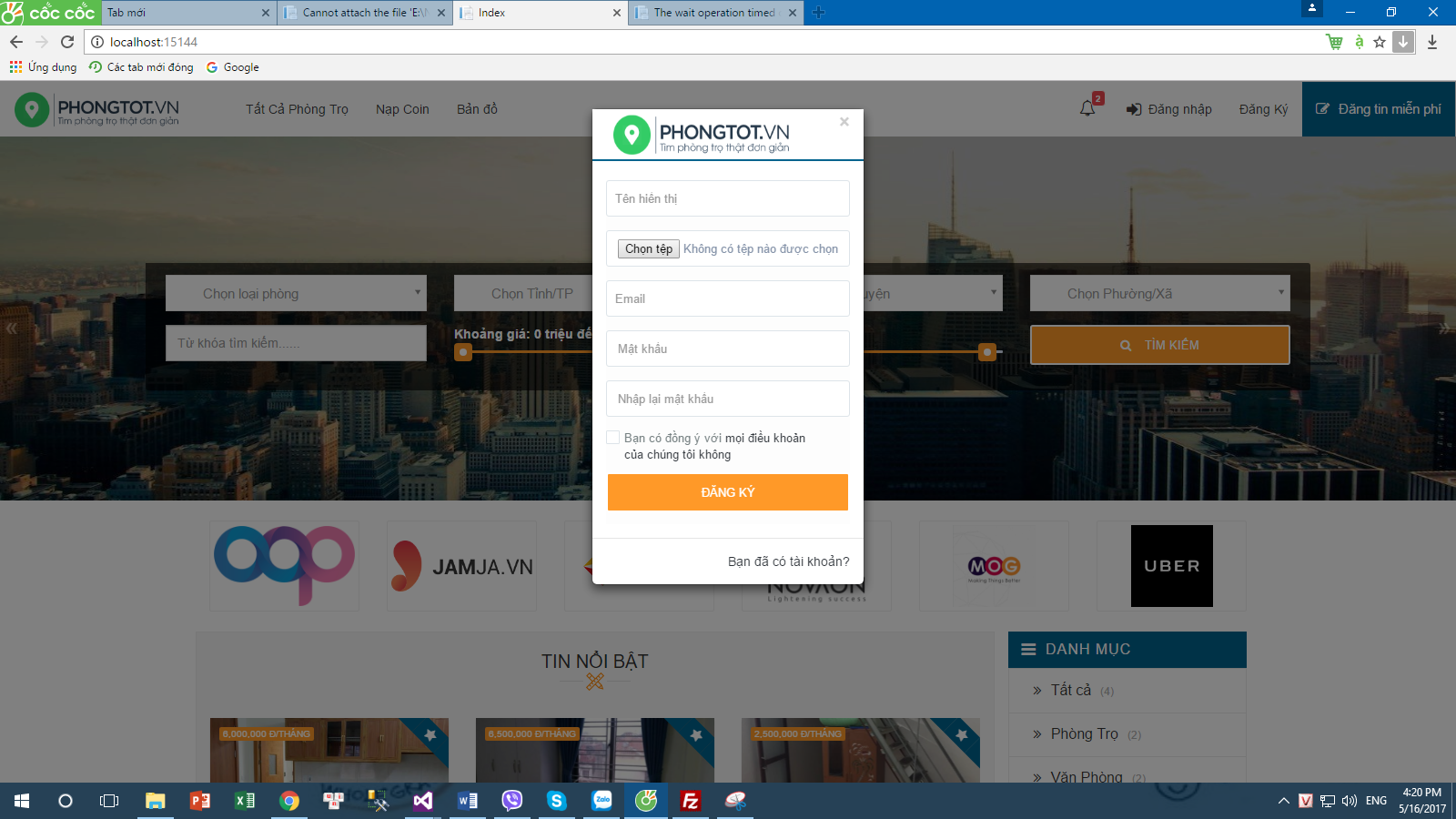
Giao diện chính của website



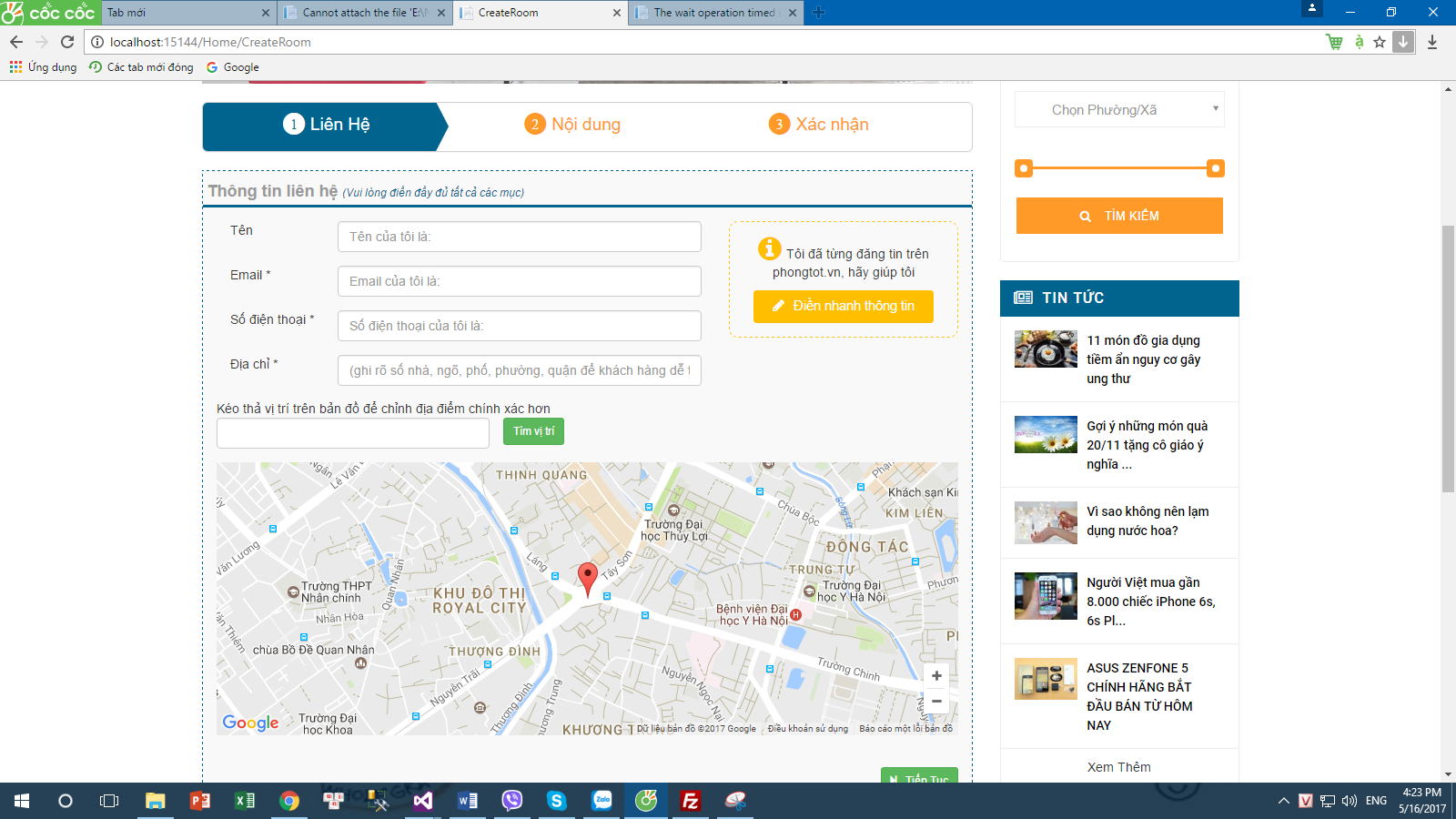
Giao diện Đăng Nhập

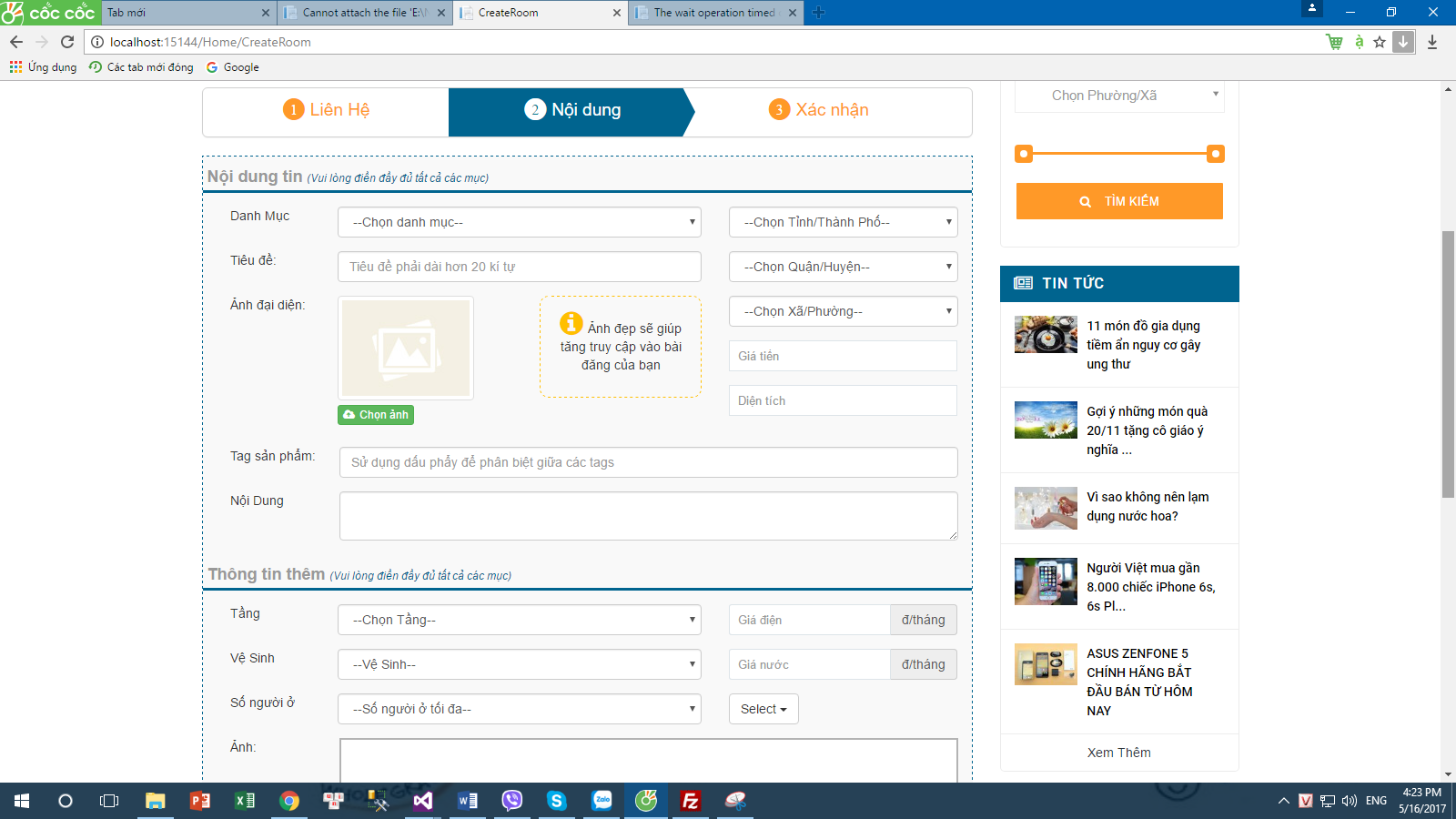


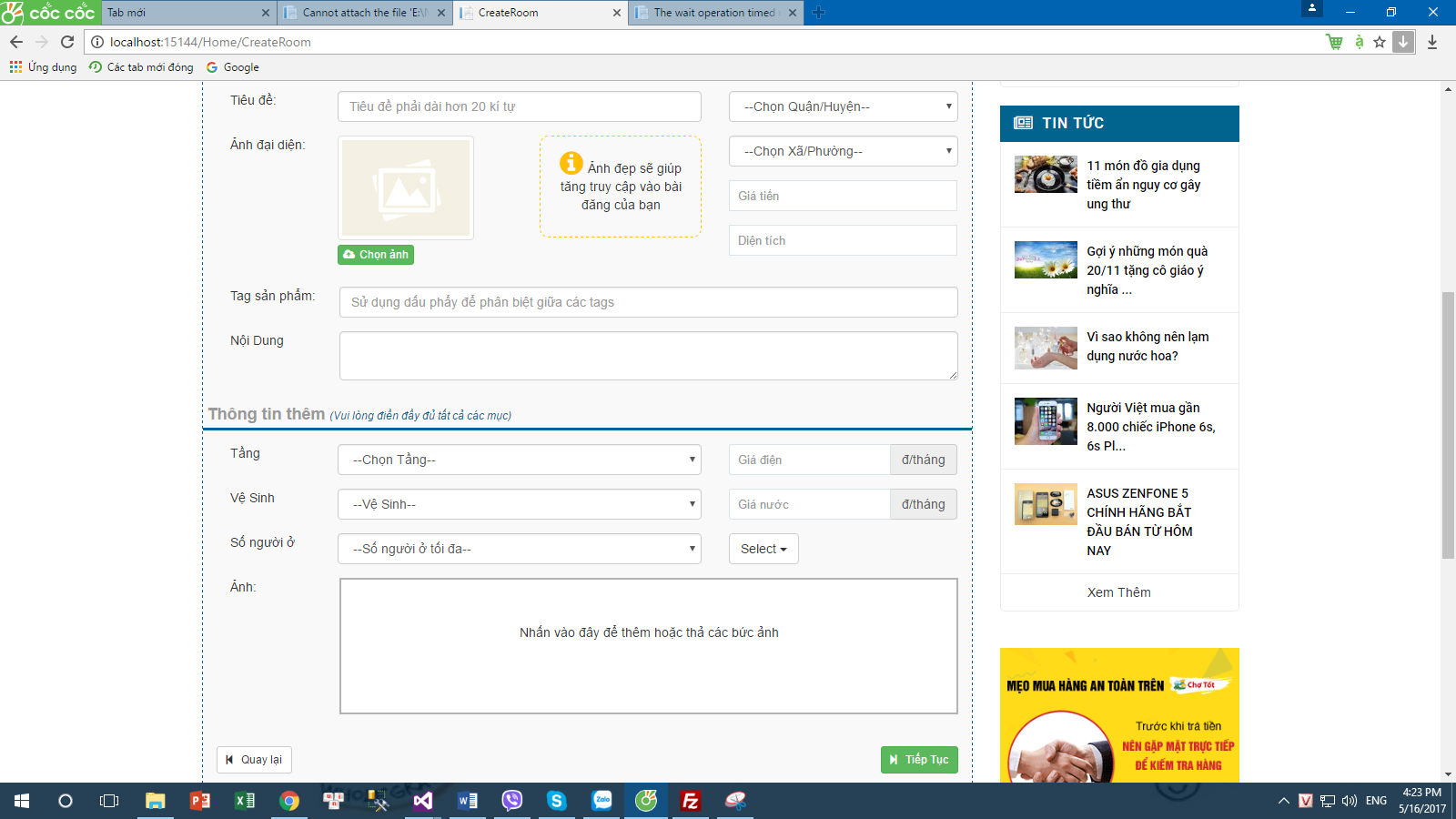
Giao diện Đăng Ký



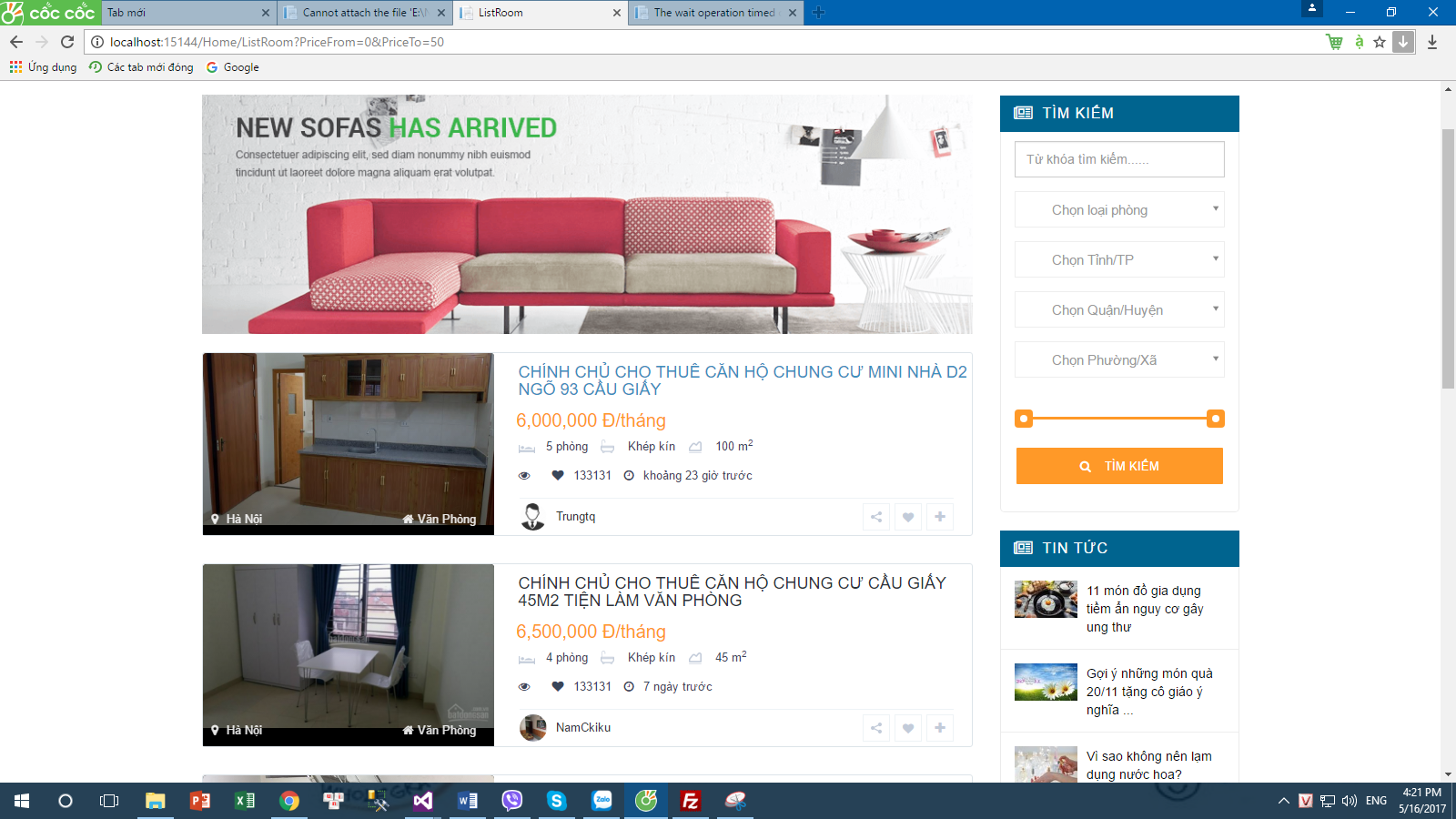
Giao diện thêm thông tin Phòng



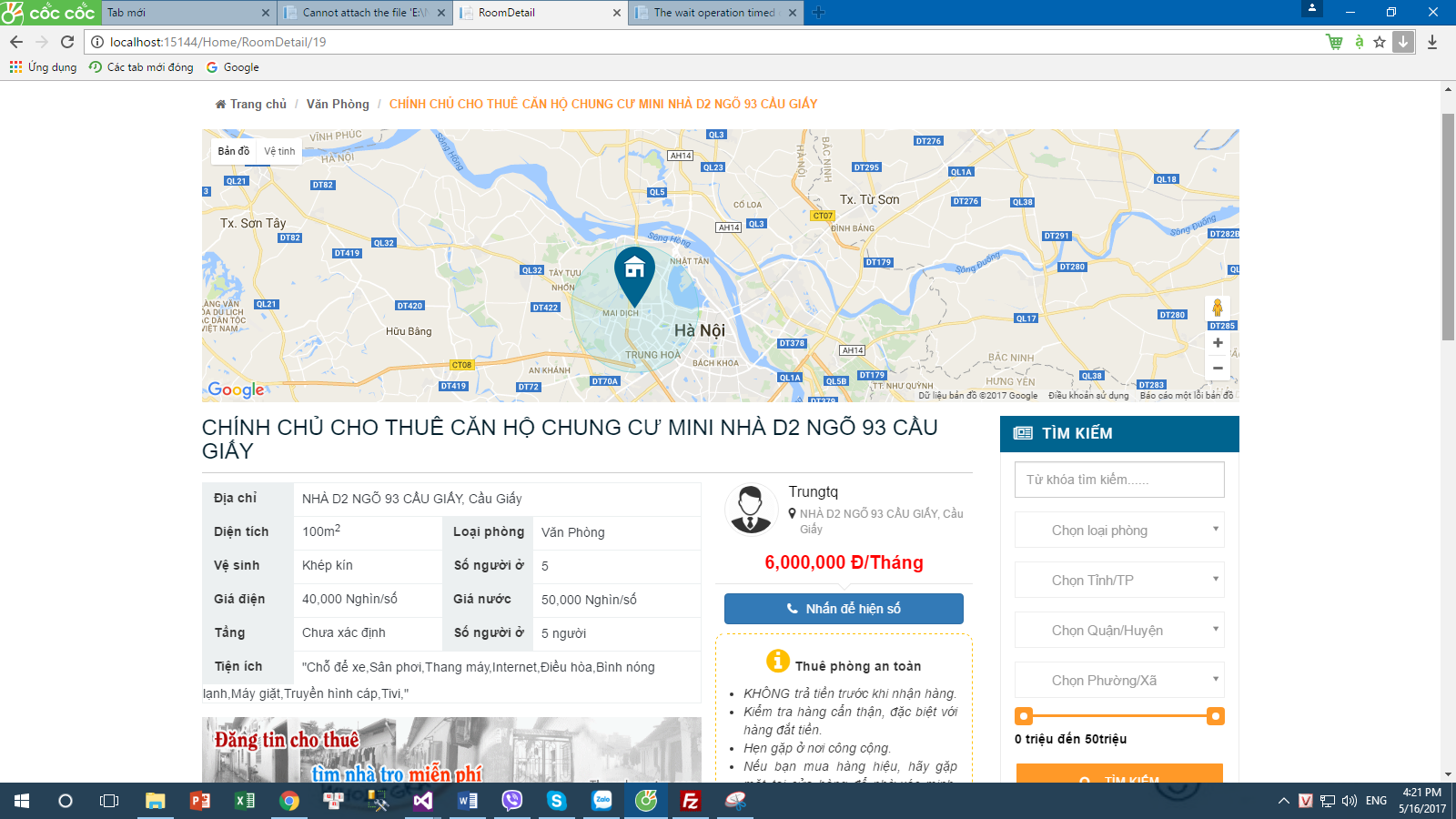


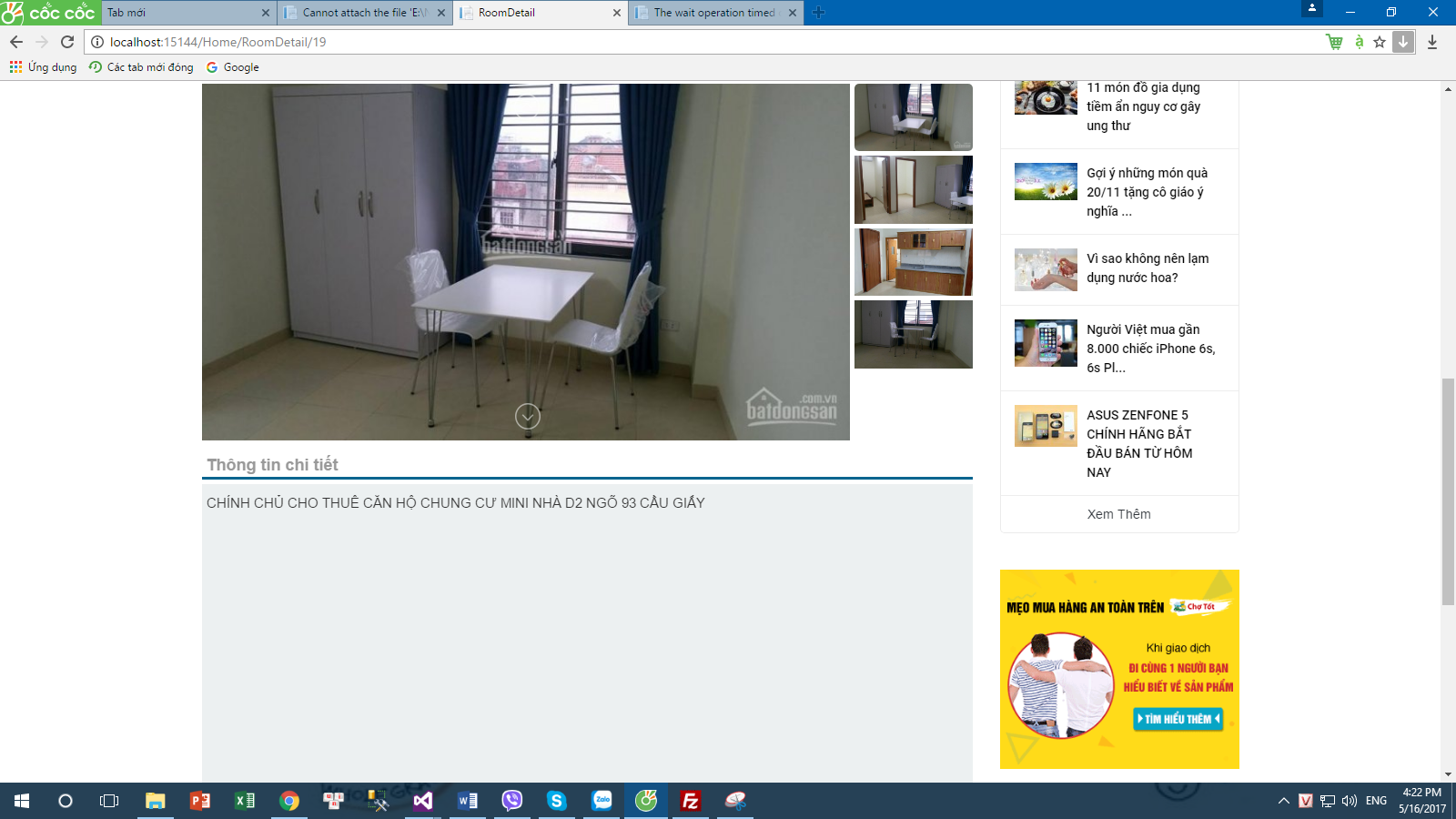


Giao diện kết quả tìm kiếm

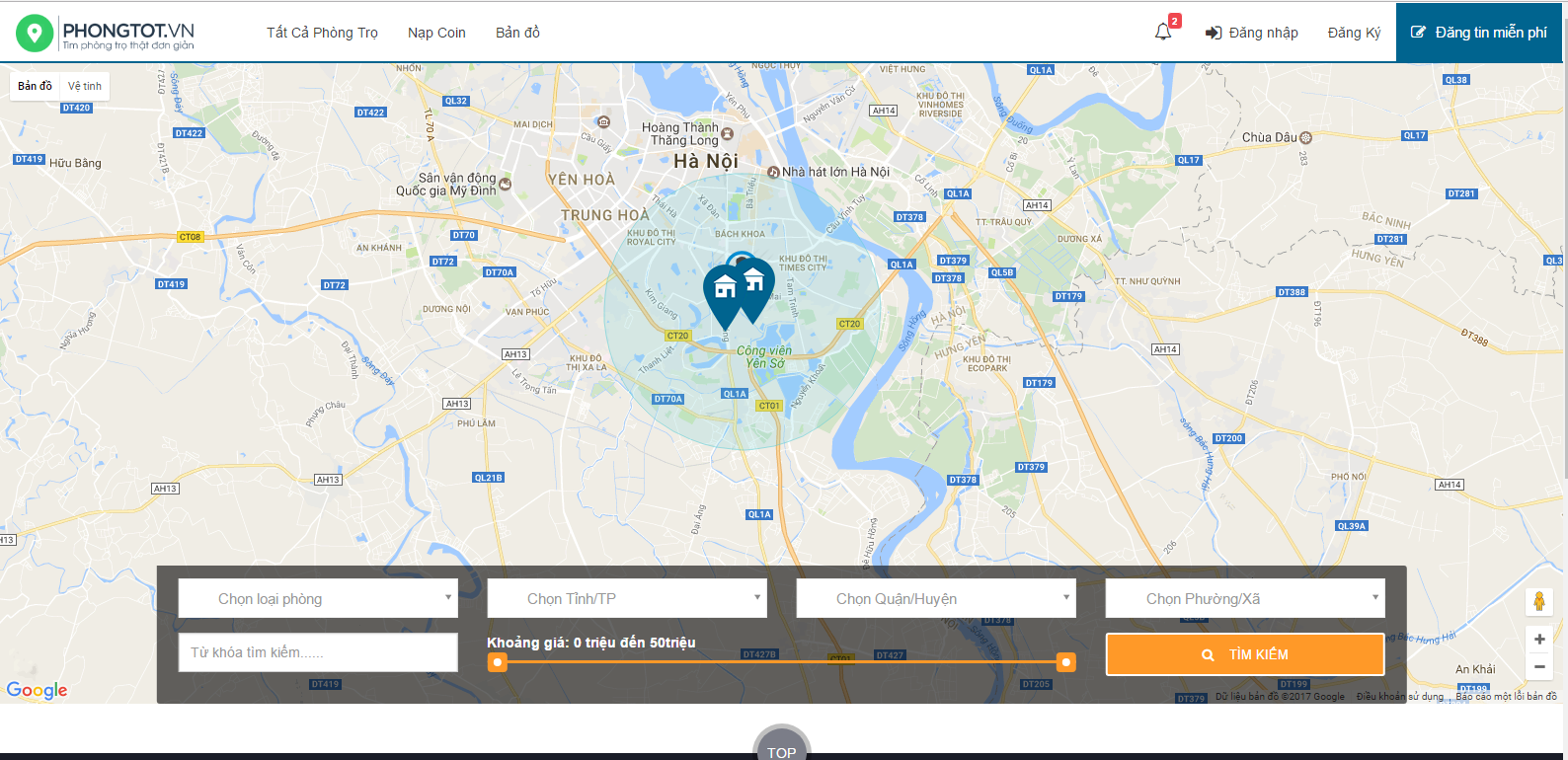


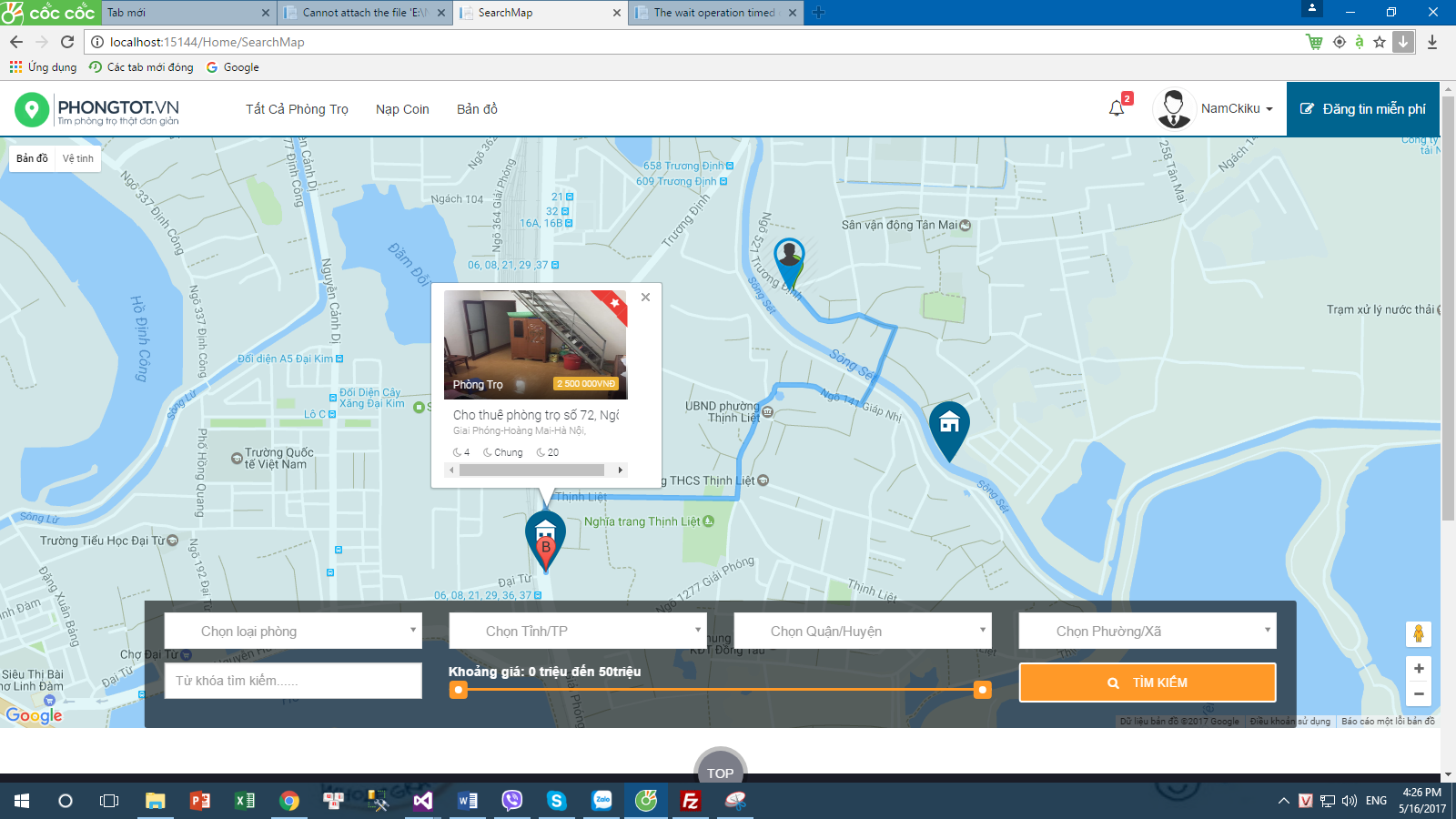
Giao diện thông tin chi tiết phòng





Giao diện trang tìm kiếm bản đồ





* 1. Cài Đặt Và Triển Khai Hệ Thống

# 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. |
| [2] | Nguyễn Văn Quyết, *Giáo trình Công nghệ Web và ứng dụng*. Hưng Yên, Việt Nam: Khoa Công nghệ Thông tin - UTEHY, 2010. |

x