|  |
| --- |
| **TRƯỜNG ĐẠI HỌC HOA SEN**  **KHOA KHOA HỌC VÀ CÔNG NGHỆ**    **BÁO CÁO ĐỀ TÀI CÔNG NGHỆ DỊCH VỤ WEB**    **Giảng viên hướng dẫn :** Thầy Võ Hoàng Hải  **Sinh viên thực hiện :** Phạm Hoàng Sơn – 093576  Đặng Thị Lan Phương - 093568  **Lớp :** QL091A  **Thời gian :** 01/09/2011 – 20/12/2011    **Tháng 11/ 20117** |

**LỜI MỞ ĐẦU**

Hiện nay kho tàng văn học Việt Nam và thế giới rất rộng lớn với nhiều thể loại, chủ đề, nội dung và tác giả.... Thế nên việc tìm kiếm một tác phẩm của một nhà văn ta yêu thích, tìm tên tác giả của một tác phẩm mà ta yêu thích là một điều không dễ dàng.

Từ đó đòi hỏi cần có một chương trình giúp cho người yêu thơ văn dễ dàng tìm được một tác phẩm hay một tác giả, hay họ có thể dễ dàng tìm kiếm cho mình những tác phẩm hay tác giả mới.

Từ những yêu cầu nêu trên nhóm đã quyết định chọn đề tài ”**Xây dựng Web service cung cấp thông tin tác giả tác phẩm văn học**”.

Nhóm đã phân tích và đưa ra một số chức năng cơ bản của một web service cung cấp thông tin tác giả tác phẩm văn học như sau:

* Khách:
  + Tìm kiếm tiểu sử theo tên tác giả.
  + Tìm kiếm tên tác giả theo tiểu sử.
  + Tìm kiếm tên tác phẩm theo tên tác giả.
  + Tìm kiếm nội dung tác phẩm theo tên tác phẩm.
  + Tìm kiếm tên tác phẩm theo thể loại.
  + Tìm kiếm tên tác phẩm theo nội dung.
* Admin:
  + Thêm, xóa, sửa tác phẩm.
  + Thêm, xóa, sửa tác giả.

Danh sách thành viên thực hiện đề tài.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ tên | MSSV | Lớp | Email |
| 1 | Phạm Hoàng Sơn | 093576 | QL091A | [son.ph3576@sinhvien.hoasen.edu.vn](mailto:son.ph3576@sinhvien.hoasen.edu.vn) |
| 2 | Đặng Thị Lan Phương | 093568 | QL091A | [phuong.dtl3568@sinhvien.hoasen.edu.vn](mailto:phuong.dtl3568@sinhvien.hoasen.edu.vn) |

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 2](#_Toc311577951)

[1. Tìm hiểu lý thuyết 6](#_Toc311577952)

[1.1 Giới thiệu Web service & SOA 6](#_Toc311577953)

[1.1.1 Tổng quan Web service 6](#_Toc311577954)

[1.1.2 Tổng quan SOA 7](#_Toc311577955)

[1.2 Các nguyên tắc thiết kế 7](#_Toc311577956)

[1.2.1 Nhóm nguyên lý thực thi 7](#_Toc311577957)

[1.2.2 Nhóm nguyên lý điều chỉnh 7](#_Toc311577958)

[1.3 Standardized Service contract 7](#_Toc311577959)

[1.4 Service Reusability 8](#_Toc311577960)

[1.5 Service Autonomy 8](#_Toc311577961)

[1.6 Service Statelessness 9](#_Toc311577962)

[1.7 Service Discoverability 10](#_Toc311577963)

[1.8 Service Loose Coupling 11](#_Toc311577964)

[1.9 Service Abstraction 11](#_Toc311577965)

[1.10 Service Composability 12](#_Toc311577966)

[2. Thiết kế chương trình 14](#_Toc311577967)

[2.1 Đối tượng phục vụ 14](#_Toc311577968)

[2.1.1 Khách: 14](#_Toc311577969)

[2.1.2 Admin: 14](#_Toc311577970)

[2.2 Kiến trúc hệ thống 14](#_Toc311577971)

[2.3 Các chức năng chính của chương trình. 15](#_Toc311577972)

[2.3.1 Tìm kiếm tiểu sử theo tên tác giả 15](#_Toc311577973)

[2.3.2 Tìm kiếm tên tác giả theo tiểu sử 15](#_Toc311577974)

[2.3.3 Tìm kiếm tên tác phẩm theo tên tác giả 16](#_Toc311577975)

[2.3.4 Tìm kiếm nội dung tác phẩm theo tên tác phẩm 16](#_Toc311577976)

[2.3.5 Tìm kiếm tên tác phẩm theo thể loại. 16](#_Toc311577977)

[2.3.6 Tìm kiếm tên tác phẩm theo nội dung 17](#_Toc311577978)

[2.3.7 Đăng nhập vào hệ thống 17](#_Toc311577979)

[2.3.8 Thêm thông tin tác giả 18](#_Toc311577980)

[2.3.9 Cập nhật thông tin tác giả 18](#_Toc311577981)

[2.3.10 Xóa thông tin tác giả 19](#_Toc311577982)

[2.3.11 Thêm thông tin tác phẩm 19](#_Toc311577983)

[2.3.12 Cập nhật thông tin tác phẩm 20](#_Toc311577984)

[2.3.13 Xóa thông tin tác phẩm 20](#_Toc311577985)

[2.3.14 Đối tượng chứa thông tin tác giả, tác phẩm 20](#_Toc311577986)

[2.3.15 Đối tượng chứa thông tin đăng nhập 22](#_Toc311577987)

[2.4 Mô tả cơ sở dữ liệu 23](#_Toc311577988)

[2.4.1 Bảng tác giả 23](#_Toc311577989)

[2.4.2 Bảng tác phẩm 23](#_Toc311577990)

[2.4.3 Bảng user 23](#_Toc311577991)

[3. Hiện thực 24](#_Toc311577992)

[3.1 Các màn hình chính của chương trình. 24](#_Toc311577993)

[3.1.1 Tìm kiếm tiểu sử theo tên tác giả. 24](#_Toc311577994)

[3.1.2 Tìm kiếm tên tác giả theo tiểu sử. 24](#_Toc311577995)

[3.1.3 Tìm kiếm tên tác phẩm theo tên tác giả. 25](#_Toc311577996)

[3.1.4 Tìm kiếm nội dung tác phẩm theo tên tác phẩm 25](#_Toc311577997)

[3.1.5 Tìm kiếm tên tác phẩm theo thể loại. 26](#_Toc311577998)

[3.1.6 Tìm kiếm tên tác phẩm theo nội dung. 26](#_Toc311577999)

[3.2 Cách cài đặt và chạy chương trình 27](#_Toc311578000)

[3.2.1 Cài đặt cơ sở dữ liệu 27](#_Toc311578001)

[3.2.2 Cài đặt hosting 27](#_Toc311578002)

[3.2.3 Chạy chương trình phía Client 27](#_Toc311578003)

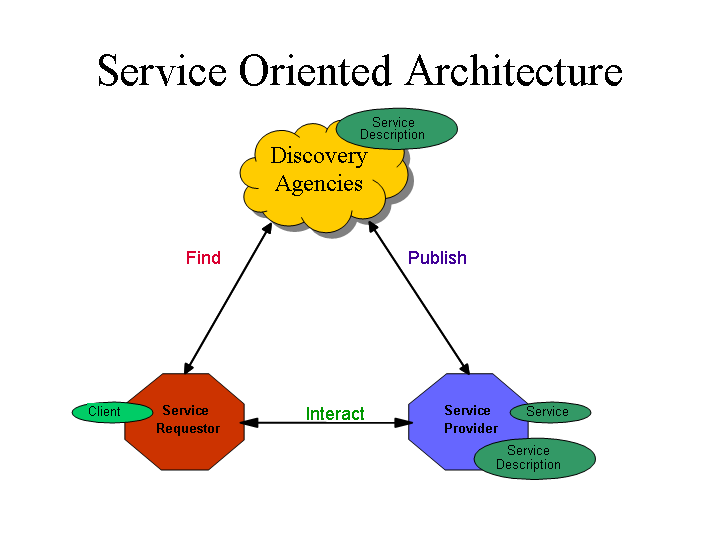
[4. Kết luận 28](#_Toc311578004)

[5. Tài liệu tham khảo 30](#_Toc311578005)

NỘI DUNG

1. Tìm hiểu lý thuyết
   1. Giới thiệu Web service & SOA
      1. Tổng quan Web service

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



Hình 1.1: Kiến trúc của Web service

Web service gồm 3 thành phần chính:

* SOAP: là giao thức truyền dữ liệu được xây dựng dựa trên XML, HTTP, và SMTP.
* WSDL: là các lược đồ XML được dùng để mô tả các chức năng, giao diện (interface) của 1 web service.
* UDDI: là tổ chức quản lý các services, nơi có sự tham gia của các tổ chức như IBM, Microsoft, ... tham gia cung cấp các dịch vụ.
  + 1. Tổng quan SOA

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

* 1. Các nguyên tắc thiết kế

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

* + 1. Nhóm nguyên lý thực thi
* Standardized Service Contract: dịch vụ đưa ra một thỏa thuận thông tin liên lạc, theo quy định chung bởi một hoặc nhiều mô tả dịch vụ và tài liệu liên quan.
* Service Reusability: các dịch vụ sẽ được chi thành các dịch vụ con với mục đích cho việc tái sử dụng các service.
* Service Autonomy: các dịch vụ sẽ có các quyền tự chủ.
* Service Statelessness: giảm việc lưu trữ các thông tin riêng.
* Service Discoverability: được thiết kế theo hướng mô tả bên ngoài cho phép người dùng tìm kiếm và sử dụng dựa trên các cơ chế khám phá có sẵng.
  + 1. Nhóm nguyên lý điều chỉnh
* Service Loose Coupling: giảm thiểu sự phụ thuộc giữa các service với nhau.
* Service Abstraction: che giấu thông tin logic của service khi hiển thị thông tin ra bên ngoài.
* Service Composability: tập hợp các service có thể phối hợp và ghép lại với nhau.
  1. Standardized Service contract

Các Service contract thông thường bao gồm mô tả kỹ thuật và phi kỹ thuật. Các tài liệu thường được miêu tả như một dạng giao diện và được biết như API. Do đó các service trong kho dịch vụ phải tuân theo các chuẩn thiết kế contract giống nhau.

Có 3 loại chuẩn hóa Service contract:

* *Chuẩn hóa chức năng (Functional Expression Standardization):* Các operation của service khi định nghĩa cần phải áp dụng các qui tắc đặt tên chuẩn, các qui tắc này được áp dụng cho các input message, output message và tên phải phù hợp với ngữ nghĩa của chúng. Điều này nhành giúp diễn giải service contract chính xác hơn, tăng khả năng liên kết nội tại và tính sử dụng lại của service, ngoài ra còn giảm khả năng trùng lắp dữ liệu.
* *Chuẩn hóa mô hình dữ liệu (Data Model Standardization*): yêu cầu phát triển các mô hình dữ liệu phải được chuẩn hóa để tạo kiến trúc dữ liệu được chuẩn hóa nhằm tăng khả năng sử dụng lại các service trong hệ thống, nhưng điều này không yêu cầu chuẩn hóa toàn bộ data model toàn hệ thống.
* *Chuẩn hóa chính sách (Policy Standardization):* các policy mô tả giới hạn sử dụng các service, để một service có thể sử dụng lại thì các behavioral requirement cần được trình bày theo phong cách phù hợp với việc sử dụng các policy expression được chuẩn hóa dựa trên các từ vựng chuẩn và phải phù hợp với những đặc điểm của ngôn ngữ.
  1. Service Reusability
* Là khả năng sử dụng lại của service.
* Mục tiêu: giúp tăng khả năng thu hồi vốn đầu tư ban đầu, tăng tính linh hoạt bằng cách thực hiện nhanh chóng các yêu cầu nghiệp vụ trong tương lai của doanh nghiệp.
* Đặc điểm thiết kế:
  + Service được xác định bởi các thuyết luôn luôn đúng của dịch vụ.
  + Service có tính dùng chung cao.
  + Service contract của service có tính dùng chung và có thể mở rộng.
  + Các service logic có thể sử dụng đồng thời.
  1. Service Autonomy

Autonomy thể hiện khả năng tự điều chỉnh, được thể hiện qua sự tự do điều khiển đưa ra quyết định cho bản thân mà không cần sự chấp thuận từ phía bên ngoài dịch vụ.

Để tăng tính tự trị đòi hỏi sự thực thi chương trình phải được cô lập hơn vì vậy sẽ làm tăng mức độ độc lập, kết quả là giúp chương trình tăng độ tin cậy và khả năng dự đoán.

Mục tiêu:

* Tăng quyền điều khiển của 1 service trên môi trường thực thi của nó.
* Tăng độ tin cậy, khả năng dự đoán đặc biệt là được kết hợp và sử dụng lại.

Có 2 loại service autonomy:

* *Runtime autonomy:* mức độ điều khiển của một service trong quá trình xử lý trong thời điểm mà service được gọi và thực thi.
* *Design-time autonomy:* mức độ tự do của chủ service để tạo sự thay đổi đối với 1 service trong thời gian tồn tại của nó.

Đo lường khả năng tự trị của service:

* *Service contract autonomy*: phạm vi khả năng của 1 contract không được trùng lắp với những contract khác. Điều này dễ dàng đạt được khi service contract được chuẩn hóa và tách rời với môi trường thực thi.
* *Shared autonomy.*
* *Service logic autonomy:* là mức độ phổ biết nhất có thể đạt được khi phát triển tùy chỉnh cho một service mới.
* *Pure autonomy.*
  1. Service Statelessness

Khi sự kết hợp các service ngày càng phức tạp thì số lượng dữ liệu hoạt động cần được quản lý trong suốt thời gian sống của composition. Đề tăng khả năng mở rộng và sự thực thi trong kho service thì các service và kiến trúc cần được thiết kế hỗ trợ sự ủy nhiệm và trì hoãn các việc quản lý trạng thái.

*Mục tiêu:* tăng khả năng mở rộng service, hỗ trợ việc thiết kế agnostic service logic và cải tiến khả năng sử dụng lại service.

*Nguyên lý thiết kế:* tùy thuộc vào service model và cách tiếp cận trì hoãn được dùng mà có các loại đặc điểm thiết kế khác nhau được thực hiện.

Các loại state:

* *Active state:* qui định của service sau khi nó được gọi và thực thi bởi một chương trình sử dụng.
* *Passive state:* một service được load vào bộ nhớ nhưng nó không thực hiện bất kỳ chức năng nào của nó một cách tích cực.
* *Stateful:* service được active và sử dụng thêm vùng nhớ tạm để lưu trữ tạm thời và xử lý state data. Stateful gồm 3 loại: session data, context data, business data.
* *Stateless:* service được active và không xử dụng thêm vùng nhớ tạm để xử lý.
  1. Service Discoverability

Khả năng khám phá, giải thích và trình bày service.

Điều này sẽ giúp cho xác định service đã tồn tại hay cần phải xây dựng mới. Để làm được điều này đòi hỏi thông tin mô tả service phải chính xác thể hiện đầy đủ khả năng của service, nếu mô tả không đầy đủ thì người dùng sẽ mất cơ hội sử dụng lại service điều này dẫn đến sự trùng lắp với các service đã tồn tại.

*Mục tiêu:* các service được đặt ở vị trí như các resource có khả năng khám phá cao, mục đích và khả năng của service được trình bày rõ ràng bởi con người cũng như chương trình phần mềm.

Đặc điểm thiết kế:

* Trang bị dữ liệu phù hợp.
* Trang bị thêm những thông tin mô tả với mục đích sử dụng của chúng và những khả năng với con người.
* Nếu một service đã được đăng ký thì các bản ghi đăng ký được phổ biến là những meta information.
* Nếu một service không được đăng ký thì các tài liệu sơ lược về service được thiết kế bổ sung cho Service contract và đó là dạng cơ bản cho các bản ghi đăng ký trong tương lai.
* Khả năng khám phá và trình bày dịch vụ.
* Cải thiện sự truyền chất lượng thông tin của dữ liệu mô tả dịch vụ.

Những loại khám phá:

* *Design-time discovery:* tiến trình khám phá được thực hiện một cách thủ công bởi con người.
* *Runtime discovery:* service có khả năng đưa ra những truy vấn discovery động và kết quả là một quá trình tự động hóa*.*
  1. Service Loose Coupling

Service loose coupling nhằm giảm sự phụ thuộc vào các service khác mà bản thân service sử dụng, cũng như giảm sự phụ thuộc vào chương trình sử dụng bản thân service.

Mục tiêu: giảm sự phụ thuộc giữa các service, tăng sự thực thi độc lập.

Đặc điểm thiết kế: sự tồn tại của một service contract phải tách rời về công nghệ và sự thực thi. Không phụ thuộc vào xử lý bên ngoài, giảm tối thiểu các yêu cầu có sự gắn kết với consumer.

Các loại service contract coupling:

* *Logic to contact coupling:* phạm vi của 1 service được giới hạn bởi contract của nó, đây là dạng liên kết mang tính tích cự.
* *Contract to logic coupling:* phạm vi của 1 service phụ thuộc vào logic bên dưới, đây là loại liên kết mang tính tích cực vì các chương trình service consumer tạo ra sụ ràng buộc với service contract.
* *Contract to technology coupling:* là sự phụ thuộc vào công nghệ giao tiếp độc quyền. Loại liên kết này không mang tính tích cực vì khi service phụ thuộc vào công nghệ sẽ dẫn đến consumer cũng sẽ phụ thuộc vào công nghệ.
* *Contract to implement coupling*: là sự phụ thuộc vào môi trường thực thi, đây cũng là loại liên kết không mang tính tích cực.
* *Contract to functional coupling*: là sự phụ thuộc của service đối với external logic.
  1. Service Abstraction

Các service contract chỉ chứa các thông tin cần thiết và các thông tin này sẽ giới hạn những thông tin gì sẽ được publish trong các service contract.

*Mục tiêu*: giữ cho số lượng và nội dung thông tin súc tích, cân bằng và ngăn ngừa sự truy cập không cho phép.

*Đặc điểm thiết kế:* thông tin sẽ được trừu tượng về công nghệ, logic vác các chức năng bên ngoài dịch vụ, thông tin về một service sẽ được điều khiển hoặc ẩn trong một môi trường cụ thể.

Có 4 loại Meta abstraction:

* *Technology information:* mô tả thực thi về kỹ thuật bên dưới. Thông tin chi tiết về kỹ thuật sẽ được ẩn đi để tạo sự tự do về thay đổi kỹ thuật mà không ảnh hưởng đến người sử dụng service.
* *Function information:* metadata sẽ mô tả service có những chức năng nào.
* *Programatic logic information:* mô tả cách thức dịch vụ thực hiện khả năng của nó. Được áp dụng ở những mức thiết kế chi tiết thấp như thuật toán, xử lý lỗi, ...
* *Quality of service information:* metadata mô tả service behavior, các giới hạn và yêu cầu tương tác. Chất lượng của service data là một giới hạn của vùng thông tin liên quan đến độ tin cậy.
  1. Service Composability

Khả năng kết hợp các service lại với nhau.

*Mục tiêu:* tất cả các mục đích của service reusability có thể áp dụng, và là một dạng sử dụng lại dịch vụ.

Các khái niệm và thuật ngữ tích hợp:

* Composition.
* Composition Instance.
* Composition member.
* Composition controller.
* Composition sub-controller.
* Composition member capability.
* Composition controller capability.
* Designated controller.
* Collective composability.
* Service activity.
* Point-to-point.
* Primitive composition.
* Complex composition.
* Composition initiator.

1. Thiết kế chương trình
   1. Đối tượng phục vụ
      1. Khách:

Đối tượng phục vụ này khá đa dạng, đa phần là những người thường xuyên tìm hiểu về văn học, mà đặc biệt là về các tác giả và tác phẩm văn học. (Ví dụ như: giáo viên, học sinh, sinh viên, nhà văn, ...)

* + 1. Admin:

Là những người quản lý chương trình thường dùng để kiểm tra chương trình hay để cập nhật lại cơ sở dữ liệu của chương trình.

* 1. Kiến trúc hệ thống

Hình 2.1: Kiến trúc hệ thống

Hệ thống chia làm 3 phần chính: Client application, WCF Service Host và WCF Service Library.

* Client applicaiton: là ứng dụng phía người sử dụng, có thể là windows application hoặc web applicaion.
* WCF Service Host: chứa các file cấu hình kết nối đến các Service Library.
* WCF Service Library: là hệ thống chính của WS cung cấp thông tin tạm trú tạm vắng, các chức năng của hệ thống sẽ được mô tả tại đây.
  1. Các chức năng chính của chương trình.
     1. Tìm kiếm tiểu sử theo tên tác giả

Chương trình cho phép khách có thể tìm kiếm thông tin tiểu sử tác giả theo tên. Khi người dùng cung cấp thông tin là họ tên của một tác giả cụ thể, chương trình sẽ cho biết được tiểu sử của tác giả đó như thế nào. Ngoài ra hệ thống còn cung cấp tên các tác phẩm tiêu biểu của tác giả đó.

**public string GetTSTGbyTG(string tacgia)**

* Tham số truyền vào: tacgia
* Kết quả trả về: dữ liệu dạng chuỗi
* [OperationContract(Name="GetTSTGbyTG")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacgia | nvarchar(128) | Tên tác giả cần tìm kiếm  Ví dụ: Tố Hữu |

Table 2.1: Tìm kiếm tiểu sử theo tên tác giả

* + 1. Tìm kiếm tên tác giả theo tiểu sử

Hệ thống cho phép người dùng tìm kiếm tên tác giả theo tiểu sử. Khi người sử dụng chương trình cung cấp thông tiểu sử của tác giả, hệ thống sẽ trả về tất cả tên các tác giả có tiểu sử phù hợp.

**public string GetTSTGbyND(string noidung)**

* Tham số truyền vào: noidung
* Kết quả trả về: dữ liệu dạng chuỗi
* [OperationContract(Name="GetTSTGbyND ")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| noidung | nvarchar(128) | 1 phần nội dung tiểu sử tác giả cần tìm kiếm  Ví dụ: sinh năm 1920, tỉnh Thừa Thiên Huế... |

Table 2.2: Tìm kiếm tên tác giả theo tiểu sử

* + 1. Tìm kiếm tên tác phẩm theo tên tác giả

Hệ thống cho phép tìm kiếm tên các tác phẩm theo tên tác giả. Người sử dụng chỉ cần nhập vào tên tác giả.

**public string GetTPbyTG(string tacgia)**

* Tham số truyền vào: tacgia
* Kết quả trả về: dữ liệu dạng chuỗi
* [OperationContract(Name="GetTPbyTG ")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacgia | nvarchar(128) | Tên tác giả cần tìm kiếm  Ví dụ: Tố Hữu |

Table 2.3: Tìm kiếm tên tác phẩm theo tên tác giả

* + 1. Tìm kiếm nội dung tác phẩm theo tên tác phẩm

Hệ thống cho phép tìm kiếm nội dung tác phẩm theo tên tác phẩm. Người sử dụng chỉ cần nhập vào tên tác phẩm.

**public string GetTPbyTP(string tacpham)**

* Tham số truyền vào: tacpham
* Kết quả trả về: dữ liệu dạng chuỗi
* [OperationContract(Name="GetTPbyTP ")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacpham | nvarchar(128) | Tên tác phẩm cần tìm kiếm  Ví dụ: Từ ấy |

Table 2.4: Tìm kiếm nội dung tác phẩm theo tên tác phẩm

* + 1. Tìm kiếm tên tác phẩm theo thể loại.

Hệ thống cho phép người sử dụng tìm kiếm tên tác phẩm theo thể loại. Người dùng chỉ cần nhập thể loại tác phẩm muốn tìm. Hệ thống sẽ trả về danh sách tên tác phẩm có thể loại phù hợp.

**public string GetTPbyTL(string theloai)**

* Tham số truyền vào: theloai
* Kết quả trả về: dữ liệu dạng chuỗi
* [OperationContract(Name="GetTPbyTL ")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| theloai | nvarchar(128) | Thể loại tác phẩm cần tìm  Ví dụ: truyện ngắn |

Table 2.5: Tìm kiếm tên tác phẩm theo thể loại

* + 1. Tìm kiếm tên tác phẩm theo nội dung

Hệ thống cho phép người sử dụng tìm kiếm tên tác phẩm theo nội dung của tác phẩm đó. Người dùng chỉ cần nhập 1 phần nội dung của tác phẩm muốn tìm. Hệ thống sẽ trả về tên tác phẩm có nội dung phù hợp.

**public string GetTPbyND(string noidung)**

* Tham số truyền vào: **noidung**
* Kết quả trả về: dữ liệu dạng chuỗi
* [OperationContract(Name="GetTPbyND ")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| noidung | nvarchar(128) | Nội dung tác phẩm cần tìm.  Ví dụ: Từ ấy trong tôi bừng nắng hạ. Mặt trời chân lí chói qua tim |

Table 2.6: Tìm kiếm tên tác phẩm theo nội dung

* + 1. Đăng nhập vào hệ thống

Hệ thống cung cấp chức năng đăng nhập, điều này bắt buộc khi người sử dụng chương trình cần thêm mới, cập nhật hoặc xóa dữ liệu.

**int Login(string username, string password)**

* Tham số truyền vào: username, password
* Kết quả trả về: kiểu dữ liệu int. 1 là đăng nhập thành công, 0 là đăng nhập thất bại.
* [OperationContract(Name = "Login")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| username | nvarchar(10) | Tài khoản đăng nhập |
| Password | nchar(10) | Mật khẩu đăng nhập |
| Result | Int | Kết quả đăng nhập, thành công hoặc thất bại. |

Table 2.12: đăng nhập vào hệ thống

* + 1. Thêm thông tin tác giả

Hệ thống cho phép người quản lý chương trình (admin) thêm mới các tác giả.

**void ThemTG (string tacgia, string noidung)**

* Tham số truyền vào: tacgia, noidung
* Kết quả trả về: không có.
* [OperationContract(Name = "ThemTG ", IsOneWay = true)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacgia | nvarchar(128) | Tên tác giả muốn thêm |
| noidung | nvarchar(128) | Tiểu sử tác giả muốn thêm |

Table 2.13: thêm thông tin tác giả

* + 1. Cập nhật thông tin tác giả

Hệ thống cung cấp chức năng cập nhật thông tin tác giả. Cho phép người quản lý (admin) cập nhật các thông tin của tác giả.

**void UpdateTG (string tacgia, string noidung)**

* Tham số truyền vào: : tacgia, noidung
* Kết quả trả về: không có
* [OperationContract(Name = "UpdateTG", IsOneWay = true)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacgia | nvarchar(128) | Tên tác giả muốn chỉnh sửa |
| noidung | nvarchar(128) | Tiểu sử tác giả muốn chỉnh sửa |

Table 2.14: Cập nhật thông tin tác giả

* + 1. Xóa thông tin tác giả

Hệ thống cung cấp chức năng xóa thông tin tác giả. Cho phép người quản lý (admin) xóa các thông tin của tác giả.

**void XoaTG (string tacgia)**

* Tham số truyền vào: tacgia
* Kết quả trả về: không có
* [OperationContract(Name = "XoaTG", IsOneWay = true)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacgia | nvarchar(128) | Tên tác giả muốn xóa |

Table 2.15: Xóa thông tin tác giả

* + 1. Thêm thông tin tác phẩm

Hệ thống cho phép người quản lý chương trình (admin) thêm mới các tác phẩm.

**void ThemTP (string tacpham, string noidung, string theloai)**

* Tham số truyền vào: tacpham, noidung, theloai
* Kết quả trả về: không có.
* [OperationContract(Name = "ThemTP ", IsOneWay = true)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacpham | nvarchar(128) | Tên tác phẩm muốn thêm |
| noidung | nvarchar(128) | Nội dung tác phẩm muốn thêm |
| theloai | nvarchar(128) | Thể loại tác phẩm muốn thêm |

Table 2.13: thêm thông tin tác phẩm

* + 1. Cập nhật thông tin tác phẩm

Hệ thống cung cấp chức năng cập nhật thông tin tác phẩm. Cho phép người quản lý (admin) cập nhật các thông tin của tác phẩm.

**void UpdateTP (string tacpham, string noidung, string theloai)**

* Tham số truyền vào: tacpham, noidung, theloai
* Kết quả trả về: không có
* [OperationContract(Name = "UpdateTP", IsOneWay = true)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacpham | nvarchar(128) | Tên tác phẩm muốn chỉnh sửa |
| noidung | nvarchar(128) | Tiểu sử tác phẩm muốn chỉnh sửa |
| theloai | nvarchar(128) | Thể loại tác phẩm muốn chỉnh sửa |

Table 2.14: Cập nhật thông tin tác phẩm

* + 1. Xóa thông tin tác phẩm

Hệ thống cung cấp chức năng xóa thông tin tác phẩm. Cho phép người quản lý (admin) xóa các thông tin của tác phẩm.

**void XoaTP (string tacpham)**

* Tham số truyền vào: tacpham
* Kết quả trả về: không có
* [OperationContract(Name = "XoaTP", IsOneWay = true)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| tacpham | nvarchar(128) | Tên tác phẩm muốn xóa |

Table 2.15: Xóa thông tin tác phẩm

* + 1. Đối tượng chứa thông tin tác giả, tác phẩm

[ServiceContract (SessionMode = SessionMode.Required)]

public interface IService1

{

[OperationContract]

int Login (string username,string password);

[OperationContract]

User GetQuyen ();

[OperationContract]

string GetTSTGbyTG (string tacgia);

[OperationContract]

string GetTSTGbyND (string tacgia);

[OperationContract]

string GetTPbyTG (string tacgia);

[OperationContract]

string GetTPbyTP (string tacpham);

[OperationContract]

string GetTPbyTL (string tacpham);

[OperationContract]

string GetTPbyND (string tacpham);

[OperationContract]

string ThemTG (string tentg, string tieusu);

[OperationContract]

string ThemTP(string tentp, string theloai, string noidung, string tentg);

[OperationContract]

string UpdateTG (string tentg, string tieusu);

[OperationContract]

string UpdateTP (string tentp, string theloai, string noidung, string tentg);

[OperationContract]

string XoaTG (string tentg, string tieusu);

[OperationContract]

string XoaTP (string tentp, string theloai, string noidung, string tentg);

[OperationContract]

void Logout();

}

* + 1. Đối tượng chứa thông tin đăng nhập

[DataContract]

public class User

{

[DataMember]

public int Quyen =-1;

[DataMember]

public string username = "";

}

* 1. Mô tả cơ sở dữ liệu
     1. Bảng tác giả

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| Id | bigint | No | Khóa chính |
| Ten | nvarchar(50) | Yes | Họ tên tác giả |
| NoiDung | nvarchar(max) | Yes | Tiểu sử tác giả |

* + 1. Bảng tác phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| Id | bigint | No | Khóa chính |
| Ten | nvarchar(50) | Yes | Tên tác phẩm |
| TheLoai | nvarchar(50) | Yes | Thể loại tác phẩm |
| NoiDung | nvarchar(max) | Yes | Nội dung tác phẩm |
| TenTacGia | nvarchar(50) | Yes | Tên tác giả của tác phẩm |

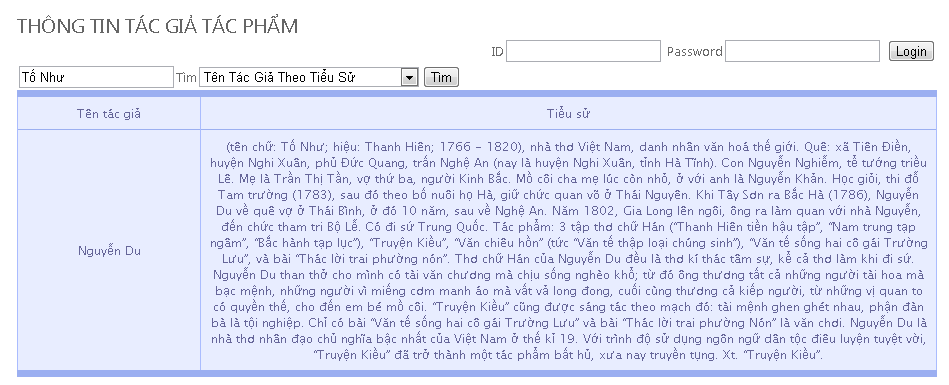
* + 1. Bảng user

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| Username | nvarchar(50) | No | Tài khoản đăng nhập |
| Password | nvarchar(50) | No | Mật khẩu đăng nhập |
| HoTen | nvarchar(50) | Yes | Họ tên người dùng |
| Quyen | int | No | Quyền sử dụng của người dùng |

1. Hiện thực
   1. Các màn hình chính của chương trình.
      1.  Tìm kiếm tiểu sử theo tên tác giả.

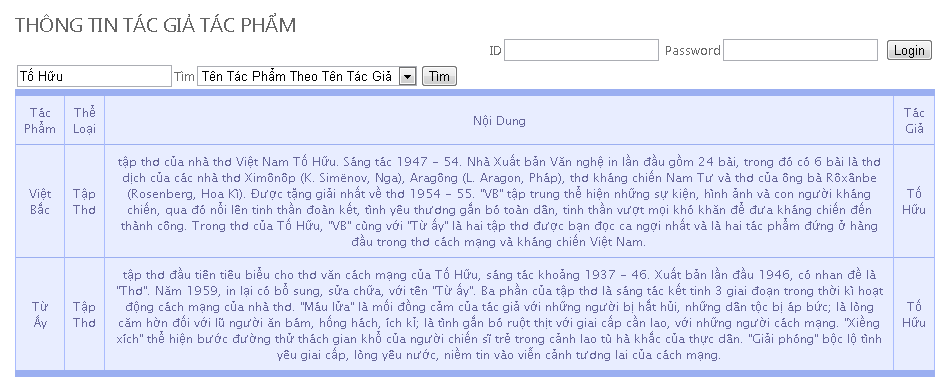
Hình 3.1: Tìm kiếm tiểu sử theo tên tác giả.

* + 1. Tìm kiếm tên tác giả theo tiểu sử.



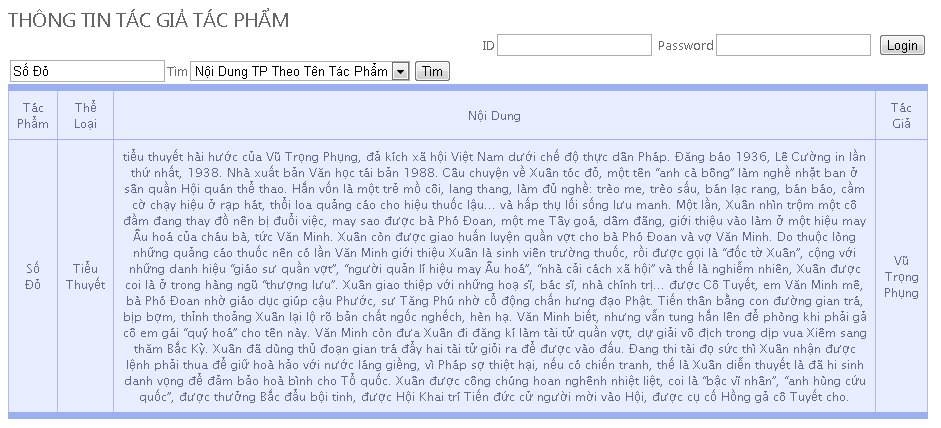
Hình 3.2: Tìm kiếm tên tác giả theo tiểu sử.

* + 1. Tìm kiếm tên tác phẩm theo tên tác giả.



Hình 3.3: Tìm kiếm tên tác phẩm theo tên tác giả

* + 1. Tìm kiếm nội dung tác phẩm theo tên tác phẩm



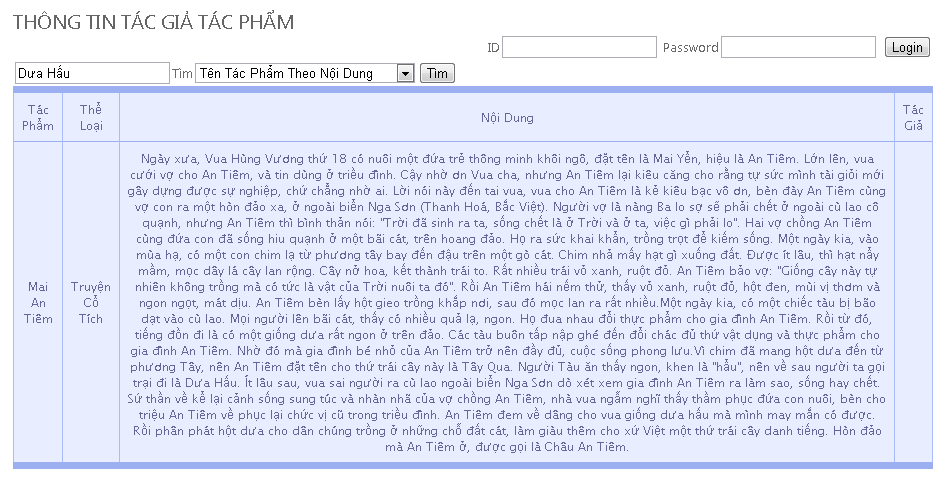
Hình 3.4: Tìm kiếm nội dung tác phẩm theo tên tác phẩm

* + 1. Tìm kiếm tên tác phẩm theo thể loại.



Hình 3.5: Tìm kiếm tên tác phẩm theo thể loại.

* + 1. Tìm kiếm tên tác phẩm theo nội dung.



Hình 3.6: Tìm kiếm tên tác phẩm theo nội dung.

* 1. Vài đoạn mã đặc biệt
     1. Service cung cấp chức năng đăng nhập

public int Login(string username, string password)

{

try

{

SqlConnection conn = new SqlConnection(conn\_taikhoan);

conn.Open();

string cmdStr = "select \* from TaiKhoan";

SqlCommand cmd = new SqlCommand(cmdStr, conn);

SqlDataReader dr = cmd.ExecuteReader();

string \_username = "";

string \_password = "";

while (dr.Read())

{

\_username = dr["Username"].ToString();

\_password = dr["Password"].ToString();

if (\_username.CompareTo(username) == 0 && \_password.CompareTo(password) == 0)

{

user.Quyen = int.Parse(dr["Quyen"].ToString());

user.username = username;

return user.Quyen;

}

}

conn.Close();

return user.Quyen = -1;

}

catch (Exception ex)

{

throw new FaultException(ex.Message);

}

}

* + 1. Service cung cấp chức năng phân quyền

public User GetQuyen()

{

return user;

}

* + 1. Service cung cấp chức năng tìm kiếm tiểu sử tác giả theo tên tác giả

public string GetTSTGbyTG(string tacgia)

{

return GetTSTGbyTGsv(tacgia, user);

}

public string GetTSTGbyTGsv(string tacgia, User user)

{

if (user.Quyen == -1)

return "Bạn không có quyền này, mời bạn đăng nhập!";

else

try

{

SqlConnection conn = new SqlConnection(conn\_tacgia);

conn.Open();

string cmdStr = "select \* from TacGia where Ten LIKE N'%" + tacgia +"%'";

SqlCommand cmd = new SqlCommand(cmdStr, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

conn.Close();

string ketqua = "<table border=1 width=100%><tr><td width=20% align=center>Tên tác giả</td><td align=center>Tiểu sử</td></tr>";

if (dt.Rows.Count == 0)

return "Chưa có thông tin về tác giả này";

else

for (int i = 0; i < dt.Rows.Count; i++)

{

ketqua += "<tr><td align=center>" + dt.Rows[i][1].ToString() + "</td>";

ketqua += "<td>" + dt.Rows[i][2].ToString() + "</td></tr>";

}

return ketqua += "</table>";

}

catch (Exception ex)

{

throw new FaultException(ex.Message);

}

}

* + 1. Service cung cấp chức năng tìm kiếm tác phẩm theo tên tác giả

public string GetTPbyTG(string tacgia)

{

return GetTPbyTGsv(tacgia, user);

}

public string GetTPbyTGsv(string tacgia, User user)

{

if (user.Quyen == -1)

return "Bạn không có quyền này, mời bạn đăng nhập!";

else

try

{

SqlConnection conn = new SqlConnection(conn\_tacpham);

conn.Open();

string cmdStr = "select \* from TacPham where TenTacGia LIKE N'%" + tacgia + "%'";

SqlCommand cmd = new SqlCommand(cmdStr, conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

conn.Close();

string ketqua = "<table border=1 style=text-align:center><tr><td>Tác Phẩm</td><td>Thể Loại</td><td>Nội Dung</td><td>Tác Giả</td></tr>";

if (dt.Rows.Count == 0)

return "Chưa có thông tin về tác giả này";

else

for (int i = 0; i < dt.Rows.Count; i++)

{

ketqua += "<tr><td>" + dt.Rows[i][1].ToString() + "</td>";

ketqua += "<td>" + dt.Rows[i][2].ToString() + "</td>";

ketqua += "<td>" + dt.Rows[i][3].ToString() + "</td>";

ketqua += "<td>" + dt.Rows[i][4].ToString() + "</td></tr>";

}

return ketqua += "</table>";

}

catch (Exception ex)

{

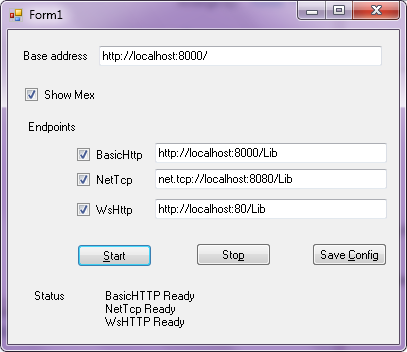
throw new FaultException(ex.Message);

}

}

* 1. Cách cài đặt và chạy chương trình
     1. Cài đặt cơ sở dữ liệu

Đầu tiên ta sẽ chạy chương trình Miscrosoft SQL Server Management Studio, sau đó chọn click phải chuột vào Database -> Attach … -> Chọn ws\_tacgia.mdf, ws\_tacpham.mdf, ws\_taikhoan.mdf.

* + 1. Cài đặt hosting

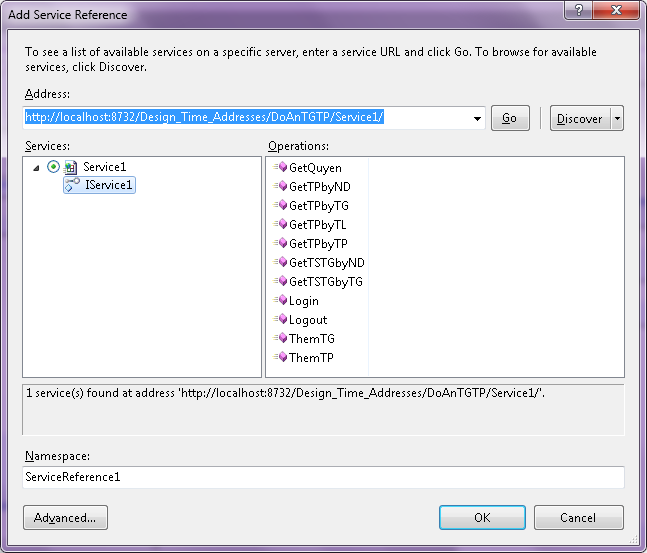
Hình 3.8: cài đặt WCF Hosting

Sau khi điền thông số như trên nhấn Start để start WCF Service Library.

* + 1. Chạy chương trình phía Client

Để Client xử dụng được các chức năng hệ thống, client phải dùng Service reference đến WCF Library.

Add service reference như hình trên để cấu hình cho client để client có thể sử dụng các chức năng của hệ thống..

Hình 3.9: cài đặt WCF Service cho Client

1. Kết luận
   1. ***Ưu điểm***

* Hoàn thành đề tài đúng thời gian đưa ra.
* Giao diện thân thiện đối với người sử dụng.
* Đáp ứng đầy đủ các chức năng cơ bản của việc cung cấp thông tin tác giả, tác phẩm.
  1. ***Khuyết điểm***
* Giải thuật chưa được tối ưu.
* Chưa xử lý các ngoại lệ.
* Tốc độ xử lý còn chậm.
* Chỉ quản lý ở mức độ đơn giản, chưa quản lý ở mức cao hơn.
* Chưa hoàn thành các chức năng thêm, xóa, sửa
  1. ***Hướng phát triển***
* Nâng cao tốc độ xử lý.
* Phát triển thêm các chức năng mới.
* Quản trị phân quyền sử dụng một cách chi tiết hơn.
* Kiểm tra thêm các ràng buộc.

1. Tài liệu tham khảo

* <http://msdn.microsoft.com>