

**《C#程序设计》选修课期末大作业报告**



题 目**：**

学 院：

专 业： 班 级：

学 号： 学 生：

授课教师：

时 间：

[1. 摘要 2](#_Toc12853)

[2. 开发背景及目的 2](#_Toc20587)

[2.1. 背景 2](#_Toc14978)

[2.2. 目的 2](#_Toc4292)

[3. 需求设计及分析 3](#_Toc26719)

[4. 程序设计 3](#_Toc19100)

[4.1. 系统功能结构图 3](#_Toc16424)

[4.2. 数据库设计 3](#_Toc17516)

[5. 应用程序详细设计 5](#_Toc7126)

[5.1. 登录设计 5](#_Toc28705)

[5.1.1. 登录代码： 5](#_Toc3611)

[5.1.2. 登录界面： 6](#_Toc9299)

[5.2. 首页设计 7](#_Toc3877)

[5.2.1. 首页代码 7](#_Toc3180)

[5.2.2. 首页界面 10](#_Toc18640)

[5.3. 数据库连接 10](#_Toc28327)

[6. 总结 14](#_Toc17677)

# 摘要

随着信息技术的发展，越来越多的专业博客浮出了水面。在这些博客里可以记载

日常发生的事情和自己的兴趣爱好，可以跟他人分享、交流自己的思想和知识，也可

以自由随意尽情地去表达自己。由于自己平时需要记载的东西很多，也很需要跟别人

交流自己的心得，因此设计了这个基于ASP.NET 的个人博客网站。该网站使用

ASP. NET和SQL Sever2019数据库技术，在VS的环境下实现了一个相对完善

的个人博客网站。。该网站系统还提供了完善的后台管理功能，方便了管理员对该网站系统进行管理和升级。后台博客管理员可以对文章的类型、文章的内容进行查看与修改。

# 开发背景及目的

## 背景

博客的个性化和平民视角使得它提供的消息更贴近人们的生活，所以很多人都

想建立自己的网络空间。过去，传统的网络交往方式主要是留言本、BBS (论坛)、

聊天室及IM(即使通信)等，但它们或多或少都存在着不足。留言本主要是用来留言，不能进行留言回复; BBS主要用来探讨问题; IM 要想发挥作用，必须要求交流的双方同时在线;而聊天室更是闲人的乐园。博客的存在，可以说是一种虚拟社区。在

这里用户可以通过网络日志的形式方便、快捷地发表自己的心得体会，及时有效并轻松地与他人交流“。

## 目的

该网站开发的目的是为了能够让很多人都拥有自己的网络空间。通过设计网站加深对课上知识的理解，并在设计中学习到- -些课上学不到的东西，充分提高学生对网站设计的学习兴趣。

# 需求设计及分析

通过查询一-些特色网站的设计理念，本网站的设计思路如下:

(1)用户分普通用户、管理员用户

(2)超级管理员可对博客管理员进行管理和设置权限

(3)博客管理员涉及对博客的类型管理、文章管理、图片管理、评论管理、链

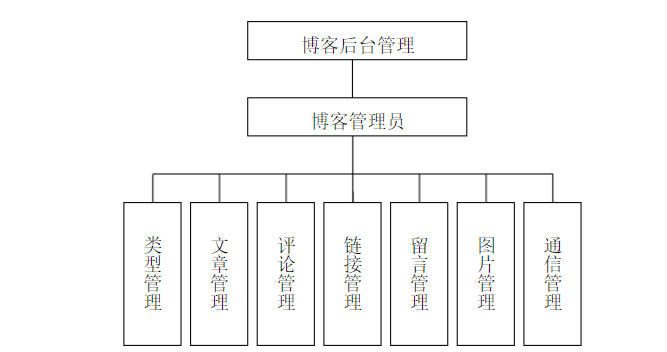
接管理、留言管理和通信管理

(4)普通用户可以阅读文章、发表回应、留言和评论

(5) - -种文章类型有多篇文章

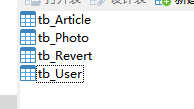
# 程序设计

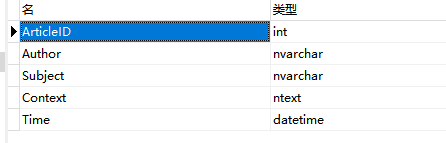
## 系统功能结构图

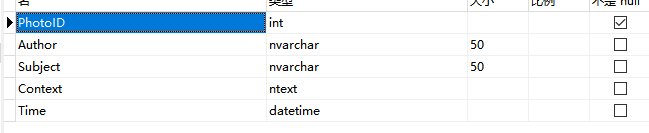


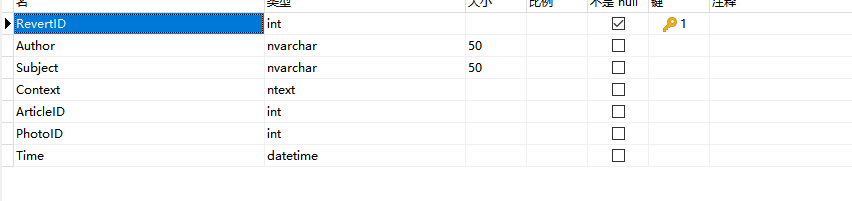
## 数据库设计

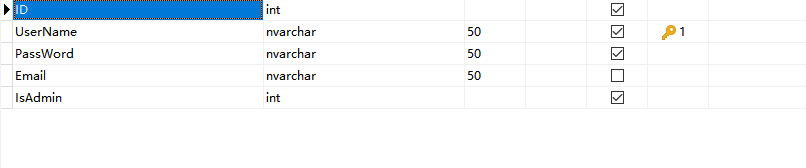
系统开发过程中，数据库设计占有重要的地位，数据库设计的原则是根据系统的整体需求而定，通过本网站进行的需求分析、网站系统功能的确定，规划出系统中使用的数据库实体对象分别为文章类型信息实体、链接信息实体、留言信息实体、文章信息实体。











# 应用程序详细设计

## 登录设计

### 登录代码：

using System;using System.Collections.Generic;using System.Linq;using System.Web;using System.Web.UI;using System.Web.UI.WebControls;using System.Data.SqlClient;

public partial class Account\_Login : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

DB db = new DB();

MessageBox message = new MessageBox();

string userName = this.txtboxUserName.Text.Trim();

string passWord = this.txtboxPassWord.Text.Trim();

SqlDataReader dr = db.reDr("SELECT \* FROM tb\_User WHERE UserName='"

+userName+"' AND PassWord='"+passWord+"'");

dr.Read();

if (dr.HasRows)

{

Session["UserName"] = userName; //用session保存用户名，作为全局使用

Response.Write(message.SendMessageBox("登陆成功", "../Blog/Index.aspx"));

}

else

{

Response.Write(message.SendMessageBox("登陆失败", "Login.aspx"));

}

dr.Close();

}

protected void Button2\_Click(object sender, EventArgs e)

{

Response.Redirect("~/Account/Register.aspx");

}

}

### 登录界面：



## 首页设计

### 首页代码

<%@ Page Title="" Language="C#" MasterPageFile="~/Blog/BlogMasterPage.master" AutoEventWireup="true" CodeFile="ArticleManager.aspx.cs" Inherits="Blog\_Article" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder2" Runat="Server">

<p>

<table style="width: 100%; height: 902px;">

<tr>

<td style="text-align: right">

<asp:Button ID="Button1" runat="server" onclick="Button1\_Click"

style="text-align: right" Text="发表文章" />

</td>

</tr>

<tr>

<td>

<table style="width:70%; height: 870px;">

<tr>

<td style="width: 969px; text-align: center;">

<asp:DataList ID="DataList1" runat="server" DataKeyField="Subject"

DataSourceID="SqlDataSource1" Width="697px" Height="16px">

<ItemTemplate>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<table style="width:100%; height: 170px;">

<tr>

<td colspan="2" style="text-align: center">

文章标题：<asp:Label ID="SubjectLabel" runat="server" Text='<%# Eval("Subject") %>' />

</td>

</tr>

<tr>

<td style="width: 354px; text-align: center">

&nbsp;</td>

<td style="text-align: center">

&nbsp;</td>

</tr>

<tr>

<td style="width: 354px; text-align: center">

作者：<asp:Label ID="AuthorLabel" runat="server" Text='<%# Eval("Author") %>' />

&nbsp;&nbsp;&nbsp;

</td>

<td style="text-align: center">

时间<asp:Label ID="TimeLabel" runat="server" Text='<%# Eval("Time") %>' />

</td>

</tr>

<tr>

<td style="width: 354px; text-align: center">

&nbsp;</td>

<td style="text-align: center">

<br />

<br />

</td>

</tr>

<tr>

<td style="text-align: center; width: 354px">

&nbsp;</td>

<td style="text-align: center">

<asp:LinkButton ID="LinkButton6" runat="server"

PostBackUrl='<%# "ShowArticle.aspx?ArticleID=" + DataBinder.Eval(Container.DataItem,"ArticleID")%>'>查看全文</asp:LinkButton>

<br />

</td>

</tr>

<tr>

<td style="text-align: right; width: 596px" colspan="2">

&nbsp;&nbsp;&nbsp;&nbsp;

</td>

</tr>

</table>

<br />

</ItemTemplate>

</asp:DataList>

<asp:SqlDataSource ID="SqlDataSource1" runat="server"

ConnectionString="<%$ ConnectionStrings:db\_BlogConnectionString %>"

SelectCommand="SELECT [Author], [Subject], [Time],[Context],[ArticleID] FROM [tb\_Article]">

</asp:SqlDataSource>

</td>

</tr>

</table>

</td>

</tr>

</table>

</p></asp:Content>

### 首页界面



## 数据库连接

using System;using System.Configuration;using System.Data;using System.Linq;using System.Web;using System.Web.Security;using System.Web.UI;using System.Web.UI.HtmlControls;using System.Web.UI.WebControls;using System.Web.UI.WebControls.WebParts;using System.Xml.Linq;

using System.Text;using System.Security.Cryptography;//MD5加密需引入的命名空间using System.Data.SqlClient;//数据库操作需引入的命名空间

/// <summary>/// DB 的摘要说明/// </summary>public class DB

{

SqlConnection myConn; //用于连接数据库

SqlCommand myCmd; //用于执行Sql语句

DataSet ds; //数据集

SqlDataAdapter adapt; //填充数据集

public DB()

{

//

// TODO: 在此处添加构造函数逻辑

//

}

/// <summary>

/// 连接数据库

/// </summary>

/// <returns>返回SqlConnection对象</returns>

public SqlConnection GetCon()

{

return new SqlConnection(ConfigurationManager.AppSettings["ConnectionString"].ToString());

}

/// <summary>

/// 执行SQL语句

/// </summary>

///<param name="cmdstr">SQL语句</param>

/// <returns>返回值为int型：成功返1，失败返回0</returns>

public int sqlEx(string cmdstr)

{

SqlConnection con = GetCon();//连接数据库

con.Open();//打开连接

SqlCommand cmd = new SqlCommand(cmdstr, con);

try

{

cmd.ExecuteNonQuery();//执行SQL 语句并返回受影响的行数

return 1;//成功返回１

}

catch (Exception e)

{

return 0;//失败返回０

}

finally

{

con.Dispose();//释放连接对象资源

}

}

/// <summary>

/// 执行SQL查询语句

/// </summary>

/// <param name="cmdstr">查询语句</param>

/// <returns>返回DataTable数据表</returns>

public DataTable reDt(string cmdstr)

{

SqlConnection con =GetCon();

SqlDataAdapter da = new SqlDataAdapter(cmdstr, con);

DataSet ds = new DataSet();

da.Fill(ds);

return (ds.Tables[0]);

}

/// <summary>

/// 执行SQL查询语句

/// </summary>

/// <param name="str">查询语句</param>

/// <returns>返回SqlDataReader对象dr</returns>

public SqlDataReader reDr(string str)

{

SqlConnection conn = GetCon();//连接数据库

conn.Open();//并打开了连接

SqlCommand com = new SqlCommand(str, conn);

SqlDataReader dr = com.ExecuteReader(CommandBehavior.CloseConnection);

return dr;//返回SqlDataReader对象dr

}

/// <summary>

/// MD5加密

/// </summary>

/// <param name="strPwd">被加密的字符串</param>

/// <returns>返回加密后的字符串</returns>

public string GetMD5(string strPwd)

{

string pwd = "";

//实例化一个md5对象

MD5 md5 = MD5.Create();

// 加密后是一个字节类型的数组，这里要注意编码UTF8/Unicode等的选择

byte[] s = md5.ComputeHash(Encoding.UTF8.GetBytes(strPwd));

//翻转生成的MD5码

s.Reverse();

//通过使用循环，将字节类型的数组转换为字符串，此字符串是常规字符格式化所得

//只取MD5码的一部分，这样恶意访问者无法知道取的是哪几位

for (int i = 3; i < s.Length - 1; i++)

{

//将得到的字符串使用十六进制类型格式。格式后的字符是小写的字母，如果使用大写（X）则格式后的字符是大写字符

//进一步对生成的MD5码做一些改造

pwd = pwd + (s[i] < 198 ? s[i] + 28 : s[i]).ToString("X");

}

return pwd;

}

/// <summary>

/// 说 明： GetDataSet数据集，返回数据源的数据表

/// 返回值： 数据源的数据表

/// 参 数： sqlStr执行的SQL语句，TableName 数据表名称

/// </summary>

public DataTable GetDataSet(string strSql, string TableName)

{

ds = new DataSet();

try

{

myConn = GetCon();//与数据库连接

adapt = new SqlDataAdapter(strSql, myConn); //实例化SqlDataAdapter类对象

adapt.Fill(ds, TableName);//填充数据集

return ds.Tables[TableName];//返回数据集DataSet的表的集合

}

catch (Exception ex)

{

throw new Exception(ex.Message, ex);

}

finally

{//断开连接，释放资源

myConn.Close();

adapt.Dispose();

ds.Dispose();

myConn.Dispose();

}

}

}

# 总结

由于知识有限，很多地方不太懂，所以该体系系统还有很多不近人意的地方。虽然在设计的时候遇到了很多问题，但是更体验到了设计的快乐，让自己在编程的同时感到了充实有意义。受益匪浅