|  |
| --- |
| **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  **Lớp :** QL092L  **Thời gian thực hiện :** 01/09/2011 – 20/12/2011  **Sinh viên thực hiện :** Dương Quốc Việt    **Tháng 11/ 20117** |

**MỤC LỤC**

1. Lý thuyết web service
2. Thiết kế chương trình
   1. Đối tượng phục vụ
   2. Kiến trúc hệ thống
   3. Chức năng chính của hệ thống
   4. Mô tả cở sở dữ liệu
3. Hiện thực chương trình
   1. Các màn hình chính của chương trình
   2. Cách thực hiện
   3. Giá trị trả về (nếu có)
   4. Cách thức cài đặt và chạy chương trình
4. Kết luận
   1. Ưu điểm
   2. Khuyết điểm
   3. Hướng phát triển trong tương lai
5. Tài liệu tham khảo

**LỜI MỞ ĐẦU**

Ngày nay khi xã hội ngày càng phát triển thì nhu cầu chuyển đổi chỗ ở do tính chất của công việc và nhu cầu đi du lịch ngày càng tăng. Do đó việc di chuyển từ địa phương này sang địa phương khác hoặc việc tạm trú, tạm vắng ngày càng gia tăng. Điều này dẫn đến các khó khăn trong công tác quản lý thông tin tạm trú, tạm vắng của một cá nhân cụ thể nào đó.

Từ đó đòi hỏi cần có một chương trình quản lý thông tin tạm trú tạm vắng để giúp cho chính quyền địa phương, cũng như các hộ kinh doanh cá nhân (khách sạn, nhà trọ) quản lý thông tin một cách nhanh chóng, dễ dàng và chính xác hơn.

Từ những yêu cầu nêu trên nhóm đã quyết định chọn đề tài ”**Xây dựng Web service cung cấp thông tin tạm trú tạm vắng**”.

Nhóm đã phân tích và đưa ra một số chức năng cơ bản của một web service cung cấp thông tin tạm trú, tạm vắng như sau:

* Quản lý thông tin tạm trú.
  + Tìm kiếm thông tin tạm trú theo họ tên của người tạm trú.
  + Tìm kiếm thông tin theo số điện thoại của người tạm trú.
  + Tìm kiếm thông tin theo chứng minh nhân dân của người tạm trú.
  + Tìm kiếm thông tin theo địa chỉ email cá nhân.
  + Liệt kê danh sách những người đang tạm trú tại địa chỉ hiện tại.
  + Thống kê giới tính theo địa chỉ hiện tại.
  + Thống kê danh sách nghề nghiệp của những người đang tạm trú tại địa chỉ hiện tại.
  + Tìm kiếm danh sách những người đang ở tại địa chỉ hiện tại theo ngày.
  + Cho phép import dữ liệu từ các file excel, csv, cơ sở dữ liệu giúp cho việc đăng ký thông tin tạm trú một cách nhanh chóng hơn, giúp người quản lý có thể sử dụng lại các danh sách đã có sẵng mà không cần nhập lại dữ liệu.
* Quản lý thông tin tạm vắng.
  + Tìm kiếm thông tin tạm vắng theo tên cá nhân.
  + Tìm kiếm thông tin tạm vắng theo số điện thoại cá nhân.
  + Tìm kiếm thông tin tạm vắng theo địa chỉ email cá nhân.
  + Tìm kiếm thông tin tạm vắng của 1 cá nhân theo chứng minh nhân dân.
  + Thống kê xem tại địa chỉ hiển tại có những cá nhân nào đăng ký tạm vắng.
  + Liệt kệ danh sách tạm vắng của địa chỉ hiện tại theo ngày.
* Quản lý thông tin chung.
  + Thêm thông tin tạm trú, tạm vắng.
  + Cập nhật thông tin tạm trú tạm vắng.
  + Xóa thông tin tạm trú, tạm vắng.
  + Đăng ký thành viên.
  + Login: đăng nhập vào hệ thống.

Danh sách thành viên thực hiện đề tài.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ tên | MSSV | Lớp | Email |
| 1 | Dương Quốc Việt | 09263L | QL092L | [viet.dq263l@sinhvien.hoasen.edu.vn](mailto:viet.dq263l@sinhvien.hoasen.edu.vn) |

**1. Tìm hiểu lý thuyết**

Phần này thực hiện trình bày về lý thuyết nghiên cứu được, ví dụ một cách tiếp cận có thể như sau:

* Giới thiệu
* Mô tả đối tượng nghiên cứu: cấu trúc chung, mô tả giao diện, phương thức hoạt động...
* Hướng dẫn lập trình

**2. Thiết kế chương trình**

***2.1 Đối tượng phục vụ.***

Chương trình được thiết kế nhằm phục vụ cho 2 đối tượng chính sau:

**2.1.1 Hộ gia đình cá nhân:** là những chủ nhà trọ, chủ khách sạn hay các doanh nghiệp thường xuyên có khách hàng cư trú lại doanh nghiệp của mình và có nhu cầu quản lý thông tin khách hàng của mình.

**2.1.2 Chính quyền địa phương:** dùng để quản lý thông tin tạm trú, tạm vắng của các cá nhân tại địa phương của mình.

***2.2 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.

***2.3 Các chức năng chính của chương trình.***

1. *GetInfoByName (tìm kiếm thông tin theo tên)*

Chương trình cho phép người quản lý tìm kiếm thông tin tạm trú, tạm vắng theo tên của một cá nhân. Khi người dùng cung cấp thông tin là họ tên của một cá nhân cụ thể, chương trình sẽ cho biết được cá nhân đó đã từng đăng ký tạm trú, tạm vắng tại những địa chỉ nào. Ngoài ra hệ thống còn cung cấp địa chỉ hiện tại đang cư trú của cá nhân đó.

**TranferRecord[] GetInfoByName(string Name, string type)**

* Tham số truyền vào: Name, type.
* Kết quả trả về: mãng đối tượng TranferRecord
* [OperationContract(Name="FindInfoByName")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Name | nvarchar(64) | Họ tên cần tìm kiếm  Ví dụ: Dương Quốc Việt |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một mãng đối tượng chứa các thông tin cá nhân như họ tên, CMND, địa chỉ tạm trú/ tạm vắng, ... |

Table 1 tìm kiếm thông tin theo tên

1. *GetInfoByPhone (tìm kiếm theo số điện thoại)*

Hệ thống cho phép người dùng tìm kiếm thông tin tạm trú, tạm vắng của một cá nhân cụ thể theo số điện thoại. Khi người sử dụng chương trình cung cấp thông tin là số điện thoại của một cá nhân, hệ thống sẽ trả về tất cả dữ liệu tạm trú/ tạm vắng của cá nhân liên quan.

**TranferRecord[] GetInfoByPhone(string Phone, string type)**

* Tham số truyền vào: Phone, type
* Kết quả trả về: mãng đối tượng TranferRecord
* [OperationContract(Name="FindInfoByPhone")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Phone | nchar(10) | Số điện thoại cần tìm kiếm  Ví dụ: 0909455034 |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một đối tượng chứa các thông tin cá nhân như họ tên, CMND, địa chỉ tạm trú/ tạm vắng, ... |

Table 2 tìm kiếm theo số điện thoại

1. *GetInfoByIdNumber (tìm kiếm theo chứng minh nhân dân)*

Hệ thống cho phép tìm kiếm thông tin tạm trú, tạm vắng của một cá nhân theo chứng minh nhân dân. Người sử dụng chỉ cần nhập vào số chứng minh nhân dân, hệ thống.

**TranferRecord[] GetInfoByIdNumber(string IDNumber, string type)**

* Tham số truyền vào: IdNumber, type
* Kết quả trả về: danh sách các đối tượng
* [OperationContract(Name="FindInfoByIdNumber")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| IDNumber | nchar(10) | Số chứng minh nhân dân/ passport  Ví dụ: 023859423 |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một đối tượng chứa các thông tin cá nhân như họ tên, CMND, địa chỉ tạm trú/ tạm vắng, ... |

Table 3 tìm kiếm theo chứng minh nhân dân

1. *GetInfoBySex (tìm kiếm theo giới tính & địa chỉ)*

Hệ thống cho phép kê liệt kê danh sách những người đang tạm trú hoặc tạm vắng theo giới tính và địa chỉ hiện tại.

**TranferRecord[]GetInfoBySex (string Address, string Sex, string type)**

* Tham số truyền vào: Address, Sex, type
* Kết quả trả về: dãy các đối tượng.
* [OperationContract(Name="ListPersonBySex")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Address | nvarchar(128) | Địa chỉ cần tìm  Ví dụ: 023859423 |
| Sex | nchar(10) | Giới tính. Ví dụ: Nam/ Nữ |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một đối tượng chứa các thông tin cá nhân như họ tên, CMND, địa chỉ tạm trú/ tạm vắng, ... |

Table 4 tìm kiếm theo giới tính & địa chỉ.

1. *GetListByAddress (tìm kiếm theo địa chỉ)*

Hệ thống cho phép người sử dụng tìm kiếm theo địa chỉ, người dùng nhập địa chỉ cần tìm. Hệ thống sẽ trả về danh sách những người đang tạm trú hoặc tạm vắng tại địa chỉ đã nhập.

**TranferRecord[] GetListByAddress(string Address, string type)**

* Tham số truyền vào: Address, type
* Kết quả trả về: mãng các đối tượng.
* [OperationContract(Name="ListPersonByAddress")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Address | nvarchar(128) | Địa chỉ cần tìm |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một đối tượng chứa các thông tin cá nhân như họ tên, CMND, địa chỉ tạm trú/ tạm vắng, ... |

Table 5 tìm kiếm theo địa chỉ

1. *GetListOccupationByAddress (thống kê nghề nghiệp theo địa chỉ)*

Ngoài các chức năng cơ bản, hệ thống còn cho phép người quản lý thống kê giới tính tại địa chỉ hiện tại, chức năng này giúp người quản lý dễ dàng trong việc thống kê giới tính cho địa phương của mình.

**TranferRecord[] GetListOccupationByAddress(string Address, string type)**

* Tham số truyền vào: Address, type
* Kết quả trả về: mãng các đối tượng.
* [OperationContract(Name="ListOccupationByAddress")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Address | nvarchar(128) | Địa chỉ cần tìm |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một đối tượng chứa các thông tin cá nhân như họ tên, CMND, địa chỉ tạm trú/ tạm vắng, ... |

Table 6 thống kê nghề nghiệp theo địa chỉ

1. *GetInfoByEmail (tìm kiếm thông tin theo địa chỉ email)*

Hệ thống cho phép người dùng tìm kiếm thông tin tạm trú, tạm vắng theo địa chỉ email.

**TranferRecord[] GetInfoByEmail(string Email, string type)**

* Tham số truyền vào: Email, type
* Kết quả trả về: mãng các đối tượng.
* [OperationContract(Name = "FindInfoByEmail")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Email | nvarchar(64) | Địa chỉ email |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một đối tượng chứa các thông tin cá nhân như họ tên, CMND, địa chỉ tạm trú/ tạm vắng, ... |

Table 7 tìm kiếm thông tin theo địa chỉ email

1. *GetListByDate (tìm kiếm danh sách theo ngày)*

Hệ thống cho phép tìm kiếm danh sách theo địa chỉ từ ngày đến ngày. Khi người dùng cung cấp thông tin là địa chỉ cần tìm, khoảng thời gian cần tìm kiếm, hệ thống sẽ cung cấp danh sách tương ứng.

**TranferRecord[] GetListByDate(string Address, DateTime FromDate, DateTime ToDate, string type)**

* Tham số truyền vào: Address, FromDate, ToDate, type.
* Kết quả trả về: danh sách đội tượng.
* [OperationContract(Name="ListPersonByDate")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Address | nvarchar(128) | Địa chỉ cần tìm |
| FromDate | Date | Ngày bắt đầu tạm trú/ tạm vắng |
| ToDate | Date | Ngày kết thúc tạm trú/ tạm vắng |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một đối tượng chứa các thông tin cá nhân như họ tên, CMND, địa chỉ tạm trú/ tạm vắng, ... |

Table 8 tìm kiếm danh sách theo ngày

1. *CountSexByAddress (thống kê giới tính tại địa chỉ hiện tại)*

Hệ thống cho phép thống kê giới tính theo địa chỉ hiện tại, chức năng giúp cơ quan địa phương thống kê giới tính, ...

**TranferRecord[] CountSexByAddress(string Address, string type)**

* Tham số truyền vào: Address, type.
* Kết quả trả về: mãng đối tượng thống kê giới tính tại địa chỉ cần tìm.
* [OperationContract(Name="CountSexByAddress")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Address | nvarchar(128) | Địa chỉ tạm trú/ tạm vắng |
| Type | nchar(10) | Loại thông tin cần tìm kiếm (tạm trú/ tạm vắng). |
| TranferRecord | Object | Kết quả trả về là một đối tượng thống kê giới tính của địa chỉ cần tìm. |

Table 9 thống kê giới tính tại địa chỉ hiện tại

1. *GetData (hiển thị tất cả thông tin tạm trú, tạm vắng)*

Chức năng liệt kê tất cả danh sách tạm trú, tạm vắng theo từng hộ gia đình hoặc địa phương.

**TranferRecord[] GetData()**

* Tham số truyền vào: không có.
* Kết quả trả về: danh sách các đối tượng chứa thông tin tạm trú, tạm vắng.
* [OperationContract(Name="LoadData")]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| TranferRecord | Object | Kết quả trả về là danh sách các đối tượng chứa thông tin tạm trú, tạm vắng. |

1. *ImportData (đăng ký thông tin tạm trú từ danh sách có sẵng)*

Cho phép import dữ liệu từ các file excel, csv, cơ sở dữ liệu giúp cho việc đăng ký thông tin tạm trú một cách nhanh chóng hơn, giúp người quản lý có thể sử dụng lại các danh sách đã có sẵng mà không cần nhập lại dữ liệu.

**void ImportData(string[] file)**

* Tham số truyền vào: danh sách các file chứa thông tin khách hàng cần đăng ký tạm trú
* Kết quả trả về: không có.
* [OperationContract(Name = "ImportDataFromFile", IsOneWay = true)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| file | array | Danh sách các file chứa thông tin khách hàng cần đăng ký tạm trú. |

Table 11 đăng ký thông tin tạm trú từ danh sách có sẵng

1. *Login (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 12 đăng nhập vào hệ thống

1. *InsertData (thêm thông tin tạm trú/ tạm vắng)*

Hệ thống cho phép người sử dụng chương trình đăng ký thông tin tạm trú, tạm vắng.

**void InsertData(TranferRecord data)**

* Tham số truyền vào: đối tượng chứa tất cả thông tin tạm trú/ tạm vắng.
* Kết quả trả về: không có.
* [OperationContract(Name = "ImportDataFromFile", IsOneWay = true)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Data | TranferRecord | Đối tượng chứa tất cả thông tin cá nhân của người muốn đăng ký thông tin tạm trú/ tạm vắng. |

Table 13 thêm thông tin tạm trú/ tạm vắng

1. *UpdateData (cập nhật thông tin tạm trú/ tạm vắng)*

Hệ thống cung cấp chức năng cập nhật thông tin tạm trú/ tạm vắng. Cho phép người quản lý cập nhật các thông tin của đối tượng tham gia tạm trú hoặc tạm vắng.

**void UpdateData(int Id, TranferRecord data)**

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

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Id | Int | Id của cá nhân sẽ cập nhật thông tin |
| Data | TransferRecord | Đối tượng chứa các thông tin như: họ tên, ngày sinh, .... đây là những thông tin cần cập nhật. |

Table 14 cập nhật thông tin tạm trú/ tạm vắng

1. *DeleteData (xóa thông tin tạm trú/ tạm vắng)*

Hệ thống cung cấp chức năng xóa thông tin cá nhân của đối tượng tham gia tạm trú/ tạm vắng.

**void DeleteData(int Id)**

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

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Id | Int | Id của cá nhân sẽ xóa thông tin |

Table 15 xóa thông tin tạm trú/ tạm vắng

**Đối tượng chứa thông tin tạm trú tạm vắng**

[DataContract]

public class TranferRecord

{

[DataMember]

public int Total;

[DataMember]

public int Id;

[DataMember]

public string FullName;

[DataMember]

public string PhoneNumber;

[DataMember]

public string Email;

[DataMember]

public DateTime Birthday;

[DataMember]

public string Sex;

[DataMember]

public string OriginalAddress;

[DataMember]

public string IDNumber;

[DataMember]

public string Occupation;

[DataMember]

public string CurrentAddress;

[DataMember]

public DateTime FromDate;

[DataMember]

public DateTime ToDate;

[DataMember]

public string Reason;

[DataMember]

public string Description;

[DataMember]

public string Type;

}

**Đối tượng chứa thông tin đăng nhập**

[DataContract]

public class UserRecord

{

[DataMember]

public string username;

[DataMember]

public string password;

}

***2.4 Mô tả cơ sở dữ liệu***

*tamtrutamvang*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| Id | Int | No | Khóa chính |
| FullName | nvarchar(64) | Yes | Họ tên |
| PhoneNumber | nchar(10) | Yes | Số điện thoại |
| Email | nchar(64) | Yes | Địa chỉ email |
| Birthday | date | Yes | Ngày sinh |
| Sex | nchar(10) | Yes | Giới tính |
| OriginalAddress | nvarchar(128) | Yes | Địa chỉ thường trú |
| IDNumber | nchar(10) | Yes | Chứng minh nhân dân/ Passport |
| Occupation | nvarchar(50) | Yes | Nghề nghiệp |
| CurrentAddress | nvarchar(128) | Yes | Địa chỉ hiện tại (tạm trú/ tạm vắng) |
| FromDate | Date | Yes | Ngày bắt đầu tạm trú/ tạm vắng |
| ToDate | Date | Yes | Ngày kết thúc tạm trú/ tạm vắng |
| Reason | Text | Yes | Lý do tạm trú/ tạm vắng |
| Description | Text | Yes | Mô tả chi tiết |
| Type | nchar(10) | Yest | Loại (tạm trú/ tạm vắng). Ví dụ: tạm trú (“TT”), tạm vắng (“TV”) |

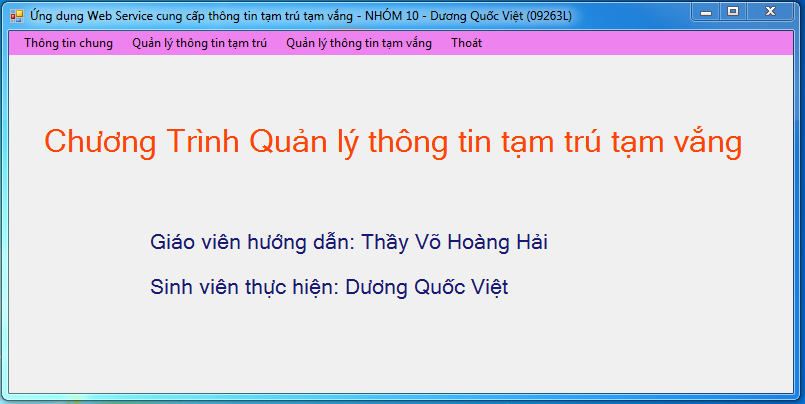
*Users*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| Username | nvarchar(10) | No | Tài khoản đăng nhập |
| Password | nchar(10) | No | Mật khẩu đăng nhập |

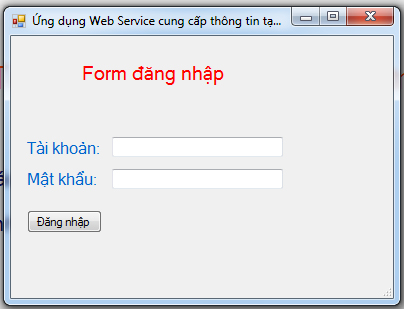
**3. Hiện thực**

**3.1 Các màn hình chính của chương trình.**

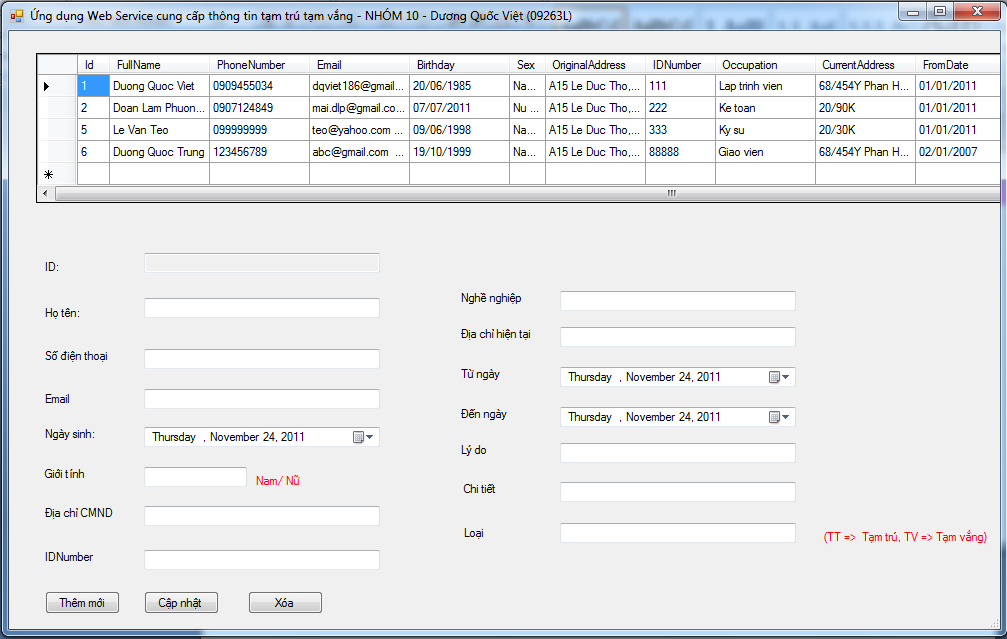
3.1.1 Giao diện chính của chương trình.



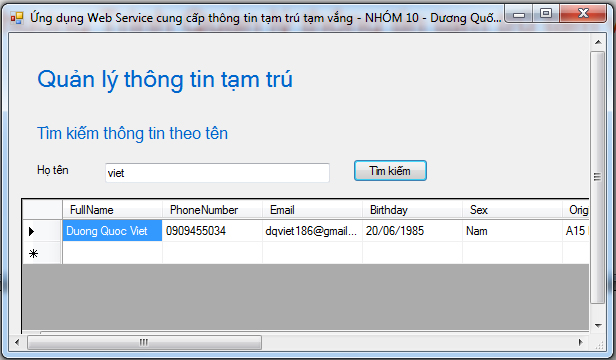
3.1.2 Form đăng nhập vào hệ thống.



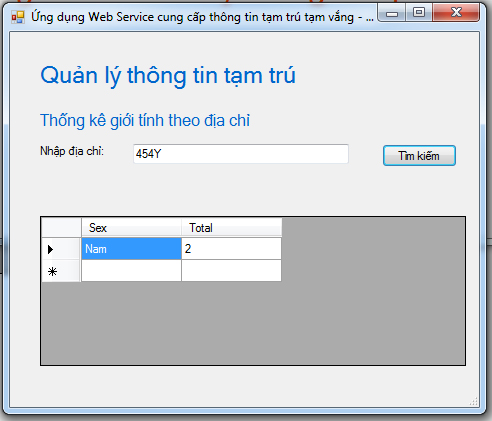
3.1.3 Form thêm, sửa, xóa thông tin tạm trú/ tạm vắng.



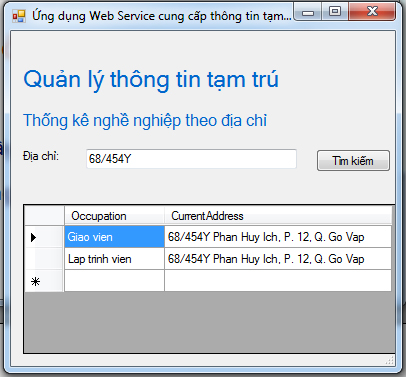
3.1.4 Tìm kiếm thông tin theo tên.



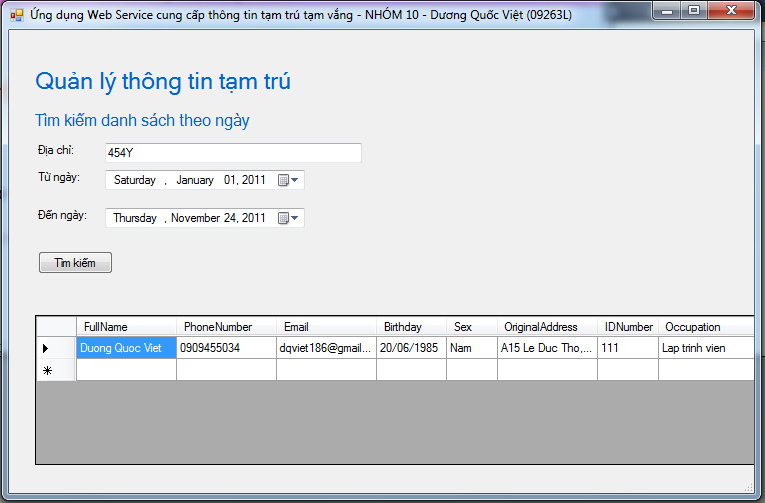
3.1.5 Thống kê giới tính theo địa chỉ hiện tại.



3.1.6 Thống kê nghề nghiệp



* + 1. Tìm kiếm thông tin theo ngày.



* 1. **Vài đoạn mã đặc biệt**

**3.2.1 Service cung cấp chức năng thêm dữ liệu**

public void InsertData(TranferRecord data)

{

try

{

string connectionString = "Data Source=mavi-PC;Initial Catalog=cnweb;Integrated Security=True";

myConnection = new SqlConnection(connectionString);

// Open connection

myConnection.Open();

string updateCement = "INSERT INTO tamtrutamvang (FullName,PhoneNumber,Email,Birthday,Sex,OriginalAddress,IDNumber,Occupation,CurrentAddress,FromDate,ToDate,Reason,Description,Type) VALUES (@FullName,@PhoneNumber,@Email,@Birthday,@Sex,@OriginalAddress,@IDNumber,@Occupation,@CurrentAddress,@FromDate,@ToDate,@Reason,@Description,@Type)";

SqlCommand cmd = new SqlCommand(updateCement, myConnection);

cmd.Parameters.AddWithValue("@FullName", data.FullName);

cmd.Parameters.AddWithValue("@PhoneNumber", data.PhoneNumber);

cmd.Parameters.AddWithValue("@Email", data.Email);

cmd.Parameters.AddWithValue("@Birthday", data.Birthday.ToString("yyyy-MM-dd"));

cmd.Parameters.AddWithValue("@Sex", data.Sex);

cmd.Parameters.AddWithValue("@OriginalAddress", data.OriginalAddress);

cmd.Parameters.AddWithValue("@IDNumber", data.IDNumber);

cmd.Parameters.AddWithValue("@Occupation", data.Occupation);

cmd.Parameters.AddWithValue("@CurrentAddress", data.CurrentAddress);

cmd.Parameters.AddWithValue("@FromDate", data.FromDate.ToString("yyyy-MM-dd"));

cmd.Parameters.AddWithValue("@ToDate", data.ToDate.ToString("yyyy-MM-dd"));

cmd.Parameters.AddWithValue("@Reason", data.Reason);

cmd.Parameters.AddWithValue("@Description", data.Description);

cmd.Parameters.AddWithValue("@Type", data.Type);

cmd.ExecuteNonQuery();

// close connection

myConnection.Close();

}

catch (PisNotFoundException exp)

{

TrackedFault tf = new TrackedFault(

Guid.NewGuid(),

exp.Message,

DateTime.Now);

throw new FaultException<TrackedFault>(

tf,

new FaultReason("PisNotFoundException"),

FaultCode.CreateReceiverFaultCode(new FaultCode("InsertInfomation")));

}

catch (Exception exp)

{

FaultReasonText reason = new FaultReasonText(exp.Message);

throw new FaultException(new FaultReason(reason), FaultCode.CreateReceiverFaultCode(new FaultCode("InsertInfomation")));

}

}

**3.2.2 Service cung cấp chức năng tìm kiếm theo ngày**

public TranferRecord[] GetListByDate(string Address, DateTime FromDate, DateTime ToDate, string type)

{

try

{

SqlConnection cn = new SqlConnection();

cn.ConnectionString = "Data Source=mavi-PC;Initial Catalog=cnweb;Integrated Security=True";

cn.Open();

string str1 = FromDate.ToString("yyyy-MM-dd");

string str2 = ToDate.ToString("yyyy-MM-dd");

string query = "SELECT \* FROM tamtrutamvang WHERE CurrentAddress like '%" + Address + "%' and FromDate >= '" + str1 + "' or ToDate <= '" + str2 + "' and Type = '" + type + "'";

SqlCommand sql = new SqlCommand(query, cn);

SqlDataAdapter adt = new SqlDataAdapter(sql);

DataSet ds = new DataSet();

adt.Fill(ds);

int rows = ds.Tables[0].Rows.Count;

data = new TranferRecord[rows];

for (int i = 0; i < rows; i++)

{

data[i] = new TranferRecord();

data[i].Id = Convert.ToInt32(ds.Tables[0].Rows[i].ItemArray[0].ToString());

data[i].FullName = ds.Tables[0].Rows[i].ItemArray[1].ToString();

data[i].PhoneNumber = ds.Tables[0].Rows[i].ItemArray[2].ToString();

data[i].Email = ds.Tables[0].Rows[i].ItemArray[3].ToString();

data[i].Birthday = Convert.ToDateTime(ds.Tables[0].Rows[i].ItemArray[4]);

data[i].Sex = ds.Tables[0].Rows[i].ItemArray[5].ToString();

data[i].OriginalAddress = ds.Tables[0].Rows[i].ItemArray[6].ToString();

data[i].IDNumber = ds.Tables[0].Rows[i].ItemArray[7].ToString();

data[i].Occupation = ds.Tables[0].Rows[i].ItemArray[8].ToString();

data[i].CurrentAddress = ds.Tables[0].Rows[i].ItemArray[9].ToString();

data[i].FromDate = Convert.ToDateTime(ds.Tables[0].Rows[i].ItemArray[10]);

data[i].ToDate = Convert.ToDateTime(ds.Tables[0].Rows[i].ItemArray[11]);

data[i].Reason = ds.Tables[0].Rows[i].ItemArray[12].ToString();

data[i].Description = ds.Tables[0].Rows[i].ItemArray[13].ToString();

data[i].Type = ds.Tables[0].Rows[i].ItemArray[14].ToString();

}

cn.Close();

return data;

}

catch (PisNotFoundException exp)

{

TrackedFault tf = new TrackedFault(

Guid.NewGuid(),

exp.Message,

DateTime.Now);

throw new FaultException<TrackedFault>(

tf,

new FaultReason("PisNotFoundException"),

FaultCode.CreateReceiverFaultCode(new FaultCode("GetListByDate")));

}

catch (Exception exp)

{

FaultReasonText reason = new FaultReasonText(exp.Message);

throw new FaultException(new FaultReason(reason), FaultCode.CreateReceiverFaultCode(new FaultCode("GetListByDate")));

}

}

**3.2.3 Cách thức lấy dữ liệu phía client**

// tao dataset

public DataSet CreateData()

{

DataSet ds = new DataSet();

ds.Tables.Add("tamtrutamvang");

//ds.Tables[0].Columns.Add("Id");

ds.Tables[0].Columns.Add("FullName");

ds.Tables[0].Columns.Add("PhoneNumber");

ds.Tables[0].Columns.Add("Email");

ds.Tables[0].Columns.Add("Birthday");

ds.Tables[0].Columns.Add("Sex");

ds.Tables[0].Columns.Add("OriginalAddress");

ds.Tables[0].Columns.Add("IDNumber");

ds.Tables[0].Columns.Add("Occupation");

ds.Tables[0].Columns.Add("CurrentAddress");

ds.Tables[0].Columns.Add("FromDate");

ds.Tables[0].Columns.Add("ToDate");

ds.Tables[0].Columns.Add("Reason");

ds.Tables[0].Columns.Add("Description");

//ds.Tables[0].Columns.Add("Type");

return ds;

}

public DataSet dataTTTV(string Address, DateTime FromDate, DateTime ToDate, string type)

{

DataSet ds = new DataSet();

ServiceReference1.pisClient proxy = new ServiceReference1.pisClient();

TTTVService.TranferRecord[] result = proxy.ListPersonByDate(Address, FromDate, ToDate,type);

ds = CreateData();

DataRow dr;

for (int i = 0; i < result.Length; i++)

{

dr = ds.Tables["tamtrutamvang"].NewRow();

//dr["Id"] = result[i].Id;

dr["FullName"] = result[i].FullName;

dr["PhoneNumber"] = result[i].PhoneNumber;

dr["Email"] = result[i].Email;

dr["Birthday"] = result[i].Birthday.ToString("dd/MM/yyyy");

dr["Sex"] = result[i].Sex;

dr["OriginalAddress"] = result[i].OriginalAddress;

dr["IDNumber"] = result[i].IDNumber;

dr["Occupation"] = result[i].Occupation;

dr["CurrentAddress"] = result[i].CurrentAddress;

dr["FromDate"] = result[i].FromDate.ToString("dd/MM/yyyy");

dr["ToDate"] = result[i].ToDate.ToString("dd/MM/yyyy");

dr["Reason"] = result[i].Reason;

dr["Description"] = result[i].Description;

//dr["Type"] = result[i].Type;

// add row to table

ds.Tables["tamtrutamvang"].Rows.Add(dr);

}

return ds;

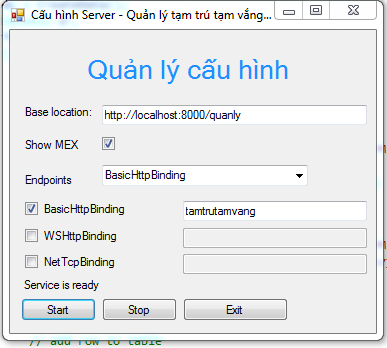
}

* 1. **Cách cài đặt và chạy chương trình.**

3.3.1 Cài đặt cơ sở dữ liệu

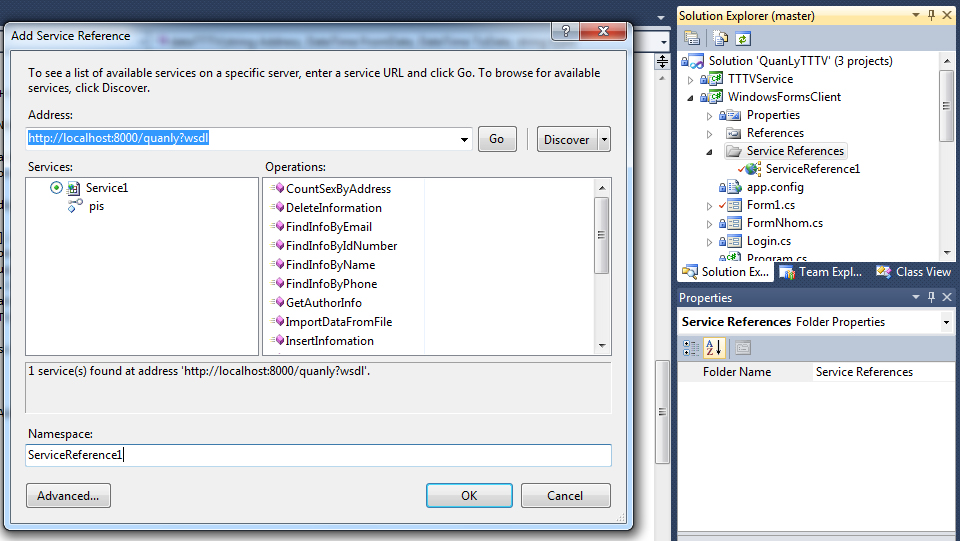
Đầu tiên ta sẽ chạy chương trình Miscrosoft SQL Server Management Studio, sau đó chọn click phải chuột vào Database -> Attach … -> Chọn cnweb.mdf

3.3.2 Cài đặt hosting



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

3.3.3 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.

**4. 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 tạm trú, tạm vắng.
* Có thể áp dụng ở mức độ quản lý cơ quản.
  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 như thống kê theo quận, tỉnh/ thành phố.
* Chưa xử lý việc import dữ liệu từ các file CSV, txt, … để đăng ký thông tin tạm trú, tạm vắng.
* Chưa hiện thực chức năng tìm kiếm theo email.
  1. ***Hướng phát triển***
* Nâng cao tốc độ xử lý.
* Hệ thống cung cấp các báo cáo, thống kê theo thời gian thực. Các bản thống kê được biểu diễn bằng các hình ảnh trực quan.
* Quản trị phân quyền sử dụng một cách chi tiết hơn như: phân quyền phân hệ, chức năng, hành động.
* Xây dựng client là web application.
* Kiểm tra các ràng buộc.

**5. TÀI LIỆU THAM KHẢO**

* <http://msdn.microsoft.com>
* <http://www.telerik.com/help/aspnet-ajax/grid-client-side-binding-to-wcf-service-adonet-dataservice.html>
* <http://www.codeproject.com/KB/combobox/valuemembersample.aspx>
* <http://www.planet-source-code.com>