博客需求分析

1. 概 述

对个人而言，博客可以调动个人的积极性，充分发挥个人的创造性。从2002年开始，国内博客开始迅猛发展起来，从最初的只有几千个尝鲜者到现在的各式各样的博客，圈子。在这三年的发展过程中，主要呈现出如下几个特点：1.博客是可以将个人信息完整呈现在互联网上的载体。2.博客具有“准实名制”趋势，有助于信任机制的建立。本文就此探讨了我国博客发展的几个要素。

  1.1个人BLOG的概念

BLOG是WEB LOG的缩写，简单来说就是网络日记。它是1997年12月由美国的Jorn Barger最早提出的，这些年全球博客获得了突飞猛进的发展。据博客搜索Technorati统计，目前全球平均每5.8秒钟诞生一个博客。它是一个新型的个人互联网出版工具，博客使用者可以很方便地用文字、链接、影音、图片建立起个性化的网络世界。

在Internet飞速发展的今天，互联网成为人们快速获取、发布和传递信息的重要渠道，它在人们政治、经济、生活等各个方面发挥着重要的作用。因此网站建设在Internet应用上的地位显而易见，倍受人们的重视。为了树立个人在网络上的形象，通过设计一个个人博客网站，以达到个人推荐、展现自我的一个平台，个人博客网站为交流提供综合的网络环境，同时也为个人的宣传，发展，提供了一个平台。

博客秉承了个人网站的自由精神，但是综合了激发创造的新模式，使其更具开放和建设性。它作为一种新表达的方式，它的传播不仅情绪，包括大量的智慧、意见和思想。某种意义上说，它也是一种新的文化现象，博客的出现和繁荣，真正凸现网络的知识价值，标志着互联网发展开始步入更高的阶段。 1.2个人BLOG具有的特点和主要功能实现一个博客网站，通常是由简短而且经常更新的帖子构成，所有文章都是按照年份和日期排列，有些类似斑竹的日记。看上去平淡无奇，毫无可炫耀之处，但它可以让每个人零成本、零维护地创建自己的网络媒体，每个都可以随时把自己的思想火花和灵感更新到博客站点上。

零机制：博客属于个人所有，是自发建立和非正式的建设。BSP不能对博客发号施令。  零技术：博客不需要作网站的专业知识，可以像发邮件一样简单地实现博客网页的呈现。技术的极度简化，包括架构和申请博客网站、编辑、上传和修改内容等。

零成本：任何一个人都可以像申请免费邮件般免费申请自己的博客网站。不需要注册域名的成本，不需要租用服务器空间的成本，不需要许多软件工具的成本。

零形式：博客页面的表现形式和文章内容的表现方式十分灵活，没有特定形式。

与个人主页相比较，博客是使用方便的、交互性强。与传统的电子邮件、BBS和ICQ这三种互联网沟通方式相比，博客是一种较严肃的沟通平台。BBS公共匿名性很强，而个人性很弱，因此缺乏约束。电子邮件和ICQ则是多用于个人间的通讯，而博客是个人性和公共性的结合。博客是个人在网上展示自己、与别人沟通交流的综合平台，它的管理比BBS简单的多。

2. 需求分析

信息时代的今天，网络已经成为人们工作、学习的一部分，不断充实和改变着人们的生活。在网络中，构建一个个个性化的博客，可以充分地表达自己的思想，通过发布文章展示个人才能，舒服个人情感；网友则可以根据主题发表个人的意见，表达自己的想法，与博主进行思想交流，乐在其中。

2.1 可行性研究

2.1.1 经济可行性

博客通过“我的文章”和“相册上传”等功能模块，吸引更多的浏览者，增加网站的访问量，提升网站的影响力；不断更新网站信息内容，使网站信心更加多元化，适合不同的访问群体；通过网站展现个人魅力，愉悦身心；通过与来访者的交流，不断积累经验，吸收新的知识，使网站始终充满新鲜活力。 2.1.2 技术可行性

本系统主要应用Servlet技术开发。Servlet技术带给程序员最大的帮助是它可以处理客户端传来的HTTP请求，并返回一个响应。Servlet是一个Java类，Java语言能够实现的功能，Servlet基本上都可以实现（图形界面以外）。总的来说，Servlet技术具有以下特点。

（1）高效

在服务器上仅有一个Java虚拟机在运行，它的优势在于当多个来自客户端的请求进行访问时，Servlet为每个请求分配一个线程而不是进程。

（2）方便

Servlet提供了大的实用工具例程，例如处理很难完成的HTML表单数据、读取和设置HTTP头、处理Cookie和跟踪会话功能。

（3）跨平台

 Servlet时用Java类编写的，它可以在不通过的操作系统平台和不同的应用服务器平台下运行。

（4）功能强大

在Servlet中，许多使用传统CGI程序很难完成的任务都可以用Servlet技术轻松完成。例如，Servlet能够直接和Web服务器交互，而普通的CGI程序则不能，Servlet还能够在各个程序之间共享数据，使得数据库连接池之类的的功能很容易实现。

（5）灵活性和可扩展性采用Servlet开放的Web应用程序，由于Java类的继承性、构造函数等特点，使得应用灵活，可随意扩展。

（6）共享数据Servlet之间可通过共享数据很容易的实现数据库连接池。它能方便的实现管理用户的请求，简化Session和获取前一页面信息的操作，而在CGI之间通信则很差。由于每个CGI程序的调用都开始一个新的进程，调用间通信要通过文件进行，因而相当缓慢。同一台服务器上的不同CGI程序之间的通信业相当麻烦。

（7）安全 有些CGI版本存在明显的安全弱点，即使是使用最新的标准和PERL等语言，系统也没有基本安装框架。而Java定义有完整的安全机制，包括SSL\CA认证、安全政策等规范。

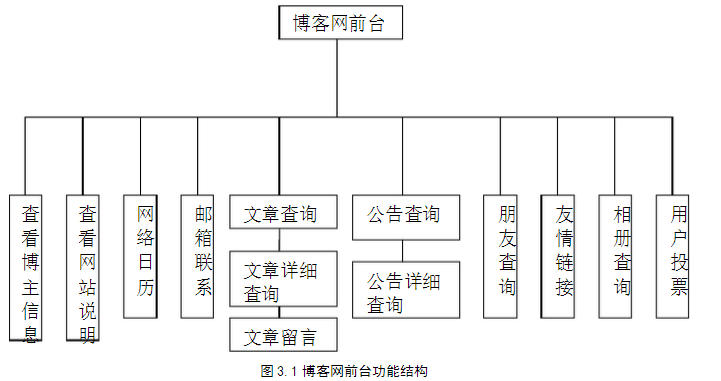
**3．系统设计**

3.1系统目标

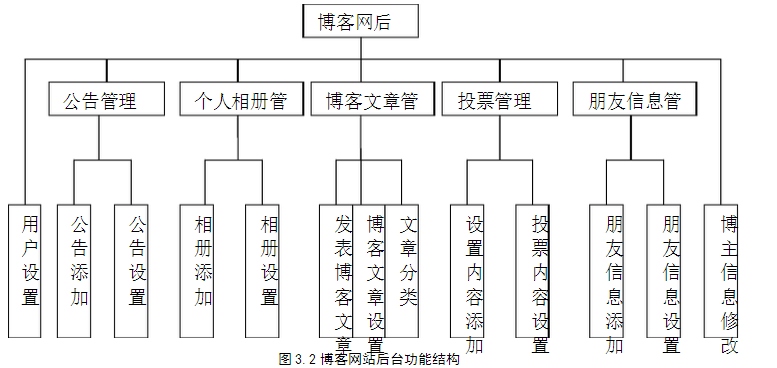
本系统主要实现网站建设者与访问者之间的交流，展现网站建设的思想。主要实现如下目标：

全面展示博客网站的主题思想 通过发表“我的文章”，表达作者的想法和观点，实现与访问者的相互交流。通过上传个人相册，使访问者对网站建设者有进一步的了解。用户可以查看文章，并可以对文章发表意见。 展现最新的的博客列表，方便访问浏览。 提供网站统计信息，使访问者了解网站的基本信息。对文章信息进行管理。通过选择文章类别名称添加文正内容，并可以查询、修改和删除文章内容。 对相册信息进行管理。通过选择相册类别名称上传相册图片信息，并可以查看和删除相册内容。可以修改管理员登陆信息。

3.2 系统功能结构

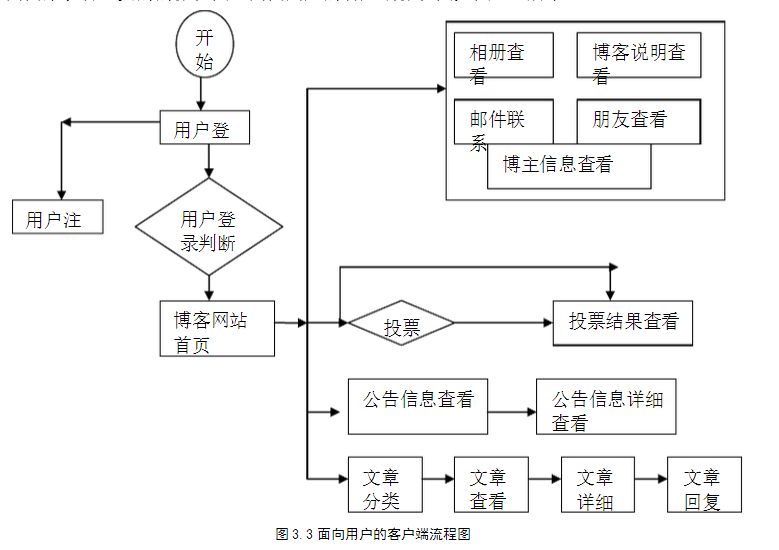
博客网站分为前台和后台。其中，根据博客前台的特点，可以将其分为博主信息、网络日历、文章查询、公告查询及相册查询等13部分，其中各个部分及其包括的具体功能模块如图所示。

根据博客网站后台的特点，可以将其分为用户设置、公告管理、个人相册管理、博客文章管理、投票管理、朋友信息管理及博主信息修改7部分，其中各个部分及其包括的具体功能模块如图所示

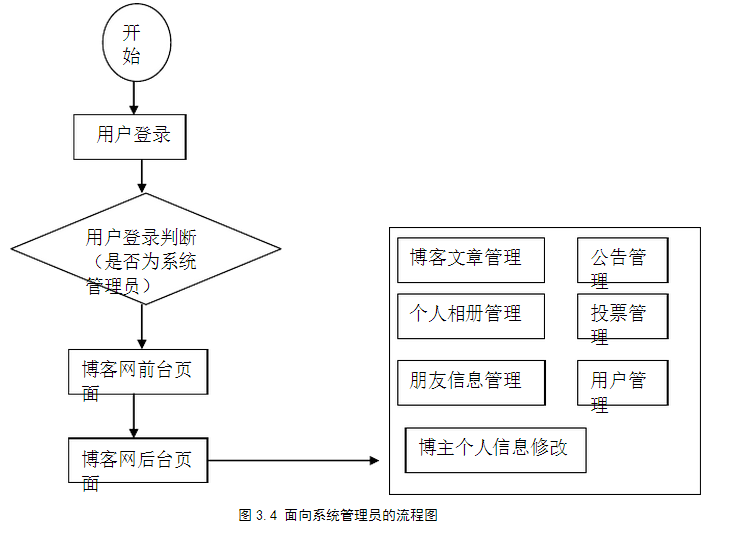


3.3 系统流程图

为了更加清晰的表达系统的业务功能模块，下面给出博客网的系统流程图。对于不同的角色，其所承担的任务也各自不同，流程图也不一样，包括面向用户的客户端流程图和面向系统管理员的的流程图。面向用户的客户端流程图如图所示。



面向系统管理员的流程图如图3.4所示。



3.4 开发环境

在开发博客网系统时，需要具备下面的软件环境。 服务器端：

操作系统：Windows xp Web服务器：Tomcat 6.0 Java 开发包：JDK 1.5以上 数据库：SQL Server2000 浏览器：IE6.0

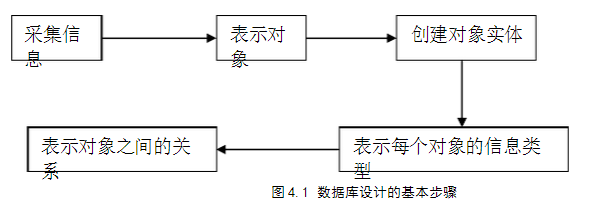
分辨率：最佳效果为1024\*768像素 客户端： 浏览器：IE6.0

分辨率：最佳效果为1024\*768像素

4数据库设计

4.1数据库需求设计

创建博客数据库的第一步是制定实施计划，该计划可以在实现数据库时做向导，也可以在数据库实现完成后，用做数据库的功能说明。数据库设计的复杂性和细节由数据库的应用程序的复杂性、大小及用户使用情况确定。这里介绍的博客数据库是一个相对简单的数据库，尽管如此，数据库设计还是需要经过如图4.1所示的基本步骤。



下面详细介绍将数据库设计的基本步骤。

4.1.1 采集信息

在创建博客数据库之前，必须对博客信息有一个透彻的理解。与博客网中所涉及的每个人进行交流，以找出他们所作的工作及他们需要从数据库中获得的内容，这是很重要的。另外，了解该博客系统是否符合实际要求及找出相关系统的问题、限制和屏蔽也很重要。

4.1.2 表示对象

在采集信息的过程中，必须标识将由博客数据库管理的主要对象或实体。对象可以是一个有形的实体（如用户），也可以是一个无形的项目（文章、相册等）。通常会有几个主要对象，在标识了这些对象之后，相关的项目就会变得清晰。数据库中每个不同的项目都应有一个对应的表。

4.1.3 建立对象实体

标识了博客系统中的对象后，使用该博客系统的一种直观表示方式对这些对象进行记录是很重要的。在数据库实现期间，可以使用数据库实体作为参考。

4.1.4 标识每个对象的信息类型

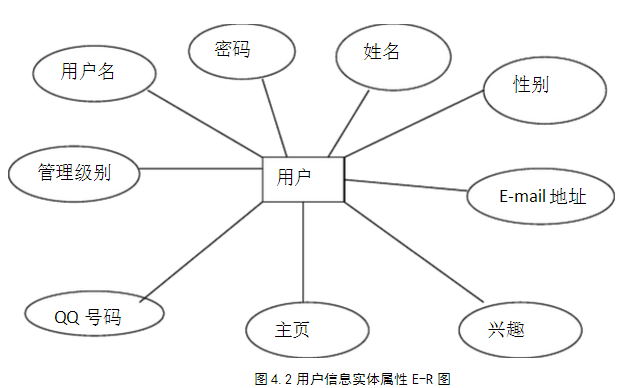
  标识必须为各对象存储的信息类型，这些信息是对象的表中的列（实际上就是数据表字段的类型）。

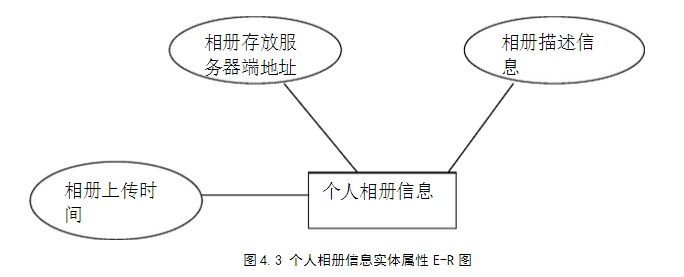
4.1.5 标识对象之间的关系

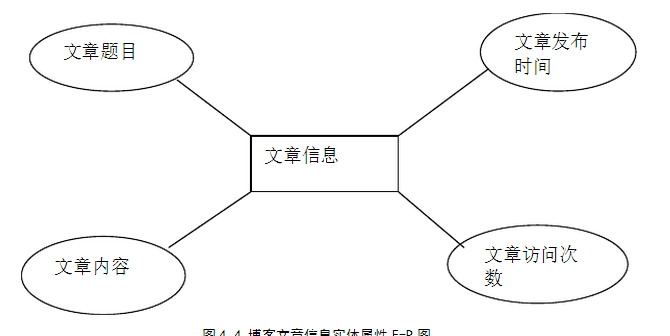
 关系数据库的一个优势是它具有联系或关联数据库中各种项目有关信息的能力。可以单独地存储独立的信息类型，但数据库可以在必要时组合数据。在设计过程中，表示对象间的关系需要查看各种表，以确定它们逻辑相关的方式及添加管系列以建立表与另一个表间的关系。

4.2 数据库概念的设计

根据以上对系统所作的需求分析和系统设计，规划出本系统中使用的数据库实体信息分别为用户信息实体、公告信息实体、个人相册信息实体、朋友信息实体、投票信息实体、博客文章信息实体及文章回复实体。下面将介绍几个关键实体的E-R图。 用户信息实体

用户信息实体包括用户名、密码、姓名、性别、QQ号码、主页、兴趣、E-mail地址、管理级别属性，用户信息实体的E-R图如图所示。个人相册信息实体

个人小相册信息实体包括相册存放服务器端地址、相册描述信息及相册上传时间属性，个人相册信息实体的E-R图如图所示。博客文章信息实体

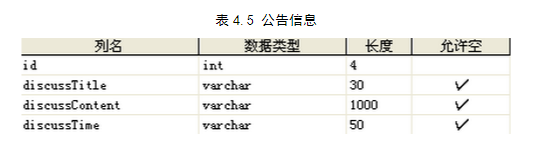
博客文章信息实体包括文章题目、文章内容、文章发布时间及文章访问次数属性。博客文章信息实体的E-R图如图4.4所示。4.3 数据库逻辑结构设计

本实例包含8个数据表，分别为个人相册信息表，用户信息表，文章信息表，文章类型信息表，公告信息表，朋友信息表，文章评语信息表，投票信息结果表。

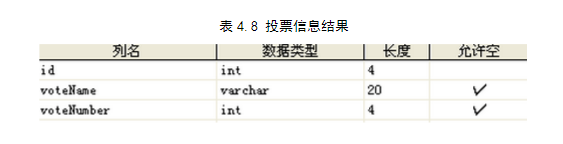
 tb\_photo↓

tb\_aritcle ↓

tb\_aritcleType↓

tb\_consumer↓tb\_discuss↓

tb\_friend↓

tb\_restore↓tb\_vote↓

5 公共模块设计

在开发过程中经常用到一些公共类，如数据库连接及操作类和字符串处理类，因此，在开发系统前首先应该编写这些公共类。下面将具体介绍博客网中所需要的公共类的编写过程。

5.1数据库连接及操作类的编写

本网站使用SQL Server 2000数据库系统。安装完成SQL Server 2000 JDBC Driver就可以编写与数据库连接的类。JDBConnection.java中不仅提供了数据库的连接，还有根据数据库获取的Statement 和 ResultSet 等，   com.my.tool.JDBConnection类封装了关于数据库的各项操作。代码如下：

public class JDBConnection {private final String url = "jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=db\_BlodMay"; //设置数据库URL驱动和数据库连接

 private final String userName = "sa";//设置连接的用户名     private final String password = "sa";//设置登陆密码

 private Connection con = null;//设置数据库连接类的对象

通过静态方法家在数据库驱动，方法代码如下：

 static {         try {Class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver").newInstance();          } catch (Exception ex) {

            System.out.println("数据库加载失败");//在控制台中输入异常信息         }

} }

创建数据库连接，方法代码如下：

  public boolean creatConnection() {

        try {

            con = DriverManager.getConnection(url, userName, password);//取得数据库连接

            con.setAutoCommit(true);//设置事务的自动提交的功能

        } catch (SQLException e) {

            System.out.println(e.getMessage());

            System.out.println("creatConnectionError!");         }

        return true;

}

对数据库的增加、修改和删除的操作，方法代码如下：

public boolean executeUpdate(String sql) {          if (con == null) {

            creatConnection();//取得数据库连接操作         }

        try {

            Statement stmt = con.createStatement();//创建声明连接

             int iCount = stmt.executeUpdate(sql);//执行SQL语句

             System.out.println("操作成功，所影响的记录数为" + String.valueOf(iCount));

      return true;//如果执行成功，则通过关键字return返回true         } catch (SQLException e) {

            System.out.println(e.getMessage());

      return false;//如果执行失败，则通过关键字return返回false         }

} 对数据库的查询操纵，方法代码如下：

public ResultSet executeQuery(String sql) {ResultSet rs;         try {

            if (con == null) {

                creatConnection();//取得数据库连接操作             }

            Statement stmt = con.createStatement();//执行声明对象连接             try {

                rs = stmt.executeQuery(sql);//执行查询的SQL语句             } catch (SQLException e) {

                System.out.println(e.getMessage());                 return null;

            }

5.2字符串处理类的编写

在使用JSP开发网站经常要过滤一些特殊字符或转换一些字符，这些功能通常需要编写一个独立的类来完成，本系统中通过以下代码实现字符的过滤和转换：

 public  static String  toChinese(String strvalue) {            try {

               if (strvalue == null) {

                   return "";//如果参数strvalue的值为空，则返回空对象                } else {

                   strvalue = new

String(strvalue.getBytes("ISO8859\_1"), "GBK");//将编码转换成GBK的格式

                   return strvalue;//返回转码内容                }

           } catch (Exception e) {

               return "";//返回查询的ResultSet结果集            }

}

在JSP中输出中文时要考虑中文的乱码问题，本实例中是利用String类的getByte()方法来解决这一问题。

5.3系统配置

本网站采用Servlet和JavaBean结合技术开发的。JavaBean技术实现对数据库的操作不需要在XML文件中进行任何的配置，而正常运行Servlet程序还需要进行适当的配置，配置文件为web.xml。下面将通过本实例详细的介绍在web.xml文件中Servlet的配置。关键代码如下：

<?xml version="1.0" encoding="UTF-8"?> <web-app version="2.4"

 xmlns="http://java.sun.com/xml/ns/j2ee"

 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee   http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd">   <servlet>

    <description>This is the description of my J2EE component</description>

    <display-name>This is the display name of my J2EE component</display-name>

    <servlet-name>ConsumerServlet</servlet-name>

    <servlet-class>com.wy.webiter.ConsumerServlet</servlet-class>   </servlet>   <servlet>

    <description>This is the description of my J2EE component</description>

    <display-name>This is the display name of my J2EE component</display-name>

    <servlet-name>FriendServlet</servlet-name>

    <servlet-class>com.wy.webiter.FriendServlet</servlet-class>   </servlet>   <servlet>

    <description>This is the description of my J2EE component</description>

    <display-name>This is the display name of my J2EE component</display-name>

    <servlet-name>VoteServlet</servlet-name>

    <servlet-class>com.wy.webiter.VoteServlet</servlet-class>   </servlet>

6 前台页面设计

6.1前台首页概述

在博客网的前台首页设计中，首先管理员发布的最新文章展现给用户，然后再提供文章的分类，显示博主的信息，最新公告及博客网导航区功能等功能。



6.2前台首页技术分析

如图6.2所示首页中的显示博主的基本信息，显示登录用户信息，显示博客的主要功能，显示博主的朋友圈及版权信息等功能，并不是仅存于在首页中，其他功能模块的子页面中也需要包括这些部分。因此，可以将这几个部分分别保存于单独的文件中，这样在需要放置相应功能时只需包含这些文件即可，如下图所示。

在JSP页面中包含文件有两种方法：一种是应用<%@include%>指令实现，另一种是应用<jsp:include>动作元素实现。本系统使用的是<jsp:include>动作元素，该动作元素用于向当前页面中包含其他的文件，这个文件可以是动态文件也可以是静态文件。

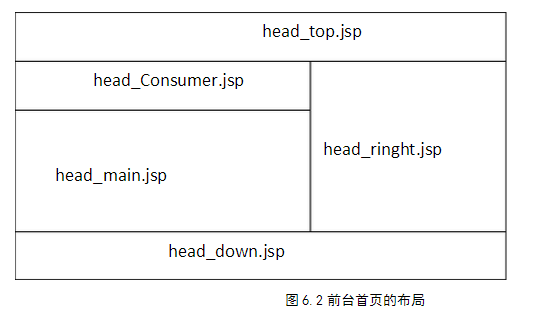
<%@include%>指令用来在JSP页面中包含另一个文件。包含的过程是静态的，即在指定文件属性值时，只能是一个包含相对路径的文件名，而不能是一个变量，也不可以在所指定的文件后面添加任何参数。其语法格式如下：

<%@include file=”fileName”%>

<jsp:include>动作元素可以指定加载一个静态或动态的文件，但运行结果不同。如果指定为静态文件，那么这种指定仅仅是把指定的文件内容加到JSP文件中去，则这个文件不被编译。如果是动态文件，那么这个文件，那么这个文件将会被编译器执行。由于在页面中包含查询模块时，只需要将文件内容添加到指定的JSP页面中即可，所以此处可以使用加载静态文件的方法包含文件。应用<jsp:include>动作元素加载静态文件的语法格式如下：

<jsp:include page=”{relativeURL | <%=expression%>}”flush=”true”/> 使用<%@include%>指令和<jsp:include>动作元素包含文件的区别是：使用<%@include%>指令包含的页面，是在编译阶段将该页面的代码插入到了主页面的代码中，最终包含页面与被包含页面生成一个文件。因此，如果被包含页面的内容有改动，需重新编译该文件。而使用<jsp:include>动作元素包含的页面可以是动态改变的，它是在JSP文件运行过程中被确定的，程序执行的是两个不同的页面，即在主页面中声明的变量，在被包含的页面中是不可见的。由此可见，当被包含的JSP页面中包含动态代码时，为了不和主页面中的代码相冲突，需要使用<jsp:include>动作元素包含文件。应用<jsp:include>动作元素包含查询页面的代码如下：

  <jsp:include page=”search.jsp” flush=”true”/>。



6.3 前台首页的布局

 应用<jsp:include>动作元素包含文件的方法进行前台首页布局的代码如下： <%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.\*" errorPage="" %>

<jsp:include page="head\_top.jsp" flush="true" />

<table width="800" border="0" align="center" cellpadding="0" cellspacing="0">   <tr>

    <td width="74"><img src="images/head\_06.jpg" width="74" height="846"></td>     <td height="846" valign="top" background="images/head\_07.jpg">  <!--登录用户部分-->

 <br><jsp:include page="head\_Consumer.jsp" flush="true" />&nbsp;

<td width="10" background="images/head\_07.jpg"><img src="images/head\_08.jpg" width="13" height="846"></td>

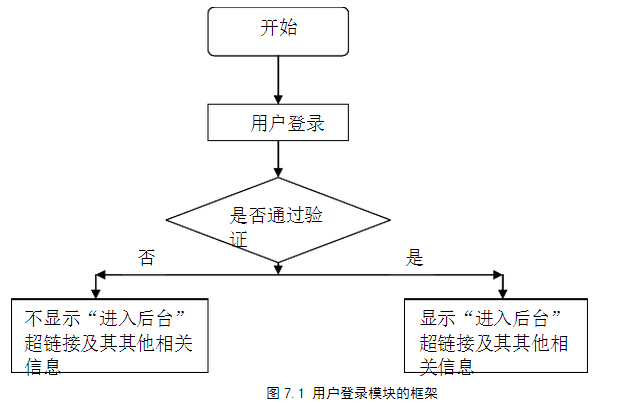
<td width="184" valign="top">

<jsp:include page="head\_right.jsp" flush="true" /> <jsp:include page="head\_down.jsp" flush="true" />

7 用户登录模块设计

7.1 用户登录模块概述

用户登录模块具有用户登录功能。博客网运行第一个页面便是用户登录页面，用户登录后，可以查看博主所发表的文章。在用户登录页面中，在用户名和密码文本框中输入用户名和密码后，单击登录按钮，系统验证成功后，用户将以会员的身份进入博客网首页，之后在网站中可以随意查看博主发表的文章。如果登录的用户是博主本人，在博主信息显示区域中，则出现“进入后台”超链接，博主可以直接单击此超连接，进入博客网的后台进行管理。用户登录模块的框架如下图所示。



7.2 用户登录模块技术分析

在用户登录模块中，涉及的数据表是用户信息表（tb\_sonsumer），在用户信息表中保存着用户名和密码等信息，根据这些信息来创建用户登录模块的form，名称为Consumer。

Servlet的核心在于它的HttpServlet，一般在Servlet里面做页面的逻辑跳转工作。用户登录模块的Servlet实现类继承了HttpServlet类，在该类中首先设置用户登录模块的ConsumerDao类的对象，该对象在每个具体的方法中实例化，doGet()在Servlet中自动被调用。该方法本身没有具体事务，它是根据通过HttpServlet的getParameter（）方法获取的method 参数值执行相应的方法的。

    在web.xml文件中配置用户登录的Servlet的具体代码如下： <servlet>

    <description>This is the description of my J2EE component</description>     <display-name>This is the display name of my J2EE component</display-name>     <servlet-name>ConsumerServlet</servlet-name>

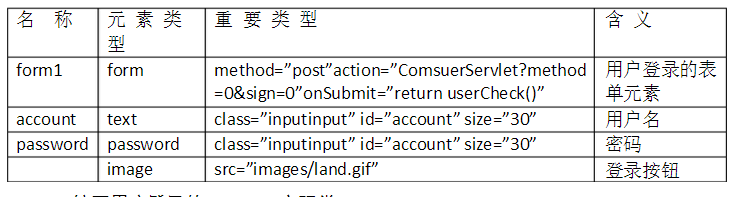
    <servlet-class>com.wy.webiter.ConsumerServlet</servlet-class>   </servlet>

7.3 用户登录模块的实现过程

用户登录模块是博客网最先使用的模块，是系统的入口。在用户登录页面中，用户可以通过输入正确的用户名和密码进入到系统，当用户没有输入用户名和密码时，系统会通过JavaScript脚步进行判断，并予以提示信息。

7.3.1 设计用户登录页面

用户登录页面主要用于收集用户输入的信息及通过自定义JavaScript函数判断输入信息是否为空，该页面所涉及的表单元素如下图所示。



7.3.2 编写用户登录的Servlet实现类

在用户登录页面的用户名和密码文本框中输入正确的用户名和密码后，单击登录按钮，网页会访问一个URL，即ConsumerServlet?method=0&sign=0。从该URL地址中可以知道用户登录模块所涉及的method参数值为0，也就是当method=0时，会调用验证用户身份的方法check-Consumer（）。

在验证用户身份的方法check-Consumer（）中，首先需要将用户名的表单信息赋值给新的String类型对象，将这个对象做参数，执行ConsumerDao类中的getConsumerForm()方法并将该方法的返回值赋给ConsumerForm类的对象，判断这个对象是否为空值，如果该对象为空值，则说明该用户输入的用户名不存在，如果该对象不为空值，则将这个对象中的密码值取出与密码表单数据比较，如果比较值为true，则说明用户输入的验证信息正确，如果比较值为false，则说明用户输入的验证信息不正确。

7.3.3 编写用户登录的ConsumerDao类的方法

用户登录使用的ConsumerDao类的方法是getConsumerForm()方法。在

getConsumerForm()中，将以用户名表单信息为参数，以用户信息表中的用户名为条件，自行查询的SQL语句，并将查询结果通过return关键字返回，如果没有查询结果，则返回空对象。

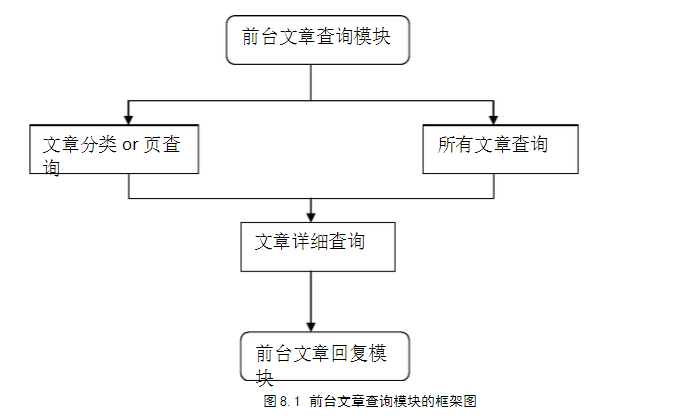
7.3.4 判断登录用户是否为博主

普通用户身份验证成功后，在博客网的首页将显示该用户的信息；如果博主登录后，则在博主显示区域中将显示“进入后台”超链接。当用户登录成功后，用户对应的form信息将保存在session中，程序员将根据保存在session中的form信息和用户登录级别信息（manageLevel）来判断登录的用户是否为博主。

8 文章查询模块

8.1文章查询模块概述

文章查询模块的主要功能时用户在浏览系统时查看发表的文章列表及文章详细内容，包括文章的题目、发表时间、该文章的浏览次数等，并可以针对发表文章进行恢复。这是博客最具代表性的特点之一。它给博主和用户提供展示文笔的平台，增加博主与用户、用户与用户之间的交流，能够使博客系统内的气氛更加活跃，内容更加丰富。前台文章查询模块主要包括所有文章查询、文章分类查询、文章内容详细查询及文章回复功能。前台文章查询模块如下图所示。



8.2 文章分类查询实现过程

在本系统中，前台文章查询模块主要应用的技术是JavaBean技术，应用<jsp:useBean>动作标签创建并使用一个JavaBean。<jsp:useBean>执行方式非常灵活，如果执行到<jsp:useBean>，将去寻找是否存在相同ID和scope。

当用户登录后，在博客网的首页右侧操作区中单击文章超链接，网页会自动弹出一个URL，即head\_Article.jsp,将分类显示文章的信息。

在前台分类显示文章信息页面中，首先通过JavaBean定义分类操作的类

（MyPagination），之后调用ArticleDao类中的queryArticle（）方法查询所有的文章信息，最后将查询的结果显示在页面中。

通过JavaBean技术调用ArticleDao类中的queryArticle（）方法实现所有文章查询的功能。在queryArticle（）实现方法中，typeId为其参数，代表文章类型的标号，当typeId属性为null时，将执行查询全部文章信息的SQL语句；当typeId的属性部位null时，将执行按文章类型查询的SQL语句。

8.3 文章详细查询的实现过程

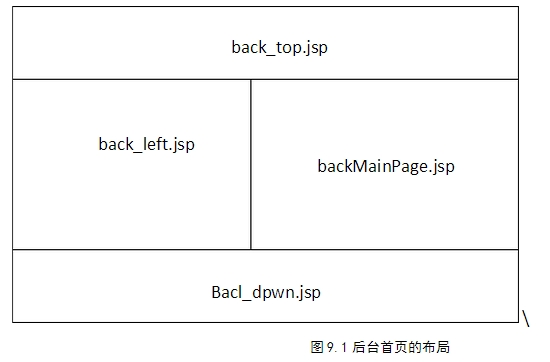
主页面的显示文章内容仅是其中的一部分，如果用户查询文章的全部信息，可以单击相应的阅读超链接，网页会自动淡出一个URL，页面为head\_ArticleForm.jsp?id=1。其中，id属性为文章的自动编号，这个编号是唯一的，因此查询文章信息也是唯一的。head\_ArticleForm.jsp页将根据这个id值显示所对应的文章信息。

首先通过request对像中的getParameter（）方法获取要查询的文章编号，之后设置<jsp:useBean>动作标签，该标签主要设定ArticleDao类实例对象，最后将执行ArticleDao类中的queryArticleForm（）方法，实现文章信息的查询功能。

9 后台设计

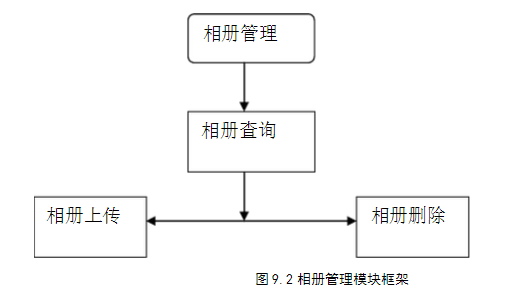
9.1 后面页面设计

在后台首页中的显示后台页面旗帜、左侧功能区及版权信息等功能，并不是仅存在于首页中，其他功能模块的子页面中也需要包括这些部分。因此，可以将这几个部分分别保存在单独的文件中，这样在需要放置相应功能时只需包含这些文件即可。如图9.1所示。



9.2后台相册设计

相册模块的主要功能时实现对相片上传、查询和删除操作，可以使博主随时更新个人博客系统内的相册文件，便于对相册文件的管理，以吸引更多的浏览者。相册管理模块的框架如图9.2所示。



在相册管理模块中，设计的数据表式相册信息表（tb\_photo），在相册信息模块中保存相册存放的服务器的路径及相册标题等信息，根据这些信息来创建相册管理模块的ActionForm类，名称PhotoForm。

相册管理模块对应的Servlet实现类的名称为PhotoServlet，其中，doGet()和doPost()方法本身没有具体的事务，而是通过HttpServletRequest的getParameter()方法获得method参数值执行相应的方法的。

**总结**

本网站系统主要使用Jsp的表现形式，结合JavaBean+Servlet技术来实现，这也恰恰说明了JavaBean+Servlet在开发Web应用程序中的优势，它充分体现了MVC的设计思想，将表示层、业务层和数据层分离，并且通过Servlet控制着这个页面的流向，使得整个系统设计思路比较清晰。

回顾整个系统中的技巧及其难点。一是当然为了保护好自己的系统，我的页面内容禁止复制。二是当在框架中设置超链接时，如果不加任何设置，则超链接页面会在当前框架页内打开，但是在某种情况下，超链接页面却需要在当前窗口中（承载当前框架页的窗口）。这里需要用一段相当难度的代码实现（当然是针对我的技术而言）。