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

**Đề án Công Nghệ và Dịch Vụ Web**

**TÊN ĐỀ TÀI:Web Service Thời Khóa Biểu Giáo Viên**

**Giảng viên hướng dẫn** : Thầy Võ Hoàng Hải

**Lớp** : QL081  
 **Thời gian thực hiện** :   
 **Nhóm sinh viên thực hiện** : Nguyễn Ngọc Tuấn 080918  
 Bùi Hoàng Thiện 080897

Văn Công Thắng 081671

**Thành phố Hồ Chí Minh**

**THÁNG 10/2011**

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc312229449)

[Nội Dung 5](#_Toc312229450)

[**I.Lý Thuyết** 5](#_Toc312229451)

[**1. Giới thiệu công nghệ** 5](#_Toc312229452)

[**2. Đặc điểm của Dịch vụ Web** 6](#_Toc312229453)

[**3. Kiến trúc của Dịch vụ Web** 8](#_Toc312229454)

[**4. Các thành phần của Dịch vụ Web** 10](#_Toc312229455)

[**5. An toàn cho dịch vụ Web** 14](#_Toc312229456)

[**6. Xây dựng một dịch vụ Web** 16](#_Toc312229457)

[**7. Tích hợp dịch vụ Web theo chuẩn** 17](#_Toc312229458)

[**II.PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH** 19](#_Toc312229459)

[**1.Kiến trúc hệ thống** 19](#_Toc312229460)

[**1.1.** **Nhóm chức năng hỗ trợ người dùng** 20](#_Toc312229461)

[1.1.1 Get lecturer 20](#_Toc312229462)

[1.1.2 Get List Lecturer 20](#_Toc312229463)

[1.1.3 Insert Lecturer 21](#_Toc312229464)

[1.1.4 Get Course 21](#_Toc312229465)

[1.1.5 Get List Course 22](#_Toc312229466)

[1.1.6 Insert Course 22](#_Toc312229467)

[1.1.7 Get Class 23](#_Toc312229468)

[1.1.8 Get List Class 23](#_Toc312229469)

[1.1.9 Insert Class 24](#_Toc312229470)

[1.1.10 Get Schedule 24](#_Toc312229471)

[1.1.11 Get List Schedule 25](#_Toc312229472)

[1.1.12 Insert Schedule 25](#_Toc312229473)

[1.1.13 Login 26](#_Toc312229474)

[1.1.14 Register 27](#_Toc312229475)

[**1.2.Hiện thực chức năng:** 28](#_Toc312229476)

[**III. KẾT LUẬN** 31](#_Toc312229477)

[**1. Ưu điểm** 31](#_Toc312229478)

[**2.Khuyết điểm** 31](#_Toc312229479)

[**3.Hướng phát triển** 31](#_Toc312229480)

[Kết Luận 33](#_Toc312229481)

# LỜI CẢM ƠN

Thông qua đồ án lần này, nhóm chúng tôi xin chân thành cảm ơn thầy Võ Hoàng Hải đã cho chúng tôi tiếp xúc và thử nghiệm với một đề tài hữu ích và sát với thực tế, hiểu rõ hơn về cách Xây dựng mọi ứng dụng Web Service. Nhờ sự giúp đỡ nhiệt tình của thầy chúng tôi đã hoàn thành đồ án đúng thòi hạn, tuy nhiên do về mặc kiến thức cũng như kinh nghiệm thực tế của chúng tôi còn hạn chế nên trong quá thực hiện đề tài vẫn không thế thiếu phần sai sót, chúng tôi rất mong nhận được sự nhận xét từ thầy để chúng tôi có thể nhận ra những sai sót và rút kinh nghiệm.

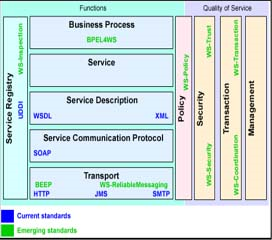
Chúng tôi xin chân thành cảm ơn đến thầy Võ Hoàng Hải – là người trực tiếp hướng dẫn, hỗ trợ chúng tôi trong suốt thời gian thực hiện đề tài.

**Trân trọng.**

# Nội Dung

## **I.Lý Thuyết**

Dịch vụ Web (Web Service) được coi là một công nghệ mang đến cuộc cách mạng trong cách thức hoạt động của các dịch vụ B2B (Business to Business) và B2C (Business to Customer). Giá trị cơ bản của dịch vụ Web 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 dịch vụ Web 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 dịch vụ Web 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… Với sự phát triển và lớn mạnh của Internet, dịch vụ Web thật sự là một công nghệ đáng được quan tâm để giảm chi phí và độ phức tạp trong tích hợp và phát triển hệ thống. Chúng ta sẽ xem xét các dịch vụ Web từ mức khái niệm đến cách thức xây dựng.



### **1. Giới thiệu công nghệ**

Theo định nghĩa của W3C (World Wide Web Consortium), dịch vụ Web 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. Dịch vụ Web 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 dịch vụ Web 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 dịch vụ Web khác. Nó bao gồm các mô đun độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.

Trước hết, có thể nói rằng ứng dụng cơ bản của Dịch vụ Web là tích hợp các hệ thống và là một trong những hoạt động chính khi phát triển hệ thống. Trong hệ thống này, các ứng dụng cần được tích hợp với cơ sở dữ liệu (CSDL) và các ứng dụng khác, người sử dụng sẽ giao tiếp với CSDL để tiến hành phân tích và lấy dữ liệu. Trong thời gian gần đây, việc phát triển mạnh mẽ của thương mại điện tử và B2B cũng đòi hỏi các hệ thống phải có khả năng tích hợp với CSDL của các đối tác kinh doanh (nghĩa là tương tác với hệ thống bên ngoài – bên cạnh tương tác với các thành phần bên trong của hệ thống trong doanh nghiệp).

Dưới đây, chúng ta sẽ xem qua những khái niệm và cách thức cơ bản nhất để xây dựng một dịch vụ Web trong tích hợp và phát triển hệ thống.

### **2. Đặc điểm của Dịch vụ Web**

a) Đặc điểm

- Dịch vụ Web cho phép client và server tương tác được với nhau ngay cả trong những môi trường khác nhau. Ví dụ, đặt Web server cho ứng dụng trên một máy chủ chạy hệ điều hành Linux trong khi người dùng sử dụng máy tính chạy hệ điều hành Windows, ứng dụng vẫn có thể chạy và xử lý bình thường mà không cần thêm yêu cầu đặc biệt để tương thích giữa hai hệ điều hành này.  
- Phần lớn kĩ thuật của Dịch vụ Web được xây dựng dựa trên mã nguồn mở và được phát triển từ các chuẩn đã được công nhận, ví dụ như XML.  
- Một Dịch vụ Web bao gồm có nhiều mô-đun và có thể công bố lên mạng Internet.  
- Là sự kết hợp của việc phát triển theo hướng từng thành phần với những lĩnh vực cụ thể và cơ sở hạ tầng Web, đưa ra những lợi ích cho cả doanh nghiệp, khách hàng, những nhà cung cấp khác và cả những cá nhân thông qua mạng Internet.  
- Một ứng dụng khi được triển khai sẽ hoạt động theo mô hình client-server. Nó có thể được triển khai bởi một phần mềm ứng dụng phía server ví dụ như PHP, Oracle Application server hay Microsoft.Net…  
- Ngày nay dịch vụ Web đang rất phát triển, những lĩnh vực trong cuộc sống có thể áp dụng và tích hợp dịch vụ Web là khá rộng lớn như dịch vụ chọn lọc và phân loại tin tức (hệ thống thư viện có kết nối đến web portal để tìm kiếm các thông tin cần thiết); ứng dụng cho các dịch vụ du lịch (cung cấp giá vé, thông tin về địa điểm…), các đại lý bán hàng qua mạng, thông tin thương mại như giá cả, tỷ giá hối đoái, đấu giá qua mạng…hay dịch vụ giao dịch trực tuyến (cho cả B2B và B2C) như đặt vé máy bay, thông tin thuê xe…  
- Các ứng dụng có tích hợp dịch vụ Web đã không còn là xa lạ, đặc biệt trong điều kiện thương mại điện tử đang bùng nổ và phát triển không ngừng cùng với sự lớn mạnh của Internet. Bất kì một lĩnh vực nào trong cuộc sống cũng có thể tích hợp với dịch vụ Web, đây là cách thức kinh doanh và làm việc có hiệu quả bởi thời đại ngày nay là thời đại của truyền thông và trao đổi thông tin qua mạng. Do vậy, việc phát triển và tích hợp các ứng dụng với dịch vụ Web đang được quan tâm phát triển là điều hoàn toàn dễ hiểu.

b) Ưu và nhược điểm

Ưu điểm:

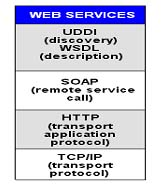
+ Dịch vụ Web cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.  
+ Sử dụng các giao thức và chuẩn mở. Giao thức và định dạng dữ liệu dựa trên văn bản (text), giúp các lập trình viên dễ dàng hiểu được.  
+ Nâng cao khả năng tái sử dụng.  
+ Thúc đẩy đầu tư các hệ thống phần mềm đã tồn tại bằng cách cho phép các tiến trình/chức năng nghiệp vụ đóng gói trong giao diện dịch vụ Web.  
+ Tạo mối quan hệ tương tác lẫn nhau và mềm dẻo giữa các thành phần trong hệ thống, dễ dàng cho việc phát triển các ứng dụng phân tán.  
+ Thúc đẩy hệ thống tích hợp, giảm sự phức tạp của hệ thống, hạ giá thành hoạt động, phát triển hệ thống nhanh và tương tác hiệu quả với hệ thống của các doanh nghiệp khác.

Nhược điểm:

+ Những thiệt hại lớn sẽ xảy ra vào khoảng thời gian chết của Dịch vụ Web, giao diện không thay đổi, có thể lỗi nếu một máy khách không được nâng cấp, thiếu các giao thức cho việc vận hành.  
+ Có quá nhiều chuẩn cho dịch vụ Web khiến người dùng khó nắm bắt.  
+ Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

### 3. Kiến trúc của Dịch vụ Web

Dịch vụ Web 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 dịch vụ Web, trong đó UDDI được sử dụng để đăng ký và khám phá dịch vụ Web 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 dịch vụ Web. Các thông điệp SOAP được gửi đi chính xác bởi HTTP và TCP/IP.

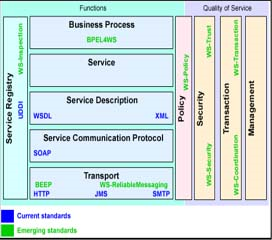


Hình 1. Chồng giao thức của dịch vụ Web

Chồng giao thức dịch vụ Web là tập hợp các giao thức mạng máy tính được sử dụng để định nghĩa, xác định vị trí, thi hành và tạo nên dịch vụ Web 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 dịch vụ Web 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. Dịch vụ Web sẽ sử dụng ngôn ngữ này để truyền tham số và các loại dữ liệu cho các thao tác và chức năng mà dịch vụ Web cung cấp.  
- Khám phá dịch vụ: tập trung dịch vụ vào trong một nơi được đăng ký, từ đó giúp một dịch vụ Web 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 dịch vụ Web 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 dịch vụ Web, chúng ta có thêm các tầng Policy, Security, Transaction, Management.

### 4. Các thành phần của Dịch vụ Web

a) 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 dịch vụ Web 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 dịch vụ Web, 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.

b)WSDL – Web Service Description Language

WSDL định nghĩa cách mô tả dịch vụ Web 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 dịch vụ Web  
- Loại thông tin: thao tác, tham số, những kiểu dữ liệu (có thể là giao diện của dịch vụ Web 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 dịch vụ Web được miêu tả trong phần này đưa ra cách thức làm thế nào để giao tiếp qua dịch vụ Web. Tên, giao thức liên kết và định dạng thông điệp yêu cầu để tương tác với dịch vụ Web được đưa vào thư mục của WSDL.

WSDL thường được sử dụng kết hợp với XML schema và SOAP để cung cấp dịch vụ Web qua Internet. Một client khi kết nối tới dịch vụ Web 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.

c) 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 dịch vụ Web.

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

Những thông tin về dịch vụ Web 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 dịch vụ Web khác nhằm xác định xem dịch vụ nào sẽ cần đến nó.

d) SOAP – Simple Object Access Protocol

Chúng ta đã hiểu cơ bản dịch vụ Web như thế nào nhưng vẫn còn một vấn đề khá quan trọng. Đó là làm thế nào để truy xuất dịch vụ khi đã tìm thấy? Câu trả lời là các dịch vụ Web có thể truy xuất bằng một giao thức là Simple Object Access Protocol – SOAP. Nói cách khác chúng ta có thể truy xuất đến UDDI registry bằng các lệnh gọi hoàn toàn theo định dạng của SOAP.

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

Thông điệp theo định dạng SOAP là một văn bản XML bình thường 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 dịch vụ Web.  
- 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 dịch vụ Web 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.

### **5. An toàn cho dịch vụ Web**

Dịch vụ Web liên kết và tương tác với các ứng dụng qua Internet, chính vì vậy bảo mật là một vấn đề được quan tâm khi các công ty tiến tới kết hợp ứng dụng với một dịch vụ Web. Việc đảm bảo an toàn cho dịch vụ Web là một vấn đề quan trọng, đặc biệt đối với những dịch vụ liên quan đến trao đổi tiền tệ, thông tin từ thị trường chứng khoán hay dịch vụ bán hàng qua mạng (liên quan đến trả tiền bằng tài khoản và có yêu cầu thông tin cá nhân của người dùng).

Trước khi có WS-Security (bảo mật cho dịch vụ Web) thì ý nghĩa thông thường của an toàn dịch vụ Web là bảo mật kênh truyền dữ liệu. Hiện nay, nó được thực hiện cho những SOAP/HTTP dựa trên cơ chế truyền thông điệp bằng cách sử dụng giao thức HTTPS. Không chỉ là an toàn ở mức truyền thông điệp, HTTPS còn cung cấp sự an toàn tới toàn bộ gói dữ liệu HTTP.

Mặc dù HTTPS không bao gồm tất cả các khía cạnh trong chuẩn an toàn chung cho dịch vụ Web nhưng nó đã cung cấp một lớp bảo mật khá đầy đủ với định danh, chứng thực, tính toàn vẹn thông điệp hay độ tin cậy.

- Đảm bảo an toàn cho dịch vụ Web:

Khái niệm về WS-Security: đây là một chuẩn an toàn bao trùm cho SOAP, nó được dùng khi muốn xây dựng những dịch vụ Web toàn vẹn và tin cậy. Toàn vẹn có nghĩa là khi có một giao dịch hay khi truyền thông tin, hệ thống và thông tin sẽ không bị chặn, giao dịch sẽ không bị mất cũng như không thể có người lấy cắp được dữ liệu trên đường truyền. WS-security được thiết kế mang tính mở nhằm hướng tới những mô hình an toàn khác bao gồm PKI, Kerberos và SSL. Nó cũng đưa ra nhiều hỗ trợ cho các cơ chế an toàn khác, nhiều khuôn dạng chữ ký và công nghệ mã hóa, đảm bảo sự an toàn, toàn vẹn thông điệp và tính tin cậy của thông điệp. Tuy nhiên, WS-security cũng chưa thể đảm bảo được tất cả yêu cầu về bảo mật và an toàn thông tin, nó chỉ là một trong những lớp của giải pháp an toàn cho dịch vụ Web.

Tính toàn vẹn tạo ra một chữ ký số hóa XML dựa trên nội dung của thông điệp. Nếu dữ liệu bị thay đổi bất hợp pháp, nó sẽ không còn thích hợp với chữ ký số hóa XML đó. Chữ ký này được tạo ra dựa trên khóa mà người gửi thông điệp tạo ra, do đó người nhận chỉ nhận thông điệp khi có chữ ký sử dụng và nội dung phù hợp. Ngược lại sẽ có một thông báo lỗi. Việc chứng thực được thực hiện giữa client và server là cách chứng thực rất cơ bản (sử dụng định danh người dùng và mật khẩu).

WS-security chỉ là một trong những lớp an toàn và bảo mật cho dịch vụ Web, vì vậy cần một mô hình an toàn chung lớn hơn để có thể bao quát được các khía cạnh khác. Các thành phần được thêm có thể là WS-Secure Conversation Describes,WS-Authentication Describes,WS-Policy Describes hay WS-Trust Describes. Chúng sẽ thực hiện việc đảm bảo an toàn hơn cho hệ thống khi trao đổi dữ liệu, mở và đóng các phiên làm việc cũng như quản lý dữ liệu cần chứng thực và chính sách chứng thực.

### **6. Xây dựng một dịch vụ Web**

Có 4 giai đoạn chính để xây dựng một dịch vụ Web là xây dựng, triển khai, tiến hành và quản lý, trong đó:

- Giai đoạn xây dựng bao gồm phát triển và chạy thử ứng dụng dịch vụ Web, xây dựng các chức năng và định nghĩa dịch vụ. Có hai cách khác nhau để tiến hành trong giai đoạn này, đó là Red-path- solod và Blue-path-dashed. Với Red- path-solod, chúng ta sẽ xây dựng một dịch vụ Web mới từ trạng thái ban đầu hoặc với một dịch vụ đã có sẵn. Từ đó, xây dựng định nghĩa service (WSDL) với các đối tượng, hàm chức năng mà chúng ta mong muốn. Nếu theo cách Blue-path-dashed, dịch vụ Web sẽ được xây dựng từ đầu hoặc từ một định nghĩa dịch vụ WSDL. Sử dụng WSDL này, xây dựng hoặc sửa đổi lại mã để thực hiện các yêu cầu mong muốn trong dịch vụ Web.

- Giai đoạn triển khai: công bố định nghĩa dịch vụ, xây dựng WSDL và triển khai mã thực thi của dịch vụ Web. Triển khai dịch vụ Web tới một ứng dụng phía server, sau đó sẽ công bố dịch vụ Web trên mạng Internet để các client có thể nhìn thấy. Sử dụng UDDI registry để công bố lên mạng.

- Giai đoạn tiến hành: tìm kiếm và gọi thực thi dịch vụ Web bởi những người dùng muốn sử dụng dịch vụ.

- Quản lý: Quản lý và quản trị dịch vụ, duy trì sự ổn định của dịch vụ, cập nhật thông tin mới, sửa lỗi khi nó xảy ra…

Để xây dựng một dịch vụ Web, chúng ta cần hiểu được những việc phải làm và nên bắt đầu từ đâu. Có 3 cách tiếp cận chủ yếu để xây dựng nên một dịch vụ Web, có thể từ một ứng dụng đã có (bottom-up); từ một định nghĩa dịch vụ, WSDL để phát sinh một ứng dụng mới (top-down) hoặc có thể từ một nhóm các dịch vụ Web hiện có, kết hợp lại với nhau để tạo nên các chức năng mới hoặc mở rộng thêm chức năng. Những hướng tiếp cận này dựa trên những gì mà chúng ta đã có, tùy thuộc vào yêu cầu của hệ thống, trong đó tối đa việc sử dụng lại các chức năng, các thành phần, môđun đã được xây dựng.

Qui trình xây dựng một dịch vụ Web bao gồm các bước sau:

1. Định nghĩa và xây dựng các chức năng, các dịch vụ mà dịch vụ sẽ cung cấp (sử dụng ngôn ngữ Java chẳng hạn).  
2. Tạo WSDL cho dịch vụ  
3. Xây dựng SOAP server  
4. Đăng ký WSDL với UDDI registry để cho phép các client có thể tìm thấy và truy xuất.  
5. Client nhận file WSDL và từ đó xây dựng SOAP client để có thể kết nối với SOAP server  
6. Xây dựng ứng dụng phía client (chẳng hạn sử dụng Java) và sau đó gọi thực hiện dịch vụ thông qua việc kết nối tới SOAP server.

Lựa chọn một ngôn ngữ, xây dựng các tiến trình nghiệp vụ và chúng ta bắt đầu tạo nên một dịch vụ Web như ý muốn. Sau đó là cung cấp dịch vụ Web này trên Internet.

### **7. Tích hợp dịch vụ Web theo chuẩn**

Để có thể thành công với dịch vụ Web chúng ta phải quan tâm đến khá nhiều vấn đề, bao gồm việc triển khai, giám sát và tích hợp hệ thống. Doanh nghiệp không những phải phát triển một ứng dụng dịch vụ Web mới mà còn phải tích hợp các ứng dụng nghiệp vụ phụ trợ của họ trong kiến trúc Dịch vụ Web. Cùng với việc triển khai và tích hợp, những nhà kinh doanh và những người sử dụng kỹ thuật cũng cần có khả năng giám sát, triển khai toàn diện để đảm bảo hoạt động kinh doanh hiệu quả và tin cậy.

- Giám sát (monitoring): Cần hỗ trợ ở cả mức công cụ và cơ sở hạ tầng để giám sát các dịch vụ Web chạy như thế nào qua toàn bộ mạng, từ một chi nhánh con của một công ty trên mạng tới các chi nhánh khác trong công ty hay giao tiếp với doanh nghiệp khác. Kết hợp thông báo theo sự kiện với các lỗi trong luồng nghiệp vụ cho những người dùng không có kinh nghiệm giám sát dịch vụ Web và các dịch vụ kế thừa khác.

- Xác định đường đi dữ liệu (Data routing): Việc thiết lập đường đi của dữ liệu giữa những thành phần của dịch vụ Web hướng tới tối đa hóa khả năng sử dụng lại. Nếu coi một thành phần (component) là một đối tượng thì mỗi thể hiện (instance) của nó sẽ không quan tâm đến các thể hiện khác của cùng thành phần đó. Những thể hiện của cùng một thành phần có thể dễ dàng được sử dụng lại trong các ứng dụng phân tán khác bởi vì chúng hoàn toàn độc lập và không phụ thuộc lẫn nhau.

- Triển khai (Deployment): Triển khai các dịch vụ Web có khả năng nâng cấp, điều khiển và cấu hình các thành phần từ xa thông qua mạng phân tán.

- Quản lý (Management): Có thể xây dựng theo kiến trúc P2P (Peer-to-Peer). Các hoạt động chính như thực thi các thành phần, định tuyến dữ liệu, xử lý luồng công việc và chuyển đổi dữ liệu được thực hiện tại các điểm cuối của mạng. Server sẽ tập trung giải quyết các hoạt động khác như quản lý, điều khiển sự kiện, chứng thực bảo mật và quản trị.

- Cấu hình và quản lý phiên bản (Configuration and version management): Sử dụng các công cụ linh hoạt để quản lý các phiên bản khác nhau của dịch vụ Web, cho phép các phiên bản được nâng cấp và điều khiển từ một công cụ quản lý tập trung. Kết hợp giữa ứng dụng và mạng giúp các kỹ sư triển khai có thể điều khiển các thành phần chạy trên nền tảng hệ thống phần cứng cụ thể bên trong mạng.

- Bảo mật (Security): các chuẩn mở như HTTP, XML, SOAP, WSDL và chuẩn bảo mật JSM được sử dụng rộng rãi khiến chúng trở thành lý tưởng để xây dựng các ứng dụng web. Đầu tiên, dịch vụ Web sử dụng những công nghệ này giống như firewall, SSL và các chứng nhận số. Dịch vụ Web thế hệ sau này sẽ kết hợp với những công nghệ có khả năng bảo mật cao hơn, giống như mã hóa XML và chứng nhận số XML.

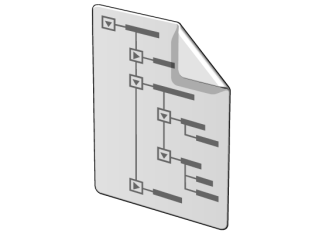
Như vậy, với một dịch vụ Web, việc giao tiếp và truyền nhận dữ liệu trở nên dễ dàng và hiệu quả hơn, đồng thời đem lại chi phí thấp hơn và tăng cường những khả năng giao tiếp thời gian thực, kết nối với mọi người trên khắp thế giới. Bản chất của nền tảng công nghệ này là kiến trúc hướng dịch vụ và sự phát triển của dịch vụ Web có tương lai rất khả quan.

## **II.PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH**

### **1.Kiến trúc hệ thống**

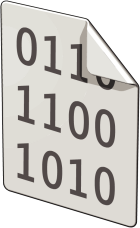


Client

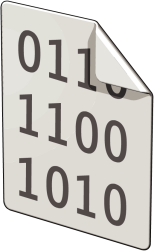
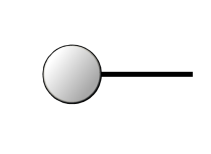
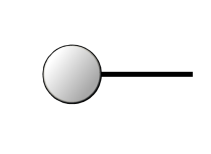
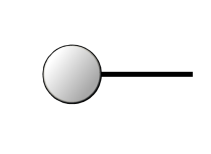


Service Host

Configuration



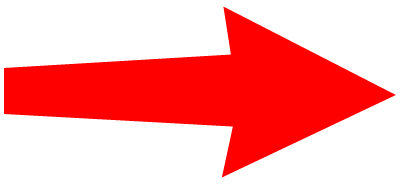
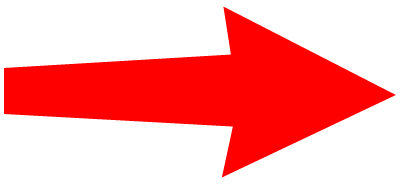
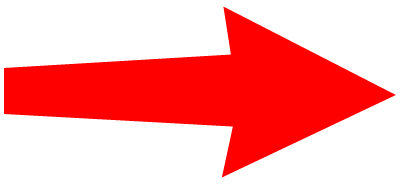
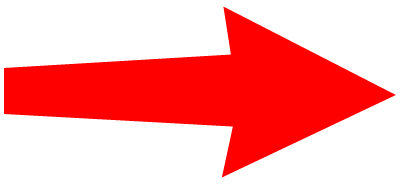
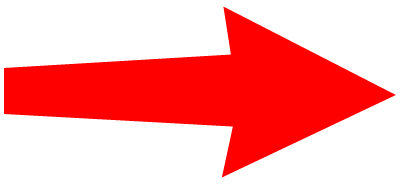
Host code



WCF Service Library

Contract

Implementation



*Hình: Hệ thống gồm 3 phần chính: Client, Service Host và WCF Service Library.*

* Client: là ứng dụng phía người sử dụng, có thể là windows application hoặc web applicaion.
* Service Host: quản lý cấu hình host.
* WCF Service Library: là thư viện cung cấp các chức năng và xử lý dữ liệu

#### **Nhóm chức năng hỗ trợ người dùng**

**1.1.1 Get lecturer**

1. **Mô tả chức năng**

Cho phép người dùng lấy thông tin giảng viên thông qua mã giảng viên.

[OperationContract]

Lecturer GetLecturer(int ID);

1. **Đối tượng sử dụng**

Sinh viên, giảng viên, admin.

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| ID\_Lecturer | Int | ID\_Lecturer là mã số của giảng viên |

**1.1.2 Get List Lecturer**

1. **Mô tả chức năng**

Cho phép người dùng xem danh sách giảng viên.

[OperationContract]

DataSet GetListLecturer();

1. **Đối tượng sử dụng**

Sinh viên, giảng viên, admin.

1. **Input**

Không có.

**1.1.3 Insert Lecturer**

1. **Mô tả chức năng**

Cho phép người sử dụng thêm giảng viên.

1. **Đối tượng sử dụng**

Admin.

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| Name\_Lecturer | nvarchar(50) | Nam\_Lecturer là họ tên của giảng viên |

**1.1.4 Get Course**

1. **Mô tả chức năng**

Cho phép người dùng xem môn học theo mã môn học.

[OperationContract]

Course GetCourse(int ID);

1. **Đối tượng sử dụng**

Giảng viên, sinh viên , admin.

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| ID\_Course | Int | ID\_Course là mã môn học |

**1.1.5 Get List Course**

1. **Mô tả chức năng**

Cho phép người dùng xem danh sách môn học.

[OperationContract]

DataSet GetListCourse();

1. **Đối tượng sử dụng**

Giảng viên, sinh viên, admin.

1. **Input**

Không có.

**1.1.6 Insert Course**

1. **Mô tả chức năng**

Cho phép người dùng thêm môn học.

[OperationContract(IsOneWay = true)]

void InsertCourse(Course cou);

1. **Đối tượng sử dụng**

Admin.

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| Name\_Course | Nvarchar(50) | Name\_Course là tên của môn học. |

**1.1.7 Get Class**

1. **Mô tả chức năng**

Cho phép người dùng xem thông tin lớp học dựa vào mã lớp.

[OperationContract]

Class GetClass(int ID);

1. **Đối tượng sử dụng**

Giảng viên, sinh viên, admin.

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| ID\_Class | Int | ID\_Class là mã lớp học |

**1.1.8 Get List Class**

1. **Mô tả chức năng**

Cho phép người dùng xem danh sách lớp học.

[OperationContract]

DataSet GetListClass();

1. **Đối tượng sử dụng**

Giảng viên ,sinh viên, admin.

1. **Input**

Không có.

**1.1.9 Insert Class**

1. **Mô tả chức năng**

Cho phép người dùng thêm lớp học.

void InsertClass(Class cla);

[OperationContract]

1. **Đối tượng sử dụng**

Admin

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| Name\_Class | Nvarchar(50) | Name\_Class là tên của môn học. |

**1.1.10 Get Schedule**

1. **Mô tả chức năng**

Cho phép người dùng xem lịch trình giảng dạy của giảng viên dựa vào mã lịch trình.

[OperationContract]

Schedule GetSchedule(int ID);

1. **Đối tượng sử dụng**

Giảng viên, sinh viên, admin.

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| ID\_Schedule | Int | ID\_Schedule là mã lịch trình. |

**1.1.11 Get List Schedule**

1. **Mô tả chức năng**

Cho phép người dùng xem danh sách lịch trình giảng dạy của giảng viên.

[OperationContract]

DataSet GetListSchedule();

1. **Đối tượng sử dụng**

Giảng viên, sinh viên, admin.

1. **Input**

Không có.

**1.1.12 Insert Schedule**

1. **Mô tả chức năng**

Cho phép người dùng thêm lịch trình giảng dạy của giảng viên.

[OperationContract(IsOneWay = true)]

void InsertSchedule(Schedule sch);

1. **Đối tượng sử dụng**

Admin.

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| ID\_Lecturer | Int | ID\_Lecturer là mã giảng viên |
| ID\_Course | Int | ID\_Course là mã môn học. |
| ID\_Class | Int | ID\_Class là mã lớp học. |
| Day | Nvarchar(50) | Phân biệt người dùng đang đăng nhập hệ thống. |
| First\_period | Int |  |
| NoPeriod | Int |  |
| Name\_Room | Nvarchar(50) | Name\_Room là tên phòng học. |
| Size | Int |  |
| Week\_Begin | Int | Tuần bắt đầu. |
| Weed\_End | Int | Tuần kết thúc. |

**1.1.13 Login**

1. **Mô tả chức năng**

Cho phép người dùng đăng nhập vào hệ thống, yêu cầu bắt buộc để thực hiện các tác vụ kế tiếp.

[OperationContract]

bool Login(string user, string pass);

1. **Đối tượng sử dụng**

Admin.

1. **Input**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| username | char(6) | Username là mã số sinh viên, mã giảng viên, bắt buộc đủ 6 ký tự |
| password | varchar(50) | Mật khẩu để đăng nhập, tối thiểu 6 ký tự, tối đa 50 ký tự, không chấp nhận chuỗi unicode |

**1.1.14 Register**

1. **Mô tả chức năng**

Cho phép người dùng đăng ký tài khoản.

[OperationContract]

bool Register(string user , string pass);

1. **Đối tượng sử dụng**

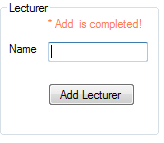
Admin.

1. **Input**

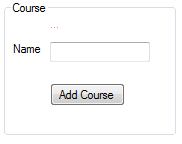
|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| username | char(6) | Username là mã số sinh viên, mã giảng viên, bắt buộc đủ 6 ký tự |
| password | varchar(50) | Mật khẩu để đăng nhập, tối thiểu 6 ký tự, tối đa 50 ký tự, không chấp nhận chuỗi unicode |

#### **1.2.Hiện thực chức năng:**

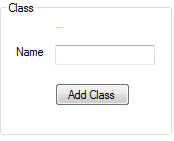
Sử dụng Get Class truyền Id\_Class để lấy tên lớp



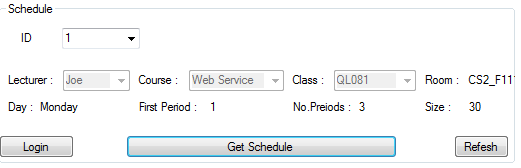
Sử dụng hàm Add Lecturer để thêm tên Giảng Viên



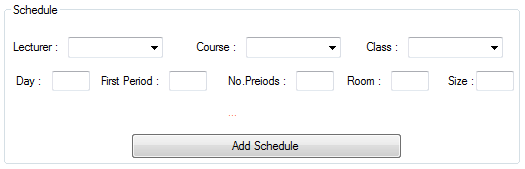
Sử dụng hàm Add Course để them tên môn học



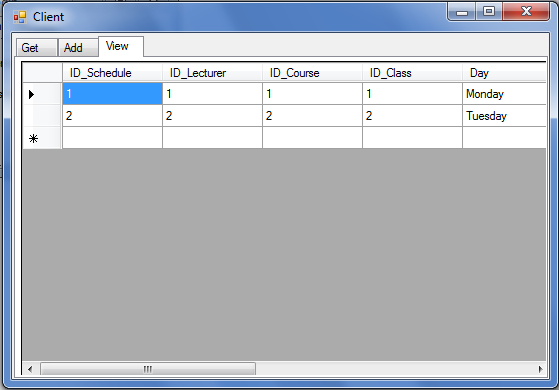
Sử dụng hàm Add Class để them tên Lớp



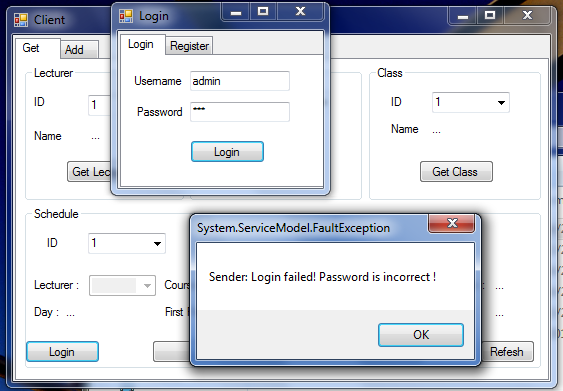
Sử dụng hàm Get Schedule để lấy thông tin Lịch trình



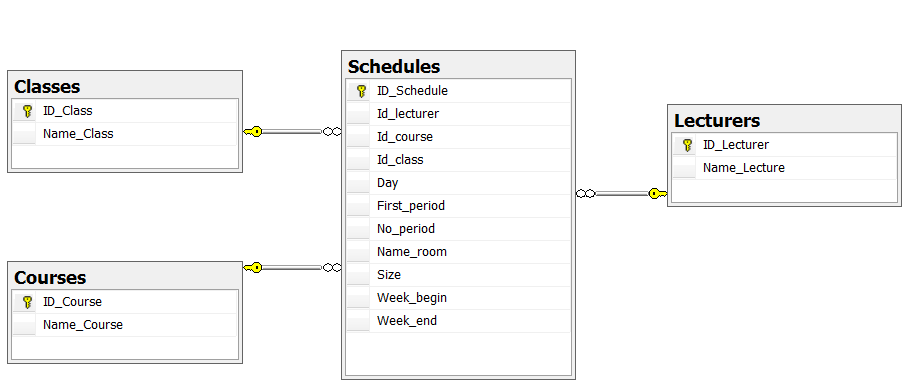
Sử dụng hàm Add Schedule để them thong tin lịch trình



Chức năng Login



Cơ Sở Dữ Liệu:



## **III. KẾT LUẬN**

### **1.** **Ưu điểm**

* Hỗ trợ đa dạng các binding: basicHttpBinding, wsHttpBinding, wsDualHttpBinding, netTcpBinding, netNamedPipeBinding,
* Hoàn thành đề tài đúng thời gian đưa ra.
* Giao diện thân thiện đối với người sử dụng.
* Đáp ứng đầy đủ các chức năng cơ bản của việc cung cấp thời khóa biểu cho giảng viên

### **2.Khuyết điểm**

* Xử lý các ngoại lệ chưa chặt chẽ.
* Chức năng client chưa phong phú
* Các quy tắc nghiệp vụ chưa hoàn chỉnh

### **3.Hướng phát triển**

* Nâng cao tốc độ xử lý.
* Kiểm tra chặc chẽ các quy tắc nghiệp vụ, đảm bảo tính đúng đắn của dữ liệu.
* Quản trị phân quyền sử dụng một cách chi tiết hơn như: phân quyền phân hệ, chức năng, hành động.
* Kiểm tra thêm các ràng buộc dữ liệu.

# Kết Luận

Trong quá trình thực hiện đề tài chúng tôi đã được tiếp xúc quá trình xây dựng một Web Service trên bộ công cụ lập trình Visual Studio. Qua đó chúng tôi đã tiếp thu được những kinh nghiệm quí báu về kinh nghiệm lập và cũng như hiểu rõ hơn về cơ chế hoạt động cũng như các ứng dụng ủa Web Service đồng thời tiếp thu được kiến thức hữu ích để hoàn thành đề tài này.