Design and Implementation of personal Information Management System in Internet

Liu Chengxia[1,2] Zhu MinLing[1]

[1]Beijing Information and Technology University
[2]Beijing University of Posts and Telecommunications

Abstract

Now people want to access their own personal information from different places and in different ways. So a system which can provide users the method to manage personal information with a Rich Interactive Application is put forward. In this system user can manage his/her address book, mails, individual assets and expenditure of family, or check related accounts, statistics, schedules and attachments. The system is designed as a distributed architecture based on Windows Communication Foundation (WCF) model, and developed by Microsoft Silverlight. At last a complete system which can provide users convenient remote access and innovative user experience is given.

Key words: Rich Interactive Application; WCF; Distributed architecture; Silverlight;

1. Introduction

As the development of Internet, people always hope that he/she can get all what he/she needed by internet. The personal information is needed everywhere and we hope that we can find it and access it anytime and anywhere. So a personal information management system is brought forward.

User can manage the address book, this can help you to find who you want contact with. Also you can send or receive mails, make individual schedules and manage the expenditure of family. An attach function of this system is to check related user accounts, of course the person that the account belongs to is your family or friends. The statistics of the data is provided as the reference to make decisions.
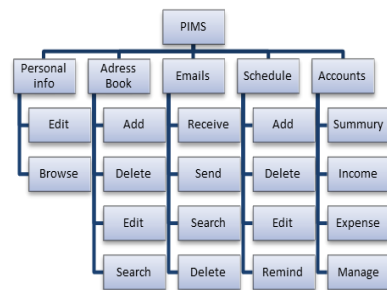

Fig. 1 System Function Module

2. Design of the System

2.1 Silverlight and WCF

Silverlight is a powerful development tool for creating engaging, interactive user experiences for Web and mobile applications. Silverlight is a free plug-in, powered by the .NET framework and compatible with multiple browsers, devices and operating systems, bringing a new level of interactivity wherever the Web works. But by using Silverlight a lot of tradition-

al function of communication, such as ADO, can't be used. To solve this Windows Communication Foundation (WCF) is used in this system.

WCF is Microsoft's unified programming model for building service-oriented applications. It enables developers to build secure, reliable, transacted solutions that integrate across platforms and interoperate with existing investments. WCF simplifies development of connected applications through a new service-oriented programming model. WCF supports many styles of distributed application development by providing a layered architecture.
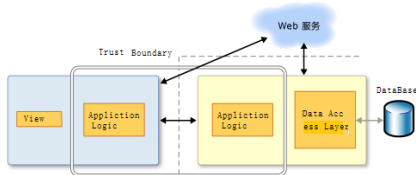

Fig.2 WCF in Distributed Architecture

In this figure, the trust boundary is divided again and by this way the distributed application can be combined together.

2.2 Data Base Design

In order to manage the personal data a personal database is needed and the database can be arranged in a database server.

The personal data include user's information, Address book, Mail, Schedule, and Account. Although the Mail's attachment can be saved individually and some log information be saved in a table.
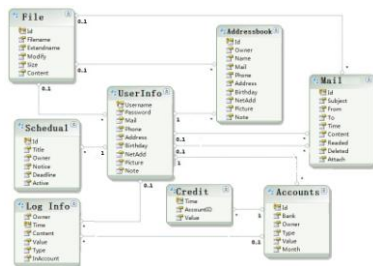

Fig.3 Data Relations

2.3 User Interface Design

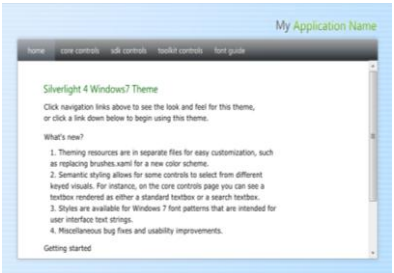Using the Window7 theme template as the base we design our own pages.

Fig.4. Windows7 theme template

There are many pages in the system. The main page is typical and be brought out as an example page. In the main page a brief review of the users personal information is shown. Also the most interested information is revealed for further processing.
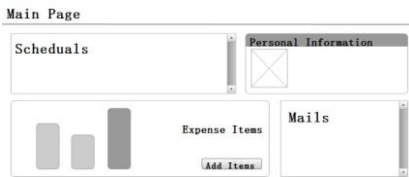

Fig.5. Main Page Design

3. Implement of the System

3.1 Cross Domain Data Service of WCF

To use the cross domain service we should start the Silverlight of WCF service and set the attributes. For example, the settings of navigation attribute (Accounts):

```
internal sealed class AccountsMetadata
{    private AccountsMetadata(){ }
    public string Bank { get; set; }
    [Include]
    [Association("FK_Credit", "Id", "AccountID")]
    public EntityCollection<Credit> Credit { get;
set; }
    public long Id { get; set; }
```

```
public string Owner { get; set; }
public bool Type { get; set; }
public Nullable<decimal> Value { get; set; }
public Nullable<byte> Month { get; set; }
}
```

In the example, [Association ("FK_Credit","Id","AccountID")] means that associating the foreign key to the primary key. So the relations of data can be mapped in the service.

3.2 WCF Service

As service-oriented applications, WCF use the publish/subscribe mode. Use the PostMail in AlterService.svc as an instance we can find how WCF works.

```
[OperationContract]
public void PostMail(string to, string content)
{
    MailSender mail = new MailSender(to);
    mail.onError += new ErrorEv-
                    -entHandler(ErrorLog);
    mail.message.Subject = "Message for-
             -warded by Fensalir System";
    mail.message.Body = content;
    mail.send();
}
```

In this instance the parameters are the receiver and the content of the mail. And after the service application of server is created the client should import it and call the service.

3.3 Page Logic Implement

In order to implement the management of the personal information the code behind the page is necessary to realize the different function logics. The address book can be modified and the delete button's function code is come out as an example:

```
private void btnDelete_Click(object sender,
RoutedEventArgs e)
{
    if (MessageBox.Show("Delete User" +
(dataGrid1.SelectedItem as User).Username,
"Delete", MessageBoxButton.OKCancel) ==
MessageBoxResult.OK)
    {
    client.Users.Remove
(dataGrid1.SelectedItem as User);
```

```
    client.SubmitChanges(new Action
<SubmitOperation> (refresh), null);
    }
}
```

Within the above code "Submit Changes" method is used to delete data. All of the pending changes are submitted in this operation. A callback method can be provided when there are codes that must execute after the asynchronous operation has finished. In this program a callback method named "refresh" is called which used to check for errors and update the user interface as needed.

Another example can be seen in schedules management. We can filter the schedules by date.

```
private bool enable(Schedule val)
{
    switch (cmbView.SelectedIndex)
    {
     case 0:    // filter by today
       if (val.Deadline == DateTime.Now.Date
          & !val.Active)
           return true;
         return false;
     case 1:    // filter by this week
       if (val.Deadline <= DateTime.Now.Date.
          AddDays(7) & !val.Active)
          return true;
        return false;
     case 2:    // all schedules
       if (!val.Active)
           return true;
        return false;
     case 3:    // finish
       if (val.Deadline >= DateTime.Now.Date.
          AddMonths(-3) & val.Active)
          return true;
        return false;
    }
    return false;
    }
```

4. Testing and Deployment

4.1 Deployment

The system is designed for personal using. So the users always use the PC in his/her own house to do the work. Of course dif-

ferent client type can be used too, such as PC, laptop, cellphone etc.
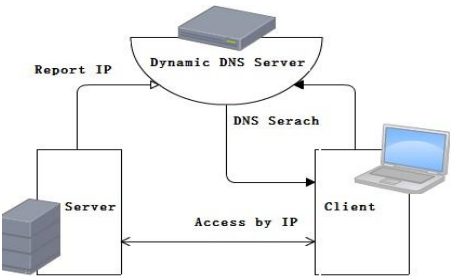

Fig.6 Schematic of the Dynamic DNS

4.2 Testing

The system is used for personal information management. So we login and do some daily work. There are many processing pages in the whole system but we can't spread them all. We will take some typical use of the system as instance to explain it.

For example, the user wants to check the emails and get the page-view below.

Fig.7 Mail Page testing

And then he/she wants to check the accounts and add a new expense item.

Fig.8 Account Review Page

In this page the accounts of the family can be reviewed and the income and expense is shown in pie graph or line graph. You can add new items and make some limits to some items. The limit is used to remind you to pay attention to the item.

Although there are many lacks of the system, such as accounts management should more comprehensive, the system provide a new thinking and it can used easily. It can help people to manage their own information more efficiently.

5. References

[1] http://zh.wikipedia.org/wiki/ WPF/E. Microsoft Silverlight [DB/OL]. 2011. 12.27.

[2] http://zh.wikipedia.org/wiki/RIA

[3] http://msdn.microsoft.com/en-us/ library/ms731082(v=vs.90).aspx.

[4] http://msdn.microsoft.com/en-us/ library/ff423067(v=VS.91).aspx