**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

**BÁO CÁO KẾT QUẢ ĐỀ TÀI KHCN**

**Tên đề tài: “Nghiên cứu và triển khai hệ thống phần mềm, cơ sở dữ liệu và website tương tác phục vụ hướng dẫn du lịch thông minh tại tỉnh Tuyên Quang”**

**Mã số: 15/2017/HĐ-KHCN-ĐT**

**Nội dung 4.8**

**Xây dựng phân hệ quản lý người dùng**

**Cơ quan chủ trì đề tài: Viện Điện tử - Viễn Thông,**

**Trường Đại học Bách Khoa Hà Nội**

**Chủ nhiệm đề tài: TS. Trần Quang Vinh**

Tuyên Quang, Tháng 11/2018

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

**BÁO CÁO KẾT QUẢ ĐỀ TÀI KHCN**

**Tên đề tài: “Nghiên cứu và triển khai hệ thống phần mềm, cơ sở dữ liệu và website tương tác phục vụ hướng dẫn du lịch thông minh tại tỉnh Tuyên Quang”**

**Mã số: 15/2017/HĐ-KHCN-ĐT**

**Nội dung 4.8**

**Xây dựng phân hệ quản lý người dùng**

|  |  |
| --- | --- |
| **Chủ nhiệm đề tài** | **Cơ quan chủ trì đề tài** |

Tuyên Quang, Tháng 11/2018

# ĐẶT VẤN ĐỀ

Trong xu thế phát triển của công nghệ hiện nay cùng với sự ra đời, phát triển của các thiết bị di động hiện đại hỗ trợ người dùng như: Điện thoại thông mình, máy tính bảng,.. kết hợp với nền tảng công nghệ Internet đang rất phổ biến và được sử dụng rộng rãi, việc du lịch trở nên ngày càng dễ dàng và tiện lợi hơn cùng với các dịch vụ mới. Bên cạnh đó là sự phát triển của các công nghệ như bản đồ số, công nghệ GPS, Wi-Fi cùng với các giao thức truyền thông đa phương tiện khác, hệ thống hướng dẫn du lịch thông minh sẽ tạo cho du khách các trải nghiệm về du lịch như xác định vị trí trên bản đồ dựa vào hệ thống định vị toàn cầu (GPS) hoặc hệ thống định vị sử dụng Wi-Fi, xem thông tin chi tiết về một địa điểm, vật thể bao gồm mô tả chi tiết, hình ảnh, video giới thiệu. Ngoài ra, khách du lịch còn có thể bình luận, nhận xét hoặc đánh dấu các địa điểm trên bản đồ và chia sẽ địa điểm đó cho người dùng khác.

Tham quan du lịch là một hình thức giải trí sau những giờ làm việc căng thẳng. Chúng ta được đi đến những địa điểm du lịch và tăng cao kiến thức về đất nước con người. Nếu bạn muốn thăm quan du lịch tại một nơi mà bạn chưa hề biết? Bạn muốn đi với các bạn mình trong dịp nào đó và không biết rõ về đường đi cũng như địa điểm các khu du lịch nơi đó? Bạn muốn tự do chụp ảnh mà không ảnh hưởng đến việc nghe thông tin mà hướng dẫn viên du lịch nói? Bạn không muốn tham gia một tour du lịch mà vẫn có thể chủ động xem thông tin địa danh một cách chi tiết nhất? Bạn có thể lựa chọn địa danh gần nhất với mình hoặc địa danh mình thích nhất?

Dịch vụ dựa trên nền tảng mạng Internet ngày càng phát triển, đời sống con người ngày càng nâng cao với các nhu cầu nâng cao chất lượng cuộc sống như du lịch, khám phá... Hơn thế nữa, các thiết bị di động thông minh ngày càng phổ biến.

Để xây dựng được ứng dụng đáp ứng được mục đích, yêu cầu đề ra, đối tượng nghiên cứu dựa trên nền tảng kiến thức về lý thuyết bản đồ số, Web Map Services, định vị sử dụng Wi-Fi, GPS, hệ quản trị cơ sở dữ liệu Mysql, máy chủ chạy Web Services Apache Tomcat và Streaming Server kèm theo các kỹ năng lập trình ứng dụng trên smartphone.

Hệ thống này nhằm cung cấp cho khách du lịch các dịch vụ thuận tiện hơn, thông minh hơn dựa trên Smart Phone, cho phép tương tác trực tiếp với bản đồ, xác định vị trí người dùng, chi tiết hóa bản đồ, cung cấp các thông tin đa phương tiện về địa điểm, đối tượng. Hệ thống cũng nhắm đến xây dựng thông tin đa phương tiện để giới thiệu địa danh lịch sử, danh lam thắng cảnh, địa danh văn hóa, thông tin vị trí địa lý như nhà hàng ẩm thực, khách sạn, khu thương mại, giao thông được lưu trữ tại máy chủ.

Khách du lịch với smartphone của mình di chuyển tới vị trí địa danh lịch sử thì thông qua hệ thống định vị vệ tinh, hệ thống phần mềm sẽ tự động phát hiện và giới thiệu về địa danh đó, những thông tin này gồm cả hình ảnh, âm thanh bằng ngôn ngữ tiếng Anh hoặc tiếng Việt tùy theo lựa chọn của khách du lịch. Nhờ vậy, khách du lịch vừa có thể theo dõi thông tin về di tích, văn hóa nơi mà mình đang thăm quan thông qua thiết bị di động của mình, vừa có thể quan sát thực tế. Đây chính là chức năng hướng dẫn du lịch từ xa của hệ thống. Đồng thời, hệ thống phần mềm có chức năng tự động dẫn hướng tour du lịch cho khách, giúp cho việc lựa chọn tuyến đường phù hợp và thăm quan được đầy đủ và thuận lợi cho hành trình của mình.

Ngoài ra, các thông tin hỗ trợ dịch vụ du lịch cũng được tích hợp trong hệ thống và có thể tương tác với du khách, nên khách du lịch có thể lựa chọn phương tiện giao thông, địa điểm phù hợp cho việc nghỉ ngơi, mua sắm và thưởng thức văn hóa ẩm thực của đồng bào dân tộc. Với mục tiêu dựa trên việc định vị vị trí của người sử dụng khi di chuyển trong khu vực có di tích lịch sử, hệ thống sẽ hiển thị những thông tin giới thiệu tương ứng với vị trí của người dùng.

Các hoạt động chính của chuyên đề này bao gồm:

- Xây dựng mô hình quản trị người dùng trên Website

# MỤC LỤC

[ĐẶT VẤN ĐỀ 3](#_Toc529280425)

[MỤC LỤC 5](#_Toc529280426)

[DANH MỤC HÌNH VẼ 6](#_Toc529280427)

[DANH MỤC BẢNG BIỂU 7](#_Toc529280428)

[CHƯƠNG 1: CÁC CÔNG CỤ CẦN THIẾT 8](#_Toc529280429)

[1.1 Giới thiệu về ASP. NETFRAMEWORK 8](#_Toc529280430)

[1.1.1 ASP.NETFRAMEWORK 8](#_Toc529280431)

[1.1.2 ASP.NET 11](#_Toc529280432)

[1.1.3 ASP.NET IDENTITY 14](#_Toc529280433)

[1.2 Mô hình MVC 15](#_Toc529280434)

[1.3 Model 18](#_Toc529280435)

[1.3.1 Giới thiệu về SQL Server 2012 18](#_Toc529280436)

[1.3.2 Các Thành Phần Của Một CSDL Trong SQL Server: 20](#_Toc529280437)

[1.3.3 Các câu lệnh cơ bản trong SQL SERVER 21](#_Toc529280438)

[CHƯƠNG 2: QUY TRÌNH THIẾT KẾ 24](#_Toc529280439)

[2.1 Yêu cầu chức năng 24](#_Toc529280440)

[2.2 Yêu cầu phi chức năng 24](#_Toc529280441)

[2.3 Quản trị người dùng trên Website 25](#_Toc529280442)

[2.3.1 Sơ đồ chức năng 25](#_Toc529280443)

[2.3.2 Sơ đồ chuẩn hoá cơ sở dữ liệu phân quyền người dùng 25](#_Toc529280444)

[CHƯƠNG 3: TRIỂN KHAI 27](#_Toc529280445)

[3.1 Quy trình 27](#_Toc529280446)

[3.2 Nội dung thực hiện 29](#_Toc529280447)

[CHƯƠNG 4: KẾT LUẬN 32](#_Toc529280448)

[TÀI LIỆU THAM KHẢO 33](#_Toc529280449)

# DANH MỤC HÌNH VẼ

[Hình 1.1 Mô hình MVC 16](#_Toc529280480)

[Hình 2.1 Sơ đồ chức năng quản lý người dùng 25](#_Toc529280481)

[Hình 2.2 Sơ đồ chuẩn hóa cơ sở dữ liệu cho tính năng phân quyền. 26](#_Toc529280482)

[Hình 3.1 Chức năng chung quản lý người dùng 27](#_Toc529280483)

[Hình 3.2 Giao diện đổi mật khẩu tài khoản cho user và doanh nghiệp 28](#_Toc529280484)

[Hình 3.3 Giao diện phân quyền cho tài khoản user và doanh nghiệp 28](#_Toc529280485)

[Hình 3.4 API lấy tất cả tài khoản trong hệ thống 29](#_Toc529280486)

[Hình 3.5 API xem thông tin chi tiết tài khoản 29](#_Toc529280487)

[Hình 3.6 API cập nhập tài khoản 30](#_Toc529280488)

[Hình 3.7Api đổi mật khẩu tài khoản 30](#_Toc529280489)

[Hình 3.8 API Phân quyền cho tài khoản 31](#_Toc529280490)

# DANH MỤC BẢNG BIỂU

[Bảng 1.1 Bảng câu lệnh thao tác dữ liệu 21](#_Toc529280491)

[Bảng 1.2 Bảng câu lệnh định nghĩa dữ liệu 21](#_Toc529280492)

[Bảng 1.3 Bảng câu lệnh điều kiển truy cập 22](#_Toc529280493)

[Bảng 1.4 Bảng câu lệnh quản lý giao dịch 22](#_Toc529280494)

[Bảng 1.5 Bảng câu lệnh lập trình 23](#_Toc529280495)

[Bảng 2.1 Bảng kê chức năng từng bảng trong cơ sở dữ liệu phân quyền 26](#_Toc529280496)

# CÁC CÔNG CỤ CẦN THIẾT

## Giới thiệu về ASP. NETFRAMEWORK

### ASP.NETFRAMEWORK

*ASP.NET Framework* là một công nghệ hỗ trợ việc xây dựng và chạy các ứng dụng trên nền tảng Framework và dịch vụ Web XML

ASP.NET Framework được thiết kế để thực hiện các mục tiêu:

* Để cung cấp một môi trường lập trình hướng đối tượng phù hợp việc xem mã đối tượng đã được lưu trữ và thực thi cục bộ, thực thi cục bộ nhưng chạy trên nền Internet, hoặc thực thi từ xa.
* Để cung cấp một môi trường thực thi mã nguồn giảm thiểu việc đóng gói phần mềm, xung đột phiên bản.
* Để cung cấp một môi trường thực thi giúp cho việc thực thi mã an toàn, kể cả các công cụ được tạo ra bởi bên hãng thứ ba hay bất kỳ hãng nào miễn sao phải tuân thủ kiến trúc .Net.
* Để cung cấp một môi trường thực thi mã trong đó loại bỏ các vấn đề hiệu suất của môi trường làm việc như script hay môi trường thông dịch.
* Để giúp cho người phát triển có kinh nghiệm vững chắc có thể nắm nhiều ứng dụng khác nhau từ ứng dụng trên nền tảng ứng dụng Window đến ứng dụng Web.
* Tạo ra chuẩn làm việc cho việc phát triển ứng dụng và thống nhất giữa các bên khi sử dụng và đảm bảo mã nguồn .NET có thể tích hợp được với bất kỳ mã nguồn khác

ASP.NET Framework bao gồm *Common Language Runtime* (CLR) và *Net Framework Class Library.*

*CLR* phổ biến là nền tảng của .NET Framework. Bạn có thể nghĩ của thời gian chạy như một tác nhân quản lý mã khi nó thực hiện, cung cấp các dịch vụ cốt lõi như quản lý bộ nhớ, quản lý luồng, và điều khiển từ xa, tăng cường thực thi mã an toàn, chính xác. Trong thực tế, các khái niệm về quản lý mã là một nguyên tắc cơ bản của thời gian chạy - runtime. Trong khí đó mã nguồn cho nhiều thời gian chạy được gọi là mã được quản lý (managed code), ngược lại gọi là mã không quản lý (unmanaged code) – mã nguồn mà không có đích tới runtime.

*Các lớp thư viện* là một bộ sưu tập toàn diện hướng đối tượng cho phép tái sữ dụng để phát triển các ứng dụng khác nhau từ dòng lệnh truyền thống (Console) cho đến đồ hoạ (GUI) đến những cải tiến mới nhất được cung cấp bởi ASP.NET, chẳng hạn như Web Forms và các dịch vụ Web XML.

.NET Framework có thể được lưu trữ bởi các thành phần không được quản lý thời gian chạy ngôn ngữ phổ biến vào các quá trình bắt đầu thực hiện các mã nguốn được quản lý, do đó tạo ra một môi trường phần mềm có thể khai thác cả hai tính năng được quản lý hoặc không được quản lý.

.NET Framework không chỉ cung cấp một số máy chủ thời gian chạy, nhưng cũng hỗ trợ sự phát triển các máy chủ thời gian chạy của bên thứ ba.

Ví dụ, ASP.NET tổ chức thời gian chạy để cung cấp một môi trường, khả năng mở rộng phía máy chủ cho các mã nguồn được quản lý.

ASP.NET làm việc trực tiếp với thời gian chạy để cho phép các ứng dụng ASP.NET và dịch vụ Web XML, cả hai đều được thảo luận sau trong chủ đề này.Internet Explorer là một ví dụ của một ứng dụng không được quản lý tổ chức trong thời gian chạy (trong các hình thức của một phần mở rộng kiểu MIME). Sử dụng Internet Explorer để lưu trữ thời gian chạy cho phép bạn nhúng các thành phần quản lý hoặc Windows Forms được điều khiển trong các tài liệu HTML.

**Common Language Runtime**

CLR phổ biến là việc quản lý bộ nhớ, quản lý luồng, thực thi mã nguồn, mã kiểm định an toàn, biên dịch, và các dịch vụ hệ thống khác. Những tính năng trên là nền tảng để mã nguồn quản lý chạy trên CLR.

Vấn đề liên quan đến an ninh, các thành phần quản lý được cung cấp ở các mức độ khác nhau, tùy thuộc vào một số yếu tố bao gồm: Internet, mạng doanh nghiệp, hoặc máy tính cục bộ. Điều này có nghĩa rằng một thành phần được quản lý có thể hoặc có thể không có khả năng để thực hiện các hoạt động truy cập tập tin, truy cập registry, hay các chức năng nhạy cảm khác, ngay cả khi nó đang được sử dụng trong cùng một ứng dụng đang hoạt động.

CLR bảo mật truy cập mã. Ví dụ, người dùng có thể tin tưởng rằng một thực thi nhúng vào trong một trang web có thể chạy một hình ảnh động trên màn hình hoặc hát một bài hát, nhưng không thể truy cập dữ liệu cá nhân của họ, hệ thống tập tin, hoặc mạng. Các tính năng bảo mật của thời gian chạy cho phép các phần mềm hợp pháp Internet triển khai được đặc biệt tính năng phong phú.

CLR cũng thực thi mạnh mẽ mã nguồn bằng cách thực hiện một cơ sở hạ tầng loại và mã nguồn xác minh nghiêm ngặt được gọi là hệ thống kiểu chung (CTS – Common Type System). CTS đảm bảo rằng tất cả các mã nguồn được quản lý tự mô tả. Microsoft và trình biên dịch ngôn ngữ của bên thứ ba tạo ra mã nguồn được quản lý có thể phù hợp với CTS. Điều này có nghĩa rằng mã nguồn được quản lý có thể sử dụng các kiểu và các trường hợp được quản lý, trong khi nghiêm chỉnh thực thi chính xác và an toàn.

Ngoài ra, môi trường quản lý của thời gian chạy loại bỏ nhiều vấn đề phần mềm phổ biến. Ví dụ, thời gian chạy sẽ tự động xử lý bố trí đối tượng và quản lý tài liệu tham khảo cho các đối tượng, giải phóng chúng khi chúng không còn được sử dụng. Quản lý bộ nhớ tự động này còn giải quyết hai lỗi ứng dụng phổ biến nhất: rò rỉ bộ nhớ và tham chiếu bộ nhớ không hợp lệ.

Thời gian chạy cũng tăng tốc hiệu quả phát triển những phần mềm. Ví dụ, các lập trình viên có thể viết các ứng dụng trong phát triển ngôn ngữ của họ lựa chọn, chưa tận dụng thời gian chạy, thư viện lớp và các thành phần được viết bằng các ngôn ngữ khác bởi các nhà phát triển khác. Bất kỳ những nhà cung cấp trình biên dịch, những người lựa chọn đều nhắm đến mục tiêu thời gian chạy. Trình biên dịch ngôn ngữ mục tiêu .NET Framework thực hiện các tính năng của .NET Framework có mã hiện tại được viết bằng ngôn ngữ đó, giúp giảm bớt quá trình chuyển đổi cho các ứng dụng hiện có.

Trong khi thời gian chạy được thiết kế cho các phần mềm của tương lai, nó cũng hỗ trợ phần mềm của ngày nay và trước đây. Khả năng cộng tác giữa các mã nguồn được quản lý hoặc không được quản lý cho phép các nhà phát triển tiếp tục sử dụng các thành phần COM và DLL khi cần thiết.

Thời gian chạy được thiết kế để nâng cao hiệu suất thực hiện. Mặc dù CLR cung cấp các tiêu chuẩn dịch vụ thời gian chạy nhưng mã nguồn quản lý là không bao giờ được dịch. Một tính năng được gọi là just-in-time (JIT) biên dịch cho phép tất cả các mã nguồn được quản lý để chạy trong các ngôn ngữ máy tính của hệ thống mà trên đó nó được thực thi. Trong khi đó, quản lý bộ nhớ loại bỏ các khả năng của bộ nhớ bị phân mảnh và làm tăng bộ nhớ cục bộ, tham chiếu gia tăng đến hiệu suất.

Cuối cùng, thời gian chạy có thể được lưu trữ hiệu suất cao, các ứng dụng phía máy chủ, chẳng hạn như Microsoft SQL Server và Internet Information Services (IIS). Cơ sở hạ tầng này cho phép bạn sử dụng mã nguồn quản lý để viết business logic của bạn, trong khi vẫn sử dụng hiệu suất cao của máy chủ doanh nghiệp tốt nhất của ngành công nghiệp hỗ trợ lưu trữ thời gian chạy.

**NET Framework Class Library**

NET Framework là một bộ sưu tập các kiểu dữ liệu có thể tái sử dụng tích hợp chặt chẽ với CLR. Thư viện lớp là hướng đối tượng, cung cấp các kiểu dữ liệu mà từ đó mã nguồn quản lý của bạn có thể dẫn xuất. Điều này không chỉ làm cho các dữ liệu .Net Framework dễ dàng để sử dụng mà còn làm giảm thời gian liên quan đến việc học tập các tính năng mới của .NET Framework. Ngoài ra, các thành phần của bên thứ ba có thể tích hợp liền mạch với các lớp trong .NET Framework.

Ví dụ, các lớp .NET Framework thực hiện một tập hợp các giao diện mà bạn có thể sử dụng để phát triển các lớp riêng của bạn. Lớp của bạn tạo ra sẽ được sử dụng chung với các lớp hiện có của .NET Framework.

Như người phát triển mong chờ với một thư viện lớp hướng đối tượng, các kiểu .NET Framework cho phép người phát triển thực hiện một loạt các nhiệm vụ lập trình phổ biến, bao gồm: các công việc như quản lý chuỗi, thu thập dữ liệu, kết nối cơ sở dữ liệu, và truy cập file. Ngoài các nhiệm vụ chung, các lớp thư viện bao gồm các kiểu dữ liệu hỗ trợ một loạt các kịch bản phát triển chuyên biệt khác [1].

Ví dụ, người phát triển có thể sử dụng .NET Framework để phát triển các kiểu ứng dụng và dịch vụ như:

* Console Application
* Windows GUI Application
* Windows Presentation Foundation (WPF) Application
* ASP.NET Application
* Windows Services
* Windows Communication Foundation (WCF) Application
* Windows Workflow Foundation (WF) Application

### ASP.NET

Từ khoảng cuối thập niên 90, ASP (Active Server Page) đã được nhiều lập trình viên lựa chọn để xây dựng và phát triển ứng dụng web động trên máy chủ sử dụng hệ điều hành Windows. ASP đã thể hiện được những ưu điểm của mình với mô hình lập trình thủ tục đơn giản, sử dụng hiệu quả các đối tượng COM: ADO (ActiveX Data Object) - xử lý dữ liệu, FSO (File System Object) - làm việc với hệ thống tập tin…, đồng thời, ASP cũng hỗ trợ nhiều ngôn ngữ: VBScript, JavaScript. Chính những ưu điểm đó, ASP đã được yêu thích trong một thời gian dài.

Tuy nhiên, ASP vẫn còn tồn đọng một số khó khăn như Code ASP và HTML lẫn lộn, điều này làm cho quá trình viết code khó khăn, thể hiện và trình bày code không trong sáng, hạn chế khả năng sử dụng lại code. Bên cạnh đó, khi triển khai cài đặt, do không được biên dịch trước nên dễ bị mất source code. Thêm vào đó, ASP không có hỗ trợ cache, không được biên dịch trước nên phần nào hạn chế về mặt tốc độ thực hiện. Quá trình xử lý Postback khó khăn, …

Đầu năm 2002, Microsoft giới thiệu một kỹ thuật lập trình Web khá mới mẻ với tên gọi ban đầu là ASP+, tên chính thức sau này là ASP.Net. Với ASP.Net, không những không cần đòi hỏi bạn phải biết các tag HTML, thiết kế web, mà nó còn hỗ trợ mạnh lập trình hướng đối tượng trong quá trình xây dựng và phát triển ứng dụng Web.

ASP.Net là kỹ thuật lập trình và phát triển ứng dụng web ở phía Server (Server-side) dựa trên nền tảng của Microsoft .Net Framework.

Hầu hết, những người mới đến với lập trình web đều bắt đầu tìm hiểu những kỹ thuật ở phía Client (Client-side) như: HTML, Java Script, CSS (Cascading Style Sheets). Khi Web browser yêu cầu một trang web (trang web sử dụng kỹ thuật client-side), Web server tìm trang web mà Client yêu cầu, sau đó gởi về cho Client. Client nhận kết quả trả về từ Server và hiển thị lên màn hình.

ASP.Net sử dụng kỹ thuật lập trình ở phía server thì hoàn toàn khác, mã lệnh ở phía server (ví dụ: mã lệnh trong trang ASP) sẽ được biên dịch và thi hành tại Web Server. Sau khi được Server đọc, biên dịch và thi hành, kết quả tự động được chuyển sang HTML/JavaScript/CSS và trả về cho Client.Tất cả các xử lý lệnh ASP.Net đều được thực hiện tại Server và do đó, gọi là kỹ thuật lập trình ở phía server.[2]

***Ưu điểm của Asp.net***

* ASP.Net cho phép bạn lựa chọn một trong các ngôn ngữ lập trình mà bạn yêu thích: Visual Basic.Net, J#, C#, …
* Trang ASP.Net được biên dịch trước. Thay vì phải đọc và thông dịch mỗi khi trang web được yêu cầu, ASP.Net biên dịch những trang web động thành những tập tin DLL mà Server có thể thi hành nhanh chóng và hiệu quả. Yếu tố này là một bước nhảy vọt đáng kể so với kỹ thuật thông dịch của ASP.
* ASP.Net hỗ trợ mạnh mẽ bộ thư viện phong phú và đa dạng của .Net Framework, làm việc với XML, Web Service, truy cập cơ sở dữ liệu qua ADO.Net, …
* ASPX và ASP có thể cùng hoạt động trong 1 ứng dụng.
* ASP.Net sử dụng phong cách lập trình mới: Code behide. Tách code riêng, giao diện riêng -> Dễ đọc, dễ quản lý và bảo trì.
* Kiến trúc lập trình giống ứng dụng trên Windows.
* Hỗ trợ quản lý trạng thái của các control
* Tự động phát sinh mã HTML cho các Server control tương ứng với từng loại Browser  
  Hỗ trợ nhiều cơ chế cache.
* Triển khai cài đặt.
* Không cần lock, không cần đăng ký DLL.
* Cho phép nhiều hình thức cấu hình ứng dụng.
* Hỗ trợ quản lý ứng dụng ở mức toàn cục.
* Global.aspx có nhiều sự kiện hơn.
* Quản lý session trên nhiều Server, không cần Cookies.
* ASP.NET là một mô hình thiết kế website thống nhất bao gồm cách dịch vụ cần thiết để xây dựng các ứng dụng website doanh nghiệp lớn. ASP.NET là một phần của NET Framework.

***Visual Web Developer:***

* Visual Web Developer là một môi trường phát triển đầy đủ tính năng để tạo ra các ứng dụng website ASP.NET.
* Các tính năng: bố trí trang web phù hợp với các trang tổng thể và sự xuất hiện trang phù hợp với các chủ đề.
* Chỉnh sửa mã: trình biên tập cho phép bạn viết mã cho các website động trong Visual Basic hoặc C#.
* Kiểm tra và gỡ rối: Máy chủ web thử nghiệm và gỡ rối giúp bạn tìm thấy lỗi trong chương trình.
* Triển khai: Công cụ tự động hóa các nhiệm vụ cho việc triển khai một ứng dụng web đến một máy chủ lưu trữ hoặc nhà cung câp lưu trữ.
* Các website dự án và ứng dụng với ASP.NET. Sử dụng Visual 2012 có thể tạo ra các dự án khác nhau với ASP.NET, bao gồm các trang web, các ứng dụng web, các dịch vụ web, và điều khiển máy chủ AJAX. Có một sự khác biệt giữa các dự án website và các dự án ứng dụng website. Một số tính năng chỉ làm việc với các dự án ứng dụng web như công cụ MVC và một số đẻ tự động hóa việc triển khai web. Các tính năng khác như đồng bộ hóa dữ liệu, làm việc với trang web và các dự án ứng dụng web.

***ASP.NET API Reference***: Cung cấp một lượng lớn các thư viện dung cho ASP.NET.

***Trang web và các điều khiển***: là một chương trình chạy trên máy chủ tự động đưa ra các trang ASP.NET. Các trang ASP.NET có thể được yêu cầu từ bất kỳ trình duyệt nào hoặc các thiết bị của khách hàng. Theo quy định bạn có thể sử dụng cùng một website cho nhiều trình duyệt.ASP.NET là công cụ thiết kế website hoàn toàn hướng đối tượng. Nó cũng cho phép đóng gói các chức năng giao diện người dung thông thường dễ sử dụng, kiểm soát để có thể tái sử dụng. Ngoài ra các chủ đề cũng giúp bạn xác định trang chủ để tạo ra và bố trí phù hợp cho các website trong ứng dụng của bạn. Khung trang ASP.NET cũng cho phép bạn xác định mô hình các URL sẽ được sử dụng trong website của bạn. Điều này giúp tối ưu hóa công cụ tìm kiếm và làm cho các URL thân thiện với người sử dụng.

***Biên dịch ASP.NET:*** ASP.NET biên dịch, cho phép tối ưu hóa hiệu xuất, tối ưu hóa ràng buộc và những lợi ích khác. Một khi mã đã được biên dịch, thời gian chạy sẽ được cải thiện.

***Bảo Mật:*** ngoài các tính năng bảo mật cung cấp một sơ sở bảo mất tiên tiến để chứng thực và cho phép người dung truy cập vào các nhiệm vụ liên quan đến an ninh. Bạn có thể xác thực bằng cách sử dụng window xác thực được cung cấp bởi IIS hoặc quản lý xác thực bằng cơ sở dữ liệu người dùng riêng.

***Cấu hình ASP.NET:*** ASP.NET sử dụng một cấu hình hệ thống cho phép bạn xác định và thiết lập cấu hình cho máy chủ web của bạn. ASP.NET thiết lập cấu hình được lưu trữ trong các tập tin XML.

***Hỗ trợ gỡ lỗi:*** ASP.NET có lợi thế gỡ lỗi trong thời gian chạy để cung cấp quá ngôn ngữ và hỗ trợ gỡ lỗi qua máy tính. Bạn có thể gỡ lỗi cả hai đối tượng quản lý và không được quản lý. Web Service Framework ASP.NET hỗ trợ các dịch vụ web XML cho phép ứng dụng trao đổi thông tin qua bức tường lửa bằng cách sử dụng các tiêu chuẩn như HTTP và XML. Bất cứ ngôn ngữ lập trình web nào cũng có thể trao đổi thông tin và truy cập các dịch vụ web XML. Ứng dụng web dựa trên mô hình MVC: ASP.NET MVC giúp các nhà phát triển web xây dựng ứng dụng dựa trên tiêu chuẩn được dễ dàng duy trì bởi nó giảm sự phụ thuộc vào các class.

***ASP.NET sử dụng dữ liệu động:*** ASP.NET sử dụng dữ liệu động cho phép bạn tạo ra các ứng dụng web hướng dữ liệu một cách dễ dàng. Bạn sẽ tìm thấy rằng bạn có thể nhận được các ứng dụng chạy dễ dàng hơn và với ít mã hơn so với trước đây. Đồng thời bạn có thể thêm các tính năng tùy chỉnh để các dữ liệu ASP.NET động để thích ứng với yêu cầu riêng của bạn [3].

### ASP.NET IDENTITY

[*ASP.NET*](http://asp.net/) Identity là một thành phần (built-in) của [*ASP.NET*](http://asp.net/), nó cung cấp cho bạn các tính năng đầy đủ và đa dạng về authentication. Có thể như: Tạo tài khoản, login với user name và password, cập nhật profile. Hoặc cũng có thể sử dụng những provider bên ngoài giống như: Facebook, Google, Twitter, Microsoft account và nhiều cái khác nữa. Bạn có thể cấu hình [*ASP.NET*](http://asp.net/) để sử dụng với SQL Server nhằm lưu trữ user name, password và dữ liệu profile.

Cấu trúc tổ chức thư mục của ASP.NET Identity:

* **App\_Start/IdentityConfig.cs**: chứa các lệnh để cấu hình ASP.NET Identity
* **Controller/AccountController**: controller chứa các action method có tác dụng xác thực người dùng như Login, Register, ForgotPassword, ...
* **Controller/ManageController**: controller chứa các action method có tác dụng quản lý user (khi user đã login vào web) như ChangePassword, SetPassword, ...
* **Model/AccountViewModels**: chứa các View Model hiển thị trong các view của AccountController
* **Model/ManageViewModels**: chứa các View Model hiển thị trong các view của ManageController
* **Model/IdentityModels**: chứa class**ApplicationUser** để quản lý thông tin user và class**ApplicationDbContext**để quản lý kết nối với database ở dạng **Entity Framework Code First** (các bạn nên có kiến thức căn bản về Entity Framework Code First) để có thể bổ sung thêm các field cho user hoặc loại bỏ bớt các field mà bạn không cần thiết 1 cách dễ dàng và ít bỡ ngỡ [4] .

## Mô hình MVC

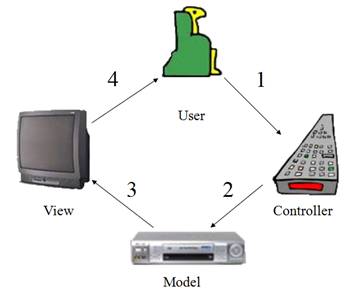
**Khái niệm về mô hình MVC**: Ý niệm chính của mô hình này là tách biệt phần ánh xạ, lưu trữ và xử lý dữ liệu (model) tách biệt hoàn toàn với thành phần trình bày giao diện kết quả cho người dùng hay phần giao diện giúp đón nhập nhập xuất cho người dùng (View). Ý niệm trên cho phép người lập trình có thể tách biệt công việc trong quá trình xây dựng chức năng cho ứng dụng và quá trình xây dựng giao diện cho người dùng.

Bên cạnh đó, ý niệm trên cho phép việc thay đổi thành phần của dữ liệu (model) sẽ không ảnh hưởng nhiều đến giao diện của người dùng vì mô hình đưa ra Model để không cho người dùng thao tác trực tiếp vào dữ liệu vật lý (Cơ sở dữ liệu hay là tập tin) mà phải thông qua Model, do vậy cho dù dữ liệu vật lý thay đổi cấu trúc nhưng cấu trúc Model cho việc truy cập, xử lý, lưu trữ dữ liệu sẽ không bị ảnh hưởng. Nhìn theo khái niệm các thành phần giao tiếp trên Model là tên hàm – tham số truyền (interface) ít khi thay đổi, nội dung thay đổi chính là cách thức cài đặt bên trong hàm. Nhưng nội dung đó người sử dụng chức năng trên giao diện không quan tâm vì đa số họ chỉ quan tâm interface là gì, giá trị nhập và kết xuất ra sao. Do vậy, đây là một trong tính linh hoạt và uyển chuyển của mô hình MVC

Ngoài ra, việc tách biệt rời rạc giữa Model và View theo phân tích của chúng ta đang thể hiện tính uy việt. Tuy nhiên, một ứng dụng có rất nhiều Model và nhiều View, do vậy, mô hình cần có một thành phần lựa chọn và kết nối các thành phần này lại với nhau theo cách hiệu quả nhất. Controller là một trong những đối tượng đưa ra để đón nhận yêu cầu nhập xuất từ người dùng, xác định model tương ứng với view nhập để đưa model xử lý, kết quả xử lý của model sẽ được chuyển đến controller để controller xác định view kết xuất để đổ kết quả xử lý và hiển thị cho người dùng.

Để hiểu rõ hơn về mô hình MVC này, tôi chủ quan đưa ra một mô hình thực tế tiếp cận khái niệm của mô hình này như sau:

* + Mô hình tiếp cận của chúng tôi được mô tả về việc công việc hàng ngày chúng ta thường làm để giải trí, đó là xem truyền hình. Câu chuyện được mô tả như sau: Một người A muốn xem các phim chọn lọc được lựa chọn từ các đĩa CD/DVD trên truyền hình của nhà A. Để đọc được các đĩa CD/DVD thì phải dùng đầu máy đọc đĩa trong nhà A. Đầu máy này đặc biệt không có nút điều khiển trực tiếp, đầu máy này nhận hàng loạt các đĩa. Tối đa là 5 cái, do vậy muốn kích hoạt và chọn đĩa xem A bắt buộc phải sử dụng remote control – điều khiển từ xa. Như vậy, chúng ta sẽ nhận thấy cách thức A xem phim như sau:
    - A sẽ sử dụng remote control điều khiển đầu máy để chọn đĩa mà A thích xem nhất để mở nó ra
    - Đĩa được chọn từ đọc máy được đọc và trình chiếu trên màn hình truyền hình
    - Khi dữ liệu đã hiển thị trên màn hình truyền hình thì A sẽ thấy được nội dung
    - Và cách thức này sẽ được lập đi lập lại khi A chọn một phim khác, đó là vẫn dùng đồ điều khiển để chọn đĩa và đầu đĩa đưa dữ liệu lên truyền hình và A thấy.
  + Mô hình tiếp cận được mô tả như hình vẽ dưới:



Hình 1.1 Mô hình MVC

* Qua hình ảnh mô tả trên, chúng ta nhận thấy rằng.
  + - Đầu máy là nơi xử lý dữ liệu, chọn lựa các thức xử lý, nội dung cần thiết nghĩa là đầu máy là Model.
    - Truyền hình chỉ làm nhiệm vụ duy nhất để trình bày kết quả mà đầu máy – Model đã thực hiện, được lựa chọn. Truyền hình không thể lựa chọn và không có cách chọn lựa là trình bày các thành phần truyền đến đã được xử lý. Truyền hình là View.
    - Thành phần hỗ trợ đưa dữ liệu từ Model đến View đó là đồ điều khiển, ngoài ra đồ điểu khiển cũng là nơi kết nối người dùng với đầu máy với truyền hình. Chức năng của điều khiển là chọn đúng model để đưa ra view. Đồ điều khiển – remote control là một Controller.
  + Dựa trên một cách tiếp cận chúng ta thấy được sự tối ưu của MVC đó là:
    - Khi chúng ta thay đổi Truyền hình đang xem thành truyền hình khác thì việc thay đổi không ảnh hưởng gì đến toàn hệ thống.
    - Tương tự như thế nếu đầu máy hư thì thay đổi máy khác vẫn không ảnh hưởng gì lớn.
    - Như vậy, việc tách biệt giúp chúng ta có nhiều lợi ích trong việc phân chia công việc và các xử lý công việc.
  + Dựa trên cách tiếp cận như trên chúng ta định nghĩa các thành phần trong mô hình MVC như sau:
    - Model: Là các thành phần hỗ trợ ánh xạ dữ liệu vật lý lên bộ nhớ, lưu trữ dữ liệu tạm thời trên bộ nhớ, hỗ trợ các cách thức xử lý dữ liệu, hỗ trợ khả năng giao tiếp và trao đổi dữ liệu giữa các đối tượng khác trong bộ nhớ và cơ sở dữ liệu. Cụ thể là một đối tượng Object trong khái niệm của lập trình hướng đối tượng OOP và mang đầy đủ khái niệm và tính chất của một Object. Trong ứng dụng Web của Java, Model sẽ là JavaBean hay Enterprise JavaBean hay Web Service
    - View: Là thành phần hỗ trợ trình bày dữ liệu hay kết quả ra màn hình, hỗ trợ nhập thông tin từ phía người dùng. Các thành phần này có khả năng truy cập Model, truy xuất Model thông qua những hành vi mà Model cho phép nhưng View không thể thay đổi các thành phần trong Model. Trong mô hình ứng dụng Web thì html, servlet, jsp … là những thành phần đại diện cho View
    - Controller: Là các thành phần hỗ trợ kết nối người dùng server, đón nhập yêu cầu người dùng, thực hiện chuyển xử lý, lựa chọn và cập nhật model và view tương ứng để trình bày về phía người dùng. Hỗ trợ kết nối giữa người model và view, giúp model xác định được view trình bày. Trong mô hình ứng dụng Web thì Servlet đóng vai trò của Controller
  + Mối quan hệ giữa các thành phần trong mô hình MVC:
    - Dựa trên bản chất của MVC, khái niệm và cách tiếp cận chúng ta nhận thấy mối quan hệ giữa controller và model là như thế nào? Giữa Controller và Model, đối tượng nào phụ thuộc đối tượng nào? Ý tưởng đó là khi chúng ta thay đổi đầu máy đọc, đồ điều khiển từ xa có thay đổi theo hay không? Hay chúng ta vẫn dùng cái điều khiển cũ (không tính khái niệm điều khiển đa năng)? Đồ điều khiển được thiết kế dựa trên đầu máy hay đầu máy được thiết kế dựa trên đồ điều khiển? Câu trả lời chúng ta quá rõ ràng, đó là controller phụ thuộc vào model bởi vì controller được thiết kế đề kết nối với model, điều khiển, truy xuất model. Do vậy, model có sự thay đổi là controller phải thay đổi theo để có thể truy xuất và điều khiển model cho phù hợp
    - Giữa Model và View trên khái niệm, đối tượng nào phụ thuộc đối tượng nào? View lệ thuộc vào Model bởi vì khi các tổ chức interface của Model thay đổi nghĩa là View phải thay đổi theo. Chúng ta tránh nhầm lẫn về khái niệm ở trên là tách biệt giữa View và Model là có lợi nhưng tại sao ở đây View lại lệ thuộc Model? Khái niệm của chúng ta ở đây đó là tách biệt dữ liệu và thành phần xử lý bên trong, hệ thống sẽ trở nên uyển chuyển khi chúng ta thành đổi thành phần cài đặt trong Model, không phải thay đổi interface Model. Và khái niệm trên đã nêu rõ giữa dữ liệu vật lý và dữ liệu được chứa trong Model. Mối quan hệ cuối cùng của chúng ta đó là giữa View và Controller. Controller sẽ là đối tượng lựa chọn View tùy thuộc theo việc yêu cầu sử dụng của người dùng, tùy thuộc theo kết quả xử lý của Model .

## Model

### Giới thiệu về SQL Server 2012

Trong một thế giới dữ liệu ngày nay, dữ liệu và các hệ thống quản lý dữ liệu đó cần phải luôn luôn được bảo đảm và ở trạng thái có sẵn. SQL Server 2012 cho phép các. Trong một thế giới dữ liệu ngày nay, dữ liệu và các hệ thống quản lý dữ liệu đó cần phải luôn luôn được bảo đảm và ở trạng thái có sẵn. SQL Server 2012 cho phép các nhà phát triển giảm được sự phức tạp của cơ sở hạ tầng trong khi đó vẫn bảo đảm cung cấp một nền tảng dữ liệu doanh nghiệp có khả năng bảo mật, khả năng mở rộng và quản lý tốt hơn, cùng với thời gian chết của ứng dụng giảm.

Những điểm mới của SQL Server 2012: giới thiệu 4 lĩnh vực chính trong toàn cảnh nền tảng dữ liệu của Microsoft:

* Nền tảng cho các nhiệm vụ then chốt - SQL Server 2012 cho phép các tổ chức có thể chạy hầu hết các ứng dụng phức tạp của họ trên một nền tảng an toàn, tin cậy và có khả năng mở rộng, bên cạnh đó còn giảm được sự phức tạp trong việc quản lý cơ sở hạ tầng dữ liệu. SQL Server 2012 cung cấp một nền tảng tin cậy và an toàn bằng cách bảo đảm những thông tin có giá trị trong các ứng dụng đang tồn tại và nâng cao khả năng sẵn có của dữ liệu. SQL Server 2012 giới thiệu một cơ chế quản lý cách tân dựa trên chính sách, cơ chế này cho phép các chính sách có thể được định nghĩa quản trị tự động cho các thực thể máy chủ trên một hoặc nhiều máy chủ. Thêm vào đó, SQL Server 2012 cho phép thi hành truy vấn dự báo với một nền tảng tối ưu.
* Sự phát triển động - SQL Server 2012 cùng với .NET Framework đã giảm được sự phức tạp trong việc phát triển các ứng dụng mới. ADO.NET Entity Framework cho phép các chuyên gia phát triển phần mềm có thể nâng cao năng suất bằng làm việc với các thực thể dữ liệu logic đáp ứng được các yêu cầu của doanh nghiệp thay vì lập trình trực tiếp với các bảng và cột. Các mở rộng của ngôn ngữ truy vấn tích hợp (LINQ) mới trong .NET Framework đã cách mạng hóa cách các chuyên gia phát triển truy vấn dữ liệu bằng việc mở rộng Visual C#® và Visual Basic® .NET để hỗ trợ cú pháp truy vấn giống SQL vốn đã có. Hỗ trợ cho các hệ thống kết nối cho phép chuyên gia phát triển xây dựng các ứng dụng cho phép người dùng mang dữ liệu cùng với ứng dụng này vào các thiết bị và sau đó đồng bộ dữ liệu của chúng với máy chủ trung tâm.
* Dữ liệu quan hệ mở rộng - SQL Server 2012 cho phép các chuyên gia phát triển khai thác triệt để và quản lý bất kỳ kiểu dữ liệu nào từ các kiểu dữ liệu truyền thống đến dữ liệu không gian địa lý mới.
* Thông tin trong toàn bộ doanh nghiệp - SQL Server 2012 cung cấp một cơ sở hạ tầng có thể mở rộng, cho phép quản lý các báo cáo, phân tích với bất kỳ kích thước và sự phức tạp nào, bên cạnh đó nó cho phép người dùng dễ dàng hơn trong việc truy cập thông tin thông qua sự tích hợp sâu hơn với Microsoft Office. Điều này cho phép CNTT đưa được thông tin của doanh nghiệp rộng khắp trong tổ chức. SQL Server 2012 tạo những bước đi tuyệt vời trong việc lưu trữ dữ liệu, cho phép người dùng hợp nhất các trung tâm dữ liệu vào một nơi lưu trữ dữ liệu tập trung của toàn doanh nghiệp.

QL Server 2012 cung cấp công nghệ và khả năng mà các tổ chức hy vọng kiểm soát được các khó khăn thách thức đang ngày càng tăng đối với việc quản lý dữ liệu và cung cấp thông tin có giá trị kịp thời đến người dùng. Với những cải thiện đáng kể trong các lĩnh vực chính của nền tảng nhiệm vụ then chốt, phát triển động, dữ liệu quan hệ mở rộng và truyền lan tin tức, các lợi ích của SQL Server 2012 thể hiện ở đây là rất có giá trị. SQL Server 2012 là một bộ phận trong toàn cảnh về nền tảng dữ liệu của Microsoft được thiết kế cho việc quản lý và làm việc với dữ liệu ngày nay là xa hơn nữa.

SQL Server 2012 là một phát hành quan trọng mang đến nhiều tính năng mới và những cải thiện quan trọng làm cho nó trở thành một phiên bản SQL Server toàn diện và mạnh mẽ nhất với những cần thiết trong việc bùng nổ dữ liệu ngày nay.

### Các Thành Phần Của Một CSDL Trong SQL Server:

**Tables:** Table là đối tượng chính của CSDL dùng lưu trữ dữ liệu cần quản lý. Mỗi table có 1 hay nhiều Field. Mỗi Field ứng với một loại dữ liệu cần lưu trữ.

Table còn có các thành phần liên quan như:

* Constraint – Ràng buột: Constraint là các chỉ định ràng buột dữ liệu trong bảng hoặc các bảng khác nhau theo một quy tắc nào đó.
* Triggers – Bẫy Lỗi: Trigger thường chứa các mã lệnh kiểm tra dữ liệu, có tính năng tự động thực hiện khi có hành động nào đó xảy ra đối với dữ liệu trong Table như Insert, Update, Delete.
* Indexs – Chỉ mục: Hổ trợ cho việc sắp xếp và tìm kiếm nhanh thông tin trên table.

**Diagram** – Sơ đồ quan hệ: Thể hiện mối quan hệ dữ liệu giữa các table.

**Views** – Khung nhìn hay table ảo: Là đối tượng dùng hiển thị dữ liệu được rút trích, tính toán từ các Table theo nhu cầu của người dùng.

**Stored Procedure** – Thủ tục nội: Chứa các lệnh T-SQL dùng thực hiện một số tác vụ nào đó. Stored Proc có thể nhận và truyền tham số. Stored Proc được biên dịch trước, do đó thời gian thực hiện nhanh khi được gọi. Có nhiều Stored Proc hệ thống được định nghĩa với tiền tố “sp\_” có nhiệm vụ thu thập thông tin từ các bảng hệ thống và rất có ích cho việc quản trị.

**User Defined Function**: Hàm do người dùng định nghĩa

**Users**: Chứa danh sách User sử dụng CSDL. Người quản trị hệ thống cao nhất có User Name là dbo, tên đăng nhập (Login Name) hệ thống mặc định là sa. Tài khoản sa luôn tồn tại và không thể bỏ đi. Để thay đổi mật khẩu của sa, cách nhanh nhất là:

* Mở trình Query Analyzer
* Thực hiện thủ tục hệ thống: EXEC SP\_PASSWORD NULL, <NewPass>

**Roles**: Các qui định vai trò và chức năng của User trong hệ thống SQL Server

**Rules**: Các qui tắc ràng buộc dữ liệu được lưu trữ trên Table

**Defaults**: Các khai báo giá trị mặc định

**User Defined Data Type**: Kiểu dữ liệu do người dùng tự định nghĩa

**Full Text Catalogs**: Tập phân loại dữ liệu Text.

### Các câu lệnh cơ bản trong SQL SERVER

SQL được theo sau bởi tập hợp duy nhất các qui tắc và nguyên tắc chỉ đạo được gọi là Cú pháp SQL. SQL chuẩn bao gồm khoảng 40 câu lệnh. Bảng sau đây liệt kê các lệnh SQL thường được sử dụng nhất trong số các câu lệnh của SQL Server:

Bảng 1.1 Bảng câu lệnh thao tác dữ liệu

|  |  |
| --- | --- |
| **Câu** **lệnh thao tác** **dữ** **liệu** | **Chức** **năng** |
| SELECT | Truy xuất dữ liệu |
| INSERT | Bổ sung dữ diệu |
| UPDATE | Cập nhật dữ liệu |
| DELETE | Xoá dữ liệu |
| TRUNCATE | Xoá toàn bộ dữ liệu trong bảng |

Bảng 1.2 Bảng câu lệnh định nghĩa dữ liệu

|  |  |
| --- | --- |
| **Câu** **lệnh định nghĩa dữ liệu** | **Chức** **năng** |
| CREATE TABLE | Tạo bảng |
| DROP TABLE | Xoá bảng |
| ALTER TABLE | Sửa đổi bảng |
| CREATE VIEW | Tạo khung nhìn |
| ALTER VIEW | Sửa đổi khung nhìn |
| DROP VIEW | Xoá khung nhìn |
| CREATE INDEX | Tạo chỉ mục |
| DROP INDEX | Xoá chỉ mục |
| CREATE SCHEMA | Tạo lược đồ cơ sở dữ liệu |
| DROP SCHEMA | Xoá lược đồ cơ sở dữ liệu |
| CREATE PROCEDURE | Tạo thủ tục lưu trữ |
| ALTER PROCEDURE | Sửa đổi thủ tục lưu trữ |
| DROP PROCEDURE | Xoá thủ tục lưu trữ |
| CREATE FUNCTION | Tạo hàm (do người sử dụng định nghĩa) |
| ALTER FUNCTION | Sửa đổi hàm |
| DROP FUNCTION | Xoá hàm |
| CREATE TRIGGER | Tạo trigger |
| ALTER TRIGGER | Sửa đổi trigger |
| DROP TRIGGER | Xoá trigger |

Bảng 1.3 Bảng câu lệnh điều kiển truy cập

|  |  |
| --- | --- |
| **Câu** **lệnh điều khiển truy cập** | **Chức** **năng** |
| GRANT | Cấp phát quyền cho người sử dụng |
| REVOKE | Thu hồi quyền từ người sử dụng |

Bảng 1.4 Bảng câu lệnh quản lý giao dịch

|  |  |
| --- | --- |
| **Câu** **lệnh quản lý giao dịch** | **Chức** **năng** |
| COMMIT | Uỷ thác (kết thúc thành công) giao dịch |
| ROLLBACK | Quay lui giao dịch |
| SAVE TRANSACTION | Đánh dấu một điểm trong giao dịch |

Bảng 1.5 Bảng câu lệnh lập trình

|  |  |
| --- | --- |
| **Câu** **lệnh lập trình** | **Chức** **năng** |
| DECLARE | Khai báo biến hoặc định nghĩa con trỏ |
| OPEN | Mở một con trỏ để truy xuất kết quả truy vấn |
| FETCH | Đọc một dòng trong kết quả truy vấn (sử dụng con trỏ) |
| CLOSE | Đóng một con trỏ |
| EXECUTE | Thực thi một câu lệnh SQL |

# QUY TRÌNH THIẾT KẾ

## Yêu cầu chức năng

Phân hệ quản lý người dùng được xây dựng với các chức năng cơ bản sau đây:

* Người dùng thông thường: Đồng bộ với người dùng trên Smartphone, có thể xem, bình luận, tạo địa điểm và quản trị địa điểm của mình.
* Người quản trị: Quản trị hệ thống cho phép tạo nhóm người dùng, cấp quyền cho nhóm người với từng module của hệ thống. Nhóm người dùng cao nhất có mọi quyền với các tính năng của hệ thống. Cho phép thêm, sửa, xóa, lấy lại mật khẩu cho các người dùng khác. Tùy thuộc vào quyền truy cập mà quản trị câp cho, nhóm người dùng có thể xem, sửa, xóa dữ liệu.

## Yêu cầu phi chức năng

Ngoài ra, phân hệ quản lý người phải đáp ứng các yêu cầu phi chức năng sau đây:

* Người dùng được phân cấp truy nhập hệ thống để đảm bảo an toàn cho hệ thống máy chủ, Database Server và các máy chủ ứng dụng.
* Đảm bảo tính chính xác bộ dữ liệu mẫu.
* Giao diện dễ nhìn, dễ sử dụng, dữ liệu trả về trong thời gian ngắn đối với các tính năng như tìm kiếm, hiển thị thông tin sự vật, địa điểm,...
* Có sự liên kết mềm dẻo giữa các tính năng của hệ thống.

## Quản trị người dùng trên Website

### Sơ đồ chức năng

Quản lý người dùng bao gồm các chức năng quản lý user (danh sách các người dùng trong hệ thống), quản lý quyền (danh sách các quyền được phân cấp cho user), chức năng quản lý nhóm người dùng (các mức phân cấp người dùng)



Hình 2.1 Sơ đồ chức năng quản lý người dùng

Chức năng quản lý tài khoản người sử dụng chỉ được thực hiện bởi tài khoản cấp admin. QTV có quyền tạo mới người dùng cấp hộ dân, phân quyền và gán quyền cho người dùng cấp admin. Đồng thười QTV cũng có quyền xóa hoặc khóa tài khoản người sử dụng cấp hộ dân. Chức năng quản lý tài khoản người sử dụng được mô tả trên Hình 2.1

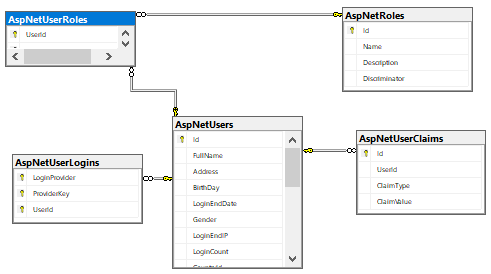
### Sơ đồ chuẩn hoá cơ sở dữ liệu phân quyền người dùng

Người dùng trong hệ thống được chia thành nhiều lớp với các chức năng và quyền hạn khác nhau được quy định bởi quản trị hệ thống.

***Người dùng thông thường:*** Người dùng có thể xem các thông tin về các vị trí, tour du lịch mà hệ thống cập nhật.

***Doanh nghiệp***: được phép quản trị trang riêng, ví dụ quản trị một nhà hàng.

***Quản trị hệ thống:*** Người dùng quản trị hệ thống. Tương tác với trang quản trị để thêm, sửa hoặc xóa dữ liệu phụ thuộc vào quyền hạn được cấp.



Hình 2.2 Sơ đồ chuẩn hóa cơ sở dữ liệu cho tính năng phân quyền.

Hình 2.2 là cơ sở dữ liệu phân quyền người dùng sau khi chuẩn hoá 3NF

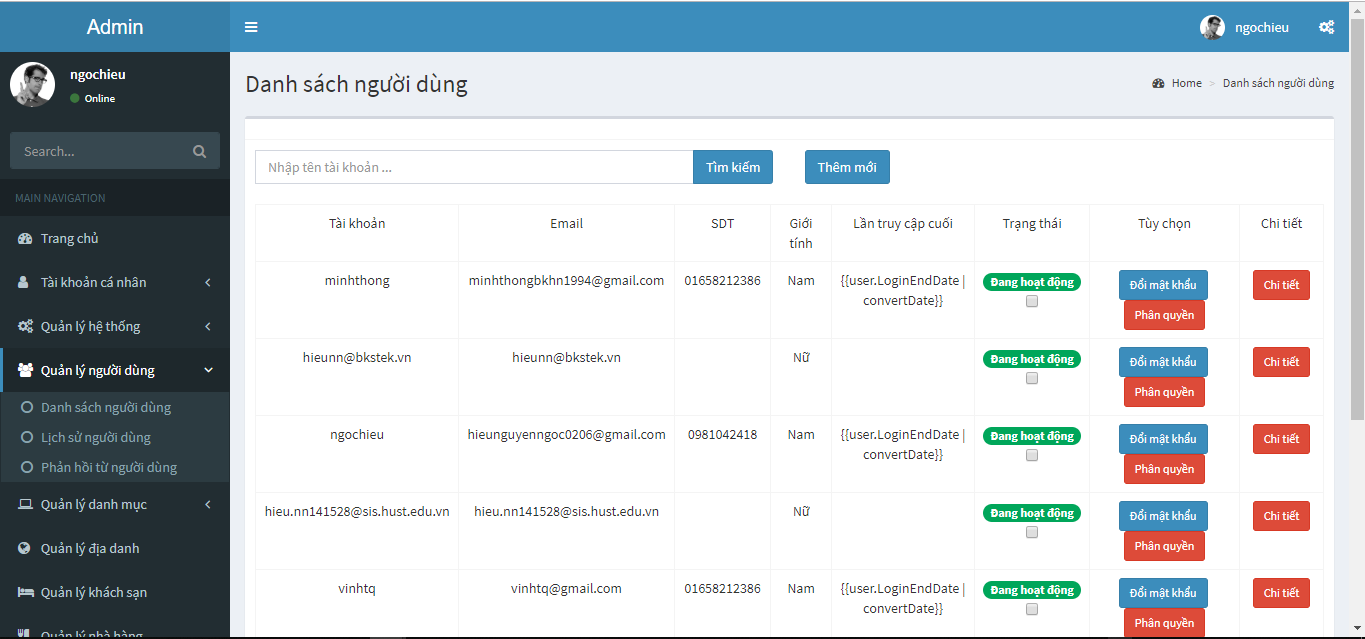
Bảng 2.1 Bảng kê chức năng từng bảng trong cơ sở dữ liệu phân quyền

|  |  |
| --- | --- |
| **Bảng** | **Chức năng của bảng** |
| AspNetUserRoles | Lưu trữ tài khoản cùng với quyền của nó |
| AspNetRoles | Lưu trữ nhóm quyền |
| AspNetUsers | Lưu trữ tài khoản |
| AspNetUserClaims | Lưu trữ các thông tin người dùng sau khi đăng nhập |
| AspNetUserLogin | Lưu trữ các thông tin đăng nhập |

# TRIỂN KHAI

## Quy trình

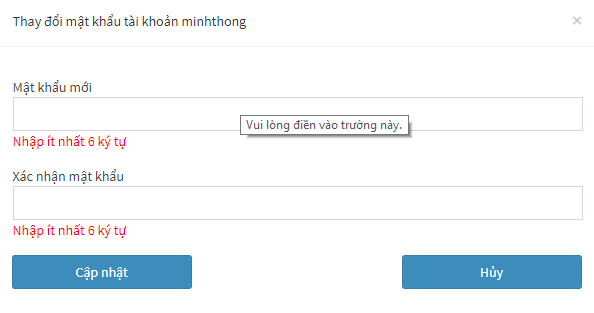
|  |  |  |
| --- | --- | --- |
| **STT** | **Tài khoản** | **Diễn giải** |
| 1 | Tài khoản Admin | Admin đăng nhập bằng tài khoản admin được cấp tất cả các quyền quản trị hệ thống, một trong số đó có quyền quản lý người dùng |



Hình 3.1 Chức năng chung quản lý người dùng

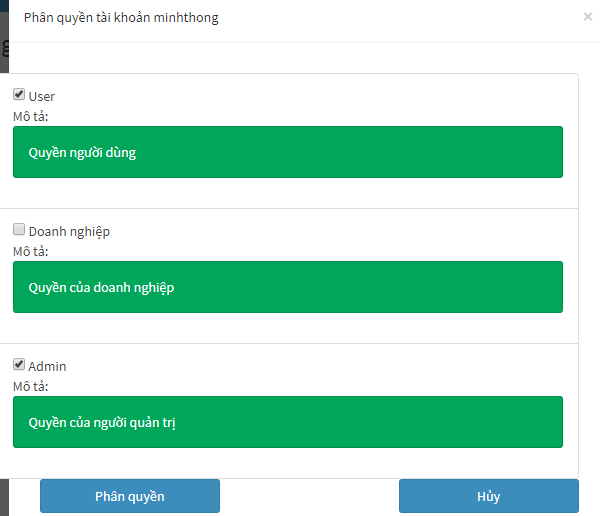
Chức năng quản lý người dùng có nhũng chức năng chính là: Danh sách người dùng hệ thống, lịch sử của người dử dụng, phản hồi của người sử dụng

|  |  |  |
| --- | --- | --- |
| **STT** | **Tài khoản** | **Diễn giải** |
| 1 | Tài khoản Admin | Tài khoản admin có quyền đổi mật khẩu, phân quyền cho các tài khoản user |
| 2 | Tài khoản User | Tài khoản User không có quyền vào phần quản trị |



Hình 3.2 Giao diện đổi mật khẩu tài khoản cho user và doanh nghiệp

Chức năng đổi mật khẩu có 2 ô nhập cho admin và sẽ kiểm tra đúng quy định cho mật khẩu hay không, nếu không đúng sẽ báo cho admin biết và không cho cập nhập mật khẩu.



Hình 3.3 Giao diện phân quyền cho tài khoản user và doanh nghiệp

Chức năng phân quyền cho các tài khoản sẽ được thêm các quyền theo giao diện như trên Hình 3.3. Quyền người dùng sẽ chỉ được vào giao diện web, quyền doanh nghiệp sẽ chỉ được chỉnh sửa những dịch vụ mà đã đăng kí với hệ thống, quyền của người quản trị sẽ được chỉnh sửa tất cả chức năng của hệ thống

## Nội dung thực hiện

* Lấy tất cả tài khoản trong hệ thống từ cơ sở dữ liệu



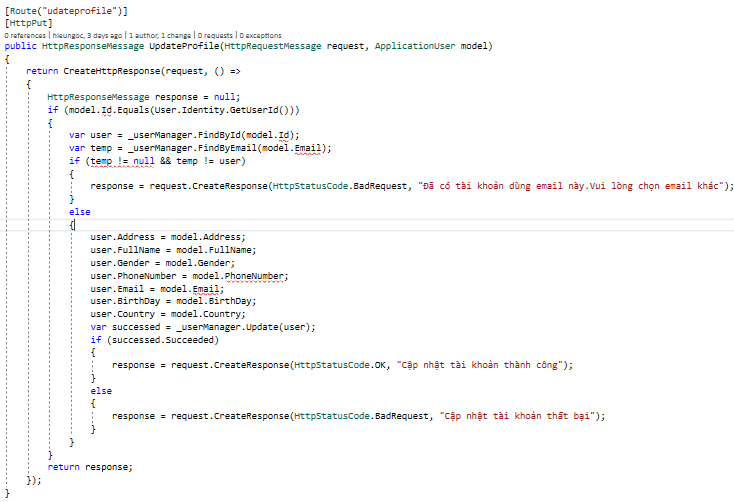
Hình 3.4 API lấy tất cả tài khoản trong hệ thống

* Xem thông tin chi tiết tài khoản



Hình 3.5 API xem thông tin chi tiết tài khoản

* Chức năng cập nhật tài khoản



Hình 3.6 API cập nhập tài khoản

* Chức năng đổi mật khẩu



Hình 3.7Api đổi mật khẩu tài khoản

* Chức năng phân quyền cho tài khoản



Hình 3.8 API Phân quyền cho tài khoản

# KẾT LUẬN

Trong tài liệu này, nhóm nghiên cứu đã trình bày kết quả nội dung công việc “Xây dựng phân hệ quản lý người dùng”. Kết quả của công việc này là cơ sở để tạo nên một website chuẩn mực, hoàn thiện và dễ bảo trì.

# TÀI LIỆU THAM KHẢO

[1] “ASP.NET - Introduction.” [Online]. Available: https://www.tutorialspoint.com/asp.net/asp.net\_introduction.htm. [Accessed: 06-Nov-2018].

[2] “Get Started with ASP.NET Web API 2 (C#),” *docs.microsoft.com*.

[3] *Announcing the .NET Framework 4.7*. .

[4] “[ASP.NET Identity] – 1 – Introduction | Tuan Tran’s Blog.” [Online]. Available: https://coding4food.net/2017/08/30/aspnet-identity-1-introduction/. [Accessed: 06-Nov-2018].