TRƯỜNG ĐẠI HỌC HOA SEN  
KHOA KHOA HỌC VÀ CÔNG NGHỆ



**CÔNG NGHỆ VÀ DỊCH VỤ WEB**

**WCF**

**TÊN ĐỀ TÀI** : QUẢN LÝ HỒ SƠ BỆNH ÁN

**GIẢNG VIÊN HƯỚNG DẪN** :

**NGƯỜI THỰC HIỆN**:

* Trần Thị Kim Hiền - 091544
* Nguyễn Hải Triều - 093602
* Dương Xuân Phúc - 093564

Tháng 11/2011

**Lời nói đầu**

Trong bối cảnh công nghệ thông tin phát triển như hiện nay.Mạng internet đã không còn lạ lẫm đối với mọi người mà ngược lại, ngày càng nhiều người sử dụng internet để phục vụ cho nhiều mục đích khác nhau của mình. Do vậy, các ứng dụng hướng dịch vụ để phục vụ cho các mục đích khác nhau của con người ngày càng phát triển. Vì thế nên mô hình hệ thống phân tán đang là xu hướng phát triển cho ngành phát triển phần mềm hiện nay và trong tương lai nhờ sự phổ biến và phát triển nhanh của ngành phát triển phần mềm hướng dịch vụ.

Hướng dịch vụ là một thay đổi lớn trong ngành phát triển phần mềm. Bất kể sử dụng SOAP hay các phương thức khác, các ứng dụng mà tương tác với nhau thông qua dịch vụ được xem là tiêu chuẩn mới. Vì thế, cùng với các nền tảng khác, Microsoft đã giới thiệu với các lập trình viên windows của mình một một mô hình mới để áp dụng cho sự thay đổi mới này – Windows Communication Foundation. Được chính thức giới thiệu như là một phần của .NET framework 3.0, và được cập nhật qua các phiên bản .NET 3.5 và mới nhất là 4.0. Là một phần đáng kể của .NET, WCF thể hiện bước phát triển đúng đắn của Microsoft.

Chúng ta sẽ tìm hiểu về công nghệ này ở phần nội dung của bài báo cáo này.

**Mục lục**

[I. TÌM HIỂU LÝ THUYẾT 5](#_Toc309802577)

[1. Giới Thiệu Tổng Quan Về WCF 5](#_Toc309802578)

[1.1. Windows Communication Foundation (WCF) là gì? 5](#_Toc309802579)

[1.2. Tại sao sử dụng WCF? 5](#_Toc309802580)

[1.3. Kiến trúc của WCF 6](#_Toc309802581)

[1.3.1 Các contracts (Các hiệp nghị) 6](#_Toc309802582)

[1.3.2 Runtime service (Dịch vụ thực thi) 7](#_Toc309802583)

[1.3.3 Message (Bản tin) 8](#_Toc309802584)

[1.3.4 Host and activation (Chứa và kích hoạt) 9](#_Toc309802585)

[1.4. Các tính năng của WCF 9](#_Toc309802586)

[1.4.1 Transaction (Giao dịch) 9](#_Toc309802587)

[1.4.2 Host (Chứa) 9](#_Toc309802588)

[1.4.3 Bảo mật 10](#_Toc309802589)

[1.5. Công cụ phát triển với WCF 10](#_Toc309802590)

[1.6. Ví dụ đầu tiên với WCF 10](#_Toc309802591)

[1.7. Sử dụng công cụ WCF Test Client để debug dịch vụ WCF 16](#_Toc309802592)

[2. Mô Hình Lập Trình Với WCF 19](#_Toc309802593)

[2.1. Sử dụng phương pháp hướng đối tượng hay hướng dịch vụ? 19](#_Toc309802594)

[2.2. Service Model (Mô hình dịch vụ) 20](#_Toc309802595)

[2.3. Các phương pháp lập trình với WCF 23](#_Toc309802596)

[2.3.1 Declarative programming (Phương pháp khai báo) 23](#_Toc309802597)

[2.3.2 Explicit programming (Phương pháp lập trình trực tiếp) 24](#_Toc309802598)

[2.3.3 Phương pháp sử dụng tập tin cấu hình 25](#_Toc309802599)

[2.4. Xây dựng một dịch vụ WCF 27](#_Toc309802600)

[2.4.1 Cài đặt WCF 27](#_Toc309802601)

[2.4.1.1 .NET Framework 3.5 SP1 27](#_Toc309802602)

[2.4.1.2 Visual Studio 2008 SP1 27](#_Toc309802603)

[2.4.2 Tạo dịch vụ WCF đầu tiên của bạn 27](#_Toc309802604)

[2.4.2.1 Tạo ứng dụng phía server 27](#_Toc309802605)

[2.4.2.2 Phát hành thông tin về dịch vụ 37](#_Toc309802606)

[2.4.2.3 Tạo ứng dụng phía client 41](#_Toc309802607)

[2.4.3 Cách khác để tạo tham chiếu ở client 43](#_Toc309802608)

[II. THIẾT KẾ CHƯƠNG TRÌNH 46](#_Toc309802609)

[1. Mô tả hệ thống 46](#_Toc309802610)

[1.1. Đối tượng sử dụng hệ thống 48](#_Toc309802611)

[1.2. Yêu cầu chức năng 48](#_Toc309802612)

[1.3. Yêu cầu phi chức năng 49](#_Toc309802613)

[2. Phân Tích – Thiết Kế Theo Chức Năng 49](#_Toc309802614)

[2.1. Use Case Diagram 49](#_Toc309802615)

[2.2. Xác định Use Case Tổng Quát 50](#_Toc309802616)

[2.3. Biểu đồ phân rã chức năng của hệ thống 50](#_Toc309802617)

[2.3.1. Biểu đồ luồng dữ liệu mức khung cảnh 51](#_Toc309802618)

[2.3.2. Biểu đồ luồng dữ liệu mức đỉnh 52](#_Toc309802619)

[2.3.3. Sơ đồ phân rã chức năng quản lý hồ sơ bệnh án 52](#_Toc309802620)

[2.3.4. Sơ đồ phân rã chức năng tìm kiếm 53](#_Toc309802621)

[2.3.5. Sơ đồ phân rã chức năng báo cáo thống kê 53](#_Toc309802622)

[2.4. Đặc tả Use Case 54](#_Toc309802623)

[3. Thiết Kế Cơ Sở Dữ Liệu 58](#_Toc309802624)

[3.1. Các Thực Thể 58](#_Toc309802625)

[3.2. Thiết Kế Cơ Sở Dữ Liệu 59](#_Toc309802626)

# TÌM HIỂU LÝ THUYẾT

1. Giới Thiệu Tổng Quan Về WCF
   1. Windows Communication Foundation (WCF) là gì?

WCF là công nghệ nền tảng nhằm thống nhất nhiều mô hình lập trình giao tiếp được hỗ trợ trong .NET 2.0 thành một mô hình duy nhất. Vào tháng 11 năm 2005, .NET 2.0 được Microsoft phát hành trong đó có cung cấp các hàm API riêng biệt cho các liên lạc dựa trên SOAP để tối đa hoá sự làm việc giữa các nền tảng sử dụng Web Services, đồng thời .NET 2.0 còn cung cấp các API để tối ưu việc liên lạc dựa trên mã nhị phân giữa các ứng dụng chạy trên hệ thống Windows gọi là .NET Remoting, các API cho các giao dịch phân tán, và API cho liên lạc dị bộ. WCF thống nhất các API này thành một mô hình duy nhất nhằm đáp ứng mô hình lập trình hướng dịch vụ.

WCF có thể sử dụng các bản tin SOAP giữa hai tiến trình, do đó làm cho các ứng dụng dựa trên WCF có thể làm việc với các tiến trình khác thông qua việc giao tiếp sử dụng bản tin SOAP. Khi một tiến trình WCF liên lạc với một tiến trình không là WCF, các bản tin SOAP được mã hoá trên cơ sở XML, nhưng khi nó liên lạc với một tiến trình WCF khác, bản tin SOAP có thể được tối ưu hoá dựa trên mã hoá nhị phân.

* 1. Tại sao sử dụng WCF?

Như phần trên đã trình bày, .NET 2.0 hỗ trợ rất nhiều phương pháp liên lạc giữa các ứng dụng khác nhau nhằm vào các mục tiêu khác nhau. Các phương pháp liên lạc này khá phức tạp và phải mất nhiều thời gian để làm chủ được công nghệ. Tuy nhiên kiến thức thu được từ việc triển khai một phương pháp ít có khả năng dùng được khi làm việc với phương pháp khác.

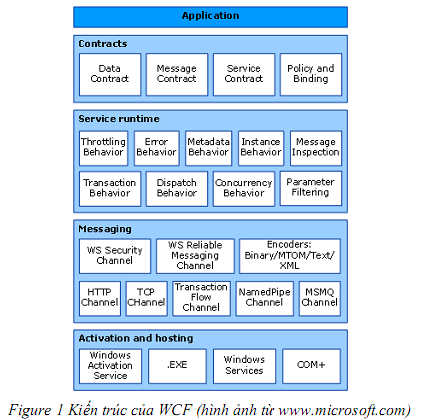
Với việc ra đời của WCF, mọi phương pháp liên lạc trước kia đều có thể thực hiện trên WCF. Do vậy nhà phát triển chỉ cần làm chủ được công nghệ WCF là có thể xây dựng các ứng dụng một cách nhanh chóng.

WCF là một mô hình lập trình cho phép nhà phát triển xây dựng các giải pháp dịch vụ đảm bảo tính ổn định, và bảo mật và thậm chí là đảm bảo giao dịch. Nó làm đơn giản hoá việc phát triển các ứng dụng nối kết và đưa ra cho nhà phát triển những giá trị mà có thể họ chưa nhận ra ngay, đó là cách tiếp cận phát triển hệ thống phân tán thống nhất, đơn giản, và quản lý được.

Do WCF được xây dựng trên cơ sở của .NET Framework 2.0 CLR, nó là tập các lớp cho phép các nhà phát triển xây dựng các ứng dụng hướng dịch vụ bằng môi trường lập trình quen thuộc của họ như VB.NET hay C#

* 1. Kiến trúc của WCF

Hình sau mô tả các lớp chủ yếu trong kiến trúc của Windows Communication Foundation (WCF)



* + 1. Các contracts (Các hiệp nghị)

Các contract trong WCF cũng giống như các hợp đồng/hiệp định mà bạn ký trong đời sống thật. Một hợp đồng bạn ký có thể chứa các thông tin như kiểu công việc bạn sẽ làm, và những thông tin mà bạn muốn đưa ra cho các bên khác. WCF contract cũng chứa các thông tin tương tự như vậy. Contract định nghĩa các đặc tả trong hệ thống bản tin.Thông thường có các loại contract sau:

* Contract dữ liệu mô tả các tham số cho các bản tin mà một dịch vụ có thể tạo ra hay sử dụng. Các tham số bản tin được định nghĩa bằng các tài liệu sử dụng ngôn ngữ đặc tả XML Schema (XSD), điều này cho phép các hệ thống hiểu XML có thể xử lý
* Tài liệu dễ dàng. Các dịch vụ khi liên lạc với nhau có thể không cần đồng ý với nhau về các kiểu, nhưng cần đồng ý về contract dữ liệu, nghĩa là đồng ý về các tham số và các kiểu trả về.
* Contract bản tin định nghĩa các phần có trong bản tin sử dụng các giao thức SOAP, và nó cho phép điều khiển sâu hơn tới các phần trong bản tin khi có yêu cầu sự chính xác như vậy.
* Contract dịch vụ đặc tả chi tiết các phương thức của dịch vụ, và được
* Phân phối như là một giao diện trong các ngôn ngữ lập trình như Visual Basic hay Visual C#. Có thể hình dung về contract dịch vụ một cách gián tiếp như sau: Đây là các kiểu dữ liệu của các bản tin của tôi, đây là nơi tôi cung cấp, và đây là các giao thức mà tôi có thể liên lạc”
* Các chính sách và các kết nối (bindings) mô tả các điều kiện cần có để giao tiếp với một dịch vụ. Các chính sách sẽ bao gồm cả các yêu cầu về bảo mật và các điều kiện khác cần phải có khi kết nối với một dịch vụ.
  + 1. Runtime service (Dịch vụ thực thi)

Lớp dịch vụ thực thi chứa các hành xử sẽ xảy ra trong quá trình thực hiện của dịch vụ, nghĩa là các hành xử thực thi của dịch vụ. Ta sẽ thấy một số các hành xử như sau:

* **Throttling behavior**: Điều khiển luồng nhằm quy định xem có bao nhiêu bản tin được xử lý
* **Error behavior**: Hành xử lỗi quy định những hành động khi lỗi xảy ra trong hệ thống
* **Metadata behavior**: Hành xử với các siêu dữ liệu quy định xem làm thế nào và khi nào thì các siêu dữ liệu được đưa ra bên ngoài dịch vụ
* **Instance behavior**: Hành xử thực thể quy định xem có bao nhiêu thực thể của dịch vụ đó được chạy
* **Transaction behavior**: Hành xử giao dịch cho phép việc rollback các giao dịch nếu xảy ra lỗi
* **Message inspection**: Kiểm tra bản tin đem lại cho dịch vụ khả năng kiểm tra tất cả hay một số phần của bản tin
* **Dispatch behavior**: Khi một bản tin được xử lý bởi nền tảng WCF, dịch vụ Dispatch behavior xác định xem bản tin được xử lý như thế nào.
* **Concurrency behavior**: Hành xử đồng thời xác định xem việc xử lý thế nào với việc đa luồng của mỗi dịch vụ hay mỗi thực thể của dịch vụ. Hành xử này giúp cho việc điều khiển số lượng luồng có thể truy nhập tới một thực thể của dịch vụ.
* **Parameter filtering**: Khi một bản tin được đưa tới một dịch vụ, sẽ xảy ra một số hành động dựa trên nội dung phần đầu đề của bản tin. Phần lọc tham số sẽ thực hiện lọc các đầu đề bản tin và thực hiện các hành động đặt sẵn dựa trên việc lọc đầu đề bản tin.
  + 1. Message (Bản tin)

Lớp bản tin là tập hợp các kênh. Mỗi kênh là một thành phần xử lý bản tin theo một cách nào đó. Một tập các kênh thường được gọi là ngăn xếp kênh.Các kênh làm việc trên bản tin và trên đầu đề của bản tin. Lớp này khác với lớp thực thi dịch vụ chủ yếu bởi sự khác nhau trong việc xử lý nội dung bản tin.

Có hai kênh khác nhau là kênh vận chuyển (transport channel) và kênh điều khiển (control channel).

* **Kênh vận chuyển** phụ trách việc đọc và ghi các bản tin từ mạng (network) hoặc từ một số điểm giao dịch bên ngoài)
* **Kênh điều khiển** thực hiện xử lý bản tin theo giao thức, thông thường làm việc bằng cách đọc và ghi thêm các đầu đề cho bản tin.
  + 1. Host and activation (Chứa và kích hoạt)

Nhìn một cách tổng thể thì một dịch vụ thực chất là một chương trình. Cũng giống như các chương trình khác, một dịch vụ cần phải chạy trong một tệp thực thi. Dịch vụ này thường được gọi là dịch vụ tự chứa.

Các dịch vụ còn có thể được chứa, hoặc chạy trong một tệp thực thi được quản lý bởi một agent bên ngoài như IIS hay Windows Activation Services (WAS). WAS cho phép WCF được kích hoạt một cách tự động khi phân phối tới một máy tính có chạy WAS.

* 1. Các tính năng của WCF

WCF có nhiều tính năng và sẽ được mô tả chi tiết trong toàn bộ khoá học.Mục này chỉ nhằm mục đích thảo luận một số tính năng của WCF.Danh sách các tính năng ở đây không phải là danh sách hoàn chỉnh, mà chỉ hy vọng là danh sách các tính năng “đỉnh” nhất.

* + 1. Transaction (Giao dịch)

Một giao dịch là một đơn vị của công việc. Một giao dịch đảm bảo chắc chắn rằng mọi thứ diễn ra trong giao dịch thành công hay thất bại đều là kết quả tổng thể. Ví dụ, nếu một giao dịch chứa ba mục công việc cần thực hiện, trong quá trình thực hiện giao dịch, một trong số các mục đó bị thất bại, khi đó cả ba mục sẽ là thất bại.Giao dịch chỉ thành công khi cả ba mục công việc đều thành công.Giao dịch thường thấy trong các thao tác với cơ sở dữ liệu.

WCF cho phép đưa vào việc xử lý giao dịch như trên với các liên lạc.Nhà phát triển có thể nhóm các liên lạc với nhau thành các giao dịch.Ở mức doanh nghiệp, tính năng này cho phép bạn thực hiện các công việc giao dịch qua các nền tảng khác nhau.

* + 1. Host (Chứa)

WCF cho phép các dịch vụ được chứa trong một số lớn các môi trường khác nhau, như Windows NT Services, Windows Forms, và ứng dụng console, cũng như ở trên IIS (Internet Information Server) và WAS (Windows Activation Services).

Chứa ứng dụng trên IIS còn có thêm các lợi điểm khác là dịch vụ có thể nhận các ưu điểm của rất nhiều tính năng có sẵn trên IIS, ví dụ IIS có thể điều khiển một cách tự động việc bắt đầu hay kết thúc một dịch vụ.

* + 1. Bảo mật

Bảo mật là tính năng không thể thiếu trong WCF nói riêng và trong liên lạc nói chung. Trong WCF, tất cả mọi thứ từ các bản tin tới các client hay server đều phải xác thực và WCF có tính năng để đảm bảo rằng các bản tin không bị lẫn trong quá trình vận chuyển. WCFbao gồm việc đảm bảo tính toàn vẹn và bảo mật của bản tin.

WCF còn cho phép bạn tích hợp ứng dụng của bạn với cơ sở hạ tầng bảo mật sẵn có, bao gồm cả các chuẩn bên ngoài môi trường Windows bằng cách sử dụng các bản tin SOAP bảo mật.

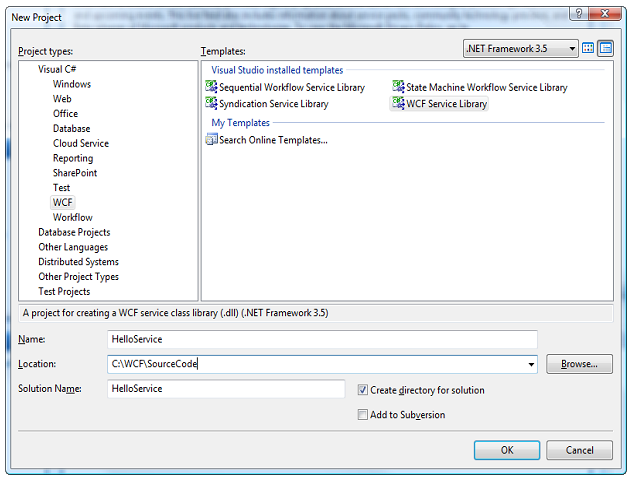
* 1. Công cụ phát triển với WCF

Để phát triển ứng dụng với WCF ta cần các phần mềm sau:

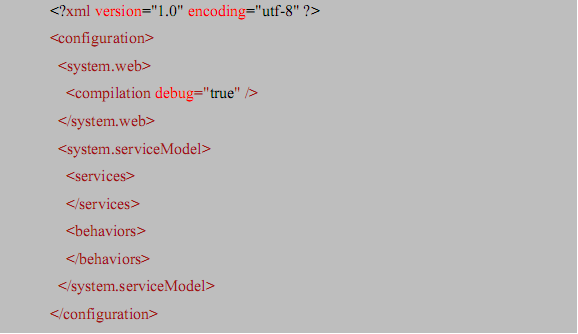
* .**NET Framework 3.5.**  tải về bộ cài của .NET Framework 3.5 tại trang web của Microsoft. Địa chỉ như sau: http://www.microsoft.com/downloads/details.aspx?familyid=ab99342f-5d1a-413d-8319-81da479ab0d7&displaylang=en
* **Microsoft Visual Studio 2008.** Bản dùng thử có thể tải về ở đây: http://msdn.microsoft.com/en-us/visualc/aa700831.aspx
* **Hoặc Microsoft Visual Studio Express Editions**. Địa chỉ tải về: http://www.microsoft.com/express/
  1. Ví dụ đầu tiên với WCF

Xây dựng ứng dụng viết chữ Hello World ra màn hình sử dụng WCF. Các bước làm việc như sau:

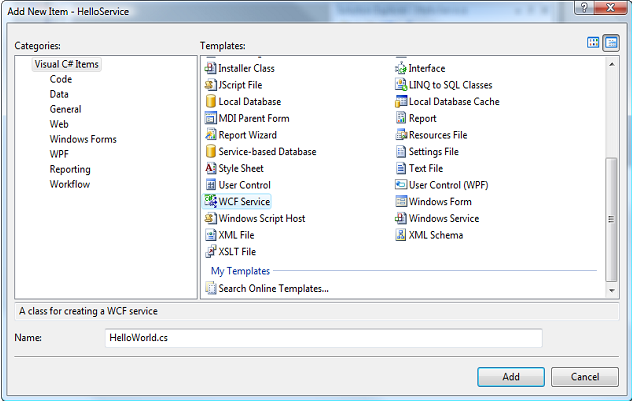
1. Mở Visual Studio 2008, chọn Create New Project, trong phần WCF chọn WCF Service Library như hình dưới. Đặt tên project là HelloService



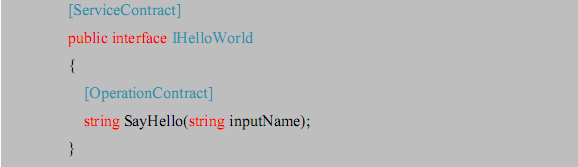
1. Sau khi project được tạo ra, xoá 2 tệp IService1.cs và Service1.cs. Đồng thời làm sạch tệp app.config thành như sau:



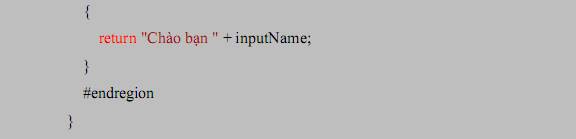
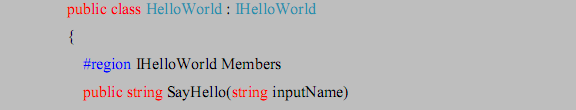
1. Kích chuột phải vào project HelloService, chọn “Add  New Item”. Sau đó chọn thêm một WCF Service, đặt tên nó là HelloWorld như hình dưới. Hệ thống sẽ thêm vào 2 tệp cho bạn:



3.1. Tệp IHelloWorld.cs chứa một giao diện gọi là IHelloWorld, đây là contract cho dịch vụ của bạn. Thay hàm DoWork thành SayHello như sau:



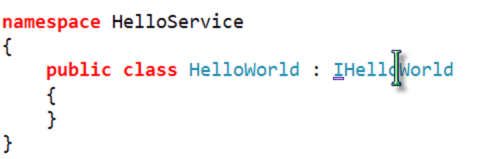
3.2. Tệp HelloWorld.cs, chứa lớp gọi là HelloWorld, lớp này sẽ cài đặt contract cho dịch vụ của bạn. Thay hàm DoWork thành SayHello như sau:



Khi thực hiện cài đặt một dịch vụ WCF, thông thường bạn định nghĩa phần giao diện trước (các interface) sau đó mới thực hiện thông qua một lớp để cài đặt giao diện đó. Bạn hoàn toàn có thể làm bằng tay để viết lại phần khai báo các phương thức đã có trong giao diện. Tuy vậy, có một cách nhanh hơn giúp bạn khai báo các phương thức cũng như thuộc tính của giao diện bằng Visual Studio. Cách làm như sau:

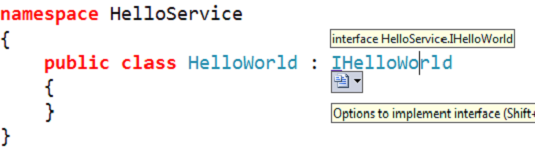
**Bước 1**. Mở file chứa class cần cài đặt. Di chuyển con trỏ tới tên giao diện như

hình dưới



*Figure 4 Di chuyển con trỏ tới tên giao diện*

**Bước 2**. Bấm chuột vào hình chữ nhật nhỏ dưới chữ IHelloWorld (giống smart tag trong Microsoft Word), xem hình dưới



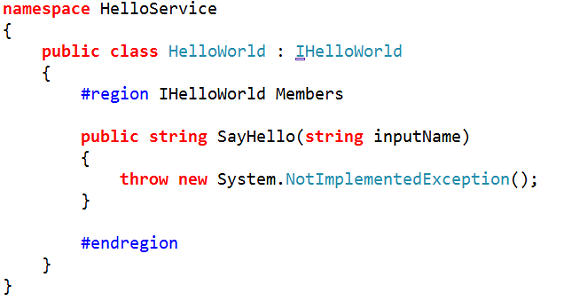
*Figure 5 Kích hoạt SmartTag*

**Bước 3**.Mở menu popup và chọn một trong 2 mục.



*Figure 6 Chọn mục menu để cài đặt giao diện*

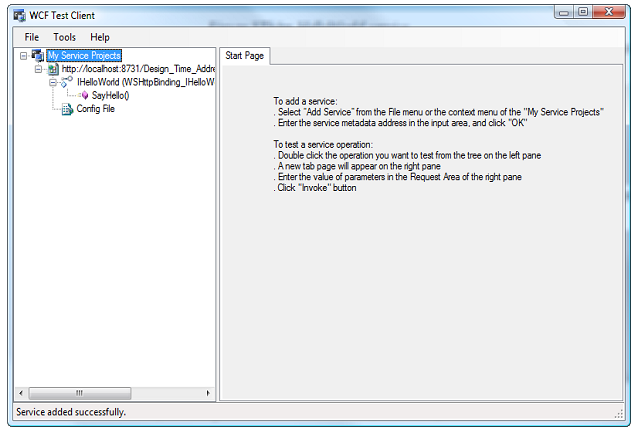
**Kết quả.**Sau khi chọn một mục menu ta sẽ được kết quả như hình dưới.



*Figure 7 Kết quả*

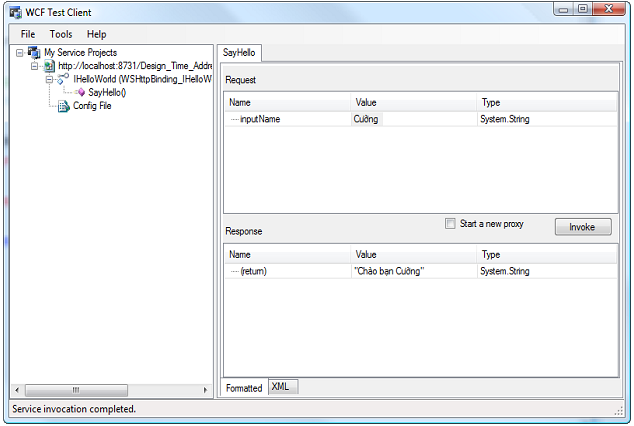
Như các bạn đã thấy, sử dụng tính năng này của Visual Studio cho phép chúng ta thực hiện tạo nhanh các khai báo cần thiết để cài đặt giao diện.

1. Vậy là chúng ta đã có một dịch vụ hoàn chỉnh. Bấm F5 để chạy. Visual Studio sẽ tự động gọi chương trình WCF Test Client (là chương trình client để test dịch vụ của WCF). Xem hình dưới.



*Figure 8 WCF Test Client*

1. Chuyển tới phương thức SayHello, đưa vào tham số, giả sử là tên của bạn, sau đó bấm nút Invoke. Kết quả xem hình dưới.



*Figure 9 Kết quả thực thi dịch vụ HelloWorld*

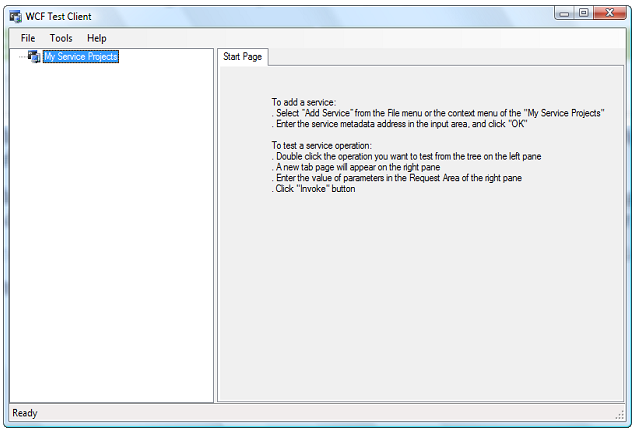
* 1. Sử dụng công cụ WCF Test Client để debug dịch vụ WCF

Công cụ WCF Test Client có thể được kích hoạt theo 2 cách.

* Cách thứ nhất như các bạn đã thấy ở ví dụ trên. Công cụ này được kích hoạt một cách tự động khi ta thực hiện debug một WCF Service Library.
* Cách thứ 2 là chạy trực tiếp và thêm vào các dịch vụ cần debug. Cách này thông thường hay sử dụng để debug dịch vụ khi host dịch vụ bởi IIS. Công cụ WCF Test Client có thể tìm thấy ở thư mục sau:

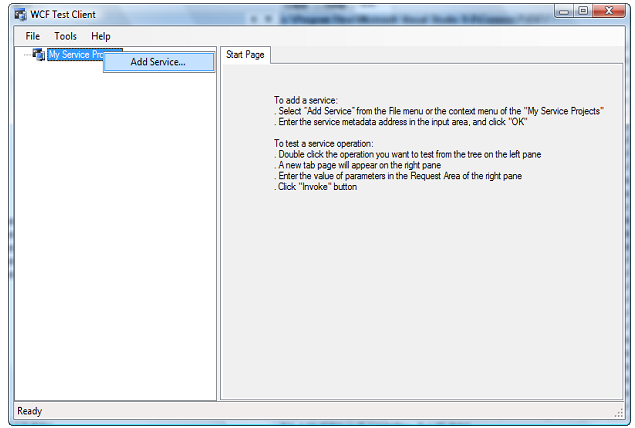
C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\WcfTestClient.exe

Khi kích hoạt trực tiếp WCF Test Client, sẽ thấy màn hình khởi động như sau:



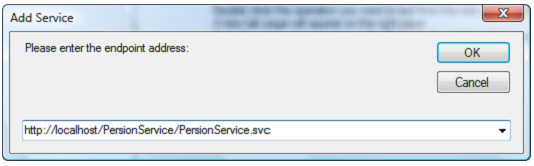
*Figure 10 Màn hình khởi động của WCT Test Client*

Bấm chuột phải vào My Service Projects, bạn có thể thêm vào các dịch vụ WCF để thực hiện debug

****

*Figure 11 Thêm một dịch vụ vào WCF Test Client*

Màn hình dưới đây là ví dụ thêm vào một dịch vụ được host trên IIS



*Figure 12 Thêm vào một dịch vụ được host trên IIS*

-----oOo-----

1. Mô Hình Lập Trình Với WCF
   1. Sử dụng phương pháp hướng đối tượng hay hướng dịch vụ?

Nếu các bạn đã học qua môn học về các phương pháp lập trình hẳn sẽ thấy có 2 phương pháp chính là hướng thủ tục và hướng đối tượng. Và phương pháp hướng đối tượng trong thời gian gần đây được phát triển rất mạnh và được hỗ trợ ngay trong các ngôn ngữ lập trình như C# hay VB.NET. Khi làm việc với các dịch vụ web bạn đã làm quen với một phương pháp nữa là lập trình hướng dịch vụ (Service-

oriented programming). Microsoft đã cung cấp nhiều công cụ trên .NET Framework để hỗ trợ phương pháp lập trình này thông qua các lớp trong không gian tên: System.Web.Services.

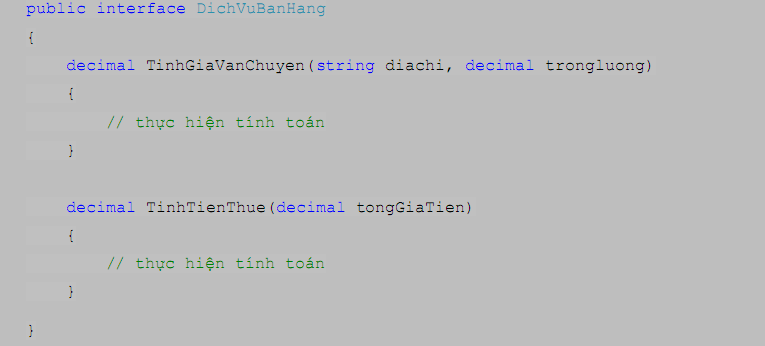
Như vậy khi làm việc với WCF bạn có hai lựa chọn là hướng đối tượng và hướng dịch vụ.Vậy ta nên sử dụng phương pháp nào?Câu trả lời là dùng cả hai. Nói một cách đơn giản là phương pháp hướng đối tượng được sử dụng để phát triển các ứng dụng trên desktop, còn phương pháp hướng dịch vụ được sử dụng để kết nối các ứng dụng đó với nhau. Điều quan trọng ở đây là làm sao để hiểu được sự khác nhau giữa hai phương pháp và hiểu được khi nào chúng được sử dụng và sử dụng như thế nào đồng thời cũng phải hiểu về các lợi ích chúng cung cấp.

Về hướng đối tượng có thể hiểu như sau. Các ứng dụng hướng đối tượng là hai hay nhiều lớp phụ thuộc lẫn nhau và chia sẻ chung các kiểu dữ liệu. Những lớp này liên lạc với nhau thông qua các lời gọi các hàm mà lớp đối tượng cung cấp.

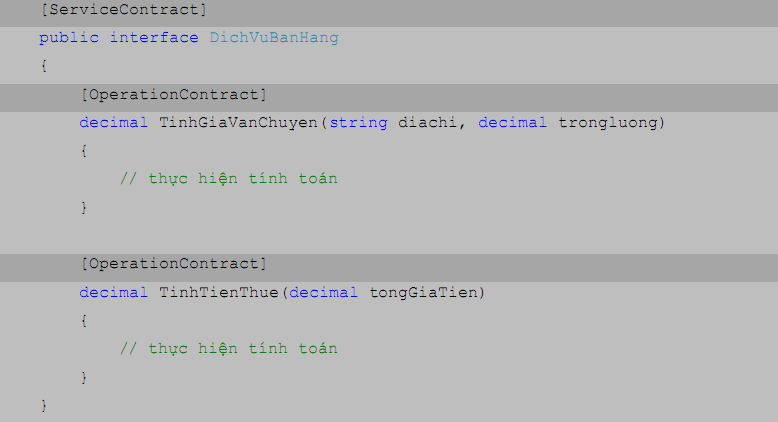
Các ứng dụng hướng dịch vụ là các chương trình không biết gì về nhau.Mỗi ứng dụng liên lạc với ứng dụng khác thông qua các bản tin. Điểm đặc biệt là các bản tin này được gửi từ một ứng dụng sang ứng dụng khác mà không quan tâm tới nền tảng mà dịch vụ đang chạy.

Khi phát triển các dịch vụ WCF, điều quan trọng là cần hiểu sự liên kết giữa hướng đối tượng và hướng dịch vụ.Khi làm việc với .NET Framework bạn chắc chắn rất quen thuộc với thuật ngữ lớp (class) và giao diện (interface).Các thuật ngữ này vẫn được sử dụng khi phát triển dịch vụ WCF.Các lớp và giao diện là phần hướng đối tượng trong WCF, còn phần hướng dịch vụ trong WCF sẽ được thấy khi bạn đưa vào các thuộc tính WCF để định nghĩa các thực thể.

Ví dụ, lớp sau đây định nghĩa một giao diện hướng đối tượng



Ta sẽ có phần hướng dịch vụ cho dịch vụ WCF khi thêm vào các thuộc tính cho giao diện ở trên



Như vậy là qua ví dụ trên các bạn có thể thấy mối liên kết giữa phương pháp hướng đối tượng và hướng dịch vụ trong WCF. Các bạn chưa cần quan tâm tới các thuộc tính [ServiceContract] và [OperationContract] vội, bởi vì những thuộc tính này, và còn nhiều thứ khác nữa sẽ được giới thiệu một cách chi tiết khi thích hợp.

* 1. Service Model (Mô hình dịch vụ)

Nếu bạn đã từng làm việc với dịch vụ web, bạn sẽ thấy mô hình này quen thuộc với bạn theo một cách nào đó. Khi bạn tạo một dịch vụ web, bạn thực sự tạo ra một dịch vụ (service).Dịch vụ web chứa một tài liệu XML để mô tả tất cả mọi thứ cần biết về dịch vụ đó.Tài liệu này được mô tả bằng ngôn ngữ Web Service Description Language (ngôn ngữ mô tả dịch vụ web). Nó chứa ba phần:

* **Service (dịch vụ):** Chứa thông tin về vị trí của dịch vụ
* **Binding:** Chứa thông tin về cách liên lạc với dịch vụ, như dịch vụ sử dụng giao thức gì, vv.
* **PortType (kiểu cổng):** Giải thích về dịch vụ sẽ làm gì

Mô hình dịch vụ trên WCF cũng tương tự như với mô hình dịch vụ web.Điểm khác biệt là ở cách đặt tên.Trong WCF các phần không được gọi là service, binding, và portType mà được gọi tương ứng là address (địa chỉ), binding, và contract.

Mô hình dịch vụ WCF được cung cấp trong không gian tên System.ServiceModel.Không gian tên này chứa rất nhiều lớp, nhưng bạn hoàn toàn không cần biết toàn bộ chúng. Để sử dụng mô hình và xây dựng dịch vụ, ta thường sử dụng một số lớp sau:

|  |  |
| --- | --- |
| **Lớp** | **Mô tả** |
| BasicHTTPBinding | Là binding mà các điểm cuối dịch vụ có thể sử dụng để liên lạc với các ứng dụng khách và dịch vụ web (ASMX) |
| NetMsmqBinding | Là binding mà các điểm cuối dịch vụ có thể sử dụng để liên lạc với các MSMQ khách và các dịch vụ khác |
| NetNamedPipeBinding | Là binding mà các điểm cuối dịch vụ có thể sử dụng để liên lạc với các ứng dụng khách/dịch vụ trên cùng một máy |
| NetTCPBinding | Là binding mà các điểm cuối dịch vụ có thể sử dụng để liên lạc với các ứng dụng khách/dịch vụ ở các máy khác nhau |
| WSHTTPBinding | Là binding mà các điểm cuối dịch vụ có thể sử dụng để liên lạc với các ứng dụng khách/dịch vụ sử dụng các giao dịch phân tán và các phiên làm việc bảo mật và tin cậy được. |
| EndpointAddress | Lớp biểu diễn địa chỉ duy nhất được cung cấp và truy xuất được cho máy khách để liên lạc với điểm cuối dịch vụ |
| EndpointAddressBuilding | Là phương pháp để tạo mới các địa chỉ đầu cuối với các giá trị tham số xác định |
| ChannelFactory | Là phương pháp trong đó các kiểu kênh khác nhau được tạo ra và quản lý, và đưa tới cho các ứng dụng khách để gửi bản tin tới các điểm cuối |
| Identity | Cách mà một định danh được xác định, cho phép xác thực giữa các điểm cuối khi trao đổi bản tin |
| MessageHeader | Biểu diễn nội dung của một đầu đề bản tin SOAP |
| ServiceHost | Phương pháp cung cấp vật chứa cho các dịch vụ |
| ReliableSession | Cung cấp truy xuất tới các thuộc tính của thành phần binding trong phiên làm việc tin cậy. |

Để định nghĩa việc liên lạc của dịch vụ, ta thường hay sử dụng các lớp sau

|  |  |
| --- | --- |
| **Lớp** | **Mô tả** |
| AddressHeader | Phần đầu đề chưa thông tin địa chỉ được sử dụng để xác định và liên lạc với một điểm cuối |
| AddressHeaderCollection | Một tập hợp các đầu đề địa chỉ |
| Binding | Tập hợp các thành phần binding, mỗi binding định nghĩa cách mà một điểm cuối liên lạc với thế giới bên ngoài |
| BindingContext | Cung cấp địa chỉ và thông tin binding cần thiết cho việc xây dựng kênh |
| BindingElement | Biểu diễn một thành phần binding, được sử dụng để xây dựng các binding |
| CustomBinding | Sử dụng để định nghĩa và xây dựng một tuỳ biến binding từ một tập các thành phần binding |
| Message | Một đơn vị của liên lạc giữa các điểm cuối |
| MessageHeader | Nội dung của đầu đề bản tin SOAP |
| MessageHeaders | Tập hợp các đầu đề bản tin |

* 1. Các phương pháp lập trình với WCF

Có một số phương pháp lập trình với WCF, mỗi phương pháp có ưu điểm và khuyết điểm riêng của nó.Điều đặc biệt về WCF là luôn có hơn một cách để giải quyết một vấn đề trong WCF, và bạn không nhất thiết phải chọn duy nhất một phương pháp nào.Trong thực tế, cách làm tốt nhất là tổ hợp các phương pháp để có được sự linh hoạt và mềm dẻo cho dịch vụ của bạn.

Có ba phương pháp hay được sử dụng khi phát triển dịch vụ WCF như sau:

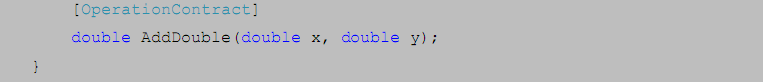
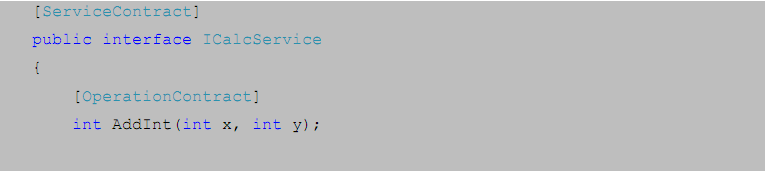
* Phương pháp khai báo
* Phương pháp lập trình trực tiếp
* Phương pháp sử dụng tập tin cấu hình
  + 1. Declarative programming (Phương pháp khai báo)

Lập trình khai báo đạt được thông qua các thuộc tính.Những thuộc tính này được sử dụng để định nghĩa các contract và xác định hành xử của dịch vụ.Chúng được sử dụng để xác định thêm các tham số để thay đổi các chi tiết của contract và hành xử dịch vụ.

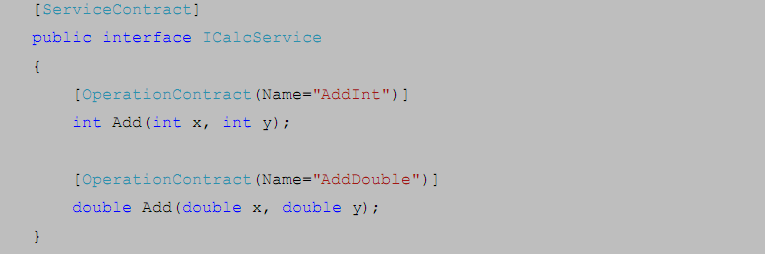
Thuộc tính ServiceContract dùng để quy định là giao diện này định nghĩa các chức năng của một dịch vụ.Thuộc tính *OperationContract* được sử dụng ở các hàm để quy định rằng hàm này được khai báo là một phần của dịch vụ.Đó là tất cả những gì cần để tạo ra một dịch vụ WCF.

Thêm nữa, bạn không nhất thiết phải sử dụng các giao diện (interface) khi cài đặt một dịch vụ, điều này cũng giống như việc bạn không cần phải sử dụng giao diện để định nghĩa một lớp.Tuy vậy bạn nhất thiết phải quy định phần nào thuộc về dịch vụ.Bạn có thể định nghĩa những phần khác cần cho giao diện, nhưng chỉ những hàm (phương thức) có gắn thuộc tính *[OperationContract].*

Ví dụ ta có một dịch vụ thực hiện phép tính cộng giữa 2 số nguyên AddInt và 2 số thực *AddDouble*. Ta khai báo dịch vụ như sau:



Như vậy dịch vụ của chúng ta sau khi khai báo sẽ có 2 phương thức (khai báo với thuộc tính *OperationContract*) là *AddInt* và *AddDouble*. Tuy nhiên khi khai báo trong C#, việc đặt tên *AddInt* và *AddDouble*, và có thể có một số hàm add cho các kiểu dữ liệu khác, có thể rút gọn lại thành một tên hàm Add mà thôi. Nhưng các dịch vụ lại không cho phép đặt trùng tên hàm như thế. Chúng ta có thể khai báo thêm với thuộc tính OperationContract để thực hiện, cách làm như sau:



Các bạn có thể thấy là chúng ta sử dụng được phép nạp chồng tên trong C# và sử dụng thêm tham số *Name* để quy định thêm tên hàm ở dịch vụ. Ngoài ưu điểm trong việc giải quyết nạp chồng tên hàm, ta còn thấy một lợi ích khác nữa là, việc quy định tham số *Name* trong thuộc tính *OperationContract* còn cho ta thêm linh hoạt trong việc đổi tên các hàm trong giao diện mà không làm thay đổi định nghĩa dịch vụ, nghĩa là các ứng dụng khác sử dụng dịch vụ này không cần phải biên dịch lại.

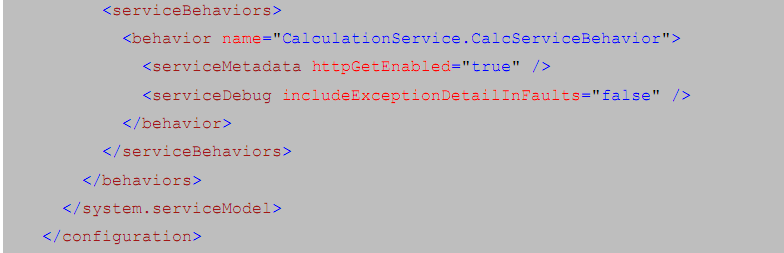
* + 1. Explicit programming (Phương pháp lập trình trực tiếp)

Là phương pháp lập trình hướng đối tượng, bạn làm việc trực tiếp với các lớp và giao diện cung cấp bởi mô hình đối tượng của WCF.Làm việc trực tiếp với mô hình đối tượng cho phép nhà phát triển tính linh hoạt cao hơn và khả năng điều khiển tốt hơn thông qua mã nguồn của họ. Thêm nữa nó cho phép điều khiển sâu hơn rất nhiều so với phương pháp khai báo và phương pháp sử dụng tập tin cấu hình.

* + 1. Phương pháp sử dụng tập tin cấu hình

Cũng giống như phương pháp khai báo, có rất nhiều thứ mà bạn có thể quy định liên quan đến hành xử của một dịch vụ thông qua tập tin cấu hình của dịch vụ.Điều hay trong cách tiếp cận này là những thay đổi ở tập tin cấu hình hoàn toàn không cần phải biên dịch lại dịch vụ mới sử dụng được.

Sau đây là ví dụ sử dụng tập tin cấu hình để định nghĩa dịch vụ tính toán trong ví dụ của phần phương pháp khai báo.



* 1. Xây dựng một dịch vụ WCF
     1. Cài đặt WCF
        1. .NET Framework 3.5 SP1

Để xây dựng một dịch vụ WCF, đầu tiên bạn cần phải cài đặt .NET Framework 3.5 SP1.Thực ra chỉ cần .NET Framework 3.0 là đủ, tuy nhiên .NET Framework 3.5 SP1 còn cung cấp thêm cho bạn nhiều tính năng nữa, nên bạn nên cài .NET Framework 3.5 SP1. Bản cài đặt của framework được Microsoft cung cấp ở trang web <http://www.microsoft.com/downloads/details.aspx?familyid=ab99342f-5d1a-413d-8319-81da479ab0d7&displaylang=en> hoặc tại <http://download.microsoft.com/download/2/0/e/20e90413-712f-438c-988e-fdaa79a8ac3d/dotnetfx35.exe>

* + - 1. Visual Studio 2008 SP1

Sau khi cài đặt .NET Framework 3.5 SP1, bạn thực hiện cài đặt Visual Studio 2008 bản Express hoặc bản Professional tuỳ theo bạn có bản nào. Nếu kinh phí hạn hẹp, bạn có thể tải về bản Visual Studio 2008 Express Edition miễn phí trên trang web của Microsoft, link ở đây: <http://go.microsoft.com/?linkid=9350817>

Giờ đây bạn đã sẵn sàng để tạo ra dịch vụ đầu tiên trên WCF

* + 1. Tạo dịch vụ WCF đầu tiên của bạn

Bạn có thể hình dung ứng dụng chúng ta sẽ xây dựng như sau. Công ty Contoso cần xây dựng một hệ thống quản lý các nhân viên của công ty. Ban đầu, chúng ta cần phải xây dựng một ứng dụng ở server

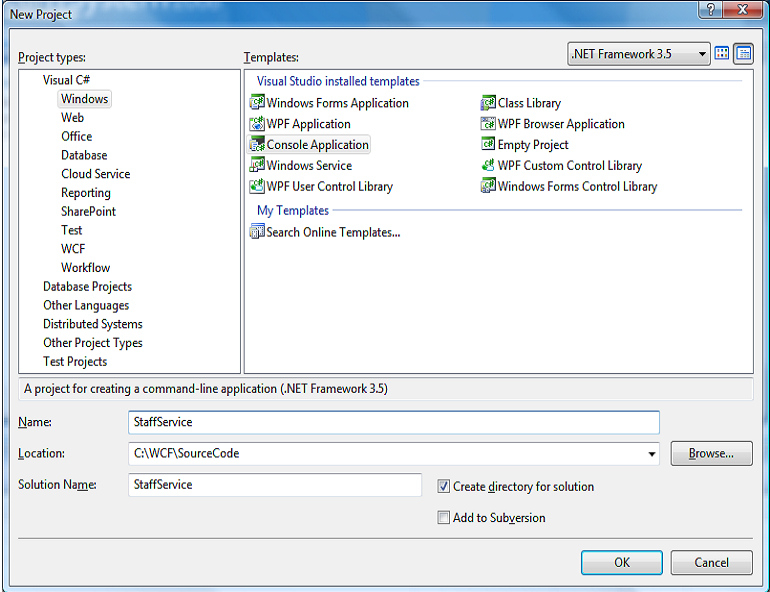
* Cung cấp danh sách các nhân viên, và
* Cho phép hỏi về ngày sinh của một nhân viên nào đó.

Sau đó cần một ứng dụng phía client để làm những việc sau

* Hiển thị danh sách các nhân viên
* Chọn một nhân viên và hiển thị ngày sinh của nhân viên đó.
  + - 1. Tạo ứng dụng phía server

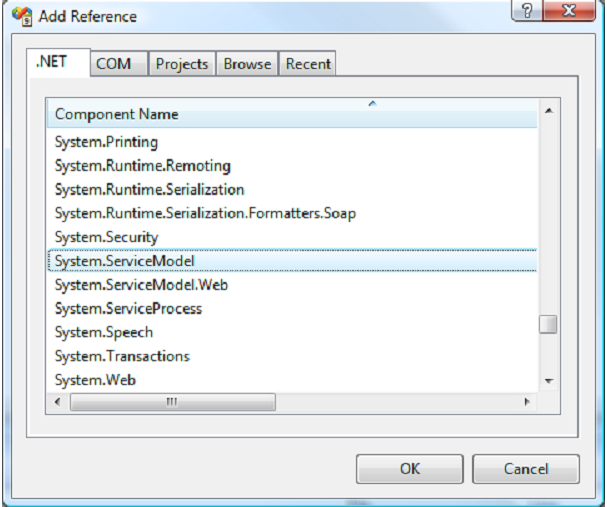
**Bước 1**. Tạo ứng dụng

1. Mở Visual Studio 2008, chọn tạo mới C# Console Project đặt tên là StaffService



*Figure 13 Tạo mới project*

2. Thêm tham chiếu tới System.ServiceModel.dll.

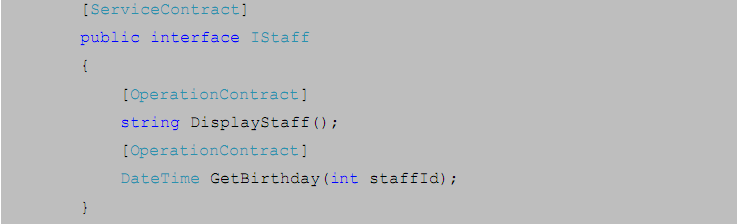


*Figure 14 Tham chiếu tới ServiceModel*

**Bước 2**. Tạo contract dịch vụ

1. Định nghĩa giao diện sẽ được sử dụng như là contract cho ứng dụng của chúng ta, thêm vào thuộc tính ServiceContract cho giao diện đó

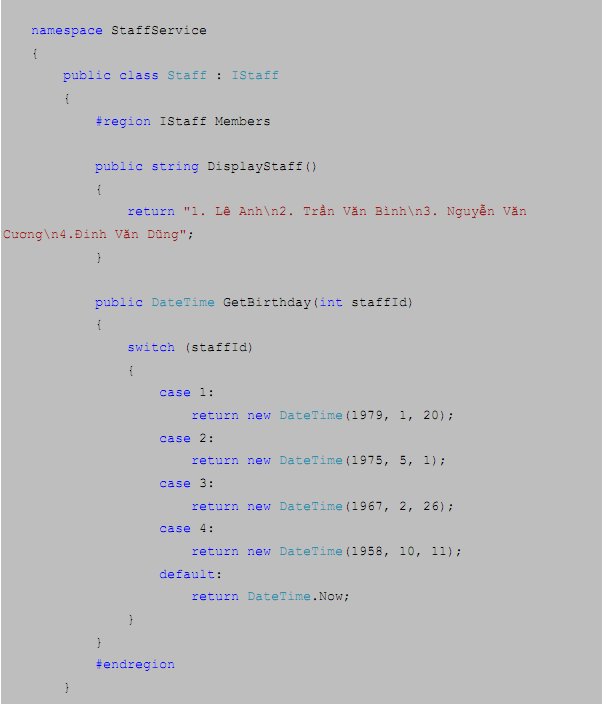
2. Định nghĩa 2 hàm như đoạn mã nguồn sau, các hàm này đóng vai trò là các contract operations miêu tả chức năng của dịch vụ



**Bước 3**. Cài đặt dịch vụ

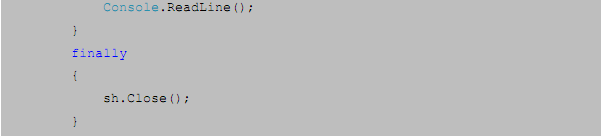
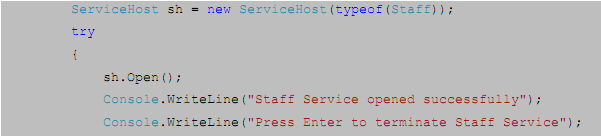
1. Định nghĩa lớp Staff để cài đặt giao diện IStaff

2. Thực hiện cài đặt cho 2 hàm được định nghĩa trong giao diện IStaff



**Bước 4**. Tạo vật chứa dịch vụ

1. Thêm đoạn mã nguồn sau vào hàm main

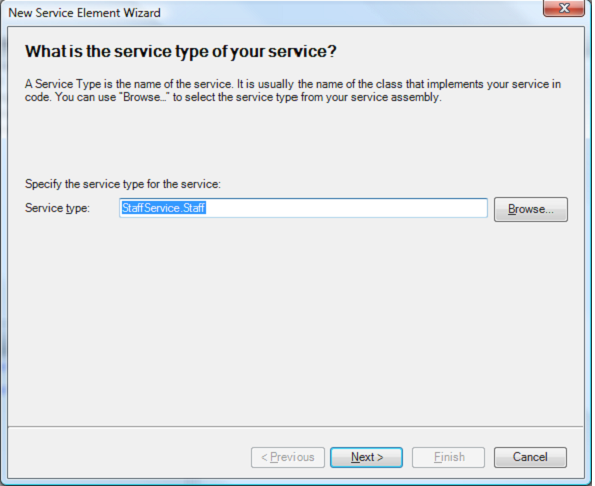


**Bước 5.** Tạo các cấu hình dịch vụ

1. Compile project của bạn, đảm bảo rằng không có lỗi xảy ra trong quá trình biên dịch

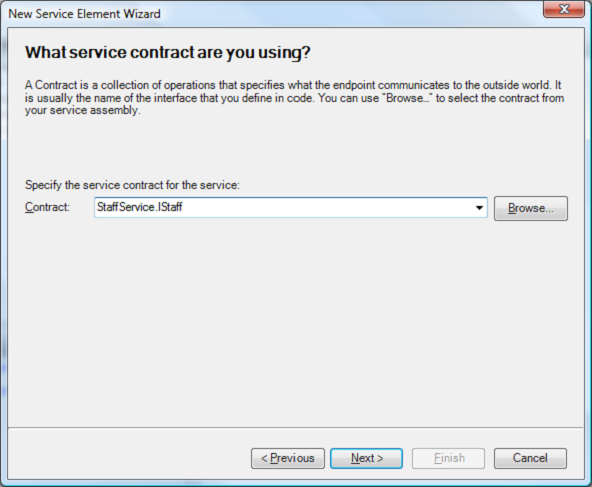
2. Mở trình soạn thảo dịch vụ, Service Configuration Editor và nạp tệp ứng dụng, bằng cách trong Visual Studio 2008, chọn menu Tools, chọn lựa chọn WCF Service Configuration Editor

3. Chọn New Config, đặt tên là Staff.Service



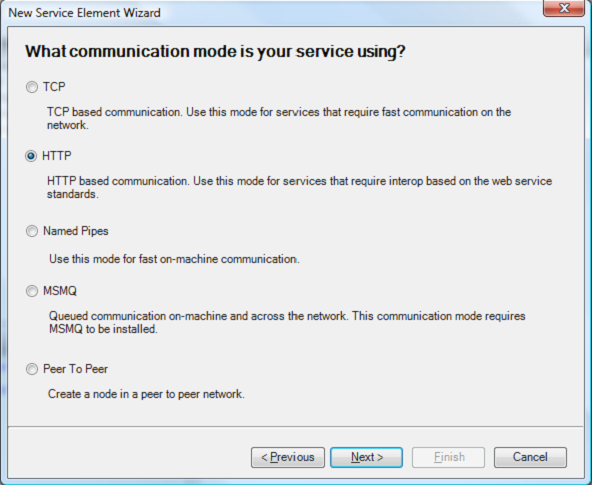
*Figure 15 Tạo config cho service*

4. Đặt contract dịch vụ là StaffService.IStaff



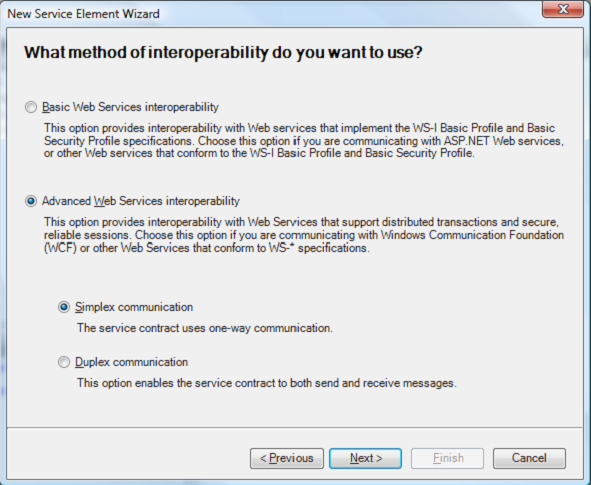
*Figure 16 Đặt tên cho contract dịch vụ*

5. Chọn cách liên lạc là HTTP



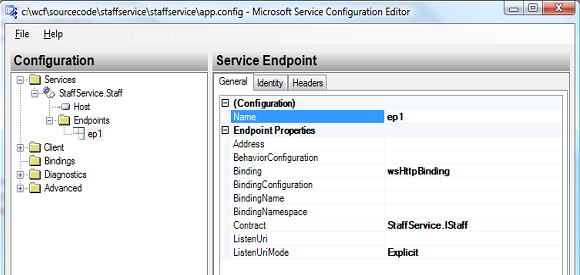
*Figure 17 Chọn cách liên lạc*

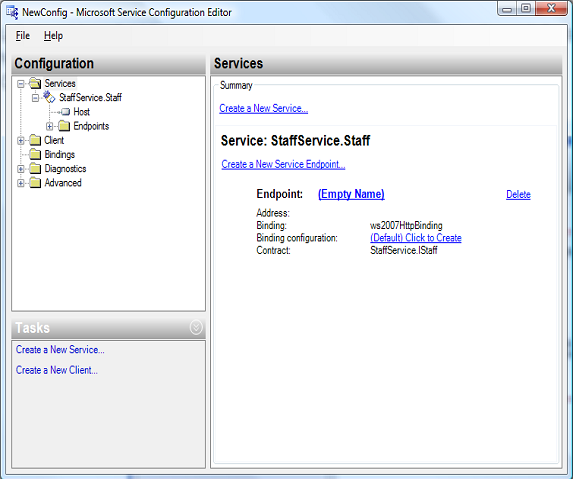
6. Tiếp theo chọn phương pháp làm việc là Advanced Web Services interoperability, và đặt kiểu liên kết là Simplex Communication



*Figure 18 Chọn Simplex communication*

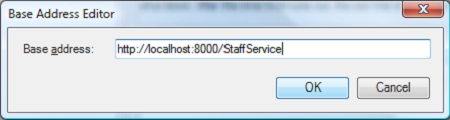
7. Đặt mặc định (trống) cho trường địa chỉ, và bấm Finish, bạn sẽ thấy kết quả sau





*Figure 19 Kết qủa cấu hình dịch vụ*

8. Giờ bạn bấm vào biểu tượng Host phía bên trái, sau đó chọn thêm mới địa chỉ cơ sở cho dịch vụ của bạn, đặt địa chỉ là <http://localhost:8000/StaffService>



*Figure 20 Thêm địa chỉ cơ sở*

9. Mở rộng nút EndPoints, chọn điểm cuối, rồi đặt tên là ep1

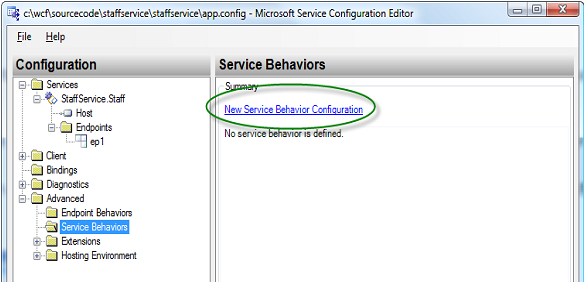
10. Kết thúc quá trình này bạn sẽ có tệp cấu hình như sau:



* + - 1. Phát hành thông tin về dịch vụ

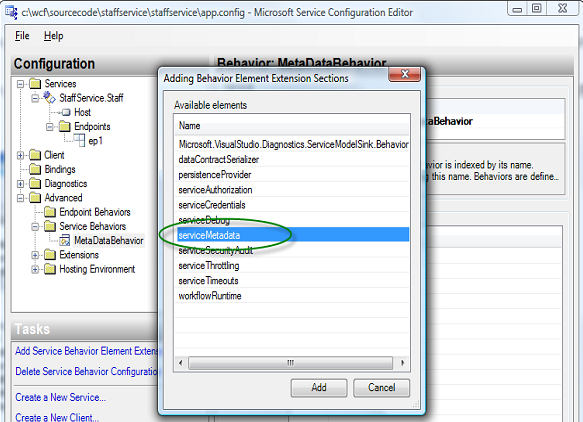
Để phát hành thông tin về dịch vụ cho các ứng dụng khác khai thác, ta cần theo các bước sau:

1. Mở rộng nút Advanced (bên trái), sau đó chọn nút Service Behaviors và bấm vào New Service Behavior Configuration ở bên phải



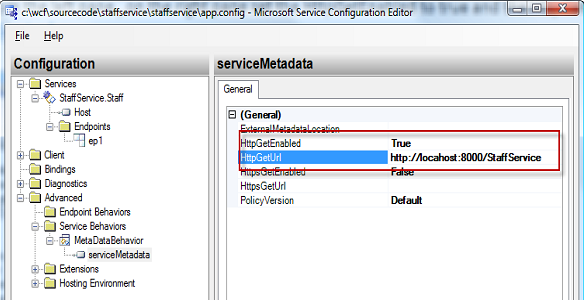
*Figure 21 Tạo mới cấu hình cho hành xử dịch vụ*

2. Đặt tên là MetaDataBehavior, bấm Add để thêm thành phần hành xử, chọn ServiceMetaData



*Figure 22 Thêm hành sử siêu dữ liệu*

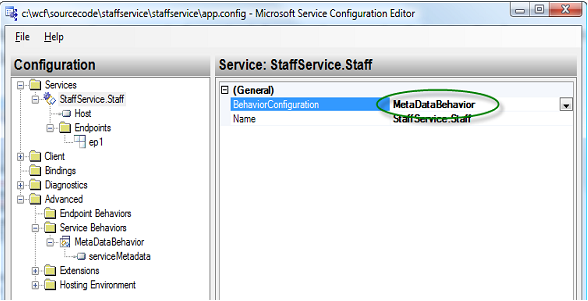
3. Chọn nút serviceMetadata ở bên trái và nhấp vào giá trị HttpGetEnabled thành true là HttpGetUrl là <http://localhost:8000/StaffService>



*Figure 23 Đặt cấu hình cho serviceMetadata*

4. Giờ bạn chọn nút StaffService.Staff và đặt Behavior Configuration với giá trị là

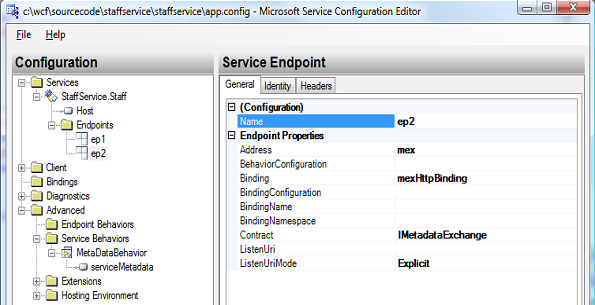
“MetaDataBehavior”



*Figure 24 Đặt BehaviorConfiguration cho dịch vụ*

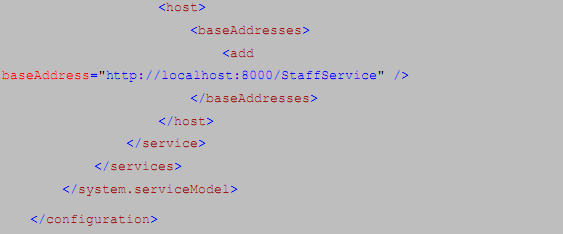
5. Cuối cùng là tạo điểm cuối để các ứng dụng có thể kết nối tới dịch vụ để lấy các thông tin về dịch vụ. Bấm chuột phải vào Endpoints vào chọn New Service Endpoint sau đó đặt cấu hình như sau:

* Name: ep2
* Address: mex
* Binding: mexHttpBinding
* Contract: ImetaDataExchange

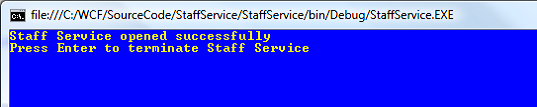


*Figure 25 Cấu hình cho điểm cuối*

6. Bấm menu Save bạn sẽ có tệp cấu hình như sau:

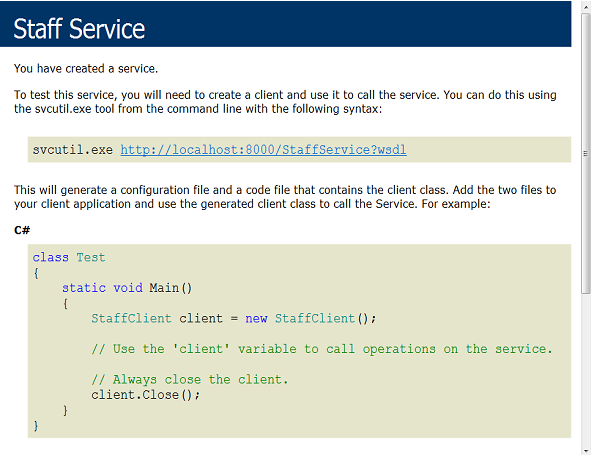


7. Giờ bạn có thể chạy dịch vụ. Sẽ không có lỗi gì xảy ra và bạn sẽ có màn hình như sau:



*Figure 26 Thực hiện dịch vụ*

8. Nếu sử dụng Internet Explorer truy xuất tới địa chỉ <http://localhost:8000/StaffService> bạn sẽ thấy màn hình sau:

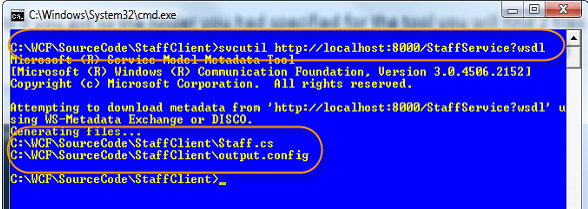


*Figure 27 Sử dụng IE để xem dịch vụ*

* + - 1. Tạo ứng dụng phía client

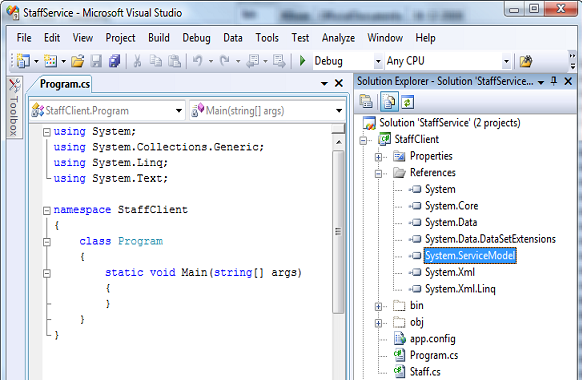
Phần này hướng dẫn các bạn tạo ứng dụng phía client để truy xuất các hàm do dịch vụ trên cung cấp. Điểm hay trong WCF là bạn chỉ cần sử dụng một dòng mã nguồn để gọi đến dịch vụ StaffService, chỉ một dòng mà thôi, những thứ khác được tạo ra tự động sử dụng công cụ svcutil.exe.

1. Mở console dòng lệnh và chuyển tới thư mục bạn muốn tạo các tệp, gõ vào lệnh sau và bấm Enter: Svcutil.exe <http://localhost:8000/StaffService?wsdl> Bạn sẽ thấy kết quả sau:



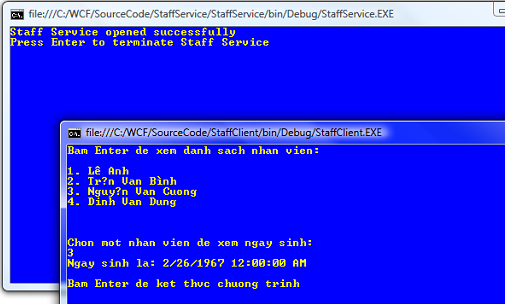
*Figure 28 Kết quả tạo ra khai báo dịch vụ*

2. Tạo ra một console project mới đặt tên là StaffClient và thêm tham chiếu tới System.ServiceModel. Thêm vào tệp vừa tạo ra là Staff.cs, đồng thời đổi tên tệp cấu hình thành app.config và thêm vào project. Kết quả như sau:



*Figure 29 Tạo client*

3. Sau khi biên dịch và chạy chương trình bạn được kết quả sau:

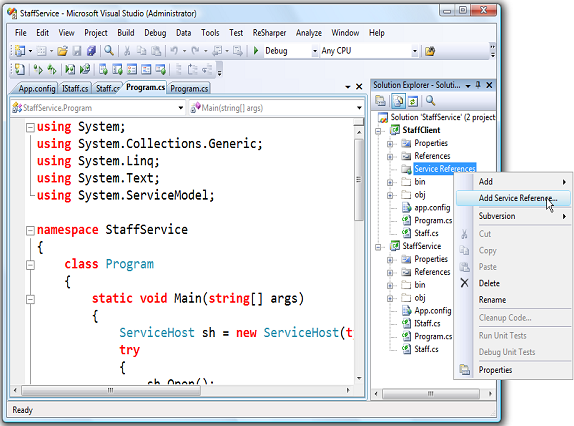


*Figure 30 Kết quả chạy chương trình*

* + 1. Cách khác để tạo tham chiếu ở client

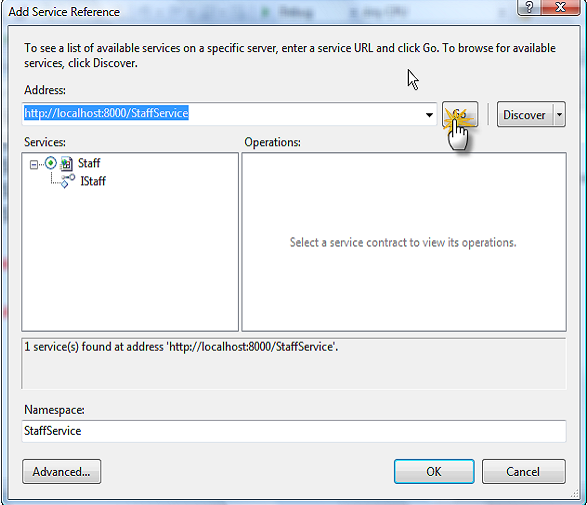
Ngoài cách thực hiện bằng dòng lệnh như ở trên, ta hoàn toàn có thể sử dụng sự giúp đỡ của công cụ Visual Studio để tạo ra tham chiếu dịch vụ phía client.Cách này thực hiện rất nhanh và rất tiện lợi. Cách làm như sau:

**Bước 1**. Chọn chuột phải vào project cần thêm tham chiếu, ở đây là StaffClient, sau đó chọn menu là Add Service Reference như hình sau:



*Figure 31 Thêm tham chiếu dịch vụ*

Bước 2.Gõ vào địa chỉ của dịch vụ cần thêm, sau đó bấm nút Go, phía dưới đặt tên cho tham chiếu dịch vụ là StaffService như hình dưới.



*Figure 32 Cấu hình các tham số cho tham chiếu dịch vụ*

Sau bước này bạn hoàn toàn có thể sử dụng lớp StaffClient để cài đặt ứng dụng phía client.

# THIẾT KẾ CHƯƠNG TRÌNH

1. Mô tả hệ thống

Nhằm lưu lại toàn bộ thông tin về quá trình điều trị của bệnh nhân:bệnh gì,triệu chứng,kết quả xét nghiệm,đơn thuốc,chống chỉ định...Bệnh nhân đi khám chỉ cần tra cứu bệnh án, bác sĩ có thể biết tiền sử bệnh tiến triển như thế nào,có cần thay đổi đơn thuốc hay không.

Việc lưu trữ hồ sơ bệnh án của bệnh nhân là rất quan trọng vì có nhiều bệnh nhân dị ứng thuốc,tiểu đường,hen,phế quản...Nếu bác sĩ không biết có thể cho đơn thuốc nhằm,gây nguy hiểm đến tính mạng.

Hệ thống quản lý bệnh nhân có nhiệm vụ: Quản lý bệnh nhân đã và đang điều trị tại bệnh viện cũng như các quá trình khám chữa bệnh của bệnh nhân, quản lý việc thanh toán viện phí của bệnh nhân với bệnh viện.

Chương trình phải đáp ứng được:

* + Cập nhật và quản lý các thông tin của bệnh nhân.
  + Xem thông tin về bệnh nhân (bệnh, tình trạng diễn tiến bệnh…)
  + Cho phép tìm kiếm theo một số tiêu chí riêng.
  + In ấn phiếu nhập viện, xuất viện, phiếu thanh toán tiền (tạm thu, thanh toán…)
  + Thống kê, tìm kiếm bệnh nhân theo từng thời gian.

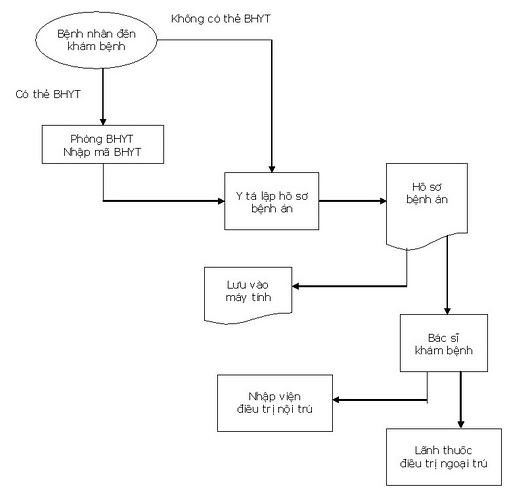
Khi một bệnh nhân nhập viện lần đầu tiên, bệnh viện lưu những thông tin sau: Họ, tên, giới tính, ngày sinh, địa chỉ, dân tộc, nghề nghiệp, thân nhân.

Bệnh nhân được chia làm hai loại: Loại có bảo hiểm y tế và không có bảo hiểm y tế. Nếu bệnh nhân có bảo hiểm y tế thì quản lý: Số thẻ bảo hiểm y tế, thời gian hiệu lực, phần trăm bảo hiểm, nơi khám bệnh ban đầu. Nếu bệnh nhân có thẻ bảo hiểm y tế là công nhân viên của một tổ chức, cơ quan nào đó thì quản lý thêm: Tên, địa chỉ, điện thoại, fax cuả cơ quan công tác.

Địa chỉ bệnh nhân và cơ quan quản lý: Số nhà, đường (thôn, ấp), xã (phường), quận (huyện), tỉnh (thành phố).

**Quy trình khám chữa bệnh:**

Bệnh nhân đến bệnh viện để khám chữa bệnh theo quy trình sau:



- Bước 1: Bệnh nhân đến nơi tiếp nhận bệnh nhân (nơi cấp số) để xác lập việc khám bệnh và được chỉ định một vị trí khám bệnh.

- Bước 2: Bệnh nhân được một bác sĩ khám bệnh.

- Bước 3: Sau khi khám xong bệnh nhân thuộc một trong hai loại: Điều trị tại nhà hay nhập viện.

* Bước 3.1: Nếu bệnh nhân điều trị tại nhà thì bác sĩ cho một toa thuốc trên đó ghi đầy đủ tên thuốc, số lượng và cách dùng. Nếu bệnh nhân có thẻ bảo hiểm y tế thì đến nơi cấp thuốc bảo hiểm y tế để nhận thuốc và trả một phần giá thuốc theo quy định phần trăm trên thẻ bảo hiểm. Ngược lại, bệnh nhân phải trả tất cả chi phí khám bệnh và tự mua thuốc.
* Bước 3.2: Nếu bệnh nhân phải nhập viện thì bác sĩ khám bệnh cho nhập viện, trên đó có ghi đầy đủ thông tin về bệnh nhân, căn bệnh dự đoán và đưa đến khoa điều trị.

- Bước 4: Tại khoa điều trị, bệnh nhân sẽ được chuẩn đoán và lập một bệnh án chi tiết. Trên bệnh án ghi đầy đủ thông tin về bệnh nhân và căn bệnh mà bệnh mắc phải. Quá trình điều trị bệnh nhân được thể hiện đầy đủ trên bệnh án. Trong một khoảng thời gian quy định tuỳ theo bệnh nhân, bệnh nhân được một bác sĩ khám, cho một toa thuốc. Trên toa thuốc ghi tên thuốc, số lượng và cách dùng.

Trong quá trình điều trị, bệnh nhân có thể sử dụng các dịch vụ hỗ trợ khám bệnh như: Xét nghiệm, X\_quang, siêu âm … Việc sử dụng cũng theo chỉ định của bác sĩ khám chữa bệnh. Mỗi dịch vụ có giá tiền riêng.

Khi bệnh nhân điều trị có thể lựa chọn loại phòng nằm điều trị với giá dịch vụ riêng cho mỗi loại phòng.

- Bước 5: Thanh toán viện phí: Trong quá trình điều trị, cứ 3 ngày, bệnh viện yêu cầu bệnh nhân thanh toán viện phí một lần (bằng cách bệnh nhân sẽ đóng một số tiền tạm ứng tuy theo khả năng của mình), sau khi xuất viện nhân viên sẽ căn cứ vào số tiền tạm ứng trên sẽ tính để biết được bệnh nhân đã thanh toán đầy đủ tiền viện phí chưa. Nếu bệnh nhân có thẻ bảo hiểm y tế thì chỉ đóng phần trăm viện phí theo bảo hiểm bao gồm tiền thuốc và các dịch vụ hỗ trợ chữa trị và khám bệnh. Riêng tiền phòng, nếu có thẻ bảo hiểm y tế, bệnh nhân cũng chỉ được bảo hiểm trên loại phòng rẻ nhất, nếu bệnh nhân nằm điều trị ở loại phòng có giá tiền cao thì tự trả phần chênh lệch.Khi xuất viện, bệnh nhân thanh toán toàn bộ số viện phí còn lại.

Trong quá trình điều trị, nếu có bệnh nhân trốn viện, không thanh toán viện phí.Bệnh viện lưu lại tất cả các thông tin về bệnh nhân trốn viện.Nếu sau khi chữa trị 3 ngày mà bệnh nhân không đóng tạm ứng tiền viện phí (hay tiền tạm ứng viện phí trước đó đã hết), thì khoa (phòng) nơi điều trị bệnh nhân trình ban lãnh đạo biết để xem xét giải quyết.

* 1. Đối tượng sử dụng hệ thống

**Người sử dụng hệ thống:** có nhiệm vụ quản lý hệ thống, quản lý các thông tin hành chính cũng như các hồ sơ bệnh án của bệnh nhân (tìm, thêm, xóa, sữa), ngoài ra còn thực hiện công tác thống kê.

* 1. Yêu cầu chức năng
* **Tiếp nhận bệnh nhân:**
* **Thông tin hành chính:** Khi bệnh nhân đến bệnh viện thì phải khai báo các thông tin cá nhân cho bác sĩ ghi vào sổ theo dõi bệnh nhân.
* **Khám nhập viện:** Bác sĩ sẽ khám và chẩn đoán bệnh ban đầu của bệnh nhân, nếu bệnh nặng bệnh nhân cần nhập viện điều trị thì sẽ yêu cầu bệnh nhân nhập viện, nếu bệnh nhẹ thì cho bệnh nhân điều trị ngoại trú.
* **Cấp số:** Bệnh nhân được đưa tới khoa điều trị.
* **Khám chữa bệnh**
* **Cung cấp dịch vụ**: Bệnh nhân có quyền yêu cầu các dịch vụ khám chữa bệnh chất lượng cao hoặc theo ý riêng của bệnh nhân sau khi tham khảo ý kiến bác sĩ hoặc theo sự chỉ định của bác sĩ.
* **Điều trị:** Trong quá trình điều trị, bệnh án của bệnh nhân sẽ được cập nhật thường xuyên.
* **Viện phí**
* **Yêu cầu thanh toán:** Khi bệnh nhân đã khỏi bệnh hoặc chuyển viện thì bệnh viện yêu cầu bệnh nhân thanh toán tiền viện phí.
* **Bảo hiểm y tế**: Nếu bệnh nhân có thẻ bao hiểm y tế hoặc dưới 6 tuổi thì đến Bảo hiểm y tế để được miễn giảm theo quy định.
* **Thanh toán**: Bệnh nhân không có thẻ bảo hiểm y tế sẽ phải thanh toán trực tiếp viện phí với bệnh viện.
* **Thống kê**: Lãnh đạo sẽ yêu cầu thống kê theo các tiêu chí riêng như thống kê bệnh nhân, theo khoa, theo phường, xã...
* **Tìm kiếm**: Lãnh đạo và bệnh nhân (người nhà bệnh nhân) sẽ được yêu cầu tìm kiếm bệnh nhân theo tiêu chí riêng như tìm theo mã bệnh nhân, tìm theo tên...
  1. Yêu cầu phi chức năng
* Dễ sử dụng
* Giao diện thân thiện
* Ngôn ngữ là tiếng Việt
* Tiện lợi cho người sử dụng
* Bố cục trình bày hợp lý
* Tốc độ truy cập nhanh

1. Phân Tích – Thiết Kế Theo Chức Năng
   1. Use Case Diagram

* **ACTOR**

Bác Sĩ

* **USE CASE**
* Bác Sĩ Đăng Nhập
* Quản Lý Bệnh Nhân
  + Tìm Kiếm Bệnh Nhân
  + Thêm Bệnh Nhân
  + Xóa Bệnh Nhân
  + Sữa Bệnh Nhân
* Quản Lý Bệnh Án
  + Tìm Kiếm Bệnh Án
  + Thêm Bệnh Án
  + Xóa Bệnh Án
  + Sữa Bệnh Án
* Thống Kê Bệnh Nhân
* Thống Kê Bệnh Án
  1. Xác định Use Case Tổng Quát

****

* 1. Biểu đồ phân rã chức năng của hệ thống



* + 1. Biểu đồ luồng dữ liệu mức khung cảnh



* + 1. Biểu đồ luồng dữ liệu mức đỉnh



* + 1. Sơ đồ phân rã chức năng quản lý hồ sơ bệnh án



* + 1. Sơ đồ phân rã chức năng tìm kiếm



* + 1. Sơ đồ phân rã chức năng báo cáo thống kê



* 1. Đặc tả Use Case
     1. **Bác Sĩ Đăng Nhập**

Để sử dụng tất cả chức năng về quản lý bệnh nhân, bệnh án, bác sĩ cần đăng nhập tài khoản của mình. Bác sĩ phải nhập chính xác username và password do bệnh viện cung cấp.

**Operationcontract:**

* Check(): thực hiện chức năng kiểm tra username và password bác sĩ nhập vào có đúng với cơ sở dữ liệu không. Nếu đúng thì trả về client là đang nhập thành công, sai thì yêu cầu nhập lại username hoặc password.
  + 1. **Tìm kiếm hồ sơ bệnh án theo loại giám định**

Khi chọn tiêu chí tìm kiếm hồ sơ bệnh án theo loại giám định, service sẽ liệt kê danh sách bệnh án theo loại giám định mình nhập vào, hiển thị tình trạng của bệnh án, những phuơng pháp điều trị và những loại thuốc đặc trị cho loại giám định đó.

**Operationcontract:**

* + 1. **Tìm kiếm hồ sơ bệnh án theo cơ quan giới thiệu**

Khi chọn tiêu chí tìm kiếm theo cơ quan giới thiệu, service sẽ liệt kê các bệnh án có tên là nơi nhập viện đầu tiên, chuẩn đoán, tình trạng bệnh nhân, đơn thuốc, các phuơng pháp điều trị khi còn ở đó.

**Operationcontract:**

* + 1. **Thống kê bệnh nhân theo dân tộc**

Chọn chức năng thống kê bệnh nhân theo dân tộc, service sẽ trả về danh sách các bệnh nhân thuộc dân tộc đó và có 2 tiêu chí là sắp xếp theo loại bệnh ở những khu vực nào và sắp xếp theo thành phố, sau đó sẽ tính toán số phần tram các loại bệnh theo khu vực hay số bệnh nhân theo thành phố.

* + 1. **Thống kê theo loại giám định**

Chọn chức năng thống kê theo loại giám định, service trả về bảng thống kê số luợng bệnh án theo tiêu chí mình chọn là theo tháng, năm, quý của loại giám định đó, kèm theo là các loại thuốc, phuơng pháp điều trị có kết quả tốt nhất.

* + 1. **Quản Lý Bệnh Nhân**
* **Tìm**

|  |  |
| --- | --- |
| Tên use case | Tìm Kiếm Bệnh Nhân |
| Tác nhân chính | Bác sĩ |
| Mức | 2 |
| Tiền điều kiện | Bác sĩ phải đăng nhập vào hệ thống |
| Kích hoạt | Bác sĩ chọn chức năng tìm kiếm trong menu |
| Đảm bảo thành công | Hệ thống hiển thị thông tin chi tiết của quản lý bệnh nhân |
| Chuỗi sự kiện chính | 1. Bác sĩ chọn chức năng tìm kiếm quản lý trong menu  2. Hệ thống hiển thị form tìm kiếm và yêu cầu bác sĩ nhập từ khóa, chọn các tiêu chí cần tìm  3. Bác sĩ nhập từ khóa, chọn các tiêu chí và nhấn nút “Tìm kiếm”  4. Hệ thống thực hiện việc tìm kiếm dựa trên từ khóa và các tiêu chí  5. Hệ thống hiển thị danh sách các thông tin tìm được  6. Bác sĩ chọn một đơn thuốc để xem thông tin chi tiết  7. Hệ thống hiển thị thông tin chi tiết. |
| Ngoại lệ | 4. Hệ thống thông báo không có thông tin cần tìm, bác sĩ có muốn tìm kiếm nữa không?  1. Bác sĩ nhấn nút “Có”, quay lại bước 2  2. Bác sĩ nhấn nút “Không”, hệ thống thoát khỏi chức năng tìm kiếm.  5. Bác sĩ nhấn nút “Thoát”, hệ thống thoát khỏi chức năng tìm kiếm.  6. Bác sĩ nhấn nút “Đóng” phía dưới thông tin chi tiết, hệ thống đóng thông tin chi tiết, quay lại danh sách các thông tin tìm được |

* **Thêm**

|  |  |
| --- | --- |
| Tên use case | Thêm Bệnh Nhân |
| Tác nhân chính | Bác Sĩ |
| Mức | 2 |
| Tiền điều kiện | Bác sĩ phải đăng nhập vào hệ thống |
| Kích hoạt | Bác sĩ chọn chức năng trong menu |
| Đảm bảo thành công | Thông tin được thêm vào CSDL |
| Chuỗi sự kiện chính | 1. Bác sĩ chọn chức năng Thêm trong menu 2. Hệ thống hiển thị form Quản lý Bệnh Nhân và yêu cầu Bác sĩ nhập thông tin của bệnh nhân khi nhập viện. 3. Bác sĩ nhập thông tin và nhấn nút “OK” 4. Hệ thống kiểm tra thông tin do bác sĩ nhập vào và xác nhận thông tin nhập vào là hợp lệ. 5. Hệ thống thêm thông tin vào CSDL. 6. Hệ thống thông báo đã thêm thành công. |
| Ngoại lệ | 4. Hệ thống thông báo thông tin không hợp lệ, bác sĩ có nhập nữa không?   1. Bác sĩ nhấn nút “có”, quay lại bước 2. 2. Bác sĩ nhấn nút “Không”, hệ thống thoát khỏi chức năng nhập thông tin. |

* **Xóa**

|  |  |
| --- | --- |
| Tên use case | Xóa Bệnh Nhân |
| Tác nhân chính | Bác sĩ |
| Mức | 3 |
| Tiền điều kiện | Bác sĩ phải tìm kiếm thông tin cần xóa |
| Kích hoạt | Bác sĩ nhấn nút “Xóa” phía dưới thông tin chi tiết của quản lý bệnh nhân |
| Đảm bảo thành công | Thông tin được xóa khỏi CSDL |
| Chuỗi sự kiện chính | 1. Bác sĩ nhấn nút “Xóa” phía dưới thông tin chi tiết của quản lý bệnh nhân.  2. Hệ thống hỏi bác sĩ có muốn xóa thông tin này không?  3. Bác sĩ nhấn nút “Có”  4. Hệ thống xóa thông tin trong CSDL  5. Hệ thống thông báo đã xóa thông tin thành công  6. Hệ thống quay lại danh sách tìm kiếm |
| Ngoại lệ | 3. Bác sĩ nhấn nút “Không”, hệ thống quay lại danh sách quản lý bệnh nhân |

* **Sửa**

|  |  |
| --- | --- |
| Tên Use case | Sửa Bệnh Nhân |
| Tác nhân chính | Bác sĩ |
| Mức | 2 |
| Tiền điều kiện | Bác sĩ phải tìm thông tin chi tiết |
| Kích hoạt | Bác sĩ nhấn nút “Sửa” phía dưới thông tin chi tiết của quản lý bệnh nhân. |
| Đảm bảo thành công | Thông tin được cập nhật vào CSDL |
| Chuỗi sự kiện chính | 1. Bác sĩ nhấn nút “Sửa” phía dưới thông tin chi tiết. 2. Hệ thống hiển thị form và yêu cầu bác sĩ nhập thông tin cần cập nhật. 3. Bác sĩ nhập thông tin và nhấn nút “OK”. 4. Hệ thống kiểm tra thông tin được cập nhật và xác nhận thông tin là hợp lệ. 5. Hệ thống cập nhật vào CSDL. 6. Hệ thống thông báo đã cập nhật thành công. |
| Ngoại lệ | 4. Hệ thống thông báo thông tin không hợp lệ, bác sĩ có cập nhật nữa không?   1. Bác sĩ nhấn nút “có”, quay lại bước 2. 2. Bác sĩ nhấn nút “không”, hệ thống thoát khỏi chức năng cập nhật . |

* + 1. **Quản Lý Bệnh Án**
* **Tìm**

|  |  |
| --- | --- |
| Tên use case | Tìm Bệnh Án |
| Tác nhân chính | Bác sĩ |
| Mức | 3 |
| Tiền điều kiện | Bác sĩ phải đăng nhập vào hệ thống |
| Kích hoạt | Bác sĩ chọn chức năng tìm kiếm trong menu |
| Đảm bảo thành công | Hệ thống hiển thị thông tin chi tiết của Bệnh Án |
| Chuỗi sự kiện chính | 1. Bác sĩ chọn chức năng tìm kiếm trong menu  2. Hệ thống hiển thị form tìm kiếm và yêu cầu bác sĩ nhập từ khóa, chọn các tiêu chí thông tin cần tìm.  3. Bác sĩ nhập từ khóa, chọn các tiêu chí và nhấn nút “Tìm kiếm”  4. Hệ thống thực hiện việc tìm kiếm dựa trên từ khóa và các tiêu chí  5. Hệ thống hiển thị danh sách các thông tin tìm được  6. Bác sĩ chọn một thông tin để xem thông tin chi tiết  7. Hệ thống hiển thị thông tin chi tiết của Bệnh Án. |
| Ngoại lệ | 4. Hệ thống thông báo không có thông tin cần tìm, bác sĩ có tìm kiếm thông tin nữa không?  1. Bác sĩ nhấn nút “Có”, quay lại bước 2  2. Bác sĩ nhấn nút “Không”, hệ thống thoát khỏi chức năng tìm kiếm.  6. Bác sĩ nhấn nút “Thoát”, hệ thống thoát khỏi chức năng tìm kiếm.  7. Bác sĩ nhấn nút “Đóng” phía dưới thông tin chi tiết, hệ thống đóng thông tin chi tiết, quay lại danh sách các thông tin tìm được. |

* **Thêm**

|  |  |
| --- | --- |
| Tên use case | Thêm Bệnh Án |
| Tác nhân chính | Bác Sĩ |
| Mức | 2 |
| Tiền điều kiện | Bác sĩ phải đăng nhập vào hệ thống |
| Kích hoạt | Bác sĩ chọn chức năng Thêm trong menu |
| Đảm bảo thành công | Thông tin nhập viện được thêm vào CSDL |
| Chuỗi sự kiện chính | 1. Bác sĩ chọn chức năng ”Thêm” trong menu 2. Hệ thống hiển thị form và yêu cầu Bác sĩ nhập thông tin bệnh án của bệnh nhân khi nhập viện. 3. Bác sĩ nhập thông tin bệnh án và nhấn nút “OK” 4. Hệ thống kiểm tra thông tin bệnh án do bác sĩ nhập vào và xác nhận thông tin bệnh án nhập vào là hợp lệ. 5. Hệ thống thêm bệnh án vào CSDL. 6. Hệ thống thông báo đã thêm bệnh án thành công. |
| Ngoại lệ | 4. Hệ thống thông báo thông tin bệnh án không hợp lệ, bác sĩ có nhập nữa không?   1. Bác sĩ nhấn nút “có”, quay lại bước 2. 2. Bác sĩ nhấn nút “Không”, hệ thống thoát khỏi chức năng nhập thông tin bệnh án. |

* **Xóa**

|  |  |
| --- | --- |
| Tên use case | Xóa Bệnh Án |
| Tác nhân chính | Bác sĩ |
| Mức | 3 |
| Tiền điều kiện | Bác sĩ phải tìm kiếm thông tin cần xóa |
| Kích hoạt | Bác sĩ nhấn nút “Xóa” phía dưới thông tin chi tiết của Bệnh Án |
| Đảm bảo thành công | Thông tin được xóa khỏi CSDL |
| Chuỗi sự kiện chính | 1. Bác sĩ nhấn nút “Xóa” phía dưới thông tin chi tiết Bệnh Án  2. Hệ thống hỏi bác sĩ có muốn xóa thông tin này không?  3. Bác sĩ nhấn nút “Có”  4. Hệ thống xóa thông tin trong CSDL  5. Hệ thống thông báo đã xóa thông tin thành công  6. Hệ thống quay lại danh sách tìm kiếm |
| Ngoại lệ | 3. Bác sĩ nhấn nút “Không”, hệ thống quay lại danh sách quản lý bệnh nhân |

* **Sửa**

|  |  |
| --- | --- |
| Tên Use case | Sửa Bệnh Án |
| Tác nhân chính | Bác sĩ |
| Mức | 2 |
| Tiền điều kiện | Bác sĩ phải tìm thông tin chi tiết |
| Kích hoạt | Bác sĩ nhấn nút “Sửa” phía dưới thông tin chi tiết của Bệnh án. |
| Đảm bảo thành công | Thông tin được cập nhật vào CSDL |
| Chuỗi sự kiện chính | 1. Bác sĩ nhấn nút “Sửa” phía dưới thông tin chi tiết. 2. Hệ thống hiển thị form và yêu cầu bác sĩ nhập thông tin cần cập nhật. 3. Bác sĩ nhập thông tin và nhấn nút “OK”. 4. Hệ thống kiểm tra thông tin được cập nhật và xác nhận thông tin là hợp lệ. 5. Hệ thống cập nhật vào CSDL. 6. Hệ thống thông báo đã cập nhật thành công. |
| Ngoại lệ | 4. Hệ thống thông báo thông tin không hợp lệ, bác sĩ có cập nhật nữa không?   1. Bác sĩ nhấn nút “có”, quay lại bước 2. 2. Bác sĩ nhấn nút “không”, hệ thống thoát khỏi chức năng cập nhật. |

1. Thiết Kế Cơ Sở Dữ Liệu
   1. Các Thực Thể

Để đưa ra mô hình các thực thểdựa vào các mẫu sau:

Hồ sơ bệnh án: là nơi lưu giữ tất cả các thông tin về bệnh nhân từ lúc vào viện điều trị đến lúc ra viện.

Các phiếu, biểu mẫu của bệnh viện.

* Hồ sơ bệnh nhân: Mã bệnh nhân, họ tên, giới tính, địa chỉ, ngày sinh, CMND, SĐT, dân tộc, nghề nghiệp, đối tượng, nhóm máu, dị ứng thuốc, tình trạng BN.
* Hồ sơ bệnh án: Mã hồ sơ bệnh án, mã bệnh nhân, tên bệnh, chuẩn đoán sơ bộ, chuẩn đoán chính, loại xét nghiệm, ngày khám, diễn biến bệnh, y lệnh, ngày lập bệnh án, ngày xét nghiệm, kết quả, mã bệnh nhân, mã bác sĩ, mã đơn thuốc.
* Chi tiết đơn thuốc: Mã đơn thuốc, mã dược phẩm, số lượng thuốc, thành tiền, ghi chú.
* Khoa: mã khoa, tên khoa.
* Bác sĩ: mã bác sĩ, họ tên, giới tính,ngày sinh,CMND, địa chỉ, số điện thoại,email, trình độ chuyên môn, mã khoa.
* Giường điều trị: mã bệnh nhân, mã giường, ngày nhận.
* Dược phẩm: mã dược phẩm, tên dược phẩm, đơn giá, nước sản xuất,hạn sử dụng, ngày sản xuất.
* Phòng bệnh: mã phòng, tên phòng, loại phòng, số giường, tình trạng.
* Giường bệnh: mã giường, loại giường, tình trạng, mã phòng.
  1. Thiết Kế Cơ Sở Dữ Liệu
  2. **Đặc Tả Các Bảng**

**Bác sĩ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| Mabs | nchar | 10 | Primary key | Mã bác sĩ |
| Tenbs | nvarchar | 50 | Not Null | Họ tên |
| Gioitinh | nchar | 5 | Not Null | Giới tính |
| Ngaysinh | Datetime |  | Not Null | Ngày sinh |
| CMND | Nvarchar | 50 | Not Null | Chứng minh nhân dân |
| Diachi | nvarchar | 50 | Null | Địa chỉ |
| Sdt | Nchar | 15 | Null | Số điện thoại |
| Email | Nvarchar | 50 | Null | Địa chỉ email |
| Trinhdochuyenmon | Nvarchar | 50 | Null | Trình độ chuyên môn |
| Makhoa | Nchar | 10 | Not null | Mã khoa |

**Dược phẩm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| Maduocpham | Nchar | 10 | Primary key | Mã dược phẩm |
| Tenduocpham | Nvarchar | 50 | Not null | Tên dược phẩm |
| Dongia | Float |  | Null | Đơn giá |
| Nuocsx | Nvarchar | 50 | Null | Nước sản xuất |
| Hansd | Datetime |  | Null | Hạn sử dụng |
| Ngaysx | Datetime |  | Null | Ngày sản xuất |

**Khoa**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| makhoa | Nchar | 10 | Primary key | Mã khoa |
| tenkhoa | nvarchar | 150 | Not Null | Tên khoa |

**Phòng bệnh**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| Maphong | Nchar | 10 | Primary key | Mã phòng |
| tenphong | nvarchar | 50 | Not Null | Tên phòng |
| Loaiphong | Nvarchar | 50 | Not null | Loại phòng |

**Hồ sơ bệnh nhân**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| MaBN | Nchar | 10 | Primary key | Mã Bênh Nhân |
| Dantoc | Nchar | 10 | Not null | Dân tộc |
| CMND | Nvarchar | 50 | Not null | Chứng minh nhân dân |
| Nghenghiep | Nvarchar | 50 | Null | Nghề nghiệp |
| Doituong | nchar | 10 | Null | Đối tượng(BHYT or đóng phí) |
| Hoten | nvarchar | 50 | Not Null | Họ tên |
| Ngaysinh | Date/Time |  | Not Null | Ngày sinh |
| Gioitinh | Nchar | 5 | Not Null | Giới tính |
| Diachi | nvarchar | 50 | Null | Địa chỉ |
| Email | Nvarchar | 50 | Null | Địab chỉ email |
| Nhommau | Nchar | 5 | Null | Nhóm máu |
| Ghichu | nvarchar | 500 | Null | Ghi chú(dị ứng thuốc) |
| Tinhtrangbn | Nvarchar | 50 | Null | Tình trạng bệnh nhân |

**Giường bệnh**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| Magiuong | Nchar | 10 | Primary key | Mã giường |
| Loaigiuong | nvarchar | 50 | Not Null | Loại giường |
| Maphong | Nchar | 10 | Not Null | Mã phòng bệnh |
| Tinhtrang | Nchar | 10 | Null | Tình trạng |

**Giường điều trị**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| Mabn | Nchar | 10 | Primary key | Mã bệnh nhân |
| Magiuong | Nchar | 10 | Primary key | Mã giường bệnh |
| Ngaynhan | Datetime |  | Null | Ngày nhận |

**Bệnh án**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| MaHSBA | Nchar | 10 | Primary key | Mã hồ sơ bệnh án |
| Tenba | Nvarchar | 500 | Not null | Tên bệnh án |
| Ngaylapba | Nchar | 10 | Null | Ngày lập bệnh án |
| Ngaykham | Nchar | 10 | Null | Ngày hẹn tái khám |
| Mabn | Nchar | 10 | Not null | Mã bệnh nhân |
| Dienbienbenh | Nvarchar | 500 | Null | Diễn biến bệnh |
| Y lệnh | Nvarchar | 500 | Null | Y lệnh |
| chuandoansobo | nvarchar | 500 | null | Chuẩn đoán sơ bộ |
| chuandoanchinh | nvarchar | 500 | null | Chuẩn đoán chính |
| Loaixn | nchar | 10 | Not null | Loại xét nghiệm |
| ketqua | Nvarchar | 500 | Not null | Kết quả |
| Ngayxn | Nchar | 10 | Null | Ngày xét nghiệm |
| Mabs | Nchar | 10 | Not null | Mã bác sĩ |
| Madonthuoc | Nchar | 10 | Not null | Mã đơn thuốc |

**Chi tiết đơn thuốc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| madonthuoc | nchar | 10 | Primary key | Mã đơn thuốc |
| Maduocpham | nchar | 10 | Primary key | Mã dược phẩm |
| Thanhtien | Float |  | Null | Thành tiền |
| Ghichu | Nvarchar | 500 | Null | Ghi chú |
| soluong | Int |  | Not Null | Số lượng |

* 1. **Mô Hình Quan Hệ**

****