

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG DẠI HỌC HOA SEN

KHOA KHOA HOC VÀ CÔNG NGHỆ

BÁO CÁO ĐỒ ÁN : CÔNG NGHỆ VÀ DỊCH VỤ WEB  
ĐỀ TÀI:

**WEBSITE CUNG CẤP THÔNG TIN NHÀ HÀNG QUÁN NƯỚC**

**GIẢNG VIÊN HƯỚNG DẪN**: Võ Hoàng Hải

**SINH VIÊN THỰC HIỆN**:

🖙 081668 NGUYỄN QUANG KHẢI.

🖙 091390 NGUYỄN THẾ ĐẠT.

🖙 080196 NGÔ TRƯỜNG LUÂN.

**LỚP MÔN HỌC**: TIN252DV01

*THÀNH PHỐ HỒ CHÍ MINH  
Năm 2011*

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Ngày tháng năm

Kí tên

**LỜI MỞ ĐẦU**

Trong xã hội ngày nay, việc giải trí ăn uống là một phần của cuộc sống. Nhưng trên thực tế có rất nhiều nhà hàng, cà phê được mở ra chính vì vậy việc xác định một nhà hàng hay quán nước là một điều khó xử. Qua những khó khăn trên nhóm chúng tôi đã xay đựng một website có khả năng cung cấp những thong tin cần thiết. Với mục đích là giúp người dùng có thể chọn lựa nhà hàng cà phê ưng ý nhất. Qua đó giúp cho việc tìm kiếm một cách đễ dàng hơn.

MỤC LỤC

[DANH MỤC HÌNH ẢNH 11](#_Toc311621806)

[NỘI DUNG 12](#_Toc311621807)

[1.TÌM HIỂU VỀ LÝ THUYẾT 12](#_Toc311621808)

[1.1WINDOWNS COMMUNICATION FOUNDATION (WCF) 12](#_Toc311621809)

[1.1.1Windows Communication Foundation là gì? 12](#_Toc311621810)

[1.1.2.Cơ bản về Windows Communication Foundation 13](#_Toc311621811)

[1.1.2.1.Address 14](#_Toc311621812)

[1.1.2.2.Binding 16](#_Toc311621813)

[1.1.2.3.Contract 18](#_Toc311621814)

[1.1.3.Những tính năng chính của Windows Communication Foundation 18](#_Toc311621815)

[1.1.3.1.Transaction 18](#_Toc311621816)

[1.1.3.2.Hosting 19](#_Toc311621817)

[1.1.3.3.Security 20](#_Toc311621818)

[1.1.3.4.Queuing 21](#_Toc311621819)

[1.1.4.Kiến trúc Windows Communication Foundation 21](#_Toc311621820)

[1.1.4.1.Contracts 22](#_Toc311621821)

[1.1.4.2.Service Runtime 22](#_Toc311621822)

[1.1.4.3.Messaging 23](#_Toc311621823)

[1.1.4.4.Activation and Hosting 24](#_Toc311621824)

[1.1.5.Cài Đặt WCF 25](#_Toc311621825)

[1.2.Service Oriented Architecture 34](#_Toc311621826)

[1.2.1.Service 34](#_Toc311621827)

[1.2.2.Các Đặc Điểm Chính Của Service 35](#_Toc311621828)

[1.2.3.Service Oriented Architecture 36](#_Toc311621829)

[1.2.4.Các actor trong SOA 36](#_Toc311621830)

[1.2.5.Ích lợi khi sử dụng SOA 37](#_Toc311621831)

[1.2.6.Thông điệp (message) trong SOA 38](#_Toc311621832)

[1.3.Web Service 39](#_Toc311621833)

[2.THIẾT KẾ CHƯƠNG TRÌNH 41](#_Toc311621834)

[2.1Mô tả ứng dụng 41](#_Toc311621835)

[2.1.1.Ứng dụng service host 41](#_Toc311621836)

[2.1.2.Ứng dụng client 41](#_Toc311621837)

[2.2.Đặc tả các chức năng (operation contracts) cà phê nhà hàng: 42](#_Toc311621838)

[2.2.1.Tìm kiếm theo địa điểm: 42](#_Toc311621839)

[2.2.2.Tìm kiếm theo không gian: 43](#_Toc311621840)

[2.2.3.Tìm kiếm theo mục đích của NH-CF: 44](#_Toc311621841)

[2.2.4.Tìm kiếm theo phong cách món ăn: 45](#_Toc311621842)

[2.2.5.Tìm kiếm theo phong cách phục vụ: 46](#_Toc311621843)

[2.2.6.Tìm kiếm theo các loại món ăn thức uống đặc sắc của NH-CF: 47](#_Toc311621844)

[2.2.7.Tìm kiếm theo nhu cầu khác: 48](#_Toc311621845)

[2.2.8.Tìm kiếm theo giá cả: 49](#_Toc311621846)

[2.2.9.Tìm kiếm theo các dịch vụ: 50](#_Toc311621847)

[2.2.10.Tìm kiếm theo tin tức khuyến mãi: 51](#_Toc311621848)

[2.2.11.Người dùng đăng ký: 52](#_Toc311621849)

[2.2.12.Người dùng đăng nhập: 53](#_Toc311621850)

[2.2.13.Người dung đánh giá: 54](#_Toc311621851)

[2.2.14.Người dùng góp ý: 55](#_Toc311621852)

[2.2.15.Người dùng đăng bài giới thiệu: 56](#_Toc311621853)

[2.2.16.So sánh: 57](#_Toc311621854)

[2.2.17.Sắp xếp: 58](#_Toc311621855)

[2.2.18.Liên hệ NH-CF: 59](#_Toc311621856)

[2.2.19.Reservation: 59](#_Toc311621857)

[3.HIỆN THỰC CHƯƠNG TRÌNH 61](#_Toc311621858)

[3.1.Giao diện chính của chương trình. 61](#_Toc311621859)

[3.2.Form đăng nhập vào hệ thống. 61](#_Toc311621860)

[3.3.Form đăng ký hệ thống. 62](#_Toc311621861)

[3.4.Tìm kiếm thông tin nhà hàng cà phê theo địa điểm. 62](#_Toc311621862)

[3.5.Tìm kiếm thông tin nhà hàng cà phê theo không gian. 63](#_Toc311621863)

[3.6.Tìm kiếm thông tin nhà hàng cà phê theo tên món ăn. 63](#_Toc311621864)

[3.7.Tìm kiếm thông tin nhà hàng cà phê theo nhu cầu. 64](#_Toc311621865)

[3.8.Tìm kiếm thông tin nhà hàng cà phê theo giá. 64](#_Toc311621866)

[64](#_Toc311621867)

[3.9.Tìm kiếm thông tin nhà hàng cà phê theo loại khuyến mãi. 65](#_Toc311621868)

[3.10.Form thể hiện thông tin nhà hàng. 65](#_Toc311621869)

[65](#_Toc311621870)

[3.11.Form thể hiện thông tin thực đơn của nhà hàng. 66](#_Toc311621871)

[66](#_Toc311621872)

[3.12.Form thể hiện thông tin khuyến mãi của nhà hàng. 66](#_Toc311621873)

[66](#_Toc311621874)

[3.13.Form thể hiện thông tin góp ý của nhà hàng và đăng góp ý của người dùng. 67](#_Toc311621875)

[67](#_Toc311621876)

[3.14.Form đặt bàn của người dùng với nhà hàng. 67](#_Toc311621877)

[67](#_Toc311621878)

[3.15.Form đăng thông tin nhà hàng cà phê. 68](#_Toc311621879)

[68](#_Toc311621880)

[3.16.Form đăng thông tin thực đơn nhà hàng cà phê. 68](#_Toc311621881)

[68](#_Toc311621882)

[3.17.Form đăng thông tin khuyến mãi nhà hàng cà phê. 69](#_Toc311621883)

[69](#_Toc311621884)

[3.17.Form đăng thông tin tác giả. 69](#_Toc311621885)

[69](#_Toc311621886)

[4.Tham Khảo. 70](#_Toc311621887)

# 

# DANH MỤC HÌNH ẢNH

*Hình 1: WCF(Windowns Communication Foundation) 6*

*Hình 2: Endpoints. 7*

*Hình 3: Address 8*

*Hình4: WS Addressing. 8*

*Hình 5:WCF supports following Transport. 9*

*Hình 6:Kiến trúc WCF 15*

*Hình 7: SOA actors 30*

*Hình 8: Web service protocol stack 33*

*Hình 10: Web service actors 34*

# NỘI DUNG

# 1.TÌM HIỂU VỀ LÝ THUYẾT

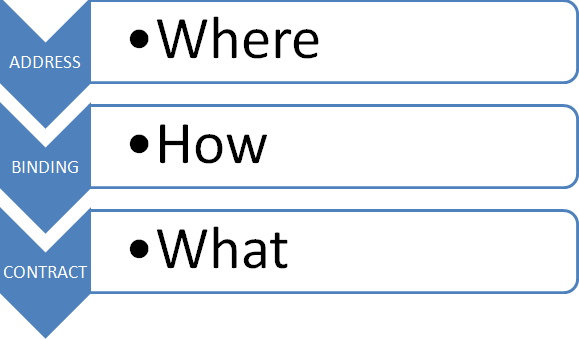
# 1.1WINDOWNS COMMUNICATION FOUNDATION (WCF)

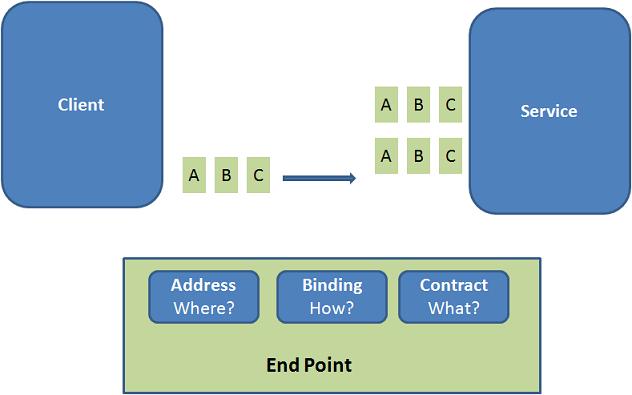
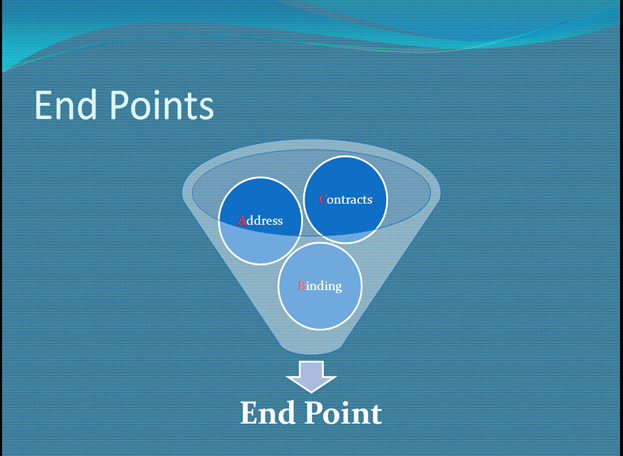
# 1.1.1Windows Communication Foundation là gì?

* Windows Communication Foundation (WCF) là một nền tảng (platform) lập trình do Microsoft cung cấp, một mở rộng của .NET framework cho việc xây dựng, cấu hình và triển khai các dịch vụ phân tán. WCF cung cấp một framework thống nhất cho việc xây dựng các giao dịch bảo mật và đáng tin cậy trên dịch vụ web.
* WCF làm đơn giản hóa việc phát triển các ứng dụng hướng dịch vụ bằng việc kết hợp các công nghệ khác nhau như: Enterprise Services, Messaging, .NET Remoting, Web Service và Web Enhancement (WSE).

*Hình 1:WCF(Windowns Communication Foundation)*

# 1.1.2.Cơ bản về Windows Communication Foundation

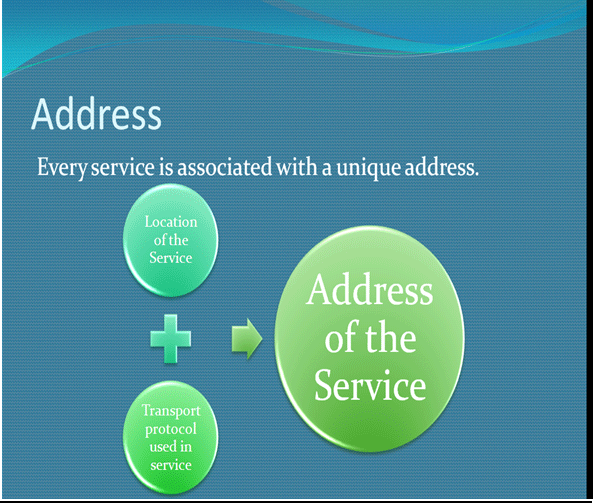
* WCF Service là một chương trình đưa ra một tập hợp các Endpoint. Mỗi Endpoint là một cổng truyền thông ra bên ngoài. Mọi việc truyền thông của WCF được diễn ra thông qua Endpoint. Cấu trúc của Endpoint bao gồm ba thành phần: Address, Binding và Contract.
* Một WCF Service có thể có nhiều Endpoint và khi client giao tiếp với service thì chỉ thông qua 1 Endpoint duy nhất.



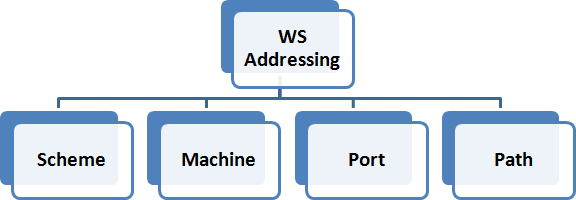
*Hình 2: Endpoints.*

# 1.1.2.1.Address

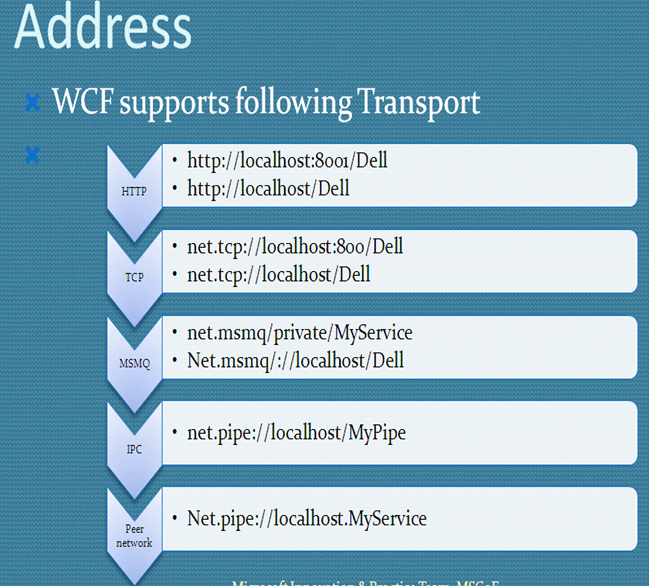
* Address là một URL (Uniform Resource Locator), chỉ định vị trí của dịch vụ và Address nên tuân theo chuẩn Web Service Addressing (WSAddressing,[scheme://<machinename>[:port]/path1/path2](scheme://%3cmachinename%3e[:port]/path1/path2))



*Hình 3: Address*



*Hình4: WS Addressing.*



*Hình 5:WCF supports following Transport.*

* Một số kiểu khác nhau của Address
* HTTP: <http://localhost:9001/Dell>, <http://localhost/Dell>
* TCP: <net.tcp://localhost:800/Dell>, <net.tcp://localhost/Dell>
* MSMQ: <net.msmq/private/MyService>, [Net.msmq/://localhost/Dell](Net.msmq/:/localhost/Dell)
* IPC: <net.pipe://localhost/MyPipe>
* Peer network: <Net.pipe://localhost.MyService>

# 1.1.2.2.Binding

* Binding cho biết một dịch vụ có thể được sử dụng như thế nào. Nó định nghĩa protocol được sử dụng, encoding được sử dụng và loại yêu cầu bảo mật nào được sử dụng như bảo mật SSL hay bảo mật thông điệp SOAP. Bao gồm:
* basisHttpBinding: Đây là một binding tương thích, thường được sử dụng để thay thế cho các dịch vụ web dựa trên ASMX trước đây. Nó hỗ trợ HTTP, HTTPS protocol và MTOM, text encoding.
* wsHttpBinding: Đây là một binding bảo mật và tương thích. Nó sử dụng SOAP trên HTTP, hỗ trợ HTTP, HTTPS protocol, MTOM, text encoding, các tính năng về độ tin cậy, giao dịch và bảo mật. Đây là binding mặc định được cung cấp bởi WCF.
* wsDualHttpBinding: Binding này hỗ trợ mọi đặc tính của wsHttpBinding, được sử dụng chủ yếu trong các Duplex Service Contract. Nó hỗ trợ thông tin liên lạc theo 2 chiều.
* webHttpBinding: Đây là binding bảo mật và tương thích. Nó hỗ trợ HTTP, HTTPS protocol. Tuy nhiên nó không sử dụng định dạng thông điệp SOAP. Binding này chủ yếu được sử dụng đối với các dịch vụ REST.
* wsFederationHttpBinding: Đây là binding bảo mật và tương thích, hỗ trợ liên bảo mật (federated security). Nó hỗ trợ HTTP, HTTPS protocol, và MTOM, text encoding. Binding này hỗ trợ việc liên định danh (federated identity).
* netTCPBinding: Đây là binding bảo mật, chỉ có thể được sử dụng nếu client cũng là một WCF machine. Nó được dùng để gửi thông điệp Binary encoded SOAP từ một máy tính sử dụng WCF này sang một máy tính sử dụng WCF khác. Nó hỗ trợ TCP protocol, các tính năng về độ tin cậy, giao dịch và bảo mật.
* netNamePipeBinding: Đây là binding bảo mật, có thể được dùng trên một máy đơn sử dụng WCF. Binding này gửi các thông điệp Binary encoded SOAP trên các đường ống được đặt tên (named pipe).
* netMSMQBinding: Đây là binding hàng đợi (queued binding), sử dụng thông điệp Binary encoded SOAP. Binding này gửi thông điệp trên MSMQ và việc thông tin ở đây nên thực hiện giữa hai máy tính.
* netPeerTCPBinding: Đây là binding bảo mật, sử dụng TCP protocol trên các máy tính ngang hàng. Việc thông tin ở đây nên được thực hiện giữa hai hay nhiều máy tính.
* msmqIntegrationBinding: Đây là binding tương thích được sử dụng cho các ứng dụng MSMQ dùng COM và C++ API.
* basicHttpContextBinding: Binding này tương tự như basicHttpBinding nhưng với nhiều thuộc tính hơn. Nó hỗ trợ HTTP cookies, cho phép các SOAP header trao đổi context. Binding chủ yếu được sử dụng trong các dịch vụ bền lâu (Durable Service).
* netTCPContextBinding: Binding này tương tự như netTCPBinding nhưng với nhiều thuộc tính hơn. Nó cho phép các SOAP header trao đổi context. Binding này chỉ yếu được sử dụng trong các dịch vụ bền lâu (Durable Service).
* wsHttpContextBinding: Binding này tương tự như wsHttpBinding nhưng với nhiều thuộc tính hơn. Nó cho phép các SOAP header trao đổi context. Binding này chỉ yếu được sử dụng trong các dịch vụ bền lâu (Durable Service).

# 1.1.2.3.Contract

Contract là tập hợp các hoạt động được chỉ ra bởi Endpoint sẽ truyền thông ra bên ngoài. Thông thường tên của Interface sẽ được đề cập trong Contract, vì vậy ứng dụng client sẽ nhận ra được các hoạt động được đưa ra bởi dịch vụ. Mỗi hoạt động là một mẫu trao đổi đơn giản như một chiều hoặc hai chiều và các request, reply. WCF đưa ra một số loại contract sau: Service Contract, Message Contract, Data Contract.

# 1.1.3.Những tính năng chính của Windows Communication Foundation

# 1.1.3.1.Transaction

* Một transaction là một tập hợp hay một nhóm gồm một hay nhiều đơn vị thao tác được thực hiện một cách trọn vẹn. Nói một cách khác, các transaction cung cấp cách gom nhóm các phần công việc riêng rẽ một cách hợp lý và thực hiện chúng như một đơn vị thao tác đơn, hay một transaction.
* Một transaction sẽ chắc chắn tất cả các thao tác sẽ cùng thành công hoặc cùng thất bại. Nếu một trong các thao tác thất bại thì toàn bộ các phần công việc sẽ được hủy bỏ. Lúc này, bất cứ thay đổi nào cũng không được thực hiện. Và việc tháo gỡ các đơn vị công việc đã được thực hiện trong một transaction được gọi là “rolling back” transaction.
* Một transaction thành công và rõ ràng có bốn đặc điểm:
* Atomic (Nguyên tử): Các transaction phải ở dạng nguyên tử, không được phân chia nhỏ ra. Điều này có nghĩa là tất cả các thao tác của transaction phải thành công hoặc thất bại hoàn toàn.
* Consistent (Nhất quán): Các transaction phải nhất quán, có nghĩa là kết quả phải chính xác với những gì mà chúng ta mong đợi.
* Isolated (Riêng biệt): Các transaction phải riêng biệt, có nghĩa là không có bất cứ thứ gì biết về transaction cho đến khi nó được ký thác an toàn (committed).
* Durable (Bền lâu): Các transaction phải bền lâu, có nghĩa là vẫn phải tồn tại sự thất bại. Bởi nếu không tồn tại sự thất bại thì việc sử dụng transaction là vô ích. Khi một transaction hoàn tất, nó sẽ được ký thác an toàn (committed). Điều này có nghĩa là những thay đổi đã có hiệu lực. Đối với một transaction bền lâu, nó phải duy trì được trạng thái ký thác an toàn của nó nếu như thất bại xảy ra.

# 1.1.3.2.Hosting

Host một WCF Service là nói đến môi trường mà service sẽ tồn tại và hoạt động. Có hai phương pháp chính để host:

* Self–Hosting: tự xây dựng môi trường để dịch vụ tồn tại. Các ứng dụng host có trách nhiệm tạo ra và kiểm soát thời gian sống của dịch vụ. Trong môi trường này, các ứng dụng host có trách nhiệm đối với tất cả các khía cạnh kiểm soát của dịch vụ, như việc định nghĩa các Endpoint và tạo ra các ServiceHost.
* Hosted: sử dụng môi trường host có sẵn như IIS. Dịch vụ sẽ có lợi là được IIS giám sát tình trạng và sự ổn định của dịch vụ, thực hiện tự động các quy trình phục hồi dịch vụ khi cần thiết.

# 1.1.3.3.Security

Windows Communication Foundation là một nền tảng ứng dụng phân tán, trong đó việc liên lạc giữa client và server thông qua các thông điệp SOAP. Do đó việc bảo mật, đảm bảo chắc chắn cho việc chuyển giao an toàn các thông điệp là điều cực kỳ quan trọng để tránh việc mất mát hoặc bị ăn cắp thông tin hay tấn công hệ thống. Việc bảo mật trong WCF được xác định trong bốn tính năng quan trọng:

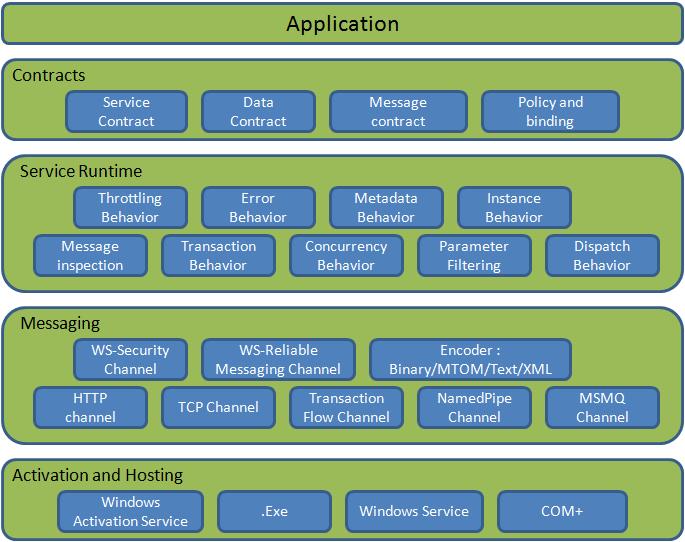
* Bảo mật (Confidentiality): tính năng này đảm bảo thông điệp được gửi bởi bên gửi chỉ có thể được xem và hiểu bởi bên nhận, các bên không có thẩm quyền sẽ không thể xem được các thông điệp được truyền giữa bên gửi và bên nhận. Điều này đạt được thông qua việc sử dụng các thuật toán mã hóa.
* Toàn vẹn (Integrity): tính năng này đảm bảo bên nhận sẽ nhận được chính xác thông điệp mà bên gửi đã gửi. Để làm được điều này, bên gửi phải sử dụng chữ ký điện tử (digital signature) để ký trên các thông điệp gửi.
* Chứng thực (Authentication): tính năng này dùng để kiểm tra người gửi và người nhận là ai và họ có được biết bởi hệ thống hay không. Thông thường việc chứng thực phải được diễn ra ở hai phía, bên nhận và bên gửi. Quá trình chứng thực cả hai phía sẽ chắc chắn thông điệp sẽ không bị thay đổi trong lúc truyền.
* Phân quyền (Authorization): việc phân quyền sẽ xảy ra sau khi chứng thực. Tại thời điểm này đã biết được người gửi hoặc người nhận là ai. Tuy nhiên, cần phải biết được liệu họ có được cấp quyền để thực hiện hành động mà họ đang yêu cầu hay không. Quá trình này sẽ chỉ ra các tài nguyên, chức năng và tính năng có thể được truy cập. Để thực hiện điều này, WCF sử dụng một số kỹ thuật như Windows Indentity, Public Key Infrastructure (PKI), X.509, Kerberos.

# 1.1.3.4.Queuing

WCF cung cấp tính năng queuing, cho phép các thông điệp được lưu trữ an toàn, và cung cấp một trạng thái thống nhất trong truyền thông. Queuing tập hợp và lưu các thông điệp gửi từ bên gửi và chuyển chúng sang bên nhận. Điều này cung cấp kỹ thuật chuyển giao thông điệp an toàn và đáng tin cậy.

# 1.1.4.Kiến trúc Windows Communication Foundation

Trung tâm của WCF là kiến trúc nhiều lớp hỗ trợ nhiều cách thức phát triển ứng dụng phân tán.



*Hình 6:Kiến trúc WCF*

# 1.1.4.1.Contracts

* Contract trong WCF cung cấp khả năng tương thích cần thiết để truyền thông với client. Thông qua các contract mà client và service đồng ý thỏa thuận các kiểu hoạt động và các cấu trúc mà chúng sẽ sử dụng trong suốt giai đoạn truyền thông qua lại. Nếu không có các contract thì sẽ không có công việc nào được hoàn thành bởi vì không có sự thỏa thuận nào xảy ra.
* Có ba contract được sử dụng để định nghĩa một WCF Service:
* Service Contract: Định nghĩa các phương thức của service, đó là những hoạt động có sẵn trên Endpoint cho client.
* Data Contract: Định nghĩa các kiểu dữ liệu trao đổi được sử dụng bởi các phương thức có sẵn của service.
* Message Contract: Cung cấp khả năng kiểm soát cơ bản các định dạng header của thông điệp trong suốt quá trình tạo thông điệp.
* Policy and Binding: Chỉ định các thông tin quan trọng như bảo mật, giao thức, thông tin khác và những policy này được thẩm vấn tìm kiếm những thứ cần thiết để thỏa mãn trước khi hai dịch vụ bắt đầu truyền thông với nhau.

# 1.1.4.2.Service Runtime

Tầng Service Runtime là tầng đặc tả và quản lý các hành vi của dịch vụ xảy ra trong suốt quá trình thực thi của dịch vụ. Một số behavior khác nhau được quản lý bởi tầng Service Runtime:

* Throttling Behavior: Xác định số lượng thông điệp được xử lý.
* Error Behavior: Chỉ định hành động sẽ được kèm theo nếu có lỗi xảy ra trong suốt quá trình thực thi của dịch vụ.
* Metadata Behavior: Kiểm soát metadata có được đưa ra bên ngoài hay không.
* Instance Behavior: Định hướng cách các thực thể của service sẽ có sẵn để xử lý thông điệp.
* Message Inspection: cung cấp cho dịch vụ khả năng kiểm tra tất cả hay một phần thông điệp.
* Transaction Behavior: cho phép các hoạt động giao dịch. Nếu bị thất bại trong quá trình thực thi của service, nó có khả năng rollback transaction.
* Dispatch Behavior: Khi thông điệp được xử lý bởi các phần của WCF, Dispatch Behavior Serice xác định cách thông điệp được quản lý và xử lý.
* Concurrency Behavior: xác định cách mỗi service hay thực thể của service quản lý tiến trình. Hoạt động này trợ giúp kiểm soát cách mà nhiều tiến trình có thể truy cập một thực thể của service được đưa ra.
* Parameter Filtering: Khi một thông điệp được thực thi bởi dịch vụ, một số hành động nhất định có thể được thực hiện dựa trên những gì hiện có trong header của thông điệp. Parameter Filtering sẽ lọc các header của thông điệp và thực thi các hành động được định sẵn trên bộ lọc của header thông điệp.

# 1.1.4.3.Messaging

Tầng Messaging định nghĩa các định dạng và mẫu dữ liệu trao đổi có thể được sử dụng trong suốt quá trình truyền thông của dịch vụ. Các ứng dụng client có thể được phát triển để truy cập tầng này, kiểm soát chi tiết việc thông tin, làm việc trực tiếp với các thông điệp và các kênh. Một số kênh và các thành phần mà tầng Messaging đưa ra:

* WS Security Chanel: thực thi đặc tả WS-Security, cho phép bảo mật thông điệp.
* WS Reliable Messaging Chanel: quá trình chuyển giao thông điệp bảo đảm được cung cấp bởi WS Reliable messaging channel.
* Encoders: cho phép chọn số lượng encoding cho thông điệp.
* HTTP Channel: nói cho service biết thông điệp chuyển giao sẽ diễn ra với giao thức HTTP.
* TCP Channel: nói cho service biết thông điệp chuyển giao sẽ diễn ra với giao thức TCP.
* Transaction Flow Channel: chi phối các mẫu thông điệp được thực hiện.
* NamedPipe Channel: cho phép quá trình liên giao tiếp.
* MSMQ Channel: nếu service cần tương thích với MSMQ, đây là kênh cho phép điều đó.

# 1.1.4.4.Activation and Hosting

Tầng Activation and Hosting cung cấp các lựa chọn khác nhau, mà trong đó các dịch vụ có thể được bắt đầu cũng như được host. Các dịch vụ có thể được host với ngữ cảnh của các ứng dụng khác hay chúng có thể được tự host. Tầng này cung cấp một số các lựa chọn hosting và activation:

* Windows Activation Service: cho phép các ứng dụng WCF tự động bắt đầu khi chạy trên máy tính khi Windows Activation Service đang chạy.
* .EXE: WCF cho phép các dịch vụ được chạy như tập tin thực thi.
* Windows Service: WCF cho phép các dịch vụ được chạy trên Windows service.
* COM+: WCF cho phép các dịch vụ chạy như ứng dụng COM+.

# 1.1.5.Cài Đặt WCF

* Cách sử dụng Windows Communication Foundation để bảo mật dịch vụ Web
* Làm thế nào để sử dụng xác thực chứng nhận và bảo mật ở tầng truyền tải trong WCF khi được gọi từ các ứng dụng Windows Form.
* Phần này sẽ trình bày quá trình sử dụng giấy chứng nhận (certificate) của khách hàng (client) để xác thực (authenticate) người dùng với dịch vụ được thực hiện bảo mật ở tầng truyền tải (transport). Trước tiên chúng ta sẽ tìm hiểu cách tạo ra và cài đặt một certificate cho client để chứng thực và một certificate cho dịch vụ để bảo mật ở tầng transport trong quá trình phát triển. Sau đó, chúng ta sẽ tìm hiểu cách cấu hình một nối kết (binding) để hiện thực IMetadataExchange trong một dịch vụ WCF, và cách tạo một tập tin svcutil.exe.config để cho phép tạo lập bộ phận trung gian (proxy) cần thiết từ client trong quá trình thực hiện bảo mật tại tầng transport và xác thực certificate khi một dịch vụ WCF được đặt trong Internet Infomation Services (IIS). Cuối cùng, chúng ta sẽ tìm hiểu cách cấu hình chính xác cài đặt bảo mật trong IIS, và làm thế nào để thử nghiệm dịch vụ với một ứng dụng tại client.
* Tóm tắt các bước:

Bước 1: Tạo và cài đặt một certificate tạm để bảo mật tầng transport

Bước 2: Tạo và cài đặt một certificate tạm cho client để xác thực certificate.

Bước 3: Tạo một dịch WCF

Bước 4: Cấu hình basicHttpBinding với xác thực certificate và bảo mật ở tầng transport

Bước 5: Cấu hình mex Endpoint để sử dụng basicHttpBinding xác thực certificate

Bước 6: Cấu hình thư mục ảo (virtual directory) để sử dụng Secure Sockets Layer (SSL) và yêu cầu các certificate từ client

Bước 7: Tạo một ứng dụng thử nghiệm tại client

Bước 8: Tạo một tập tin cấu hình Svcutil tại máy client

Bước 9: Tạo một proxy với công cụ svcutil.exe

Bước 10: Thử nghiệm ứng dụng tại client với dịch vụ WCF

**Bước 1: Tạo và cài đặt một certificate tạm để bảo mật tầng transport**

Trong bước này, chúng ta sẽ tạo và cài đặt một certificate tạm để bảo mật tầng transport ở máy chủ (server) đặt dịch vụ. Chúng ta cũng được yêu cầu cài đặt CA gốc trên client để chuỗi xác nhận tin tưởng thành công khi duyệt qua dịch vụ trong Microsoft Internet Explorer, tạo ra proxy server, và gọi dịch vụ từ proxy. Để hoàn tất việc này, hãy xem lại phần “Làm thế nào để tạo và cài đặt các chứng nhận tạm trong WCF để bảo mật ở tầng truyền tải trong quá trình phát triển”.

**Bước 2: Tạo và cài đặt một certificate tạm cho client để xác thực certificate**

Trong bước này, chúng ta sẽ tạo và cài đặt một certificate tạm cho client để xác thực certificate trên client. Chúng ta cũng được yêu cầu cài đặt CA gốc và CRL trên server để chuỗi tin tưởng và xác nhận thu hồi thành công. Để hoàn tất các bước tạo những certificate sử dụng trong việc xác thực certificate, hãy xem lại phần “Làm thế nào để tạo và cài đặt các chứng nhận tạm cho client trong WCF để xác thực chứng nhận trong quá trình phát triển”.

**Bước 3: Tạo một dịch WCF**

Trong bước này, chúng ta sẽ tạo ra một dịch vụ WCF trong Visual Studio

Trong Visual Studio, trên menu **File**, click **New Web Site**

Trong phần **Templates**, chọn **WCF Service**. Hãy chắc chắn rằng **Location** được thiết lập là **Http** và đặc tả thư mục ảo được tạo ra trong Path là **https** (Ví dụ: <https://ServerName/WCFTestService>)

Lưu ý: Tên server cần phải khớp với tên certificate, hoặc là tên của NetBIOS hoặc tên DNS.

Trong hộp thoại **New Web Site**, chọn **OK** để tạo một thư mục ảo và WCF service.

**Bước 4: Cấu hình basicHttpBinding với xác thực certificate và bảo mật ở tầng transport**

Trong bước này, chúng ta sẽ cấu hình WCF service để sử dụng chứng thực thông qua certificate và bảo mật ở tầng transport.

* Chọn chuột phải trên tập tin **web.config** của dịch vụ WCF và sau đó click **Edit WCF Configuration**.
* Trong trình soạn thảo cấu hình, trong phần **Configuration**, mở nút **Service** và sau đó mở thư mục **Enpoints**.
* Chọn nút **[Empty Name]** đầu tiên và thiết lập thuộc tính **Name** là **basicHttpEndpoint**. Mặc định, thuộc tính này được để trống bởi vì nó là thuộc tính tùy chọn. Tiếp tục thiết lập thuộc tính **Binding** là **basicHttpBinding** bằng việc chọn lựa chọn này từ drop-down list.
* Chọn tab **Identity** và sau đó xóa đi giá trị ở thuộc tính **Dns**.
* Trong trình soạn thảo cấu hình, chọn thư mục **Bindings** rồi chọn **New Binding Configuration**.
* Trong hộp thoại **Create a New Binding**, chọn **basicHttpBinding**. Chọn **OK**.
* Đặt tên cho cấu hình binding thông qua thuộc tính **Name** với một số tên hợp lý và dễ nhận biết, ví dụ: **basicHttpEndpointBinding**.
* Chọn tab **Security** và chắc chắn rằng thuộc tính **Mode** được thiết lập là **Transport**.
* Đặt **TransportClientCredentialType** là **Certificate** bằng việc chọn lựa chọn này từ drop-down list.
* Trong phần **Configuration**, chọn nút **basicHttpEndpoint**.
* Đặt thuộc tính **BindingConfiguration** là **basicHttpEndpointBinding** bằng việc chọn lựa chọn này từ drop-down list. Điều này sẽ liên kết cài đặt cấu hình binding với binding.
* Trong trình soạn thảo cấu hình, trên menu **File**, chọn **Save**.
* Trong Visual Studio, mở tập tin cấu hình lên và bỏ phần thẻ identity.

<!--<identity>

<dns value="" />

</identity>-->

* Trong Visual Studio, kiểm tra lại cấu hình của chúng ta. Cấu hình sẽ có dạng như sau:

…

<system.serviceModel>

<bindings>

<basicHttpBinding>

<binding name="basicHttpEndpointBinding">

<security mode="Transport">

<transport clientCredentialType="Certificate" />

</security>

</binding>

</basicHttpBinding>

</bindings>

<services>

<service behaviorConfiguration="ServiceBehavior" name="Service">

<endpoint address="" binding="basicHttpBinding" bindingConfiguration="basicHttpEndpointBinding"

name="basicHttpEndpoint" contract="IService" />

<endpoint address="mex" binding="basicHttpBinding" contract="IMetadataExchange" />

</service>

</services>

…

**Bước 5: Cấu hình mex Endpoint để sử dụng basicHttpBinding xác thực certificate**

Trong bước này, chúng ta sẽ thay đổi cấu hình của metadata exchange (mex) endpoint từ **mexHttpBinding** (mặc định) để sử dụng **basicHttpBinding** với việc cấu hình chúng ta đã tạo từ bước trước để sử dụng xác thực certificate. **mexHttpBinding** không thể dùng cho việc chứng thực thông qua certificate bởi vì Web site yêu cầu SSL, và **mexHttpEndpoint** cũng không được sử dụng bởi vì không hỗ trợ việc cấu hình xác thực certificate trong IIS. Để tạo một proxy cho WCF service được đặt trong IIS với một cấu trúc chứng thực thông qua certificate, chúng ta cần một endpoint để hiện thực **IMetadataExchange** với **basicHttpBinding**, với một cấu hình bảo mật cho phép xác thực certificate.

* Trong trình soạn cấu hình, trong phần **Configuration**, mở nút **Service** và sau đó mở nút **Endpoints**.
* Chọn nút **[Empty name]** thứ hai và đặt thuộc tính **Name** là **mexEnpoint**. Mặc định, giá trị tại thuộc tính này sẽ rỗng bởi vì nó là một thuộc tính tùy chọn.
* Click thuộc tính **Binding** để thay đổi thành **basicHttpbinding** bằng việc chọn lựa chọn này từ drop-down list.
* Click thuộc tính **BindingConfiguration** và thay đổi thành **basicHttpEndpointBinding** bằng việc chọn lựa chọn này từ drop-down list. Điều này sẽ liên kết **mexEndpoint** với cài đặt cấu hình binding để cấu hình việc xác thực certificate.
* Trong trình soạn thảo cấu hình, trên menu **File**, chọn **Save**.
* Trong Visual Studio, kiểm tra lại cấu hình. Cấu hình sẽ giống như bên dưới. Lưu ý thằng endpoint bây giờ là một basicHttpbinding endpoint hiện thực IMetadataExchange.

…

<system.serviceModel>

<bindings>

<basicHttpBinding>

<binding name="basicHttpEndpointBinding">

<security mode="Transport">

<transport clientCredentialType="Certificate" />

</security>

</binding>

</basicHttpBinding>

</bindings>

<services>

<service behaviorConfiguration="ServiceBehavior" name="Service">

<endpoint address="" binding="basicHttpBinding" bindingConfiguration="basicHttpEndpointBinding"

name="basicHttpEndpoint" contract="IService" />

<endpoint address="mex" binding="basicHttpBinding" bindingConfiguration="basicHttpEndpointBinding"

name="mexEndpoint" contract="IMetadataExchange" />

</service>

</services>

<behaviors>

<serviceBehaviors>

<behavior name="ServiceBehavior">

<!-- To avoid disclosing metadata information, set the value below to false and remove the metadata endpoint above before deployment -->

<serviceMetadata httpGetEnabled="false" httpsGetEnabled="true"/>

<!-- To receive exception details in faults for debugging purposes, set the value below to true. Set to false before deployment to avoid disclosing exception information -->

<serviceDebug includeExceptionDetailInFaults="true"/>

</behavior>

</serviceBehaviors>

</behaviors>

</system.serviceModel>

…

**Bước 6: Cấu hình thư mục ảo (virtual directory) để sử dụng Secure Sockets Layer (SSL) và yêu cầu các certificate từ client**

Trong bước này, chúng ta sẽ cấu hình thư mục ảo trong IIS để sử dụng mã hóa SSL và để yêu cầu client cung cấp certificate.

* Chọn **Start** và sau đó chọn **Run**.
* Trong hộp thoại **Run**, gõ **inetmgr** và sau đó chọn OK.
* Trong hộp thoại **Internet Information Service (IIS) Manager**, mở nút **(local computer)** và sau đó mở nút **Web Sites**.
* Mở nút **Default Web Site** và sau đó chuột phải vào thư mục ảo, chọn **Properties**.
* Trong hộp thoại **Virtual Directory Properties**, chọn tab **Directory Security** và sau đó chọn **Edit** trong mục **Secure communications**. Chọn **Require Secure Channel (SSL)** và sau đó chọn **Require Client Certificates**.

Bây giờ chúng ta có thể duyệt dịch vụ bằng cách sử dụng Internet Explorer đi đến địa chỉ <https://ServerName/WCFTestService>. Internet Explorer sẽ nhắc chọn một certificate từ một danh sách các certificate được cài đặt trong tài khoản người dùng và các nơi lưu trữ cá nhân.

**Bước 7: Tạo một ứng dụng thử nghiệm tại client**

Trong bước này, chúng ta sẽ tạo một ứng dụng Windows Form để thử nghiệm dịch vụ WCF.

* Chọn chuột phải vào solution, chọn **Add** và sau đó chọn **New Project**.
* Trong hộp thoại **Add New Project**, trong phần **Templates**, chọn **Windows Forms Application**.
* Đặt tên cho ứng là **TestClient** và sau đó chọn **OK**.

**Bước 8: Thêm tham chiếu dịch vụ WCF vào ứng dụng tại client**

Trong bước này, chúng ta sẽ thêm một tham chiếu từ ứng dụng Windows Form của client đến dịch vụ WCF đã được xây dựng tại các bước trên.

Lưu ý: khi thực hiện bước này, chúng ta nên tắt chế độ chứng thực certificate tại dịch vụ WCF, bằng cách thay đổi giá trị thuộc tính **clientCredentialType** thành **None** tại thẻ **transport** được chứa bên trong thẻ **security** trong tập tin cấu hình của dịch vụ WCF. Đồng thời tắt chế độ **Require client certificate** trong IIS tương ứng với dịch vụ WCF đang sử dụng. Nếu không thực hiện bước này, việc tham chiếu đến dịch vụ WCF sẽ báo lỗi. Sau khi thực hiện xong các bước tham chiếu đến dịch vụ WCF, chúng ta cần trả các giá trị vừa thay đổi trong cấu hình dịch vụ và cấu hình IIS trở lại ban đầu để đảm bảo tính bảo mật.

* Chọn chuột phải vào project client, chọn **Add Service Reference**.
* Tại hộp thoại **Add Service Reference**, chọn **Advanced**.
* Tại hộp thoại **Service Reference Settings**, chọn **Add Web Reference**.
* Trong hộp thoại **Add Web Reference**, điền URL của dịch vụ WCF, chọn **Go** để tìm kiếm dịch vụ. Sau khi tìm kiếm dịch vụ thành công, tiếp tục đặt tên cho tham chiếu dịch vụ web này (Ví dụ: WCFTestService), sau đó chọn **Add Reference**.

Bây giờ trong ứng dụng Windows Form, chúng ta sẽ thấy xuất hiện mục **WCFTestService** bên trong thư mục **Web References** của ứng dụng, và một tập tin cấu hình **app.config**.

**Bước 9: Thử nghiệm ứng dụng tại client với dịch vụ WCF**

Trong bước này, chúng ta sẽ truy cập vào WCF Service, truyền vào certificate của người dùng để chắc rằng việc xác thực bằng certificate làm việc hiệu quả.

Trong ứng dụng tại client, kéo một button vào trên form.

Double click vào button để hiện ra code bên dưới.

Tạo một thực thể của proxy và gọi hàm **GetData** của WCF Service. Đoạn code sẽ giống như sau:

private void button1\_Click(object sender, EventArgs e)

{

WCFTestService.Service proxy = new WCFTestService.Service();

X509Certificate cert = new X509Certificate(@"E:\exporttempcert.pfx", "123456789");

proxy.ClientCertificates.Add(cert);

MessageBox.Show(proxy.GetData(123, true));

}

Chọn chuột phải trên project client và sau đó chọn **Set as StartUp Project**.

Chạy ứng dụng client bằng cách nhấn F5 hoặc Ctrl+F5.

Trong khi bạn nhấn vào button trên form, thì thông điệp “**You entered: 123**” sẽ xuất hiện.

# 1.2.Service Oriented Architecture

# 1.2.1.Service

* Về mặt định nghĩa, service là một hệ thống có khả năng nhận một hay nhiều yêu cầu xử lý và sau đó đáp ứng lại bằng cách trả về một hay nhiều kết quả. Quá trình nhận yêu cầu và trả kết quả về được thực hiện thông qua các interface đã được định nghĩa trước đó. Thông thường việc giao tiếp này được thực hiện trên các interface đã được chuẩn hóa và sử dụng rộng rãi.
* Một ví dụ đơn giản của service chính là hoạt động của một nhà hàng. Khi khách hàng vào nhà hàng và gọi thức ăn, khách hàng đang tiến hành gởi yêu cầu cho service “phục vụ khách hàng” của nhà hàng. Nhân viên nhà hàng nhậc được yêu cầu của khách, nếu món ăn khách hàng yêu cầu nhà hàng không có hoặc đã hết, nhân viên nhà hàng sẽ từ chối hoặc đề nghị khách hàng gọi món khác. Nếu nhà hàng có thể đáp ứng được yêu cầu của khách, món ăn sẽ được chế biến và mang ra cho khách hàng thưởng thức sau một khoảng thời gian chờ. Ở đây, yêu cầu chính là món ăn mà khách hàng muốn thưởng thức, còn kết quả trả về của service phục vụ nhà hàng chính là từ chối (nếu nhà hàng không đáp ứng được yêu cầu của khách) hay là món ăn mà khách hàng cần.
* Một hệ thống được thiết kế theo kiểu hướng service (service oriented) là một hệ thống trong đó các chức năng của hệ thống được xây dựng dựa trên các service có độ kết dính thấp. Các service trong hệ thống giao tiếp với nhau thông qua việc gởi nhận các thông điệp (message).

# 1.2.2.Các Đặc Điểm Chính Của Service

* Có ranh giới rõ ràng (Boundaries Are Explicit):

Mỗi service được xây dựng dựa trên các interface chuẩn hóa đã được sử dụng rộng rãi.  
Chi tiết hiện thực của mỗi service sẽ không được thể hiện ra bên ngoài. Mỗi service chỉ công bố một số các interface của nó cho user có thể dùng để gởi các yêu cầu và nhận kết quả trả về.

* Tính tự trị (Autonomous):

Về mặt lý thuyết, mỗi service có tính độc lập cao, có thể được build và đưa vào sử dụng mà không phụ thuộc vào các service khác.

* Share the Schema and Contract, Not the Class:  
  Về mặt trao đổi dữ liệu, các service không truyền các class và type. Thay vào đó, các class và type sẽ được đặc tả hình thức (data được đặc tả trong schema, behavior được đặc tả thành các contract )
* Service Compatibility Is Based on Policy:  
  Sự tương thích giữa các service được căn cứ vào các policy.   
  Tương thích về mặt cấu trúc dựa trên các đặc tả hình thức bao gồm contract (dựa trên Web Service Description Language (WSDL) hoặc Business Process Execution Language for Web Services (BPEL4WS)) và schema (XSD)  
  Sự tương thích dựa trên policy cung cấp khả năng phân tích cũng như đảm bảo sự tương thích giữa các service.

# 1.2.3.Service Oriented Architecture

* Service Oriented Architecture (SOA) cung cấp cơ chế cho phép các hệ thống hoạt động trên các platform khác nhau có thể giao tiếp với nhau.
* Một hệ thống được xây dựng theo mô hình SOA bao gồm các service thỏa mãn các tính chất của service ở mục 2.2. Mỗi service trong hệ thống có thể được sửa đổi một cách độc lập với các service khác nhằm mục đích đáp ứng một yêu cầu mới từ thực tế.

# 1.2.4.Các actor trong SOA

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

*Hình 7: SOA actors*

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

# 1.2.5.Ích lợi khi sử dụng SOA

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

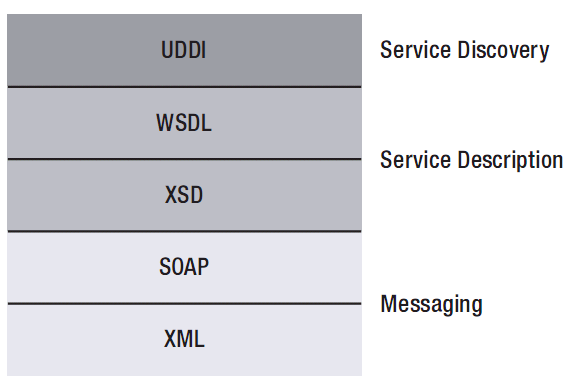
* Lợi ích kinh tế:
* Doanh nghiệp có điều kiện tập trung thời gian để tìm kiếm các giải pháp cho các bài toán liên quan đến kinh tế.
* Thúc đẩy sự phát triển của hệ thống hiện có cũng như cung cấp khả năng mở rộng hệ thống trong tương lai.
* Lợi ích kỹ thuật:
* Hệ thống xây dựng theo mô hình SOA đảm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

# 1.2.6.Thông điệp (message) trong SOA

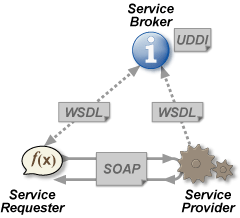
* So với kiểu thiết kế Component-Based, điểm khác biệt chính của SOA là cung cấp khả năng giao tiếp giữa các thành phần trong hệ thống (service) sử dụng thông điệp (message) dựa trên các chuẩn giao tiếp đã được chuẩn hóa (HTTP, FTP, SMTP, ...). Chính nhờ đặc điểm này, hệ thống SOA trở nên độc lập với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

# 1.3.Web Service

Web service là một hiện thực của service sử dụng web protocol để giao tiếp. Web service cung cấp hệ thống phần mềm hỗ trợ việc giao tiếp giữa các máy tính trên network.

Cấu trúc Web service

*Hình 8: Web service protocol stack*

* Hình trên mô tả các layer hình thành nên Web service. Hình dưới mô tả các Web service actor tương ứng với các layer này.
* Tương tự với SOA, có 3 actor chính tham gia vào Web service.
* Service Provider: Dùng Web Services Description Language (WSDL) để mô tả dịch vụ mà mình có thể cung cấp cho Service Broker (tương tự với Service Registry trong SOA).
* Service Broker: Lưu trữ thông tin về các service được cung cấp bởi các Service Provider. Cung cấp chức năng tìm kiếm hỗ trợ Service Requester (Service Consumer trong SOA) trong việc xác định Service Provider phù hợp. Thành phần chính của Service Broker là Universal Discovery, Description, and Integration (UDDI) repositories.
* Service Requester: Dùng WSDL để đặc tả nhu cầu sử dụng (loại service, thời gian sử dụng, resource cần thiết, mức giá ...) và gởi cho Service Broker. Bằng việc sử dụng UDDI và chức năng tìm kiếm của Service Broker, Service Requester có thể tìm thấy Service Provider thích hợp. Ngay sau đó, giữa Service Requester và Service Provider thiết lập kênh giao tiếp sử dụng SOAP để thương lượng giá cả và các yếu tố khác trong việc sử dụng service.

*Hình 9: Web service actors*

# 2.THIẾT KẾ CHƯƠNG TRÌNH

# 2.1Mô tả ứng dụng

# 2.1.1.Ứng dụng service host

* Service host theo đề tài chúng tôi thực hiện có nhiệm vụ cung cấp thông tin nhà hàng, cà phê cho các ứng dụng client sử dụng, service host được thiết kế đơn giản, cho phép người sử dụng có thể tự cấu hình các đường dẫn, endpoint, tắt/mở service tùy theo ý thích của mình.
* Các bước thực hiện:
* Nhập base address cho service, check vào phần show MEX để ứng dụng client có thể đọc được service.
* Chọn kiểu binding và relative address cho các endpoint, chương trình cho phép chạy cùng lúc 3 enpoint với các kiểu binding như BasicHttpBinding, WsHttpBinding, NetTcp.
* Sau đó ta nhấn Start để chạy service, ứng dụng thông báo “service is running” là service đã khởi chạy thành công, khi ta nhấn Stop để dừng service, ứng dụng thông báo “service is stopped” là service đã dừng lại.
* Service host này sẽ do những người quản trị, bảo trì và nâng cấp service quản lý. Service host này chứa hầu hết các chức năng phục vụ cho quy trình cung cấp thông tin nhà hàng,cà phê, hỗ trợ cho các bên liên quan có thể tìm được những thông tin hữu ích 1 cách nhanh chóng, đảm bảo cho công việc được thuận lợi, cũng như đánh giá được chất lượng của nhà hàng, cà phê.

# 2.1.2.Ứng dụng client

* Client sẽ tham chiếu đến service host để sử dụng các chức năng mà service host cung cấp, cụ thể client sẽ làm nhiệm vụ trình bày, xử lý các dữ liệu nhận được, phản hồi một số thông tin lại cho service. Client sẽ được thiết kế theo dạng 1 website.
* Client sẽ cung cấp các chức năng như tạo tài khoản cho thành viên , đăng nhập, chức năng tìm kiếm thông tin nhà hàng (theo tên,địa điểm, vùng miền, món ăn, loại hình….), xem thông tin nhà hàng, cà phê, báo cáo thống kê thông tin nhà hàng, cà phê cũng như việc hỗ trợ in ấn, xuất file theo các định dạng mình muốn, gửi comment về cho quản lý thông tin nhà hàng, cà phê.
* Client này chúng tôi thiết kế sẽ hỗ trợ cho 2 hướng người dùng chủ yếu là người dụngvàcác quản lý nhà hàng (quản lý dại diện, chủ nhà hàng).

# 2.2.Đặc tả các chức năng (operation contracts) cà phê nhà hàng:

# 2.2.1.Tìm kiếm theo địa điểm:

* Chức năng tìm kiếm các nhà hàng, cà phê (NH-CF)theo thành phố, tỉnh, quận, phường mà người dùng yêu cầu. Thêm nữa, tìm kiếm các nhà hàng cà phê theo 1 địa điểm cụ thể mà khách hàng yêu cầu.
* Đặc tả hàm:
* Intput: Timdiadiem(thanhpho, tinh, quan, phuong)
* Output: Danh sách các NH-CF có các thuộc tính thành phố, tỉnh, quận, phường trùng với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| thanhpho | string | tên thành phố được nhập để tìm kiếm NH-CF theo tên thành phố. |
| tinh | string | têntỉnh được nhập để tìm kiếm NH-CF theo tên tỉnh. |
| quan | string | tênquận hoặc số quận được nhập để tìm kiếm NH-CF theo quận. |
| phuong | int | số phường được nhập để tìm kiếm NH-CF theo phường. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo địa điểm trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn hoặc nhậpvào form tìm kiếm trên các thuộc tính thành phố, tỉnh, quận, phường. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm nào đó, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó

# 2.2.2.Tìm kiếm theo không gian:

* Chức năng tìm kiếm các nhà hàng, cà phê có không gian theo ý của người dùng, ví dụ: như nhà hàng-cà phê phục vụ trong nhà, hay sân vườn, hay cả hai, ánh sáng nhạt, vàng, huyền ảo.., âm nhạc của nhà hàng cà phê là nhạc nhẹ, hòa tấu, hay nhạc dance, âm lương như thế nào có DJ phục vụ không và vào lúc nào (ví dụ sáng NH-CF thường mở nhạc nhẹ, thời gian tối thì nhạc dance và có DJ phục vụ).
* Đặc tả hàm:
* Input: Timkhonggian(vitri, anhsang, amnhac, dj, thoigian)
* Output: Danh sách các NH-CF có các không gian trùng với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| Loai | string | Loại không gian được nhập để tìm kiếm NH-CF theo loại không gian. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo không gian trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn trong form tìm kiếm trên các thuộc tính vị trí, ánh sáng, âm nhạc, dj, thời gian bằng cách check theo các yêu cầu tìm kiếm của mình trên mỗi thuộc tính. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm nào đó, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó.

# 2.2.3.Tìm kiếm theo mục đích của NH-CF:

* Chức năng tìm kiếm mục đích của nhà hàng cà phê là Âu, Á, hay truyền thống, tông màu của quán, tranh ảnh trang trí như thế nào theo yêu cầu của người dùng.
* Đặc tả hàm:
* Input: Timphongcach(phongcach, tongmau, trangtri)
* Output: Danh sách các NH-CF có các thuộc tính phong cách, tông màu, trang trí trùng với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| phongcach | string | phong cách phục vụ được nhập để tìm kiếm NH-CF theo phong cách phục vụ. |
| tongmau | string | tông màu của quán được nhập để tìm kiếm NH-CF theo tông màu. |
| trangtri | string | phong cách trang trí được nhập để tìm kiếm NH-CF theo phong cách trang trí. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo phong cách trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn trong form tìm kiếm trên các thuộc tính phong cách, tông màu, trang trí bằng cách check theo các yêu cầu tìm kiếm của mình trên mỗi thuộc tính. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm nào đó, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó.

# 2.2.4.Tìm kiếm theo phong cách món ăn:

* Chức năng tìm kiếm theo các loại món ăn, thức uống được phục vụ trong NH-CF, Các món ăn Nhật, Hàn, Pháp, Việt, Trung, Ấn, Thái,.. Thức uống phục vụ là cà phê, nước trái cây, bia, rượu theo yêu cầu của người dùng.
* Đặc tả hàm:
* Input: Timmonan(monan, thucuong)
* Output: Danh sách các NH-CF có các thuộc tính món ăn, thức uống trùng với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| monan | string | phong cách món ăn được nhập để tìm kiếm NH-CF theo phong cách món ăn. |
| thucuong | string | phong cách thức uống được nhập để tìm kiếm NH-CF theo phong cách nước uống. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo phong cách ăn uống trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn trong form tìm kiếm trên các thuộc món ăn, thức uống bằng cách check vào các món ăn thức uống hoặc nhập vào tên các món ăn thức uống mà họ muốn tìm. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm nào đó, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó.

# 2.2.5.Tìm kiếm theo phong cách phục vụ:

* Chức năng tìm kiếm theo phong cách phục vụ của NH-CF theo yêu cầu của người dùng như phong cách tự phục vụ (buffet) , phục vụ lẩu. phục vụ theo bang chuyền, phục vụ bánh, phục vụ thức ăn nhanh, có người phục vụ tại bàn không theo yêu cầu người dùng.
* Đặc tả hàm:
* Input: Timphucvu(phucvu)
* Output: Danh sách các NH-CF có các thuộc tính phong cách phục vụ trùng với yêu cầu tìm kiếm của người dung.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| phucvu | string | phong cách phục vụ được nhập để tìm kiếm NH-CF theo phong cách phục vụ. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo phong cách phục vụ trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn trong form tìm kiếm trên thuộc tính về các phong cách phục vụ phổ biến được hệ thống định sẵn bằng cách check vào các ô với các phong cách phục vụ đó. Ngoài ra, người dùng có thể tìm theo các phong cách phục vụ khác theo nhu cầu của mình không có trong danh mục phong cách phục vụ của hệ thống bằng cách nhập các phong cách phục vụ ấy vào trong ô phong cách phục vụ khác, mỗi phong cách phục vụ người dùng nhập them sẽ cách nhau bằng 1 dấu phẩy. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm và các ô check, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó.

# 2.2.6.Tìm kiếm theo các loại món ăn thức uống đặc sắc của NH-CF:

* Chức năng tìm kiếm những món ăn thức uống nổi bật, có những yếu tố đặc biệt như nổi tiếng ngon, rẻ, độc đáo của NH-CF theo sở thích của người dùng.
* Đặc tả hàm:
* Input: Timdacsan(ngon, re,docdao, tenmonan, tenthucuong)
* Output: Danh sách các NH-CF có các thuộc tính ngon rẻ, độc đáo theo tên món ăn thức uống phù hợp với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| ngon | boolean | Tìm kiếm với tiêu chí món ăn thức uống ngon hay không. |
| re | boolean | Tìm kiếm với tiêu chí món ăn thức uống rẻ hay không. |
| docdao | boolean | Tìm kiếm với tiêu chí món ăn thức uống độc đáo hay không. |
| tenmonan | string | tên món ăn cần tìm thỏa các tiêu chí tìm kiếm trên được nhập để tìm kiếm NH-CF theo các tiêu chí đó. |
| tenthucuong | string | tênthức uống cần tìm thỏa các tiêu chí tìm kiếm trên được nhập để tìm kiếm NH-CF theo các tiêu chí đó. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo đặc sản trên được trả về. |

Binding: BasicHttpBinding

* Tiêu chí đánh giá món ăn thức uống ngon, rẻ, độc đáo là do NH-CF đã nổi tiếng về những điều đó từ lâu đời và hầu hết mọi người ở địa phương đều biết, hoặc tiêu chí có thể do phần lớn thành viên đánh giá cao và được xét duyệt là đúng sự thật. Người dùng sẽ check trong form tìm kiếm các thuộc tính ngon, rẻ và nhập tên món ăn thức uống ngon, rẻ mà người dùng muốn tìm kiếm. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô check nào đó, hệ thống sẽ không tìm kiếm theo thuộc tính đó. Nếu người dùng bỏ trống ô tên món ăn và tên thức uống, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó. Nếu người dùng check vào ô độc đáo, thì các ô tên món ăn và tên thức uống sẽ bị mờ, hệ thống sẽ tự động tìm các món ăn thức uống độc đáo tại các NH-CF cho người dùng thỏa điều kiện tìm kiếm ngon, rẻ của họ hay không. Nếu muốn tìm món ăn độc đáo theo tên, người dùng dùng chức năng Timmonan.

# 2.2.7.Tìm kiếm theo nhu cầu khác:

* Chức năng tìm kiếm theo các nhu cầu khác của người dùng như NH-CF có nhạc sống không, có Hát với nhau không,..
* Đặc tả hàm:
* Input: Timnhucau(nhucau)
* Output: Danh sách các NH-CF có các thuộc tính về nhu cầu trùng với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| nhucau | string | các nhu cầu khác của người dùng được nhập để tìm kiếm NH-CF theo các nhu cầu thêm đó. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo nhu cầu trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn trong form tìm kiếm trên thuộc tính về các nhu cầu phổ biến được hệ thống định sẵn bằng cách check vào các ô với các nhu cầu đó. Ngoài ra, người dùng có thể tìm theo các nhu cầu khác của mình không có trong danh mục nhu cầu của hệ thống bằng cách nhập các nhu cầu ấy vào trong ô nhu cầu khác, mỗi nhu cầu người dùng nhập them sẽ cách nhau bằng 1 dấu phẩy. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm và các ô check, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó.

# 2.2.8.Tìm kiếm theo giá cả:

* Chức năng tìm kiếm theo giá cả của các NH-CF theo phạm vi giá cả, giá được sắp xếp từ thấp đến cao, hay cao đế thấp cho đối tượng khách hàng bình thường hay khách đặc biệt như VIP, đối tác, khách hàng than thiết,.. theo yêu cầu người dùng.
* Đặc tả hàm:
* Input: Timgia(doituong, giatu, giaden)
* Output: Danh sách các NH-CF với các món ăn thức uống có mức giá trung bìnhphù hợp với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| doituong | string | đối tượng người dùng được nhập để tìm kiếm NH-CF theo phạm vi đối tượng đó. |
| giatu | double | giá thấp nhất được nhập để tìm kiếm NH-CF theo nhu cầu từ giá đó. |
| giaden | double | giácao nhất được nhập để tìm kiếm NH-CF theo nhu cầu đến giá đó. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo giá trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn trong form tìm kiếm trên thuộc tính đối tượng khách mà mình muốn tìm kiếm. Sau đó, người dùng có thể sắp xếp thuộc tính giá cả theo giá từ cao xuống thấp hay từ thấp lên cao tuy theo yêu cầu tìm kiếm của mình, hoặc người dùng có thể nhập vào ô giá từ với mức giá thấp nhất và giá đến với mức giá cao nhất trong phạm vi giá cả mà họ muốn tìm kiếm. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF có giá cả các món ăn thức uống thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm nào đó, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó.

# 2.2.9.Tìm kiếm theo các dịch vụ:

* Chức năng tìm kiếm theo các dịch vụ có tại các NH\_CF như dịch vụ tổ chức yến tiệc, lễ kỷ niệm, lễ cưới, tiệc sinh nhật, phục vụ tại nhà, tổ chức hội nghị, và tổ chức các sự kiện khác theo yêu cầu người dùng.
* Đặc tả hàm:
* Input: Timdichvu(dichvu)
* Output: Danh sách các NH-CF có các thuộc tính về dịch vụ trùng với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| dichvu | string | cácdịch vụ cần tìm của người dùng được nhập để tìm kiếm NH-CF theo các dịch vụ đó. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo dịch vụ trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn trong form tìm kiếm trên thuộc tính về các dịch vụ phổ biến được hệ thống định sẵn bằng cách check vào các ô với các dịch vụ đó. Ngoài ra, người dùng có thể tìm theo các nhu cầu khác về dịch vụ của mình không có trong danh mục dịch vụ của hệ thống bằng cách nhập các dịch vụ them vào ấy vào trong ô dịch vụ khác, mỗi dịch vụ người dùng nhập them sẽ cách nhau bằng 1 dấu phẩy. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm và các ô check, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó

# 2.2.10.Tìm kiếm theo tin tức khuyến mãi:

* Chức năng tìm kiếm các NH-CF theo tin tức khuyến mãi như giảm giá mới khai trương, giảm giá vào các dịp lễ, có các voucher, coupon cho các sản phẩm, dịch vụ của NH-CF.
* Đặc tả hàm:
* Input: Timkhuyenmai(khuyenmai)
* Output: Danh sách các NH-CF có các thuộc tính về khuyến mãi trùng với yêu cầu tìm kiếm của người dùng.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| khuyenmai | string | Các nhu cầu khác của người dùng được nhập để tìm kiếm NH-CF theo các nhu cầu thêm đó. |
| nhahang | list<NH-CF> | danh sách các NH-CF thỏa các điều kiện tìm kiếm theo nhu cầu trên được trả về. |

Binding: BasicHttpBinding

* Người dùng sẽ chọn trong form tìm kiếm trên thuộc tính về các sự kiện khuyến mãi phổ biến được hệ thống định sẵn bằng cách check vào các ô với các sự kiện khuyến mãi đó. Ngoài ra, người dùng có thể tìm theo các nhu cầu khác về sự kiện khuyến mãi của mình không có trong danh mục sự kiện khuyến mãi của hệ thống bằng cách nhập các sự kiện khuyến mãi them vào ấy vào trong ô sự kiện khuyến mãi khác, mỗi sự kiện khuyến mãi người dùng nhập them sẽ cách nhau bằng 1 dấu phẩy. Hệ thống sẽ thực hiện câu lệnh truy xuất vào CSDL để tìm ra các NH-CF thỏa tất cả các điều kiện trên. Nếu người dùng bỏ trống ô tìm kiếm và các ô check, hệ thống sẽ tìm kiếm tất cả, không ràng buộc điều kiện với thuộc tính đó

# 2.2.11.Người dùng đăng ký:

* Chức năng cho phép người dùn đăng ký để trở thành thành viên để có thể đóng góp những đánh giá, góp ý, hay gửi bài của mình.
* Đặc tả hàm:
* Input:Userdangky(username, password, e-mail, hoten, diachi, sodienthoai)
* Output: Hệ thống hiện trang thong báo đăng ký thành viên thành công cho người dùng và chuyển về trang chủ của hệ thống.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| username | string | usename tài khoản do người dùng nhập vào để đăng ký. |
| password | string | password tài khoản do người dùng nhập vào để đăng ký. |
| e-mail | string | e-mail của người dùng nhập vào để đăng ký. |
| hoten | string | họ tên của người dùng do họ nhập vào. |
| diachi | string | địa chỉ của người dùng do họ nhập vào. |
| sodienthoai | string | số điện thoại của người dùng do họ nhập vào. |
| ketqua | thanhvien | khi hệ thống duyệt thành công, hàm trả về kiểu thành viên insert vào CSDL bảng thành viên. |

Binding: BasicHttpBinding

* Người dùng nhập username, password, e-mail, họ tên, địa chỉ, số điện thoại của mình vào form đăng ký thành viên của hệ thống và chọn đăng ký.Sau duyệt qua tất cả thong tin của người dùng đều hợp lệ, hệ thống sẽ thêm thành viên vào CSDL của hệ thống và gửi mail kích hoạt tài khoản vào e-mail của người dùng.Người dùng sau khi kích hoạt tài khoản sẽ nhận được thông báo đăng ký tài khoản thành công.

# 2.2.12.Người dùng đăng nhập:

* Chức năng cho phép người dùng đăng nhập vào tài khoản của mình để có thể đóng góp những đánh giá, góp ý, hay gửi bài của mình.
* Đặc tả hàm:
* Input: Userdangnhap(username, password)
* Output: Hệ thống hiện trang thông báo đăng nhập thành công cho người dùng và chuyển về trang chủ của hệ thống.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| username | string | usename tài khoản do người dùng nhập vào để đăng nhập. |
| password | string | password tài khoản do người dùng nhập vào để đăng nhập. |
| ketqua | boolean | khi so sánh username, password do người dùng nhập vào, hàm sẽ trả về giá trị đăng nhập được hay không. |

Binding: BasicHttpBinding

* Người dùng sẽ nhập username và password vào form đăng nhập của hệ thống và chọn đăng nhập. Hệ thống sẽ thực hiện câu lệnh kiểm tra thông tin người dùng nhập có trùng với tài khoản thành viên trong CSDL chưa.Sau khi xét duyệt thành công, hệ thống sẽ chuyển tình trạng người dùng thành thành viên, thông báo đăng nhập thành công và chuyển thành viên về trang chủ của hệ thống.

# 2.2.13.Người dung đánh giá:

* Chức năng cho phép người dùng đã đăng ký và đăng nhập vào hệ thống gửi đánh giá(vote) của mình về các NH-CF hoặc trên đánh giá góp ý của các thành viên khác.
* Đặc tả hàm:
* Input: Danhgia(username, vote)
* Output: Hệ thống cập nhật những đánh giá của thành viên về các NH-KS hoặc trên đánh giá hay góp ý của các thành viên khác.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| username | string | username của thành viên thực hiên chức năng vote. |
| vote | boolean | chức năng vote có được chọn hay không. |
| vote | boolean | trả về 1 vote được cộng cho đối tượng được vote. |

Binding: BasicHttpBinding

* Thành viên khi xem các NH-CF mà họ biết qua và yêu thích có thể dùng chức năng vote cho các NH-CF này. Ngoài ra, thành viên có thể vote cho ý kiến của các thành viên khác nếu họ đồng tình với các ý kiến đó.Hệ thống sẽ cập nhật các đánh giá của thành viên đến các đối tượng được đánh giá.

# 2.2.14.Người dùng góp ý:

* Chức năng cho phép người dùng đã đăng ký và đăng nhập vào hệ thống gửi góp ý(commend) của mình về các NH-CF hoặc trên đánh giá góp ý của các thành viên khác.
* Đặc tả hàm:
* Input: Gopy(username, gopy)
* Output: Hệ thống cập nhật những góp ý của thành viên về các NH-KS hoặc trên đánh giá hay góp ý của các thành viên khác.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| username | string | username của thành viên thực hiên chức năng góp ý. |
| gopy | string | bài góp ý của thành viên. |
| gopy | string | trả về 1 bài góp ý được update cho đối tượng được góp ý. |

Binding: BasicHttpBinding

* Thành viên khi xem các NH-CF có thể dùng chức này đưa ra các góp ý của mình về các NH-CF này.Ngoài ra, thành viên có thể đóng góp ý kiến về những góp ý của các thành viên khác. Hệ thống sẽ duyệt các góp ý này,nếu hợp lệ các đóng góp này sẽ được cập nhật, ngược lại nếu đóng góp bậy bạ, sai đề tài, sai sự thật sẽ bị phạt từ xóa bài đến ban account.

# 2.2.15.Người dùng đăng bài giới thiệu:

* Chức năng cho phép người dùng có thể giơi thiệu 1 NH-CF mà mình thích đến các thành viên khác, hay các NH-CF muốn giới thiệu mình, qua việc post thêm hình ảnh, các thông tin về NH-CF đó, NH-CF được giới thiệu sẽ được đăng lên sau khi đã được duyệt. (loại các trường hợp viết bậy, thong tin sai sự thật,…)
* Đặc tả hàm:
* Input: Dangbai(username,dangbai)
* Output: Sau khi xét duyệt, hệ thống đăng tải những bài viết giới thiệu của thành viên về các NH-CF hoặc những bài quảng cáo của các NH-CF muốn quảng cáo họ.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| username | string | username của thành viên thực hiên chức năng đăng bài. |
| dangbai | NH-CF | bài đăng gồm tất cả các thông tin của NH-CF. |
| ketqua | NH-CF | trả về đối tượng NH-CF được insert vào CSDL nếu hệ thống duyệt thành công. |

Binding: BasicHttpBinding

* Thành viên có thể sử dụng chức năng đăng bài để đăng lên những bài viết và hình ảnh về các NH-CF họ yêu thích mà vẫn chưa có trên danh sách các NH-CF của hệ thống để giới thiệu các NH-CF này đến với các thành viên khác.Ngoài ra, các NH-CF có thể sử dụng chức năng này để đăng bài quảng cáo cho NH-CF của họ cũng như tạo thêm những sự lựa chọn khác cho người dùng.Các bài viết được đăng lên sẽ được xét duyệt, nếu như hợp lệ thì bài viết sẽ được đăng tải và NH-CF mới sẽ được cập nhật vào CSDL của hệ thống.Ngược lại nếu bài viết không hợp lệ, lạc đề, bậy bạ, sai sự thật thì người đăng sẽ bị phạt từ cảnh cáo, xóa bài viết đến ban account vĩnh viễn.

# 2.2.16.So sánh:

* Chức năng cho phép người dùng so sánh giữa cac NH-CF theo các thông tin mà họ cần so sánh để tìm ra NH-CF phù hợp với nhu cầu của họ nhất.
* Đặc tả hàm:
* Input:Sosanh(tennhahang, diachi, monan, thucuong, khonggian, trangtri, dichvu, phucvu, khuyenmai)
* Output: Hệ thống hiển thị cho người dùng trang so sánh các NH-CF có các thông tin theo nhu cầu so sánh của họ.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| tennhahang | string | tên nhà NH-CF được chọn để so sánh. |
| diachi | boolean | có so sánh theo tiêu chí địa chỉ hay không. |
| monan | boolean | có so sánh theo tiêu chí món ăn hay không. |
| thucuong | boolean | có so sánh theo tiêu chí thức uống hay không. |
| khonggian | boolean | có so sánh theo tiêu chí không gian hay không. |
| trangtri | boolean | có so sánh theo tiêu chí trang trí hay không. |
| dichvu | boolean | có so sánh theo tiêu chí dịch vụ hay không. |
| phucvu | boolean | có so sánh theo tiêu chí phục vụ hay không. |
| khuyenmai | boolean | có so sánh theo tiêu chí khuyến mãi hay không. |
| dsketqua | list<NH-CF> | danh sách các NH-CF được đem so sánh với các tiêu chỉ được chọn. |

Binding: BasicHttpBinding

* Khi xem các NH-CF họ chọn chức năng đưa các NH-CF đó vào danh sách các NH-CF cần so sánh. Sau khi chọn xong hết, người dùng chuyển sang trang so sánh và check vào các thông tin mà họ cần so sánh của các NH-CF đó. Sau đó, hệ thống sẽ truy suất vào CSDL của hệ thống để lấy các thông tin được check để so sánh của các NH-CF này và hiển thị trang so sánh có các NH-CF chỉ với các thông tin cần được so sánh đó. Nếu người dùng bỏ trống các ô check, hệ thống sẽ so sánh tất cả, không ràng buộc điều kiện với các thuộc tính đó.

# 2.2.17.Sắp xếp:

* Chức năng cho phép người dùng sắp xẹp lại các thong tin của NH-CF theo các tiêu chí như giá cả, đánh giá người dùng (rate) từ thấp đến cao hay cao đến thấp, hoặc sắp xếp món ăn thức uống, địa điểm, dịch vụ,.. theo nhóm phù hợp nhu cầu của người dùng.
* Đặc tả hàm:
* Input:Sapxep(tennhahang, diachi, monan, thucuong, khonggian, trangtri, dichvu, phucvu, khuyenmai)
* Output: Hệ thống hiển thị cho người dùng trang đã sắp xếp các NH-CF có các thông tin theo nhu cầu sắp xếp của họ.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| tennhahang | string | tên NH-CF được chọn để sắp xếp. |
| diachi | boolean | có sắp xếp theo tiêu chí địa chỉ hay không. |
| monan | boolean | có sắp xếp theo tiêu chí món ăn hay không. |
| thucuong | boolean | có sắp xếp theo tiêu chí thức uống hay không. |
| khonggian | boolean | có sắp xếp theo tiêu chí không gian hay không. |
| trangtri | boolean | có sắp xếp theo tiêu chí trang trí hay không. |
| dichvu | boolean | có sắp xếp theo tiêu chí dịch vụ hay không. |
| phucvu | boolean | có sắp xếp theo tiêu chí phục vụ hay không. |
| khuyenmai | boolean | có sắp xếp theo tiêu chí khuyến mãi hay không. |
| sapxep | string | phương thức sắp xếp do người dùng chọn. |
| dsketqua | list<NH-CF> | danh sách các NH-CF được đem sắp xếp với các tiêu chỉ được chọn. |

Binding: BasicHttpBinding

* Người dùng chuyển sang trang sắp xếp và check vào các thông tin mà họ cần sắp xếp lại của các NH-CF, sau đó chọn sắp xếp từ cao xuống thấp hay từ thấp đến cao hoặc theo nhóm. Sau đó, hệ thống sẽ truy suất vào CSDL của hệ thống để lấy các thông tin được check rồi sắp xếp lại các thông tin đó của các NH-CF theo yêu cầu của người dùng và hiển thị trang sắp xếp cho người dùng có các NH-CF với các thông tin đã được sắp xếp theo nhu cầu của người dùng. Nếu người dùng bỏ trống các ô check, hệ thống sẽ sắp xếp tất cả, không ràng buộc điều kiện với các thuộc tính đó.

# 2.2.18.Liên hệ NH-CF:

* Chức năng hỗ trợ người dùng lien hệ nhanh với NH-CF qua e-mail, chat, fax, hoặc điện thoại để hỏi đáp về các sản phẩm, dịch vụ,...
* Đặc tả hàm:
* Input:Lienhe(lienhe)
* Output: Hệ thống hiển thị cho người dùng trang liên hệ với các NH-CF tùy theo hình thức mà họ muốn liên hệ.

|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| lienhe | string | hình thức liên hệ |
| lienhe | string | hình thức liên hệ |

Binding: BasicHttpBinding

* Khi xem các NH-CF người dùng chọn chức năng liên hệ với các NH-CF đó qua việc chọn liên hệ qua e-mail, chat, điện thoại với nhân viên, hay gửi fax cho NH-CF đó. Sau khi chọn xong hết, người dùng chuyển sang trang liên hệ tùy thuộc vào hình thức mà người dùng muốn liên hệ với NH-CF đó để hỏi đáp về các sản phẩm, dịch vụ của NH-CF theo nhu cầu của họ.

# 2.2.19.Reservation:

* Chức năng hỗ trợ thành viên đã đăng nhập đặt chỗ nhanh tại NH\_CF theo nhu cầu của họ như giờ đặt bàn, vị trí ngồi..Server sẽ liên hệ với nhà hàng để trả lời họ rằng yêu cầu đặt bàn đó của họ có thực hiện được hay không và giúp họ đặt chỗ tại các NH-CF này.
* Đặc tả hàm:
* Input: Datcho(username, ban, thoigian, dichvu)
* Output: Hệ thống trả lời nhanh chóng cho thành viên rằng yêu cầu đặt chỗ của họ với các NH-CF mà họ muốn đặt chỗ có thực hiện được hay không.

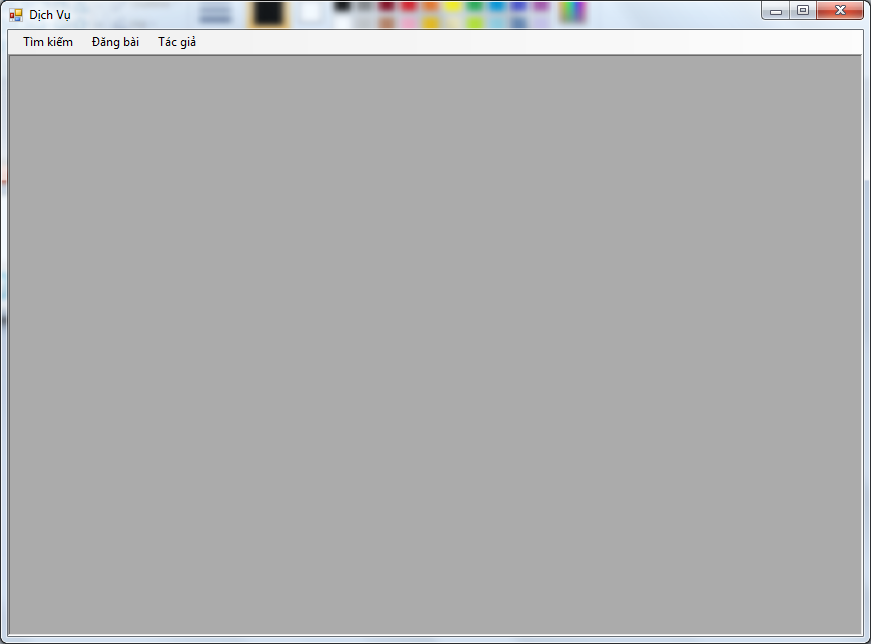
|  |  |  |
| --- | --- | --- |
| Các tham số | | |
| Tên | Kiểu dữ liệu | Mô tả |
| username | string | username của thành viên thực hiện chức năng đặt chỗ. |
| ban | string | vị trí bàn mà thành viên muốn đặt chỗ. |
| thoigian | datetime | thời gian thành viên sẽ check-in đến NH-CF đó. |
| dichvu | string | các dịch vụ khác mà thành viên yêu cầu khi đặt chỗ. |
| ketqua | boolean | kết quả việc đặt chỗ của thành viên đó với các yêu cầu trên có thực hiện được hay không. |

Binding: BasicHttpBinding

* Khi xem các NH-CF thành viên chọn chức năng đặt chỗ với với các NH-CF đó qua việc chọn chức năng đặt chỗ và điền thông tin về vị trí bàn, thời gian check-in, và dịch vụ thêm mà họ muốn. Hệ thống sẽ dựa vào những thông tin này và truy xuất ngay đến tình trạng bàn, dịch vụ của NH-CF tại thời điểm đó có đáp ứng được nhu cầu của khách hàng hay không. Nếu tất cả đều phù hợp, hệ thống sẽ gửi lại thông báo có thể đặt bàn và kèm theo các nội quy của NH-CF như cách tính phí phục vụ, phí đền bù nếu không đến hoặc đến trễ khi đã đặt chỗ, thông tin mà NH-CF cần thêm từ khách hàng như số PIN,… qua tin nhắn của thành viên đó và yêu cầu xác nhận yêu cầu đặt chỗ và điền thông tin thêm nếu có. Sau khi xác nhận yêu cầu đặt chỗ, hệ thống sẽ gửi các thông tin cần thiết của khách hàng đến NH-CF đó để hoàn tất thủ tục đặt chỗ, nếu thành công, NH-CF đó sẽ gửi thông báo thành công đến hệ thống và hệ thống chuyển thông báo đó về thành viên đó qua tin nhắn.

# 3.HIỆN THỰC CHƯƠNG TRÌNH

# 3.1.Giao diện chính của chương trình.



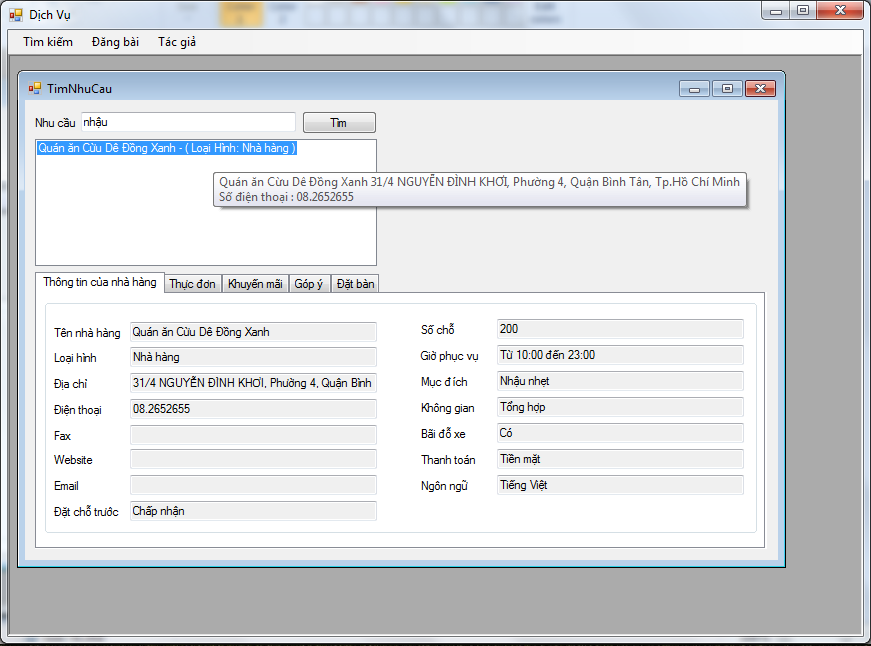
# C:\Users\Khin\Pictures\dangnhap.png3.2.Form đăng nhập vào hệ thống.

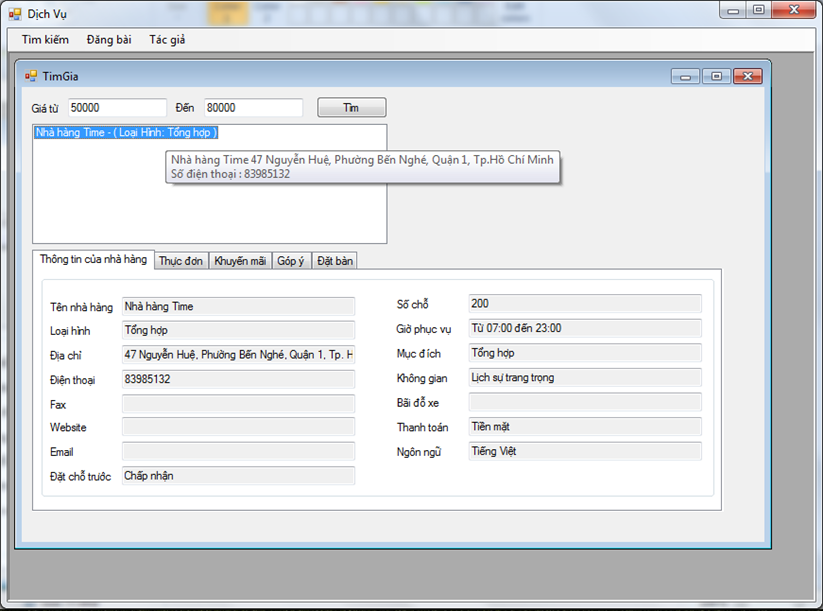
# C:\Users\Khin\Pictures\dangky.png3.3.Form đăng ký hệ thống.

# C:\Users\Khin\Pictures\timdiadiem1.png3.4.Tìm kiếm thông tin nhà hàng cà phê theo địa điểm.

# 3.5.Tìm kiếm thông tin nhà hàng cà phê theo không gian.

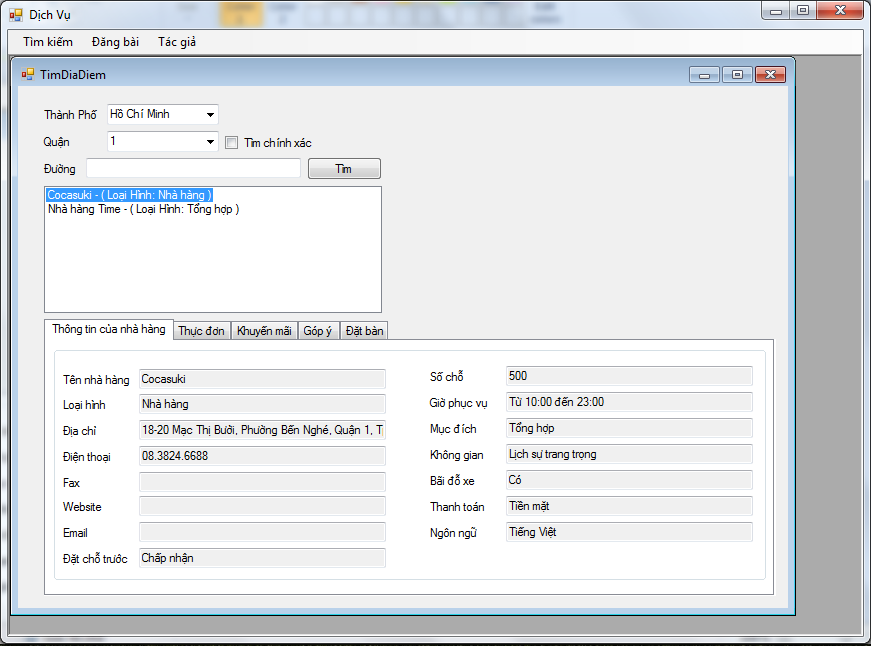
# C:\Users\Khin\Pictures\timmonan.pngC:\Users\Khin\Pictures\timkhonggian.png 3.6.Tìm kiếm thông tin nhà hàng cà phê theo tên món ăn.

**3.7.Tìm kiếm thông tin nhà hàng cà phê theo nhu cầu.**

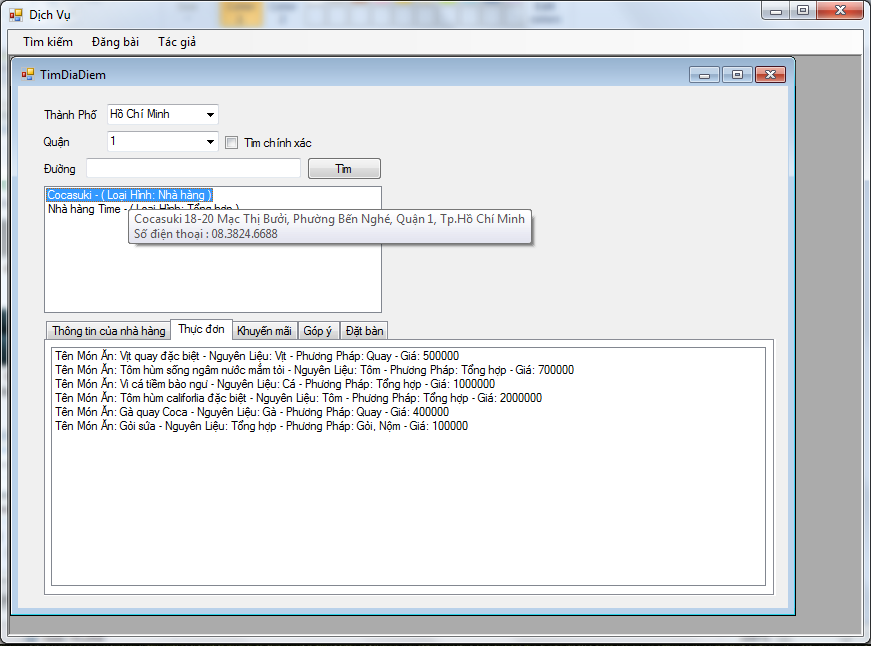
**3.8.Tìm kiếm thông tin nhà hàng cà phê theo giá.**

# 3.9.Tìm kiếm thông tin nhà hàng cà phê theo loại khuyến mãi.

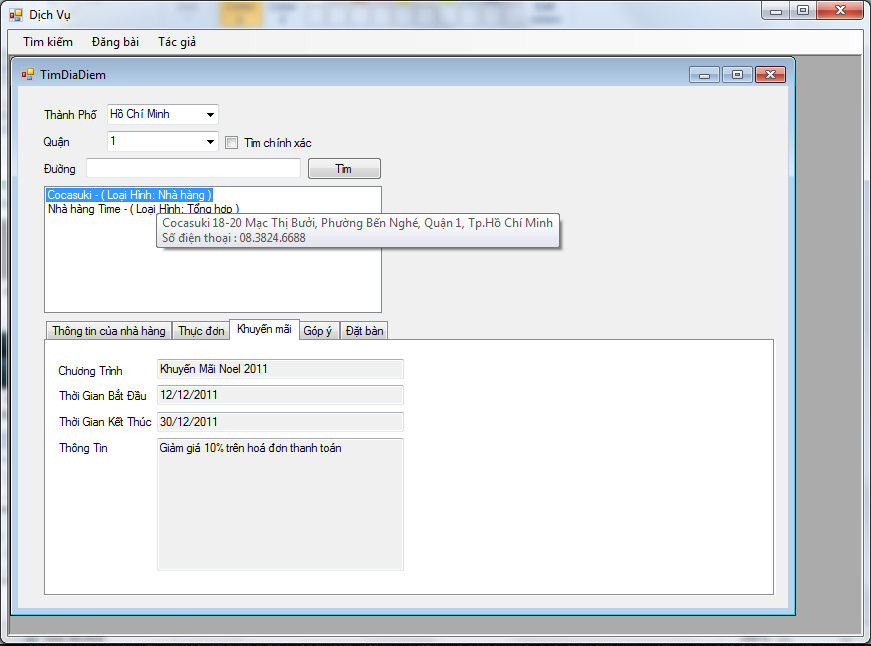
# C:\Users\Khin\Pictures\timkhuyenmai.png3.10.Form thể hiện thông tin nhà hàng.



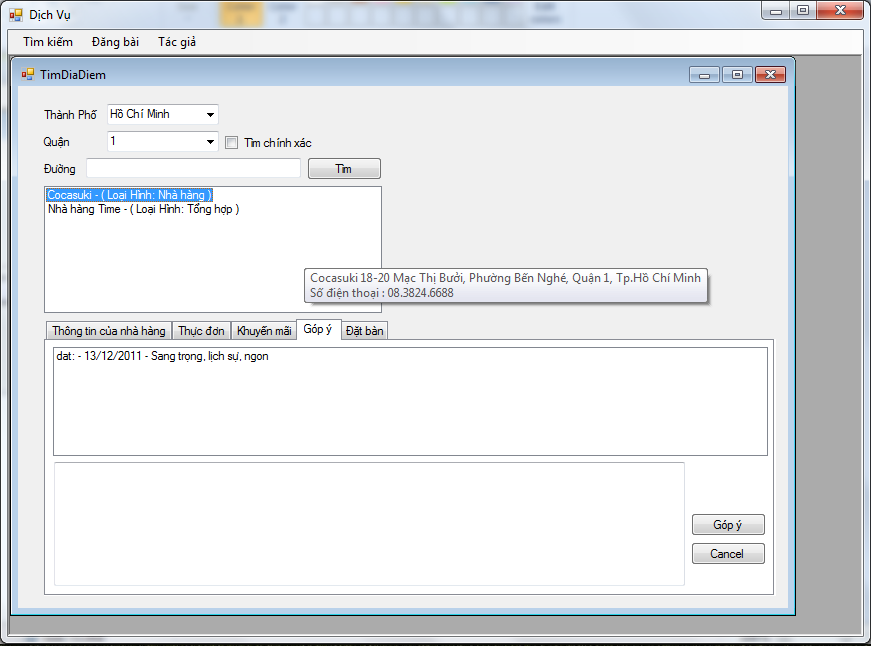
# 3.11.Form thể hiện thông tin thực đơn của nhà hàng.



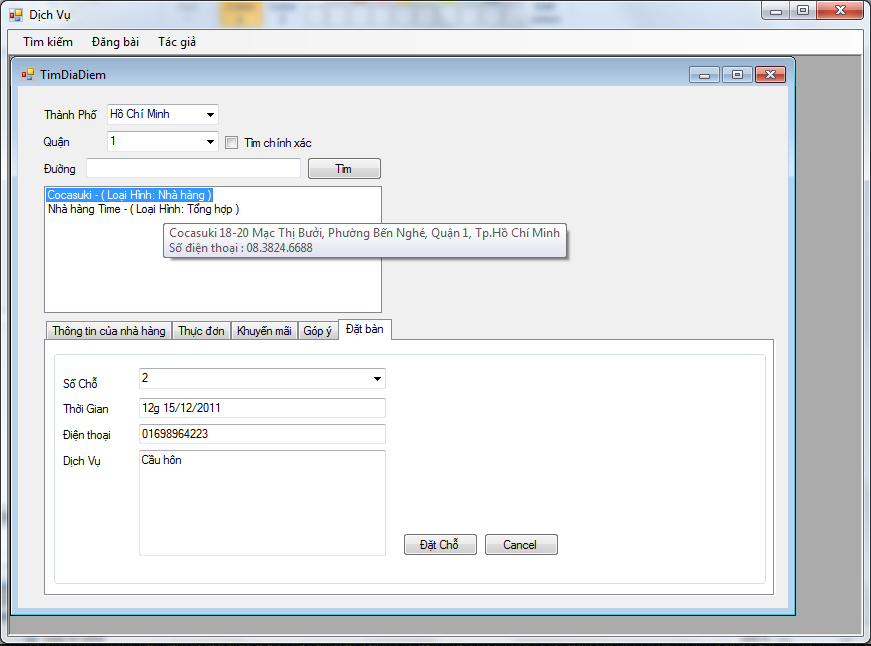
# 3.12.Form thể hiện thông tin khuyến mãi của nhà hàng.



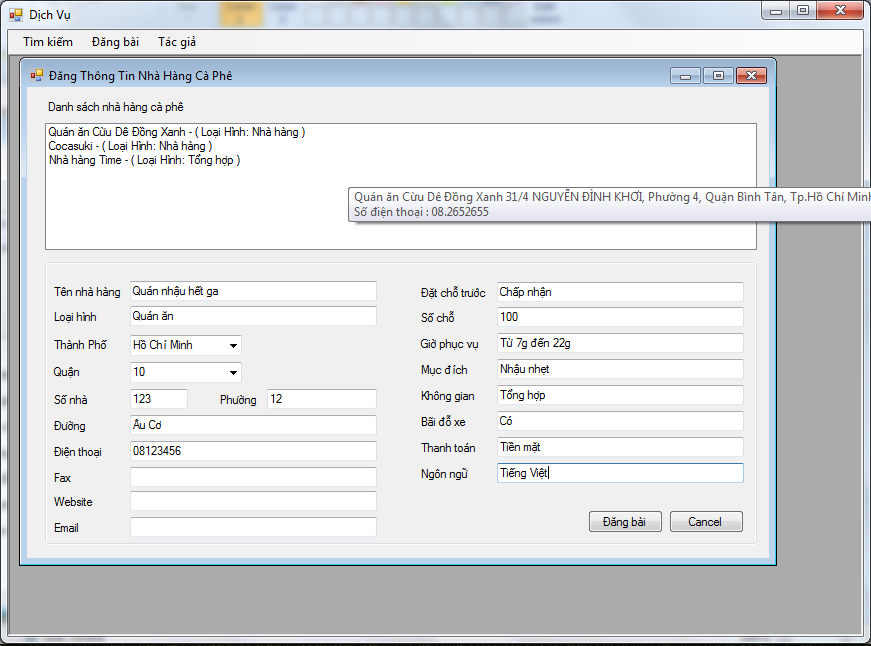
# 3.13.Form thể hiện thông tin góp ý của nhà hàng và đăng góp ý của người dùng.



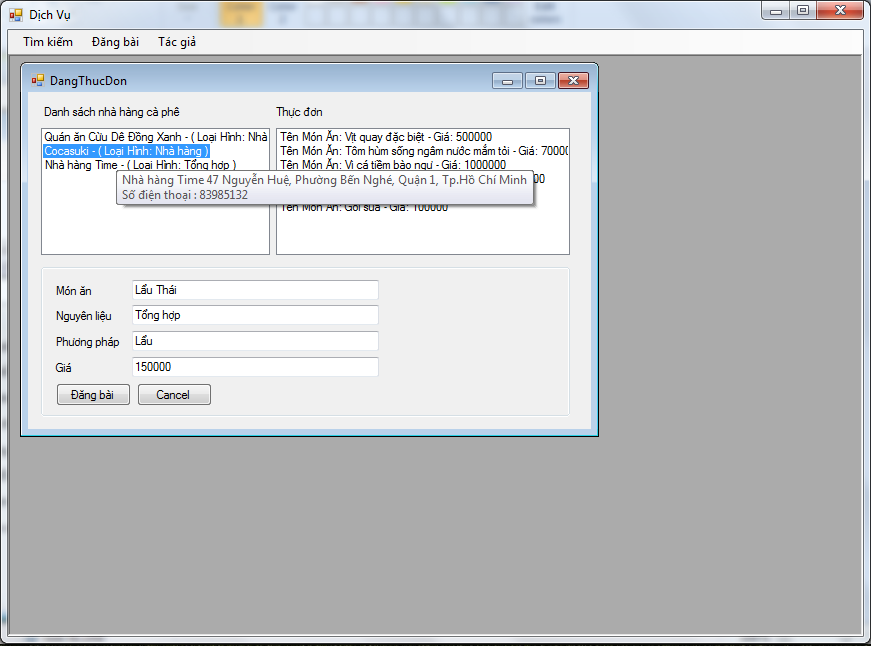
# 3.14.Form đặt bàn của người dùng với nhà hàng.



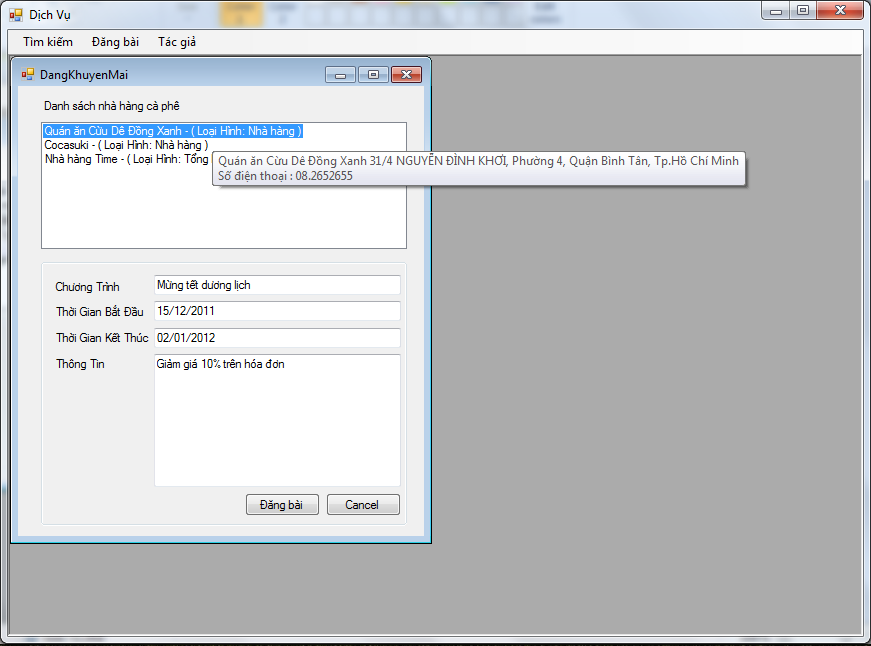
# 3.15.Form đăng thông tin nhà hàng cà phê.



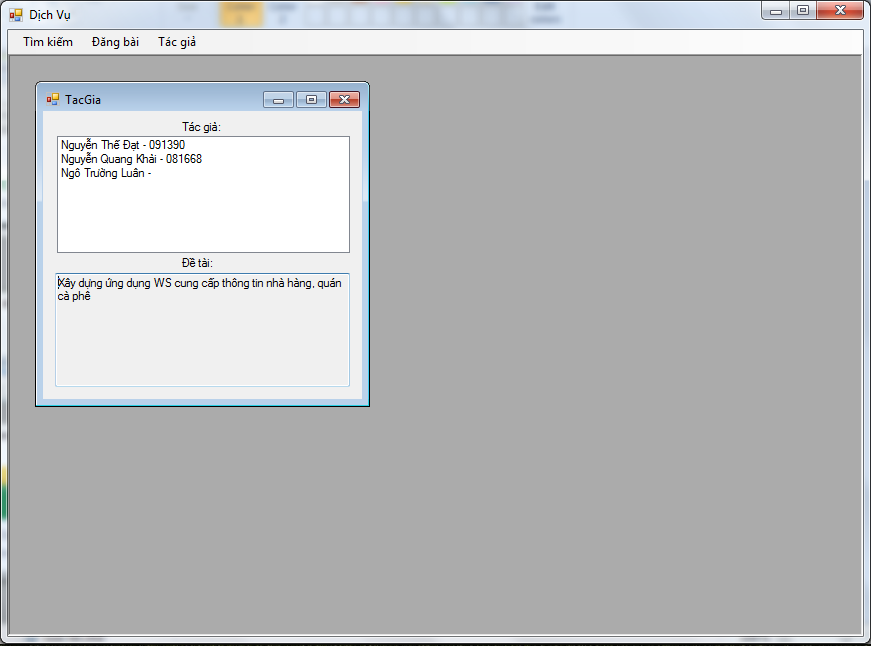
# 3.16.Form đăng thông tin thực đơn nhà hàng cà phê.



# 3.17.Form đăng thông tin khuyến mãi nhà hàng cà phê.



# 3.17.Form đăng thông tin tác giả.



# 4.Tham Khảo.

<http://www.codeproject.com/KB/WCF/WCFOverview.aspx>

<http://www.wcftutorial.net/WCF-Architecture.aspx>

<http://www.wcftutorial.net/EndPoint.aspx>