

1

ONLINE DIARY

SỔ NHẬT KÝ TRỰC TUYẾN

Qua chương này, bạn sẽ tạo được một sổ nhật ký trực tuyến (online diary) và trình quản lý danh bạ liên lạc (contacts manager). Vậy chính xác thì sổ nhật ký trực tuyến và trình quản lý danh bạ liên lạc làm được việc gì? Sử dụng một giao diện dựa trên lịch biểu, bạn có thể thêm, xóa, và chỉnh sửa một mục nhật ký cho bất kỳ ngày nào. Bạn cũng có thể tạo các sự kiện, chẳng hạn như ghi nhớ ngày sinh của một người nào đó. Các sự kiện không chỉ là ngày sinh mà còn có thể là các cuộc họp, cuộc hẹn,...

Hệ thống này có một hệ thống đăng nhập (gồm tên người dùng và mật khẩu), để chỉ có bạn (chứ không ai khác) mới có thể xem nhật ký của bạn. Đây là điểm khác biệt so với blog. Hệ thống này là một trình quản lý danh bạ liên lạc và nhật ký cá nhân—là nơi đặt những suy nghĩ và những ghi chú mà bạn không muốn mọi người nhìn thấy. Còn blog là nơi bạn muốn tất cả mọi người nhìn thấy.

Toàn bộ dự án này cho thấy sức mạnh của *ASP.NET 2.0* và dễ dàng tạo nên các dự án thế này. Đã qua rồi cái thời phải viết hàng trăm dòng mã để thực hiện đăng nhập, tạo người dùng mới,... Chương này tận dụng các thành phần bảo mật mới của *ASP.NET 2.0* nhằm cho bạn thấy rằng rất dễ dàng tạo nên các dự án hữu ích và thú vị.

Phần đầu tiên sẽ hướng bạn sử dụng sổ nhật ký và các màn hình chính của nó. Phần tiếp theo cho bạn cái nhìn tổng quan về thiết kế hệ thống. Sau đó, bạn bước vào các thành phần của hệ thống, cũng như cách kết hợp chúng với nhau. Trong phần cuối cùng, bạn sẽ cài đặt sổ nhật ký.

1.1

Sử dụng Online Diary

Mỗi người dùng có một sổ nhật ký trực tuyến riêng, để truy cập cần phải đăng nhập. Nhập tên người dùng là *demo* với mật khẩu là *password#* để đăng nhập làm người dùng thử. Màn hình đăng nhập được thể hiện trong hình 1-1.



Hình 1-1

Mặc dù màn hình này có thể gợi cho bạn nhiều điều kiểm tra và nhiều mã lệnh tạo nên chức năng bảo mật, thật ra với các điều kiểm tra bảo mật mới trong *ASP.NET 2.0* thì mọi việc trở nên rất dễ dàng và chẳng có nhiều việc để làm.

Chương 1: Online Diary

Nếu bạn chưa đăng ký, liên kết *Bạn chưa có tài khoản? Nhấp vào đây để đăng ký!* sẽ đưa bạn đến trang đăng ký, được mô tả trong hình 1-2.

Hình 1-2

Hình này cho thấy một điều kiểm khác nữa trong số các điều kiểm bảo mật mới trong *ASP.NET 2.0*; việc tạo một quy trình đăng ký giờ đây chỉ là thêm một điều kiểm vào một form!

Nếu quên mật khẩu, bạn có thể nhấp vào liên kết *Bạn quên mật khẩu?*, và bạn sẽ được dẫn đến trang nhắc mật khẩu (xem hình 1-3).

Hình 1-3

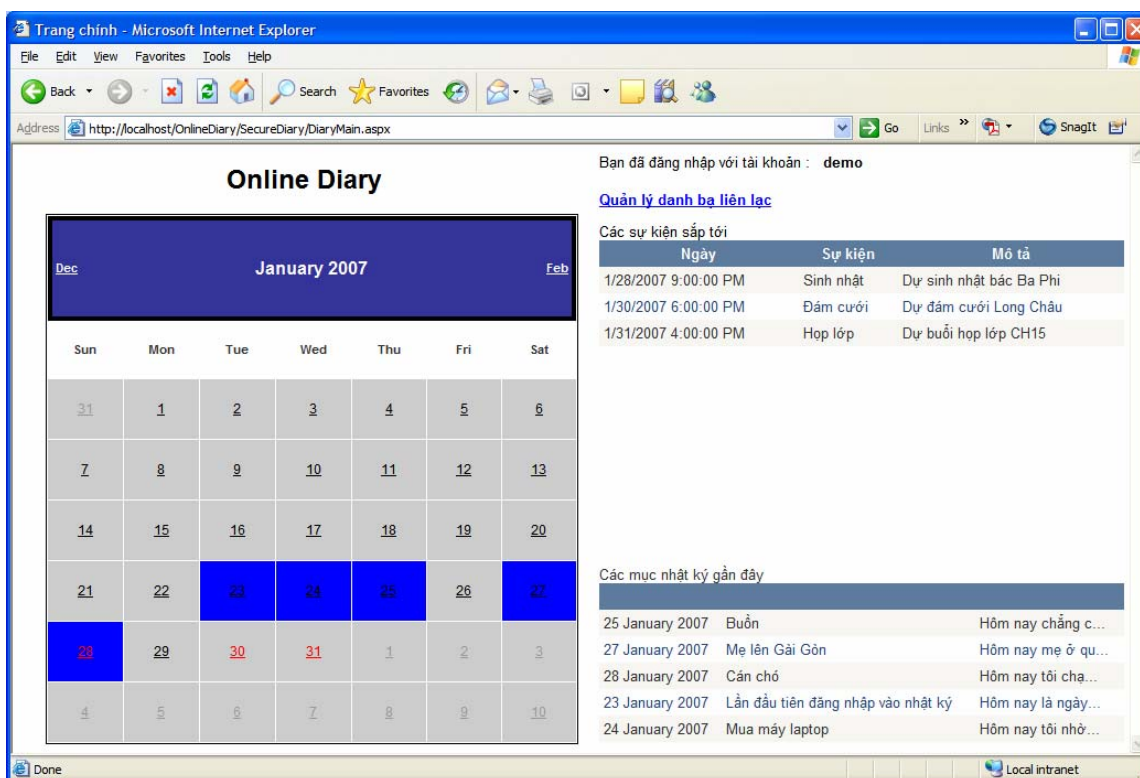
Sau khi đăng nhập, bạn đến trang nhật ký chính, được hiển thị trong hình 1-4.

Trên trang này, bạn thấy một lịch biểu theo tháng. Các ngày có các mục nhật ký được đánh dấu bằng nền màu xanh. Các ngày có các sự kiện được đánh dấu bằng chữ màu đỏ. Cũng để ý rằng, phía bên phải là các sự kiện sắp tới và các mục nhật ký gần đây.

Nhấp vào một ngày sẽ đưa bạn đến vùng nhập mục nhật ký cho ngày hôm đó; và thêm, sửa, xóa các sự kiện (xem hình 1-5)

Bạn cũng có thể điều hướng sổ nhật ký của mình từ đây bằng một lịch biểu nhỏ bên phải.

Thêm một mục nhật ký bằng cách nhập vào hộp *Tiêu đề* và hộp *Nội dung*, sau đó nhấp nút *Lưu nhật ký*.



Hình 1-4



Hình 1-5

Các sự kiện diễn ra vào một ngày cụ thể được liệt kê ở góc dưới hình 1-5. Bạn có thể chỉnh sửa và xóa các sự kiện, hoặc nhấp vào liên kết *Thêm sự kiện mới* để thêm một sự kiện mới. Các trang sửa và thêm sự kiện gần như giống nhau. Hình 1-6 là một ví dụ của trang sửa sự kiện.

Trong trang sửa sự kiện, bạn có thể nhập tên sự kiện, mô tả ngắn về sự kiện, thời gian bắt đầu sự kiện, và sự kiện kéo dài trong bao lâu.

28 January 2007

Sửa sự kiện

Tên sự kiện

Mô tả sự kiện

Giờ bắt đầu

Thời lượng (tính bằng phút)

Hình 1-6

Trở lại trang nhật ký chính (hình 1-4), bạn sẽ thấy liên kết *Quản lý danh bạ liên lạc* (xem hình 1-7).

Bạn đã đăng nhập với tài khoản : **demo**

[Quản lý danh bạ liên lạc](#)

Các sự kiện sắp tới

Ngày	Sự kiện	Mô tả
1/28/2007 9:00:00 PM	Sinh nhật	Dự sinh nhật bác Ba Phi
1/30/2007 6:00:00 PM	Đám cưới	Dự đám cưới Long Châu
1/31/2007 4:00:00 PM	Họp lớp	Dự buổi họp lớp CH15

Hình 1-7

Nhấp vào liên kết này, bạn sẽ được dẫn đến trang quản lý sổ liên lạc (xem hình 1-8).

Danh bạ liên lạc

[Trang chính](#)

[Thêm liên lạc mới](#)

	Họ và chữ lót	Tên	Điện thoại	E-mail
Xóa Sửa	Đinh Phan Chí	Tâm		dpctam@yahoo.com
Xóa Sửa	Lê Ngọc	Sơn		lnson@yahoo.com
Xóa Sửa	Nguyễn Ngọc Bình	Phương		nnbphuong@yahoo.com
Xóa Sửa	Thái Kim	Phụng		tkphung@yahoo.com

Hình 1-8

Tại đây, bạn sẽ thấy danh sách các liên lạc. Bạn có thể chỉnh sửa và xóa liên lạc bằng cách nhấp vào liên kết phù hợp. Bạn cũng có thể thêm một liên lạc mới bằng cách nhấp vào liên kết *Thêm liên lạc mới*, liên kết này sẽ dẫn bạn đến trang thêm liên lạc (xem hình 1-9).

Hình 1-9

Hiện tại, chức năng danh bạ liên lạc khá đơn giản, không có các chức năng như liên kết các sự kiện và các cá nhân, tự động gửi mail đến các cá nhân trong danh bạ liên lạc để nhắc họ về một sự kiện.

Bạn đã thấy những gì *Online Diary* làm được, bây giờ bạn có thể xem nó “đã làm điều đó như thế nào” (câu nói quen thuộc của một chương trình truyền hình)! Phần kế tiếp mô tả bản thiết kế tổng quan và hệ thống gắn kết với nhau như thế nào. Bạn sẽ tìm hiểu cơ sở dữ liệu và các lớp mà hệ thống sử dụng.

1.2

Thiết kế Online Diary

Hệ thống nhật ký được chia thành kiến trúc ba tầng. Tất cả dữ liệu và mã lệnh hiệu chỉnh dữ liệu trực tiếp nằm trong tầng truy xuất dữ liệu—sự kết hợp của các bảng cơ sở dữ liệu và các thủ tục tồn trữ.

Phía trên tầng truy xuất dữ liệu là tầng nghiệp vụ, tầng này cung cấp tất cả các quy tắc và tính thông minh của hệ thống. Tầng nghiệp vụ được tổ chức thành bảy lớp.

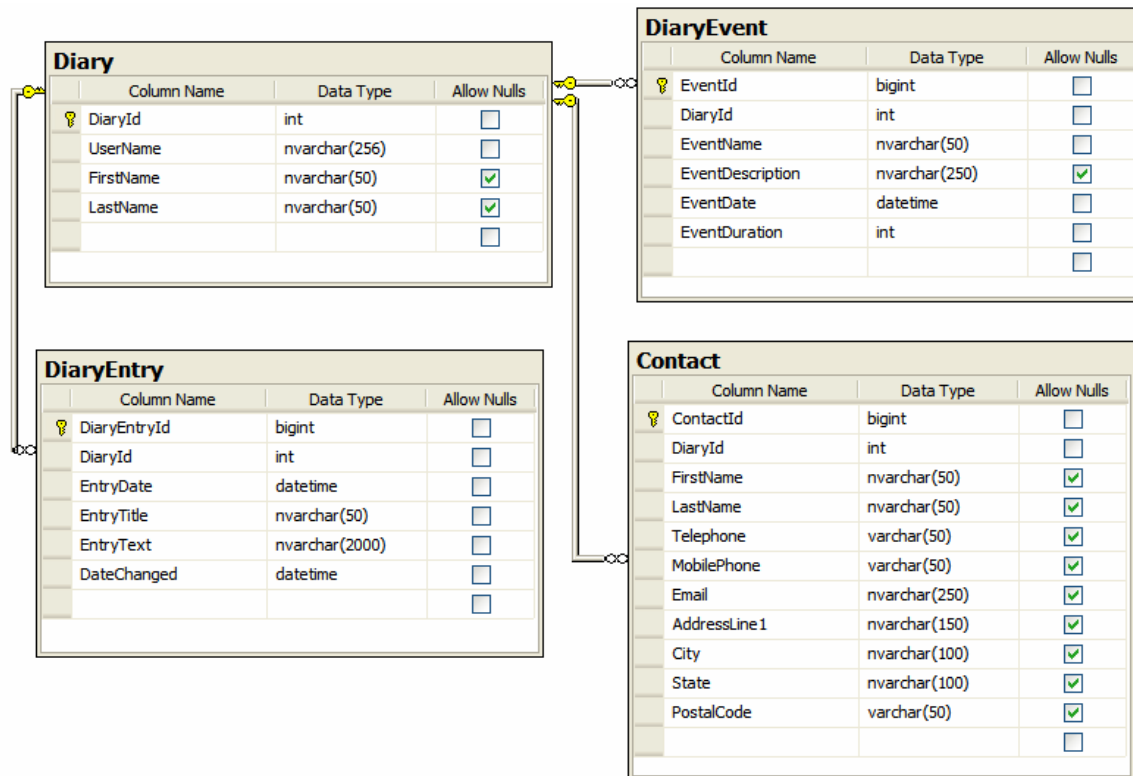
Được thảo luận sau cùng là tầng trình bày. Tầng này bao gồm một số file *.aspx*, sử dụng tầng nghiệp vụ và tầng truy xuất dữ liệu để tạo giao diện của nhật ký.

1.2.1 Tầng truy xuất dữ liệu

Online Diary sử dụng cơ sở dữ liệu *SQL Server 2005 Express*. Tuy nhiên, không có lý do gì không thể thay đổi cơ sở dữ liệu khác. Nếu cơ sở dữ liệu hỗ trợ thủ tục tồn trữ thì theo lý thuyết, cần thay đổi chuỗi kết nối và tạo những thủ tục tồn trữ tương ứng với những thủ tục tồn trữ hiện có trong cơ sở dữ liệu *SQL Server*. Nếu cơ sở dữ liệu không hỗ trợ thủ tục tồn trữ (chẳng hạn, *MS Access*) thì thay đổi mã lệnh là cần thiết nhưng không khó.

Chương 1: Online Diary

Hình 1-10 mô tả các bảng trong cơ sở dữ liệu *Online Diary* (DiaryDB).



Hình 1-10

Cơ sở dữ liệu mặc định được tạo bằng các tính năng membership của *ASP.NET 2.0* cũng được sử dụng. Để nối kết việc đăng nhập và các chi tiết nhật ký, trường *UserName* trong cơ sở dữ liệu *DiaryDB* phải lấy giá trị gốc của nó từ cơ sở dữ liệu membership. Các chi tiết về membership nằm trong cơ sở dữ liệu *ASPNETDB* mà *Visual Web Developer Express* tạo cho bạn. Mặc dù nó chỉ có một vài bảng, bạn không bao giờ truy xuất chúng thông qua mã lệnh. Nó được truy xuất bởi các điều kiện *Login* mới—toàn bộ công việc được thực hiện phía hậu trường!

Dự án này chỉ sử dụng bảng *aspnet_Users* (xem hình 1-11) để đăng nhập và cung cấp tên người dùng cho *DiaryDB*. Bạn có thể mở rộng cơ sở dữ liệu membership để thêm các chức năng khác như xác định sự trải nghiệm của người dùng hoặc quy định các cấp độ membership khác nhau (admin, user, operator).

dbo.aspnet_Users(ASPNETDB.MDF)		
Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
UserId	uniqueidentifier	<input type="checkbox"/>
UserName	nvarchar(256)	<input type="checkbox"/>
LoweredUserName	nvarchar(256)	<input type="checkbox"/>
MobileAlias	nvarchar(16)	<input checked="" type="checkbox"/>
IsAnonymous	bit	<input type="checkbox"/>
LastActivityDate	datetime	<input type="checkbox"/>
		<input type="checkbox"/>

Hình 1-11

Các bảng của cơ sở dữ liệu *Online Diary* và vai trò của chúng được liệt kê trong bảng sau:

Tên bảng	Mô tả
Diary	Chứa thông tin chi tiết của tất cả người dùng, gồm ID và tên của họ.
DiaryEntry	Chứa tất cả mục nhật ký của tất cả người dùng.
DiaryEvent	Chứa tất cả sự kiện nhật ký của tất cả người dùng
Contact	Chứa thông tin chi tiết của tất cả liên lạc cho các sổ nhật ký.

Khóa liên kết tất cả các bảng với nhau là trường `DiaryId`. Nó là khóa chính trong bảng `Diary` và là khóa ngoại trong tất cả các bảng khác. Tại sao không sử dụng trường `UserName`? Về cơ bản là tốc độ—sẽ dễ và nhanh hơn khi kết bảng và tìm kiếm trên một trường kiểu số nguyên so với các trường kiểu ký tự.

Mọi truy xuất đến cơ sở dữ liệu đều thông qua thủ tục tồn trữ. Xét thủ tục tồn trữ sau:

`DeleteContact`

`DeleteContact` xóa một liên lạc khỏi cơ sở dữ liệu. Tên của mỗi thủ tục tồn trữ đã thể hiện mục đích của thủ tục tồn trữ đó nên không cần giải thích nhiều. Khi thảo luận về mã lệnh, bạn sẽ xem xét các thủ tục tồn trữ kỹ hơn.

1.2.2 Tầng nghiệp vụ

Tầng nghiệp vụ được tổ chức thành bảy lớp. Trong đó, bốn lớp chính là:

- `OnlineDiary`
- `DiaryEntry`
- `DiaryEvent`
- `Contact`

Các lớp này thực hiện hầu hết công việc giữ dữ liệu nhật ký tạm thời, thu lấy và lưu nó vào cơ sở dữ liệu. Lớp đầu tiên được thảo luận là `OnlineDiary`.

❑ Lớp `OnlineDiary`

Lớp này chỉ có hai phương thức công khai chia sẻ, được mô tả chi tiết trong bảng sau:

Phương thức	Kiểu trả về	Mô tả
<code>InsertDiary(ByVal UserName As String, ByVal FirstName As String, LastName As String)</code>	không	Thêm một người dùng mới vào cơ sở dữ liệu <i>Online Diary</i> .
<code>GetDiaryIdFromUserName(ByVal UserName As String)</code>	Integer	Tìm kiếm <code>UserName</code> trong cơ sở dữ liệu và trả về <code>DiaryId</code> tương ứng.

Mục đích của lớp `OnlineDiary` là cung cấp hai phương thức chia sẻ liên quan đến một nhật ký trực tuyến. Nó cũng có thể được sử dụng để mở rộng hệ thống nhật ký và thêm các chức năng mới vào hệ thống, không chỉ là một phần cụ thể như danh bạ liên lạc.

❑ Lớp `Contact`

Chương 1: Online Diary

Lớp Contact thể hiện hóa một liên lạc—một người hay một thứ gì đó mà bạn muốn lưu thông tin liên lạc. Nó đóng gói mọi thứ để làm việc với các liên lạc, bao gồm lưu trữ và thu lấy thông tin liên lạc trong cơ sở dữ liệu.

Nó có hai phương thức khởi dựng, được mô tả trong bảng sau:

Phương thức khởi dựng	Mô tả
New(ByVal Diaryid as Integer)	Tạo một đối tượng Contact mới với tất cả các thuộc tính được gán bằng các giá trị mặc định.
New(ByVal ContactId As Long)	Tạo một đối tượng Contact mới với các thuộc tính được lấy từ cơ sở dữ liệu bằng đối số ContactId.

Với một đối tượng Contact đã được tạo, việc lưu nó chỉ đơn giản là gọi phương thức Save(). Lớp này sẽ xem xét đó có phải một liên lạc mới cần được chèn vào cơ sở dữ liệu hay không, hay là một liên lạc đã tồn tại cần được cập nhật. Ngoài phương thức Save(), lớp Contact còn có hai phương thức Delete() và hai phương thức GetContacts(), chúng được mô tả trong bảng sau:

Phương thức	Kiểu trả về	Mô tả
Save()	không	Lưu một đối tượng Contact đã có đầy đủ dữ liệu. Nếu nó là một liên lạc mới, Save() gọi thủ tục InsertNewContact và các chi tiết của liên lạc được chèn vào cơ sở dữ liệu. ContactId mới được trả về từ cơ sở dữ liệu và được gán cho mContactId. Nếu liên lạc này đã tồn tại trong cơ sở dữ liệu, Save() gọi UpdateContact để cập nhật cơ sở dữ liệu với các giá trị trong đối tượng Contact.
DeleteContact()	không	Xóa đối tượng Contact khỏi cơ sở dữ liệu với ContactId bằng với mContactId của đối tượng Contact. Các giá trị của đối tượng Contact được khởi tạo lại bằng các giá trị mặc định.
DeleteContact(ByVal ContactId As Long)	không	Phương thức chia sẻ này xóa đối tượng Contact khỏi cơ sở dữ liệu với giá trị ContactId bằng với đối số ContactId của phương thức.

<code>GetContactsByFirstLetter(ByVal DiaryId As Integer,Optional ByVal FirstLetterOfSurname As Char)</code>	SqlDataReader	<p>Phương thức chia sẻ này trả về một đối tượng SqlDataReader gồm một danh sách các liên lạc mà chữ cái đầu tiên trong phần họ trùng với đối số FirstLetterOfSurname. Đối số này là tùy chọn; nếu để trống, tất cả các đối tượng Contact bất chấp chữ cái đầu tiên của phần họ sẽ có mặt trong các hàng của DataSet.</p>
<code>GetContactsByFirstLetterAsCollection(ByVal DiaryId As Integer,Optional ByVal FirstLetterOfSurname As Char)</code>	ContactCollection	<p>Phương thức chia sẻ này trả về một đối tượng ContactCollection gồm các đối tượng Contact mà chữ cái đầu tiên trong phần họ trùng với đối số FirstLetterOfSurname. Đối số này là tùy chọn; nếu để trống, tất cả các đối tượng Contact bất chấp chữ cái đầu tiên của phần họ sẽ có mặt trong các hàng của DataSet.</p>

Lớp Contact chứa các thuộc tính sau:

Thuộc tính	Kiểu	Mô tả
ContactId	Long	Mỗi liên lạc được đại diện bởi một ID duy nhất. ID này được sinh tự động bởi bảng Contact trong cơ sở dữ liệu mỗi khi một liên lạc mới được thêm vào.
FirstName	String	Tên.
LastName	String	Họ (và chữ lót).
Email	String	Địa chỉ e-mail.
Telephone	String	Số điện thoại cố định.
MobilePhone	String	Số điện thoại di động.
AddressLine1	String	Địa chỉ nhà.
City	String	Tên tỉnh thành.

State	String	Tên quốc gia.
PostalCode	String	Mã vùng.

❑ Lớp ContactCollection

Lớp ContactCollection kế thừa từ lớp System.Collections.CollectionBase. Mục đích của lớp ContactCollection là lưu trữ một tập hợp các đối tượng Contact.

Lớp ContactCollection chỉ có một thuộc tính duy nhất:

Thuộc tính	Kiểu	Mô tả
Item(ByVal Index As Integer)	Integer	Trả về đối tượng Contact tại vị trí Index trong tập hợp.

Các phương thức công khai của lớp ContactCollection được mô tả trong bảng sau:

Phương thức	Kiểu trả về	Mô tả
Add(ByVal NewContact As Contact)	không	Thêm một đối tượng Contact vào tập hợp ContactCollection.
Add(ByVal ContactId As Long)	không	Tạo một đối tượng Contact mới. ContactId được truyền cho phương thức khởi dựng của đối tượng Contact để đảm bảo có được các chi tiết của liên lạc từ cơ sở dữ liệu. Sau đó, đối tượng Contact mới được thêm vào tập hợp ContactCollection.
Remove(ByVal Index as Integer)	không	Loại bỏ đối tượng Contact khỏi tập hợp tại vị trí Index.

Đó là các lớp làm việc với danh bạ liên lạc; bây giờ xét đến hai lớp làm việc với các mục nhật ký.

❑ Lớp DiaryEntry

Lớp DiaryEntry thể hiện hóa một mục trong nhật ký. Nó đóng gói mọi thứ để làm việc với các mục nhật ký, bao gồm: tạo, cập nhật, và thu lấy dữ liệu của mục nhật ký. Nó thụ lý mọi truy xuất cơ sở dữ liệu cho các mục nhật ký.

Nó có ba phương thức khởi dựng, được mô tả trong bảng sau:

Phương thức khởi dựng	Mô tả
New(ByVal DiaryId as Integer)	Tạo một đối tượng DiaryEntry mới với tất cả các thuộc tính được gán bằng các giá trị mặc định.
New(ByVal DiaryEntryId As Long)	Tạo một đối tượng DiaryEntry mới với các thuộc tính được lấy từ cơ sở dữ liệu bằng đối số DiaryEntryId.
New(ByVal DiaryId AS Integer, ByVal EntryDate As Date)	Tạo một đối tượng DiaryEntry mới với các thuộc tính được lấy từ cơ sở dữ liệu bằng đối số DiaryId và EntryDate.

Với một đối tượng `DiaryEntry` đã được tạo, chỉ việc gọi phương thức `Save()` để lưu nó. Giống như phương thức `Save()` của lớp `Contact`, lớp `DiaryEntry` sẽ xét đó có phải một mục nhật ký mới cần được chèn vào cơ sở dữ liệu, hay là một mục nhật ký đã tồn tại cần được cập nhật. Cũng như cho phép lấy các chi tiết của một mục nhật ký, lớp `DiaryEntry` cung cấp thêm các phương thức để lấy các chi tiết của một số mục nhật ký ở dạng một tập hợp hoặc một `sqlDataReader`. Các phương thức của lớp này được mô tả trong bảng sau:

Phương thức	Kiểu trả về	Mô tả
<code>Save()</code>	không	Lưu một đối tượng <code>DiaryEntry</code> đã có đầy đủ dữ liệu. Nếu nó là một mục mới, <code>Save()</code> gọi thủ tục <code>InsertNewDiaryEntry</code> và các chi tiết mục nhật ký được chèn vào cơ sở dữ liệu. <code>DiaryEntryId</code> mới được trả về từ cơ sở dữ liệu và được gán cho <code>mDiaryEntryId</code> . Nếu mục nhật ký này đã tồn tại trong cơ sở dữ liệu, <code>Save()</code> gọi <code>UpdateContact</code> để cập nhật cơ sở dữ liệu với các giá trị trong đối tượng <code>DiaryEntry</code> .
<code>GetDaysInMonthWithEntries(ByVal DiaryId As Integer, ByVal Month As Integer, ByVal Year As Integer)</code>	Mảng Boolean	Phương thức chia sẻ này trả về một mảng Boolean cho biết những ngày nào có mục nhật ký. Chỉ số mảng ứng với ngày trong tháng (ví dụ: 1 là ngày thứ nhất, 2 là ngày thứ hai,...).
<code>GetDiaryEntriesByDate(ByVal DiaryId As Integer, ByVal FromDate As Date, ByVal ToDate As Date)</code>	<code>SqlDataReader</code>	Phương thức chia sẻ này trả về một đối tượng <code>SqlDataReader</code> gồm các mục nhật ký nằm giữa đối số <code>FromDate</code> và <code>ToDate</code> .
<code>GetDiaryEntriesByDateAsCollection(ByVal DiaryId As Integer, ByVal FromDate As Date, ByVal ToDate As Date)</code>	<code>DiaryEntryCollection</code>	Tạo một đối tượng <code>DiaryEntryCollection</code> gồm các đối tượng <code>DiaryEntry</code> có <code>EntryDate</code> nằm giữa đối số <code>FromDate</code> và <code>ToDate</code> .
<code>GetDiaryEntriesRecentlyChanged(ByVal DiaryId As Integer)</code>	<code>SqlDataReader</code>	Trả về một <code>SqlDataReader</code> gồm các mục nhật ký được tạo gần đây.

Chương 1: Online Diary

Ngoài các phương thức khởi tạo và các phương thức trên, lớp `DiaryEntry` có các thuộc tính sau:

Thuộc tính	Kiểu	Mô tả
<code>EntryTitle</code>	<code>String</code>	Tiêu đề mục nhật ký của một ngày.
<code>EntryText</code>	<code>String</code>	Nội dung mục nhật ký của một ngày.
<code>EntryDate</code>	<code>Date</code>	Ngày tạo mục nhật ký.

Lớp làm việc với các mục nhật ký là `DiaryEntryCollection`, lớp này sẽ được giải thích tiếp theo.

❑ Lớp `DiaryEntryCollection`

Lớp `DiaryEntryCollection` kế thừa từ lớp `System.Collections.CollectionBase`. Mục đích của nó là lưu trữ một tập hợp các đối tượng `DiaryEntry`.

Lớp này chỉ có duy nhất một thuộc tính, được mô tả trong bảng sau:

Thuộc tính	Kiểu	Mô tả
<code>Item(ByVal Index As Integer)</code>	<code>Integer</code>	Trả về đối tượng <code>DiaryEntry</code> tại vị trí <code>Index</code> trong tập hợp.

Cùng với thuộc tính `Item()`, lớp `DiaryEntryCollection` có ba phương thức công khai:

Phương thức	Kiểu trả về	Mô tả
<code>Add(ByVal New DiaryEntry As DiaryEntry)</code>	không	Thêm một đối tượng <code>DiaryEntry</code> vào tập hợp <code>DiaryEntryCollection</code> .
<code>Add(ByVal DiaryEntryId As Long)</code>	không	Tạo một đối tượng <code>DiaryEntry</code> mới. <code>DiaryEntryId</code> được truyền cho phương thức khởi dựng của đối tượng <code>DiaryEntry</code> để đảm bảo có được dữ liệu của mục nhật ký từ cơ sở dữ liệu. Sau đó, đối tượng <code>DiaryEntry</code> mới được thêm vào tập hợp <code>DiaryEntryCollection</code> .
<code>Remove(ByVal Index as Integer)</code>	không	Loại bỏ đối tượng <code>DiaryEntry</code> ra khỏi tập hợp tại vị trí <code>Index</code> .

Chúng ta đã thảo luận về các lớp liên quan đến danh bạ liên lạc và các mục nhật ký. Phần tiếp theo sẽ thảo luận về các sự kiện nhật ký.

❑ Lớp `DiaryEvent`

Lớp `DiaryEvent` thể hiện hóa một mục trong nhật ký. Nó đóng gói mọi thứ để làm việc với các mục nhật ký, bao gồm tạo, cập nhật, và thu lấy dữ liệu của các sự kiện nhật ký. Nó thụ lý mọi truy xuất cơ sở dữ liệu cho các sự kiện nhật ký.

Lớp `DiaryEvent` có ba phương thức khởi dựng, được mô tả trong bảng sau:

Phương thức khởi dựng	Mô tả
-----------------------	-------

New(ByVal DiaryId As Integer)	Tạo một đối tượng DiaryEvent mới với tất cả các thuộc tính được gán bằng giá trị mặc định.
New(ByVal EntryId As Long)	Tạo một đối tượng DiaryEvent mới với các thuộc tính được lấy từ cơ sở dữ liệu bằng đối số EventId.
New(ByVal DiaryId As Integer, ByVal EventDate As Date)	Tạo một đối tượng DiaryEvent mới với các thuộc tính được lấy từ cơ sở dữ liệu bằng đối số DiaryId và EventDate.

Với một đối tượng DiaryEvent đã được tạo, chỉ việc gọi phương thức Save() để lưu nó. Lớp này xét đó có phải một sự kiện nhật ký mới cần được chèn vào cơ sở dữ liệu, hay là đã tồn tại cần được cập nhật. Lớp DiaryEvent cũng có hai phương thức Delete(). Thứ nhất là phương thức chia sẻ, do đó không cần tạo DiaryEvent, chỉ cần thông số EventId. Nó được sử dụng bởi một số thành phần truy xuất dữ liệu nội tại đi cùng với *ASP.NET 2.0*. Thứ hai là phương thức đối tượng, dùng để xóa sự kiện được tham chiếu bởi đối tượng DiaryEvent hiện tại. Cũng như cho phép lấy các chi tiết của một mục nhật ký, lớp DiaryEvent cung cấp thêm các phương thức để lấy các chi tiết của một số sự kiện nhật ký ở dạng tập hợp hay SqlDataReader.

Bảng sau mô tả chi tiết các phương thức này:

Phương thức	Kiểu trả về	Mô tả
Save()	không	Lưu một đối tượng DiaryEvent đã có đầy đủ dữ liệu. Nếu nó là một mục mới, Save() gọi thủ tục InsertNewDiaryEvent và các chi tiết được chèn vào cơ sở dữ liệu. EventId mới được trả về từ cơ sở dữ liệu và được gán cho mEventId. Nếu mục này đã tồn tại trong cơ sở dữ liệu, Save() gọi UpdateDiaryEvent để cập nhật cơ sở dữ liệu với các giá trị trong đối tượng DiaryEvent.
GetDaysInMonthWithEvents(ByVal DiaryId As Integer, ByVal Month As Integer, ByVal Year As Integer)	Mảng Boolean	Phương thức chia sẻ này trả về một mảng Boolean cho biết những ngày nào có sự kiện. Chỉ số mảng ứng với ngày trong tháng (ví dụ: 1 là ngày thứ nhất, 2 là ngày thứ hai,...).
GetDiaryEventsByDate(ByVal DiaryId As Integer, ByVal FromDate As Date, ByVal ToDate As Date)	SqlDataReader	Phương thức chia sẻ này trả về một đối tượng SqlDataReader gồm các sự kiện nhật ký từ FromDate đến ToDate.

GetDiaryEventsByDateAsCollection(ByVal DiaryId As Integer, ByVal FromDate As Date, ByVal ToDate As Date)	DiaryEventCollection	Tạo một đối tượng DiaryEventCollection mới gồm các đối tượng DiaryEvent có EntryDate từ FromDate đến ToDate.
DeleteEvent()	không	Xóa khỏi cơ sở dữ liệu sự kiện có EventId bằng với mEventId. Các giá trị của đối tượng DiaryEvent được khởi tạo lại bằng giá trị mặc định.
DeleteEvent(ByVal EventId As Long)	không	Phương thức chia sẻ này xóa khỏi cơ sở dữ liệu sự kiện có giá trị EventId bằng với đối số EventId của phương thức.

Ngoài các phương thức khởi dựng và các phương thức công khai, lớp DiaryEvent có bốn thuộc tính sau:

Thuộc tính	Kiểu	Mô tả
EventDescription	String	Mô tả sự kiện.
EventName	String	Tên ngắn của sự kiện.
EventDate	Date	Ngày bắt đầu sự kiện.
EventDuration	Integer	Thời lượng của sự kiện (tính theo phút)

Phần kế tiếp sẽ xét đến lớp thụ lý tập hợp nhật ký: DiaryEventCollection.

❑ Lớp DiaryEventCollection

Lớp DiaryEventCollection kế thừa từ lớp System.Collections.CollectionBase. Mục đích của nó là lưu trữ một tập các đối tượng DiaryEvent. Lớp này có các phương thức sau:

Phương thức	Kiểu trả về	Mô tả
Add(ByVal NewDiaryEvent As DiaryEvent)	không	Thêm một đối tượng DiaryEvent vào tập hợp DiaryEventCollection.
Add(ByVal DiaryEventId As Long)	không	Tạo một đối tượng DiaryEvent mới. DiaryEventId được truyền cho phương thức khởi dựng của đối tượng DiaryEvent để đảm bảo có được các chi tiết của sự kiện từ cơ sở dữ liệu. Sau đó, đối tượng DiaryEvent mới được thêm vào tập hợp DiaryEventCollection.
Remove(ByVal Index As Integer)	không	Loại bỏ đối tượng DiaryEvent khỏi tập hợp tại vị trí Index.

Lớp này chỉ có duy nhất một thuộc tính:

Thuộc tính	Kiểu	Mô tả
Item(ByVal Index As Integer)	Integer	Trả về đối tượng DiaryEvent tại vị trí Index trong tập hợp.

Đó là tổng quan về tất cả các lớp và bản thiết kế, các phương thức và các thuộc tính của chúng. Phần kế tiếp sẽ đi sâu vào mã lệnh và các trang .aspx liên quan đến tầng trình bày.

1.3

Mã lệnh Online Diary

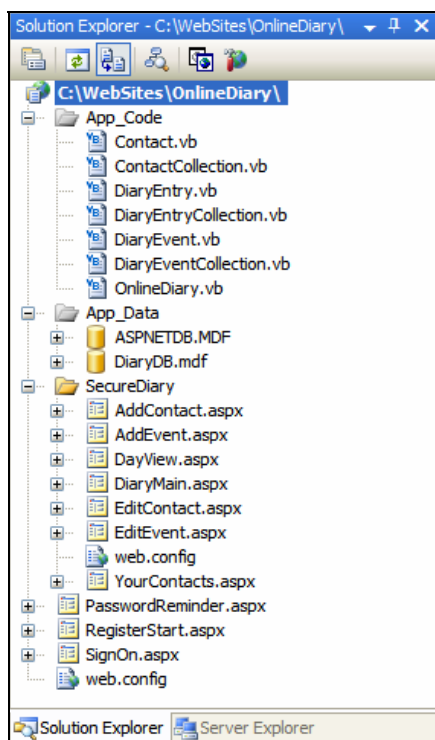
Phần này đi sâu vào từng trang quan trọng và cho biết chúng tương tác với nhau thế nào, cũng như chúng sử dụng các lớp trong tầng nghiệp vụ ra sao. Phần này không phân tích từng dòng của mỗi trang, nhưng phần nào cũng cho bạn một cái nhìn tổng quan về cách ứng dụng làm việc và đi sâu hơn vào những chỗ cần thiết.

Thảo luận về dự án này được tiếp cận theo lối dựa trên chức năng. Thay vì thảo luận từng trang cụ thể, các phần sau đây sẽ thảo luận theo từng quy trình—chẳng hạn như quy trình đăng ký.

Chúng ta bắt đầu với phần tổng quan về file và cấu trúc file.

1.3.1 Cấu trúc file

Tổng quan về cấu trúc file được thể hiện trong hình 1-12



Hình 1-12

Bảy file lớp được lưu trữ trong thư mục *App_Code*. Thư mục *App_Data* chứa hai cơ sở dữ liệu: cơ sở dữ liệu đăng nhập (*ASPNETDB.MDF*) và cơ sở dữ liệu *Online Diary*.

(*DiaryDB.mdf*). Các trang cần phải đăng nhập trước khi xem được lưu trong thư mục *SecureDiary*. Thư mục gốc chứa các trang đăng nhập, các trang đăng ký, và các trang nhắc mật khẩu. Về cơ bản, bạn phải đăng nhập thì mới có thể xem các thứ.

1.3.2 Đăng ký, đăng nhập, và bảo mật

Ứng dụng *Online Diary* sử dụng các điều kiểm Login mới để cung cấp các tính năng quản lý người dùng nhật ký, bao gồm: đăng ký người dùng mới, đăng nhập, và nhắc mật khẩu.

Các điều kiểm Login giúp tiết kiệm thời gian, cho phép nhiều chức năng phức tạp, chỉ cộng thêm một ít việc và hầu như không có mã lệnh nào! *ASP.NET 2.0* có bảy điều kiểm loại này:

- ✓ Login—Cho phép người dùng đăng nhập, xác nhận tên người dùng và mật khẩu.
- ✓ LoginView—Cho phép hiển thị các template khác nhau dựa vào trạng thái người dùng có đăng nhập hay không, cũng như vai trò thành viên của người đó.
- ✓ PasswordRecovery—Cung cấp chức năng nhắc mật khẩu cho những người dùng quên mật khẩu.
- ✓ LoginStatus—Hiển thị trạng thái người dùng: đăng nhập hay đăng xuất.
- ✓ LoginName—Hiển thị tên người dùng hiện tại.
- ✓ CreateUserWizard—Tạo một người dùng mới trong vài bước đơn giản.
- ✓ ChangePassword—Cho phép người dùng thay đổi mật khẩu.

Tuy nhiên, dự án *Online Diary* chỉ sử dụng các điều kiểm: Login, LoginName, CreateUserWizard, và ChangePassword.

❑ Đăng nhập

Trang *SignOn.aspx* chứa một điều kiểm Login. Cơ sở dữ liệu người dùng được tạo bằng các công cụ quản trị website. Khi hoàn tất việc này, một cơ sở dữ liệu mới có tên là *ASPNETDB.MDF* xuất hiện trong thư mục *App_Data* của dự án.

Thẻ đánh dấu của điều kiểm Login như sau:

```
<asp:Login ID="Login1" runat="server" BackColor="#F7F6F3" BorderColor="#E6E2D8"
  BorderPadding="4" BorderStyle="Solid" BorderWidth="1px"
  CreateUserText="Bạn chưa có tài khoản? Nhấp vào đây để đăng ký!"
  CreateUserUrl="~/RegisterStart.aspx" DestinationPageUrl="~/SecureDiary/DiaryMain.aspx"
  Font-Names="Verdana" Font-Size="0.8em" ForeColor="#333333" Height="197px"
  PasswordRecoveryText="Bạn quên mật khẩu?" PasswordRecoveryUrl="~/PasswordReminder.aspx"
  Style="z-index: 100; left: 78px; position: absolute; top: 55px" Width="315px">
  <LoginButtonStyle BackColor="#FFFBFF" BorderColor="CCCCCC"
    BorderStyle="Solid" BorderWidth="1px"
    Font-Names="Verdana" Font-Size="0.8em" ForeColor="#284775" />
  <TextBoxStyle Font-Size="0.8em" />
  <TitleTextStyle BackColor="#5D7B9D" Font-Bold="True"
    Font-Size="0.9em" ForeColor="White" />
  <InstructionTextStyle Font-Italic="True" ForeColor="Black" />
</asp:Login>
```

Đặc tính quan trọng cần lưu ý là *DestinationPageUrl*. Đặc tính này xác định người dùng sẽ được chuyển đến đâu khi họ nhập vào một tên người dùng và mật khẩu hợp lệ. Trong dự án *Online Diary*, đó là trang *DiaryMain.aspx*—trung tâm của giao diện *Online Diary*.

Để cho phép người dùng mới đăng ký, CreateUserText được gán bằng một thông điệp thân thiện *Bạn chưa có tài khoản? Nhấp vào đây để đăng ký!*; URL dùng để đăng ký được chỉ định trong CreateUserUrl.

Trong trường hợp người dùng đã đăng ký nhưng quên mật khẩu, đặc tính PasswordRecoveryText hiển thị thông điệp *Bạn quên mật khẩu?* và PasswordRecoveryUrl thiết lập URL mà người dùng sẽ được chuyển đến khi họ cần tìm lại mật khẩu.

Mã lệnh duy nhất mà bạn cần viết là trong sự kiện LoggedIn của điều kiểm Login, sự kiện này phát sinh khi người dùng nhập tên và mật khẩu thành công:

```
Protected Sub Login1_LoggedIn(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles Login1.LoggedIn
    Dim DiaryId As Integer = GetDiaryIdFromUserName(Login1.UserName)
    Session("DiaryId") = DiaryId
End Sub
```

Sự kiện này sử dụng tên người dùng để tìm DiaryId của người dùng trong cơ sở dữ liệu *Online Diary*. Sau đó, DiaryId được lưu trong biến Session.

Trang *SignOn.aspx* cũng cho phép đăng ký người dùng mới.

❑ Đăng ký người dùng mới

Trang *RegisterStart.aspx* xử lý việc đăng ký một người dùng mới. Giống như trang *SignOn.aspx*, trang này cũng sử dụng một điều kiểm Login mới, đó là điều kiểm CreateUserWizard. Thẻ đánh dấu của điều kiểm CreateUserWizard được trình bày trong đoạn mã sau:

```
<asp:CreateUserWizard ID="CreateUserWizard1" runat="server" BackColor="#F7F6F3"
    BorderColor="#E6E2D8" BorderStyle="Solid" BorderWidth="1px" Font-Names="Arial"
    Font-Size="Small" Style="z-index: 100; left: 58px; position: absolute; top: 43px"
    Height="164px" Width="330px" FinishDestinationPageUrl="~/SignOn.aspx"
    AnswerLabelText="Câu trả lời bảo vệ:" ConfirmPasswordLabelText="Nhập lại mật khẩu:"
    CreateUserButtonText="Tạo tài khoản" PasswordLabelText="Mật khẩu:"
    QuestionLabelText="Câu hỏi bảo vệ:" UserNameLabelText="Tên người dùng:">
    <SideBarStyle BackColor="#5D7B9D" BorderWidth="0px" Font-Size="Small"
        VerticalAlign="Top" />
    <SideBarButtonStyle BorderWidth="0px" Font-Names="Arial" ForeColor="White" />
    <NavigationButtonStyle BackColor="#FFF8FF" BorderColor="#CCCCCC" BorderStyle="Solid"
        BorderWidth="1px" Font-Names="Arial" ForeColor="#284775" />
    <HeaderStyle BackColor="#5D7B9D" BorderStyle="Solid" Font-Bold="True" Font-Size="Small"
        ForeColor="White" HorizontalAlign="Left" />
    <CreateUserButtonStyle BackColor="#FFF8FF" BorderColor="#CCCCCC" BorderStyle="Solid"
        BorderWidth="1px" Font-Names="Arial" ForeColor="#284775" Font-Size="Small" />
    <ContinueButtonStyle BackColor="#FFF8FF" BorderColor="#CCCCCC" BorderStyle="Solid"
        BorderWidth="1px" Font-Names="Arial" ForeColor="#284775" />
    <StepStyle BorderWidth="0px" />
    <TitleTextStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
    <WizardSteps>
        <asp:CreateUserWizardStep runat="server">
            <ContentTemplate>
                <table border="0" style="font-size: 100%; width: 100%; font-family: Arial;
                    height: 164px; text-align: center;">
                    <tr>
                        <td align="center" colspan="2" style="font-weight: bold; color: white;
                            background-color: #5d7b9d; height: 24px;">Đăng ký tài khoản mới</td>
                    </tr>
                    <tr>
                        <td align="right">
                            <asp:Label ID="UserNameLabel" runat="server"
                                AssociatedControlID="UserName">Tên người dùng:</asp:Label>
                        </td>
```

```

        <td align="left">
            <asp:TextBox ID="UserName" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="UserNameRequired"
                runat="server" ControlToValidate="UserName"
                ErrorMessage="User Name is required."
                ToolTip="User Name is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td align="right">
            <asp:Label ID="PasswordLabel" runat="server"
                AssociatedControlID="Password">Mật khẩu:</asp:Label>
        </td>
        <td align="left">
            <asp:TextBox ID="Password" runat="server"
                TextMode="Password"></asp:TextBox>
            <asp:RequiredFieldValidator ID="PasswordRequired" runat="server"
                ControlToValidate="Password"
                ErrorMessage="Password is required."
                ToolTip="Password is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td align="right">
            <asp:Label ID="ConfirmPasswordLabel" runat="server"
                AssociatedControlID="ConfirmPassword">Nhập lại mật khẩu:</asp:Label>
        </td>
        <td align="left">
            <asp:TextBox ID="ConfirmPassword" runat="server"
                TextMode="Password"></asp:TextBox>
            <asp:RequiredFieldValidator ID="ConfirmPasswordRequired"
                runat="server" ControlToValidate="ConfirmPassword"
                ErrorMessage="Confirm Password is required."
                ToolTip="Confirm Password is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td align="right">
            <asp:Label ID="EmailLabel" runat="server"
                AssociatedControlID="Email">E-mail:</asp:Label>
        </td>
        <td align="left">
            <asp:TextBox ID="Email" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="EmailRequired" runat="server"
                ControlToValidate="Email" ErrorMessage="E-mail is required."
                ToolTip="E-mail is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td align="right">
            <asp:Label ID="QuestionLabel" runat="server"
                AssociatedControlID="Question">Câu hỏi mật:</asp:Label></td>
        <td align="left">
            <asp:TextBox ID="Question" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="QuestionRequired" runat="server"
                ControlToValidate="Question"
                ErrorMessage="Security question is required."
                ToolTip="Security question is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>

```

```

        <td align="right">
            <asp:Label ID="AnswerLabel" runat="server"
                AssociatedControlID="Answer">Câu trả lời mật:</asp:Label>
        </td>
        <td align="left">
            <asp:TextBox ID="Answer" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="AnswerRequired" runat="server"
                ControlToValidate="Answer"
                ErrorMessage="Security answer is required."
                ToolTip="Security answer is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
        </td>
    </tr>
</tr>
<tr>
    <td align="center" colspan="2">
        <asp:CompareValidator ID="PasswordCompare" runat="server"
            ControlToCompare="Password" ControlToValidate="ConfirmPassword"
            Display="Dynamic" ErrorMessage="Hai mật khẩu không trùng nhau."
            ValidationGroup="CreateUserWizard1"></asp:CompareValidator>
    </td>
</tr>
<tr>
    <td align="center" colspan="2" style="color: red">
        <asp:Literal ID="ErrorMessage" runat="server"
            EnableViewState="False"></asp:Literal>
    </td>
</tr>
</table>
</ContentTemplate>
</asp:CreateUserWizardStep>
<asp:WizardStep ID="personalDetailsStep" runat="server" Title="User Details">
    <table border="0" style="font-size: 100%; font-family: Arial; z-index: 100;
        left: 0px; position: absolute; top: 0px;" width="100%">
        <tr>
            <td align="center" colspan="2" style="font-weight: bold;
                color: white; background-color: #5d7b9d; height: 24px;">
                Thông tin cá nhân
            </td>
        </tr>
        <tr>
            <td align="right" style="height: 26px; width: 162px;">
                <label for="UserName">
                    Tên:</label></td>
            <td style="width: 235px; height: 26px">
                <asp:TextBox ID="firstNameTextBox" runat="server"
                    CausesValidation="True"></asp:TextBox>&nbsp;
            </td>
        </tr>
        <tr>
            <td align="right" style="width: 162px">
                <label for="Password">
                    Họ và chữ lót:</label></td>
            <td style="width: 235px">
                <asp:TextBox ID="lastNameTextBox" runat="server"
                    CausesValidation="True"></asp:TextBox>&nbsp;
            </td>
        </tr>
        <tr>
            <td align="center" colspan="2" style="height: 18px">&nbsp;</td>
        </tr>
        <tr>
            <td align="center" colspan="2" style="color: red">&nbsp;</td>
        </tr>
    </table>
</asp:WizardStep>
<asp:CompleteWizardStep runat="server">

```

```

<ContentTemplate>
<table border="0" style="font-size: 100%; font-family: Arial; z-index: 100;
left: 0px; position: absolute; top: 0px;" width="100%">
  <tr>
    <td align="center" colspan="2" style="font-weight: bold; color: white;
background-color: #5d7b9d; text-align: center;">Hoàn tất</td>
  </tr>
  <tr>
    <td style="text-align: center;">Tài khoản của bạn đã được tạo.</td>
  </tr>
  <tr>
    <td align="right" colspan="2">
      <asp:Button ID="ContinueButton" runat="server" BackColor="#FFBFFF"
        BorderColor="#CCCCC" BorderStyle="Solid" BorderWidth="1px"
        CausesValidation="False" CommandName="Continue"
        Font-Names="Arial" ForeColor="#284775" Text="Continue"
        ValidationGroup="CreateUserWizard1" />
    </td>
  </tr>
</table>
</ContentTemplate>
</asp:CompleteWizardStep>
</WizardSteps>
</asp:CreateUserWizard>

```

Hầu hết thẻ đánh dấu và các đặc tính liên quan đến các thiết lập style. Tuy nhiên, đặc tính quan trọng nhất là `FinishDestinationPageUrl`. Đây là nơi người dùng được chuyển đến khi quá trình đăng ký hoàn tất. Trong *Online Diary*, đó là trang *SignOn.aspx*.

Bạn có thể thấy một số thẻ `WizardStep` trong thẻ đánh dấu giống như sau:

```
<asp:WizardStep ID="personalDetailsStep" runat="server" Title="User Details">
```

`CreateUserWizard` làm việc trên cơ sở từng bước một. Phải có ít nhất một bước cho phép người dùng chọn tên đăng nhập, mật khẩu, và câu hỏi mật (xem hình 1-13).

Bước này và các style của nó có thể được hiệu chỉnh (hình 1-13 hiển thị các giá trị đã được Việt hóa). Điều kiểm này đảm nhận việc chèn dữ liệu người dùng mới vào cơ sở dữ liệu người dùng.

The screenshot shows a web form for creating a new account. The title is 'Online Diary' and the subtitle is 'Đăng ký tài khoản mới'. The form includes the following fields and labels:

- Tên người dùng: (Username)
- Mật khẩu: (Password)
- Nhập lại mật khẩu: (Repeat Password)
- E-mail:
- Câu hỏi mật: (Security Question)
- Câu trả lời mật: (Security Answer)
- Tạo tài khoản (Create Account) button

Hình 1-13

Bước thứ hai (xem hình 1-14) được hiển thị sau khi người dùng được tạo.

Hình 1-14

Màn hình này yêu cầu họ tên của người dùng. Khi đó, tùy bạn lưu dữ liệu ở đâu, bạn thực hiện điều này trong sự kiện `FinishButtonClick` của điều khiển `CreateUserWizard`:

```
Protected Sub CreateUserWizard1_FinishButtonClick(ByVal sender As Object, _
    ByVal e As System.Web.UI.WebControls.WizardNavigationEventArgs) _
    Handles CreateUserWizard1.FinishButtonClick
    Dim myTextBox As TextBox
    Dim UserName, FirstName, LastName
    myTextBox = CreateUserWizard1.FindControl("firstNameTextBox")
    FirstName = myTextBox.Text
    myTextBox = CreateUserWizard1.FindControl("lastNameTextBox")
    LastName = myTextBox.Text
    UserName = CreateUserWizard1.UserName
    OnlineDiary.InsertDiary(UserName, FirstName, LastName)
End Sub
```

Bước này tạo một nhật ký mới và lưu trữ họ tên của người dùng. `UserName` được lấy từ thuộc tính `UserName` của điều khiển `CreateUserWizard`, phương thức chia sẻ `InsertDiary()` được sử dụng để chèn người dùng mới vào cơ sở dữ liệu *Online Diary*.

Người ta đôi lúc cũng quên mất mật khẩu của mình. May mắn thay, *ASP.NET 2.0* có khả năng nhắc mật khẩu.

❑ Nhắc mật khẩu

Hầu như chẳng cần mã lệnh nào, bạn có thể tạo tính năng nhắc mật khẩu cho *Online Diary* bằng điều khiển `PasswordRecovery`. Hầu như tất cả các thiết lập của nó là các giá trị mặc định hoặc có liên quan đến style. Chỉ có một dòng mã trong sự kiện `SendingMail`:

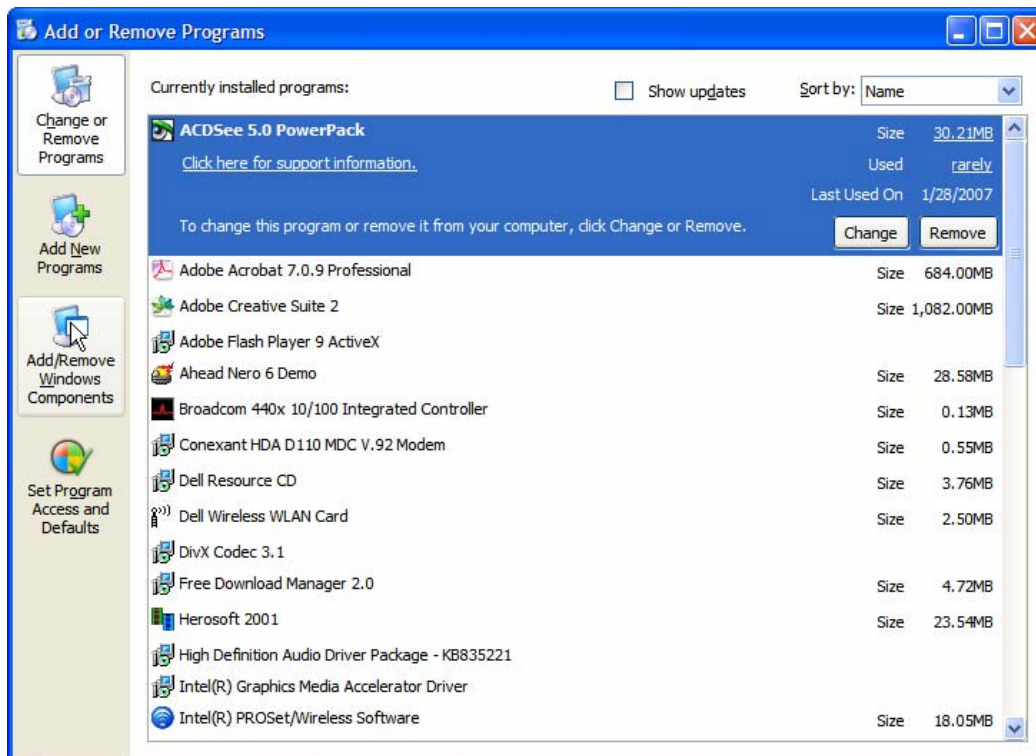
```
Protected Sub PasswordRecovery1_SendingMail(ByVal sender As Object, _
    ByVal e As System.Web.UI.WebControls.MailMessageEventArgs) _
    Handles PasswordRecovery1.SendingMail
    returnToLogOnHyperLink.Visible = True
End Sub
```

Sự kiện `SendingMail` phát sinh khi người dùng nhấn nút *Gửi e-mail* và hiển thị liên kết *Trở về màn hình đăng nhập* (người dùng không phải đoán tiếp theo sẽ đi đâu).

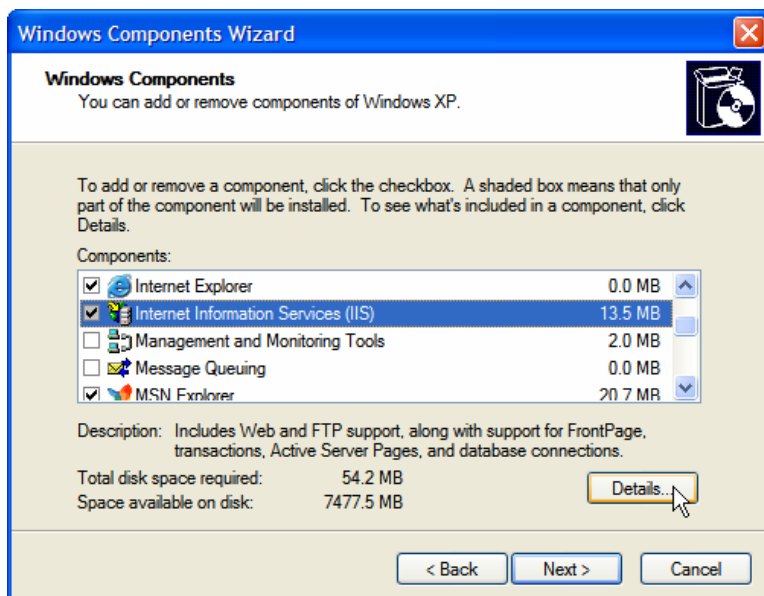
Công việc chính là cấu hình *SMTP Server* để gửi e-mail nhắc mật khẩu. *Visual Web Developer* không có *SMTP Server*. Tuy nhiên, bạn có thể sử dụng dịch vụ *SMTP* của *IIS* trong *Windows XP/2000*. Các bước cài đặt như sau:

1. Vào *Start | Control Panel | Add or Remove Programs*.

2. Trong hộp thoại *Add or Remove Programs*, chọn *Add/Remove Windows Components*.

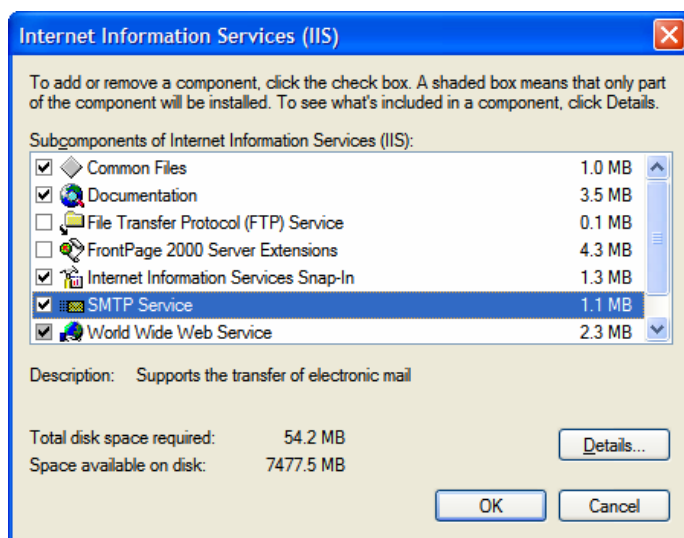


3. Trong hộp thoại *Windows Components Wizard*, chọn *Internet Information Server (IIS)* rồi nhấp nút *Details*.



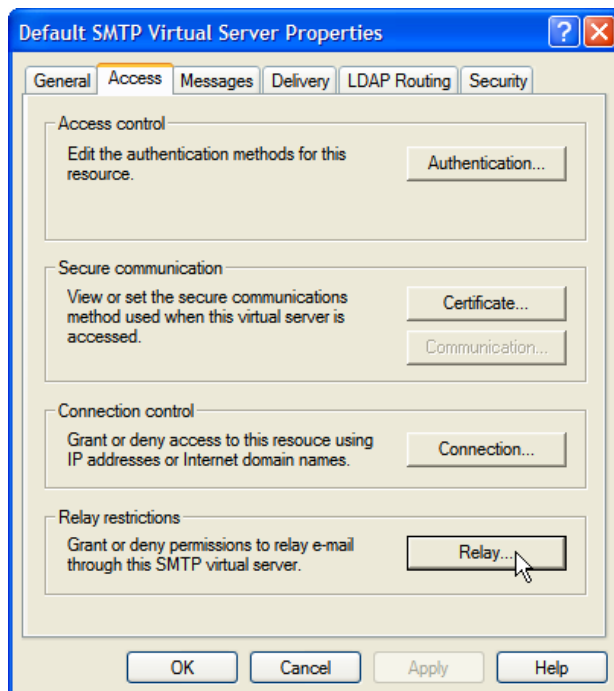
Hình 1-15

4. Trong hộp thoại *Internet Information Server (IIS)*, đánh dấu chọn *SMTP Service* rồi nhấp *OK*.



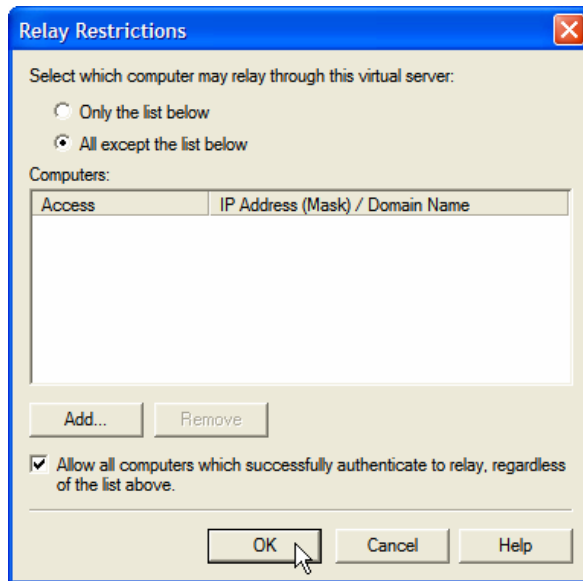
Hình 1-16

5. Trong hộp thoại *Windows Components Wizard*, nhấp *Next* để cài tiến hành đặt dịch vụ *SMTP*.
6. Vào *Start | Control Panel | Administrative Tools | Internet Information Services*.
7. Trong *IIS*, nhấp phải vào *Default SMTP Virtual Server* và chọn *Properties*.
8. Trong hộp thoại *Default SMTP Virtual Server Properties*, chọn thẻ *Access* rồi nhấp nút *Relay*.



Hình 1-17

9. Trong hộp thoại *Relay Restrictions*, chọn *All except the list below* rồi nhấp *OK*.



Hình 1-18

Khi dịch vụ *SMTP* đã được cài đặt, thêm phần mã in đậm vào giữa thẻ `<configuration>` trong file *Web.config*:

```
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <connectionStrings>
    <add name="DiaryDBConnectionString" connectionString=
      "Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\DiaryDB.mdf;
      Integrated Security=True;User Instance=True"
      providerName="System.Data.SqlClient" />
    </connectionStrings>
  <system.web>
    <roleManager enabled="true" />
    <authentication mode="Forms"/>
    <compilation debug="true"/>
  </system.web>
  <system.net>
    <mailSettings>
      <smtp from="admin@OnlineDiary.com">
        <network host="localhost" password="" userName="" />
      </smtp>
    </mailSettings>
  </system.net>
</configuration>
```

1.3.3 Xem lịch biểu trực tuyến

Trang *DiaryMain.aspx* là trung tâm của ứng dụng. Nó hiển thị một lịch biểu của tháng hiện tại, cho biết những ngày nào có các sự kiện hay các mục nhật ký tương ứng với chúng. Nó cũng hiển thị một danh sách các sự kiện sắp đến và các mục nhật ký cho tháng hiện tại.

Để hiển thị khi một ngày có các sự kiện hoặc một mục nhật ký, sự kiện `OnDayRender` của điều khiển `Calendar` được sử dụng:

```
Protected Sub Calendar1_OnDayRender(ByVal sender As Object, _
  ByVal e As System.Web.UI.WebControls.DayRenderEventArgs) Handles Calendar1.DayRender
  If Not e.Day.IsOtherMonth Then
    If entryArrayOfDays Is Nothing Then
      entryArrayOfDays = GetDaysInMonthWithEntries(Session("DiaryId"), _
        e.Day.Date.Month, e.Day.Date.Year)
    End If
```

```

If eventArrayOfDays Is Nothing Then
    eventArrayOfDays = GetDaysInMonthWithEvents(Session("DiaryId"), _
        e.Day.Date.Month, e.Day.Date.Year)
End If
If entryArrayOfDays(CInt(e.Day.DayNumberText)) Then
    e.Cell.BackColor = Drawing.Color.Blue
End If
If eventArrayOfDays(CInt(e.Day.DayNumberText)) Then
    e.Cell.ForeColor = Drawing.Color.Red
End If
End If
End Sub

```

Hai khối `If` đầu tiên trong đoạn mã trên đảm bảo `entryArrayOfDays` và `eventArrayOfDays` chứa dữ liệu về các ngày có mục nhật ký hay sự kiện. Chúng đều là mảng `Boolean`; nếu một ngày có mục nhật ký hay sự kiện, phần tử mảng cho ngày hôm đó chứa giá trị `True`. Các mảng này nhận dữ liệu từ các hàm chia sẻ `GetDaysInMonthWithEntries()` và `GetDaysInMonthWithEvents()` của lớp `DiaryEntry` và `DiaryEvent`.

Trong hai khối `If` tiếp theo, mã lệnh kiểm tra các ngày trong tháng có được thể hiện là có mục nhật ký hay sự kiện không. Nếu có sự kiện, phần văn bản của ngày đó được tô màu đỏ. Nếu có mục nhật ký, phần nền của ngày đó được tô màu xanh.

Ngoài điều kiểm `Calendar`, trang chính còn có hai điều kiểm `GridView`. `GridView` phía trên hiển thị các sự kiện sắp đến, `GridView` phía dưới hiển thị các mục nhật ký gần đây. Cả hai điều kiểm này đều lấy dữ liệu từ điều kiểm `ObjectDataSource`—một điều kiểm mới trong *ASP.NET 2.0*. Trước đây, các điều kiểm nguồn dữ liệu tương tác trực tiếp với cơ sở dữ liệu, rất dễ sử dụng—chỉ cần đặt lên một trang, thiết lập vài thuộc tính,... Tuy nhiên, đó không phải là một lối viết mã hay. Việc tách tầng truy xuất dữ liệu, tầng nghiệp vụ, và tầng trình bày là một lối tốt đã được xem xét, nhưng như thế phải bỏ đi các điều kiểm nguồn dữ liệu dễ-sử-dụng!

Tuy nhiên, `ObjectDataSource` cho phép bạn được cả hai: các điều kiểm dữ liệu dễ-sử-dụng và sử dụng các lớp để phân tách tầng nghiệp vụ, tầng dữ liệu, và tầng trình bày. Thay vì kết nối trực tiếp đến cơ sở dữ liệu, `ObjectDataSource` lấy dữ liệu từ một trong các lớp này. Ví dụ, `diaryEntriesObjectDataSource` trên *DiaryMain.aspx* lấy dữ liệu từ phương thức `GetDiaryEntriesRecentlyChanged()` của lớp `DiaryEntry`, thẻ đánh dấu của nó như sau:

```

<asp:ObjectDataSource ID="diaryEntriesObjectDataSource" runat="server"
    SelectMethod="GetDiaryEntriesRecentlyChanged" TypeName="DiaryEntry">
    <SelectParameters>
        <asp:SessionParameter DefaultValue="-1" Name="DiaryId"
            SessionField="DiaryId" Type="Int32" />
    </SelectParameters>
</asp:ObjectDataSource>

```

Đặc tính `TypeName` chỉ định tên lớp, và đặc tính `SelectMethod` chỉ định phương thức nào của lớp đó sẽ cung cấp dữ liệu. `GetDiaryEntriesRecentlyChanged()` là phương thức chia sẻ:

```

Public Shared Function GetDiaryEntriesRecentlyChanged(ByVal DiaryId As Integer) _
    As SqlDataReader
    Dim diaryDBConn As New SqlConnection(connectionString)
    Dim sqlString As String = "GetRecentDiaryEntries"
    Dim sqlCommand As New SqlCommand(sqlString, diaryDBConn)
    sqlCommand.CommandType = CommandType.StoredProcedure

    sqlCommand.Parameters.AddWithValue("@DiaryId", DiaryId)

    diaryDBConn.Open()
    Dim entrySQLDR As SqlDataReader = sqlCommand.ExecuteReader(CommandBehavior.CloseConnection)
    sqlCommand = Nothing

```

Chương 1: Online Diary

```
Return entrySQLDR
End Function
```

Phương thức này trả về một đối tượng SqlDataReader gồm những dữ liệu mà điều khiển ObjectDataSource sẽ sử dụng.

Việc hiển thị dữ liệu chỉ là vấn đề trở một điều khiển vào ObjectDataSource:

```
<asp:GridView ID="recentEntriesGridView" runat="server" AutoGenerateColumns="False"
Caption="Các mục nhật ký gần đây" CaptionAlign="Left" CellPadding="4"
DataSourceID="diaryEntriesObjectDataSource"
ForeColor="#333333" GridLines="None" Style="z-index: 105; left: 535px;
position: absolute; top: 321px" Width="476px" Height="208px">
  <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
  <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
  <Columns>
    <asp:BoundField DataField="EntryDate" />
    <asp:BoundField DataField="EntryTitle" />
    <asp:BoundField DataField="EntryText" />
  </Columns>
  <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
  <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />
  <HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
  <EditRowStyle BackColor="#999999" />
  <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
</asp:GridView>
```

Trong thẻ đánh dấu của điều khiển GridView, đặc tính DataSourceID chỉ định nguồn dữ liệu—đó là điều khiển ObjectDataSource. Ngoài ra, thẻ đánh dấu này chỉ định những cột nào được hiển thị bằng cách thiết lập AutoGenerateColumns là False. Bước cuối cùng là tạo danh sách các cột:

```
<Columns>
  <asp:BoundField DataField="EntryDate" />
  <asp:BoundField DataField="EntryTitle" />
  <asp:BoundField DataField="EntryText" />
</Columns>
```

Ngoài việc cho phép hiển thị dữ liệu, điều khiển ObjectDataSource cũng có thể cập nhật, chèn, và xóa các mẫu tin trong cơ sở dữ liệu.

1.3.4 Tạo, chỉnh sửa, và xem một mục nhật ký

Trang *DayView.aspx* cho phép chỉnh sửa nhật ký. Trang này chứa một form cho phép bạn nhập tiêu đề và nội dung mục nhật ký. Nó cũng hiển thị bất cứ mục nhật ký nào đang tồn tại.

Tất cả công việc được hoàn tất bằng cách sử dụng lớp DiaryEntry. Sự kiện Page_Load tạo một lớp DiaryEntry mới, truyền cho phương thức khởi dựng DiaryId của người dùng hiện tại cũng như ngày mà trang này nhắc đến:

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
  mDiaryEntry = New DiaryEntry(CInt(Session("DiaryId")), CDate(dayShownLabel.Text))
  changeDayCalendar.SelectedDate = CDate(dayShownLabel.Text)
  changeDayCalendar.VisibleDate = changeDayCalendar.SelectedDate
  If Not IsPostBack Then
    entryTextTextBox.Text = mDiaryEntry.EntryText
    entryTitleTextBox.Text = mDiaryEntry.EntryTitle
  End If
End Sub
```

mDiaryEntry là một biến toàn cục, được sử dụng để chứa đối tượng DiaryEntry ứng ngày đang được chỉnh sửa.

Phương thức khởi dựng, được trình bày trong đoạn mã sau, thực hiện lấy dữ liệu:

```

Public Sub New(ByVal DiaryId As Integer, ByVal EntryDate As Date)
    mDiaryId = DiaryId
    If mDiaryId > 0 Then

        Try
            Dim diaryDBConn As New SqlConnection(conString)
            Dim sqlString As String = "GetDiaryEntryByDate"
            Dim sqlCommand As New SqlCommand(sqlString, diaryDBConn)
            sqlCommand.CommandType = CommandType.StoredProcedure

            sqlCommand.Parameters.AddWithValue("@DiaryId", mDiaryId)
            sqlCommand.Parameters.AddWithValue("@EntryFromDate", EntryDate)
            sqlCommand.Parameters.AddWithValue("@EntryToDate", EntryDate)

            diaryDBConn.Open()
            Dim diaryEntrySQLDR As SqlDataReader = _
                sqlCommand.ExecuteReader(CommandBehavior.CloseConnection)
            sqlCommand = Nothing

            If diaryEntrySQLDR.Read() Then
                mDiaryEntryId = CLng(diaryEntrySQLDR("DiaryEntryId"))
                mEntryDate = CDate(diaryEntrySQLDR("EntryDate"))
                mEntryTitle = diaryEntrySQLDR("EntryTitle").ToString
                mEntryText = diaryEntrySQLDR("EntryText").ToString
            Else
                mDiaryEntryId = -1
                mEntryDate = EntryDate
            End If

            diaryEntrySQLDR.Close()
            diaryEntrySQLDR = Nothing
            diaryDBConn.Close()
            diaryDBConn = Nothing

        Catch ex As Exception
            mDiaryEntryId = -1
        End Try

    End If
End Sub

```

Thủ tục tồn trữ GetDiaryEntryByDate được gọi để lấy dữ liệu. Nếu chưa có mục nhật ký nào cho ngày hôm đó, mDiaryEntryId được gán -1 và tất cả các thuộc tính khác được gán các giá trị mặc định. Ngược lại, chúng nhận dữ liệu từ cơ sở dữ liệu.

Khi hộp tiêu đề hoặc hộp nội dung mục nhật ký thay đổi, mDiaryEntry sẽ được cập nhật:

```

Protected Sub entryTitleTextBox_TextChanged(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles entryTitleTextBox.TextChanged
    mDiaryEntry.EntryTitle = entryTitleTextBox.Text
End Sub

Protected Sub entryTextTextBox_TextChanged(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles entryTextTextBox.TextChanged
    mDiaryEntry.EntryText = entryTextTextBox.Text
End Sub

```

Việc lưu các thay đổi sẽ diễn ra khi bạn nhấp nút *Lưu nhật ký*:

```

Protected Sub saveDiaryEntryButton_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles saveDiaryEntryButton.Click
    mDiaryEntry.Save()
End Sub

```

Mọi thứ liên quan đến việc gọi phương thức Save() của đối tượng DiaryEntry:

Chương 1: Online Diary

```
Public Sub Save()
    If mDiaryEntryId = -1 Then
        InsertNewDiaryEntry()
    Else
        UpdateDiaryEntry()
    End If
End Sub
```

Dựa vào mDiaryEntryId có bằng -1 hay không, phương thức này chèn một mục nhật ký mới vào cơ sở dữ liệu hoặc cập nhật mục nhật ký đã tồn tại. Phương thức riêng InsertNewDiaryEntry() chèn một mục nhật ký mới:

```
Private Sub InsertNewDiaryEntry()
    If mDiaryId <> -1 Then
        Dim diaryDBConn As New SqlConnection(connectionString)
        Dim sqlString As String = "InsertDiaryEntry"
        Dim sqlCommand As New SqlCommand(sqlString, diaryDBConn)
        sqlCommand.CommandType = CommandType.StoredProcedure

        sqlCommand.Parameters.AddWithValue("@DiaryId", mDiaryId)
        sqlCommand.Parameters.AddWithValue("@EntryDate", mEntryDate)
        sqlCommand.Parameters.AddWithValue("@EntryTitle", mEntryTitle)
        sqlCommand.Parameters.AddWithValue("@EntryText", mEntryText)
        sqlCommand.Parameters.Add("@NewDiaryEntryId", SqlDbType.BigInt)
        sqlCommand.Parameters("@NewDiaryEntryId").Direction = ParameterDirection.ReturnValue

        diaryDBConn.Open()
        sqlCommand.ExecuteNonQuery()
        mDiaryEntryId = CLng(sqlCommand.Parameters("@NewDiaryEntryId").Value())

        diaryDBConn.Close()
        sqlCommand = Nothing
        diaryDBConn = Nothing
    End If
End Sub
```

Phương thức riêng UpdateDiaryEntry() cập nhật mục nhật ký đã tồn tại:

```
Private Sub UpdateDiaryEntry()
    If mDiaryEntryId <> -1 Then
        Dim diaryDBConn As New SqlConnection(connectionString)
        Dim sqlString As String = "UpdateDiaryEntry"
        Dim sqlCommand As New SqlCommand(sqlString, diaryDBConn)
        sqlCommand.CommandType = CommandType.StoredProcedure

        sqlCommand.Parameters.AddWithValue("@DiaryEntryId", mDiaryEntryId)
        sqlCommand.Parameters.AddWithValue("@EntryDate", mEntryDate)
        sqlCommand.Parameters.AddWithValue("@EntryTitle", mEntryTitle)
        sqlCommand.Parameters.AddWithValue("@EntryText", mEntryText)

        diaryDBConn.Open()
        sqlCommand.ExecuteNonQuery()
        diaryDBConn.Close()
        sqlCommand = Nothing
        diaryDBConn = Nothing
    End If
End Sub
```

Phần tiếp theo sẽ thảo luận về mã lệnh chỉnh sửa, xem, và xóa các sự kiện.

1.3.5 Tạo, chỉnh sửa, và xem các sự kiện nhật ký

Các sự kiện được tạo bằng cách nhấp vào liên kết *Thêm sự kiện mới* trên trang *DayView.aspx*. Liên kết này sẽ dẫn bạn đến một form trên trang *AddEvent.aspx*. Khi nút *Lưu sự kiện* được nhấp, sự kiện Click sẽ tạo một đối tượng DiaryEvent mới với dữ liệu của các thuộc tính được

lấy từ form, và sau đó gọi phương thức `Save()`. Dòng chảy mã lệnh giống hệt phương thức `Save()` của đối tượng `DiaryEvent`. Những chỗ mà chức năng giống nhau hoặc tương tự nhau, tên phương thức trên các đối tượng khác nhau được giữ giống nhau. Điều này giảm thiểu nhầm lẫn và khiến mọi thứ dễ dàng hơn.

Tất cả các sự kiện liên quan đến một ngày cụ thể được trình bày trong trang *DayView.aspx*. Điều kiểm `ObjectDataSource` trên trang *DayView.aspx* lấy dữ liệu từ phương thức chia sẻ `GetDiaryEventsByDate()` của đối tượng `DiaryEvent`. Thẻ đánh dấu cho điều kiểm `ObjectDataSource` như sau:

```
<asp:ObjectDataSource ID="eventsObjectDataSource" runat="server"
    SelectMethod="GetDiaryEventsByDate" TypeName="DiaryEvent" DeleteMethod="DeleteEvent">
    <SelectParameters>
        <asp:SessionParameter DefaultValue="-1" Name="DiaryId"
            SessionField="DiaryId" Type="Int32" />
        <asp:ControlParameter ControlID="dayShownLabel" DefaultValue=""
            Name="FromDate" PropertyName="Text" Type="DateTime" />
        <asp:ControlParameter ControlID="dayShownLabel" DefaultValue=""
            Name="ToDate" PropertyName="Text" Type="DateTime" />
        <asp:Parameter DefaultValue="0" Name="MaxRows" Type="Int32" />
    </SelectParameters>
    <DeleteParameters>
        <asp:Parameter Name="EventId" Type="Int64" />
    </DeleteParameters>
</asp:ObjectDataSource>
```

Chú ý rằng `SelectParameters` và `DeleteParameters` được thiết lập để chỉ định dữ liệu truyền cho phương thức `GetDiaryEventsByDate()` (dùng để thu lấy dữ liệu) và phương thức `DeleteEvent()` (dùng để xóa các sự kiện nhật ký).

Điều kiểm `GridView` được kết với `ObjectDataSource` trong đoạn mã trên:

```
<asp:GridView ID="eventsGridView" runat="server" AutoGenerateColumns="False"
    CellPadding="4" DataSourceID="eventsObjectDataSource" ForeColor="#333333"
    GridLines="None" Height="1px" PageSize="5" Style="z-index: 108; left: 78px;
    position: absolute; top: 357px" Width="542px" DataKeyNames="EventId">
    <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
    <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
    <Columns>
        <asp:HyperLinkField DataNavigateUrlFields="EventId" Text="Sửa"
            DataNavigateUrlFormatString="~/SecureDiary/EditEvent.aspx?EventId={0}" />
        <asp:CommandField ShowDeleteButton="True" DeleteText="Xóa" />
        <asp:BoundField DataField="EventName" HeaderText="Sự kiện" />
        <asp:BoundField DataField="EventDescription" HeaderText="Mô tả" />
    </Columns>
    <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
    <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />
    <HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
    <EditRowStyle BackColor="#999999" />
    <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
</asp:GridView>
```

Ngoài ra, tham số `AutoGenerateColumns` được thiết lập là `False`, và các cột được chỉ định như sau:

```
<Columns>
    <asp:HyperLinkField DataNavigateUrlFields="EventId" Text="Sửa"
        DataNavigateUrlFormatString="~/SecureDiary/EditEvent.aspx?EventId={0}" />
    <asp:CommandField ShowDeleteButton="True" DeleteText="Xóa" />
    <asp:BoundField DataField="EventName" HeaderText="Sự kiện" />
    <asp:BoundField DataField="EventDescription" HeaderText="Mô tả" />
</Columns>
```

Chương 1: Online Diary

Liên kết này sẽ dẫn người dùng đến trang *EditEvent.aspx*. URL chứa thông số chuỗi truy vấn *EventId*, ban đầu được gán {0}, lúc thực thi sẽ được thay bằng giá trị của cột thứ nhất đối với mỗi dòng trong *DataSet*.

Ngoài ra, mã lệnh cũng chỉ định một nút *Xóa* trên mỗi dòng trong khung lưới:

```
<asp:CommandField ShowDeleteButton="True" DeleteText="Xóa" />
```

Khi bạn nhấp vào nút *Xóa*, điều kiểm *GridView* sẽ yêu cầu điều kiểm *ObjectDataSource* gọi phương thức *DeleteEvent()* của lớp *DiaryEvent*. Đặc tính *DataKeyNames* trong thẻ đánh dấu của điều kiểm *GridView* chỉ định trường khóa chính cần được sử dụng để xóa dòng.

Trở lại việc chỉnh sửa sự kiện: khi nhấp vào liên kết *Sửa*, bạn được đưa đến trang *EditEvent.aspx*. *EventId* của liên kết *Sửa* được truyền làm một thông số URL. Trang *EditEvent.aspx* hầu như giống hệt trang *AddEvent.aspx* đã được thảo luận trước đây. Điểm khác biệt chính là khi khởi tạo trang. Hàm thụ lý sự kiện *Page_Init* được trình bày trong đoạn mã sau, và đây là nơi mà các chi tiết của sự kiện được nhập vào form:

```
Protected Sub Page_Init(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Init
```

```
    Dim EventBeingEdited As New DiaryEvent(CLng(Request.QueryString("EventId")))
    eventNameTextBox.Text = EventBeingEdited.EventName
    eventDescriptionTextBox.Text = EventBeingEdited.EventDescription
    dayShownLabel.Text = EventBeingEdited.EventDate.Day & " " & _
        MonthName(EventBeingEdited.EventDate.Month) & " " & EventBeingEdited.EventDate.Year

    Dim NewListItem As ListItem, HourCount, MinuteCount

    For HourCount = 0 To 23
        If HourCount < 10 Then
            NewListItem = New ListItem("0" & HourCount, HourCount.ToString)
        Else
            NewListItem = New ListItem(HourCount.ToString, HourCount.ToString)
        End If
        If EventBeingEdited.EventDate.Hour = HourCount Then
            NewListItem.Selected = True
        End If
        StartHourDropDownList.Items.Add(NewListItem)
    Next

    For MinuteCount = 0 To 59
        If MinuteCount < 10 Then
            NewListItem = New ListItem("0" & MinuteCount.ToString, MinuteCount.ToString)
        Else
            NewListItem = New ListItem(MinuteCount.ToString, MinuteCount.ToString)
        End If
        If EventBeingEdited.EventDate.Minute = MinuteCount Then
            NewListItem.Selected = True
        End If
        StartMinuteDropDownList.Items.Add(NewListItem)
    Next

    Dim itemToSelect As ListItem
    itemToSelect = _
        eventDurationDropDownList.Items.FindByValue(EventBeingEdited.EventDuration.ToString())
    itemToSelect.Selected = True

    EventBeingEdited = Nothing
End Sub
```


EventId được trích từ các thông số URL và được sử dụng để tạo một đối tượng DiaryEvent mới. Việc đưa dữ liệu vào các textbox của sự kiện khá dễ dàng, nhưng các chi tiết về thời gian và thời lượng của sự kiện thì liên quan đến việc đưa dữ liệu vào các hộp xổ xuống giờ và phút, và phải đảm bảo giá trị chính xác được chọn. Điều này được thực hiện bằng cách lập giờ từ 0-23 và phút từ 0-59.

1.3.6 Quản lý danh bạ liên lạc

Quản lý danh bạ liên lạc là phần cuối cùng của *Online Diary* mà bạn sẽ khảo sát, và phần này sử dụng những nguyên tắc giống như các phần khác. *YourContacts.aspx* là trang quản lý danh bạ liên lạc chính. Tại đây, danh sách các liên lạc hiện tại được hiển thị, bao gồm các tùy chọn: thêm, chỉnh sửa, và xóa liên lạc.

Tất cả các liên lạc được hiển thị bằng điều khiển DataSource và GridView; các nguyên tắc cũng giống như khi hiển thị, xóa, và chỉnh sửa các sự kiện nhật ký. Lúc này, lớp Contact được sử dụng cho việc chỉnh sửa và hiển thị thông tin liên lạc.

Trang chính dùng để hiển thị các liên lạc là *YourContacts.aspx*, trong đó điều khiển GridView liệt kê tất cả các liên lạc hiện có:

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" CellPadding="4"
    DataSourceID="ObjectDataSource1" ForeColor="#333333" GridLines="None"
    Style="z-index: 101; left: 36px; position: absolute; top: 137px" DataKeyNames="ContactId">
    <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
    <Columns>
        <asp:CommandField ShowDeleteButton="True" DeleteText="Xóa" />
        <asp:HyperLinkField DataNavigateUrlFields="ContactId"
            DataNavigateUrlFormatString="~/SecureDiary/EditContact.aspx?ContactId={0}"
            Text="Sửa" />
        <asp:BoundField DataField="LastName" HeaderText="Họ và chữ lót" />
        <asp:BoundField DataField="FirstName" HeaderText="Tên" />
        <asp:BoundField DataField="Telephone" HeaderText="Điện thoại" />
        <asp:BoundField DataField="Email" HeaderText="E-mail" />
    </Columns>
    <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
    <EditRowStyle BackColor="#999999" />
    <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />
    <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
    <HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
    <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
</asp:GridView>
```

Nó lấy dữ liệu từ điều khiển ObjectDataSource ObjectDataSource1, đến phiên điều khiển này kết nối đến phương thức chia sẻ GetContactByFirstLetter() của lớp Contact:

```
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
    SelectMethod="GetContactsByFirstLetter" TypeName="Contact" DeleteMethod="DeleteContact">
    <SelectParameters>
        <asp:SessionParameter DefaultValue="6" Name="DiaryId"
            SessionField="DiaryId" Type="Int32" />
        <asp:Parameter Name="FirstLetterOfSurname" Type="Char" />
    </SelectParameters>
    <DeleteParameters>
        <asp:ControlParameter ControlID="GridView1" Name="ContactId"
            PropertyName="SelectedValue" Type="Int64" />
    </DeleteParameters>
</asp:ObjectDataSource>
```

Thông số DeleteMethod của điều khiển ObjectDataSource cũng được gắn với DeleteContact của lớp Contact. Điều khiển GridView đã được thiết lập hiển thị một liên kết để xóa mỗi liên lạc, và phương thức này thực hiện thao tác xóa:

Chương 1: Online Diary

```
Public Shared Sub DeleteContact(ByVal ContactId As Long)
    Dim diaryDBConn As New SqlConnection(connectionString)
    Dim sqlString As String = "DeleteContact"
    Dim sqlCommand As New SqlCommand(sqlString, diaryDBConn)
    sqlCommand.CommandType = CommandType.StoredProcedure

    sqlCommand.Parameters.AddWithValue("@ContactId", ContactId)
    diaryDBConn.Open()
    sqlCommand.ExecuteNonQuery()
    diaryDBConn.Close()
    sqlCommand = Nothing
    diaryDBConn = Nothing
End Sub
```

GridView cũng có liên kết *Sửa*; khi nhấp vào, người dùng sẽ được chuyển đến trang *EditContact.aspx*:

```
<asp:HyperLinkField DataNavigateUrlFields="ContactId"
    DataNavigateUrlFormatString="~/SecureDiary/EditContact.aspx?ContactId={0}"
    Text="Sửa" />
```

ContactId tương ứng được truyền trong URL làm dữ liệu URL.

Liên kết *Thêm liên lạc mới* trên trang *YourContacts.aspx* sẽ đưa bạn đến form nhập các thông tin liên lạc như: tên, e-mail, số điện thoại,... Trang này và trang *EditContact.aspx* hoạt động tương tự nhau ngoại trừ một chi tiết quan trọng: trang *EditContact.aspx* lấy các thông tin liên lạc (để chỉnh sửa) từ lớp Contact. Điều này diễn ra trong sự kiện Page_Load:

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    If IsPostBack Then
        Dim currentContact As New Contact(CLng(Request.QueryString("ContactId")))
        currentContact.FirstName = firstNameTextBox.Text
        currentContact.LastName = lastNameTextBox.Text
        currentContact.AddressLine1 = addressLine1TextBox.Text
        currentContact.City = cityTextBox.Text
        currentContact.PostalCode = postalCodeTextBox.Text
        currentContact.State = stateTextBox.Text
        currentContact.Telephone = telephoneTextBox.Text
        currentContact.MobilePhone = mobilePhoneTextBox.Text
        currentContact.Email = emailTextBox.Text
        currentContact.SaveContact()
        currentContact = Nothing
        Response.Redirect("YourContacts.aspx")
    Else
        Dim currentContact As New Contact(CLng(Request.QueryString("ContactId")))
        firstNameTextBox.Text = currentContact.FirstName
        lastNameTextBox.Text = currentContact.LastName
        addressLine1TextBox.Text = currentContact.AddressLine1
        cityTextBox.Text = currentContact.City
        postalCodeTextBox.Text = currentContact.PostalCode
        stateTextBox.Text = currentContact.State
        telephoneTextBox.Text = currentContact.Telephone
        mobilePhoneTextBox.Text = currentContact.MobilePhone
        emailTextBox.Text = currentContact.Email
        currentContact = Nothing
    End If
End Sub
```

Câu lệnh If xác định đây là một postback (form đã được submit về chính nó) hay trang vừa được nạp. Nếu là một postback, bạn cần lưu dữ liệu và rồi trở về trang danh bạ liên lạc. Nếu là trang mới nạp, bạn cần tạo một đối tượng Contact mới, và sử dụng dữ liệu từ đó để đổ dữ liệu vào các trường của form.

Trang *AddContact.aspx* cũng giống như vậy, trừ việc không phải đổ những dữ liệu đã có, bởi một liên lạc mới không có dữ liệu trước!

1.4

Cài đặt Online Diary

Một trong những điều tuyệt vời của *ASP.NET 2.0* là dễ dàng cài đặt các ứng dụng web từ máy này lên máy khác. Để cài đặt ứng dụng này trên PC của bạn, bạn chỉ việc chép toàn bộ thư mục và các file từ CD-ROM đính kèm vào một thư mục trên PC của bạn (chẳng hạn *C:\Websites*). Trong *Visual Web Developer*, hãy chọn *File | Open Web Site* và tìm đến thư mục mà bạn đã chép các file. Sau đó, nhấn *F5* để chạy ứng dụng.

Nếu đã cài đặt *IIS*, bạn có thể tạo thư mục ảo *OnlineDiary* cho ứng dụng này và duyệt <http://localhost/OnlineDiary/SignOn.aspx>.

Nếu muốn hiệu chỉnh ứng dụng *Online Diary*, bạn hãy tham khảo phần tiếp theo.

1.5

Mở rộng Online Diary

Dự án này cung cấp một điểm khởi đầu tốt cho một hệ thống quản lý danh bạ liên lạc và nhật ký trọn vẹn. Ví dụ, bạn có thể cải tiến việc quản lý và hiển thị danh bạ liên lạc. Dù hoạt động tốt đối với danh bạ liên lạc nhỏ, nhưng hệ thống sẽ trở nên cồng kềnh khi danh bạ liên lạc tăng. Tổ chức hiển thị các liên lạc theo chữ cái đầu tiên của phần họ (hay theo chữ cái đầu tiên của phần tên) là một cách giải quyết vấn đề này. Lớp *Contact* đã có một phương thức cho phép điều đó và bạn có thể sử dụng. Một chọn lựa khác là thêm tính năng tìm kiếm dựa trên một hay nhiều thông số như e-mail, tên,...

Cách này hay cách kia, bạn cũng sẽ cần hiệu chỉnh trang *YourContacts.aspx*. Hiện tại khi hiển thị các liên lạc trong điều khiển *GridView*, nó cho hiện tất cả. Lớp *Contact* đã có một hàm nhận vào chữ cái đầu tiên của phần họ làm thông số. Tuy nhiên, thông số đó là tùy chọn. Bằng việc thêm một danh sách xổ xuống gồm các chữ cái từ A đến Z, bạn có thể sử dụng nó để truyền chữ cái đầu tiên của phần họ làm thông số.

Sau đó, bạn cần cho điều khiển *ObjectDataSource* biết thông số mới. Nếu gọi danh sách xổ xuống là *DropDownBox1* thì bạn cần thêm phần tử sau vào thẻ đánh dấu *ObjectDataSource*:

```
<asp:ControlParameter ControlID="DropDownList1"
    Name="FirstLetterOfSurname" PropertyName="SelectedValue" Type="Char" />
```

Giờ đây, chỉ những ai có phần họ được chỉ định trong danh sách xổ xuống mới được hiển thị.

Một vài cải tiến nữa có thể được xem xét:

- ✓ Tính năng upload và lưu trữ ảnh của những người trong danh bạ liên lạc.
- ✓ Tính năng chỉ định những ai sẽ quan tâm đến một sự kiện.
- ✓ Lưu trữ các chi tiết như ngày sinh của một người và có bộ nhắc tự động.
- ✓ Có bộ nhắc tự động đối với các sự kiện được mail cho bạn.
- ✓ Thêm tính năng hỗ trợ tin nhắn điện thoại di động, cho phép bạn gửi e-mail hay text message đến các những người trong danh bạ liên lạc.

1.6

Kết chương

Qua chương này, bạn đã biết cách tạo một hệ thống quản lý danh bạ liên lạc và nhật ký đầy đủ chức năng, tất cả chỉ với một ít mã lệnh nhờ các điều kiện và chức năng mới của *ASP.NET 2.0*. Các điều kiện bảo mật mới giúp tiết kiệm rất nhiều thời gian viết mã. Trong chương này, chúng đã được sử dụng để tạo người dùng và đăng nhập. Tuy nhiên, chúng cũng có thể cung cấp nhiều chức năng hơn như tạo các vai trò người dùng khác nhau, sau đó cho phép bạn chỉ định người dùng có thể làm gì và không thể làm gì dựa vào vai trò của họ. Hoặc bạn có thể để người dùng xác định cảm quan về các trang bằng tài khoản của họ và các điều kiện mới của *ASP.NET 2.0*.

Một điều kiện tuyệt vời khác mà bạn đã khám phá trong chương này là `ObjectDataSource`. Các điều kiện nguồn dữ liệu trước đây khiến công việc dễ dàng hơn, nhưng thiết kế mã kém, và bạn phải từ bỏ kiến trúc ba tầng. Giờ đây với điều kiện `ObjectDataSource`, bạn có thể có được kiến trúc ba tầng cho các dự án với thiết kế tốt, tính bảo trì cao.

Trong chương tiếp theo, bạn sẽ tạo một dự án chia sẻ file và tìm hiểu thêm một vài tính năng mới của *ASP.NET 2.0*.