

1

Introducing the Project: TheBeerHouse

This chapter introduces the project that we're going to develop in this book. I'll explain the concept behind the sample website that is the subject of this book, but as you read along you should keep in mind that this is a general-purpose, data-driven, content-based style of website that can easily be modified to meet the needs of a myriad of real-world website requirements. Although we'll use many of the older features of ASP.NET, the clear focus of this book is on showing you how to leverage the powerful new features of ASP.NET 3.5 SP1 such as ASP.NET AJAX, and the ADO.NET Entity Framework. I'll also integrate some basic search engine optimization techniques and third-party services for social networking and SPAM filtering to make a much richer site.

This book follows a "Problem-Design-Solution" approach in each chapter: the Problem section explains the business requirements for the module designed in that chapter; the Design section is used to develop the roadmap for meeting those requirements, and the Solution section is where we write the code to implement our design. This is unlike traditional computer books because the focus is not on teaching basic concepts but rather on showing you how to apply your knowledge to solve real-world business problems.

If you are new to ASP.NET, this is perhaps not the best book to start with, but if you're generally familiar with the basic concepts of web development and ASP.NET (any version of ASP.NET), you're ready to put that knowledge to use, and perhaps you want to learn about the new features in ASP.NET 3.5 SP1. Then, fasten your seat belt!

TheBeerHouse was originally created by Marco to serve as an online presence for a local pub in his hometown of Bologna, Italy. It was meant to serve as a way to reach out to the pub's mostly young clientele. This is still the primary goal of the site, but as technology matures, so does the average user's expectations of a site. The success of the site has driven website traffic and demand for TheBeerHouse merchandise much higher. This demand means a new site is needed to help extend the brand and the sharing of information with patrons. It also provides an opportunity to

Chapter 1: Introducing the Project: TheBeerHouse

upgrade the underlying plumbing with some new technologies for both the user interface and the business layers.

Problem

Although the owner has always used traditional, printed marketing ads for her pub, and has a popular website, she wants to expand into social networking because many of her patrons are already active on Facebook, Twitter, and other popular social networking sites. Retaining features with which users are familiar is important because users are used to reading about new specials and events, and receiving newsletters with promotions and coupons. Now you want to extend their capability to browse photos of past events and to see a true calendar of upcoming events. Since the typical patron of TheBeerHouse is a young student who is active online, these upgrades should go a long way in extending TheBeerHouse brand, and, hopefully, profits!

The owner also realizes that, just as online marketing and frontend experiences have evolved, so has the technology that drives the Internet. Building a flexible and scalable business tier is also important to the owner. Object Relational Mapping (ORM) frameworks have come of age since the last version of the site. She is also interested in building some client software for back-office management with Windows Presentation Foundation (WPF) and maybe even a mobile client in the next phase, so creating a common business library is very important.

Design

At the beginning of a project, you think about your client's needs and how you might meet those needs, and possibly even expand on them to give your client more functionality than the minimum needed, while still staying within your time limits and budgetary guidelines. In this scenario your client is a pub owner who wants to have a website to promote her pub, providing online information about upcoming events, reports about past events, and more. The current site can be expanded in many ways, to create a new site that has a lot more interesting things, good for its users (who are also potential customers for the physical pub) and for the store owner. We can begin by writing down a list of features that a modern content-based site should have, and a few reasons why they are useful:

- ❑ The site will need to provide a rich interactive user experience (now known as UX). The experience users have on the website directly affects the impression they have of the pub. This means that users must find information and desired activities easy to browse and interact with.

Attention should also be given to cross-browser compatibility, that is, ensuring that the site looks good and behaves well on different platforms and browsers. While Internet Explorer is still the dominant browser, FireFox, Opera, and Safari are growing in popularity. You can't know in advance which browser your customers will use, as you might in the case of an intranet site for a corporation, for example.

- ❑ A successful content-based site owes its popularity to its users. Loyal users who regularly visit the site, help write content, and participate in polls and special events are those who guarantee that the site will keep growing. To build a vibrant community of active members, users must have some sort of *identity*, something that describes and distinguishes them from other members. Because of this, the site needs a registration feature, as part of a larger authentication/authorization infrastructure. This will also be used to grant and restrict access to some areas of the site.

- ❑ Extending the concept of identity and interaction is leveraging popular social networking sites like Facebook, MySpace, Twitter, and YouTube. These sites will give TheBeerHouse another avenue of extending its brand, interacting with patrons, and allowing them to interact with TheBeerHouse site. Most social networking sites, such as Twitter, have available APIs that can be used by the site to publish content. Twitter allows you to post and monitor your Twitter feed through any client you want. All of these external sites have been designed to add value to the overall experience for a patron of TheBeerHouse.
- ❑ The site needs a constant supply of fresh content to stay alive and vibrant. If the content becomes stale, visitors will lose interest in the site and won't visit it anymore. A pub's site can't be very good unless it has regular updates about upcoming events, parties, and concerts. What's the point in visiting the site if it doesn't display photos that were shot at the last party? To facilitate a constant stream of new content, the site needs some mechanism that enables the editor to easily update it with dynamic content. The content should also be thoughtfully organized so that visitors can find, and even subscribe to, the type(s) of information in which they are most interested. Furthermore, the editor who will be in charge of the content updates will probably not be a technical person, so you must build some simple administration pages that make updates easy, even for nontechnical people.
- ❑ The Beer House owes much of its success to its nightly entertainment, themed parties and other events. The new site includes a Calendar of Events for visitors to keep track of what is happening at the Beer House.
- ❑ Once the site has new content ready to be read, the site's manager must have some way to inform its users about this. Not all users visit the site every day, so the site manager must be proactive and notify the customers about recent updates. If customers have registered on the site, providing their e-mail address, they might also have requested to receive a newsletter notifying them about recent changes and additions to the site. Of course, there are also other ways to syndicate news, such as exposing RSS (Really Simple Syndication) feeds to which a user can register and then control from their favorite RSS reader, and get automatic notifications about news without having to visit the site daily to get the information.
- ❑ A site like this can also be a good opportunity to get feedback from customers about a variety of issues: What do they like most in a pub? What brand of beer do they prefer? Do they want to listen to live music while drinking with friends, or perhaps they don't like all that noise? Establishing some kind of user-to-site communication is important, and if you get a good number of responses, it can even lead to strategic decisions and changes that may improve the business.
- ❑ If the presence of some sort of user-to-site communication is important, user-to-user communication may be even more so, because that's the central point of creating a community of loyal users, who come to the site frequently to chat, discuss the news posted on the site, ask for suggestions from the others about upcoming events, and more. This translates into more traffic on the site and a feeling of membership that will pay off in both the short and long run.
- ❑ Once the store has a user base, the store's owner may decide to expand it so that it supports an online store. In fact, the pub already offers a catalog of products for beer enthusiasts, such as glasses, T-shirts, key chains, and more. If the site has a lot of traffic, it may be a good way to promote these products so that people can place orders without even visiting the pub in person. And once users see a product and like it, they can rate that product to tell other people how much they like it. The online store must be easy to manage by nontechnical people, because it might possibly be the pub's owner who adds and edits products, and manages the orders. Thus, there must be a module with a simple and intuitive UI that automates as many operations as possible, and guides the user through the tasks.

Chapter 1: Introducing the Project: TheBeerHouse

- ❑ Demonstrating how lively and fun TheBeerHouse is very important in fostering dedicated patrons. Photos and videos are a great way to share this atmosphere. Adding a photo gallery to the site and taking advantage of YouTube.com and other video-sharing sites are great ways to share what life is like in TheBeerHouse.
- ❑ With the site offering news and articles, lists of products, user-to-user discussions, and other dynamic content, it's easy to imagine that the home page could easily become crowded, and possibly more difficult to read and understand because of too much information. It would be good if the user herself could build her own home page, according to what she is interested in. Maybe she wants to read about upcoming events but doesn't care about shopping online for gadgets? Great, you want to give her the capability to do that, by adding and deleting content to and from the home page, or maybe just moving around the existing content so that it's placed and organized in a way that she finds more comfortable and useful for her. This type of customization is done on some large sites such as Windows Live and My MSN, for example, and is a great example of personalization, which helps encourage users to decide to register on the site.
- ❑ As mentioned previously, the pub is typically visited by a lot of customers coming from many different countries, and the pub's owner expects the same to happen on the website. Because of this, the site must be partially or fully translated into multiple languages, making it easy for most users to understand it. Not only must text be translated, but information such as dates and numbers should also be displayed according to the user's preferred locale settings, so that nobody will misunderstand an announcement about an upcoming party or event.
- ❑ Optimizing the site for search engine exposure is also a high priority. Adding social networking features to the online strategy should help gain valuable inbound links to assist with search engine ranking. But you still need to make sure that common search engine optimization techniques are applied to the site to help with overall ranking and placement of the site for targeted keyword phrases.

To recap everything in a few words, the TheBeerHouse site will have everything a modern content-based site will have, including dynamic articles and news, polls for user-to-site communication, forums for user-to-user communication, newsletters and RSS feeds to notify members about new content on the site, an e-commerce store for selling products online, home page personalization, and content localization. It will also have a photo gallery, calendar of events, social networking, and OpenId integration. Although the sample project is built around a fictitious pub, you'll recognize in this list of requirements the common features of the majority of content- and commerce-based sites you find online now, and sites that you're likely to develop in the near future, or maybe even sites you're developing right now.

Solution

The Solution section of each chapter will contain the instructions and actual code for implementing all the features and requirements outlined and designed in the previous sections. However, this first chapter gives you a more detailed description of exactly what the following chapters will cover, so that you can get a good idea of what the final result will be like.

In Chapter 2, you'll build the site's design, the graphics, and the layout that are shared among all pages of the site, through the use of master pages and nested master pages. You will also use themes and Cascading Style Sheets (CSS) to create a couple of different visual appearances for the same master

page, and create a mechanism to enable users to select their own favorite theme from a drop-down list, so that they can change the colors and overall appearance of the site according to their taste and possible visual impediments. Finally, a flexible and easy-to-maintain navigation system will be built by means of a custom `SiteMapProvider` and the `Menu` and `SiteMapPath` controls.

In Chapter 3, you'll lay down the foundations for building a flexible, and easily configurable and instrumented site. First, a data access layer (DAL) will be built from the ADO.NET Entity Framework and a SQL Server database. Then a business logic layer will be built on the top of the DAL to expose the data in an object-oriented way, with the required validation logic, transaction management, event logging, and caching. Finally, you'll look at the UI and presentation layer, which take advantage of the new `ListView` control and ASP.NET AJAX to quickly generate complex, interactive, and feature-rich, data-driven pages.

In Chapter 4, you'll integrate the membership infrastructure introduced in ASP.NET 2.0 into the site, to create user registration forms and supporting logic to authenticate/authorize users. You'll also see how to use the `Profile` module, which allows you to declaratively define user-level properties that are automatically persisted to a durable medium, quite different from the well-known traditional `Session` state variables, which last only as long as the user browses the site on one occasion. You will also see how to take advantage of ASP.NET AJAX's built-in membership, role, and profile interfaces. You will build a complete management console to enable administrators to see the list of members, disable members that behave badly on the site, and view and edit each user's profile.

In Chapter 5, you'll build a sort of content management system, a module that enables administrators to completely manage the site's articles from an intuitive UI, accessible also by nontechnical users. The module will integrate with the built-in membership system to secure the module and track the authors of the articles, and will have a syndication service that publishes an RSS feed of recent content for a specific category, or for every category, and will support ratings and comments, among many other features. It will also automatically post matching content on Twitter, alerting followers about new content and make use of user and search-engine-friendly URLs, that take advantage of the article title in the URL and a custom URL Rewriting `HttpModule`. The result will be quite powerful, enabling the editor to prepare richly formatted content in advance, and schedule it for automatic publication and retirement, so that the site's content updates are as simple as possible, and require the least effort and time. At the end of the chapter, you will have experienced a many of the things you can do with the new `ListView` control, the ADO.NET Entity Framework, interacting with external APIs and custom `HttpHandlers`.

In Chapter 6, you'll implement a solution for creating and managing multiple dynamic polls on the website. It will feature an administration console for managing the polls through a web browser and a user control that enables you to plug different polls into any page you want with just a couple of lines of code, as well as a history page for viewing archived polls.

In Chapter 7, the site will be enriched with a complete module for sending out newsletters to members who registered for them in their profile page. The module will enable you to send out the e-mail newsletters from a background thread, instead of the main thread that processes the page request, so that the page won't risk timeouts, and more important, so that the editor will not be left with a blank page for minutes at a time. AJAX will be used to implement real-time feedback about the newsletter being sent in the background. Finally, end users will be able to look at past newsletters listed on an archive page. To implement all this, you'll use advanced features such as multithreaded programming, the new script callback feature, and new classes for sending e-mails.

Chapter 1: Introducing the Project: TheBeerHouse

In Chapter 8, you'll create a forums system from scratch, which supports multiple sub-forums with optional moderation; lists threads and replies through custom pagination and with different sorting options; has wide support for standard RSS feeds, configurable user rating, signatures, and quoting; and offers other features typical of most recent forum software. Complete administration features (for deleting, editing, approving, moving, and closing threads and posts) will also be provided.

In Chapter 9, you'll add a working e-commerce store with most of the essential features, including a complete catalog and order management system; a persistent shopping cart; integrated online payment via credit cards; product ratings; product stock availability; rich formatting of a product's descriptions, including text and images; configurable shipping methods and order statuses; and much more. All this will be implemented in relatively few pages, since it will leverage the good foundations built in previous chapters, such as the built-in membership and profile systems, and other features and controls, such as the `ListView`, `Wizard`, and `MultiView` controls.

In Chapter 10, you'll add a photo gallery that features the capability for administrators to add and manage photo albums, photos, and descriptive data about the photos. It also implements an AJAX lightbox to add some special effects when viewing an image.

In Chapter 11, you'll add an event calendar to which TheBeerHouse staff will add events at each of their locations. Patrons can subscribe to these events via RSS or add them to their calendar applications via an `iCal` object. This will be done by using the `Calendar` control and a custom `httpHandler`. New events will also be published directly to TheBeerHouse Twitter feed.

In Chapter 12, you'll make the site's home page fully localizable to an additional language and will support the user's preferred locale settings when displaying dates and numbers. All this can be done easily with ASP.NET, thanks to its automatic resource generation, implicit and explicit localization expressions, strongly typed and dynamically compiled global resources, and good Visual Studio Designer support.

Finally, in Chapter 13, you'll look the different ways to deploy an ASP.NET site, either on a local IIS server or to a remote production site, or to an inexpensive shared hosting server. The ASP.NET compilation model enables you to use a simple `XCOPY` deployment that includes everything, but lacks protection of source code, and takes a little time to compile on first requests. If that's a problem for you, you will see how you can use the new command-line tools and Visual Studio's wizards to precompile the site and generate one or more compiled assemblies to deploy. You'll also learn how to deploy the local SQL Server Express database to a remote full-featured SQL Server instance, and how you can create installer packages for distributing the application to automate as many installation tasks as possible.

Summary

You now have an overview of an aggressive plan to develop a highly functional content-based website that shows you how to use ASP.NET 3.5, ASP.NET AJAX, the ADO.NET Entity Framework, and various external APIs to their full capacity. This chapter gave you a broad idea about what we're going to discuss, design, and implement throughout the rest of the book.

In each chapter, you'll learn something new about ASP.NET and web programming, and at the end of the book you will have created a real-world site with most of the features required by modern content-centric sites and e-commerce stores. Furthermore, the site you develop in this book may provide a good deal more functionality than any site you've designed in the past, and the relatively small development effort will enable you to do more than you thought possible in a small amount of time. One of Microsoft's key goals with the .NET platform is to help you "fall into the pit of success." ASP.NET is designed to make developers' jobs easier: to reduce the amount of effort required to implement common functionality, thereby giving them more time to focus on business needs, and enabling them to offer more advanced functionality to empower users and site administrators, while keeping the site maintainable and scalable. This book will help you judge whether Microsoft has met this goal. Let the adventure begin!

