|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC HOA SEN**  **KHOA KHOA HỌC & CÔNG NGHỆ** |
|  |

Tên đề tài : Chia sẻ thông tin địa điểm

**Giảng viên bộ môn : Võ Hoàng Hải**

**Lớp : HT330DV01**

**Khóa : QL08**

**Nhóm sinh viên thực hiện : Nguyễn Bá Hiệp**

**Huỳnh Hữu Tạo**

**Trần Đại Sơn**

***Tp.HCM, năm 2011***

Lời nhận xét của giảng viên

Mục lục

[1. Tìm hiểu lý thuyết 2](#_Toc311723261)

[1.1. Giới thiệu Web service & SOA 2](#_Toc311723262)

[1.1.1. Tổng quan Web service 2](#_Toc311723263)

[1.1.2. Tổng quan SOA 3](#_Toc311723264)

[1.2. Các nguyên tắc thiết kế 3](#_Toc311723265)

[1.2.1. Nhóm nguyên lý thực thi 3](#_Toc311723268)

[1.2.2. Nhóm nguyên lý điều chỉnh 3](#_Toc311723272)

[1.3. Standardized Service contract 3](#_Toc311723273)

[1.4. Service Reusability 4](#_Toc311723274)

[1.5. Service Autonomy 4](#_Toc311723275)

[1.6. Service Statelessness 5](#_Toc311723276)

[1.7. Service Discoverability 6](#_Toc311723277)

[1.8. Service Loose Coupling 7](#_Toc311723278)

[1.9. Service Abstraction 7](#_Toc311723279)

[1.10. Service Composability 8](#_Toc311723280)

[2. Thiết kế chương trình 10](#_Toc311723281)

[2.1. Đối tượng phục vụ 10](#_Toc311723282)

[2.2. Các chức năng chính của chương trình 10](#_Toc311723283)

[2.2.1. ThemThongTin (Thêm thông tin) 10](#_Toc311723284)

[2.2.2. Rating (Rating cho địa điểm) 11](#_Toc311723285)

[2.2.3. TimThongTin (tìm kiếm lấy thông tin) 11](#_Toc311723286)

[2.2.4. ThemHinhAnh (Thêm hình ảnh) 12](#_Toc311723287)

[2.2.5. Báo cáo địa điểm sai 12](#_Toc311723288)

[2.2.6. Thay đổi thông tin 13](#_Toc311723289)

[2.2.7. Xóa thông tin 14](#_Toc311723290)

[2.2.8. Đăng nhập 14](#_Toc311723291)

[2.3. Mô tả cơ sở dữ liệu 16](#_Toc311723292)

[2.3.1. Bảng Table\_DiaDiem 16](#_Toc311723296)

[2.3.2. Bảng PhanLoai 17](#_Toc311723297)

[2.3.3. Bảng LinhVuc 17](#_Toc311723298)

[2.3.4. Bảng BaoCaoSai 17](#_Toc311723299)

[2.3.5. Bảng tài khoản 18](#_Toc311723300)

[3. Hiện thực Chương Trình 19](#_Toc311723301)

Lời mở đầu

Với tốc độ phát triển của xã hội Việt Nam như hiện nay thì thông tin là một điều khá là quan trọng. Với việc tìm kiếm thông tin về một địa điểm chính xác thì rất là hữu ích với mọi người. Nhưng để có được thông tin trong một thành phố hoặc cả nước và thông tin phải chính xác là một điều hết sức khó khăn. Vì vậy nhóm phát triển phần mềm “chia sẻ thông tin địa điểm” nhằm kêu gọi mọi người cùng nhau đóng góp các thông tin chính xác về các địa điểm gần khu vực mà họ sống để giúp đỡ cho nhưng người sử dụng “tìm kiếm địa điểm”. Với phần mềm này mọi người có điện thoại và có phần mềm của nhóm thì chức năng phần mềm người dùng chỉ việc chụp và điền thông tin chính xác về địa điểm. Khi đó trang web của nhóm với sự hỗ trợ của webservice sẽ đưa thông tin về địa điểm đó cho người sử dụng xem.

Từ những yêu cầu nêu trên nhóm đã quyết định chọn đề tài ”**Xây dựng Web service tra cứu địa điểm**”.

Nhóm đã phân tích và đưa ra một số chức năng cơ bản như sau:

Quản lý thông tin:

* Nhận thông tin
* Rating cho địa điểm
* Lấy thông tin địa điểm
* Thêm hình ảnh
* Báo cáo sai
* Thay đổi thông tin
* Xóa thông tin
* Login

Thành viên nhóm :

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | Mã số sinh viên | Email | Điện thoại |
| Nguyễn Bá Hiệp | 080824 | [Hiep.nb0824@sinhvien.hoasen.edu.vn](mailto:Hiep.nb0824@sinhvien.hoasen.edu.vn) | 0908828061 |
| Huỳnh Hữu Tạo | 080887 | [Tao.hh0887@sinhvien.hoasen.edu.vn](mailto:Tao.hh0887@sinhvien.hoasen.edu.vn) | 0908182013 |
| Trần Đại Sơn | 080881 | [Son.td0881@sinhvien.hoasen.edu.vn](mailto:Son.td0881@sinhvien.hoasen.edu.vn) | 01267856772 |

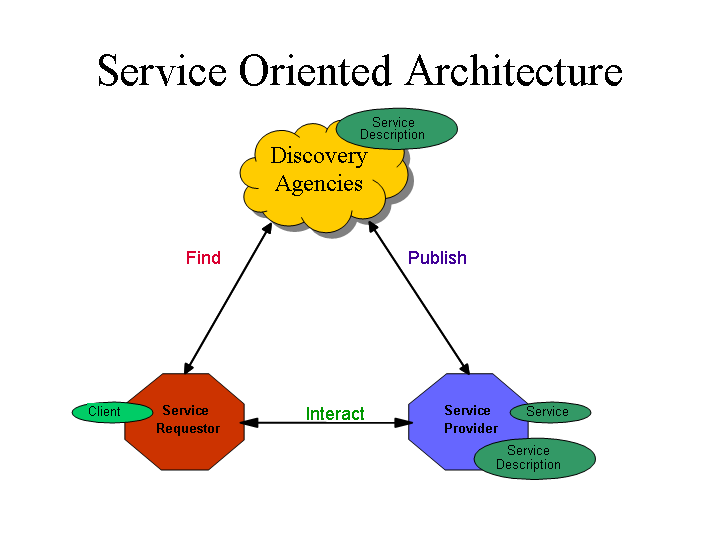
Nội dung

# Tìm hiểu lý thuyết

## Giới thiệu Web service & SOA

### Tổng quan Web service

Web service phát triển dựa trên khái niệm căn bản của quá trình phân bố các components. Trong đó tất cả các hoạt động sẽ được thực hiện thông qua Internet.

****

Hình 1.1: Kiến trúc của Web service

Web service gồm 3 thành phần chính:

SOAP: là giao thức truyền dữ liệu được xây dựng dựa trên XML, HTTP, và SMTP.

WSDL: là các lược đồ XML được dùng để mô tả các chức năng, giao diện (interface) của 1 web service.

UDDI: là tổ chức quản lý các services, nơi có sự tham gia của các tổ chức như IBM, Microsoft, ... tham gia cung cấp các dịch vụ.

### Tổng quan SOA

Kiến trúc hướng dịch vụ (SOA) cho phép các đơn vị logic đơn lẻ tồn tại tự chủ nhưng không cô lập so với các đơn vị khác. Các đơn vị logic này vẫn phải phù hợp với một tập hợp các nguyên tắc cho phép chúng phát triển độc lập, trong khi vẫn duy trì đủ sự tương đồng và các tiêu chuẩn. Trong SOA, các đơn vị này của logic được gọi là dịch vụ. Có thể nghĩ SOA là 1 tập hợp các services.

## Các nguyên tắc thiết kế

Có 8 nguyên tắc thiết kế theo hướng kiến trúc và được chi thành 2 nhóm như sau:



### Nhóm nguyên lý thực thi

* Standardized Service Contract: dịch vụ đưa ra một thỏa thuận thông tin liên lạc, theo quy định chung bởi một hoặc nhiều mô tả dịch vụ và tài liệu liên quan.
* Service Reusability: các dịch vụ sẽ được chi thành các dịch vụ con với mục đích cho việc tái sử dụng các service.
* Service Autonomy: các dịch vụ sẽ có các quyền tự chủ.
* Service Statelessness: giảm việc lưu trữ các thông tin riêng.
* Service Discoverability: được thiết kế theo hướng mô tả bên ngoài cho phép người dùng tìm kiếm và sử dụng dựa trên các cơ chế khám phá có sẵng.



### Nhóm nguyên lý điều chỉnh

* Service Loose Coupling: giảm thiểu sự phụ thuộc giữa các service với nhau.
* Service Abstraction: che giấu thông tin logic của service khi hiển thị thông tin ra bên ngoài.
* Service Composability: tập hợp các service có thể phối hợp và ghép lại với nhau.

## Standardized Service contract

* Các Service contract thông thường bao gồm mô tả kỹ thuật và phi kỹ thuật. Các tài liệu thường được miêu tả như một dạng giao diện và được biết như API. Do đó các service trong kho dịch vụ phải tuân theo các chuẩn thiết kế contract giống nhau.
* Có 3 loại chuẩn hóa Service contract.
* Chuẩn hóa chức năng (Functional Expression Standardization): Các operation của service khi định nghĩa cần phải áp dụng các qui tắc đặt tên chuẩn, các qui tắc này được áp dụng cho các input message, output message và tên phải phù hợp với ngữ nghĩa của chúng. Điều này nhành giúp diễn giải service contract chính xác hơn, tăng khả năng liên kết nội tại và tính sử dụng lại của service, ngoài ra còn giảm khả năng trùng lắp dữ liệu.
* Chuẩn hóa mô hình dữ liệu (Data Model Standardization): yêu cầu phát triển các mô hình dữ liệu phải được chuẩn hóa để tạo kiến trúc dữ liệu được chuẩn hóa nhằm tăng khả năng sử dụng lại các service trong hệ thống, nhưng điều này không yêu cầu chuẩn hóa toàn bộ data model toàn hệ thống.
* Chuẩn hóa chính sách (Policy Standardization): các policy mô tả giới hạn sử dụng các service, để một service có thể sử dụng lại thì các behavioral requirement cần được trình bày theo phong cách phù hợp với việc sử dụng các policy expression được chuẩn hóa dựa trên các từ vựng chuẩn và phải phù hợp với những đặc điểm của ngôn ngữ.

## Service Reusability

* Là khả năng sử dụng lại của service.
* Mục tiêu: giúp tăng khả năng thu hồi vốn đầu tư ban đầu, tăng tính linh hoạt bằng cách thực hiện nhanh chóng các yêu cầu nghiệp vụ trong tương lai của doanh nghiệp.
* Đặc điểm thiết kế:
  + Service được xác định bởi các thuyết luôn luôn đúng của dịch vụ.
  + Service có tính dùng chung cao.
  + Service contract của service có tính dùng chung và có thể mở rộng.
  + Các service logic có thể sử dụng đồng thời.

## Service Autonomy

* Autonomy thể hiện khả năng tự điều chỉnh, được thể hiện qua sự tự do điều khiển đưa ra quyết định cho bản thân mà không cần sự chấp thuận từ phía bên ngoài dịch vụ.
* Để tăng tính tự trị đòi hỏi sự thực thi chương trình phải được cô lập hơn vì vậy sẽ làm tăng mức độ độc lập, kết quả là giúp chương trình tăng độ tin cậy và khả năng dự đoán.
* Mục tiêu:
  + Tăng quyền điều khiển của 1 service trên môi trường thực thi của nó.
  + Tăng độ tin cậy, khả năng dự đoán đặc biệt là được kết hợp và sử dụng lại.
* Có 2 loại service autonomy:
  + Runtime autonomy: mức độ điều khiển của một service trong quá trình xử lý trong thời điểm mà service được gọi và thực thi.
  + Design-time autonomy: mức độ tự do của chủ service để tạo sự thay đổi đối với 1 service trong thời gian tồn tại của nó.
* Đo lường khả năng tự trị của service:
  + Service contract autonomy: phạm vi khả năng của 1 contract không được trùng lắp với những contract khác. Điều này dễ dàng đạt được khi service contract được chuẩn hóa và tách rời với môi trường thực thi.
  + Shared autonomy.
  + Service logic autonomy: là mức độ phổ biết nhất có thể đạt được khi phát triển tùy chỉnh cho một service mới.
  + Pure autonomy.

## Service Statelessness

* Khi sự kết hợp các service ngày càng phức tạp thì số lượng dữ liệu hoạt động cần được quản lý trong suốt thời gian sống của composition. Đề tăng khả năng mở rộng và sự thực thi trong kho service thì các service và kiến trúc cần được thiết kế hỗ trợ sự ủy nhiệm và trì hoãn các việc quản lý trạng thái.
* Mục tiêu: tăng khả năng mở rộng service, hỗ trợ việc thiết kế agnostic service logic và cải tiến khả năng sử dụng lại service.
* Nguyên lý thiết kế: tùy thuộc vào service model và cách tiếp cận trì hoãn được dùng mà có các loại đặc điểm thiết kế khác nhau được thực hiện.
* Các loại state:
  + Active state: qui định của service sau khi nó được gọi và thực thi bởi một chương trình sử dụng.
  + Passive state: một service được load vào bộ nhớ nhưng nó không thực hiện bất kỳ chức năng nào của nó một cách tích cực.
  + Stateful: service được active và sử dụng thêm vùng nhớ tạm để lưu trữ tạm thời và xử lý state data. Stateful gồm 3 loại: session data, context data, business data.
  + Stateless: service được active và không xử dụng thêm vùng nhớ tạm để xử lý.

## Service Discoverability

* Khả năng khám phá, giải thích và trình bày service.
* Điều này sẽ giúp cho xác định service đã tồn tại hay cần phải xây dựng mới. Để làm được điều này đòi hỏi thông tin mô tả service phải chính xác thể hiện đầy đủ khả năng của service, nếu mô tả không đầy đủ thì người dùng sẽ mất cơ hội sử dụng lại service điều này dẫn đến sự trùng lắp với các service đã tồn tại.
* Mục tiêu: các service được đặt ở vị trí như các resource có khả năng khám phá cao, mục đích và khả năng của service được trình bày rõ ràng bởi con người cũng như chương trình phần mềm.
* Đặc điểm thiết kế:
  + Trang bị dữ liệu phù hợp.
  + Trang bị thêm những thông tin mô tả với mục đích sử dụng của chúng và những khả năng với con người.
  + Nếu một service đã được đăng ký thì các bản ghi đăng ký được phổ biến là những meta information.
  + Nếu một service không được đăng ký thì các tài liệu sơ lược về service được thiết kế bổ sung cho Service contract và đó là dạng cơ bản cho các bản ghi đăng ký trong tương lai.
  + Khả năng khám phá và trình bày dịch vụ.
  + Cải thiện sự truyền chất lượng thông tin của dữ liệu mô tả dịch vụ.
* Những loại khám phá:
  + Design-time discovery: tiến trình khám phá được thực hiện một cách thủ công bởi con người.
  + Runtime discovery: service có khả năng đưa ra những truy vấn discovery động và kết quả là một quá trình tự động hóa.

## Service Loose Coupling

* Service loose coupling nhằm giảm sự phụ thuộc vào các service khác mà bản thân service sử dụng, cũng như giảm sự phụ thuộc vào chương trình sử dụng bản thân service.
* Mục tiêu: giảm sự phụ thuộc giữa các service, tăng sự thực thi độc lập.
* Đặc điểm thiết kế: sự tồn tại của một service contract phải tách rời về công nghệ và sự thực thi. Không phụ thuộc vào xử lý bên ngoài, giảm tối thiểu các yêu cầu có sự gắn kết với consumer.
* Các loại service contract coupling:
  + Logic to contact coupling: phạm vi của 1 service được giới hạn bởi contract của nó, đây là dạng liên kết mang tính tích cự.
  + Contract to logic coupling: phạm vi của 1 service phụ thuộc vào logic bên dưới, đây là loại liên kết mang tính tích cực vì các chương trình service consumer tạo ra sụ ràng buộc với service contract.
  + Contract to technology coupling: là sự phụ thuộc vào công nghệ giao tiếp độc quyền. Loại liên kết này không mang tính tích cực vì khi service phụ thuộc vào công nghệ sẽ dẫn đến consumer cũng sẽ phụ thuộc vào công nghệ.
  + Contract to implement coupling: là sự phụ thuộc vào môi trường thực thi, đây cũng là loại liên kết không mang tính tích cực.
  + Contract to functional coupling: là sự phụ thuộc của service đối với external logic.

## Service Abstraction

* Các service contract chỉ chứa các thông tin cần thiết và các thông tin này sẽ giới hạn những thông tin gì sẽ được publish trong các service contract.
* Mục tiêu: giữ cho số lượng và nội dung thông tin súc tích, cân bằng và ngăn ngừa sự truy cập không cho phép.
* Đặc điểm thiết kế: thông tin sẽ được trừu tượng về công nghệ, logic vác các chức năng bên ngoài dịch vụ, thông tin về một service sẽ được điều khiển hoặc ẩn trong một môi trường cụ thể.
* Có 4 loại Meta abstraction:
  + Technology information: mô tả thực thi về kỹ thuật bên dưới. Thông tin chi tiết về kỹ thuật sẽ được ẩn đi để tạo sự tự do về thay đổi kỹ thuật mà không ảnh hưởng đến người sử dụng service.
  + Function information: metadata sẽ mô tả service có những chức năng nào.
  + Programatic logic information: mô tả cách thức dịch vụ thực hiện khả năng của nó. Được áp dụng ở những mức thiết kế chi tiết thấp như thuật toán, xử lý lỗi, ...
  + Quality of service information: metadata mô tả service behavior, các giới hạn và yêu cầu tương tác. Chất lượng của service data là một giới hạn của vùng thông tin liên quan đến độ tin cậy.

## Service Composability

* Khả năng kết hợp các service lại với nhau.
* Mục tiêu: tất cả các mục đích của service reusability có thể áp dụng, và là một dạng sử dụng lại dịch vụ.
* Các khái niệm và thuật ngữ tích hợp:
  + Composition.
  + Composition Instance.
  + Composition member.
  + Composition controller.
  + Composition sub-controller.
  + Composition member capability.
  + Composition controller capability.
  + Designated controller.
  + Collective composability.
  + Service activity.
  + Point-to-point.
  + Primitive composition.
  + Complex composition.
  + Composition initiator.

# Thiết kế chương trình

## Đối tượng phục vụ

* Tất cả mọi người muốn chia sẽ thông tin về địa điểm thú vị mà mình biết.
* Những người muốn tìm kiếm thông tin, hình ảnh về một địa điểm nào đó.
* Một đối tượng nào đó muốn quảng bá về địa điểm của mình.

## Các chức năng chính của chương trình

### ThemThongTin (Thêm thông tin)

Chức năng nhận thông tin về địa điểm từ người dùng. Khi người dùng gửi thông tin lên, web service sẽ nhận thông tin và lưu vào cơ sở dữ liệu.

**ThemThongTin(**int Maphanloai, string Ten, string Diachi, string Lng, string Lat, string Lienlac, string Avatar, string Hinhanh**)**

Tham số truyền vào: Maphanloai, Ten, Diachi, Quan, Thanhpho, Lng, Lat, Lienlac, Avatar, Hinhanh.

Kết quả trả về: chuỗi TranferRecord

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Maphanloai | int | Mã phân loại |
| Ten | Nvarchar(100) | Tên địa điểm  VD: Đại Học Hoa Sen |
| DiaChi | Nvarchar(50) | Địa chỉ của địa điểm  VD: Lô 2 CVPM Quang Trung |
| Lng | Nvarchar(50) | Kinh độ của địa điểm  VD: 10.09876 |
| Lat | Nvarchar(50) | Vĩ độ của địa điểm  VD: 10.3443002 |
| LienHe | Nvarchar(50) | Liên hệ của địa điểm  VD: Điện thoại và mail |
| Avatar | Nvarchar(50) | Avatar của địa điểm |
| HinhAnh | Nvarchar(50) | Hình ảnh của địa điểm |

Table 2.1: Thêm thông tin địa điểm

### Rating (Rating cho địa điểm)

Khi người sử dụng chấm điểm thì sẽ cộng điểm cũ với điểm người sử dụng mới chấm và chia cho tổng số lần đã chấm để ra điểm trung bình. Field Rating sẽ lưu 2 thông tin [điểm hiện tại, số lần chấm].

**Rating(string Ten, string Rating)**

Tham số truyền vào: Ten, Rating

Kết quả trả về: chuỗi tên + rating.

[OperationContract]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Ten | Nvarchar(50) | Tên địa điểm |
| Rating | Rating | Dùng chấm điểm  VD: 1-2-3-4-5 (Chấm điểm: 4) |

Table 2.2: Rating

### TimThongTin (tìm kiếm lấy thông tin)

Nếu người sử dụng chỉ tìm kiếm theo mã phân loại thì sẽ tìm kiếm theo phân loại tương ứng và trả về những kết quả được xếp hạng cao.

Nếu người sử dụng chỉ tìm kiếm theo tên thì sẽ tìm kiếm trong cơ sở dữ liệu các địa điểm có tên đó và trả về những địa điểm được xếp hạng cao.

Khi người dùng muốn tìm những địa điểm gần mình thì truyền vào tọa độ thì sẽ kết hợp với các tiêu chí tìm kiếm khác để trả về địa điểm gần tọa độ hiện tại.

**TimThongTin(int ID\_Phanloai, string TenDiaDiem, String Location)**

Tham số truyền vào: ID\_PhanLoai, TenDiaDiem, Location

Kết quả trả về: danh sách các đối tượng

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| ID\_Phanloai | Integer | Mã phân loại của địa điểm cần tìm. |
| TenDiaDiem | Nvarchar(50) | Tên địa điểm cần tìm. Vd: Đại học Hoa Sen. |
| Location | Nvarchar(50) | Thông tin về tọa độ cần tìm. Vd:  10.99876, 101.67542. |

Table 2.3: Tìm thông tin địa điểm

### ThemHinhAnh (Thêm hình ảnh)

Địa điểm đó sẽ được thêm vào các hình ảnh đã gửi lên. Người dùng chụp hình, sau đó mã hóa ra dạng stringbase64 rồi gửi lên web service, webservice sẽ nhận chuỗi mã hóa để giải mã thành hình.

**ThemHinhAnh (int ID\_DiaDiem, string HinhAnh)**

Tham số truyền vào: Ten, HinhAnh

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| ID\_DiaDiem | integer | Mã địa điểm |
| HinhAnh | nvarchar(50) | Tên file hình ảnh |
|  | String | Kết quả trả về là một chuỗi tên địa điểm, hình ảnh. |

Table 2.4: Thêm hình ảnh

### Báo cáo địa điểm sai

Cho phép người dùng thông tin đến hệ thống là địa điểm hiện tại đang cung cấp là sai thông tin. Khi người dùng phát hiện thông tin địa điểm sai thì sẽ chọn Báo cáo sai và hệ thống sẽ lưu vào cơ sở dữ liệu số lần địa điểm đó bị báo cáo sai và xóa khỏi danh sách địa điểm nếu cần thiết.

**Void BaoCaoSai(int id\_DiaDiem)**

Tham số truyền vào: id\_DiaDiem

|  |  |  |
| --- | --- | --- |
| Name | Description | Commend |
| Id\_DiaDiem | Integer | Id của thông tin địa điểm sai. |

Table 2.5: Báo cáo sai

### Thay đổi thông tin

Cho phép người quản lý thay đổi thông tin nếu thấy thông tin của một địa điểm bị đưa lên là sai. Khi người dùng đưa các thông số này vào thì sẽ update thông tin mới thay bằng thông tin cũ.

**Void ThayDoiThongTin**(int id\_DiaDiem, int id\_PhanLoai, String Ten, String Location, String DiaChi, String LienLac, String Avatar, String HinhAnh, float Rating)

Tham số truyền vào: id\_PhanLoai, Ten, Location, DiaChi, LienLac, Avatar, HinhAnh, Rating.

|  |  |  |
| --- | --- | --- |
| Name | Description | Commend |
| Id\_DiaDiem | Integer | ID của địa điểm sẽ bị thay đổi. |
| Id\_PhanLoai | Integer | Id của phân loại |
| Ten | Nvarchar(50) | Tên của địa điểm  Ví dụ: Trường Đại Học Hoa Sen CS1 |
| Lat | Nvarchar(50) | Kinh độ của địa điểm  Vd: 10.09876 |
| Lng | Nvarchar(50) | Vĩ độ của địa điểm  Vd: 106.23425 |
| DiaChi | Nvarchar(200) | Địa chỉ của địa điểm  Vd: 2 Tản Viên, phường Tân Sơn Nhất, quận Tân Bình, thành phố Hồ Chí Minh |
| LienLac | Nvarchar(100) | Thông tin liên lạc của địa chỉ.  Vd: email [abc@email.com](mailto:abc@email.com) |
| Avatar | Nvarchar(50) | Lưu tên của image sẽ làm avatar.  Vd : hinh001.jpg |
| HinhAnh | Nvarchar(200) | Lưu tên của các hình ảnh của địa điểm.  Vd: hinh002.jpg, hinh005.jpg |
| Rating | Float | Lưu thông tin điểm của địa điểm.  Vd: 4.5 |

Table 2.6: Thay đổi thông tin

### Xóa thông tin

Cho phép người quản lý xóa thông tin của địa điểm khi địa điểm đó bị thay đổi hoặc không còn tồn tại nữa. Khi người quản lý đưa thông tin của địa điểm thì địa điểm đó sẽ bị xóa khỏi cơ sở dữ liệu.

**Void XoaThongTin(int id\_DiaDiem)**

Tham số truyền vào: id\_DiaDiem

|  |  |  |
| --- | --- | --- |
| Name | Description | Commend |
| Id\_DiaDiem | Integer | ID của địa điểm muốn xóa thông tin. |

Table 2.7: Xóa thông tin

### Đăng nhập

Cho phép người quản lý đăng nhập vào hệ thống để thực hiện các chức năng như sửa thông tin, xóa thông tin. Khi người dùng đăng nhập thì hệ thống sẽ kiểm tra thông tin tài khoản, nếu đúng thì đăng nhập thành công, sai thì đăng nhập thất bại.

**int DangNhap(String Username, String Password)**

Tham số truyền vào: Username, Password

Kiểu dữ liệu trả về: true đăng nhập thành công, false là đăng nhập thất bại.

|  |  |  |
| --- | --- | --- |
| Name | Description | Commend |
| Username | Nvarchar(10) | Tên đăng nhập của tài khoản  Vd: admin |
| Password | Nvarchar(10) | Mật khẩu của tài khoản  Vd:123456 |
| Result | Int | Trả về 1 nếu đăng nhập thành công, trả về 0 nếu đăng nhập thất bại |

Table 2.8: Đăng nhập

Đối tượng chứa thông tin địa điểm

**[DataContract]**

**public class Thongtindiadiem**

**{**

**[DataMember]**

**public string Tenlinhvuc;**

**[DataMember]**

**public int Malinhvuc;**

**[DataMember]**

**public string Tenphanloai;**

**[DataMember]**

**public int Maphanloai;**

**[DataMember]**

**public string Ten;**

**[DataMember]**

**public string Lng;**

**[DataMember]**

**public string Lat;**

**[DataMember]**

**public string Diachi;**

**[DataMember]**

**public string Quan;**

**[DataMember]**

**public string Thanhpho;**

**[DataMember]**

**public string Lienlac;**

**[DataMember]**

**public string Avatar;**

**[DataMember]**

**public int Trangthai;**

**}**

**[DataContract]**

**public struct CustomFaultMsg**

**{**

**/// <summary>**

**/// Thuộc tính này được sử dụng để chứa thông tin chi tiết về lỗi xảy ra.**

**/// Thông tin về lỗi sẽ được gửi từ dịch vụ tới client**

**/// </summary>**

**[DataMember]**

**public string Message { get; set; }**

**}**

## Mô tả cơ sở dữ liệu



### Bảng Table\_DiaDiem

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| Id | Int | No | Khóa chính |
| Id\_Phanloai | Int | No | Khóa ngoại từ bảng PhanLoai |
| Ten | Nvarchar(100) | No | Tên địa điểm |
| Lng | Nvarchar(50) | No | Kinh độ |
| Lat | Nvarchar(50) | No | Vĩ độ |
| DiaChi | Nvarchar(50) | Yes | Địa chỉ |
| LienHe | Nvarchar(50) | Yes | Liên hệ |
| Avatar | Nvarchar(50) | Yes | Avatar |
| HinhAnh | Nvarchar(50) | Yes | Hình ảnh |
| SoNguoiRaTing | Int | Yes | Số người chấm điểm |
| Rating | Float | Yes | Chấm điểm |

### Bảng PhanLoai

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| ID | Int | No | Khóa chính |
| ID\_LinhVuc | Int | No | Khóa ngoại từ bảng LinhVuc |
| PhanLoai | Nvarchar(100) | Yes | Tên phân loại |

### Bảng LinhVuc

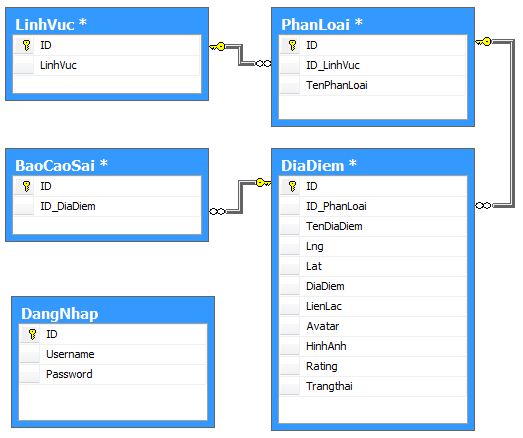
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| ID | Int | No | Khóa chính |
| LinhVuc | Nvarchar(50) | Yes | Tên lĩnh vực |

### Bảng BaoCaoSai

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| ID | Int | No | Khóa chính |
| ID\_Diadiem | Nvarchar(50) | No | Khóa ngoại từ bảng table\_DiaDiem |

### Bảng tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| ID | Int | No | Khóa chính |
| Username | Nvarchar(10) | No | Tên đăng nhập |
| Password | Nvarchar(10) | No | Mật khẩu |



Hình 3 : Bảng Cơ sở dữ liệu

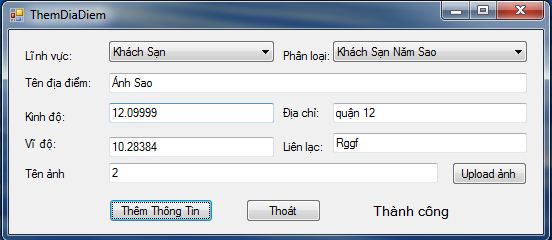
# Hiện thực Chương Trình

**Màn hình chính** : 

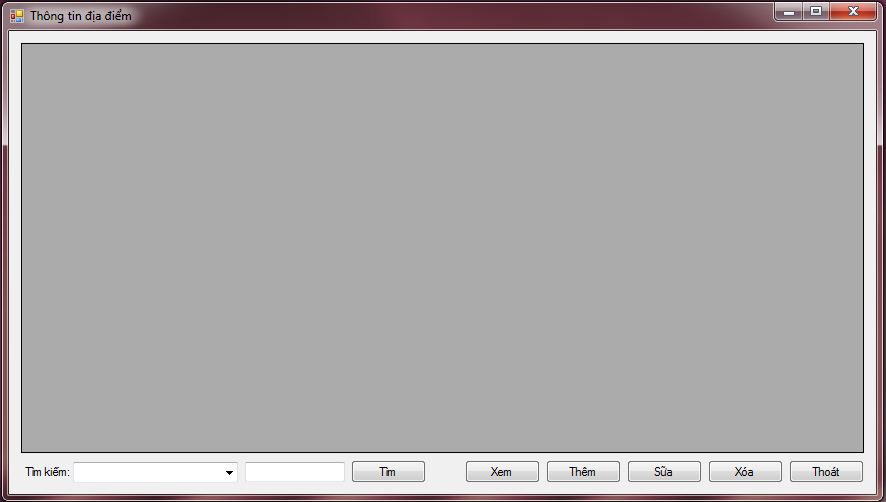
**Màn hình đăng nhập :**



**Màn hình thêm thông tin :**



**Màn hình tìm kiếm, xem, xáa, sữa :**



Kết luận

Sau khi kết thúc môn học “Công nghệ và dịch vụ web”, thì nhóm đã hiện thực được 1 phần so với kế hoạch ban đầu. Đó là đã đưa thông tin từ người dùng đăng tải từ điện thoại lên trang web cho người sử dụng xem. Với những gì nhóm đã hiện thực được, đó cũng là một ưu điểm đối với nhóm trong quãng thời gian ngắn ngũi này.

Bên cạnh đó, nhóm vẫn còn những khuyết điểm: phần mềm chỉ có những chức năng chính yếu(cho người dùng xem địa điểm cần tìm), và chưa có những chức năng hỗ trợ người dùng để tiện lợi hơn như tìm kiếm theo khu vực, tìm kiếm theo loại(ví dụ: công ty, nhà trường ….), chưa quản lý được thông tin của người dùng đưa lên.

Nếu phần mềm này tiếp tục được phát triển, thì đầu tiên nhóm sẽ khắc phục những khuyết điểm trên, và ngoài ra nhóm tiếp tục phát triển tìm đường đi ngắn nhất, đo khoảng cách đường chân chim, mở rộng dịch vụ cho các công ty. Công ty nào muốn có hình ảnh logo riêng trên bản đồ thì đăng kí với nhóm. Nhóm phát triển tìm kiếm địa điểm theo loại thức ăn (ví dụ: tìm địa điểm của tất cả các quán bán cháo lòng,....) và kết hợp với tìm kiếm theo địa điểm (ví dụ: các quán cơm ở quận 1), phân loại dữ liệu cũ và mới, đánh giá (vote rate) cho thông tin tốt nhất và chính xác nhất về địa điểm đó. Nhóm cố gắng phát triển được về thời tiết hiện tại theo từng khu vực như khu vực nào đang có mưa thì sẽ hiển thị mưa lớn hay nhỏ, có gió, hay có nắng, có mây không. Đây cũng là điều mới lạ mà nhóm muốn phát triển và mở rộng.

Tài liệu tham khảo

Các trang web tham khảo :

[www.diadiem.com](http://www.diadiem.com)

<http://maps.google.com/>