

The Florida State University  
College of Arts and Sciences

**Real Estate Listing Web Application**  
**Using Microsoft .NET Framework SDK1.1 and SQL2000**

By Hang Wu

April 2004

A project submitted to the  
Department of Computer Science  
In partial fulfillment of the requirement for the  
Degree of Master of Science

Major Professor: **Dr. Daniel G. Schwartz**

# TABLE OF CONTENTS

<b>Acknowledgements</b> .....	3
<b>Abstract</b> .....	4
<b>1. Introduction</b> .....	5
1.1 Project Requirements Specifications .....	5
1.2 Project Design Specifications .....	6
1.3 Developer Environment .....	8
1.4 Technology Applied in this project.....	10
<b>2. Architecture Design</b> .....	12
2.1 Building Web Application in .NET .....	12
2.2 Flow Chart .....	13
<b>3. Database Design</b> .....	15
3.1 SQL 2000 and Relational Database .....	15
3.2 Database Table and Database Diagram .....	16
3.3 Sample of Database Scripts .....	17
<b>4. Implementation Detail</b> .....	19
4.1 Working with Web Objects .....	19
4.2 Creating Web Surfaces .....	19
4.3 DataGrid Control .....	21
4.4 Building Customer User Control .....	22
4.5 Maintaining State Information, Session, Cookie .....	23
4.6 Storing and Retrieving Data using ADO.NET .....	24
4.7 Maintaining Security.....	25
<b>5. Real Estate Web Application Running on the Sever</b> .....	26
5.1 Build and Run on the server.....	26
5.2 Test and Run the Project.....	26
<b>6. Future Extensions to the Project</b> .....	37
6.1 Using Object-Oriented Domain Model.....	37
6.2 Providing Web Services.....	38
6.3 Functionality Extensions.....	38
<b>7. Conclusion</b> .....	40
<b>8. User Manual</b> .....	41
<b>9. References</b> .....	43

## **Acknowledgements**

I would like to express my sincere gratitude and appreciation to all those who gave me help to complete this project report. I want to thank the Department of Computer Science for giving me the resources to do the project and using departmental data.

I am deeply indebted to my supervisor Prof. Dr. D. Schwartz from Florida State University whose help, stimulating suggestions and encouragement helped me in all the time of doing the project and writing the report.

Especially, I would like to give me special thanks to my parents whose love enabled me to complete this work.

## **Abstract**

The aim of this project was to build a real estate web application using Microsoft ASP .NET and SQL 2000. The programming environment was Visual Studio .NET 2003 using the C# programming language. The real estate system provides functionality for buyers, allowing them to search for houses by features or address. It provides functionality for the seller, allowing them to log into the system and add new advertisements or delete existing ones. For this purpose, each user is provided a login account with login ID and password access. This web application makes use of .NET features, including web controls, html controls, customer web controls, forms authentication, ADO .NET, error handling. The programming tasks included building the customer web controls and deploying these on the IIS web server. The system architecture is based on the three-tier enterprise architecture, comprised of a data layer, business logic, and a presentation layer.

# 1. Introduction

The aim of this project was to develop a prototype real estate listing service using Microsoft technology. The objective was to explore the capabilities of this technology as compared to the corresponding Java-based technology provided by Sun Microsystems. A similar real estate listing service using the latter approach, and created using the IBM Websphere development environment was undertaken in a previous project by Ms. Susmitha Athota [1].

There are some important issues in developing the real estate web application. First, the search time should be kept to a minimum. This depends on choosing the appropriate development environment and using good development techniques. Second, the web application should provide the services that both buyer and seller want. Third, the web application should have a friendly user interface.

This project is developed based on the ASP.NET using C# and the SQL 2000 database engine. ASP.NET is part of the Microsoft .NET framework, which is an integrated and managed environment for the development and execution of native code. ASP.NET is a platform for creating web applications that run on Windows servers using IIS and the .NET framework.

## 1.1 Project Requirements Specifications

This Real Estate Web application provides functionalities for both seller and buyer. The buyers can browse the web site without any security and they can

search for houses based on state/city/price/features. Also they can search the houses based on state/city/address. For the seller, a security issue is added. The seller needs to create an account in the system and every time he logs in, the previous advertisements posted by him are presented. He can then delete any of these, and he can add new ones.

The functionalities are as follows:

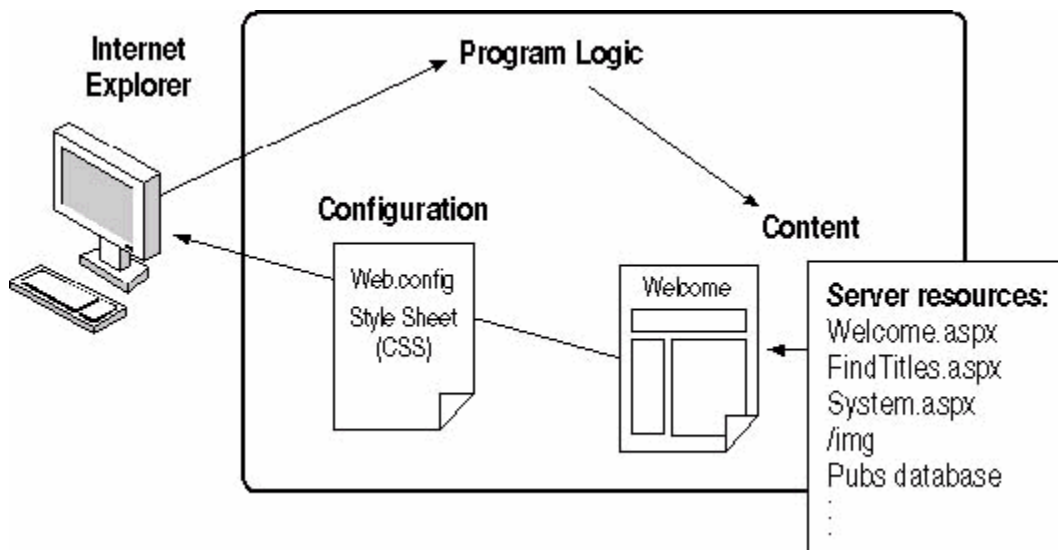
1. The buyer can search for houses according to detailed information: state/city/price/property/bedroom/bathroom/additional features.
2. The buyer can search for houses by state/city/street name.
3. The results are displayed five per page and the buyer can view the next or previous page.
4. The results can be sorted and viewed by selecting the sorting item and order.
5. When a buyer clicks the link for each listed item, he will be redirected to a page that lists the detail information for this house.
6. A seller can create an account in the system, and each seller has a username and a password.
7. The seller can log into the system, and his existing advertisements are displayed.
8. The seller can delete existing advertisements from the database.
9. The seller can add new advertisements and provide the detailed information and photo.

## 1.2 Project Design Specifications

The Real Estate Web application consists of three parts: content, program logic, and Web configuration information. The content of the project is presented through the web forms which use web controls and HTML components like conventional HTML page. The program logic, the project's executable is stored in a .dll file called an assembly. The assembly is compiled to intermediate code, and common language runtime does the final compilation before running the project. The back end of the project uses SQL2000 which is a relational database management system. ADO.NET is used to create a tight tier between the data tier and business logic.

The whole process of running this web application would be:

When client requests one of the web forms pages from the browser, IIS starts the project's executable. The .dll composes a response to the client based on content of the web form page and the program logic which generates the dynamic content. IIS gives back the HTML response to the client. Figure 1-1 shows the sequence.



## Figure 1-1

### 1.3 Developer Environment

The project is hosted by windows server by IIS (Internet Information Services) and .NET framework together with Microsoft SQL2000. It is developed in the Visual Studio. NET 2003 integrated development environment.

Internet Information Services: It is a very powerful web service which provides a highly reliable, manageable, and scalable web application infrastructure for windows server. It organizes web sites and hosts web applications together with system administration.

Microsoft .NET Framework: It is the infrastructure for developing distributed applications for the internet. It is made of two parts, an execution engine called the common language runtime(CLR), a class library which includes the whole set of ASP.NET classes as well as windows programming classes, other programming tasks such as file access, data type conversion, array and string manipulation and so on. Figure 1-2 shows how .NET application works.



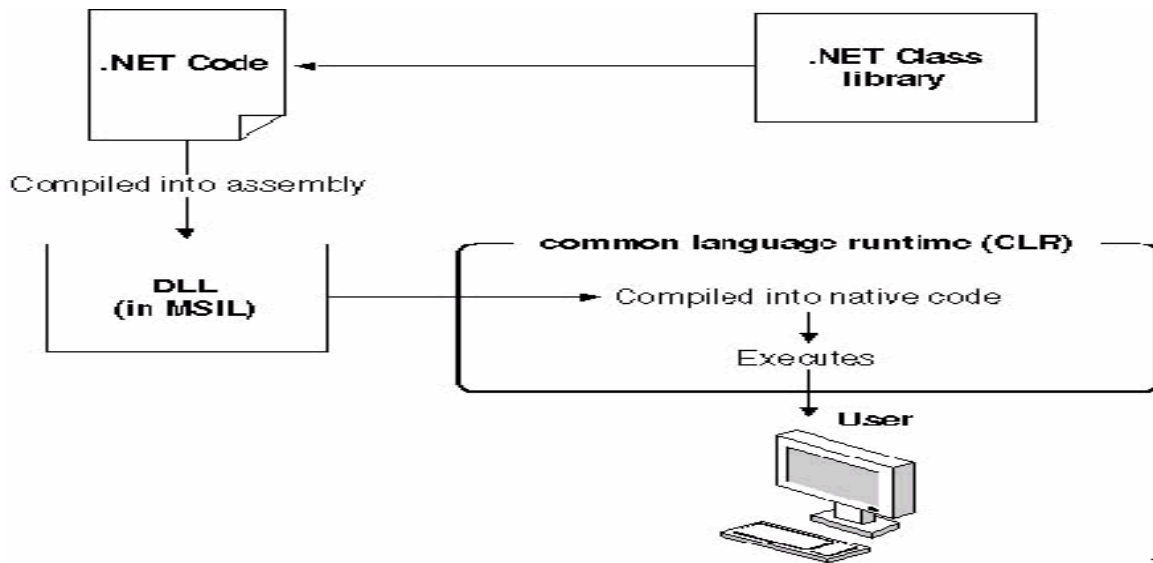


Figure 1-2

ASP.NET: It is the next generation ASP. It is part of the .NET framework and it supports two kinds of applications: web application and web services. It is a very powerful tool for making dynamic and interactive web page.

ADO.NET: .NET provides databases access through the set of tools and namespaces collectively referred to as Microsoft ADO.NET. There are three layers: the physical data store which could be a SQL database or an XML file, the Data Provider which interacts between the program and the database, the DataSet which stores disconnected data on the local memory. Figure 1-3 shows the ADO.NET model in action:

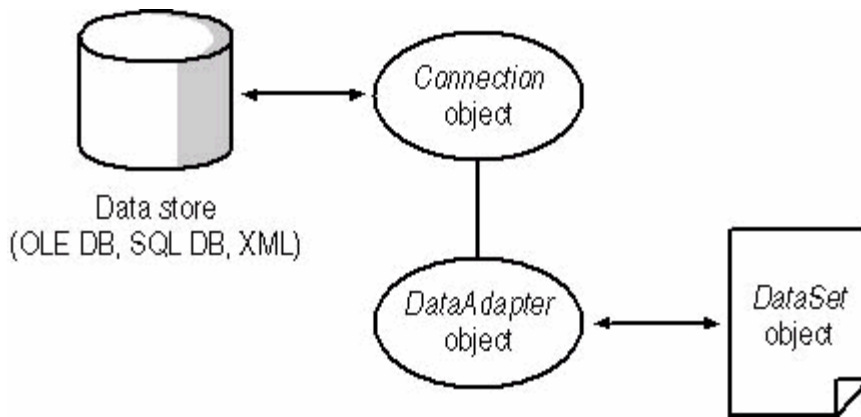


Figure 1-3

SQL2000: It is an RDBMS which uses Transact-SQL to send requests between a client computer and a SQL Server 2000 computer. RDBMS manages and stores the data by organizing the data into rows and columns within the database. It consists of the database engine, various applications, tools and databases.

Visual Studio .NET: The Visual Studio .NET provides a very friendly programming environment. It enables developers to quickly create data-driven and distributed applications using familiar Visual Basic techniques and reusable controls.

#### 1.4 Technology Applied in this project

There are some specific technologies that I have used in this project. It improves the performance of web applications.

Using Session as data container: Instead of getting the house record data from the database every time, the project fills the data into the session. It

cuts time down by limiting the transmitting time from the server to the back end SQL2000 sever.

Using Datagrid web control for displaying data: Datagrid is .NET provided web control which has powerful functionality in displaying data. Instead of using complicated presentation logic such as dynamic generating html code or xslt files, the project writes code using Datagrid which gives optimal running performance.

Using Forms Authentication:

This project uses forms authentication to control access to web pages by identifying users. It stores user name and password into the database. Each buyer will have a cookie associated if they have been authenticated. Also the password is hashed and it provides high security.

## 2. Architecture Design

### 2.1 Building Web Application in .NET

Web applications provide content from a server to client machines over the internet and the users view the web applications through a web browser.

This project uses client/server architecture. It is hosted on the web server and responds from other clients, as shown in Figure 2-1:

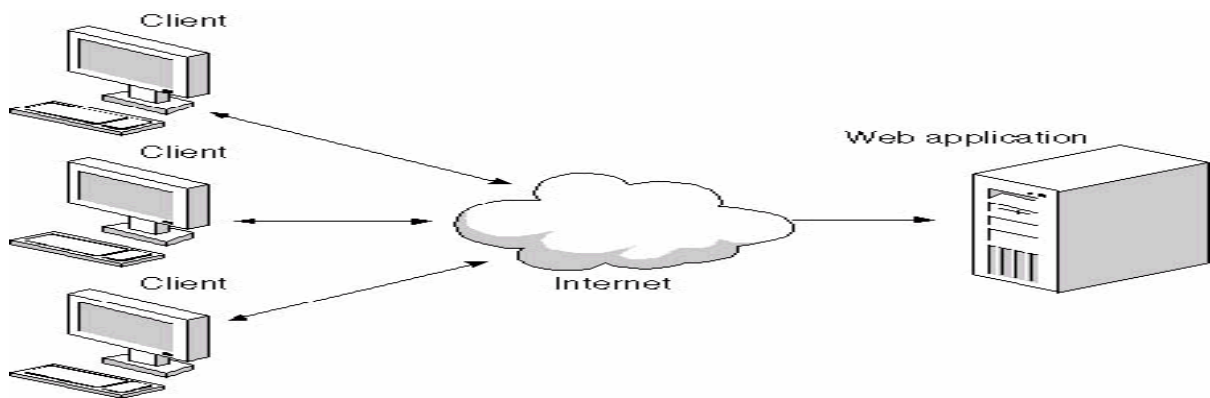


Figure 2-1

The real estate web application runs under Microsoft Internet Information Services. IIS manages the project, passes requests from the clients and returns the response based on executable code, web forms, html page, image files and other content. These requests and responds as passed over internet using HTTP, as show in Figure 2-2

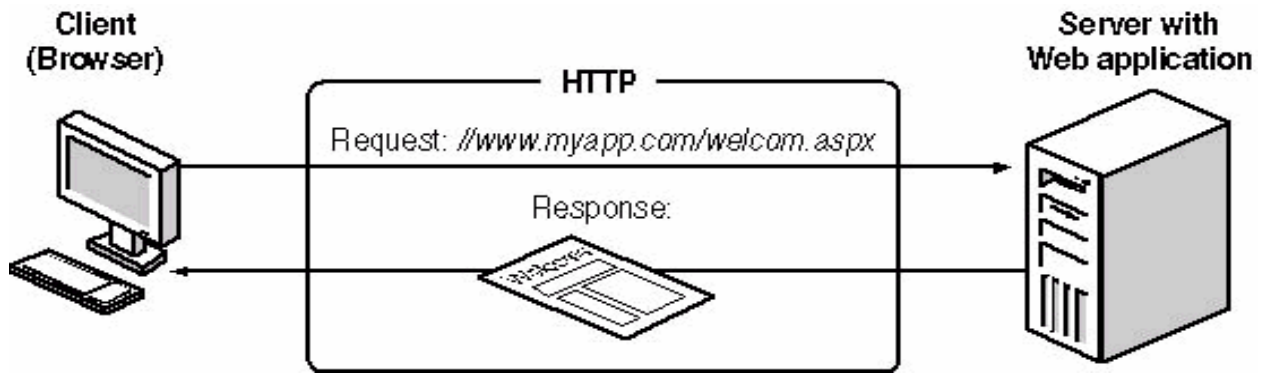


Figure 2-2

## 2.2 Flow Chart

The flow chart is a visual representation of the sequence of the content of this project and it shows all the steps of the project. It describes how these web forms are organized and invoked in order. The index page of this web site is FrontPage.aspx. From this page, users can have three options: Buyer searches the houses based on the state/city/features, buyers searches the houses based on the address name, seller logs into the system. Figure 2-3 shows the sequence order of this project:

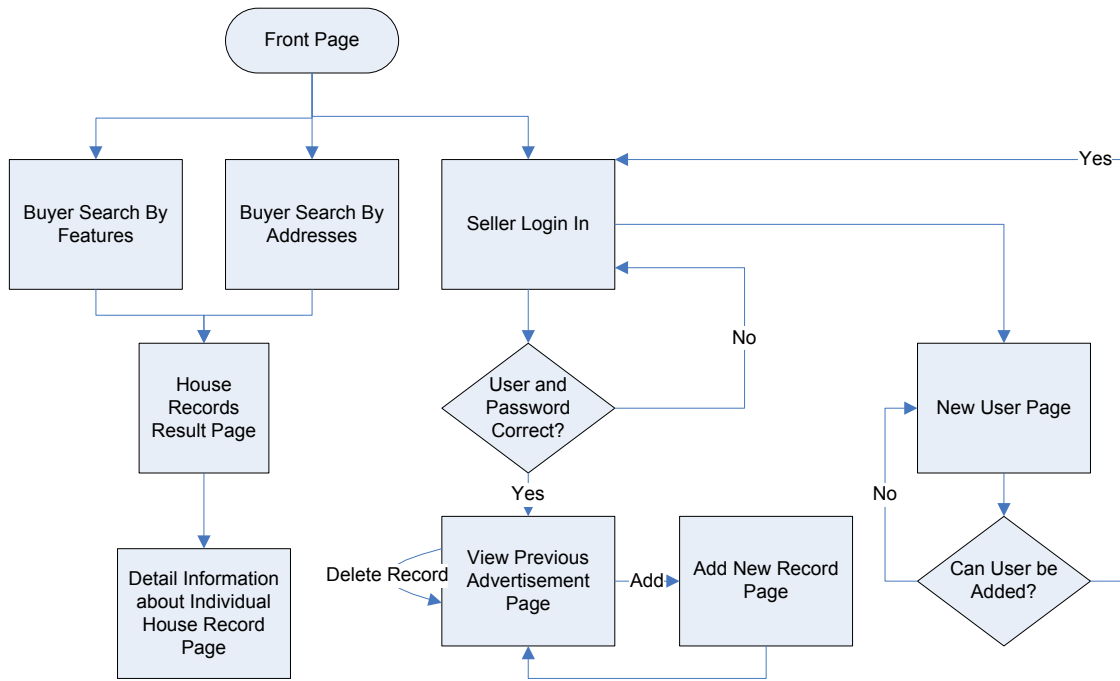


Figure 2-3

### **3. Database Design**

A good database design will contribute to software development and improve performance. It is always very important to keep in mind the data retrieval, storage and modification efficiency in mind. Good data modeling will provide easier and more intuitive access than others. When this project is developed, I have gone through the following step:

1. First, I analyzed the requirement of the real estate web application. Since it provides services for both seller and buyer, three tables are needed. House table contains individual house item. Seller table contains seller login and password information. Besides, a relationship table is also needed.
2. Second, I spent time modeling the data, using the standard method ER Diagramming, adding fields into each table.
3. Third, I tried out the ideas, tested them through limited trials and made necessary changes.

#### 3.1 SQL 2000 and Relational Database

SQL Server 2000 has a good connection with Microsoft .NET platform and provides two fundamental services to .NET platform. The first service is the SQL Server service, which is a high performance, scalable relational database engine. The second service is SQL Server 2000 Analysis Service. SQL provide a relational database management system (RDBMS). A Database is similar a data file in that it is a storage place for data. The developer runs the application that access data from SQL Server and presents it to the user in a friendly format. A relational database stores all its

data into tables, and nothing more. It uses mathematical set theory and normalization process to effectively organize data.

### 3.2 Database Table and Database Diagram

Based on the project description and requirement, three tables are designed.

**HouseRecord Table:** It contains the detailed information of each house records, including state/city/price and so on. The primary key is houseId which uses unique identifier type.

**Users Table:** It contains each seller username and password. The forms authentication is based on this table. User can create account and store information is Users table. The password is stored the hashed value of the original string. The primary key is UserName.

**UserHouse Table:** It is the relationship table between HouseRecord and Users Table. It has two foreign keys, UserName and HouseId.

Figure 3-1 shows the database diagram.



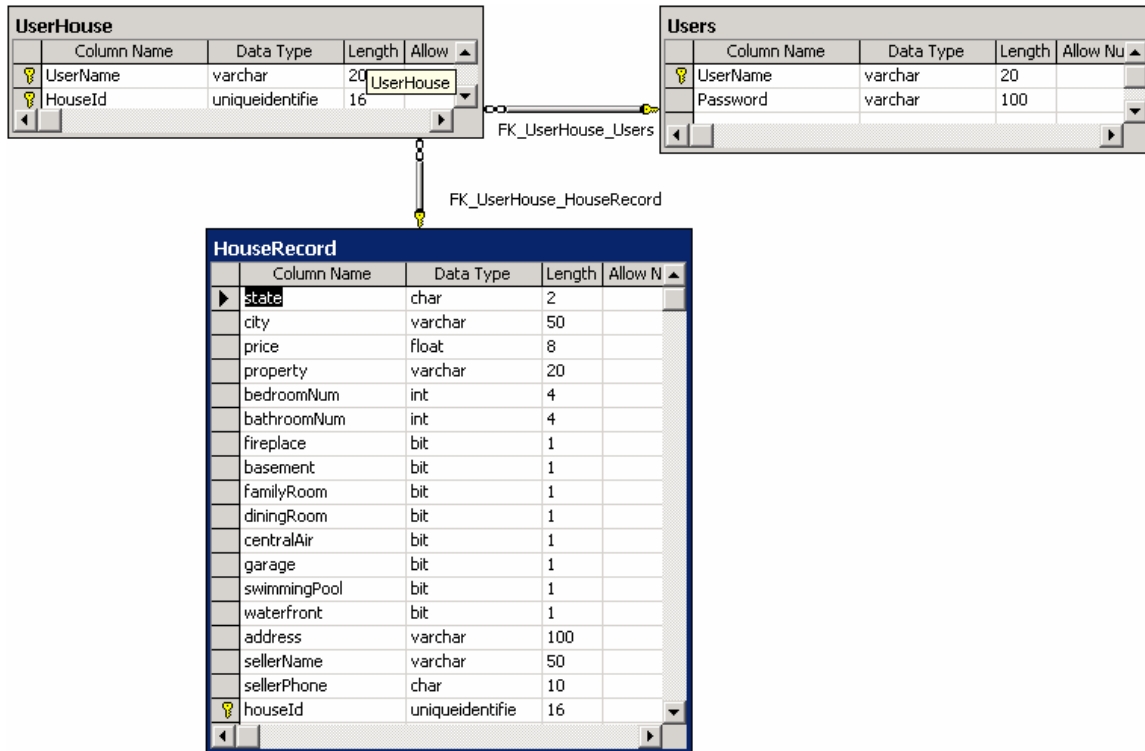


Figure 3-1

### 3.3 Sample of Database Scripts

I give some sample of the database scripts that are used to create the database and tables. Please refer to realestate.sql for detailed information.

Here is a piece of creating HouseRecord Table which stores the information of each house record.

```
CREATE TABLE [dbo].[HouseRecord] (
    [state] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [city] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL ,
    [price] [float] NOT NULL ,
    [property] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [bedroomNum] [int] NOT NULL ,
    [bathroomNum] [int] NOT NULL ,
    [fireplace] [bit] NOT NULL ,
    [basement] [bit] NOT NULL ,
    [familyRoom] [bit] NOT NULL ,
    [diningRoom] [bit] NOT NULL ,
    [centralAir] [bit] NOT NULL ,
```

```
    [garage] [bit] NOT NULL ,
    [swimmingPool] [bit] NOT NULL ,
    [waterfront] [bit] NOT NULL ,
    [address] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [sellerName] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [sellerPhone] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [houseId] [uniqueidentifier] NOT NULL
) ON [PRIMARY]
GO
```

## 4. Implementation Detail

This chapter describes in detail the .NET technologies used in this project. .NET Framework has an advantage in developing web application because ASP.NET uses a powerful set of web and HTML control, complete version of library classes, strong connection with ADO.Net, integrated security with IIS.

### 4.1 Working with Web Objects

Visual C# has a full set of object-oriented programming concept including classes, abstract classes, interfaces and overloading or overriding members. The system library classes are organized in namespaces. We can take a look at the statements in this project to allow developer to use class library:

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Configuration;
```

The project code itself has its own namespace also, the two namespaces in this project are: `namespace RealEstate.Seller`, `namespace RealEstate`.

### 4.2 Creating Web Surfaces

This project uses ASP.NET controls to create user interfaces for multi-page web forms. There are two styles of form layout: Grid layout and Flow layout. The web controls that have been used in this project are: Label, TextBox, Button, DropDownList, DataGrid (which will be described in detail), CheckBox, RequiredFieldValidator, CompareValidator.

We can take some examples how these web controls are used.

**Button Control:** In the html view of .aspx page, the following code is needed.

```
<asp:button id="subBtn" runat="server" Text="Submit"
BackColor="White"></asp:button>
```

In the code-behind of .cs file: an event-handler is associated with the button control, when the button is clicked, the code in the block will be executed.

```
private void creBtn_Click(object sender, System.EventArgs e)
{
    Response.Redirect("CreateNew.aspx");
}
```

**DropDownList Control:** In the html view of .aspx page, the following code is needed.

```
<asp:DropDownList id="SortList" runat="server" AutoPostBack="True">
<asp:ListItem Value="Price ASC" Selected="True">Price</asp:ListItem>
<asp:ListItem Value="Price DESC">Price Desc</asp:ListItem>
</asp:DropDownList>
```

In the code-behind of .cs file: an even-handler is associated with the DropDownList control, when the selected item is changed, the code in the block will be executed.

```
private void SortList_SelectedIndexChanged(object sender,
System.EventArgs e)
{
    DataView dv = (DataView)Session["records"];
    dv.Sort = SortList.Selected.Value;
    Session["records"] = dv;
    ResultGrid.DataSource = dv;
    ResultGrid.DataBind();
}
```

### 4.3 DataGrid Control

The DataGrid control is very important in this project. It is used in the result page to display the house records. It is a very powerful control in ASP.NET and it provided rich functions to manipulate displaying data. It has a property design dialog which can be used to set the display features in design mode, also developer can write some code to provide function in DataGrid, such as sorting, paging and so on. We can take a look at the code.

The following code is in the html view of Result.aspx to declare the datagrid:

```
<asp:DataGrid id="ResultGrid" runat="server" BorderColor="#E7E7FF"
BorderStyle="None" BorderWidth="1px" BackColor="White" CellPadding="3"
GridLines="Horizontal" AutoGenerateColumns="False" align="center"
PageSize="5" AllowPaging=True OnPageIndexChanged=ResultGrid_Page>
<SelectedItemStyle Font-Bold="True" ForeColor="#F7F7F7"
BackColor="#738A9C"></SelectedItemStyle>
<AlternatingItemStyle BackColor="#F7F7F7"></AlternatingItemStyle>
<ItemStyle ForeColor="#4A3C8C" BackColor="#E7E7FF"></ItemStyle>
<HeaderStyle Font-Bold="True" ForeColor="#F7F7F7"
BackColor="#4A3C8C"></HeaderStyle>
<FooterStyle ForeColor="#4A3C8C" BackColor="#B5C7DE"></FooterStyle>
<Columns>
<asp:TemplateColumn HeaderText="Search Result">
<ItemTemplate>
<table width="600" height="150" border="0" cellpadding="2"
cellspacing="2">
.
.
</table>
</ItemTemplate>
</asp:TemplateColumn>
</Columns>
<PagerStyle HorizontalAlign="Right" ForeColor="#4A3C8C"
BackColor="#E7E7FF" Mode="NumericPages"></PagerStyle>
```

```
</asp:DataGrid>
```

The code behind in Result.aspx.cs to bind the DataView to the DataGrid:

```
ResultGrid.DataSource = Session["records"];  
ResultGrid.DataBind();
```

The code behind in Result.aspx.cs to sort the data and redisplay using

DataGrid:

```
DataView dv = (DataView)Session["records"];  
dv.Sort = SortList.SelectedValue;  
Session["records"] = dv;  
ResultGrid.DataSource = dv;  
ResultGrid.DataBind();
```

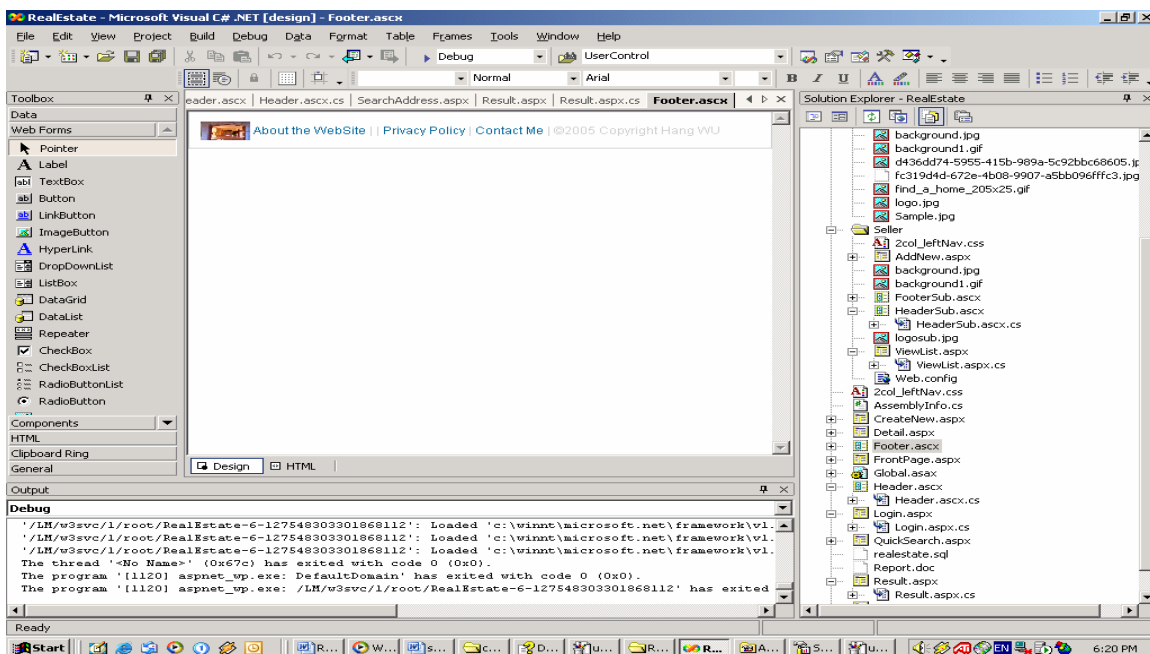
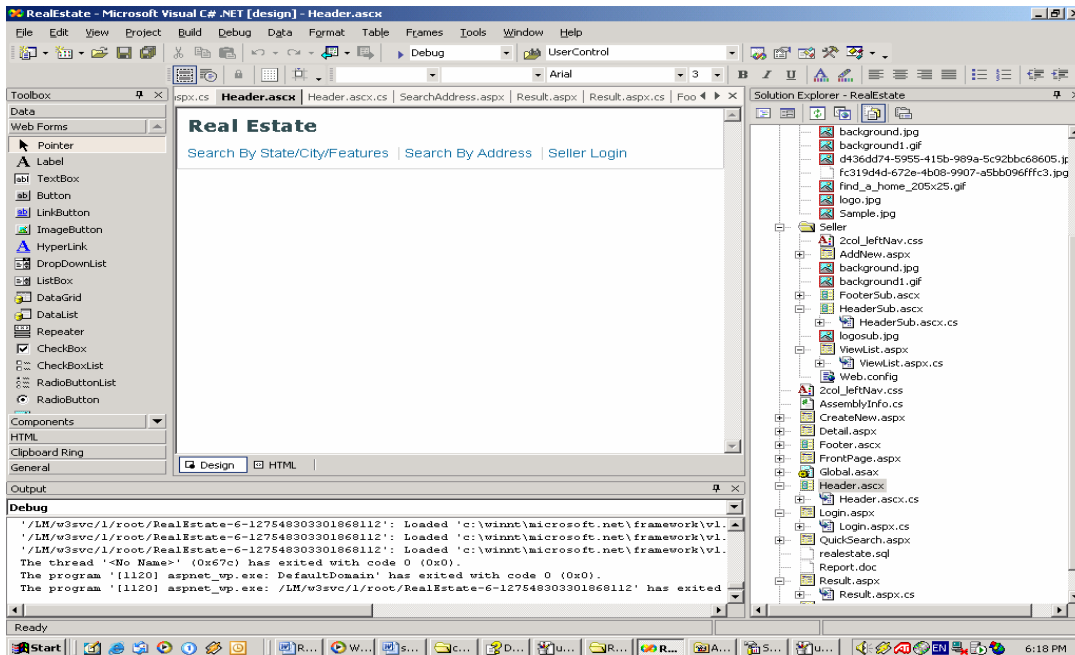
The code behind in Result.aspx.cs to page the DataGrid:

```
ResultGrid.CurrentPageIndex = e.NewPageIndex;  
ResultGrid.DataSource = (DataView)Session["records"];  
ResultGrid.DataBind();
```

#### 4.4 Building Customer User Control

In this project I developed two web user controls page: Header.ascx and Footer.ascx. They combine existing server and HTML controls to create functional units which could be reused in each page as header and footer.

The visual interface of the control is shown:



These two controls can be used on web forms throughout this project

We can take a look at the code in using header control here:

```
<%@ Register TagPrefix="uc1" TagName="Header" Src="Header.ascx" %>
<uc1:Header id="Header1" runat="server"></uc1:Header>
```

## 4.5 Maintaining State Information, Session, Cookie

HTTP is stateless protocol and it is important to retain data on web forms between requests. ASP.NET provides different mechanisms: Query strings, cookies, view state, session state, application state.

Query strings can send additional information along with an address, as shown here: `Response.Redirect("ViewList.aspx?uName=" + name)`. To retrieve a query string in code, use the `QueryString` method, as shown here: `name = Request.QueryString["uName"]`.

Developer can store any type of object into session and retrieve it later. In this project the session state is used to store the `DataBView` for the lifetime of a user session. To store a session is shown as: `Session["records"]=dt`; to retrieve a session variable is shown as:

```
DataView dv =(DataView)Session["records"];
```

#### 4.6 Storing and Retrieving Data using ADO.NET

ADO.NET is used to access and modify data stored in SQL database. The `DataSet` is the central object in ADO.NET. Table rows and columns collection can be used get data items in a `DataSet`, as show in Figure 4-1

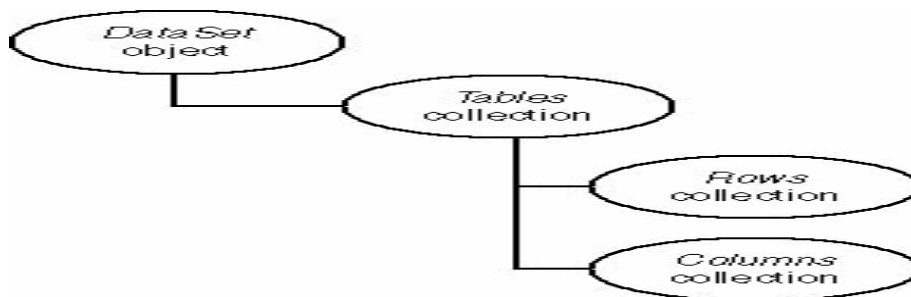


Figure 4-1

Using data-access objects in code follows the same sequence:

1. Create the data connection object
2. Create the command object



3. Create the data adapter object use the command object
4. Create a data set object
5. Invoke methods on the adapter object to fill the data set

Take a look at some code in this project as an example:

```

SqlConnection conn = new
SqlConnection(ConfigurationSettings.AppSettings["ConnectionString
"]);
SqlCommand command = new SqlCommand(s, conn);
SqlDataAdapter adapter = new SqlDataAdapter(command);
DataTable dt = new DataTable();
conn.Open();
adapter.Fill(dt);
conn.Close();

```

## 4.7 Maintaining Security

There are three ways to deal with security in ASP.NET: using windows authentication, using forms authentication and using passport authentication. This project uses forms authentication from database. The root folder allows all users and the seller folder only allows authenticated user. The authentication mode is Web.config is set to forms as shown:

```

<authentication mode = "Forms">
  <forms loginUrl="Login.aspx">
  </forms>
</authentication>

```

The login page is Login.aspx and someone access pages in seller folder will automatically redirected to Login.aspx. The authentication process is shown as Figure 4-2:

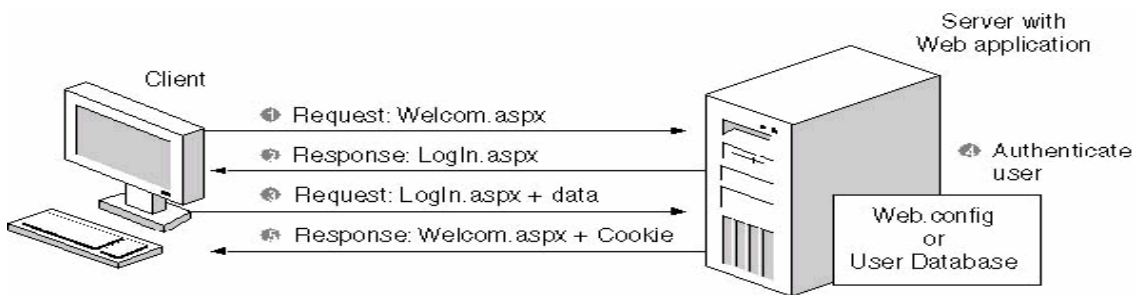


Figure 4-2

## 5. Real Estate Web Application Running on the Sever

After developing the project, it is time to publish the completed web application on a web server for client use. Also, it is important to monitor and maintain the application when it is in use. This project uses Internet Information Server 5.0 on local machine as the web server to host and run the deployed project.

### 5.1 Build and Run on the server

Building a web application in Visual Studio .NET requires little effort. When I click run while developing or debugging the application, it is built automatically. But there are two steps:

1. Set the build options: Set the build option in the configuration manager to Release and set the compilation element in Web.config file to

```
<compilation defaultLanguage="c#" debug="false" />
```

2. Identify the application: Include information about the application such as the application's title, author and version in Assembly Info file.

```
[assembly: AssemblyTitle("Real Estate Web Application")]  
[assembly: AssemblyDescription("House Search and Seller")]  
[assembly: AssemblyConfiguration("")]  
[assembly: AssemblyCompany("")]  
[assembly: AssemblyProduct("Computer Science Department, FSU")]  
[assembly: AssemblyCopyright("Hang Wu, 2005")]  
[assembly: AssemblyTrademark("")]  
[assembly: AssemblyCulture("")]  
  
[assembly: AssemblyVersion("1.0.*")]
```

### 5.2 Test and Run the Project

After the project has been built to a release version of web application and IIS is running on the web server, we can run and test the software. Make

sure there is a virtual folder on IIS called RealEstate. The goal of testing is to ensure that the software meets project requirement. We can go through the web site and see the feature.

FrontPage.aspx (Home Page): It is first page that the client will see. From here, sellers could go the login page and buyer could go the search page.

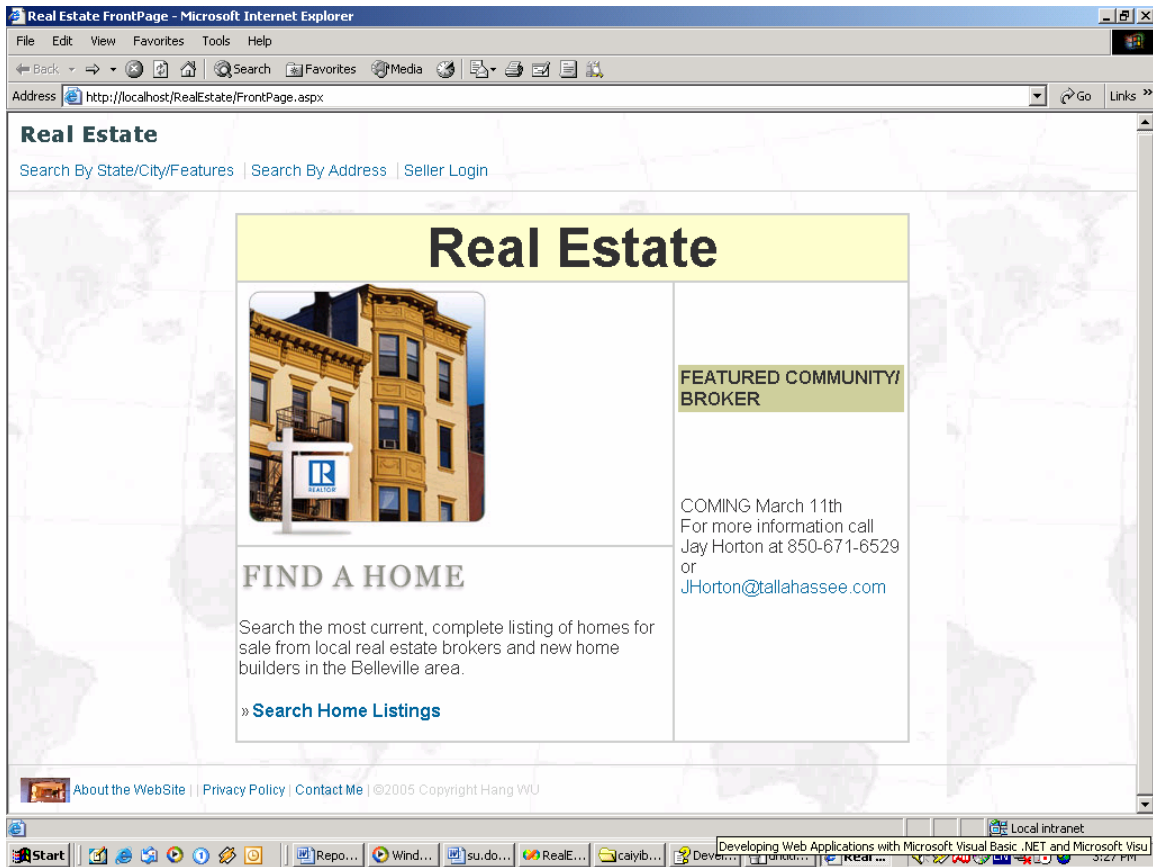


Figure 6-1

(QuickSearch.aspx) Search by Feature Page: Buyer can search the houses by giving the detailed searching parameters, such as state/city/price/bedroom/bathroom/property type/other optional features. By clicking the search button, it will lead to the result page. By clicking clear button, it will clear out the form.

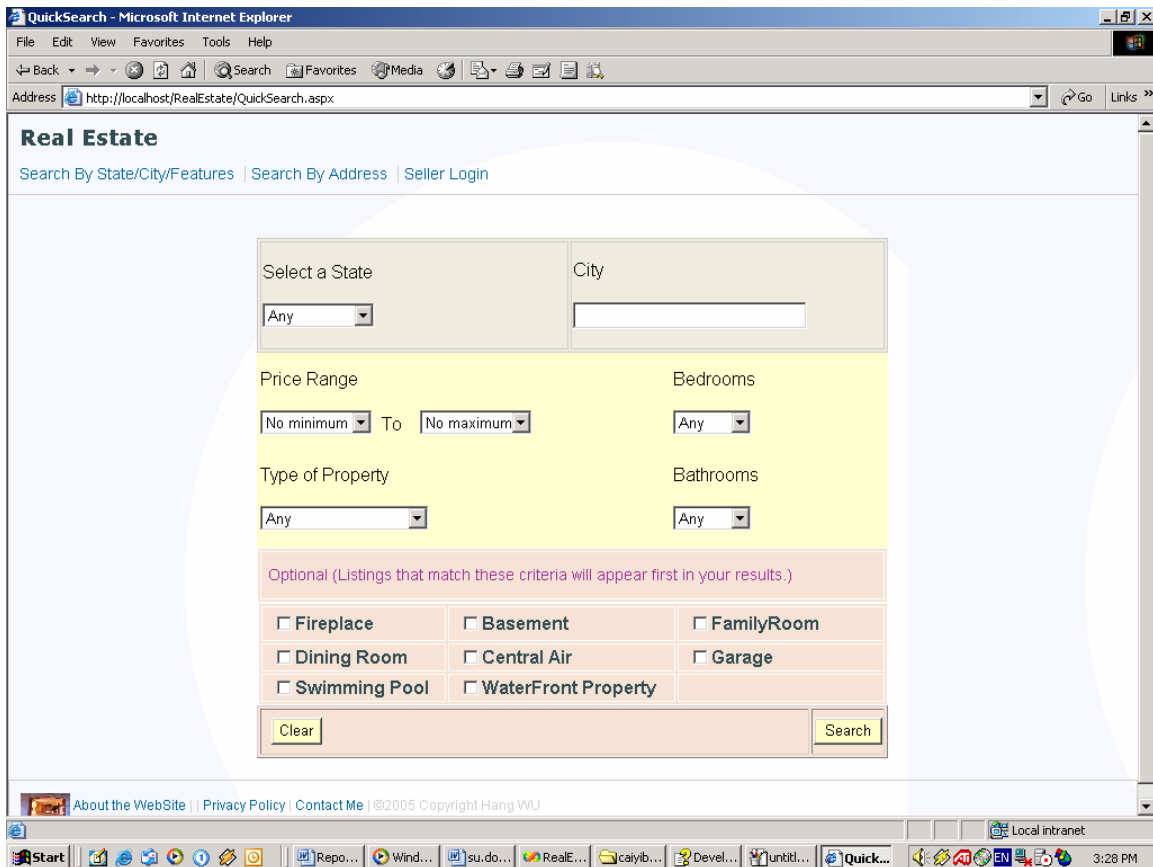


Figure 6-2

SearchAddress.aspx (Search Page by Address Name): It is very convenient for buyer who wants to search a house on a specific street name. User inputs the state/city and key word of the street.

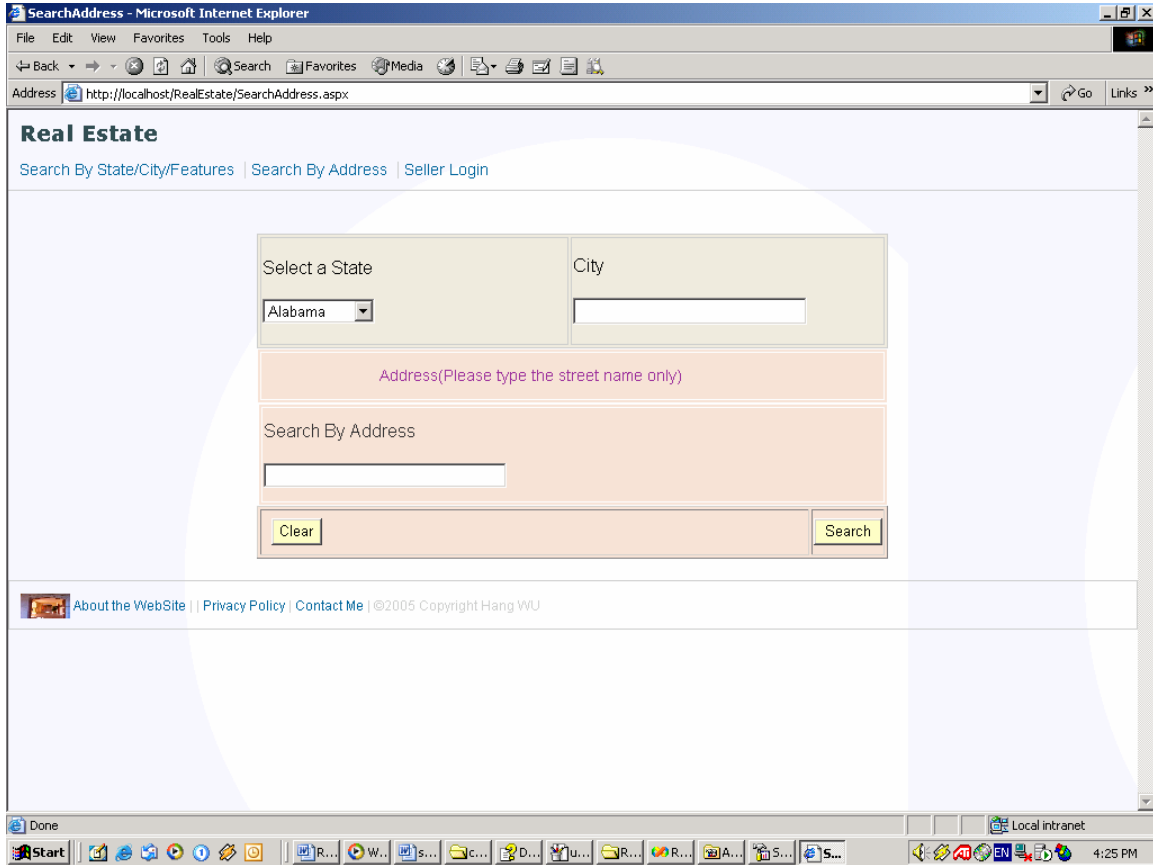


Figure 6-3

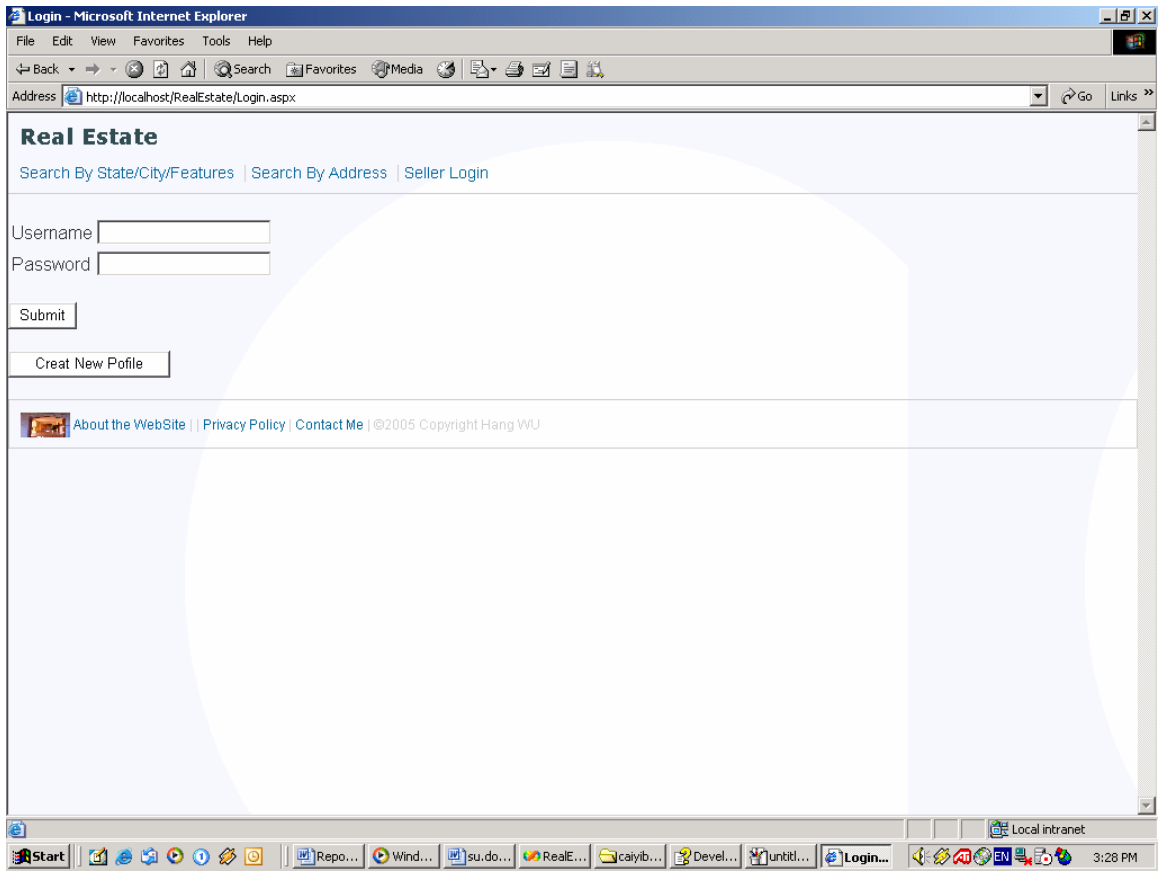


Figure 6-4

Result.aspx (Showing the result Page): Five Results are shown on each page and user can click the page number to view other pages. Also, user can sort the result page by choosing the DropDownList. By clicking the picture, the user can view the detail page.

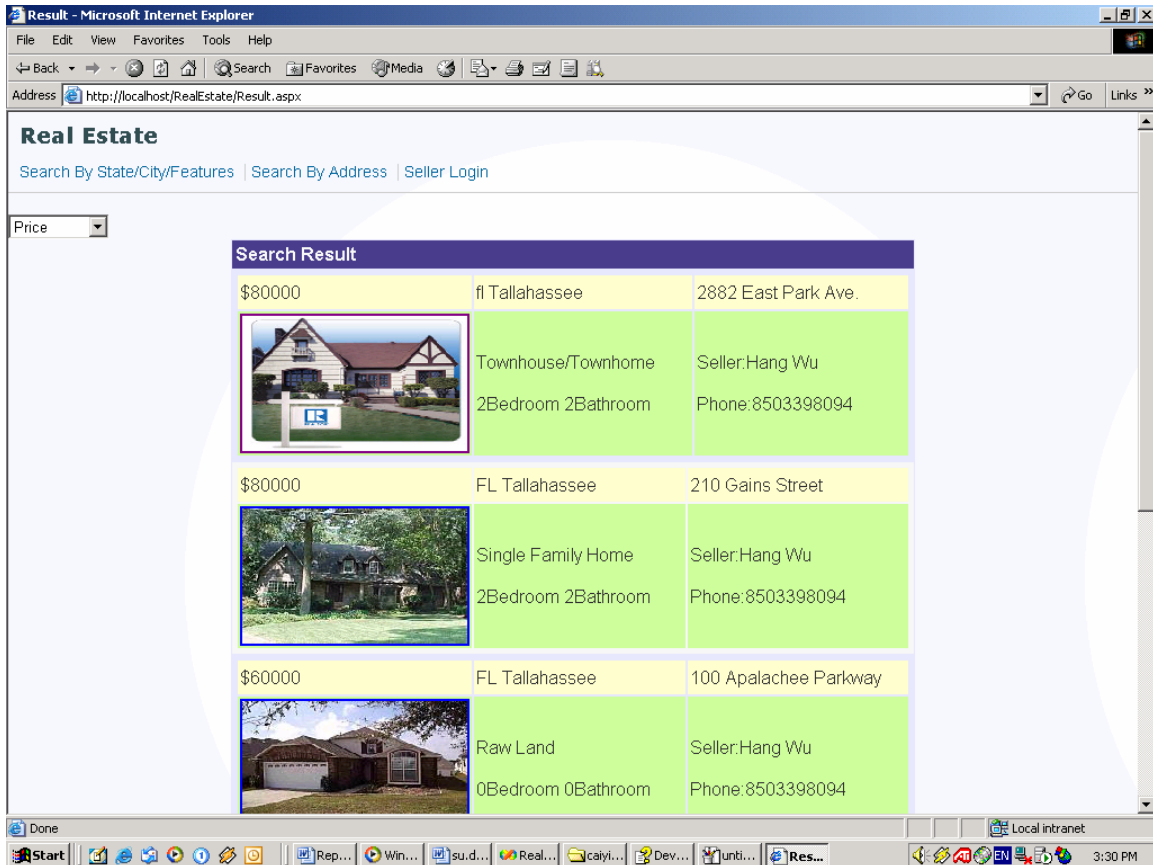


Figure 6-5

Detail.aspx: It displays the detail information of each house record, including the optional features.

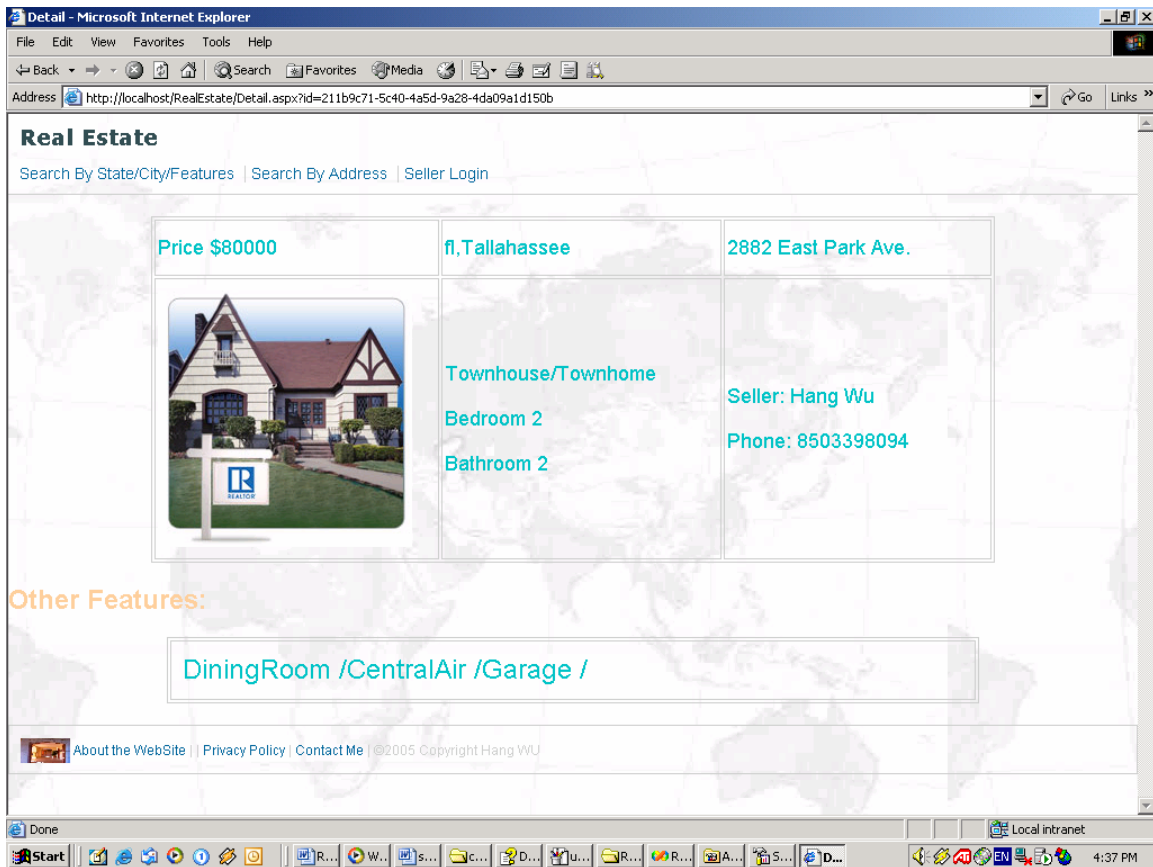


Figure 6-6



Login.aspx (Buyer Login Page): Buyer inputs the use name and password and clicks ‘Submit’ button. If it succeeds, buyer will go to his profile page. If it fails, buyer has to type again. If he is a new user, he can create his own user name and password by clicking the ‘Create New Profile’ button.

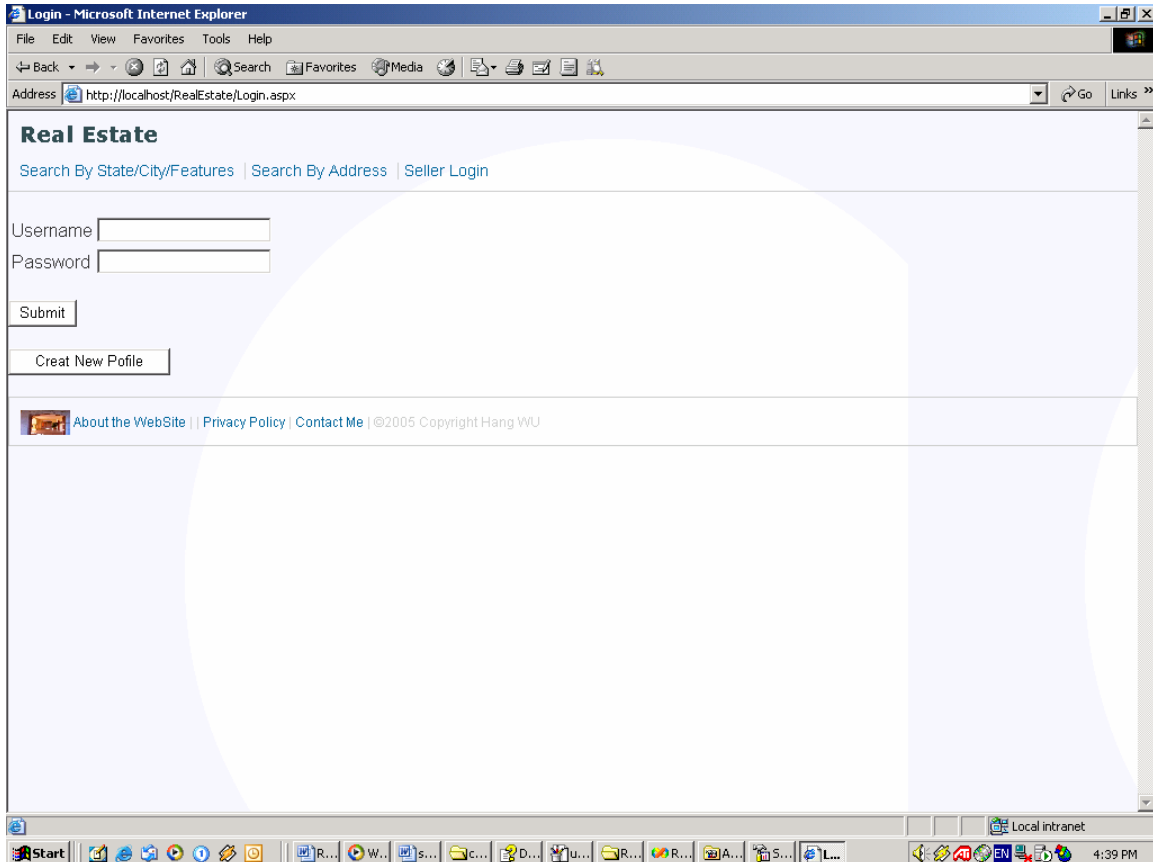


Figure 6-7

ViewList.aspx (Buyer Profile Page): This page will display the previous advertisements the buyer has submits. He can delete each advertisement by clicking the 'Delete' button. Also, by click 'Add' button, user can add a new advertisement.

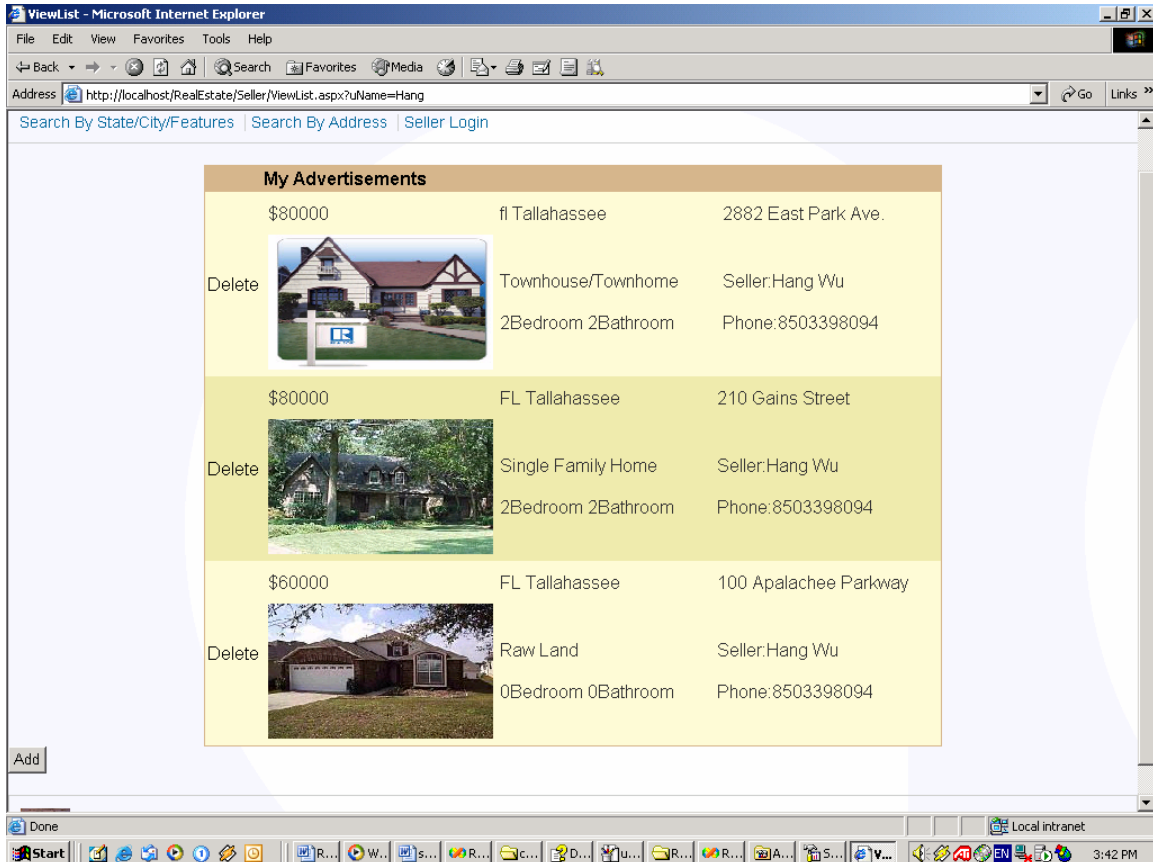


Figure 6-8

AddNew.aspx: Buyer inputs the required information and adds a new advertisement. A new house record will be added into the database.

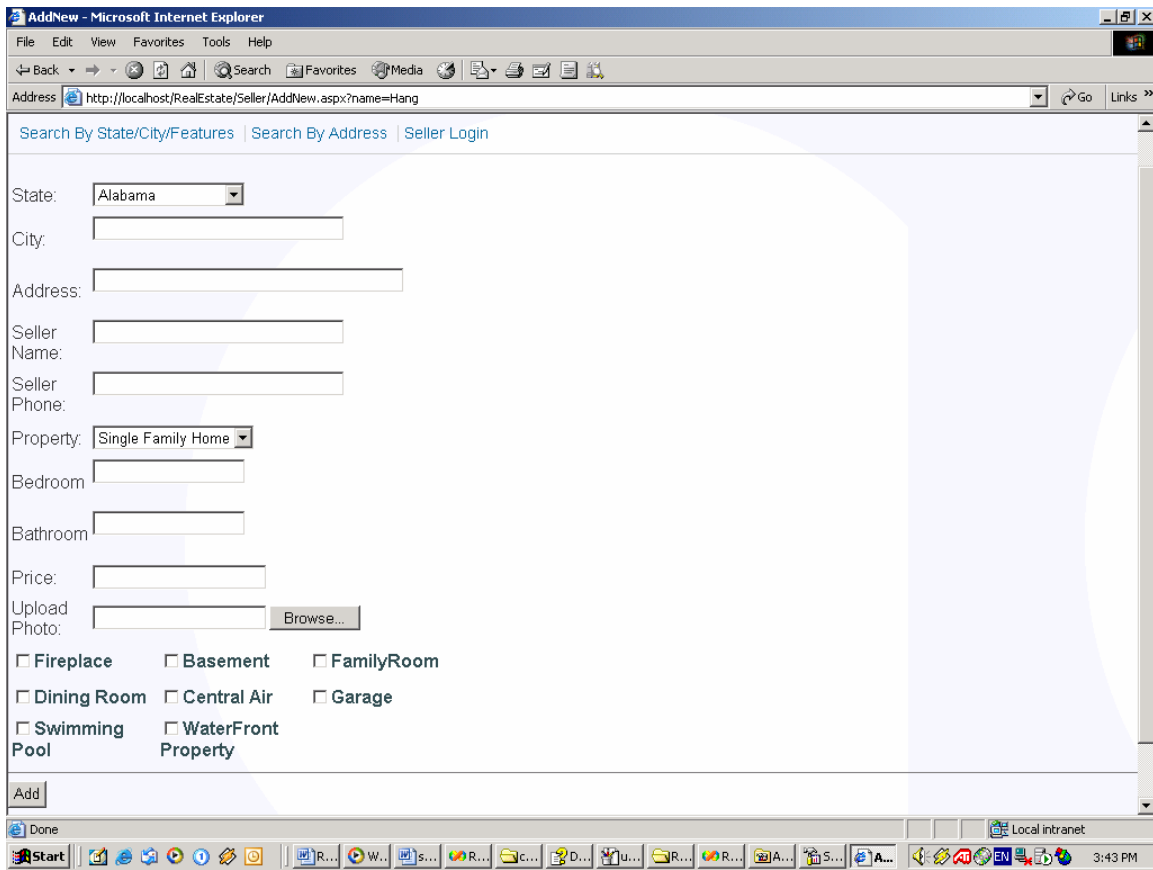


Figure 6-9

CreateNew.aspx: New user can create his/her account on this page. The password is hashed and stored into the database.

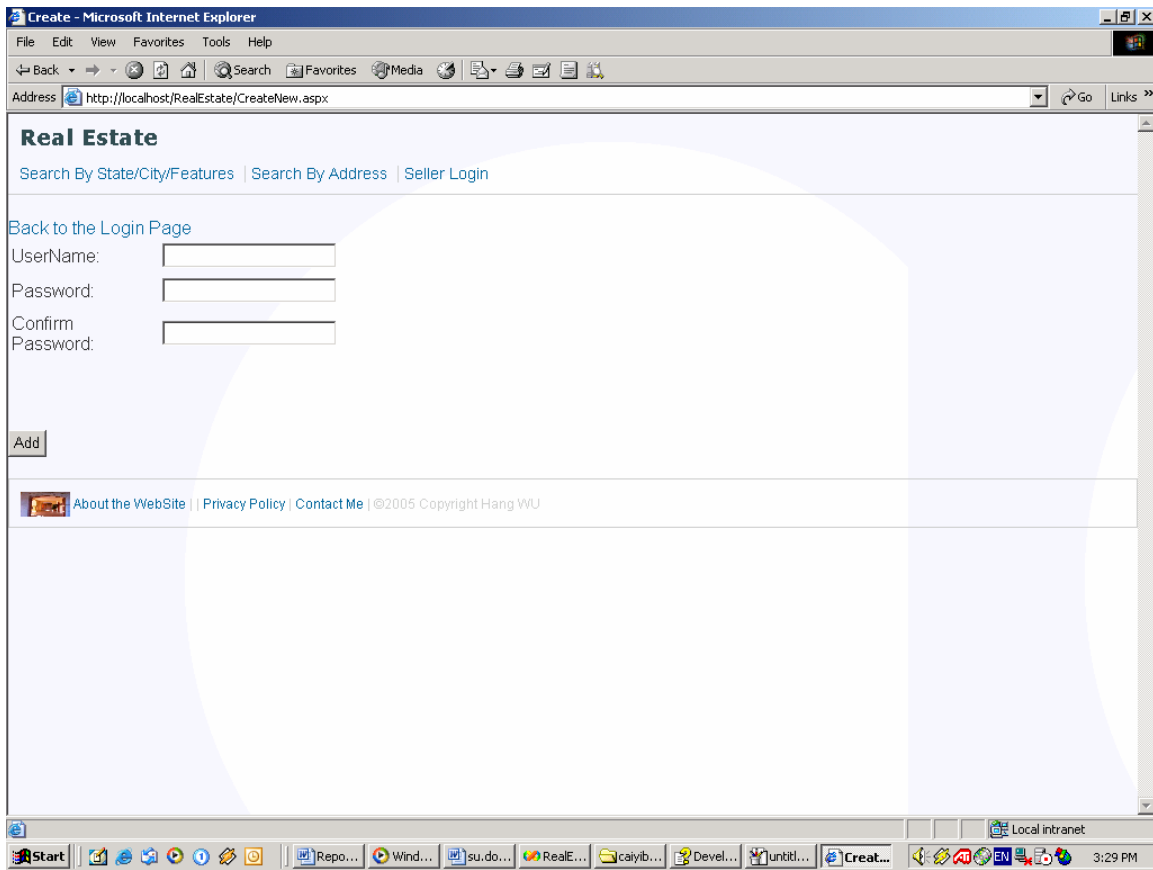


Figure 6-10

## **6. Future Extensions to the Project**

This project is developed as a master's project and we still have some extension which could be made to the project if it is going to be developed as commercial product. We can use pure object-oriented domain model to deal the database access tier using Visual Studio 2005 beta and SQL Server 2003. In doing this, we use a better architecture design which will improve code efficiency and running performance. Besides, we can build XML web services programming model that enables other applications to consuming real estate web services built by us using standard protocol such as HTTP, XML, XSD, SOAP and web services description language (WSDL). We can also extend some functionality in the project which will be described in detail in the following sections.

### **6.1 Using Object-Oriented Domain Model**

Visual Studio .NET 2005 beta and SQL server 2005 add a significant feature of object-oriented domain model. Instead of using traditional Relational Database Model, the database access layer treat each table as class model and each row as instance object. It gives a friendly development thought to developer and makes the business logic more convenient when talking to back end. Currently, there is a third-party tool called NEO which could generate the domain model for database automatically. In using object-oriented domain model, the database is transport to developer and application using these domain classes to handle data processing instead of

using disconnected DataSet or typed DataSet. Figure 7-1 shows the two different architectures.

## 6.2 Providing Web Services

XML and web services are used to build highly scalable, loosely coupled, distributed application using standard web protocols such as HTTP, XML, and SOA. An XML web service is a component that implements the program logic and provides functionality to other applications. There are some examples such as Amazon.com, Google.com. WSDL is used to describe XML web service and XML-based messaging is used to send and receive data.

Visual Studio .NET IDE has a powerful support to develop XML web services. We can create a real estate web application which provides application logic to a client application. I can give an example here, I can create a search function to search the houses based on the feature. The client can call my web service and provide parameters to the search function and the web service will return dataset containing the result house records list back to the client. We can create other convenient functions and the client can easy build their own custom web application while still using the programming logic from the server web services.

## 6.3 Functionality Extensions

If we are going to develop commercial real estate web application based on this project, we can add some more feature as the following:

1. We can provide a login page for buyers and store their username and password in the database.
2. The buyer could add the house record into their wishing cart and next time when they login in, they could view their cart.
3. By using the AdRotator web control we could add more advertisement on the web site or links to other web site.
4. Refine this web site and make it friendly and pretty.

## **7. Conclusion**

This Real Estate Web Application is a typical .NET web application using ASP.NET and SQL 2000 in the C# programming language. It uses a client/sever architecture based on the HTTP protocol. It is developed in Microsoft's Visual Studio .NET programming environment. Some ways in which this system could be enhanced with additional functionalities have been discussed. Whereas this system was developed using Visual Studio .NET 2003, a future version might use the newer 2005 version (currently still in beta testing), which provides an object-oriented domain model.



## **8. User Manual**

### **Server System Requirements:**

Microsoft .NET Framework 1.1 installed

Visual Studio .NET 2003 installed(optional)

SQL Server 2000 installed

Internet Information Service installed

Windows 95/98/NT/Me/2000/XP/2003 server

### **Client System Requirements:**

Windows 95/98/NT/Me/2000/XP/2003 server (optional)

Internet Explore 5.0 or higher/Netscape Navigator/Mozilla Firefox installed

### **Server Installation:**

1. Install IIS and start the web server in windows operation system.
2. Install Microsoft .NET Framework 1.1 and register ASP.NET.
3. Install SQL Server 2000
4. Create a visual directory called RealEstate and point it to the RealEstate physical directory containing the source code.
5. Run the database scripts called realstate.sql in the package

### **Client Installation:**

1. Install Web Brower

2. Brower <http://www.address/RealEstate/FrontPage.aspx>

## 9. References

1. Susmitha Athota, A Real Estate Listing Web Service using IBM Websphere, Master's Degree Project Report, Department of Computer Science, Florida State University, Fall 2004.
2. Developing Web Applications with Microsoft Visual Basic .NET and Microsoft Visual C# .NET// Microsoft Corporation.
3. Developing XML Web Services and Server Components with Microsoft Visual Basic .NET and Microsoft C# .NET / Microsoft Corporation.
4. Developing Windows-Based Applications with Microsoft Visual Basic .NET and Microsoft Visual C# .NET / Microsoft Corporation.
5. Microsoft SQL Server 2000 Database Design and Implementation /Microsoft Corporation.
6. C# Programming Introduction/Deitel Developer Series.
7. <http://asp.net>
8. <http://msdn.microsoft>
9. <http://www.omg.org>
10. <http://www.edm2.com/>