

CÁC GIẢI PHÁP LẬP TRÌNH ASP.NET 2.0

Nguyễn Ngọc Bình Dương - Thái Kim Dũng - Lê Ngọc Sơn - Nguyễn Hoàng Thanh Nhân

CÁC GIẢI PHÁP LẬP TRÌNH ASP.NET 2.0

Nguyễn Ngọc Bình Phương - Thái Kim Dung - Lê Ngọc Sơn - Nguyễn Hoàng Thanh Nhân

Chịu trách nhiệm xuất bản: TS. Nguyễn Xuân Thủy

Biên tập: Đất Việt - DVPUB

Trình bày bìa: Nguyễn Thị Thanh Thủy

Chế bản & đưa bản in: Nguyễn Ngọc Bình Phương

Nhà sách Đất Việt

Địa chỉ: 225 Nguyễn Tri Phương, Quận 5, TP. Hồ Chí Minh

Điện thoại: (08) 2 652 039 - E-mail: datviet@dvpub.com.vn

Website: <http://dvpub.com.vn> (hay <http://dvpub.vn>)

In 1000 cuốn, khổ 19x27cm, tại XN in 86 05.

Số đăng ký KHXB - /XB-QLXB

do Cục Xuất Bản cấp ngày tháng năm 2007.

In xong và nộp lưu chiểu tháng năm 2007.

CÁC GIẢI PHÁP LẬP TRÌNH ASP.NET 2.0

Nguyễn Ngọc Bình Dương - Thái Kim Dung - Lê Ngọc Sơn - Nguyễn Hoàng Thanh
Nhà

TẬP MỘT

NHÀ XUẤT BẢN GIAO THÔNG VẬN TẢI

LỜI NÓI ĐẦU

CÁC GIẢI PHÁP LẬP TRÌNH ASP.NET 2.0

Các giải pháp lập trình ASP.NET 2.0 là quyển sách hoàn hảo cho những ai muốn tìm hiểu nhanh các tính năng và công cụ mới trong phiên bản 2.0 của ASP.NET. Bạn có thể đọc một quyển sách theo lối truyền thống, và mất nhiều thời gian để nghĩ xem phải làm gì để viết một phần mềm cho ra trò. Nhưng nếu sử dụng quyển sách này, bạn có thể hiểu và thử nghiệm các dự án vô cùng thiết thực chỉ trong một thời gian ngắn.

ASP.NET 2.0 là một công nghệ “nóng”, là sự kế tục và cải tiến mạnh so với phiên bản 1.1, nhờ vào một số tính năng hấp dẫn nhắm đến việc trang bị cho nhà phát triển web các công cụ cần cho phần lớn các dự án. Các công cụ này bao gồm các điều kiểm đăng nhập mới, cho phép nhà phát triển nhanh chóng kéo và thả một bộ các điều kiểm lên bề mặt thiết kế, thiết lập một vài thuộc tính, và có ngay một hệ thống bảo mật. Một tính năng mới khác là những cải tiến cho giao diện người dùng lúc thiết kế nhằm kết các điều kiểm dữ liệu. Các điều kiểm hấp dẫn khác cũng tồn tại, nhưng không phải là mục tiêu của sách này. Chúng ta chỉ dừng lại ở một số thay đổi đã được kiến lập trong ASP.NET 2.0, bao gồm kiến trúc của các website, tốc độ của các yêu cầu trang, việc quản lý thông tin có trạng thái, và hiệu quả phát triển toàn diện. Nếu bạn đã chọn ASP.NET 2.0, ắt nỗ lực lập trình của bạn sẽ dễ dàng hơn rất nhiều.

Sách này lý tưởng cho bất kỳ lập trình viên nào đã có kinh nghiệm với .NET và muốn tìm hiểu công nghệ mới. Bất kỳ ai đã có kinh nghiệm với .NET 1.1 hoặc đã từng hiện thực các giải pháp quy mô toàn phần sẽ là ứng viên hoàn hảo để lĩnh hội tất cả các tư liệu trong quyển sách này. Bạn sẽ có những hiểu biết tổng quát về cách làm việc của một cơ sở dữ liệu cơ bản, cùng với những kinh nghiệm trong việc phát triển phần mềm để truy xuất một đối tượng cơ sở dữ liệu, chẳng hạn một thủ tục tồn trữ.

Nếu là người mới bắt đầu, bạn có thể nhảy đến chương 6, Photo Album. Chương này có thiết kế “nhẹ nhàng”, nhắm đến việc cung cấp một lối tiếp cận đơn giản cho một dạng website phổ biến, đó là website trình diễn ảnh gia đình. Chương này lý giải các khái niệm cơ bản về việc chia sẻ ảnh trong một website và thêm/xóa ảnh. Đây là một cách hay để sớm nắm bắt các khái niệm cơ bản, trước khi đi sâu vào các khái niệm và thiết kế phức tạp hơn trong quyển sách này.

Lối tiếp cận hay nhất đối với người đọc là tìm ra một chương mà mình quan tâm và nhảy đến đó. Không một kiến thức nào bị mất khi đọc theo lối cảm hứng như thế. Điều này có nghĩa đây là một sách hướng dẫn hữu ích và cũng là một sách tra cứu nhanh nhằm đào sâu mã lệnh và học thật nhanh. Các chương hoàn toàn độc lập và là những bản hiện thực khác nhau với nhiều tính năng trùng lặp. Một số chương có thể lặp lại một khái niệm cho một quy trình nào đó khá rườm rà, trong khi một số chương có thể lệch hướng khỏi các tính năng và lối sử dụng nhất quán nhằm cung cấp một lối tiếp cận bình thường cho những đòi hỏi đặc trưng của website. Trong một số trường hợp cần một sự lý giải đầy đủ về một đề tài đã được thảo luận, bạn sẽ nhận thấy có một chương trước đó đã mô tả đề tài này một cách rất chi tiết.

Sách này bao gồm 12 dự án và bạn có thể sử dụng ngay chỉ với một ít cài đặt. Mỗi dự án đều có các chỉ dẫn cài đặt mã nguồn vào máy cục bộ. Mục tiêu của chương và mã lệnh dự án đi kèm là giúp bạn hiểu và nhanh chóng hiệu chỉnh dự án để cải tiến hay hiện thực một số tính năng. Qua những bài học được lặp đi lặp lại, bạn sẽ biết nhiều hơn về lối thiết kế và tạo các dự án trong phiên bản 2.0 của ASP.NET. Sách này khảo sát một số tiến bộ vượt bậc từ phiên bản 1.1, nhưng không ở mức chi tiết. Hầu hết mã lệnh và tư liệu đã được “cân đo” với mục đích lý giải các khái niệm trong phiên bản 2.0 một cách trọn vẹn.

Chúng tôi xin chân thành cảm ơn các bạn Thái Thanh Phong, Trần Thế Quân, Đinh Phan Chí Tâm, Nguyễn Quang Nam, và Lê Trần Nhật Quỳnh đã có những đóng góp quý báu cho quyển sách; cảm ơn Nhà xuất bản Giao thông Vận tải và Nhà sách Đất Việt đã tạo điều kiện cho quyển sách này đến với bạn đọc.

Do lần đầu tiên xuất bản nên quyển sách này khó tránh khỏi những thiếu sót. Rất mong nhận được ý kiến đóng góp và nhận xét của các bạn để lần tái bản sau được hoàn thiện hơn.

Webpage: <http://dvpublish.com.vn/dv/details.aspx?itemid=332>

Email: docgia@dvpublish.com.vn

Trân trọng
CÁC TÁC GIẢ

MỘT SỐ CHỈ DẪN

CÁC GIẢI PHÁP LẬP TRÌNH ASP.NET 2.0

■ Cấu trúc của sách

Quyển sách này được thiết kế giống như một tài liệu tham khảo cho các dự án mã nguồn mở. Hầu hết lập trình viên không cần hấp thụ mọi thông tin về một chủ đề cụ thể theo kiểu truyền thống; mà nhiều khi lập trình viên tìm thấy câu trả lời trong mã lệnh, rồi mới đọc nội dung hay tư liệu kèm theo đó. Quyển sách này nhằm đến việc thỏa mãn khuynh hướng đó, các chủ đề và các khái niệm được truyền đạt từ căn bản đến nâng cao, xuyên suốt 12 dự án tương ứng với 12 chương sau:

TẬP MỘT

- Chương 1. ONLINE DIARY
- Chương 2. FILE SHARE
- Chương 3. CHAT SERVER
- Chương 4. SURVEY ENGINE
- Chương 5. BLOG
- Chương 6. PHOTO ALBUM

TẬP HAI

- Chương 7. CMS
- Chương 8. CUSTOMER SUPPORT
- Chương 9. WEBSHOP
- Chương 10. APPOINTMENT BOOKING
- Chương 11. GREETING CARD
- Chương 12. BUG BASE

Cấu trúc của mỗi chương tuân theo khuôn mẫu chung sau:

- ✓ Sử dụng ứng dụng,
- ✓ Thiết kế ứng dụng,
- ✓ Mã lệnh ứng dụng,
- ✓ Cài đặt ứng dụng,
- ✓ Mở rộng ứng dụng.

Mỗi dự án được thiết kế với các module, các file lớp, và các điều kiểm có thể tái sử dụng. Các lớp và các file quan trọng sẽ được nêu và phân tích chi tiết ở mỗi chương.

Các chương 4, 6, và 9 đề cập các chủ đề quen thuộc và dễ tiếp cận; trong khi các chương 3, 10, và 12 đề cập các chủ đề nâng cao, chắc chắn bạn sẽ học được nhiều thứ từ đó. Kết thúc các bài học, bạn sẽ có thể tự tạo cho mình một website, hiện thực cơ chế bảo mật với một kiến trúc có tổ chức, sử dụng các điều kiểm kết dữ liệu mới và hiệu quả hơn, cơ chế kết dữ liệu dựa vào đối tượng, và nhiều tính năng khác nữa.

■ Quy ước về font chữ

Quyển sách này sử dụng các quy ước về font chữ như sau:

- ✓ Font Tahoma—Dùng cho tên riêng, tên file và thư mục, đường dẫn, nhãn, liên kết, và đôi khi để nhấn mạnh.
- ✓ Font Lucida Sans Typewriter—Dùng cho nội dung file; các phân tử cơ sở dữ liệu (như bảng, khung nhìn, thủ tục tồn trữ, truy vấn, trường, kiểu dữ liệu); các đoạn chương trình cùng các phân tử mã lệnh (như không gian tên, lớp, kiểu, hàm, thủ tục, phương thức, thuộc tính, đặc tính, tham số, biến, giá trị, đối tượng, thể hiện, sự kiện, thẻ XML và thẻ HTML).

■ Yêu cầu về hệ thống

Để có thể chạy được những ứng dụng được trình bày trong quyển sách này, bạn sẽ cần những phần mềm sau đây:

- ✓ Microsoft .NET Framework version 2.0;
- ✓ Microsoft Visual Studio 2005 hay Visual Web Developer 2005 Express Edition;
- ✓ Windows 2000 Professional/Server SP4, Windows XP Professional SP2, Windows Server 2003 SP1, hay Windows Vista;
- ✓ Microsoft SQL Server 2005 Express Edition;
- ✓ Microsoft Internet Information Services (IIS).

Yêu cầu tối thiểu về phần cứng là bộ vi xử lý 600 MHz, RAM 192 MB. Bạn cũng cần khoảng 5 GB dung lượng đĩa cứng còn trống để cài đặt Visual Studio 2005 và thư viện MSDN. Những giá trị này là mức tối thiểu, quá trình phát triển sẽ dễ dàng hơn trên một hệ thống với bộ vi xử lý cao, dung lượng RAM lớn và đĩa cứng còn trống nhiều.

- ✎ Bạn sẽ được hướng dẫn cài đặt Microsoft Internet Information Services (IIS) và Visual Studio 2005 Professional Edition trong phụ lục A, “Một số vấn đề liên quan đến Visual Studio 2005”.

■ Sử dụng CD-ROM đính kèm

CD-ROM đính kèm theo sách chứa toàn bộ mã nguồn, cũng như gói cài đặt cho các ứng dụng được trình bày trong quyển sách này.

Thư mục gốc của CD-ROM gồm ba thư mục:

- ✓ Websites—Chứa toàn bộ mã nguồn của các ứng dụng,
- ✓ PrecompiledWebsites—Chứa các ứng dụng đã được biên dịch,
- ✓ WebSetupProjects—Chứa các dự án cài đặt ứng dụng,
- ✓ ASP.NETMaker3—Chứa file cài đặt công cụ sinh mã ASP.NET Maker 3 và dự án mẫu.

Ba thư mục đầu có cấu trúc giống nhau, gồm sáu thư mục con ứng với sáu ứng dụng được trình bày trong tập một này:

- ✓ OnlineDiary—Sổ nhật ký trực tuyến,
- ✓ FileShare—Kho lưu trữ file trực tuyến,
- ✓ ChatServer—Hệ thống hội thoại trực tuyến,
- ✓ SurveyEngine—Bộ máy khảo sát trực tuyến,
- ✓ Blog—Tập san cá nhân trực tuyến,
- ✓ PhotoAlbum—Bộ sưu tập ảnh trực tuyến.

Bạn có hai tùy chọn khi cài đặt một ứng dụng:

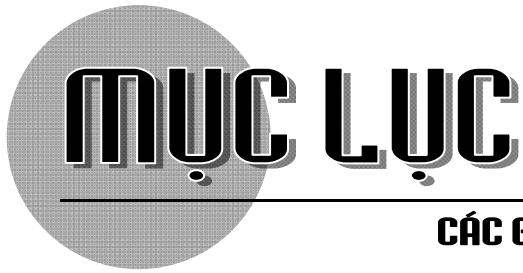
- ✓ Nếu muốn cài đặt ứng dụng bằng trình cài đặt, chạy file WebSetupProjects\xyz\Debug\xyz.msi (xyz là tên ứng dụng). Như thế, ứng dụng sẽ được cài đặt vào Internet Information Services (IIS).
- ✓ Nếu muốn cài đặt ứng dụng bằng tay, chép thư mục Websites\xyz\ vào đĩa cứng. Như thế, bạn có thể mở và chỉnh sửa ứng dụng bằng Visual Studio 2005 (hay Visual Web Developer).

Bạn sẽ được chỉ dẫn cả hai tùy chọn cài đặt này tại mỗi chương.

Bạn không cần chép thư mục PrecompiledWebsites và WebSetupProjects vào đĩa cứng, vì bạn sẽ biết cách tạo hai thư mục này trong phụ lục A, “Một số vấn đề liên quan đến Visual Studio 2005”.

- ✎ Nếu không đọc được nội dung trong CD-ROM đính kèm, bạn có thể trực tiếp liên hệ với chúng tôi theo địa chỉ sau để nhận CD-ROM khác:

Nhà sách Đất Việt
225 Nguyễn Tri Phương, Quận 5, TP. Hồ Chí Minh - ĐT: (08) 2 652 039



MỤC LỤC

CÁC GIẢI PHÁP LẬP TRÌNH ASP.NET 2.0

LỜI NÓI ĐẦU	5
MỘT SỐ CHỈ DẪN	7
MỤC LỤC	11
Chương 1. ONLINE DIARY	17
1.1 Sử dụng Online Diary	19
1.2 Thiết kế Online Diary	23
1.2.1 Tầng truy xuất dữ liệu	23
1.2.2 Tầng nghiệp vụ	25
Lớp OnlineDiary	25
Lớp Contact	25
Lớp ContactCollection	28
Lớp DiaryEntry	28
Lớp DiaryEntryCollection	30
Lớp DiaryEvent	30
Lớp DiaryEventCollection	32
1.3 Mã lệnh Online Diary	33
1.3.1 Cấu trúc website	33
1.3.2 Đăng ký, đăng nhập, và bảo mật	34
Đăng nhập	34
Đăng ký người dùng mới	35
Nhắc mật khẩu	39
1.3.3 Xem lịch biểu trực tuyến	40
1.3.4 Tạo, chỉnh sửa, và xem một mục nhật ký	42

1.3.5 Tạo, chỉnh sửa, và xem các sự kiện nhật ký	44
1.3.6 Quản lý danh bạ liên lạc	46
1.4 Cài đặt Online Diary	48
1.5 Mở rộng Online Diary	49
1.6 Kết chương.....	50
Chương 2. FILE SHARE	51
2.1 Sử dụng File Share	54
2.2 Thiết kế File Share	56
2.2.1 Upload file	57
2.2.2 Gửi e-mail.....	57
2.2.3 Cấu trúc website.....	58
2.2.4 Mô hình dữ liệu và các đối tượng cơ sở dữ liệu	59
Bảng Email.....	59
Bảng Contact	60
Bảng Resource	60
2.2.5 Theme và skin	64
2.2.6 Mô hình bảo mật	64
2.2.7 Các lớp chính yếu	65
Lớp EmailContent	65
Lớp Resource	66
Lớp Config	67
Lớp Utilities	68
2.3 Mã lệnh File Share	68
2.3.1 Một số file quan trọng	68
Web.config	69
config.vb.....	69
Resource.vb	71
resourceDB.vb	72
2.3.2 Webform.....	74
Default.aspx	74
Login.aspx.....	75
Download.aspx.....	75
2.3.3 Các điều kiểm người dùng	77
header.ascx.....	77
footer.ascx.....	77
navigation.ascx	77
2.4 Cài đặt File Share.....	78
Cài đặt File Share trên máy server.....	78
Cài đặt File Share để phát triển cục bộ	79
2.5 Mở rộng File Share.....	80
2.6 Kết chương.....	82
Chương 3. CHAT SERVER.....	83
3.1 Sử dụng Chat Server.....	86
3.2 Thiết kế Chat Server.....	88

3.2.1	Gửi thông điệp bằng Callback	88
3.2.2	Cấu trúc website	91
3.2.3	Mô hình dữ liệu	91
	Bảng Category	92
	Bảng Room	92
	Bảng Message	93
	Bảng User	93
3.2.4	Theme và skin	93
3.2.5	Các lớp chính yếu	94
	Lớp ChatRoom	94
	Lớp ChatRoomDB	95
	Lớp Config	95
3.3	Mã lệnh Chat Server	96
3.3.1	Một số file quan trọng	96
	Web.config	96
	Config.vb	97
	ChatRoom.vb	98
	ChatRoomDB.vb	98
3.3.2	Webform	100
	SignIn.aspx	100
	Default.aspx	100
	ChatRoom.aspx	101
	ChatWindow.aspx	103
3.3.3	Các điều kiểm người dùng	105
	header.ascx	105
	footer.ascx	105
	navigation.ascx	106
3.4	Cài đặt Chat Server	106
	Cài đặt Chat Server trên máy server	106
	Cài đặt Chat Server để phát triển cục bộ	107
3.5	Mở rộng Chat Server	108
3.6	Kết chương	110
Chương 4. SURVEY ENGINE		111
4.1	Sử dụng Survey Engine	114
	Thêm một khảo sát mới	117
	Chỉnh sửa một khảo sát	120
4.2	Thiết kế Survey Engine	121
4.2.1	Kết đối tượng và kết dữ liệu SQL Server	121
4.2.2	Cấu trúc website	122
4.2.3	Mô hình dữ liệu và các đối tượng cơ sở dữ liệu	123
4.2.4	Theme và skin	125
4.2.5	Mô hình bảo mật	126
4.2.6	Các lớp chính yếu	127
	Lớp SurveyBase	127

Lớp Survey.....	128
Lớp SurveyDB.....	129
Lớp Config	130
4.3 Mã lệnh Survey Engine	131
4.3.1 Một số file quan trọng	131
Web.config	131
Survey.vb	131
Config.vb	132
SurveyDB.vb	133
4.3.2 Webform.....	134
Default.aspx	134
Login.aspx.....	135
TakeSurvey.aspx	135
4.3.3 Các điều kiểm người dùng	135
header.ascx.....	136
footer.ascx.....	136
navigation.ascx	136
SurveyResults.ascx.....	137
CurrentSurvey.ascx.....	138
4.4 Cài đặt Survey Engine	139
Cài đặt Survey Engine trên máy server	139
Cài đặt Survey Engine để phát triển cục bộ	139
4.5 Mở rộng Survey Engine.....	140
4.6 Kết chương.....	144
Chương 5. BLOG	145
5.1 Sử dụng Blog	148
5.2 Thiết kế Blog	151
5.2.1 Tầng nghiệp vụ.....	151
Lớp BlogEntry	151
Lớp BlogManager	152
Lớp UserManager	154
5.2.2 Tầng truy xuất dữ liệu	154
Lớp BlogManagerDB.....	156
Lớp UserManagerDB	157
Mô hình dữ liệu	158
Thủ tục tồn trữ và truy vấn	159
5.2.3 Các lớp trợ giúp	160
5.3 Mã lệnh Blog	160
5.3.1 Các file trong thư mục gốc	160
Web.config	160
Login.aspx.....	161
Global.asax	163
5.3.2 Viết mã độc lập provider.....	164
5.3.3 Thư mục Controls.....	167

BlogEntriesFilter.ascx	167
BlogEntries.ascx	169
5.3.4 Thụ lý và ghi nhận lỗi	175
Cấu hình.....	175
Thụ lý và ghi nhận lỗi	176
5.4 Cài đặt Blog.....	178
Cài đặt Blog bằng trình cài đặt	178
Nhúng Blog vào một ứng dụng mới hay có sẵn	179
5.5 Mở rộng Blog.....	181
5.6 Kết chương.....	186
Chương 6. PHOTO ALBUM.....	187
6.1 Sử dụng Photo Album	190
6.2 Thiết kế Photo Album	196
6.2.1 Làm sao để mọi thứ ăn khớp với nhau?.....	196
Lưu trữ hình ảnh	196
Hiển thị hình ảnh	196
Cấu trúc website	197
Theme và skin.....	198
Mô hình dữ liệu	200
Mô hình bảo mật	201
6.2.2 Các lớp chính yếu	201
Lớp Photo	201
Lớp PhotoCollection.....	202
Lớp PhotoDB	202
6.3 Mã lệnh Photo Album	204
6.3.1 Các file trong thư mục gốc	205
Web.config	205
Masterpage.master và Admin.master	205
Web.sitemap	205
6.3.2 Webform.....	206
Photos.aspx	206
Login.aspx.....	207
Admin.aspx	208
Editphotos.aspx.....	209
6.3.3 Các file trong thư mục Secure.....	210
6.3.4 Các điều kiểm người dùng	210
header.ascx.....	210
footer.ascx.....	211
navigation.ascx	211
6.4 Cài đặt Photo Album.....	212
Cài đặt Photo Album trên máy server.....	212
Cài đặt Photo Album để phát triển cục bộ	212
6.5 Mở rộng Photo Album	213
6.6 Kết chương.....	219

PHỤ LỤC A. MỘT SỐ VẤN ĐỀ LIÊN QUAN ĐẾN VISUAL STUDIO 2005	221
A.1 Cài đặt Visual Studio 2005	221
Cài đặt Internet Information Services (IIS) và dịch vụ SMTP	222
Cài đặt Visual Studio 2005 và thư viện MSDN	225
A.2 Các công cụ cấu hình mới trong ASP.NET 2.0	234
Web Site Administration Tool	234
Trang thuộc tính ASP.NET trong IIS	239
A.3 Tạo gói cài đặt website	241
Biên dịch ứng dụng	241
Tạo dự án cài đặt	244
Cài đặt và sử dụng ứng dụng	248
A.4 Tạo kịch bản cơ sở dữ liệu	252
PHỤ LỤC B. GIỚI THIỆU CÔNG CỤ SINH MÃ ASP.NET MAKER 3	255
B.1 Cài đặt ASP.NET Maker 3	255
B.2 Một số tính năng nổi trội của ASP.NET Maker 3	257
B.3 Sử dụng ASP.NET Maker 3	264
THUẬT NGỮ	271
TÀI LIỆU THAM KHẢO	275

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 khung 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 (tập san cá nhân trực tuyến, sẽ được thảo luận trong chương 5). 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 Online Diary 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 Online Diary 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. Cuối cùng, bạn sẽ cài đặt Online Diary và có những hướng cải tiến cho nó.

1.1

Sử dụng Online Diary

Mỗi người dùng có một sổ nhật ký trực tuyến riêng, để truy cập cần phải đăng nhập. ả 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 và nhiều mã lệnh tạo nên chức năng bảo mật, thật ra với các điều kiểm bảo mật mới trong ASP.NET 2.0 thì mọi việc trở nên rất dễ dàng và chẳng có nhiều việc để làm.

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.

Online Diary

Đăng ký tài khoản mới

Tên người dùng:

Mật khẩu:

Nhập lại mật khẩu:

E-mail:

Câu hỏi mật:

Câu trả lời mật:

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).

Online Diary

Nhắc mật khẩu

Nhập tên người dùng để nhận mật khẩu

Tên người dùng:

Hình 1-3

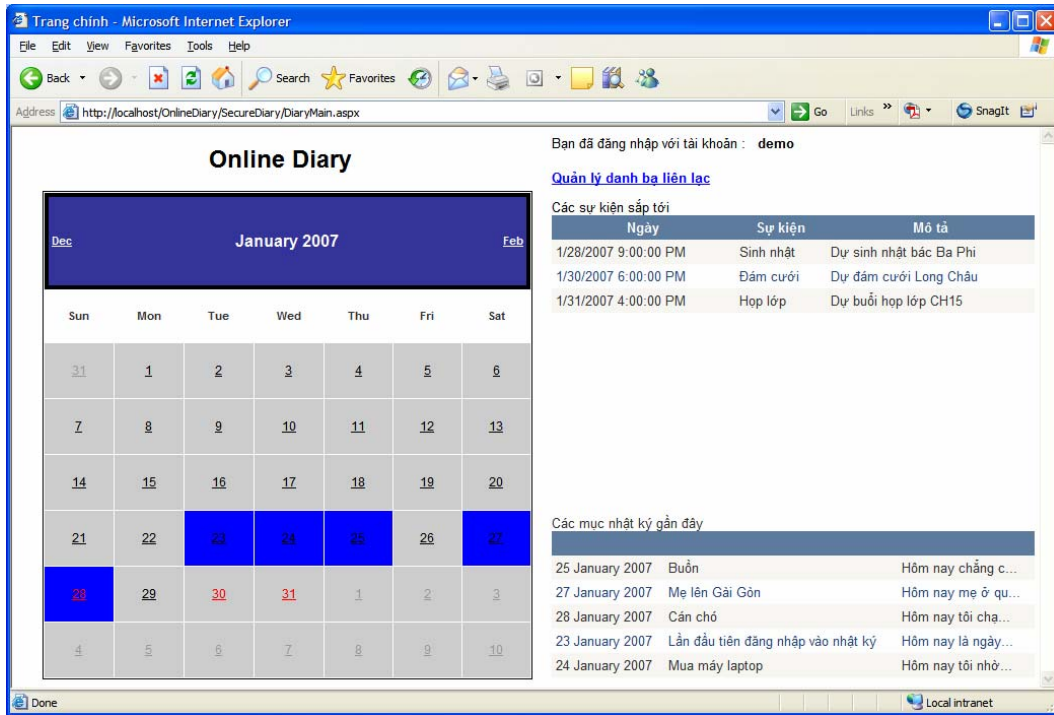
Sau khi đăng nhập, bạn đến trang nhật ký chính, được hiển thị trong hình 1-4.

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.

Ấp 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 số 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

Ấnhấp vào liên kết này, bạn sẽ được dẫn đến trang quản lý danh bạ 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	Đình Phan Chí	Tâm		dpctam@yahoo.com
Xóa Sửa	Lê Ngọc	Sơn		lnson@yahoo.com
Xóa Sửa	Lê Trần Vy	Quỳnh		vyquynh2001@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).

Thêm liên lạc

Tên Điện thoại cố định

Họ và chữ lót Điện thoại di động

Địa chỉ E-mail

Tỉnh thành

Quốc gia [Hủy bỏ](#)

Mã vùng

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. 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.

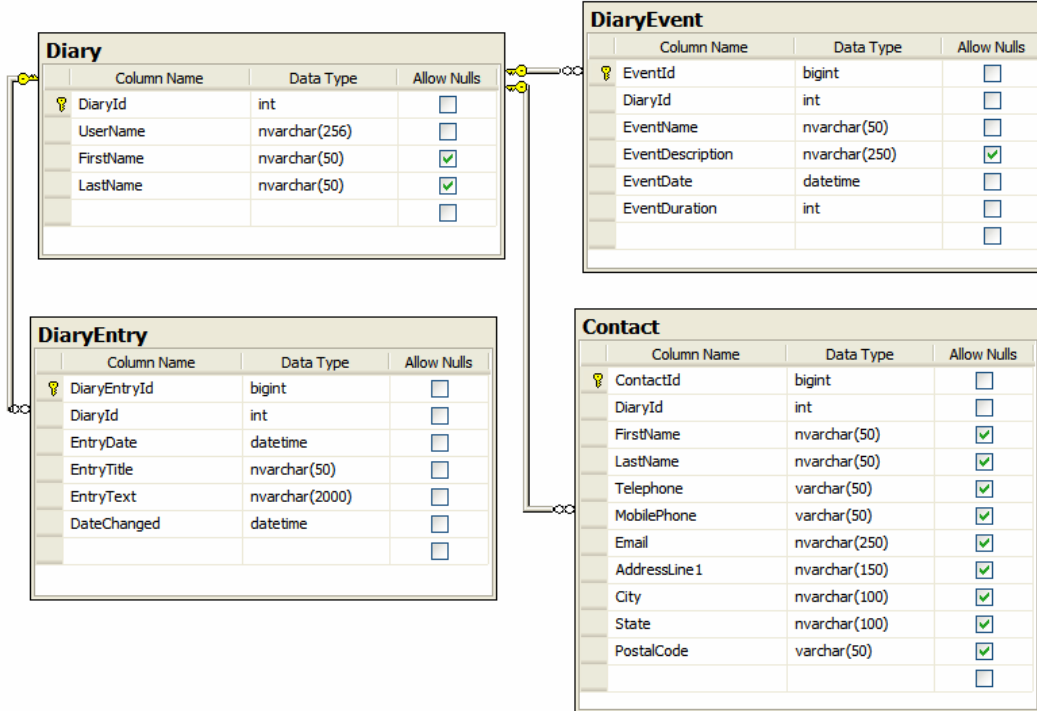
Phía trên tầng truy xuất dữ liệu là tầng nghiệp vụ, tầng này cung cấp tất cả các quy tắc và tính thông minh của hệ thống. Tầng nghiệp vụ được tổ chức thành bảy lớp.

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. Ắt ếu 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. Ắt ếu cơ sở dữ liệu không hỗ trợ thủ tục tồn trữ (chẳng hạn, Microsoft Access) thì thay đổi mã lệnh là cần thiết nhưng không khó.

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 tạo cho bạn. Mặc dù nó chỉ có một vài bảng, bạn không bao giờ truy xuất chúng thông qua mã lệnh. ả ó được truy xuất bởi các điều kiện đăng nhập 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 quy định các cấp độ membership (admin, user, operator).

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"/>

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`. ả ó 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 Online Diary.
<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. ả ó 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. Ồ 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.

Ồ 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. Ồ 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. Ồ ầu 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. Ồ ầu 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.

GetContactsByFirstLetter(ByVal DiaryId As Integer,Optional ByVal FirstLetterOfSurname As Char)	SqlDataReader	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.
GetContactsByFirstLetterAsCollection (ByVal DiaryId As Integer,Optional ByVal FirstLetterOfSurname As Char)	ContactCollection	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.

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)	Contact	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.       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.       thụ lý mọi truy xuất cơ sở dữ liệu cho các mục nhật ký.

         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 g�n 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 <code>Boolean</code>	Phương thức chia sẻ này trả về một mảng <code>Boolean</code> 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

Ở ngoài các phương thức khởi dựng 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>DiaryEntry</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 sự kiện trong nhật ký. Ở đó đóng gói mọi thứ để làm việc với các sự kiện nhật ký, bao gồm: tạo, cập nhật, và thu lấy dữ liệu của các sự kiện nhật ký. Ở đó thu 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
-----------------------	-------

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

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 tham số `EventId`. ả ó 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
<code>Save()</code>	không	Lưu một đối tượng <code>DiaryEvent</code> đã có đầy đủ dữ liệu. ả ếu nó là một mục mới, <code>Save()</code> gọi thủ tục <code>InsertNewDiaryEvent</code> và các chi tiết được chèn vào cơ sở dữ liệu. <code>EventId</code> mới được trả về từ cơ sở dữ liệu và được gán cho <code>mEventId</code> . ả ếu mục này đã tồn tại trong cơ sở dữ liệu, <code>Save()</code> gọi <code>UpdateDiaryEvent</code> để cập nhật cơ sở dữ liệu với các giá trị trong đối tượng <code>DiaryEvent</code> .
<code>GetDaysInMonthWithEvents(ByVal DiaryId As Integer, ByVal Month As Integer, ByVal Year As Integer)</code>	Mảng <code>Boolean</code>	Phương thức chia sẻ này trả về một mảng <code>Boolean</code> 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,...).
<code>GetDiaryEventsByDate(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 sự kiện nhật ký từ <code>FromDate</code> đến <code>ToDate</code> .

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.

Để gọi 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)	DiaryEvent	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

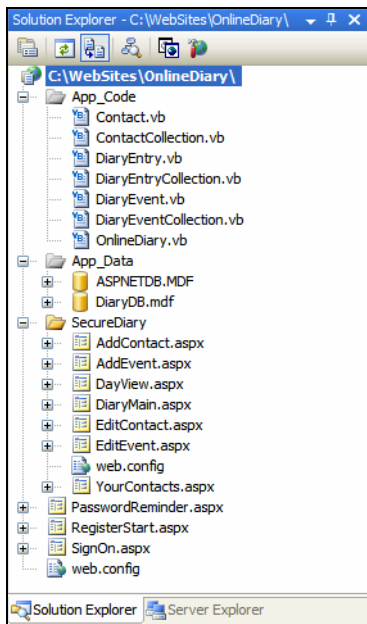
Phần này đi sâu vào từng trang quan trọng và cho biết chúng tương tác với nhau thế nào, cũng như chúng sử dụng các lớp trong tầng nghiệp vụ ra sao. Phần này không phân tích từng dòng của mỗi trang, nhưng phần nào cũng cho bạn một cái nhìn tổng quan về cách ứng dụng làm việc và đi sâu hơn vào những chỗ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

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



Hình 1-12

Bây 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 đăng nhập 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 đăng nhập 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 PasswordRecovery.

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.

Mã 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 đăng nhập mới, đó là điều kiểm CreateUserWizard. Mã 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"
    UserNameLabelText="Tên người dùng:" CreateUserButtonText="Tạo tài khoản"
    PasswordLabelText="Mật khẩu:" ConfirmPasswordLabelText="Nhập lại mật khẩu:"
    QuestionLabelText="Câu hỏi mật:" AnswerLabelText="Câu trả lời mật:" >
    <SideBarButtonStyle 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="UserNameLabel1" 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>
    <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="#FFFFFF"
      BorderColor="#CCCCCC" 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 mã 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 mã đánh dấu giống như sau:

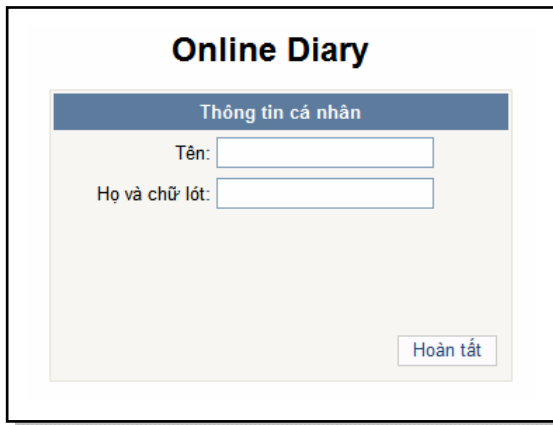
```
<asp:WizardStep ID="personalDetailsStep" runat="server" Title="User Details">
```

`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.

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.

Chúng 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).

Bạn cần cấu hình SMTP server để gửi e-mail nhắc mật khẩu. Bạn có thể sử dụng dịch vụ SMTP của Internet Information Services (IIS). Xem hướng dẫn cài đặt trong phụ lục A, “Một số vấn đề liên quan đến Visual Studio 2005”.

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.

ả goài 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, mã 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(conString)
  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
  Return entrySQLDR
End Function
```

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

Việc hiển thị dữ liệu chỉ là vấn đề trở một điều kiểm 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>
```

Chương 1. Online Diary

Trong mã đánh dấu của điều khiển GridView, đặc tính DataSourceID chỉ định nguồn dữ liệu—đó là điều khiển ObjectDataSource. Ồ goài ra, mã 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>
```

Ồ goài 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. Ồ ó 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 với 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
      End If
    Catch
    End Try
  End If
End Sub
```

```

        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. Ắt ối 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. Ắ ngược lại, chúng nhận dữ liệu từ cơ sở dữ liệu.

Khi hộp tiêu đề hoặc hộp nội dung mục nhật ký thay ối, `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 ối 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 ối tượng `DiaryEntry`:

```

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(conString)
        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
    End If
End Sub

```

```

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

diaryDBConn.Close()
sqlCmd = 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(conString)
        Dim sqlString As String = "UpdateDiaryEntry"
        Dim sqlCmd As New SqlCommand(sqlString, diaryDBConn)
        sqlCmd.CommandType = CommandType.StoredProcedure

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

        diaryDBConn.Open()
        sqlCmd.ExecuteNonQuery()
        diaryDBConn.Close()
        sqlCmd = 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. ả hững 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 kiện ObjectDataSource trên trang DayView.aspx lấy dữ liệu từ phương thức chia sẻ GetDiaryEventsByDate() của đối tượng DiaryEvent. Mã đánh dấu cho điều kiện 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>

```

```
</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 kiện `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>
```

Ở 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>
```

Liên kết này sẽ dẫn người dùng đến trang `EditEvent.aspx`. URL chứa tham 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`.

Ở 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 kiện `GridView` sẽ yêu cầu điều kiện `ObjectDataSource` gọi phương thức xóa được chỉ định của lớp cung cấp dữ liệu. Trong trường hợp này, đó là phương thức `DeleteEvent()` của lớp `DiaryEvent`. Đặc tính `DataKeyNames` trong mã đánh dấu của điều kiện `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 tham 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 tham 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 lặp 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>

```

Ả ó lấy dữ liệu từ điều kiểm ObjectDataSource ObjectDataSource1, đến phiên điều kiểm 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>

```

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

```

Public Shared Sub DeleteContact(ByVal ContactId As Long)
  Dim diaryDBConn As New SqlConnection(conString)
  Dim sqlCommand As String = "DeleteContact"
  Dim sqlCmd 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. Ắt ếu là một postback, bạn cần lưu dữ liệu và rồi trở về trang danh bạ liên lạc. Ắt ếu 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.

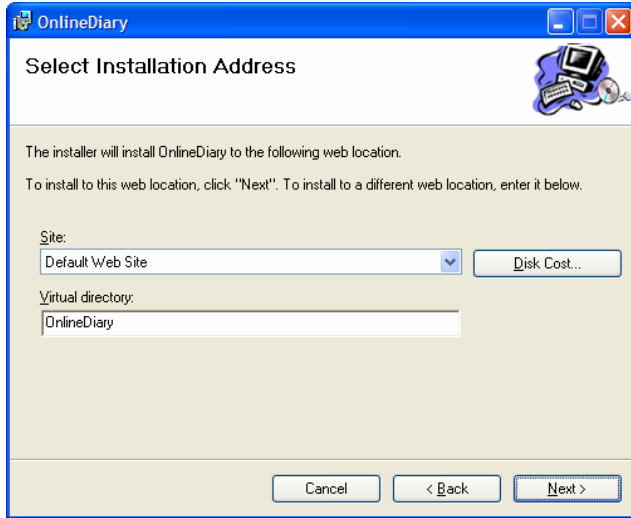
Trang AddContact.aspx cũng giống như vậy, trừ việc không phải đổ những dữ liệu đã có, bởi một liên lạc mới không có dữ liệu trước!

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 máy tính của bạn, bạn chỉ việc chép thư mục Websites\OnlineDiary\ từ CD-ROM đính kèm vào đĩa cứng, chẳng hạn C:\Websites\OnlineDiary\. Trong Visual Studio 2005 hay Visual Web Developer, chọn File | Open Web Site và tìm đến thư mục C:\Websites\OnlineDiary\. Sau đó, nhấn F5 để chạy ứng dụng.

Ắt ếu đã cài đặt IIS, bạn có thể chạy file WebSetupProjects\OnlineDiary\Debug\OnlineDiary.msi từ CD-ROM đính kèm. Quá trình này cài đặt những file cần thiết vào thư mục C:\inetpub\wwwroot\OnlineDiary\. Chú ý rằng, trong lúc cài đặt, có một màn hình yêu cầu bạn xác nhận tên thư mục ảo, đó là nơi ứng dụng sẽ được cài đặt trong IIS (xem hình 1-20). Tên thư mục ảo rất quan trọng, bởi vì nó sẽ cho phép duyệt http://localhost/tên_thư_mục_ảo. Bạn hãy giữ nguyên tên mặc định là OnlineDiary.



Hình 1-20

Sau khi cài đặt hoàn tất, bạn có thể duyệt ứng dụng theo địa chỉ <http://localhost/OnlineDiary/SignOn.aspx>.

Nếu muốn cải tiến 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 tham 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 tham số. Tuy nhiên, tham 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 tham số.

Sau đó, bạn cần cho điều khiển ObjectDataSource biết tham số mới. Nếu gọi danh sách xổ xuống là DropDownList1 thì bạn cần thêm phần tử sau vào mã 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 kiểm và chức năng mới của ASP.NET 2.0. Các điều kiểm 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 kiểm mới của ASP.NET 2.0.

Một điều kiểm tuyệt vời khác mà bạn đã khám phá trong chương này là `ObjectDataSource`. Các điều kiểm 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 kiểm `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.

2

FILE SHARE

KHO LƯU TRỮ FILE TRỰC TUYẾN

à ếu bạn từng cố gắng gửi một e-mail với file đính kèm có dung lượng lớn và đã thất bại, bạn không phải là người duy nhất gặp trường hợp này! Việc bạn có thể đính kèm file hay tải liệu vào e-mail và gửi nó qua Internet là một ý tưởng đột phá trong lịch sử khoa học máy tính. Để gửi một tài liệu qua Internet, kết nối Internet của bạn phải đủ nhanh để upload file. à goài ra, file phải đủ nhỏ để truyền qua kết nối Internet trước khi sự kiện timeout xảy ra. à ếu nhà cung cấp dịch vụ Internet giới hạn kích thước file được truyền qua kết nối mà họ cung cấp, các chức năng e-mail có thể bị lảm trở ngại. Hơn nữa, việc đính kèm file vào e-mail có thể chiếm không gian trên server, và phải được xem xét một cách cẩn trọng. Một số nhà cung cấp e-mail phổ biến phải cân đối hàng triệu người sử dụng e-mail, và phải có chính sách lưu trữ file sao cho hợp lý. Thông thường, có sự giới hạn về kích thước file đính kèm được gửi qua e-mail server. Một số nhà cung cấp cho phép đính kèm lên đến 10MB; số khác còn cho phép lớn hơn. Hiện tượng này là một trở ngại từ nhiều năm nay bởi vì người dùng không thể gửi các file có dung lượng lớn cho đồng nghiệp và bạn bè thông qua kết nối Internet. à gười dùng phải làm gì?

Một giải pháp cho việc gửi các file đính kèm có dung lượng lớn vào e-mail là sử dụng website trung gian—thường được gọi là “file share”—đóng vai trò là một kho lưu trữ file trực tuyến. Website này có thể gửi một thông báo khi file được gửi đến server và cung cấp một liên kết cho người dùng nhấp vào để tải file về. Theo cách này, thực tế là bạn không phải gửi e-mail, mà là upload file. Giải pháp này đã được mô phỏng nhiều lần bởi nhiều website khác nhau. Chương này sử dụng “file share” nhằm giới thiệu một số tính năng mạnh và mới trong ASP.NET 2.0.

Các tính năng chủ yếu của File Share bao gồm:

- ✓ Khả năng upload một file lên website, chỉ định địa chỉ e-mail để gửi file này đến đó thông qua một liên kết trong e-mail để tải file về.
- ✓ Tự động gửi một e-mail đến người nhận, với thông điệp tùy biến và liên kết để tải file về.
- ✓ Có thể thay đổi nội dung của e-mail được gửi tự động, sử dụng các biến đặc trưng cho các giá trị của người gửi, người nhận, liên kết, và một thông điệp tùy biến gửi đến người nhận.
- ✓ Khả năng chỉ định thông tin SMTP server và thông tin tài khoản e-mail ở dạng một mục cấu hình, chứ không phải là một giá trị được viết cứng.
- ✓ Khả năng thay đổi cảm quan toàn bộ website bằng cách hiệu chỉnh một mục trong file cấu hình.

Chương này cũng phân tích các thành phần cấu thành website, bao gồm các điều kiểm và kỹ thuật đặc biệt cùng môi trường phát triển ASP.NET 2.0:

- ✓ Điều kiểm Login
- ✓ Điều kiểm PasswordRecovery
- ✓ Điều kiểm LoginStatus
- ✓ Điều kiểm Menu
- ✓ Điều kiểm SiteMapDataSource
- ✓ Theme
- ✓ Điều kiểm FileUpload

Phần “Thiết kế File Share” đi sâu vào bản thiết kế ứng dụng; bao gồm các thành phần thiết yếu liên quan đến kỹ thuật và cấu trúc của website, cũng như các lớp, các thành viên của mỗi lớp, giải thích về các bảng cơ sở dữ liệu, mối quan hệ và giá trị của chúng.

Phần “Mã lệnh File Share” tập trung vào mã lệnh của các lớp hay module quan trọng. Một số tiêu điểm: các webform dùng để upload file lên hệ thống, chèn dữ liệu vào các bảng cơ sở dữ liệu.

Hai phần cuối xét cách trích xuất và tùy biến File Share trong môi trường phát triển, cũng như cách cài đặt nó.

2.1

Sử dụng File Share

Sử dụng File Share cực kỳ dễ. Website này chỉ có một vài vùng chức năng, bởi mục đích của nó chỉ là upload file và gửi e-mail.

Ấu website File Share đã được cài đặt thành công (tham khảo phần “Cài đặt File Share”), bạn có thể duyệt theo địa chỉ <http://localhost/FileShare/> (xem hình 2-1).



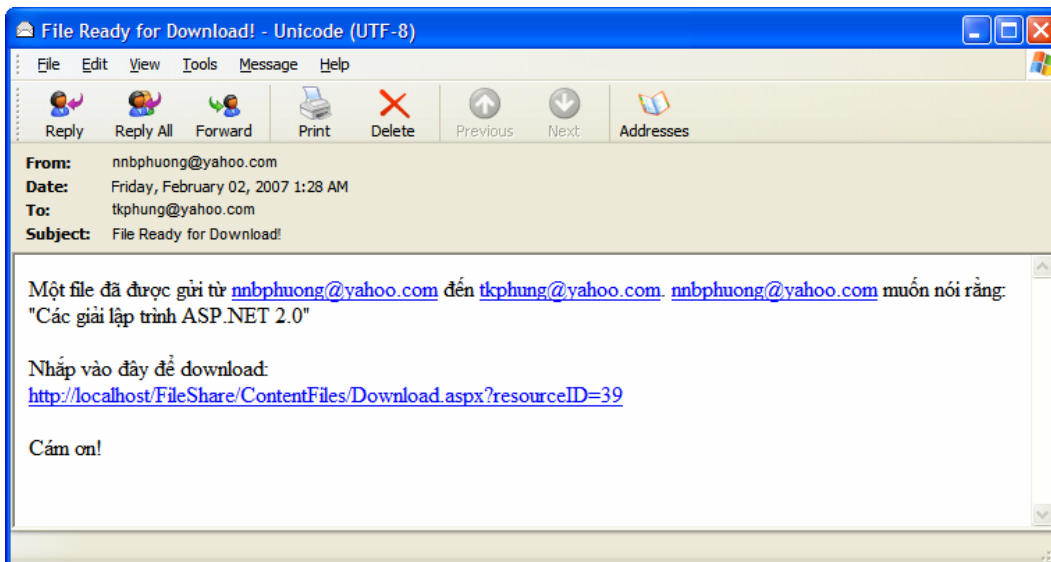
Hình 2-1

Trình đơn chính gồm các liên kết sau:

- ✓ Trang chủ
- ✓ Giới thiệu
- ✓ Liên hệ
- ✓ Admin

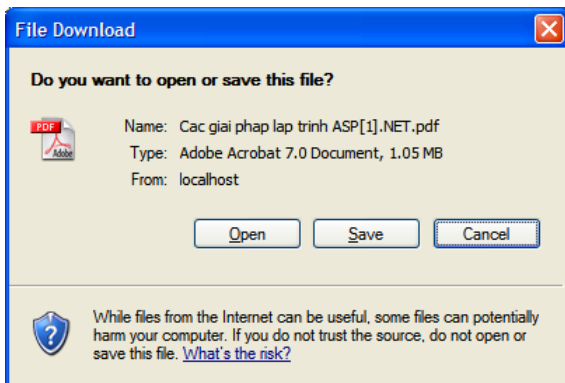
Trên trang chủ, có tổng cộng ba bước cần thiết để gửi một file có dung lượng lớn lên website. Các bước này gồm: lấy địa chỉ e-mail của người nhận, file cần gửi, một chú thích hay thông điệp đến người nhận, và địa chỉ e-mail của người gửi. Một khi đã có đầy đủ các trường này, nhấp nút Gửi để thực hiện upload và gửi e-mail.

Hình 2-2 là một ví dụ về một e-mail được gửi đến người nhận.



Hình 2-2

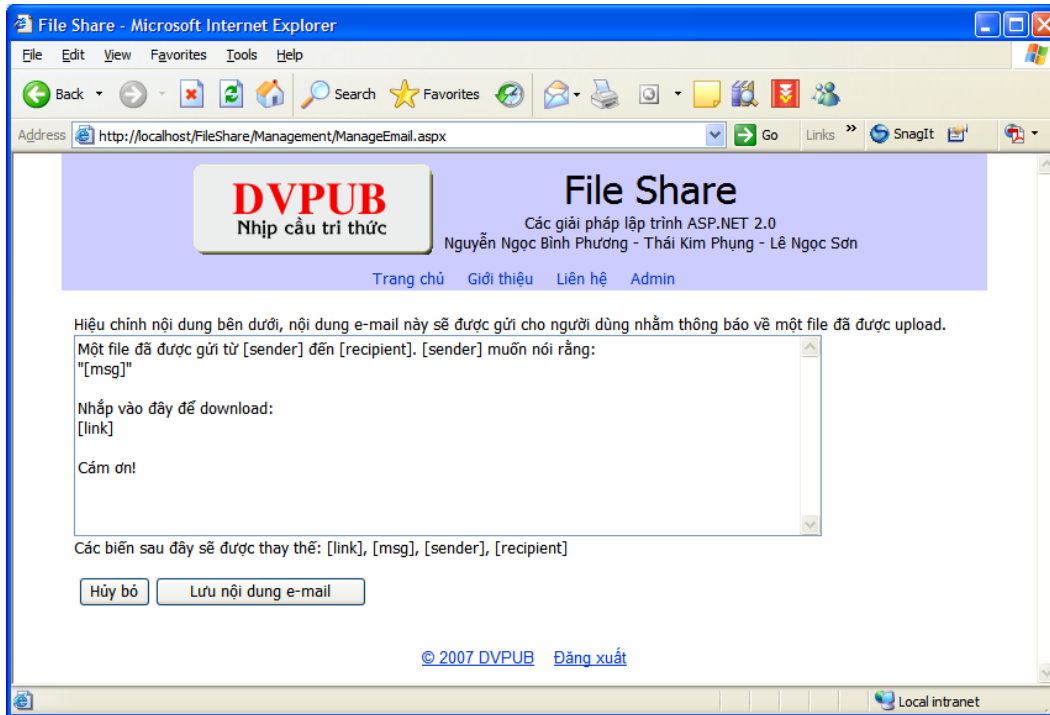
E-mail này chứa một liên kết mà thông qua đó, file gốc được gửi đến người nhận. Khi nhấp vào liên kết này, hộp thoại như hình 2-3 xuất hiện. Ấn Save để mở hộp thoại Save As, và chọn nơi lưu file. Bước này hoàn tất tác vụ gửi một file lớn đến e-mail của người nhận thông qua File Share.



Hình 2-3

Khi nhấp liên kết Admin trên trình đơn chính, bạn sẽ được chuyển đến màn hình đăng nhập nếu bạn chưa đăng nhập vào website và chưa tạo một phiên làm việc. Trang này chứa một điều kiện kiểm tra Login và một điều kiện kiểm tra passwordRecovery. Ắp Admin cho tên đăng nhập và password# cho mật khẩu, sau đó nhấp nút Đăng nhập.

Khi đã đăng nhập vào website, bạn được đưa đến trang quản trị, như hình 2-4.



Hình 2-4

Giao diện này cung cấp một lối tùy biến các e-mail gửi file đến người nhận. Các biến được sử dụng là liên kết [link], thông điệp [msg], e-mail người gửi [sender], e-mail người nhận [recipient]. Các biến này sẽ được thay thế trong phần thân của thông điệp e-mail.

Chương này bao quát các điểm chính của việc phát triển ứng dụng. Bạn sẽ đi chi tiết vào các file lớp, các phương thức và thuộc tính của chúng. Ắ ngoài ra, bạn cũng sẽ tìm hiểu về cơ sở dữ liệu, mô hình dữ liệu, và các đối tượng cơ sở dữ liệu.

Phần kế tiếp bàn về bản thiết kế ứng dụng File Share, bao gồm các lớp và các đối tượng cơ sở dữ liệu.

2.2

Thiết kế File Share

Bản thiết kế File Share dựa trên một số ý tưởng, bao gồm:

- ✓ File lưu trên server được xem như một lớp Resource.
- ✓ Các phương thức được sử dụng để lưu và lấy nội dung e-mail được lưu trữ trong lớp EmailContent.

- ✓ Mỗi đối tượng lớp nghiệp vụ có một đối tượng lớp dữ liệu nhằm thu lấy dữ liệu từ cơ sở dữ liệu hoặc chèn dữ liệu vào cơ sở dữ liệu.
- ✓ Bản thiết kế này cho thấy tầng nghiệp vụ và tầng dữ liệu được phân chia hợp lý.

Trong các phần tiếp theo sau, bạn sẽ biết cách upload file và gửi e-mail; thấy rõ cấu trúc của File Share; hiểu rõ mô hình dữ liệu và các đối tượng cơ sở dữ liệu, các theme trong website, và mô hình bảo mật. Bạn cũng sẽ tìm hiểu các lớp và phạm vi ảnh hưởng của chúng bên trong kiến trúc của website.

2.2.1 Upload file

Điều kiểm FileUpload được sử dụng để upload một file lên server. Ở đó hiển thị một điều kiểm TextBox bên cạnh nút Browse, cho phép người dùng chọn một file từ máy cục bộ của họ để upload lên server. Thể hiện FileUpload1 của điều kiểm FileUpload trưng ra các thuộc tính như FileName hay FileBytes, mà trước ASP.NET 2.0 khó lòng có được. Điều kiểm FileUpload cũng không tự động lưu file vào server khi người dùng chọn file và submit form chứa điều kiểm này. Logic trong form phải lưu file vào đĩa một cách tường minh. Mã lệnh này lưu file bằng phương thức SaveAs, phương thức này lưu file vào đường dẫn đã được chỉ rõ trên hệ thống file cục bộ của server.

2.2.2 Gửi e-mail

Để gửi e-mail trong ASP.NET 2.0, có nhiều điểm cần xem xét trong quá trình hoạch định và phát triển. Thứ nhất là sử dụng một SMTP server hợp lệ, với một tài khoản e-mail hợp lệ. Tài khoản e-mail được sử dụng phải cho phép quyền “relay mail”.

Các lớp mail của ASP.NET 2.0 thuộc không gian tên System.Net.Mail, cung cấp các tính chất và nội dung thiết yếu của một thông điệp mail. Lớp con SmtplibClient gửi e-mail đến SMTP server đã được chỉ định.

File Web.config cung cấp các thiết lập cấu hình cho SMTP server, bao gồm:

- ✓ EmailFrom
- ✓ EmailSubject
- ✓ SmtplibServer
- ✓ MailUser
- ✓ MailPassword
- ✓ MailPort
- ✓ EmailFormatSelected

Chúng được truy xuất từ lớp Utilities, trình bày nội dung của một biến cấu trúc. Biến cấu trúc này được khai báo ở phần đầu của lớp Utilities, như sau:

```

''' <summary>
''' MailSettings là cấu trúc dùng để định nghĩa thông tin mail server
''' </summary>
Public Structure MailSettings
    Public MailServer As String
    Public MailPort As Integer
    Public MailFrom As String
    Public MailUser As String
    Public MailPassword As String
End Structure

```

Thực tế, việc gửi e-mail được thực hiện trong lớp Utilities, bên trong hàm sau:

Chương 2. File Share

```

''' <summary>
''' SendEmail được sử dụng để gửi một e-mail với các thiết lập đã được xác định
''' </summary>
Public Shared Sub SendEmail(ByVal MsgTo As String, ByVal MsgFrom As String, _
    ByVal MsgSubject As String, ByVal MsgText As String)

    Dim SmtSettings As MailSettings
    SmtSettings = GetSmtSettings()
    Dim SmtC1 As New SmtClient(SmtSettings.MailServer, SmtSettings.MailPort)
    SmtC1.Credentials = GetCredentials(SmtSettings)
    Dim MailMsg As New MailMessage(MsgFrom, MsgTo)
    MailMsg.Subject = MsgSubject
    MailMsg.Body = MsgText

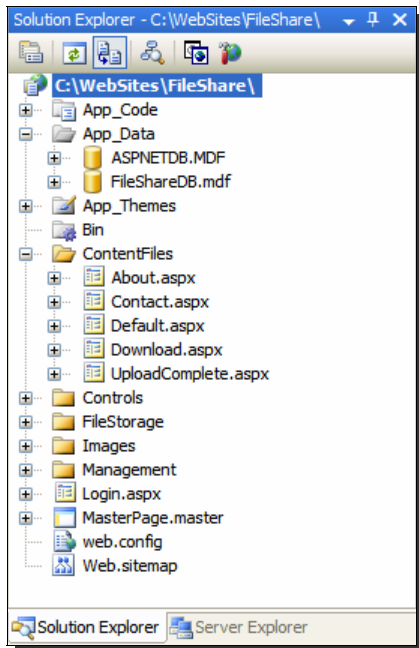
    SmtC1.Send(MailMsg)
End Sub

```

Đoạn mã trên trình bày cách sử dụng các lớp e-mail được xây dựng trong ASP.NET 2.0, và cách File Share hiện thực chức năng e-mail.

2.2.3 Cấu trúc website

Hình 2-5 cho thấy các file và thư mục của dự án trong Solution Explorer:



Hình 2-5

Cấu trúc file của website ASP.NET 2.0 đã được chuẩn hóa một ít từ các phiên bản trước của nó. Các chuẩn hóa này phải thực hiện với các quy ước đặt tên cho các thư mục bên trong website. Các thành phần của dự án được liệt kê trong bảng sau:

Thành phần	Mô tả
App_Code	Chứa lớp tầng nghiệp vụ (như resource.vb) và lớp tầng dữ liệu (như resourceDB.vb)
App_Data	Thư mục .NET chuẩn cho các file cơ sở dữ liệu.

App_Themes	Thư mục theme, gồm hai theme dùng cho website.
ContentFiles	Các file webform ASPX chuẩn cho việc hiển thị nội dung.
Controls	Lưu trữ tất cả điều kiện người dùng.
FileStorage	Thư mục chứa các file đã được upload để gửi e-mail cho người nhận.
Images	Lưu trữ các hình ảnh cho header và các trang khác.
Management	Chứa các trang webform quản trị được bảo mật.
[các file linh tinh]	Bao gồm trang đăng nhập, file config, file sitemap, và file master tại thư mục gốc của website.

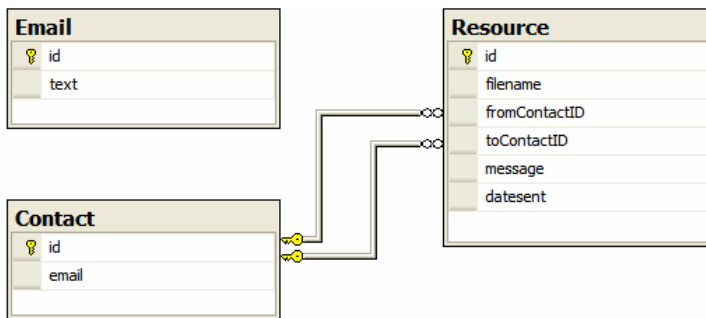
Một trong các thành phần thiết yếu của website File Share là cơ sở dữ liệu. Cơ sở dữ liệu này được cấu thành từ một file SQL Server 2005 Express, chứa đầy đủ các đối tượng cơ sở dữ liệu bên trong đó. Phần tiếp theo tập trung vào file cơ sở dữ liệu, cụ thể là các thủ tục tồn trữ và các bảng.

2.2.4 Mô hình dữ liệu và các đối tượng cơ sở dữ liệu

Mô hình dữ liệu rất đơn giản, chỉ cần lưu ba phần tử dữ liệu cơ bản:

- ✓ Email
- ✓ Resource
- ✓ Contact

Mỗi tài nguyên (resource) chứa các tham chiếu đến bảng Contact, với địa chỉ e-mail của người gửi và nhận các file được upload lên hệ thống. Hình 2-6 hiển thị lược đồ của các bảng cơ sở dữ liệu.



Hình 2-6

Phần dưới đây sẽ trình bày chi tiết ba bảng này.

Bảng Email

Tên trường	Kiểu dữ liệu	Mô tả
id	Int	Định danh duy nhất cho mẫu tin này.
text	nvarchar (MAX)	Nội dung e-mail được lưu ở dạng văn bản, người dùng có thể chỉnh sửa trong phần quản trị của website.

Bảng Contact

Tên trường	Kiểu dữ liệu	Mô tả
id	Int	Định danh duy nhất cho mẫu tin này.
email	varchar (200)	Địa chỉ e-mail của người này.

Bảng Resource

Tên trường	Kiểu dữ liệu	Mô tả
id	Int	Định danh duy nhất cho mẫu tin này.
filename	varchar (300)	Tên của file được upload
fromContactID	Int	ID của người đã gửi file này.
toContactID	Int	ID của người sẽ nhận file này.
message	nvarchar (300)	Thông điệp mà người gửi đã cung cấp cùng với file được upload.
datesent	Datetime	Thời điểm file được upload.

Âm ba bảng này, một số thủ tục tồn trữ cũng được sử dụng. Chúng tuân theo một kiểu đặt tên nhất quán với các chương khác, như sau:

- ✓ `sprocTableNameSelectList`
- ✓ `sprocTableNameSelectSingleItem`
- ✓ `sprocTableNameInsertUpdateItem`

Theo kiểu như thế, các thủ tục tồn trữ sau sẽ được sử dụng trong ứng dụng này:

- ✓ `sprocEmailInsertUpdateItem`
- ✓ `sprocEmailSelectSingleItem`
- ✓ `sprocResourceInsertUpdateItem`
- ✓ `sprocResourceSelectSingleItem`

Quy ước đặt tên này cho phép bạn tìm thấy dễ dàng và nhanh chóng các thủ tục tồn trữ áp dụng cho một bảng cụ thể; và biết chúng là lấy, chèn, cập nhật, hay xóa.

Có một vài thủ tục tồn trữ bạn cần xem qua. Thủ tục tồn trữ đầu tiên, `sprocResourceSelectSingleItem`, là một câu lệnh `SELECT` dựa vào tham số `@id`, lấy một mẫu tin tài nguyên từ cơ sở dữ liệu và trả nó về cho đối tượng gọi.

```
ALTER PROCEDURE dbo.sprocResourceSelectSingleItem
' Lấy một tài nguyên cụ thể từ cơ sở dữ liệu
(@id int)
```

```
AS
select * from Resource where id = @id
```

Thủ tục tồn trữ trên được gọi từ tầng dữ liệu `ResourceDB.vb`, trong hàm `GetResourceFileName`.

Tương tự như vậy, thủ tục tồn trữ kế tiếp, `sprocEmailSelectSingleItem`, được sử dụng để lấy một mẫu tin từ bảng `Email`. Không có tham số `@id` trong thủ tục này, bởi vì nó giả định rằng

hiện tại bạn chỉ lưu trữ một mẫu tin trong bảng Email. Ắ ều bạn muốn thêm các phiên bản hay các kiểu e-mail khác vào hệ thống để dùng sau này, đây chính là nơi quản lý thông tin đó.

```
ALTER PROCEDURE dbo.sprocEmailSelectSingleItem
/*
-- Lấy phần thân của e-mail từ cơ sở dữ liệu
*/
AS
select top 1 * from Email
```

Chuyển sang hai thủ tục tồn trữ khác, mức độ phức tạp tăng không đáng kể. Dưới đây là thủ tục tồn trữ tiếp theo, sprocEmailInsertUpdateItem, được sử dụng để cập nhật nội dung e-mail trong một mẫu tin của bảng Email. Ắ ó nhận một tham số @text, đó là nội dung e-mail mẫu sẽ được gửi đến người nhận.

```
ALTER PROCEDURE dbo.sprocEmailInsertUpdateItem
/*
-- Chèn hay cập nhật nội dung e-mail vào cơ sở dữ liệu
*/
(@text nvarchar(MAX))
AS
UPDATE Email SET [text] = @text
```

Thủ tục tồn trữ cuối cùng, sprocResourceInsertUpdateItem, khá phức tạp nhưng đừng quá lo lắng bởi ý tưởng cơ bản của nó khá đơn giản.

```
ALTER PROCEDURE dbo.sprocResourceInsertUpdateItem
/*
-- Chèn hay cập nhật một tài nguyên vào cơ sở dữ liệu
*/
(@id int,
 @filename varchar(300),
 @fromContactEmail varchar(300),
 @toContactEmail varchar(300),
 @message varchar(1000))
AS

DECLARE @returnValue int
Declare @fromContactID int
Declare @toContactID int
/*
----- fromContactID -----
*/
-- Chèn các mẫu tin liên lạc nếu chúng chưa tồn tại...
if((select count(*) from contact where email = @fromContactEmail)=0)
begin
insert into contact (email) values (@fromContactEmail)
SET @fromContactID = SCOPE_IDENTITY() -- Trích ID của liên lạc từ câu lệnh INSERT
end
else
begin
-- Trích ID của liên lạc từ câu lệnh SELECT
SET @fromContactID = (select id from contact where email = @fromContactEmail)
end

/*
----- toContactID -----
*/
if((select count(*) from contact where email = @toContactEmail)=0)
begin
insert into contact (email) values (@toContactEmail)
SET @toContactID = SCOPE_IDENTITY() -- Trích ID của liên lạc từ câu lệnh INSERT
end
```

```

else
  begin
    -- Trích ID của liên lạc từ câu lệnh SELECT
    SET @toContactID = (select id from contact where email = @toContactEmail)
  end

-- Chèn một mẫu tin tài nguyên mới
IF (@id IS NULL)
  BEGIN
    INSERT INTO
      Resource
      (
        filename,
        fromContactID,
        toContactID,
        message
      )
    VALUES
      (
        @filename,
        @fromContactID,
        @toContactID,
        @message
      )
    SET @returnValue = SCOPE_IDENTITY()
  END
ELSE
  BEGIN
    UPDATE
      Resource
    SET
      filename = @filename,
      fromContactID = @fromContactID,
      toContactID = @toContactID,
      message = @message
    WHERE
      Id = @id
    SET @returnValue = @id
  END

select @returnValue

```

Thủ tục này được sử dụng để chèn thông tin tài nguyên vào cơ sở dữ liệu, thêm các liên lạc mới vào bảng Contact, và các tài nguyên vào bảng Resource. Nó sử dụng phương pháp “upsert”, tức là sẽ cập nhật nếu mẫu tin đã tồn tại, hoặc chèn nếu mẫu tin chưa có.

Thủ tục `sprocResourceInsertUpdateItem` tuân theo các bước dưới đây:

Kiểm tra xem địa chỉ e-mail của người gửi (`@fromContactEmail`) có tồn tại trong hệ thống hay không:

```

/*
----- fromContactID -----
*/
-- Chèn các mẫu tin liên lạc nếu chúng chưa tồn tại...
if((select count(*) from contact where email = @fromContactEmail)=0)
...

```

Nếu không, thủ tục tồn trữ này thêm địa chỉ e-mail như một mẫu tin liên lạc mới, lấy giá trị ID gán cho biến cục bộ `@fromContactID` nhằm chèn vào bảng Resource sau này. Nếu mẫu tin đã tồn tại, thực hiện câu lệnh SELECT để lấy dữ liệu cho `@fromContactID`:

```

...
begin
  insert into contact (email) values (@fromContactEmail)

```

```

        SET @fromContactID = SCOPE_IDENTITY() -- Trích ID của liên lạc từ câu lệnh INSERT
    end
else
    begin
        -- Trích ID của liên lạc từ câu lệnh SELECT
        SET @fromContactID = (select id from contact where email = @fromContactEmail)
    end

```

Phần tiếp theo của thủ tục tồn trữ thực hiện điều giống như vậy, ngoại trừ lúc này với tham số @toContactEmail và đưa dữ liệu vào biến @toContactID.

Sau khi có ID của người nhận và người gửi, bạn tập trung vào việc chèn mẫu tin tài nguyên vào cơ sở dữ liệu. Phần dưới đây được sử dụng để chèn một mẫu tin tài nguyên mới vào bảng Resource, trả về ID mới của tài nguyên vào biến @resourceID:

```

-- Chèn một mẫu tin tài nguyên mới
IF (@id IS NULL)
    BEGIN
        INSERT INTO
            Resource
        (
            filename,
            fromContactID,
            toContactID,
            message
        )
        VALUES
        (
            @filename,
            @fromContactID,
            @toContactID,
            @message
        )
        SET @returnValue = SCOPE_IDENTITY()
    END
...

```

Câu lệnh ELSE dưới đây theo ngay sau mệnh đề IF, với trường hợp tham số @id khác rỗng. Khi đó, ứng dụng truyền một ID cho thủ tục tồn trữ, cho biết một mẫu tin tài nguyên đã tồn tại, và thủ tục tồn trữ thực hiện cập nhật thay cho chèn.

```

...
ELSE
    BEGIN
        UPDATE
            Resource
        SET
            filename = @filename,
            fromContactID = @fromContactID,
            toContactID = @toContactID,
            message = @message
        WHERE
            Id = @id
        SET @returnValue = @id
    END

```

```
select @returnValue
```

Đoạn mã trên thực hiện truy vấn UPDATE, và trả về biến kết quả @resourceID. Một khi biến @resourceID được gửi ngược về đối tượng gọi (tầng dữ liệu), quá trình chèn một tài nguyên vào hệ thống hoàn tất.

2.2.5 Theme và skin

Dự án File Share cung cấp một lối đơn giản để áp dụng theme và skin cho mỗi trang của website mà không phải hiệu chỉnh phần mã đánh dấu HTML cho bất cứ trang nào (thậm chí trang master). Bạn có thể áp dụng một theme cho toàn bộ website bằng cách điều chỉnh file Web.config trỏ đến tên theme (giả sử theme này tồn tại trong dự án, dưới thư mục App_Themes). Điều này được thực hiện bên trong mỗi form ASP.NET bằng cách sử dụng đoạn mã sau trong sự kiện khởi tạo trước của form.

```

''' <summary>
''' Sự kiện này phát sinh để khởi tạo trang. Nó cho phép thiết lập theme và tiêu đề trang,
''' các giá trị này được lấy từ Web.config thông qua các thuộc tính chia sẻ của lớp Config.
''' </summary>
Protected Sub Page_PreInit(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles Me.PreInit
    Page.Theme = Config.CurrentTheme
    Page.Title = Config.PageTitle
End Sub

```

Về cơ bản, đoạn mã này truy xuất các thuộc tính của lớp Config (được lấy từ file Web.config), và gán thành viên Theme của trang bằng giá trị theme hiện tại. Theo cách này, chỉ với một thay đổi nhỏ trong file Web.config, cảm quan của website sẽ thay đổi! Vị trí chính xác để đổi theme cho website là trong phần appSettings của Web.config:

```

<!--
<add key="CurrentTheme" value="CleanBlue" />
-->
<add key="CurrentTheme" value="CleanRed" />

```

Đoạn mã này hiển thị một trong hai theme (chỉ có một được kích hoạt). Bạn hãy hoán đổi hai giá trị này để thay đổi theme.

2.2.6 Mô hình bảo mật

File Share sử dụng ASP.NET 2.0 Forms Authentication với SQL Server Security Provider. Sử dụng provider này trong công cụ ASP.NET Security Administration, bạn thiết kế ban đầu sinh ra một cơ sở dữ liệu bảo mật mới. Cơ sở dữ liệu này sẽ được đưa vào dự án và được sử dụng để chứa tất cả thông tin tài khoản người dùng và các thiết lập bảo mật. Mô hình bảo mật này hiện thực Forms Authentication bên trong các điều kiểm bảo mật mới của ASP.NET 2.0, chẳng hạn như các điều kiểm được sử dụng để đăng nhập, hiển thị trạng thái đăng nhập, và khôi phục mật khẩu. Sử dụng công cụ ASP.NET Security Administration, bạn có thể tùy biến và thay đổi các thiết lập phù hợp, mà không phải thực thi dự án với vai trò quản trị.

Mô hình bảo mật này được sử dụng trong một số phần của ứng dụng, chẳng hạn thư mục Management của website. Nó cho phép bạn đăng nhập vào website và trở thành một người dùng được xác thực. Form Login.aspx tự động được nạp mỗi khi bạn cố truy xuất vào bất kỳ file ASPX nào trong thư mục Management mà bạn chưa được xác thực trước đó. Đây là một ý tưởng của mô hình bảo mật ASP.NET 2.0, được hiện thực thông qua Role and Membership Provider. Cấu hình như thế được thể hiện bởi điều kiểm ASP.NET Login:

```
<asp:Login ID="Login1" runat="server" />
```

Trên đây là ví dụ về một thư mục website mật và sử dụng “truy xuất dựa trên vai trò” (role-based access) để truy xuất các trang trong thư mục đó thông qua ASP.NET 2.0 Configuration Tool. Về bản chất, công cụ này được sử dụng cho việc quản lý các quyền bảo mật. ASP.NET 2.0 Configuration Tool có thể được truy xuất trong Visual Studio 2005 bằng cách nhấp vào Website |

ASP.NET Configuration trên trình đơn chính. Thẻ Security của công cụ này cho phép bạn hiệu chỉnh các thiết lập của bất kỳ thư mục nào bên trong website nhằm cho phép hoặc giới hạn khả năng truy xuất dựa trên những vai trò mà bạn có thể định nghĩa và gán cho người dùng. Kết quả là file Web.config được tạo ra trong thư mục mà bạn đã chỉ định giới hạn truy xuất. Ví dụ dưới đây là một file Web.config kết xuất:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.web>
    <authorization>
      <allow roles="Administrator" />
      <deny users="?" />
    </authorization>
  </system.web>
</configuration>
```

Hai mục chính trong file cấu hình này là phần cốt lõi của các thiết lập bảo mật. Về bản chất, chúng là các câu lệnh XML định nghĩa các quyền bảo mật cho thư mục đó, có phân cấp bên trong website, chép đề file Web.config trong thư mục gốc của website, cũng như file machine.config trên server. Trong file này, mục <deny users="?" /> nghĩa là thư mục này không cho phép người dùng chưa được xác thực, dấu chấm hỏi biểu thị điều đó. Mục <allow roles="Administrator" /> nghĩa là thư mục này cho phép truy xuất với vai trò Administrator.

Chỉ một tài khoản được tạo ra để sử dụng File Share, và một vai trò được gán cho tài khoản này. Tài khoản này như sau:

Tên người dùng	Mật khẩu	Mô tả tài khoản
Admin	password#	Â người dùng này được gán vai trò Administrator.

Vai trò dưới đây đã có trong cơ sở dữ liệu bảo mật và được tham chiếu bên trong ứng dụng:

Vai trò	Mô tả vai trò
Administrator	Vai trò này có thể đăng nhập vào phần quản trị, chỉnh sửa nội dung e-mail.

Bạn có thể kiểm soát việc truy xuất đến các phần tử form, các hàm, và các thư mục file bằng các vai trò bảo mật được hiện thực bởi ASP.NET Configuration Tool, hoặc thông qua mã lệnh kịch bản VB.NET của chính bạn. Với mô hình bảo mật ASP.NET 2.0 cơ bản này, bảo mật ứng dụng được thực hiện ở mức tối thiểu.

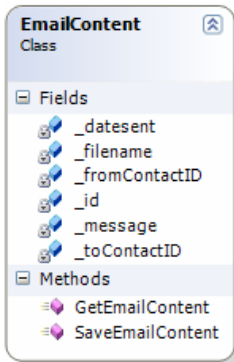
Trong phần tiếp theo, bạn sẽ đi sâu vào các lớp của ứng dụng, tìm hiểu về cách phân luồng thông tin bên trong website.

2.2.7 Các lớp chính yếu

File Share có một số lớp trình bày tầng nghiệp vụ và tầng dữ liệu của ứng dụng. Trong các cấu trúc lớp cơ bản này, bạn sẽ tìm thấy các phương thức và các thuộc tính cung cấp các tính năng trong ứng dụng.

Lớp EmailContent

Lớp EmailContent (xem hình 2-7) là lớp cho phép lưu và lấy nội dung e-mail từ cơ sở dữ liệu.



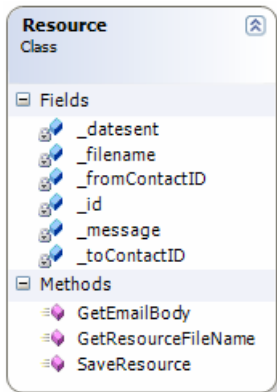
Hình 2-7

Các phương thức của lớp EmailContent được mô tả trong bảng sau:

Phương thức	Kiểu trả về	Mô tả
SaveEmailContent()	không	Lưu trữ nội dung e-mail, thông qua lớp EmailContentDB.
GetEmailContent()	String	Thu lấy e-mail từ cơ sở dữ liệu, thông qua lớp EmailContentDB.

Lớp Resource

Lớp Resource (xem hình 2-8) được sử dụng làm tầng nghiệp vụ của ứng dụng. Các phương thức của nó được đánh dấu là công khai và chia sẻ, nên dễ dàng sử dụng chúng bên trong các form và các điều kiện của ứng dụng. Điều này có nghĩa: bạn không cần tạo một thể hiện của lớp Resource để gọi các phương thức của nó. Thay vào đó, sử dụng cú pháp Resource.MethodName() trong bất kỳ webform hoặc điều kiện VB.NET nào của ứng dụng để thực thi hàm.



Hình 2-8

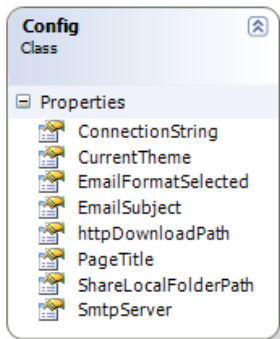
Bảng sau mô tả các thành viên có thể truy xuất của lớp Resource:

Phương thức	Kiểu trả về	Mô tả
GetEmailBody()	String	Trả về phần thân HTML của thông điệp e-mail sẽ được gửi đi.

GetResourceFileName()	String	Trả về tên tài nguyên (tên file) từ ID của tài nguyên.
SaveResource()	Integer	Lưu một tài nguyên mới vào cơ sở dữ liệu, truyền vào thông tin người gửi, thông tin người nhận, thông điệp, và tên file.

Lớp Config

Lớp Config (xem hình 2-9) được sử dụng như trình quản lý cấu hình của ứng dụng. Về bản chất, nó là điểm truy xuất chính cho tất cả các thiết lập cấu hình mà bất kỳ tầng ứng dụng nào cũng có thể truy xuất đến.



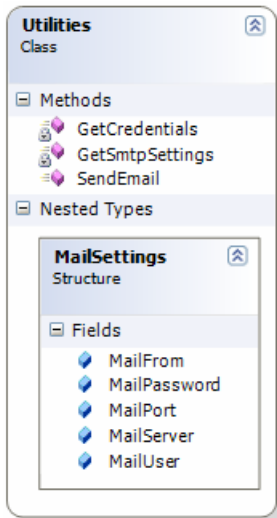
Hình 2-9

Bảng sau mô tả các thành viên có thể truy xuất của lớp Config:

Phương thức	Kiểu trả về	Mô tả
ConnectionString()	String	Thuộc tính chuỗi kết nối, được lấy từ file Web.config.
CurrentTheme()	String	Theme hiện tại của website, được định nghĩa trong file Web.config.
EmailFormatSelected()	String	Kiểu định dạng e-mail, các giá trị có thể được sử dụng là Text và Html.
EmailSubject()	String	Tiêu đề cho tất cả e-mail, báo cho người dùng biết về file được upload.
httpDownloadPath()	String	Mục cấu hình xác định đường dẫn của file Download.aspx, được thiết lập trong file Web.config.
PageTitle()	String	Giá trị tiêu đề HTML mà mỗi trang hiển thị, được định nghĩa trong file Web.config.
ShareLocalFolderPath()	String	Đường dẫn thư mục cục bộ cho tất cả các file được upload lên server, được lấy từ file Web.config.
SntpServer()	String	Mục cấu hình xác định địa chỉ và tên của SMTP server.

Lớp Utilities

Lớp Utilities được sử dụng cho chức năng gửi e-mail của ứng dụng File Share (hình 2-10).



Hình 2-10

Bảng sau mô tả các thành viên có thể truy xuất của lớp Utilities:

Phương thức	Kiểu trả về	Mô tả
GetCredentials()	System.Net .NetworkCredential	Tạo và trả về một tham chiếu đối tượng lớp System.Net.NetworkCredential với các giá trị cấu hình thích hợp.
GetSmtpSettings()	MailSettings (cấu trúc)	Được sử dụng để thu lấy các giá trị của file Web.config và gán chúng cho các thuộc tính của thể hiện cấu trúc.
SendEmail()	không	Được sử dụng để gửi e-mail, với các thiết lập đã được định sẵn.

Bạn đã xem qua các lớp của ứng dụng. Phần kế tiếp, “Mã lệnh File Share”, sẽ đi sâu vào việc phát triển và dẫn bạn qua tất cả các đoạn mã thiết yếu.

2.3

Mã lệnh File Share

Phần này giải thích các file mã lệnh chính trong dự án File Share. Bạn sẽ xem chi tiết các file này trong mỗi thư mục và tìm hiểu chúng tương tác thế nào và được sử dụng trong dự án ra sao.

2.3.1 Một số file quan trọng

Thư mục gốc của File Share chứa một số file quan trọng, bao gồm các trang ASPX chính, các trang cấu hình và định dạng.

Web.config

Web.config lưu trữ các mục cấu hình quan trọng dùng trong ứng dụng. Mục `SqlServerConnectionString` điều khiển kết nối đến file cơ sở dữ liệu SQL Server 2005 Express `FileShareDB.mdf`:

```
<connectionStrings>
  <add name="ConnectionString" connectionString="Data Source=(local)\SqlExpress;
    AttachDbFilename=|DataDirectory|\FileShareDB.mdf;Integrated Security=True;
    User Instance=True" providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Web.config cũng chứa thông tin quản lý các thiết lập SMTP để gửi e-mail.

```
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
<appSettings>
  <add key="EmailFrom" value="admin@FileShare.com"/>
  <add key="EmailSubject" value="File Ready for Download!"/>
  <add key="SmtpServer" value="127.0.0.1"/>
  <add key="MailUser" value="myalias"/>
  <add key="MailPassword" value="mypassword"/>
  <add key="MailPort" value="25"/>
  <add key="EmailFormatSelected" value="Text"/>
  <add key="PageTitle" value="File Share"/>
  <add key="ShareLocalFolderPath" value="C:\Inetpub\wwwroot\FileShare\FileStorage\"/>
  <add key="httpDownloadPath" value="http://localhost/FileShare/ContentFiles"/>
  <!--
  <add key="CurrentTheme" value="CleanBlue" />
  -->
  <add key="CurrentTheme" value="CleanRed"/>
</appSettings>
```

Các mục cấu hình SMTP này là nơi bạn có thể chỉ định các thiết lập mail server mà ứng dụng sẽ sử dụng để gửi e-mail đến người nhận. File `Web.config` cũng được sử dụng để hiệu chỉnh các theme dùng cho toàn bộ website. Vấn đề này đã được giải thích chi tiết trong phần “Theme và skin”.

config.vb

Lớp `Config` được sử dụng như một đối tượng nghiệp vụ sẵn dùng cho các giá trị và các thiết lập thông qua sự hiện diện của một số thành viên tĩnh. Các thành viên của nó được liệt kê như các thuộc tính nhằm trừu tượng hóa vị trí lưu trữ các giá trị này. Giá trị của các thuộc tính này được lưu trữ trong file `Web.config`, và lớp `Config` lấy chúng khi cần thiết.

```
Imports Microsoft.VisualBasic
```

```
Public Class Config
```

```
    ‘‘‘ <summary>
    ‘‘‘ Thuộc tính chuỗi kết nối, được lấy từ file Web.config
    ‘‘‘ </summary>
    Public Shared ReadOnly Property ConnectionString() As String
        Get
            Return
                ConfigurationManager.ConnectionStrings("ConnectionString").ConnectionString
        End Get
    End Property

    ‘‘‘ <summary>
    ‘‘‘ Theme hiện tại của website, được định nghĩa trong file Web.config
    ‘‘‘ </summary>
    Public Shared ReadOnly Property CurrentTheme() As String
        Get
```

```

        Return ConfigurationManager.AppSettings("CurrentTheme").ToString()
    End Get
End Property

''' <summary>
''' Giá trị tiêu đề HTML mà mỗi trang hiển thị, được định nghĩa trong file Web.config
''' </summary>
Public Shared ReadOnly Property PageTitle() As String
    Get
        Return ConfigurationManager.AppSettings("PageTitle").ToString()
    End Get
End Property

''' <summary>
''' Đường dẫn thư mục cục bộ cho tất cả các file được upload lên server, được lấy từ file Web.config
''' </summary>
Public Shared ReadOnly Property ShareLocalFolderPath() As String
    Get
        Return ConfigurationManager.AppSettings("ShareLocalFolderPath").ToString()
    End Get
End Property

''' <summary>
''' Tiêu đề cho tất cả e-mail, báo cho người dùng biết về file được upload
''' </summary>
Public Shared ReadOnly Property EmailSubject() As String
    Get
        Return ConfigurationManager.AppSettings("EmailSubject").ToString()
    End Get
End Property

''' <summary>
''' Mục cấu hình xác định phần thân của e-mail là Html hay Text
''' </summary>
Public Shared ReadOnly Property EmailFormatSelected() As String
    Get
        Return ConfigurationManager.AppSettings("EmailFormatSelected").ToString()
    End Get
End Property

''' <summary>
''' Mục cấu hình xác định địa chỉ và tên của SMTP server
''' </summary>
Public Shared ReadOnly Property SmtServer() As String
    Get
        Return ConfigurationManager.AppSettings("SmtServer").ToString()
    End Get
End Property

''' <summary>
''' Mục cấu hình xác định đường dẫn của file Download.aspx, được thiết lập trong file Web.config
''' Giá trị mặc định là: "http://localhost/FileShare/ContentFiles/"
''' </summary>
Public Shared ReadOnly Property httpDownloadPath() As String
    Get
        Return ConfigurationManager.AppSettings("httpDownloadPath").ToString()
    End Get
End Property
End Class

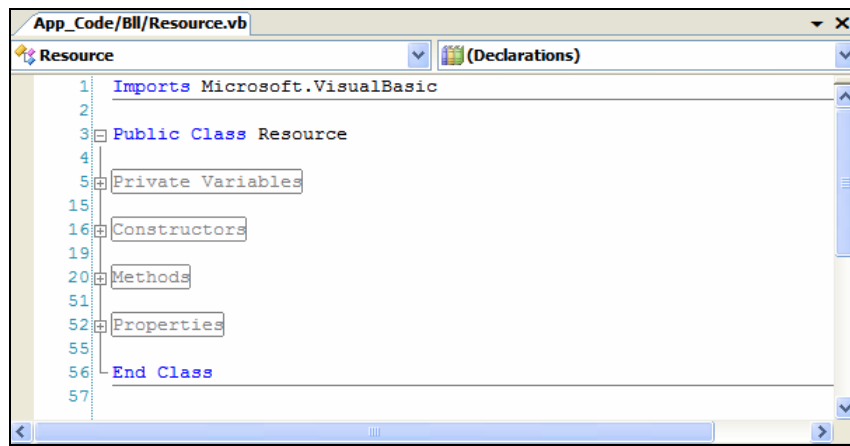
```

Trong đoạn mã trên, các thuộc tính được đánh dấu là `Public Shared ReadOnly`, điều này cho phép chúng được truy xuất từ bất cứ đâu trong dự án bằng ký hiệu `config-chấm` (`config.`). Ví dụ, `config.ConnectionString()` trả về chuỗi kết nối từ lớp `Config`, mà không phải thể hiện hóa một đối tượng `Config` trước.

Resource.vb

Lớp Resource được sử dụng để thu lấy và lưu trữ thông tin tài nguyên được gửi lên website. Lớp này đóng vai trò là tầng nghiệp vụ và cung cấp một mức trừu tượng giữa các yêu cầu cho các mẫu tin cơ sở dữ liệu và giao diện người dùng.

Bằng cách sử dụng các thẻ #Region trong file lớp Resource.vb, Visual Studio IDE cho phép trang được nhóm thành các phần có trật tự. Các phần thường được sử dụng để nhóm mã lệnh theo cách này bao gồm: Variables, Constructors, Methods, và Properties. Điều này không ảnh hưởng đến các assembly .NET, nhưng là một cách hay để bảo trì mã lệnh. Hình 2-11 hiển thị trực quan các vùng mã lệnh đã được nhóm bên trong Visual Studio IDE.



Hình 2-11

Một trong các phương thức quan trọng của Resource là SaveResource, mã lệnh như sau:

```

''' <summary>
''' Lưu <see cref="Resource" /> vào cơ sở dữ liệu
''' </summary>
''' <param name="filename">Tên file tài nguyên.</param>
''' <param name="fromContactEmail">E-mail của người gửi.</param>
''' <param name="message">Thông điệp về tài nguyên.</param>
''' <param name="toContactEmail">E-mail của người nhận.</param>
''' <param name="ID">Tham số tùy chọn: id của tài nguyên.</param>
Public Shared Function SaveResource(ByVal filename As String,
    ByVal fromContactEmail As String, ByVal toContactEmail As String, _
    ByVal message As String) As Integer
    Return ResourceDB.SaveResource(filename, fromContactEmail, toContactEmail, message)
End Function

```

Phương thức này cung cấp phương tiện để chuyển một đối tượng Resource đến tầng dữ liệu. Nó nhận năm tham số:

- ✓ filename
- ✓ fromContactEmail
- ✓ toContactEmail
- ✓ message
- ✓ ID (tùy chọn)

Các tham số này thể hiện toàn bộ khung nhìn của lớp Resource trong hệ thống.

resourceDB.vb

Lớp resourceDB là tầng dữ liệu của ứng dụng. Nó cung cấp các lời gọi phương thức để lấy thông tin từ cơ sở dữ liệu, cũng như chèn hay cập nhật dữ liệu bên trong cơ sở dữ liệu. Lớp này được xem là file hay đối tượng duy nhất có khả năng truy xuất đến các file cơ sở dữ liệu. Theo cách này, bạn có thể tách các thao tác đặc thù dữ liệu ra khỏi tầng nghiệp vụ. Bạn có thể thấy rằng điều này giúp nhà phát triển không viết mã lệnh truy xuất dữ liệu trùng nhau và thích ứng với chức năng bảo trì logic truy xuất dữ liệu có cấu trúc và có tổ chức. Điều này cũng hỗ trợ ứng dụng được tách thành các tầng một cách hợp lý với tính khả thi trong việc di dời và mở rộng ứng dụng trên các server riêng biệt bất cứ lúc nào.

Dưới đây là phương thức SaveResource của lớp resourceDB:

```

''' <summary>
''' Lưu <see cref="Resource" /> vào cơ sở dữ liệu
''' </summary>
''' <param name="filename">Tên file tài nguyên.</param>
''' <param name="fromContactEmail">E-mail của người gửi.</param>
''' <param name="message">Thông điệp về tài nguyên.</param>
''' <param name="toContactEmail">E-mail của người nhận.</param>
''' <param name="ID">Tham số tùy chọn: id của tài nguyên.</param>
Public Shared Function SaveResource(ByVal filename As String, _
    ByVal fromContactEmail As String, ByVal toContactEmail As String, _
    ByVal message As String, Optional ByVal ID As Integer = Nothing) As Integer

    Using mConnection As New SqlConnection(Config.ConnectionString)
        Dim mResourceID As Integer
        ' Tạo một đối tượng câu lệnh
        Dim mCommand As SqlCommand = New
            SqlCommand("sprocResourceInsertUpdateItem", mConnection)
        ' Thiết lập kiểu câu lệnh là 'thủ tục tồn trữ'
        mCommand.CommandType = CommandType.StoredProcedure
        ' Thêm vào các tham số
        If ID > 0 Then
            mCommand.Parameters.AddWithValue("@id", ID)
        Else
            mCommand.Parameters.AddWithValue("@id", DBNull.Value)
        End If
        mCommand.Parameters.AddWithValue("@filename", filename)
        mCommand.Parameters.AddWithValue("@fromContactEmail", fromContactEmail)
        mCommand.Parameters.AddWithValue("@toContactEmail", toContactEmail)
        mCommand.Parameters.AddWithValue("@message", message)
        ' Mở kết nối và thực thi thủ tục tồn trữ
        mConnection.Open()
        mResourceID = mCommand.ExecuteScalar()
        mConnection.Close()
        Return mResourceID
    End Using
End Function

```

Một phương thức khác là GetEmailBody(), phương thức này trả về một biến chuỗi là phần thân của mẫu e-mail được sử dụng để gửi đến người nhận file. Dưới đây là mã lệnh của phương thức này:

```

''' <summary>
''' Trả về phần thân HTML của thông điệp e-mail sẽ được gửi đi
''' </summary>
''' <param name="msg">Thông điệp do người dùng cung cấp để đưa vào phần thân của e-mail.</param>
Public Shared Function GetEmailBody(ByVal msg As String, _
    ByVal id As Integer, ByVal SenderEmail As String, _
    ByVal RecipientEmail As String) As String
    Dim emailBody As String = ""

```



```

Try
    Using mConnection As New SqlConnection(Config.ConnectionString)
        Dim mLink As String
        mLink = Config.httpDownloadPath & "Download.aspx?resourceID="

        Dim mCommand As SqlCommand = New _
            SqlCommand("sprocEmailSelectSingleItem", mConnection)
        mCommand.CommandType = CommandType.StoredProcedure
        mConnection.Open()
        Using mDataReader As SqlDataReader = _
            mCommand.ExecuteReader(CommandBehavior.CloseConnection)
            If mDataReader.Read() Then
                ' Lấy nội dung e-mail từ bảng Email
                emailBody = mDataReader.GetString( _
                    mDataReader.GetOrdinal(Config.EmailFormatSelected))
                ' Thay thế giá trị của các biến động
                emailBody = emailBody.Replace("[msg]", msg)
                emailBody = emailBody.Replace("[link]", mLink & id.ToString())
                emailBody = emailBody.Replace("[sender]", SenderEmail)
                emailBody = emailBody.Replace("[recipient]", RecipientEmail)
            End If
            mDataReader.Close()
        End Using
    End Using
Catch ex As Exception
    ' Bằng cách gọi lệnh "Throw", bạn dựng lên lỗi file Global.asax, và nó sẽ sử dụng
    ' trang thụ lý lỗi mặc định để xử lý/hiển thị lỗi tùy biến cho người dùng
    Throw
End Try
Return emailBody
End Function

```

Đoạn mã trên thực hiện các bước sau:

Tạo một đối tượng `SqlCommand` mới, truyền cho nó tên thủ tục tồn trữ và kết nối:

```
Using mConnection As New SqlConnection(Config.ConnectionString)
```

Tạo một biến cục bộ để nối thành liên kết thực dựa trên kiểu thiết kế vận hành theo cấu hình:

```
Dim mLink As String
mLink = Config.httpDownloadPath & "Download.aspx?resourceID="
```

Gán `CommandType` là `StoredProcedure`, và cung cấp tên của thủ tục tồn trữ là `sprocEmailSelectSingleItem`:

```
Dim mCommand As SqlCommand = New SqlCommand( _
    "sprocEmailSelectSingleItem", mConnection)
mCommand.CommandType = CommandType.StoredProcedure
```

Tạo một `SqlDataReader` mới:

```
Using mDataReader As SqlDataReader = _
    mCommand.ExecuteReader(CommandBehavior.CloseConnection)
```

Gọi phương thức `Execute` của đối tượng câu lệnh. Phương thức này thực thi thủ tục tồn trữ `sprocEmailSelectSingleItem` và trả về kết quả là một giá trị chuỗi trong một dòng một-mẫu-tin:

```
mCommand.ExecuteReader(CommandBehavior.CloseConnection)
```

Gán giá trị theo thứ tự của dữ liệu bên trong `DataReader` cho biến chuỗi `emailBody`:

```
If mDataReader.Read() Then
```

```

' Lấy nội dung e-mail từ bảng Email
emailBody = mDataReader.GetString( _
    mDataReader.GetOrdinal(Config.EmailFormatSelected))
...

```

Thay thế giá trị của các biến động: thông điệp từ người gửi, liên kết được sử dụng để tải file, địa chỉ e-mail của người gửi, và địa chỉ e-mail của người nhận.

```

' Thay thế giá trị của các biến động
emailBody = emailBody.Replace("[msg]", msg)
emailBody = emailBody.Replace("[link]", mLink & id.ToString())
emailBody = emailBody.Replace("[sender]", SenderEmail)
emailBody = emailBody.Replace("[recipient]", RecipientEmail)

```

Trả về giá trị chuỗi cho đối tượng gọi:

```
Return emailBody
```

Hàm này cung cấp chức năng lấy phần thân của thông điệp e-mail từ cơ sở dữ liệu, và trả về nội dung văn bản cho đối tượng gọi.

2.3.2 Webform

Webform là các trang ASPX chuẩn, chứa giao diện người dùng đồ họa phía client của ứng dụng. Các phần dưới đây sẽ giới thiệu một số webform đặc biệt quan trọng trong dự án File Share.

Default.aspx

File Default.aspx dĩ nhiên là trang đầu tiên được nạp khi truy xuất website. Bên trong trang này là các điều kiện dùng để bắt lấy thông tin về file khi người dùng upload file lên hệ thống.

Một số hàm đặc trưng trong trang này cần được xem xét. Thứ nhất là một sự kiện với logic có phần dư thừa trong toàn bộ các trang ASPX của ứng dụng. Đây là sự kiện khởi tạo trước của trang, mã lệnh như sau:

```

''' <summary>
''' Sự kiện này phát sinh để khởi tạo trang. Nó cho phép thiết lập theme và tiêu đề trang,
''' các giá trị này được lấy từ Web.config thông qua các thuộc tính chia sẻ của lớp Config.
''' </summary>
Protected Sub Page_PreInit(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles Me.PreInit
    Page.Theme = Config.CurrentTheme
    Page.Title = Config.PageTitle
End Sub

```

Page.Theme là thuộc tính theme của tham chiếu Page. Bằng cách thiết lập thuộc tính Theme là giá trị CurrentTheme của lớp Config, bạn đã gán một theme lúc thực thi. Kiểu gán này thực sự lý tưởng, bởi vì mỗi trang có thể sử dụng động theme—được điều khiển thông qua file Web.config, mà không cần bất kỳ thay đổi nào trên các file khác.

Một điểm khác cần quan tâm là hàm xử lý sự kiện Click của nút btnSend. Hàm này thực hiện upload file, lưu thông tin về file, và gửi một e-mail đến người nhận.

Ba quá trình cho việc gửi và lấy file về như sau:

- ✓ Dữ liệu văn bản về file được chèn vào cơ sở dữ liệu.
- ✓ File thực tế được upload lên thư mục lưu trữ của server.
- ✓ Một e-mail được gửi đến người nhận, báo rằng có thể tải file về.

Mã lệnh cho sự kiện Click như sau:

```

Protected Sub btnSend_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles btnSend.Click
    If FileUpload1.FileName <> "" Then
        ' Upload file lên server
        FileUpload1.SaveAs(Config.ShareLocalFolderPath + FileUpload1.FileName)

        ' Lưu thông tin vào cơ sở dữ liệu
        Dim ResourceID As Integer = Resource.SaveResource( _
            FileUpload1.FileName, txtSenderEmail.Text, _
            txtRecipientEmail.Text, txtMessage.Text)

        ' Lấy phần thân của thông điệp e-mail
        Dim emailBody As String = Resource.GetEmailBody( _
            txtMessage.Text, ResourceID, txtSenderEmail.Text, _
            txtRecipientEmail.Text)

        ' Gửi e-mail đến người nhận
        Utilities.SendEmail(txtRecipientEmail.Text, _
            txtSenderEmail.Text, Config.EmailSubject, emailBody)
        Server.Transfer("UploadComplete.aspx", True)
    End If
End Sub

```

Sự kiện này thực hiện phần logic cốt yếu nhất bên trong ứng dụng, và là điểm nút cho nỗ lực lập trình để làm chủ một ứng dụng chia sẻ file theo kiểu này. Sự kiện này cực kỳ quan trọng bởi vì nó thực hiện các chức năng upload, lưu trữ, gửi e-mail. Ắt hẳn muốn thêm các tính năng vào ứng dụng, bạn có thể bắt đầu từ đây và thực hiện theo cách của bạn trong các tầng sâu hơn của ứng dụng.

Login.aspx

Trang đăng nhập chứa điều kiểm Login và điều kiểm PasswordRecovery. Trang này nằm ở thư mục gốc của website và không sử dụng trang master. Các điều kiểm đăng nhập chứa mã đánh dấu HTML (được thể hiện trong mã lệnh bên dưới) định nghĩa các giá trị cụ thể cho trang đích và các giá trị văn bản của các điều kiểm.

```

<fieldset style="height: 128px; width: 270px;">
<asp:Login ID="Login1" runat="server" DestinationPageUrl="~/Management/ManageEmail.aspx">
    ...
</asp:Login>
</fieldset>
<br />
<br />
<fieldset style="height: 118px; width: 270px;">
<asp>PasswordRecovery ID="PasswordRecovery1" runat="server">
    ...
</asp>PasswordRecovery>
</fieldset>

```

Đoạn mã đánh dấu HTML trên chứa các định nghĩa cho điều kiểm Login và PasswordRecovery cùng các thuộc tính của chúng.

Download.aspx

Webform Download.aspx được sử dụng để truy xuất các file được lưu trữ bên trong website mà không bắt người dùng phải xem một trang web đầy những quảng cáo.

Sự kiện Page_Load của trang này như sau:

```

''' <summary>
''' Sự kiện này phát sinh để xử lý việc hiển thị hộp thoại download file
''' </summary>

```

Chương 2. File Share

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    DisplayDownloadDialog(Config.ShareLocalFolderPath & _
        Resource.GetResourceFileName(Request.QueryString("resourceID")))
End Sub
```

Sự kiện này gọi hàm `DisplayDownloadDialog`, truyền `Config.ShareLocalFolderPath` và một biến chuỗi truy vấn (`resourceID`). `Config.ShareLocalFolderPath` là thuộc tính của lớp `Config`, tham khảo đến thư mục lưu trữ file trên website. Điều này tính đến một kết xuất trực tiếp của đối tượng file bằng mã lệnh, bởi vì đó là một con trỏ cục bộ chỉ đến file này trên server. Biến chuỗi truy vấn `resourceID` là ID của mẫu tin tài nguyên mà trang sẽ tham chiếu khi trả về một file cho client. Mỗi file trong File Share có một số ID, được nhận biết qua biến `resourceID`. ID này cho phép bạn truy vấn dữ liệu và trích ra tên file để tải về.

Dưới đây là hàm `DisplayDownloadDialog` từ trang code-behind `download.aspx.vb`:

```
Sub DisplayDownloadDialog(ByVal PathVirtual As String)
    Dim strPhysicalPath As String
    Dim objFileInfo As System.IO.FileInfo
    Try
        ' strPhysicalPath = Server.MapPath(PathVirtual)
        strPhysicalPath = PathVirtual
        ' Thoát nếu file không tồn tại
        If Not System.IO.File.Exists(strPhysicalPath) Then
            Exit Sub
        End If
        objFileInfo = New System.IO.FileInfo(strPhysicalPath)
        Response.Clear()

        ' Thêm các header để hiển thị hộp thoại
        Response.AddHeader("Content-Disposition", _
            "attachment; filename=" & objFileInfo.Name)
        Response.AddHeader("Content-Length", objFileInfo.Length.ToString())
        Response.ContentType = "application/octet-stream"
        Response.WriteFile(objFileInfo.FullName)
    Catch ex As Exception
        ' Bằng cách gọi lệnh "Throw", bạn dựng lên lỗi file Global.asax, và nó sẽ sử dụng
        ' trang thụ lý lỗi mặc định để xử lý/hiển thị lỗi tùy biến cho người dùng
        Throw
    Finally
        Response.End()
    End Try
End Sub
```

Đoạn mã này cung cấp một file dưới dạng hộp thoại Open or Save, đã được giới thiệu trong phần “Sử dụng File Share”.

- ✎ Thay cho một trang web hiển thị các quảng cáo, File Share cung cấp một file dạng stream trực tiếp đến đối tượng gọi HTTP từ cú nhấp chuột vào một liên kết. Đáp ứng của trang được xóa trắng, các header được thêm vào đó, và đối tượng file được ghi vào đó trước khi nó được trả về cho đối tượng gọi. Điều này cho phép trang trả về một file—một lối tiếp cận nhanh gọn để download file.

Trang download có thể mở rộng theo kiểu: ứng dụng được hiệu chỉnh để một trang khác được nạp trước với một liên kết đến trang `download.aspx`, truyền vào đó một biến chuỗi truy vấn `resourceID`. Ồn ào thế, một file có thể được download hay download lại theo lối an toàn hơn, bằng cách sử dụng một nút hoặc một nút liên kết để khởi tạo yêu cầu. Cách này thường được sử dụng một cách toàn diện, bởi vì có nhiều trang download đầy những quảng cáo và phần tử có vấn đề—đây là nguyên nhân khiến cho việc download thất bại.

2.3.3 Các điều kiểm người dùng

Một số điều kiểm người dùng đặc thù trong website này giúp điều hướng và hiển thị nội dung cho nhiều trang. ả hằm tạo và sử dụng mã lệnh có khả năng tái sử dụng, các điều kiểm người dùng đã được tạo ra để áp dụng cho nhiều trang của website, tùy thuộc vào bản chất của các điều kiểm.

header.ascx

Điều kiểm người dùng header được sử dụng làm phần trên của website. ả ếu một thứ nào đó cần được hiển thị ở phần trên của một trang web, bạn hãy thêm nó vào điều kiểm header để nó có thể được nhìn thấy ở tất cả các trang.

Đoạn mã dưới đây trình bày toàn bộ mã nguồn của điều kiểm header:

```
<%@ Control Language="VB" AutoEventWireup="false" CodeFile="header.ascx.vb"
    Inherits="Controls_header" %>
<div style="text-align: center">
    <table>
        <tr>
            <td></td>
            <td><h1><% Response.Write(Page.Title) %></h1></td>
        </tr>
    </table>
</div>
```

Chú ý rằng các thẻ <% Response.Write(Page.Title) %> được sử dụng để in tiêu đề của website, giá trị này được lấy từ file Web.config.

footer.ascx

Điều kiểm người dùng footer được sử dụng làm phần dưới của website, cho mỗi trang có sử dụng trang master. Đây là một điều kiểm được tham chiếu bên trong trang master. Theo cách này, nó được nhân bản cho tất cả các trang với dáng vẻ giống hệt nhau.

ả ội dung của điều kiểm footer như sau:

```
<%@ Control Language="VB" AutoEventWireup="false" CodeFile="footer.ascx.vb"
    Inherits="Controls_footer" %>
<a href="http://dvpublish.com.vn" target="_blank">&copy; 2007 DVPUB</a>&nbsp; &nbsp; &nbsp;
<asp.LoginStatus ID="LoginStatus1" runat="server"
    LogoutAction="RedirectToLoginPage" LogoutPageUrl="~/Login.aspx"
    LoginText="Đăng nhập" LogoutText="Đăng xuất" />
```

Đoạn mã trên gồm một tham chiếu đến điều kiểm LoginStatus—một điều kiểm mới trong ASP.NET 2.0. Điều kiểm này hiển thị một nút liên kết cho chức năng đăng nhập và đăng xuất:

- ✓ Khi người dùng đã đăng nhập vào website, điều kiểm LoginStatus hiển thị nút liên kết Đăng xuất. ả hắp vào nút liên kết Đăng xuất, người dùng sẽ đăng xuất khỏi website và được đưa đến trang đăng nhập.
- ✓ Khi người dùng đã đăng xuất khỏi website, điều kiểm LoginStatus hiển thị nút liên kết Đăng nhập. ả hắp vào nút liên kết Đăng nhập, người dùng sẽ được đưa đến trang đăng nhập.

navigation.ascx

Điều kiểm người dùng navigation được sử dụng để cung cấp một trình đơn có thể tái sử dụng trên mỗi trang. Bản thân điều kiểm Menu là một điều kiểm mới trong ASP.NET 2.0, kết với điều kiểm SiteMapDataSource cũng là một điều kiểm mới trong .NET Framework 2.0. Điều kiểm

SiteMapDataSource được sử dụng để kết đến một file XML, các file của website được liệt kê như các mục trong file XML. Đây là nơi bạn có thể thay đổi dữ liệu cho trình đơn của website.

Dưới đây là mã đánh dấu HTML của điều kiểm navigation:

```
<%@ Control Language="VB" AutoEventWireup="false" CodeFile="navigation.ascx.vb"
    Inherits="Controls_navigation" %>
<asp:Menu ID="Menu1" runat="server" DataSourceID="SiteMapDataSource1"
    Orientation="Horizontal" StaticDisplayLevels="2"></asp:Menu>
<asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server" />
```

File XML của điều kiểm SiteMapDataSource như sau:

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
    <siteMapNode url="ContentFiles/default.aspx" title="Trang chủ">
        <siteMapNode url="ContentFiles/about.aspx" title="Giới thiệu" />
        <siteMapNode url="ContentFiles/contact.aspx" title="Liên hệ" />
        <siteMapNode url="Management/ManageEmail.aspx" title="Admin" />
    </siteMapNode>
</siteMap>
```

Để thêm một trang vào trình đơn của website, bạn phải chép và dán (với các hiệu chỉnh cần thiết) một mục của file XML trên. Theo cách này, trang master (chứa tham chiếu đến điều kiểm navigation) cho bạn thấy được trình đơn của website trên mỗi trang.

Phần tiếp theo giải thích chi tiết cách cài đặt và cấu hình các file nguồn của website và cách triển khai website trên một server.

2.4

Cài đặt File Share

Đã đến lúc tìm hiểu cách cài đặt File Share. Bạn có thể cài đặt File Share theo hai kiểu: cài đặt trên máy server (File Share là một ứng dụng thực thụ) hay cài đặt để phát triển cục bộ (hiệu chỉnh mã nguồn File Share trong Visual Studio 2005 hay Visual Web Developer).

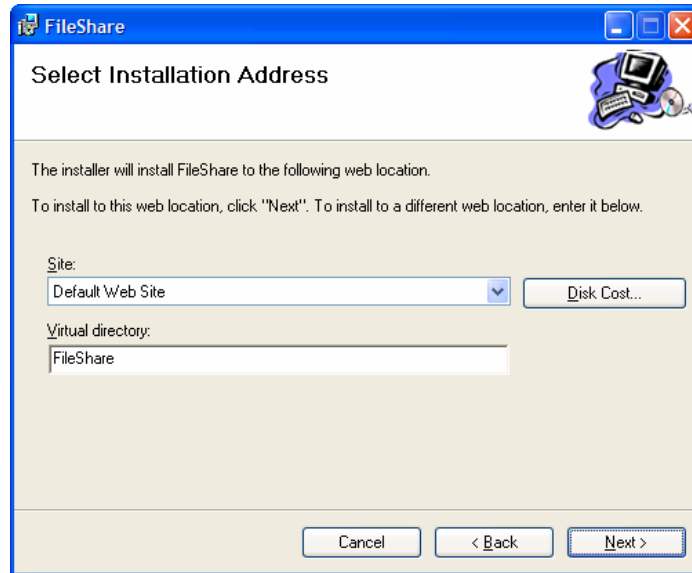
Bởi ứng dụng có liên quan đến việc gửi e-mail nên cấu hình một tài khoản e-mail là vấn đề cấp thiết cho việc thực thi dự án. Ắ nhiên nhà cung cấp mail không cấp SMTP relay cho các tài khoản người dùng, và cần phải cẩn thận khi sử dụng các giao thức bảo mật mail server. Ứng dụng mẫu này giả định bạn có thể truy xuất đến một mail server với các quyền relay được cấp cho một tài khoản người dùng hợp lệ.

Cài đặt File Share trên máy server

Phần này giả định .NET Framework 2.0 đã được cài đặt rồi. Ắ ếu bạn muốn cài đặt File Share như một website thực thụ trên một máy tính hay máy server, không có hiệu chỉnh hay mở rộng gì cả, thực hiện theo các bước sau:

Chạy file WebSetupProjects\FileShare\Debug\FileShare.msi từ CD-ROM đính kèm. Quá trình này cài đặt những file cần thiết vào thư mục C:\inetpub\wwwroot\FileShare\. Chú ý rằng, trong lúc cài đặt, có một màn hình yêu cầu bạn xác nhận tên thư mục ảo, bạn hãy giữ nguyên tên mặc định là FileShare (xem hình 2-12).

Ắ hắp Next để cài đặt ứng dụng, và đóng chương trình cài đặt khi đã hoàn tất.

**Hình 2-12**

Duyệt website theo địa chỉ <http://localhost/FileShare/>. Ứng dụng File Share sẽ xuất hiện. Để thử nghiệm phần quản trị, nhấp vào liên kết Admin và đăng nhập với tên người dùng là Admin và mật khẩu là password#.

Cài đặt File Share để phát triển cục bộ

Phần này giả định .NET Framework 2.0 đã được cài đặt, cùng với Visual Studio 2005 hay Visual Web Developer. Nếu bạn muốn mở dự án bằng Visual Studio 2005 hay Visual Web Developer, thực hiện theo các bước sau:

Chép thư mục Websites\FileShare\ từ CD-ROM đính kèm vào đĩa cứng, chẳng hạn C:\Websites\FileShare\.

Mở Visual Studio 2005 hay Visual Web Developer. Chọn File | Open Web Site và tìm đến thư mục C:\Websites\FileShare\. Khi đó, cửa sổ Solution Explorer chứa tất cả các file của dự án.

Trong phần appSettings của file Web.config, hiệu chỉnh các giá trị của mail server để có thể gửi e-mail. Thuộc tính PageTitle cũng được thay đổi tại đây để hiển thị trên thanh tiêu đề của mỗi trang:

```
<appSettings>
  <add key="EmailFrom" value="admin@FileShare.com"/>
  <add key="EmailSubject" value="File Ready for Download!"/>
  <add key="SmtpServer" value="127.0.0.1"/>
  <add key="MailUser" value="myalias"/>
  <add key="MailPassword" value="mypassword"/>
  <add key="MailPort" value="25"/>
  <add key="EmailFormatSelected" value="Text"/>
  <add key="PageTitle" value="File Share"/>
  <add key="ShareLocalFolderPath"
    value="C:\Inetpub\wwwroot\FileShare\FileStorage"/>
  <add key="httpDownloadPath" value="http://localhost/FileShare/ContentFiles"/>
  <!--
  <add key="CurrentTheme" value="CleanBlue"/>
-->
```

```
<add key="CurrentTheme" value="CleanRed"/>
</appSettings>
```

Ấn phím F5 để thực thi ứng dụng trong môi trường phát triển. Phần đáng quan tâm nhất trong quá trình này có lẽ là việc sử dụng và cấu hình tài khoản mail server trong file Web.config.

Nếu muốn cải tiến ứng dụng File Share, bạn hãy tham khảo phần tiếp theo.

2.5

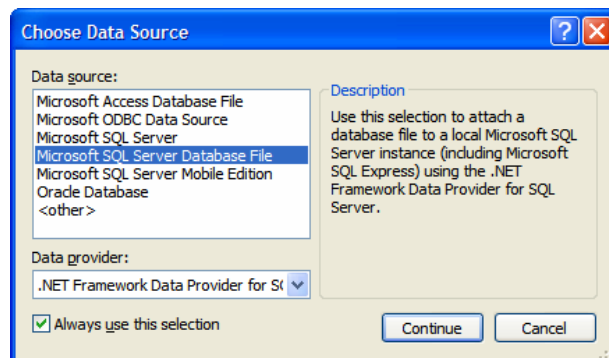
Mở rộng File Share

Mặc dù là một ứng dụng xuất phát rất hay cho việc học ASP.NET 2.0, nhưng File Share không bao hàm tất cả các tính năng mà bạn có thể muốn biết. Một số cải tiến cho File Share có thể bao gồm:

- ✓ **Khả năng quản lý**—Sẽ rất hay nếu có thể nhận biết những file nào hiện có trên server, chúng đã được truy xuất bao nhiêu lần,... Thậm chí khả năng xóa file nếu nó không được truy xuất hay đã quá cũ.
- ✓ **Lập báo cáo**—Sẽ rất hay nếu có một số báo cáo về các tài nguyên, những đáp ứng của chúng, và tỉ lệ phần trăm của mỗi tài nguyên.
- ✓ **E-mail phản hồi**—Người gửi file có thể muốn nhận một e-mail cho biết rằng người nhận đã download file đó.
- ✓ **Bộ đếm**—Khả năng theo dõi số lượt truy cập website, bao gồm số lần file được download.

Nếu muốn hiện thực tính năng gửi thêm e-mail, bạn hãy tuân theo các bước dưới đây:

Trong Visual Studio 2005 hay Visual Web Developer, mở website từ C:\Inetpub\wwwroot\FileShare\. Mở Server Explorer bằng cách chọn View | Server Explorer từ thanh trình đơn. Tìm cơ sở dữ liệu của bạn bên dưới nút Data Connections. Nếu cơ sở dữ liệu của bạn không được liệt kê, hãy nhấp phải vào nút Data Connections, chọn Add Connection, chọn Microsoft SQL Server Database File (xem hình 2-13), và chuyển đến file FileShareDB.mdf trong thư mục App_Data của website. Khi đó, kết nối của bạn sẽ được đưa vào cửa sổ Server Explorer. Giờ đây bạn đã có một cây cơ sở dữ liệu hợp lệ, mở rộng nó để thấy thư mục Stored Procedures. Nhấp phải vào thư mục này và chọn Add New Stored Procedure.



Hình 2-13

Thêm thủ tục tồn trữ mới như sau:

```
CREATE PROCEDURE dbo.sprocResourceSelectEmailInfo
/*
-- Lấy thông tin e-mail người gửi của một tài nguyên cụ thể từ cơ sở dữ liệu
*/
    (@id int)
AS

SELECT dbo.Contact.email
    FROM dbo.Contact INNER JOIN dbo.Resource
    ON dbo.Contact.id = dbo.Resource.fromContactID
    WHERE (dbo.Resource.id = @id)
```

Thủ tục tồn trữ mới này, `sprocResourceSelectEmailInfo`, sẽ cung cấp địa chỉ e-mail của người gửi đối với một file tài nguyên cụ thể.

Mở rộng thư mục `App_Code`, và rồi mở rộng thư mục `Dal` để thấy file `EmailContentDB.vb`. Trong lớp này, bạn phải thêm một hàm mới là `GetEmailSenderForResource`, hàm này sẽ thực thi thủ tục tồn trữ mà bạn vừa tạo. Hàm này nhận tham số nguyên `resourceID`, và trả về một chuỗi là địa chỉ e-mail của người gửi.

Mã lệnh của hàm này như sau:

```
''' <summary>
''' Lấy địa chỉ e-mail của người gửi file từ cơ sở dữ liệu
''' </summary>
Public Shared Function GetEmailSenderForResource( _
    ByVal resourceID As Integer) As String
    Dim mSender As String = ""

    Try
        Using mConnection As New SqlConnection(Config.ConnectionString)
            Dim mCommand As SqlCommand = New SqlCommand( _
                "sprocResourceSelectEmailInfo", mConnection)
            mCommand.CommandType = CommandType.StoredProcedure
            mCommand.Parameters.AddWithValue("@id", resourceID)
            mConnection.Open()
            Using mDataReader As SqlDataReader = _
                mCommand.ExecuteReader(CommandBehavior.CloseConnection)
                If mDataReader.Read() Then
                    mSender = mDataReader.GetString( _
                        mDataReader.GetOrdinal("email"))
                End If
            mDataReader.Close()
        End Using
    End Using

    Catch ex As Exception
        ' Bằng cách gọi lệnh "Throw", bạn dựng lên lỗi file Global.asax, và nó sẽ sử dụng
        ' trạng thái lỗi mặc định để xử lý/hiển thị lỗi tùy biến cho người dùng
        Throw
    End Try

    Return mSender
End Function
```

Trong thư mục `App_Code` và bên trong thư mục con `Bll`, mở file `EmailContent.vb`. Thêm một hàm mới vào lớp này, với tên là `GetEmailSenderForResource`. Hàm này nhận tham số nguyên `resourceID`, và trả về một chuỗi là địa chỉ e-mail của người gửi.

Dưới đây là nội dung của hàm GetEmailSenderForResource:

```
''' <summary>
''' Lấy địa chỉ e-mail của người gửi file từ cơ sở dữ liệu
''' </summary>
Public Shared Function GetEmailSenderForResource( _
    ByVal resourceID As Integer) As String
    Return EmailContentDB.GetEmailSenderForResource(resourceID)
End Function
```

Mở trang Download.aspx và nhấn F7 để mở file code-behind Download.aspx.vb. Bạn cũng có thể nhấp phải vào trang Download.aspx và chọn View Code. Trong trang này, thêm đoạn mã dưới đây vào khối Try-Catch của hàm DisplayDownloadDialog. Logic này sẽ thu lấy biến chuỗi truy vấn resourceID, sử dụng nó để truy vấn cơ sở dữ liệu và trích địa chỉ e-mail của người gửi để gửi một thông điệp cho người này.

```
Dim resourceID As Integer = Request.QueryString("resourceID")
Dim senderEmail As String = EmailContent.GetEmailSenderForResource(resourceID)
Dim emailBody As String = "File của bạn đã được download vào ngày " & _
    System.DateTime.Today.ToLongDateString
Utilities.SendEmail(senderEmail, Config.EmailFrom, _
    "Your file was downloaded", emailBody)
```

Giờ đây bạn đã hiện thực một cải tiến đơn giản, hiểu biết của bạn về kiến trúc ứng dụng chắc chắn tốt hơn rất nhiều. Bạn đã thấy và phát triển theo lối tiếp cận phân tầng thiết kế. Điều này cho phép bạn tiến nhanh khi thêm chức năng vào ứng dụng này, hoặc tự viết một ứng dụng cho riêng mình.

2.6

Kết chương

Chương này khảo sát một số điều kiểm mới của ASP.NET Framework 2.0 như: Login, LoginStatus, PasswordRecovery, SiteMap, SiteMapDataSource, Menu. Bạn đã tìm hiểu một bản thiết kế ứng dụng đơn giản; sử dụng các lớp thuộc tầng nghiệp vụ, các lớp thuộc tầng dữ liệu, và một cấu trúc cơ sở dữ liệu cơ bản. Bạn cũng biết được các điều kiểm mới này có thể giúp bạn tiết kiệm rất nhiều thời gian và công sức.

Phần thiết kế File Share cho bạn ý niệm về các lớp được sử dụng; cũng như các thuộc tính, phương thức, và mục đích chung của chúng. Phần này cũng giải thích các thủ tục tồn trữ, các thiết kế bảng, các mối quan hệ, và các thực thể dữ liệu được kết sinh.

Phần cuối chương trình bày cách trích xuất và tùy biến File Share trong môi trường phát triển, cũng như cách cài đặt nó.