**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

# **TRƯỜNG ĐẠI HỌC KHOA HỌC – ĐẠI HỌC HUẾ**

KHOA CÔNG NGHỆ THÔNG TIN

🙞✪🙜



**TÁC GIẢ: MAI VĂN LONG**

**19T1021124**

**DỰ ÁN WEBSITE QUẢN LÝ KHÁCH SẠN**

**JADE DRAGON**

**KHÓA LUẬN TỐT NGHIỆP**

**CHUYÊN NGÀNH CÔNG NGHỆ PHẦN MỀM**

*GIẢNG VIÊN HƯỚNG DẪN: THẦY NGUYỄN HOÀNG HÀ*

**TP.HUẾ – NĂM 2023**

Phần mở đầu

***Lý do chọn đề tài:*** Đời sống con người ngày càng càng phát triển, dẫn dến nhu cầu thỏa mãn, tận hưởng cho bản thân ngày càng cao. Theo đó là sự phát triển mạnh mẽ của các ngành du lịch, đặc biệt là về lĩnh vực khách sạn, khu nghỉ dưỡng.

Nhận thấy được sự phát triển bừa bộn của nhiều khách sạn cũng như các khu nghĩ dưỡng liên tục mọc lên một cách rải rác khắp vùng miền. Việc này ảnh hưởng rất lớn đến việc kinh doanh của các doanh nhân quản lý khách sạn cũng như mang lại nhiều rắc rối cho người dân do thiếu sự tổng hợp và tương tác giữa đôi bên.   
Cùng với mô hình MVC giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp và bảo trì. Do đó tôi đã chọn đề tài thiết kế ra một trang website quản lý khách sạn bằng C#, cụ thể là mô hình MVC 5. Với đề tài này, tôi mong muốn người dân có thể dễ dàng tiếp cận, tham khảo cũng như tìm hiểu trước nơi mình muốn đặt chân đến nghỉ ngơi bằng cách sử dụng internet, bên cạnh đó là giúp cho những doanh nhân làm về lĩnh vực này dễ dàng kết nối với khách hàng và thẩm định thị trường của chính mình.

***Mục tiêu:*** Wed quản lý khách sạn thực chất là một khối nghiệp vụ phức tạp, đòi hỏi phải có được một hệ thống quản lý chặt chẽ, nghiệp vụ cao. Đề tài này vừa mang giá trị thực tế vừa mang lại giá trị tri thức cũng như kinh nghiệm cho tôi trong việc tiếp cận với ngành công nghệ thông tin. Vì sau khi thực hiện đề tài này, tôi đã được bồi dưỡng rất nhiều về hệ thống các kiến thức đã từng học và nhiều ứng dụng công nghệ mới phù hợp với thực tế. Bên cạnh đó là trang bị thêm kiến thức về nghiệp vụ quản lý trong hệ thống của một công ty, tổ chức và hơn hết là nâng cao trình độ chuyên môn của bản thân.

Trong khuôn khổ đề tài này, tôi chủ yếu chú ý đến việc tiếp cận và kết nối của các doanh nhân với khách hàng và khách hàng với các doanh nhân nên tôi đã thực hiện hệ thống với các nhiệm vụ cơ bản như:

Bên cạnh đó là lồng ghép các công nghệ mới để tăng cường các yếu tố bảo mật cũng như sự thuận tiện trong khi làm việc với hệ thống

Mục lục

[Phần mở đầu 1](#_Toc134669107)

[Chương 1, Tổng quan về ASP.NET 3](#_Toc134669108)

[1. Giới thiệu 3](#_Toc134669109)

[2. Các thành phần trong ASP.NET 5](#_Toc134669110)

[3. Mô hình MVC 6](#_Toc134669111)

[3.1 Giới thiệu về MVC 7](#_Toc134669112)

[3.2. Ưu và nhược điểm MVC 8](#_Toc134669113)

[3.2.1. Ưu điểm: 8](#_Toc134669114)

[3.2.2. Nhược điểm: 9](#_Toc134669115)

[3.3. Vì sao nên sử dụng mô hình MVC 9](#_Toc134669116)

[3.3.1. Quy trình phát triển nhanh hơn 9](#_Toc134669117)

[3.3.2. Khả năng cung cấp nhiều chế độ View 9](#_Toc134669118)

[3.3.3. Các sửa đổi không ảnh hưởng đến toàn bộ mô hình 10](#_Toc134669119)

[3.3.4. MVC Model trả về dữ liệu mà không cần định dạng 10](#_Toc134669120)

[4. Các bước để xây dựng Website bằng MVC 5 10](#_Toc134669121)

[Chương 2, Xây dựng Website quản lý khách sạn 11](#_Toc134669122)

[1. Phát biểu bài toán 11](#_Toc134669123)

[2. Phân tích thiết kế và cài đặt 11](#_Toc134669124)

[2.2. Phân tích 11](#_Toc134669125)

[2.3. Thiết kế 11](#_Toc134669126)

[2.4. Cài đặt 11](#_Toc134669127)

[3. Hướng dẫn sử dụng 11](#_Toc134669128)

[Phần Kết luận 13](#_Toc134669129)

Chương 1, Tổng quan về ASP.NET

1. Giới thiệu

* Trên thế giới phát triển ứng dụng và web hiện nay, có rất nhiều framework và ngôn ngữ được sử dụng bởi các nhà phát triển. Mỗi framework đều có những tính năng và lợi ích riêng, và trong số đó, *ASP.NET* đã trở thành một trong những nền tảng mã nguồn mở phổ biến và được ưa chuộng bởi các nhà phát triển vì những ưu điểm vượt trội của nó.
* *ASP.NET* là một framework phát triển ứng dụng web được phát triển bởi Microsoft. *ASP.NET* được xây dựng trên nền tảng *.NET Framework*, cho phép các nhà phát triển tạo ra các ứng dụng web động và tương tác mạnh mẽ với các nguồn dữ liệu.
* Phiên bản *ASP.NET* đầu tiên được triển khai là 1.0 và phiên bản *ASP.NET* mới nhất là phiên bản 4.6. *ASP.NET* được thiết kế để tương thích với giao thức *HTTP. HTTP* là giao thức chuẩn được sử dụng trên tất cả các ứng dụng web. cung cấp cho các nhà phát triển các công cụ mạnh mẽ để tạo ra các ứng dụng web chất lượng cao, đáp ứng được các yêu cầu phức tạp của doanh nghiệp. Nó hỗ trợ nhiều ngôn ngữ lập trình như *C#, VB.NET, F#* và các công nghệ như *HTML, CSS* và *JavaScript* để xây dựng các ứng dụng web đáp ứng và thân thiện với người dùng.
* *ASP.NET* cũng cung cấp nhiều tính năng để giúp các nhà phát triển xây dựng các ứng dụng web bảo mật và đáng tin cậy. Nó có thể tích hợp với các công nghệ bảo mật như *OAuth, OpenID và SSL* để đảm bảo an toàn cho người dùng và dữ liệu.
* Ngoài ra, *ASP.NET* còn có khả năng xử lý tải cao, giúp các ứng dụng web hoạt động mượt mà và hiệu quả hơn, đáp ứng được nhu cầu của các doanh nghiệp với lượng truy cập lớn.

Với các tính năng mạnh mẽ và khả năng tích hợp tốt với các công nghệ khác, *ASP.NET* là một lựa chọn lý tưởng cho các nhà phát triển ứng dụng web mong muốn xây dựng các ứng dụng chất lượng cao và đáp ứng nhu cầu của khách hàng.

ASP viết đầy đủ là ***Active Server Pages***, và .NET là viết tắt của ***Network Enabled Technologies****.*

1. Các thành phần trong ASP.NET

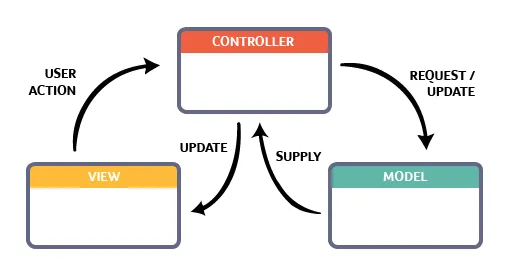
**ASP.NET** là một **framework** phát triển ứng dụng web được phát triển bởi Microsoft. Nó cung cấp cho các nhà phát triển nhiều thành phần để tạo ra các ứng dụng web đáp ứng và thân thiện với người dùng. Dưới đây là một số thành phần quan trọng của **ASP.NET**:

* **ASP.NET Web Forms**: Web Forms là một công nghệ cho phép các nhà phát triển tạo ra các ứng dụng web truyền thống và động. Nó cho phép bạn tạo các trang web động bằng cách sử dụng các control trên trình duyệt như các textbox, label và button.
* **ASP.NET MVC**: MVC (Model-View-Controller) là một kiến trúc phát triển ứng dụng web được sử dụng phổ biến trong ASP.NET. Nó cho phép tách biệt các phần của ứng dụng web như dữ liệu, giao diện và logic xử lý, giúp dễ dàng quản lý và bảo trì ứng dụng.
* **ASP.NET Web API**: Web API là một công nghệ cho phép các nhà phát triển tạo ra các dịch vụ web RESTful (Representational State Transfer) để truyền tải dữ liệu giữa các ứng dụng.
* **ASP.NET SignalR**: SignalR là một thư viện cho phép các ứng dụng web thời gian thực với tính năng đẩy thông báo tới người dùng, kết nối thời gian thực và gửi dữ liệu giữa các ứng dụng.
* **Entity Framework**: Entity Framework là một ORM (Object-Relational Mapping) cho phép các nhà phát triển truy cập vào cơ sở dữ liệu một cách dễ dàng thông qua mã hóa đối tượng. Nó cho phép bạn tương tác với cơ sở dữ liệu như các đối tượng thay vì viết các truy vấn SQL trực tiếp.
* **ASP.NET Identity**: Identity là một khung xác thực và quản lý người dùng giúp cho các nhà phát triển có thể xây dựng tính năng đăng nhập, đăng ký và quản lý người dùng vào ứng dụng web của họ.

Tất cả các thành phần trên đều là những công nghệ quan trọng của **ASP.NET**, giúp cho các nhà phát triển xây dựng các ứng dụng web đáp ứng và thân thiện với người dùng, và làm cho **ASP.NET** trở thành một trong những nền tảng phát triển ứng dụng web phổ biến nhất trên thế giới.

Và ở dự án này, tôi chọn **ASP.NET MVC** để hoàn thành dự án của mình

1. Mô hình MVC



3.1 Giới thiệu về MVC

**MVC** là viết tắt của cụm từ **“Model-View-Controller“**. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. **MVC** là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. **MVC** chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

* **Model (dữ liệu)**: Quản lí xử lí các dữ liệu.
* **View (giao diện)**: Nới hiển thị dữ liệu cho người dùng.
* **Controller (bộ điều khiển)**: Điều khiển sự tương tác của hai thành phần **Model** và **View**

**MVC** được tiến sĩ **Trygve Reenskaug** đưa vào [ngôn ngữ lập trình](https://vietnix.vn/ngon-ngu-lap-trinh/) **Smalltalk-76** khi ông đến trung tâm Nghiên cứu **Xerox Palo Alto (PARC)** vào giữa năm 1970. Sau đó, việc triển khai trở nên phổ biến trong các phiên bản khác của **Small- Talk**. Năm 1988, các bài báo *“The Journal of Object Technology” – JOT* mang lại bước tranh toàn cảnh về **MVC** mang liệu sự hiệu quả tốt nhất.

A diagram of a process

Description automatically generated with low confidence

**Luồng xử lý trong của mô hình MVC**, bạn có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

* Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
* Sau đó, **Controller** xử lý **input** của user rồi giao tiếp với**Model** trong MVC.
* Model chuẩn bị data và gửi lại cho Controller.
* Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

*Ở đây,****View*** *không giao tiếp trực tiếp với* ***Model****. Sự tương tác giữa****View*** *và* ***Model****sẽ chỉ được xử lý bởi****Controller.***

* 1. Ưu và nhược điểm MVC
     1. Ưu điểm:
* Đầu tiên, nhắc tới ưu điểm mô hình **MVC** thì đó là băng thông ([**Bandwidth**](https://vietnix.vn/bandwidth-la-gi/)) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn.
* Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng.
* Một lợi thế chính của **MVC** là nó tách biệt các phần **Model**, **Controller** và **View** với nhau.
* Sử dụng mô hình **MVC** chức năng **Controller** có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau
* Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau.
* Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.
* Hỗ trợ [**TTD**](https://en.wikipedia.org/wiki/Test-driven_development) (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các won [**test case**](https://vietnix.vn/test-case-la-gi/).
* Phiên bản mới nhất của **MVC** hỗ trợ trợ thiết kế responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ **View** của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống.
  + 1. Nhược điểm:
* Bên cạnh những ưu điểm MVC mang lại thì nó cũng có một số nhược điểm cần khắc phục.
* MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với với các dự án nhỏ, lẻ vì khá là cồng kềnh và mất thời gian.
  1. Vì sao nên sử dụng mô hình MVC
     1. Quy trình phát triển nhanh hơn
* **MVC** hỗ trợ phát việc phát triển nhanh chóng và song song. Nếu một **mô hình MVC** được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên **View** và một developer khác có thể làm việc với **Controller** để tạo logic nghiệp vụ cho ứng dụng web đó.
* Do đó, **ứng dụng mô hình MVC** có thể được hoàn thành nhanh hơn ba lần so với các ứng dụng mô hình khác.
  + 1. Khả năng cung cấp nhiều chế độ View
* Trong **mô hình MVC**, bạn có thể tạo nhiều **View** cho chỉ mộtmô hình. Ngày nay, nhu cầu có thêm nhiều cách mới để truy cập ứng dụng và đang ngày càng tăng. Do đó, việc sử dụng **MVC** để phát triển chắc chắn là một giải pháp tuyệt vời.
* Hơn nữa, với phương pháp này, việc nhân bản code rất hạn chế. Vì nó tách biệt dữ liệu và logic nghiệp vụ khỏi màn hình.
  + 1. Các sửa đổi không ảnh hưởng đến toàn bộ mô hình
* Đối với bất kỳ ứng dụng web nào, người dùng có xu hướng thay đổi thường xuyên. Bạn có thể quan sát thông qua những thay đổi thường xuyên về màu sắc, font chữ, bố cục màn hình. Hay là thêm hỗ trợ thiết bị mới cho điện thoại hay máy tính bảng…
* Việc thêm một kiểu view mới trong MVC rất đơn giản. Vì phần Model không phụ thuộc vào phần View. Do đó, bất kỳ thay đổi nào trong Model sẽ không ảnh hưởng đến toàn bộ kiến trúc.
  + 1. MVC Model trả về dữ liệu mà không cần định dạng
* **MVC pattern** có thể trả về dữ liệu mà không cần áp dụng bất kỳ định dạng nào. Do đó, các thành phần giống nhau có thể được sử dụng với bất kỳ giao diện nào.
* ***Ví dụ:*** tất cả loại dữ liệu đều có thể được định dạng bằng HTML. Ngoài ra, nó cũng có thể được định dạng bằng **Macromedia Flash** hay **Dream Viewer.**

1. Các bước để xây dựng Website bằng MVC 5

Khi bạn đã hiểu **MVC là gì**, thì nó giúp ích cho bạn có được một kiến thức cần thiết và nó là một trong các kỹ năng cần thiết khi bạn làm lập trình. Để sử dụng tốt **mô hình kiến trúc**này bạn cần có các kỹ năng và kiến thức dưới đây:

* Hiểu rõ về mô hình kiến trúc phần mềm (MVC).
* Hiểu cách sử dụng framework.
* Có kiến thức cơ bản về lập trình hướng đối tượng.
* Có khả năng logic và hiển thị nội dung, đảm bảo được rằng Model và View độc lập với nhau.

Sau khi đã nắm rõ được những kiễn thức cơ bản này, ta đi vào thực hành:

* **Bước 1:** Gõ tìm kiếm trong trình duyệt cụm từ “***download visual studio***” hoặc truy cập vào đường link này: [*https://visualstudio.microsoft.com/downloads/*](https://visualstudio.microsoft.com/downloads/)
* **Bước 2:** Tải phần mềm ***Visual studio*** *về.* Bạn có thể tải bản 2019 hoặc 2022 *(Bản 2022 sẽ tập hợp nhiều tính năng mới).*
* **Bước 3:** Chạy phần mềm ***Visual studio***

A screenshot of a computer

Description automatically generated with low confidence

*A screenshot of a computer

Description automatically generated*

*Tiếp theo chúng ta sẽ click chọn những gói hổ trợ để hoành thành việc cài đặt và làm một website bằng MVC 5*

*A screenshot of a computer

Description automatically generated*

*Chờ một chút để bản cài đặt hoàn thành*

* **Bước 4:** Ở phần mềm ***Visual Studio*** ta chọn vào ***Create a new project***

A screenshot of a computer

Description automatically generated

* **Bước 5:** Tìm kiếm và chọn vào ***ASP.NET Web Application (.NET Framework)***

A screenshot of a computer

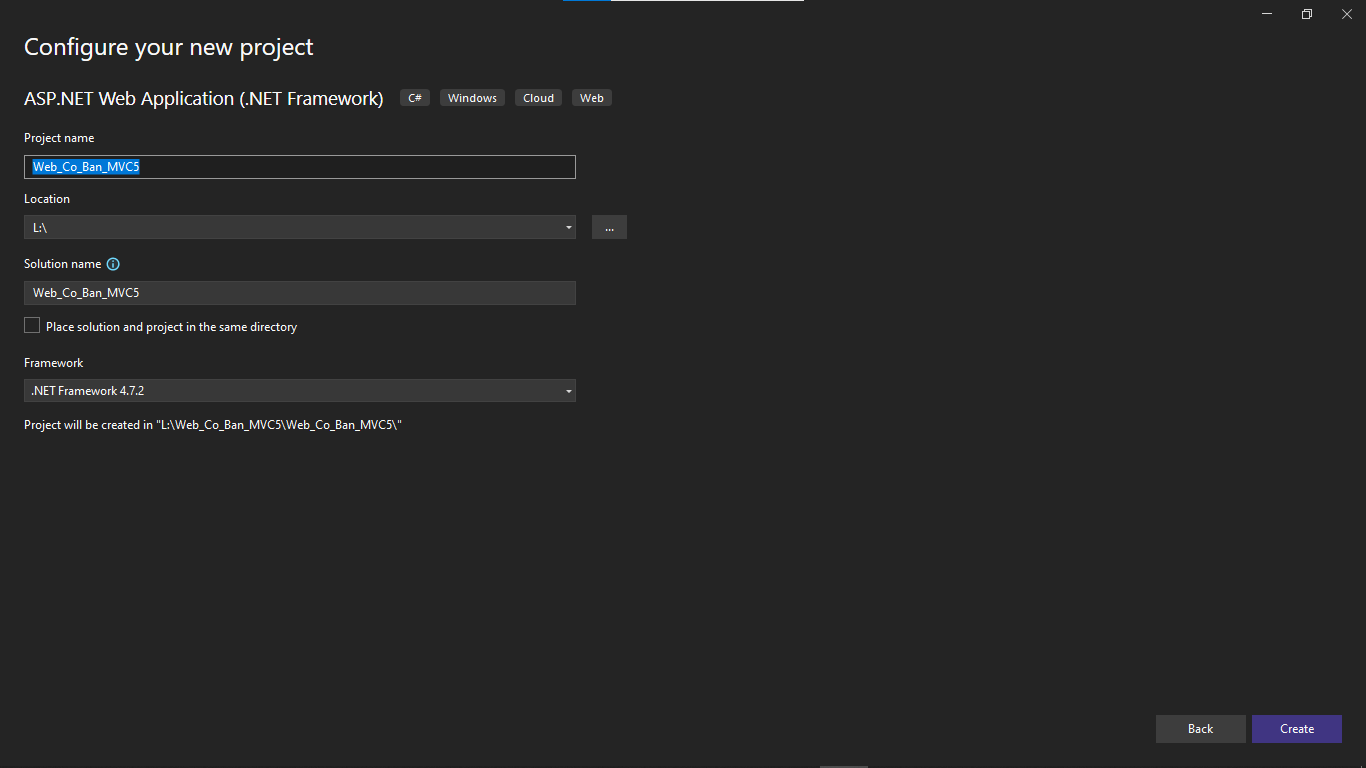
Description automatically generated

* **Bước 6:** Chúng ta đặt tên, chọn đường dẫn và nhấn ***Create***

**A screenshot of a computer

Description automatically generated**

* **Bước 7:** Chọn ***Empty*** và chọn ***MVC*** ở ***Add folders & references*** hoặc chúng ta có thể chọn luôn ***MVC*** (loại này nó sẽ tạo sẵn 1 bài web cơ bản và cài sẵn những ***NuGet Packages***)

****

**A screenshot of a computer

Description automatically generated**

*Đây là bố cục của một dự án vừa mới tạo*

* **Bước 8:** Ở trong dự án, ta chọn vào thư mục ***App\_Start*** và chọn ***RouteConfig.cs***, Ở file cs này sẽ là nơi quyết định khi người dùng truy cập vào trang web của chúng ta thì nó sẽ chạy đến giao diện nào đầu tiên.

**A screenshot of a computer program

Description automatically generated with medium confidence**

*Ta có thể thay đổi tên của Controller và Action theo ý thích*

* **Bước 9:** Ở thư mục ***View***, chúng ta tạo 1 thư mục có tên là ***Shared***. Ở trong thư mục này sẽ là nơi để lưu trữ các tập tin mà bạn muốn sử dụng chung trong nhiều trang web hoặc các phần khác nhau của dự án của bạn

A screenshot of a computer

Description automatically generated

*Click phải vào Shared > Add > New Item > Web > Razor > chọn Add*

*Ở file này chúng ta sẽ code html, css, js hoặc dẫn link css, js vào vẫn được, những file cshtml trong View khi kết nối với file này thì sẽ không cần phải code lại những thứ đó nữa.*

* **Bước 10:** Chúng ta chọn vào thư mục Controller, ***click phải > chọn add > chọn controller > chọn MVC 5 Controller – Empty*** (có 2 loại ở dưới sẽ hỗ trợ chúng ta tùy theo chức năng)

**A screenshot of a computer

Description automatically generated**

*Nhấn chọn* ***add*** *đặt tên và chọn* ***add*** *(Lưu ý phải đặt tên controller giống như tên đã đặt ở* ***RouteConfig.cs*** *để khi chạy web thì nó sẽ chạy đến đúng hướng trang đã chọn trong* ***RouteConfig.cs****)*

* **Bước 11:** Click chuột phải vào hàm muốn tạo giao diện > chọn Add View… > chọn MVC 5 View > chọn Add

**A screen shot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

*Đặt tên View, chọn đường dẫn file Razor chúng ta đã tạo ở trên > chọn Add*

*(Lưu ý: tương tự như Controller, đặt tên View phải giống như tên đã đặt ở Action)*

* **Bước 12:** Xuất hiện file ***Index.cshtml***, chúng ta thử gõ vài dòng code vào đó. Chọn ***Build*** và sau đó Click chuột phải vào ***Project Web\_Co\_Ban\_MVC5*** > Hover chuột vào ***View*** > chọn ***View in Browser (Google Chrome)*** để chạy bài

*A screenshot of a computer

Description automatically generated*

A screenshot of a computer

Description automatically generated

*Vậy là 1 trang web đơn giản bằng MVC 5 đã hoàn thành*

*A screenshot of a computer

Description automatically generated with medium confidence*

*Viết thêm code html và css để vào Razor và web hoàn thành*

**Tổng kết:**

* + Tải phần mềm Visual Studio và tạo dự án **ASP.NET Web Application**
* Đặt tên Controller và Action ở ***RouteConfig.cs*** của thư mục ***App\_Start***
* Tạo thư mục ***Shared*** và tạo 1 file ***Razor*** cshtml (Có thể làm hoặc không)
* Tạo file ***controller*** và file ***cshtml*** ở ***View*,** đưa code vào.
* Click chuột phải vào dự án, chọn view > chọn *View in Browser…* hoặc nhấn tổ hợp phím *Ctrl + F5*

Chương 2, Xây dựng Website quản lý khách sạn

1. Phát biểu bài toán

Mục tiêu của dự án là xây dựng một trang web đặt phòng khách sạn trực tuyến với giao diện thân thiện, chức năng tìm kiếm nhanh chóng và dịch vụ hỗ trợ khách hàng tốt nhất. Trang web sẽ cung cấp thông tin chi tiết về các khách sạn, bao gồm hình ảnh, tiện nghi, đánh giá của khách hàng, giá cả và bản đồ chỉ dẫn.

Để đáp ứng được mục tiêu này, dự án sẽ cần có các chức năng chính sau:

* **Tìm kiếm nhanh chóng:** Khách hàng có thể tìm kiếm các khách sạn theo nhiều tiêu chí khác nhau, bao gồm vị trí, giá cả, loại hình,…
* **Đặt phòng trực tuyến:** Khách hàng có thể đặt phòng trực tuyến theo hai hình thức là thanh toán tại quầy hoặc thanh toán toàn bộ thông qua trang web với các phương thức thanh toán đa dạng và bảo mật.
* **Quản lý đơn đặt phòng:** Khách hàng có thể quản lý đơn đặt phòng của mình thông qua trang web, ví dụ như xem lại thông tin đặt phòng, hủy đơn và chỉnh sửa đơn…
* **Hỗ trợ khách hàng:** Trang web sẽ cung cấp dịch vụ hỗ trợ khách hàng trực tuyến để giải đáp mọi thắc mắc và yêu cầu của khách hàng.
* **Quản lý thông tin khách sạn:** Chủ khách sạn có thể đăng ký tài khoản và cập nhật thông tin về khách sạn của mình, bao gồm hình ảnh, địa chỉ, tiện nghi, giá cả, trạng thái phòng và thống kê.

Để đảm bảo chất lượng và tính bảo mật của trang web, dự án sẽ sử dụng các công nghệ mới nhất và tuân thủ các quy chuẩn, tiêu chuẩn trong lĩnh vực phát triển phần mềm và bảo mật thông tin.

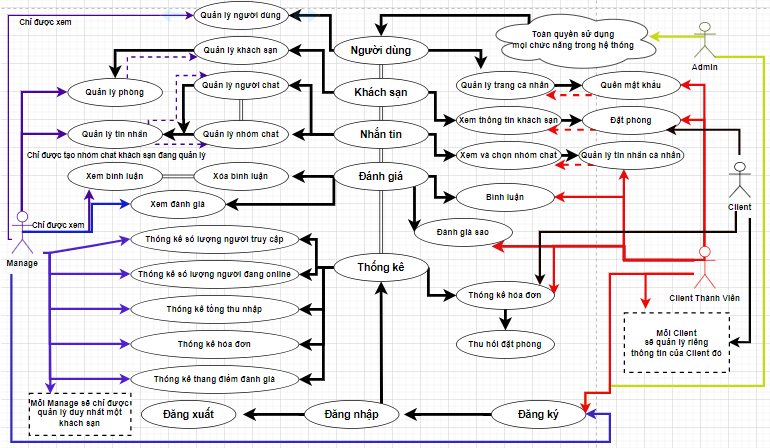
1. Phân tích thiết kế và cài đặt
   1. Phân tích nghiệp vụ
      1. Các công việc cần quản lý:

* Quản lý người dùng: Trong quản lý khách sạn, sẽ có 2 đối tượng người dùng được quản lý, đó là:
* Nhân viên lễ tân:đối tượng này sẽ được chủ doanh nghiệp quản lý
* Khách hàng: đối tượng này sẽ được nhân viên lễ tân và chủ doanh nghiệp khách sạn quản lý
* Quản lý khách hàng: Mỗi khách hàng của khách sạn đều được quản lý những thông tin sau: Họ tên, số điện thoại, gmail, chứng minh nhân dân hoặc căn cước công dân, địa chỉ, ảnh đại diện, tên đăng nhập, mật khẩu.
* Quản lý khách sạn và quản lý phòng: các khách sạn, các phòng của khách sạn sẽ được quản lý dựa vào định danh (mã khách sạn, mã phòng), loại hình, giá,… và được chia nhỏ những khách sạn cho đối tượng Nhân viên lễ tân quản lý. Khách sạn gồm nhiều loại phòng tương ứng với các mức giá và chất lượng của phòng. Việc quản lý phòng sẽ giúp biết được tình trạng phòng trống hay đã đặt để phục vụ công việc đặt phòng.
* Quản lý đặt phòng: Tất cả các thông tin đặt phòng đều được quản lý dựa trên mã hóa đơn, mã khách hàng, ngày đến, ngày đi, số phòng và mã khách sạn,…. Khách hàng có thể chọn hình thức là thanh toán tại quầy (sẽ cọc 20% tổng số tiền thanh toán) hoặc là thanh toán toàn bộ.
  + 1. Quy trình quản lý
* Hoạt động đặt phòng:
* khách hàng sẽ liên lạc với lễ tân thông qua điện thoại hoặc trực tiếp tại khách sạn, thông báo cụ thể về nhu cầu của mình (ngày đến, ngày đi, số lượng phòng, loại phòng. . . ).
* Sau đó, nhân viên lễ tân sẽ kiểm tra danh sách phòng, sổ ghi khách đăng kí, đối chiếu khả năng đáp ứng của khách sạn đối với nhu cầu của khách và trả lời.
* Nếu có thể đáp ứng được nhu cầu của khách, lễ tân sẽ yêu cầu khách hàng cung cấp thêm một số thông tin phục vụ cho nhu cầu đặt phòng: Thông tin khách hàng: số CMND, tên khách hàng, địa chỉ.
* Thông tin đặt phòng: ngày đến, ngày đi, loại phòng. Những thông tin này được dùng để lễ tân điền vào thông tin khách hàng vào sổ ghi khách đặt phòng.
* Hoạt động thuê phòng:
* Khi khách đến thuê phòng, nếu khách hàng đã  
  đăng kí trước, lễ tân sẽ cho khách nhận phòng với thông tin đã ghi trong sổ khách đặt phòng) và cập nhật lại thông tin khách hàng (nếu có sai sót thông tin).
* Sau đó, sẽ làm thủ tục giao phòng cho khách và cập nhật lại trạng thái hóa đơn đặt phòng.
* Lưu lại thông tin thuê phòng của khách hàng vào phiếu thuê phòng (lấy từ thông tin hóa đơn đặt phòng). Nếu khách hàng chưa đăng kí trước khách hàng phải cung cấp một số thông tin sau: Thông tin khách hàng: tên người nhận phòng, CMND, số điện thoại, giới tính, địa chỉ; Thông tin thuê phòng: loại phòng, mã phòng, ngày đến, ngày đi.
* Hoạt động trả phòng:
* Khi khách hàng trả phòng, khách hàng phải thanh toán tiền tại quầy lễ tân. Lễ tân sẽ thanh toán tất cả các khoản cho khách thông qua Hóa đơn có nội dung sau: số CMND, tổng tiền, ngày lập, nhân viên lễ tân sẽ in chi tiết hóa đơn (Thông tin phòng, tổng tiền phòng,… ) giao cho khách hàng và lưu lại thông tin hóa đơn và trả lại giấy CMND cho khách hàng.
  1. Thiết kế
     1. Biểu đồ Use Case

Dưới đây *(Hình 1)* là biểu đồ Use Case tổng quát, có 4 đối tượng tham gia vào hệ thống:

1. Admin: chủ hệ thống website.
2. Manage: chủ doanh nghiệp khách sạn.
3. Client Thành Viên: Những khách hàng có đăng ký tài khoản.
4. Client: Những khách hàng không đăng ký tài khoản.

* Biểu đồ tổng quát



*Hình 1: Biểu đồ Use case tổng quát*

Trong *(Hình 1),* đối tượng Admin sửa dụng hết toàn bộ mọi chức năng có trong hệ thống, quản lý 3 đối tượng: Manage, Client thành viên. Client.

* Phân rã Use case:
* Phân rã Use case người dùng

A picture containing text, diagram, screenshot, font

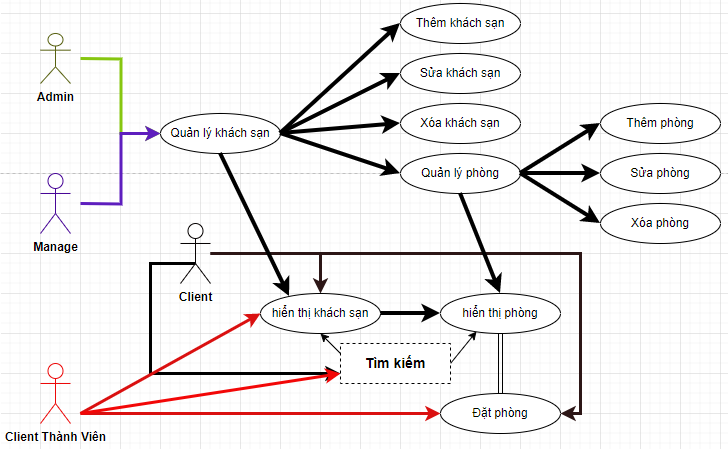
Description automatically generated

* Phân rã Use case thống kê

A picture containing text, diagram, line, plan

Description automatically generated

* Phân rã Use case khách sạn



* Phân rã Use case nhắn tin

A diagram of a chat group

Description automatically generated with low confidence

* Phân rã Use case đánh giá

A picture containing diagram, line, plot, text

Description automatically generated

* Phân rã Use case đăng ký, đăng nhập, đăng xuất

A picture containing diagram, text, line, circle

Description automatically generated

* + 1. Đặc tả

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Đặc tả Use case người dùng** Sẽ có 3 đối tượng sử dụng Use case này:   * Admin: tạo ra và quản lý. * Manage: chỉ được quyền sử dụng ở chế độ xem danh sách khách hàng hoặc sử dụng chức năng quản lý trang cá nhân. * Client thành viên: được sử dụng chức năng quản lý trang cá nhân. | | | | | |
| **Use Case** | | | | | **Chức năng** |
| **Quản lý người dùng** | | | | | Hệ thống cho phép Admin quản lý toàn bộ thông tin của người dùng. Use case được kích hoạt khi admin chọn chức năng “Quản lý người dùng trong menu chính của Admin”. Giao diện ban đầu đưa ra toàn bộ danh sách các người dùng và phân quyền.   * **Tạo người dùng mới:** Hệ thống hiển thị các trường tên, mật khẩu, địa chỉ, số điện thoại, gmail, phân quyền. Khi admin bấm ‘Lưu’, người dùng với phân quyền tương ứng được thêm vào hệ thống. * **Sửa người dùng hiện có:** Hệ thống hiển thị các trường tên, mật khẩu, phân quyền,… và kèm theo thông tin của người dùng đã đăng ký. Khi Admin sửa đổi thông tin và bấm “Lưu”. người dùng với phân quyền tương ứng được sửa đổi trong hệ thống. * **Xóa người dùng:** Ở giao diện danh sách người dùng có nút “Xóa”. Khi Admin Click vào thì mọi thông tin của người dùng đó sẽ bị xóa bỏ khỏi hệ thống. Trừ hóa đơn sẽ chỉ xóa thông tin người đó, giữ lại thông tin đặt khác như mã hóa đơn, tên khách sạn, …. |
| **Quản lý trang cá nhân** | | | | | Hệ thống cho phép 3 đối tượng trên quản lý toàn bộ thông tin cá nhân. Use case được kích hoạt khi chọn chức năng “Trang cá nhân trong menu chính của Client”. Giao diện ban đầu đưa ra toàn bộ thông tin của người dùng đó.   * **Sửa đổi thông tin:** Khi User sửa đổi thông tin và Click nút lưu. Hệ thống sẽ tiếp nhận và thay đổi thông tin. * **Quên mật khẩu:** Khi User Click vào quên mật khẩu. Hệ thống sẽ tự động quét gmail của người dùng và gửi mã về cho người dùng qua gmail để xác nhận. khi người dùng xác nhận đúng thì hệ thống sẽ bắt đầu cho thay đổi mật khẩu và hệ thống sẽ thay đổi lại mật khẩu như đã điền bằng mã hóa MD5. |
| **Use Case** | | | | | **Chức năng** |
| **Đặc tả Use case khách sạn** Sẽ có 4 đối tượng sử dụng Use case này:   * Admin: tạo ra và quản lý khách sạn và phòng của khách sạn đó. * Manage: chỉ được quyền sử dụng ở phạm vi quản lý một khách sạn. * Client thành viên: được quyền sử dụng chức năng tìm kiếm, hiển thị, đặt phòng. * Client: tương tự như Client thành viên. được quyền sử dụng chức năng tìm kiếm, hiển thị, đặt phòng. | | | | | |
| **Quản lý khách sạn** | | | | Hệ thống sẽ cho phép Admin tạo ra nhiều khách sạn và Manage tạo ra duy nhất một khách sạn. Từ đó nhập thông tin phòng của những khách sạn đó.   * **Thêm mới khách sạn:** Hệ thống hiển thị một giao diện bản đồ, khi Admin, Manage Click vào 1 khu vực tùy chọn trên bản đồ và hiển thị các trường tên, số điện thoại, gmail, địa chỉ thì hệ thống sẽ hiển thị thông tin mà khu vực đã click vào. Khi admin, manage bấm ‘Lưu’, khách sạn với vị trí tương ứng được thêm vào hệ thống và có mặt trên bản đồ. * **Sửa khách sạn hiện có:** Hệ thống hiển thị các thông tin tên khách sạn, địa chỉ, số điện thoại đã đăng ký. Khi Admin, Manage sửa đổi thông tin và bấm “Lưu”. người dùng với phân quyền tương ứng được sửa đổi trong hệ thống. * **Xóa khách sạn:** Ở giao diện danh sách khách sạn có nút “Xóa”. Khi Admin, Manage Click vào thì mọi thông tin của khách sạn đó sẽ bị xóa bỏ khỏi hệ thống. * **Thêm mới phòng và quản lý:** Khi click vào sửa khách sạn thì sẽ có một nút “chi tiết phòng”. Khi click vào nút “chi tiết phòng” giao diện taoh phòng sẽ hiện ra và thực hiện thêm, sửa, xóa. | |
| **Hiển thị khách sạn, phòng.**  **Đặt phòng** | | | | Hệ thống sẽ cho phép Client*(Client chưa thành viên và Client thành viên)* tìm kiếm xem thông tin khách sạn, danh sách phòng và đặt phòng:   * **Tìm kiếm:** Hệ thống cho phép tìm kiếm dựa trên những tiêu chí của khách sạn và của phòng. * **Đặt phòng:** Một danh khách phòng của khách sạn hiện ra, sẽ có những phòng đang trống và những phòng đang có khách được phân biệt. Client có thể chọn này đến và đi để xem khoảng thời gian đó còn trống những phòng nào để đặt. Khi người dùng đã chọn phòng, bước đến công việc cuối là lựa chọn hình thức thanh toán: *Thanh toán tại quầy* và *Thanh toán toàn bộ.* Sau khi đã lựa chọn và thanh toán thì Client sẽ được hỗ trợ thanh toán qua ví VNPay. Hoàn thành thanh toán, Client sẽ được nhận một hóa đơn qua website và qua gmail. | |
| **Use Case** | | | | **Chức năng** | |
| **Đặc tả Use case nhắn tin** Sẽ có 3 đối tượng sử dụng Use case này:   * Admin: Tạo Group chat và nhắn tin hoặc cũng có thể chat riêng với Manage, Client thành viên. * Manage: Được tạo Group chat khách sạn đang quản lý và gửi tin nhắn hoặc cũng có thể chat riêng với Admin, Client thành viên. * Client thành viên: được quyền truy cập vào group chat mà Admin và Manage đã tạo để chat hoặc có thể chat riêng với Admin, Manage. | | | | | |
| **Quản lý tin nhắn** | | | Hệ thống sẽ cho phép tạo Group và chat:   * **Tạo group chat:** Admin và Manage sẽ được quyền tạo group. Admin được quyền thêm hoặc xóa toàn bộ group, còn Manage chỉ được quyền xóa group mà đã tự tạo. * **Gửi tin nhắn:** Sau khi lựa chọn chat group hay là chat user, người dùng gỡ tin nhắn hoặc chọn ảnh rồi nhấn gửi. Hệ thống sẽ tiếp nhận và hiển thị trên giao diện chat. * **Xóa tin nhắn:** Hệ thống cho phép * Admin được quyền xóa toàn bộ tin nhắn. * Manage và Client chỉ được xóa tin nhắn mà cá nhân tự gửi. | | |
| **Use Case** | | | **Chức năng** | | |
| **Đặc tả Use case đánh giá** Sẽ có 3 đối tượng sử dụng Use case này:   * Admin: được quyền xem bình luận và xóa bình luận, chỉ được xem đánh giá sao. * Manage: chỉ được quyền xem bình luận và đánh giá sao, không được sửa đổi hoặc xóa. * Client thành viên: được quyền bình luận, xóa bình luận và đánh giá sao. | | | | | |
| **Quản lý bình luận, đánh giá sao** | | Hệ thống chỉ cho phép Client viết bình luận và xóa bình luận.  Khi Client đã từng đặt phòng và sử dụng phòng, hệ thống sẽ cho phép Client đánh giá dựa trên thanh điểm 5 sao.  Trường hợp Admin muốn xóa bình luận vi phạm thì click vào nút xóa, hệ thống sẽ xóa nội dung tin nhắn đó. | | | |
|  | | | | | |
| **Use Case** | **Chức năng** | | | | |
| **Đặc tả Use case thống kê** Sẽ có 4 đối tượng sử dụng Use case này:   * Admin, Manage: được quyền xem các thông tin đã được hệ thống thông kê * Client thành viên: được quyền xem lịch sử đặt phòng và số sao đánh giá. * Client: Phải sử dụng tìm kiếm để xem được thông tin hóa đơn đã đặt. | | | | | |
| **Quản lý thống kê** | | | Hệ thống sẽ thống kê 5 loại:   * (1)Thống kê số lượng người truy cập. * (2)Thống kê số lượng người đang online. * (3)Thống kê tổng thu nhập. * (4)Thống kê hóa đơn. * (5)Thống kê thang điểm đánh giá *(5 sao)* | | |

* + 1. Biểu đồ trình tự

*Biểu đồ trình tự (Sequence diagram) là biểu đồ tương tác theo trật tự thời gian của các giao tiếp bằng thông điệp giữa các đối tượng, biểu đồ được đọc từ đỉnh xuống đáy. Mỗi UC có nhiều luồng dữ liệu. Mỗi biểu đồ trình tự biểu diễn một luồng dữ liệu.*

Từ bài toán trên ta có các biểu đồ trình tự sau:

* **Biểu đồ trình tự đăng nhập:**

**A picture containing text, diagram, number, font

Description automatically generated**

* Biểu đồ trình tự thêm mới:

A picture containing text, screenshot, line, number

Description automatically generated

* Biểu đồ trình tự tìm kiếm:

A picture containing text, diagram, line, screenshot

Description automatically generated

* Biểu đồ trình tự đặt phòng:

A picture containing text, number, font, line

Description automatically generated

* + 1. Biểu đồ lớp

*Biểu đồ lớp chỉ ra trừu tượng thế giới thực, tập trung vào giải thích cấu trúc tĩnh từ góc nhìn tổng quát. Thông thường mỗi hệ thống có vài biểu đồ lớp. Một số biểu đồ lớp trong số đó hiển thị lớp và quan hệ giữa các lớp, một vài biểu đồ lớp khác hiển thị gói lớp và quan hệ giữa các gói. Có thể tạo rất nhiều biểu đồ lớp để mô tả toàn bộ bức tranh hệ thống. Các biểu đồ lớp giúp người phát triển phần mềm quan sát và lập kế hoạch cấu trúc hệ thống trước khi viết mã trình. Nó đảm bảo hệ thống được thiết kế tốt ngay từ ban đầu.*

* Biểu đồ lớp cho chức năng quản lý khách hàng

A screenshot of a computer

Description automatically generated with medium confidence

* Biểu đồ lớp cho chức năng quản lý khách sạn

A screenshot of a computer

Description automatically generated with low confidence

* Biểu đồ lớp cho chức năng tìm kiếm khách sạn

A screenshot of a computer

Description automatically generated with low confidence

* Biểu đồ lớp cho chức năng quản lý Phòng

A screenshot of a computer

Description automatically generated with low confidence

* Biểu đồ lớp cho chức năng tìm kiếm phòng

A picture containing text, diagram, screenshot, line

Description automatically generated

* Biểu đồ lớp cho chức năng đặt phòng

A picture containing text, diagram, screenshot, line

Description automatically generated

* 1. Cài đặt

1. Hướng dẫn sử dụng

Phần Kết luận

Xây dựng website quản lý khách sạn trực tuyến thực sự cần thiết đối với mọi khách sạn để đáp ứng nhu cầu của du khách và cung cấp cho họ trải nghiệm phù hợp.