|  |
| --- |
| **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](#_Toc312269998)

[1.1. Giới thiệu Web service & SOA 2](#_Toc312269999)

[1.1.1. Web service 2](#_Toc312270000)

[1.1.2. Tổng quan về SOA 7](#_Toc312270001)

[1.1.2.1. Các actor trong SOA 8](#_Toc312270002)

[1.1.2.2. Ích lợi khi sử dụng SOA 9](#_Toc312270003)

[1.1.2.3. Thông điệp (message) trong SOA 10](#_Toc312270004)

[1.1.3. Sự khác biệt giữa SOA và WebSerive 11](#_Toc312270005)

[2. Thiết kế chương trình 12](#_Toc312270006)

[2.1. Đối tượng phục vụ 12](#_Toc312270007)

[2.2. Các chức năng chính của chương trình 12](#_Toc312270008)

[2.2.1. ThemThongTin (Thêm thông tin) 12](#_Toc312270009)

[2.2.2. Rating (Rating cho địa điểm) 13](#_Toc312270010)

[2.2.3. TimThongTin (tìm kiếm lấy thông tin) 13](#_Toc312270011)

[2.2.4. ThemHinhAnh (Thêm hình ảnh) 14](#_Toc312270012)

[2.2.5. Báo cáo địa điểm sai 14](#_Toc312270013)

[2.2.6. Thay đổi thông tin 15](#_Toc312270014)

[2.2.7. Xóa thông tin 16](#_Toc312270015)

[2.2.8. Đăng nhập 16](#_Toc312270016)

[2.3. Mô tả cơ sở dữ liệu 18](#_Toc312270017)

[2.3.1. Bảng Table\_DiaDiem 18](#_Toc312270021)

[2.3.2. Bảng PhanLoai 19](#_Toc312270022)

[2.3.3. Bảng LinhVuc 19](#_Toc312270023)

[2.3.4. Bảng BaoCaoSai 19](#_Toc312270024)

[2.3.5. Bảng tài khoản 19](#_Toc312270025)

[3. Hiện thực Chương Trình 21](#_Toc312270026)

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

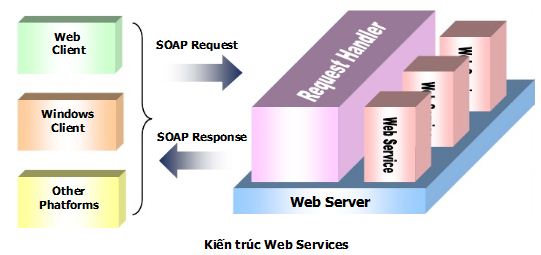
## Tìm hiểu lý thuyết

## Giới thiệu Web service & SOA

### Web service

#### Web Service là gì

WebService là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML. WebService là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu. Một WebService được tạo nên bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng nhìn thấy và có thể truy cập đến những dịch vụ mà nó thực hiện, đồng thời có thể yêu cầu thông tin từ WebService khác. Nó bao gồm các module độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.Web Service có thể được coi là thế hệ kế tiếp của các dịch vụ phân tán trên mạng như DCOM, CORBA, RMI,...nhưng không giống như các dịch vụ phân tán trước đó Web Service có thể được gọi bất kỳ ở đâu và trên bất kỳ nền tảng nào (Theo Microsoft). Vì cách thức giao tiếp với Web Service thông qua chuẩn XML được hầu hết các ngôn ngữ lập trình và các nền tảng hỗ trợ.  
  
 Giá trị cơ bản của WebService dựa trên việc cung cấp các phương thức theo chuẩn trong việc truy nhập đối với hệ thống đóng gói và hệ thống kế thừa. Các phần mềm được viết bởi những ngôn ngữ lập trình khác nhau và chạy trên những nền tảng khác nhau có thể sử dụng WebService để chuyển đổi dữ liệu thông qua mạng Internet theo cách giao tiếp tương tự bên trong một máy tính. Tuy nhiên, công nghệ xây dựng WebService không nhất thiết phải là các công nghệ mới, nó có thể kết hợp với các công nghệ đã có như XML, SOAP, WSDL, UDDI



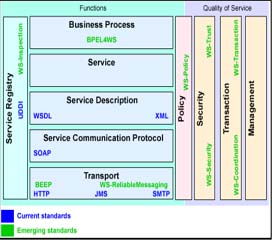
#### **Kiến trúc của **WebService****

WebService gồm có 3 chuẩn chính: SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) và UDDI (Universal Description, Discovery, and Integration). Hình 1 mô tả chồng giao thức của WebService, trong đó UDDI được sử dụng để đăng ký và khám phá WebService đã được miêu tả cụ thể trong WSDL. Giao tác UDDI sử dụng SOAP để nói chuyện với UDDI server, sau đó các ứng dụng SOAP yêu cầu một WebService. Các thông điệp SOAP được gửi đi chính xác bởi HTTP và TCP/IP.

  
Hình 1. Chồng giao thức của WebService

Chồng giao thức WebService là tập hợp các giao thức mạng máy tính được sử dụng để định nghĩa, xác định vị trí, thi hành và tạo nên WebService tương tác với những ứng dụng hay dịch vụ khác. Chồng giao thức này có 4 thành phần chính:

- Dịch vụ vận chuyển (Service Transport): có nhiệm vụ truyền thông điệp giữa các ứng dụng mạng, bao gồm những giao thức như HTTP, SMTP, FTP, JSM và gần đây nhất là giao thức thay đổi khổi mở rộng (Blocks Extensible Exchange Protocol- BEEP).  
- Thông điệp XML: có nhiệm vụ giải mã các thông điệp theo định dạng XML để có thể hiểu được ở mức ứng dụng tương tác với người dùng. Hiện tại, những giao thức thực hiện nhiệm vụ này là XML-RPC, SOAP và REST.  
- Mô tả dịch vụ: được sử dụng để miêu tả các giao diện chung cho một WebService cụ thể. WSDL thường được sử dụng cho mục đích này, nó là một ngôn ngữ mô tả giao tiếp và thực thi dựa trên XML. WebService sẽ sử dụng ngôn ngữ này để truyền tham số và các loại dữ liệu cho các thao tác và chức năng mà WebService cung cấp.  
- Khám phá dịch vụ: tập trung dịch vụ vào trong một nơi được đăng ký, từ đó giúp một WebService có thể dễ dàng khám phá ra những dịch vụ nào đã có trên mạng, tốt hơn trong việc tìm kiếm những dịch vụ khác để tương tác. Một WebService cũng phải tiến hành đăng ký để các dịch vụ khác có thể truy cập và giao tiếp. Hiện tại, UDDI API thường được sử dụng để thực hiện công việc này.

Kiến trúc sâu hơn được mô tả trong Hình 2:  


Trong đó, tầng giao thức tương tác dịch vụ (Service Communication Protocol) với công nghệ chuẩn là SOAP. SOAP là giao thức nằm giữa tầng vận chuyển và tầng mô tả thông tin về dịch vụ, cho phép người dùng triệu gọi một dịch vụ từ xa thông qua một thông điệp XML. Ngoài ra, để các dịch vụ có tính an toàn, toàn vẹn và bảo mật thông tin, trong kiến trúc WebService, chúng ta có thêm các tầng Policy, Security, Transaction, Management.

#### Các thành phần của WebService

##### XML – eXtensible Markup Language

Là một chuẩn mở do W3C đưa ra cho cách thức mô tả dữ liệu, nó được sử dụng để định nghĩa các thành phần dữ liệu trên trang web và cho những tài liệu B2B. Về hình thức, XML hoàn toàn có cấu trúc thẻ giống như ngôn ngữ HTML nhưng HTML định nghĩa thành phần được hiển thị như thế nào thì XML lại định nghĩa những thành phần đó chứa cái gì. Với XML, các thẻ có thể được lập trình viên tự tạo ra trên mỗi trang web và được chọn là định dạng thông điệp chuẩn bởi tính phổ biến và hiệu quả mã nguồn mở.

Do WebService là sự kết hợp của nhiều thành phần khác nhau nên nó sử dụng các tính năng và đặc trưng của các thành phần đó để giao tiếp. XML là công cụ chính để giải quyết vấn đề này và là kiến trúc nền tảng cho việc xây dựng một WebService, tất cả dữ liệu sẽ được chuyển sang định dạng thẻ XML. Khi đó, các thông tin mã hóa sẽ hoàn toàn phù hợp với các thông tin theo chuẩn của SOAP hoặc XML-RPC và có thể tương tác với nhau trong một thể thống nhất.

##### WSDL – Web Service Description Language

WSDL định nghĩa cách mô tả WebService theo cú pháp tổng quát của XML, bao gồm các thông tin:

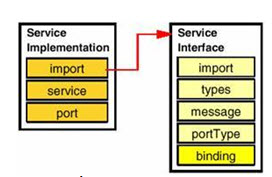
- Tên dịch vụ  
- Giao thức và kiểu mã hóa sẽ được sử dụng khi gọi các hàm của WebService  
- Loại thông tin: thao tác, tham số, những kiểu dữ liệu (có thể là giao diện của WebService cộng với tên cho giao diện này).

Một WSDL hợp lệ gồm hai phần: phần giao diện (mô tả giao diện và phương thức kết nối) và phần thi hành mô tả thông tin truy xuất CSDL. Cả hai phần này sẽ được lưu trong 2 tập tin XML tương ứng là tập tin giao diện dịch vụ và tập tin thi hành dịch vụ. Giao diện của một WebService được miêu tả trong phần này đưa ra cách thức làm thế nào để giao tiếp qua WebService. Tên, giao thức liên kết và định dạng thông điệp yêu cầu để tương tác với WebService được đưa vào thư mục của WSDL.

WSDL thường được sử dụng kết hợp với XML schema và SOAP để cung cấp WebService qua Internet. Một client khi kết nối tới WebService có thể đọc WSDL để xác định những chức năng sẵn có trên server. Sau đó, client có thể sử dụng SOAP để lấy ra chức năng chính xác có trong WSDL.

##### Universal Description, Discovery, and Integration (UDDI)

Để có thể sử dụng các dịch vụ, trước tiên client phải tìm dịch vụ, ghi nhận thông tin về cách sử dụng và biết được đối tượng nào cung cấp dịch vụ. UDDI định nghĩa một số thành phần cho biết các thông tin này, cho phép các client truy tìm và nhận những thông tin được yêu cầu khi sử dụng WebService.

- Cấu trúc UDDI :  
+ Trang trắng – White pages: chứa thông tin liên hệ và các định dạng chính yếu của WebService, chẳng hạn tên giao dịch, địa chỉ, thông tin nhận dạng… Những thông tin này cho phép các đối tượng khác xác định được dịch vụ.  
+ Trang vàng – Yellow pages: chứa thông tin mô tả WebService theo những loại khác nhau. Những thông tin này cho phép các đối tượng thấy được WebService theo từng loại với nó.  
+ Trang xanh – Green pages: chứa thông tin kỹ thuật mô tả các hành vi và các chức năng của WebService.  
+ Loại dịch vụ – tModel:  chứa các thông tin về loại dịch vụ được sử dụng. 

Những thông tin về WebService được sử dụng và công bố lên mạng sử dụng giao thức này. Nó sẽ kích hoạt các ứng dụng để tìm kiếm thông tin của WebService khác nhằm xác định xem dịch vụ nào sẽ cần đến nó

##### SOAP – Simple Object Access Protocol

SOAP là một giao thức giao tiếp có cấu trúc như XML. Nó được xem là cấu trúc xương sống của các ứng dụng phân tán được xây dựng từ nhiều ngôn ngữ và các hệ điều hành khác nhau. SOAP là giao thức thay đổi các thông điệp dựa trên XML qua mạng máy tính, thông thường sử dụng giao thức HTTP.

Một client sẽ gửi thông điệp yêu cầu tới server và ngay lập tức server sẽ gửi những thông điệp trả lời tới client. Cả SMTP và HTTP đều là những giao thức ở lớp ứng dụng của SOAP nhưng HTTP được sử dụng và chấp nhận rộng rãi hơn bởi ngày nay nó có thể làm việc rất tốt với cơ sở hạ tầng Internet.

Cấu trúc một thông điệp theo dạng SOAP bao gồm các phần tử sau:

- Phần tử gốc – envelop: phần tử bao trùm nội dung thông điệp, khai báo văn bản XML như là một thông điệp SOAP.  
- Phần tử đầu trang – header: chứa các thông tin tiêu đề cho trang, phần tử này không bắt buộc khai báo trong văn bản. Header còn có thể mang những dữ liệu chứng thực, những chứ ký số, thông tin mã hóa hay cài đặt cho các giao dịch khác.  
- Phần tử khai báo nội dung chính trong thông điệp – body, chứa các thông tin yêu cầu và thông tin được phản hồi.  
- Phần tử đưa ra các thông tin về lỗi -fault, cung cấp thông tin lỗi xảy ra trong qúa trình xử lý thông điệp.

Một SOAP đơn giản trong body sẽ lưu các thông tin về tên thông điệp, tham chiếu tới một thể hiện của dịch vụ, một hoặc nhiều tham số. Có 3 kiểu thông báo sẽ được đưa ra khi truyền thông tin: request message(tham số gọi thực thi một thông điệp), respond message (các tham số trả về, được sử dụng khi yêu cầu được đáp ứng) và cuối cùng là fault message (thông báo tình trạng lỗi).

Kiểu truyền thông: Có 2 kiểu truyền thông

- Remote procedure call (RPC): cho phép gọi hàm hoặc thủ tục qua mạng. Kiểu này được khai thác bởi nhiều WebService.  
- Document: được biết đến như kiểu hướng thông điệp, nó cung cấp giao tiếp ở mức trừu tượng thấp, khó hiểu và yêu cầu lập trình viên mất công sức hơn.

Hai kiểu truyền thông này cung cấp các định dạng thông điệp, tham số, lời gọi đến các API khác nhau nên việc sử dụng chúng tùy thuộc vào thời gian và sự phù hợp với WebService cần xây dựng.

Cấu trúc dữ liệu: Cung cấp những định dạng và khái niệm cơ bản giống như trong các ngôn ngữ lập trình khác như kiểu dữ liệu (int, string, date…) hay những kiều phức tạp hơn như struct, array, vector… Định nghĩa cấu trúc dữ liệu SOAP được đặt trong namespace SOAP-ENC.

Mã hóa: Giả sử service rquester và service provider được phát triển trong Java, khi đó mã hóa SOAP là làm thế nào chuyển đổi từ cấu trúc dữ liệu Java sang SOAP XML và ngược lại, bởi vì định dạng cho Web Service chính là XML. Bất kỳ một môi trường thực thi SOAP nào cũng phải có một bảng chứa thông tin ánh xạ nhằm chuyển đổi từ ngôn ngữ Java sang XML và từ XML sang Java – bảng đó được gọi là SOAPMappingRegistry. Nếu một kiểu dữ liệu được sử dụng dưới một dạng mã hóa thì sẽ có một ánh xạ tồn tại trong bộ đăng ký của môi trường thực thi SOAP đó.

### Tổng quan về SOA

SOA - Service Oriented Architecture (Kiến trúc Định hướng Dịch vụ), theo định nghĩa của DotNetGuru, là 'Khái niệm về hệ thống trong đó mỗi ứng dụng được xem như một nguồn cung cấp dịch vụ'.  
  
Dịch vụ là yếu tố then chốt trong SOA. Có thể hiểu dịch vụ như là hàm chức năng (module phần mềm) thực hiện qui trình nghiệp vụ nào đó. Một cách cơ bản, SOA là tập hợp các dịch vụ kết nối 'mềm dẻo' với nhau (nghĩa là một ứng dụng có thể 'nói chuyện' với một ứng dụng khác mà không cần biết các chi tiết kỹ thuật bên trong), có giao tiếp (dùng để gọi hàm dịch vụ) được định nghĩa rõ ràng và độc lập với nền tảng hệ thống, và có thể tái sử dụng. SOA là cấp độ cao hơn của phát triển ứng dụng, chú trọng đến qui trình nghiệp vụ và dùng giao tiếp chuẩn để giúp che đi sự phức tạp kỹ thuật bên dưới.  
  
Thiết kế SOA tách riêng phần thực hiện dịch vụ (phần mềm) với giao tiếp gọi dịch vụ. Điều này tạo nên một giao tiếp nhất quán cho ứng dụng khách (client) sử dụng dịch vụ bất chấp công nghệ thực hiện dịch vụ. Thay vì xây dựng các ứng dụng đơn lẻ và đồ sộ, nhà phát triển sẽ xây dựng các dịch vụ tinh gọn có thể triển khai và tái sử dụng trong toàn bộ quy trình nghiệp vụ. Điều này cho phép tái sử dụng phần mềm tốt hơn, cũng như tăng sự linh hoạt vì nhà phát triển có thể cải tiến dịch vụ mà không làm ảnh hưởng đến ứng dụng client sử dụng dịch vụ.  
  
Thật ra triết lý SOA không hoàn toàn mới, DCOM và CORBA cũng có kiến trúc tương tự. Tuy nhiên, các kiến trúc cũ ràng buộc các thành phần với nhau quá chặt, ví dụ như các ứng dụng phân tán muốn làm việc với nhau phải đạt được 'thỏa thuận' về chi tiết tập hàm API, một thay đổi mã lệnh trong thành phần COM sẽ yêu cầu những thay đổi tương ứng đối với mã lệnh truy cập thành phần COM này.  
  
Ưu điểm quan trọng nhất của SOA là khả năng kết nối 'mềm dẻo' (nhờ sự chuẩn hóa giao tiếp) và tái sử dụng. Các dịch vụ có thể được sử dụng với trình client chạy trên nền tảng bất kỳ và được viết với ngôn ngữ bất kỳ. (Ví dụ, ứng dụng Java có thể liên kết với WebService .NET và ngược lại).  
  
SOA dựa trên 2 nguyên tắc thiết kế quan trọng:  
  
\* Module: Tách vấn đề lớn thành nhiều vấn đề nhỏ.  
\* Đóng gói: Che đi dữ liệu và lô-gic trong từng module (hay 'hộp đen') đối với truy cập từ ngoài.

## Các actor trong SOA

.3.1.1 mô tả các actor tham gia trong một hệ thống xây dựng theo SOA.

* Service Provider: Cung cấp stateless service phục vụ cho một nhu cầu nào đó. User (service consumer) không cần quan tâm đến vị trí thực sự mà service họ cần sử dụng đang hoạt động.
* Serive Consumer: User sử dụng service được cung cấp bởi Service Provider
* Service Registry: Nơi lưu trữ thông tin về các service của các Service Provider khác nhau, Service Consumer dựa trên những thông tin này để tìm kiếm và lựa chọn Service Provider phù hợp.

Hình 1.3.1.1: SOA actors

Service Provider sẽ đăng kí thông tin về service mà mình có thể cung cấp (các chức năng có thể cung cấp, khả năng của hệ thống (resource, performance), giá cả dịch vụ, ...) vào Service Registry. Service Consumer khi có nhu cầu về một service nào đó sẽ tìm kiếm thông tin trên Service Registry. Ngoài chức năng hỗ trợ tìm kiếm, Service Registry còn có thể xếp hạng các Service Provider dựa trên các tiêu chí về chất lượng dịch vụ, bầu chọn từ các khách hàng đã sử dụng service, ... Những thông tin này sẽ hỗ trợ thêm cho quá trình tìm kiếm của Service Consumer. Khi đã xác định được Service Provider mong muốn, Service Consumer thiết lập kênh giao tiếp trực tiếp với Service Provider nhằm sử dụng service hoặc tiến hành thương lượng thêm (về mặt giá cả, resource sử dụng, ...)

## Ích lợi khi sử dụng SOA

Sử dụng mô hình SOA trong việc thiết kế hệ thống mang lại lợi ích về mặt kinhtế cũng như kỹ thuật.

* Lợi ích kinh tế
  + Doanh nghiệp có điều kiện tập trung thời gian để tìm kiếm các giải pháp cho các bài toán liên quan đến kinh tế.
  + Thúc đẩy sự phát triển của hệ thống hiện có cũng như cung cấp khả năng mở rộng hệ thống trong tương lai.
* Lợi ích kỹ thuật
  + Hệ thống xây dựng theo mô hình SOA đảm bảo các service trong hệ thống có tính độc lập cao (độ kết dính thấp) (autonomous và loose coupling).
  + Ở góc nhìn người sử dụng, vị trí các service có tính trong suốt (transparency), việc di dời các service đến một máy tính khác không ảnh hưởng khả năng phục vụ yêu cầu khách hàng.
  + Hoạt động của các service có tính động, hành vi của các service tùy thời đểm, tùy yêu cầu cần xử lý mà có sự khác nhau (late binding).

## Thông điệp (message) trong SOA

So với kiểu thiết kế Component-Based, điểm khác biệt chính của SOA là cung cấp khả năng giao tiếp giữa các thành phần trong hệ thống (service) sử dụng thông điệp (message) dựa trên các chuẩn giao tiếp đã được chuẩn hóa (HTTP, FTP, SMTP, ...). Chính nhờ đặc điểm này, hệ thống SOA trở nên độc lập với platform (platform independent). Các service hoạt động trên nền các platform khác nhau vẫn có thể giao tiếp với nhau nhờ vào các interface giao tiếp đã được chuẩn hóa để cộng tác xử lý một tác vụ nào đó.

Sử dụng thông điệp (message) để giao tiếp có các lợi thế sau:

* Cross-platform: thông điệp (message) trở thành ngôn ngữ chung của các platform và các ngôn ngữ lập trình khác nhau. Điều này đảm bảo các service trên các platform khác nhau hoạt động với cấu trúc dữ liệu đặc thù của platform đó.
* Asynchronous communications: hoạt động gởi nhận thông điệp được thực hiện theo cơ chế Fire-and-Forget. Sender và Receiver không cần phải chờ thông điệp trả lời sau khi đã gởi đi một thông điệp. Điều này giúp cho Sender và Receiver tiếp tục xử lý công việc sau khi gởi thông điệp mà không cần dừng thực thi để chờ thông điệp trả lời.
* Reliable communication: các thông điệp từ Sender có thể được gởi đến một service trung gian có nhiệm vụ lưu trữ (store) các thông điệp. Service trung gian sẽ gởi (forward) thông điệp cho Receiver khi Receiver có thể xử lý yêu cầu tiếp theo. Cơ chế Store-and-Forward này đảm bảo các thông điệp sẽ không bị thất lạc trong trường hợp Receiver bị quá tải và không thể nhận thêm yêu cầu mới.
* Thread management: Việc trao đổi thông điệp theo cơ chế bất đồng bộ giúp ứng dụng không cần ngừng thực thi để chờ một tác vụ kết thúc mà có thể tạo ra các thread xử lý các công việc khác nhau.
* Remote communication: Các thông điệp lưu trữ thông tin về các đối tượng dữ liệu dưới dạng đặc tả hình thức thay thế việc phải serialization and deserialization các đối tượng dữ liệu truyền qua mạng khi ứng dụng thực hiện remote call một ứng dụng khác.
* End-to-end security: Thông điệp có thể lưu trữ thông tin về security context của kênh giao tiếp. Điều này cung cấp khả năng điều khiển liên quan đến security như authentication and authorization.

### Sự khác biệt giữa SOA và WebSerive

Đặc điểm chính của SOA là tách rời phần giao tiếp với phần thực hiện dịch vụ. Điều này có thể làm bạn liên tưởng đến một công nghệ được đề cập nhiều gần đây: Dịch vụ web. Dịch vụ web cho phép truy cập thông qua định nghĩa giao thức-và-giao tiếp. SOA và dịch vụ web thoạt trông có vẻ giống nhau nhưng chúng không phải là một.  
  
Về cơ bản, SOA là kiến trúc phần mềm phát xuất từ định nghĩa giao tiếp và xây dựng toàn bộ mô hình ứng dụng như là mô hình các giao tiếp, hiện thực giao tiếp và phương thức gọi giao tiếp. Giao tiếp là trung tâm của toàn bộ triết lý kiến trúc này; thực ra, tên gọi 'kiến trúc định hướng giao tiếp' thích hợp hơn cho SOA. Dịch vụ và module phần mềm nghiệp vụ được truy cập thông qua giao tiếp, thường theo cách thức yêu cầu - đáp trả. Ngay cả với yêu cầu dịch vụ 1 chiều thì nó vẫn là yêu cầu trực tiếp có chủ đích từ một phần mềm này đến một phần mềm khác. Một tương tác định hướng dịch vụ luôn bao hàm một cặp đối tác: nguồn cung cấp dịch vụ và khách hàng sử dụng dịch vụ.  
  
Định nghĩa cơ bản của dịch vụ web dựa trên một nền tảng khác: Tập hợp các công nghệ WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol) và UDDI (Universal Description, Discovery ang Integration), cho phép xây dựng các giải pháp lập trình cho vấn đề tích hợp ứng dụng và truyền thông điệp. Theo thời gian, các công nghệ này có thể hoàn thiện hay có thể được thay bằng công nghệ khác tốt hơn, hiệu quả hơn hay ổn định hơn. (Hiện tại thì các công nghệ này làm việc tốt và hiệu quả).  
  
Rõ ràng, theo định nghĩa thì dịch vụ web là đặc tả công nghệ còn SOA là triết lý thiết kế phần mềm. Dịch vụ web đưa ra giải pháp kỹ thuật để thực hiện SOA, nhưng SOA cũng có thể thực hiện với các giải pháp kỹ thuật khác không phải dịch vụ web (và không phải tất cả dịch vụ web đều có kiến trúc SOA). Tuy vậy, SOA và dịch vụ web có mối quan hệ tương hỗ: sự phổ biến của dịch vụ web giúp thúc đẩy sự phát triển của SOA, và kiến trúc tốt của SOA sẽ giúp dịch vụ web thành công.

## 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

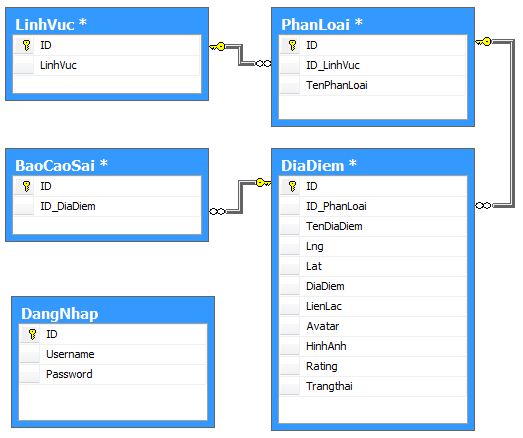
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| ID | Int | No | Khóa chính |
| LinhVuc | Nvarchar(50) | Yes | Tên lĩnh vực |

## 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

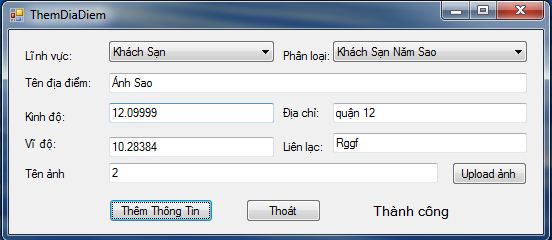
## Hiện thực Chương Trình

**Màn hình chính** : 

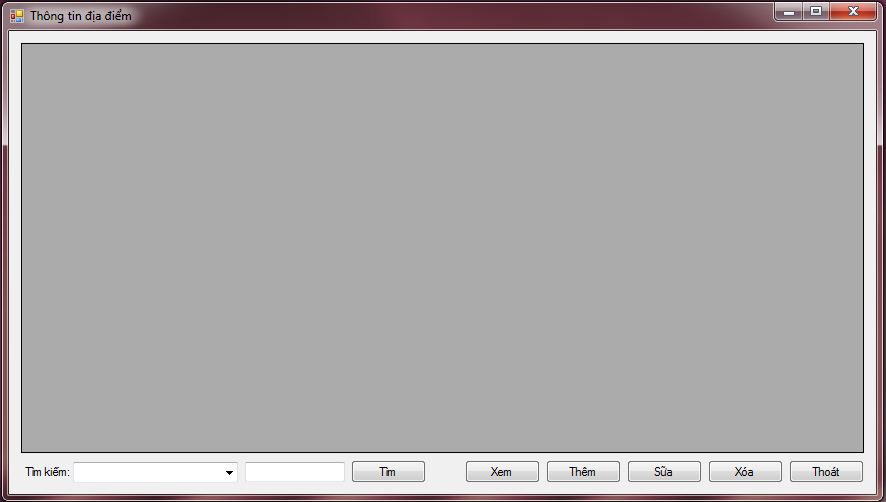
**Màn hình đăng nhập :**



**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/>