

电子商务网站设计 指导书

电子商务专业

2013 年 6 月

第一部分 电子商务网站课程设计概述

应用网站课程设计涵盖了数据库应用系统的开发和 Web 程序设计 2 方面的内容。“Web 程序设计”强调从总体结构、系统分析这一角度来研究计算机系统。任何一个应用网站的开发都离不开后台数据库的支持，所以应用网站课程设计将网页制作、Web 应用程序编制和数据库应用有效结合起来，开发一个数据库技术支持的 WEB 应用系统。应用网站课程设计的目的和要求如下：

一、课程设计目的和意义

1. 加深对讲授内容的理解

WEB 程序设计理论课强调实用性，注重学生动手能力的培养。本课程设计将有助于加深对 Web 程序设计的基本概念、基本原理、设计原则和数据库操纵方法的理解，同时，通过大量实践环节学习和掌握有关数据库技术的基本理论、设计方法在应用网站系统开发的具体运用。

2. 通过应用网站课程设计，掌握 web 应用系统设计与开发的方法，培养自主开发能力。

应用网站课程设计侧重于理论应用、系统设计和程序开发过程。结合实际需要开发一个特定的“网络广告系统”，使学生能够运用数据库的基本知识设计数据库，掌握面向对象程序设计和分析的基本思想和基本方法，能够自主设计并实现小型的软件，最终具备一定的自主开发能力，可以自行开发小型的数据库应用程序。使同学能够主动查阅与数据库设计、Web 应用程序开发的相关资料，具备文献检索能力。进而能够与国内外 IT 行业对人才的需求接轨，为后面的毕业设计和将来能够较好地适应社会需求打下基础。

3. 培养自学以及主动解决问题的能力

通过本次设计，使同学能够主动查阅与应用网站相关资料，掌握一些课堂上老师未曾教授的知识，从而达到培养学生自学以及主动解决问题的能力目的，为后面的毕业设计打下坚实的基础。

二、课程设计实施步骤

同学们可按下列步骤完成所选题目的设计并写出设计报告。

第一步：问题分析。

在对所选题目进行调研的基础上，明确该选题要做什么。依据调查结果，进一

步分析和表达用户的需求。绘制数据流程图：

(1)基本符号

(2)数据流程图绘制方法：自顶向下，分层绘制。

(3)数据流程图绘制规则

①每张数据流程图须从左往右绘制，即从产生数据的外部实体开始到使用数据的外部实体结束。

②对含义明显的数据流，其名称可以省略。

③尽量避免数据流的交叉。

④对于需在两个设备上进行的处理，应避免直接相连。可以在它们之间加一个数据存储。

⑤如果一个外部实体提供给某一处理的数据流过多，可将它们合并成一个综合的数据流。

⑥下层图中的数据流应与上层图中的数据流守恒。

⑦对于大而复杂的系统，其图中的各元素应加以编号。通常在编号之首页以字母，用以表示不同的元素，可以用 P 表示处理，用 D 表示数据流，用 F 表示数据存储，用 S 表示外部实体。

第二步：数据库设计与实现。包括：

数据库的概念结构（E-R）图：

（1）画出系统各部分（子系统）E-R 模型图

（2）消除冲突和冗余，合并各部分 E-R 模型图，形成总体 E-R 模型图

（3）若系统较简单可直接画出系统总体 E-R 模型图；

逻辑与物理结构设计：将 E-R 图转换为关系模型，及设计数据库中的表、视图（如果使用）、存储过程（如果使用）的结构和定义（可以用 SQL 脚本提供）；

设计系统查询功能及要求，写出主要的查询 SQL 语句。

第三步：系统功能设计与实现，同时对设计结果进行评价与总结。对设计结果的合理性、规范程度和实际运行的结果以报告的形式进行提交。

四、课程设计要求

本次课程设计的目标是运用数据库和 Web 程序设计知识完成一个网站新闻管理系统的设计与开发工作。为了充分发挥学生的主观能动性，课程设计不指定开发环境和开发工具，学生可以任意选择自己喜欢的编程工具和数据库系统（建议使用开发平台 J2EE，数据库采用 My SQL）。课程设计结束后，学生以文档的形式提交课程设计成果，软件要实际上机检查，要求具有一定的实用性。

设计报告具体要求如下：

- (1) 问题描述。包括此问题的理论和实际两个方面。
- (2) 基本思路及关键问题的解决方法：.....
- (3) 分析设计文档：系统流程图，数据库设计（E-R 图），功能模块图，系统架构图，类的设计，界面设计，主要功能的实现方法.....
- (4) 测试报告：调试过程中出现的问题及相应解决方法:.....

第二部分 课程设计指导

一、项目开发的一般流程

1. 需求确定

通过各种手段（头脑风暴、会议、询问、原型——界面原型和业务原型）确定系统的功能和性能。

2. 分析与设计

1) 架构分析与设计

2) 业务逻辑分析

3) 业务逻辑设计

4) 界面设计

3. 开发环境搭建

4. 开发-测试-开发-测试（螺旋递增式开发）

5. 文档编写

二、需求分析

1. 问题的提出：

传统的网站管理方式有两种：一是静态 HTML 页面，更新信息时需要重新制作页面，然后上传页面并修改相应的链接，这种方式因为效率太低已不多用；二是基于 J2EE 和脚本语言，将动态网页和数据库结合，通过应用程序来处理程序，这是目前较为流行的做法。网络广告系统充分发挥网络的优势，实现了网络广告的动态管理，使得对网络广告的管理更加方便、及时、安全，降低了升级和维护的难度，提高了工作效率。

2. 系统需要解决的问题及功能：

根据实现业务的不同进行功能分析。

三、系统分析与设计

1. 架构分析与设计

逻辑架构：3 层架构、n 层架构；MVC 等；

物理架构：Web 服务器的分布；数据库服务器的分布；

技术解决方案的确定：Java；Open Source；

2. 业务逻辑分析

(Unified Process, Use Case)

-根据需求分析业务逻辑

-他们会使用本系统做什么

-通常他们使用本系统的步骤是什么样的

-会有哪些明显的类来支撑本系统的运行

-会有哪些不同的提示返回给用户

-...

本阶段与需求的确定密切相关，通常在确定需求的时候会进行相关的分析

3. 业务逻辑设计

-根据需求的分析来确定具体的类

-确定类的属性

-确定类的接口（方法）

-确定类之间的关系

-确定用户操作流程在设计上的反映

-进行数据库的设计（不同的项目步骤可能不尽相同）

-...（设计类...，设计数据库...）

4. 界面设计

-设计系统的界面风格：颜色，style

-设计系统的具体“模拟”界面

能够从头走到尾：方便进行需求的确定；方便程序员的开发

（界面设计：注意命名机制，包括文件名、字段名等；风格要统一，最好画图表示出来，不用文字）

四、开发环境搭建

-开发工具的确定

-配置管理工具的确定

- 测试工具的确定
- 文件服务器/配置服务器的确定
- ...

五、开发-测试-开发-测试

- 按照设计进行开发
- 迅速开发原型
- 进行迭代开发
- 提早进行测试
- 单元测试
- 黑盒测试
- 性能测试
- 应用性测试

(确定先写什么, 再写什么, 然后。。。。最后。。。。; 主要的先写, 基础的先写)

第三部分 选题描述

这一部分讲述如何用 JSP 技术建立一个简单的网络广告系统, 其目的是掌握一般 Web 应用中常用基本模块的开发方法。JSP 引擎为 Tomcat6.0; 系统采用 MVC 模式实现各个模块功能; 数据库连接操作使用加载纯 Java 驱动程序方式, 本系统采用的是 MySQL 数据库。

一、系统模块构成

- 1.会员注册: 新会员填写表单, 包括会员名、E-mail 地址等信息。如果输入的会员名已经被其他用户注册使用, 系统提示新用户更改自己的会员名。
- 2.会员登录: 输入会员名、密码。如果用户输入的会员名或密码有错误, 系统将显示错误信息。
- 3.上传产品照片: 如果登录成功, 用户可以使用该模块上传产品照片。
- 4.浏览会员的广告: 成功登录的会员可以分页浏览其他会员所做的广告, 如果用户直接进入该页面或没有成功登录就进入该页面, 将被链接到“会员登录”页面。
- 5.修改密码: 成功登录的会员可以在该页面修改自己的登录密码, 如果用户直接进入该页面或没有成功登录就进入该页面, 将被链接到“会员登录”页面。

6.修改注册信息：成功登录的会员可以在该页面修改自己的注册信息，比如联系电话、通信地址等，如果用户直接进入该页面或没有成功登录就进入该页面，将被链接到“会员登录”页面。

7.退出登录：成功登录的用户可以使用该模块退出登录。

二、数据库设计

系统采用加载纯 Java 数据库驱动程序方式访问数据库。使用 MySQL5.5 建立一个数据库 Friend,该库有一个 member 表。会员的注册信息存入 member 表中,member 表的主键是 logname, 各个字段值的说明如下：

logname: 存储会员登录名字。

password: 存储会员登录密码。

advertiseTitle: 存储会员的广告标题。

phone: 存储会员的电话。

email: 存储会员的 E-mail 地址。

message: 存储会员的广告词。

pic: 存储会员的广告产品的照片的名字。

member 表的详细结构设计如图所示。

	列名	数据类型	长度	允许空
?	logname	char	10	
	password	char	10	
	advertiseTitle	nvarchar	50	
	phone	nvarchar	50	✓
	email	nvarchar	50	✓
	message	nvarchar	50	✓
	pic	nvarchar	50	✓

三、系统管理

本系统使用的 Web 服务目录是为 ch10, ch10 是在 Tomcat 安装目录的 webapps 目录下建立的 Web 服务目录。现在需要在当前 Web 服务目录下建立如下的目录结构：

ch10\WEB-INF\classes

然后根据 servlet 的包名，在 classes 下再建立相应的子目录，比如 Servlet 类的包名为 myservlet.control, 那么在 classes 下建立子目录：\myservlet\control；如果

Javabeans 类的包名为 mybean.data, 那么在 classes 下建立子目录: \mybean\data。为了让 Tomcat 服务器起用上述目录, 必须重新启动 Tomcat 服务器。

1. 页面管理

本系统所用的 JSP 页面全部保存在 Web 服务目录 ch10 中。

所有的页面将包括一个导航条, 该导航条由注册、登录、上传广告照片、浏览会员的广告、修改密码、修改个人信息组成。为了便于维护, 其他页面通过使用 JSP 的 <%@include...%> 标记将导航条文件 head.txt 嵌入自己的页面, head.txt 保存在 ch10 中。head.txt 的内容如下:

```
<%@ page contentType="text/html;charset=GB2312" %>
<CENTER><Font size=5><P>网 络 广 告 系 统</Font></CENTER>
<table cellSpacing="1" cellPadding="1" width="560" align="center" border="0" >
  <tr valign="bottom">
    <td><A href="register.jsp"><font size=2>会员注册</font></A></td>
    <td><A href="login.jsp"><font size=2>会员登录</font></A></td>
    <td><A href="upload.jsp"><font size=2>上传广告照片</font></A></td>
    <td><A href="choiceLookType.jsp"><font size=2>浏览会员的广告</font></A></td>
    <td><A href="choiceModifyMess.jsp"><font size=2>修改注册信息</font></A></td>
    <td><A href="modifyPassword.jsp"><font size=2>修改密码</font></A></td>
    <td><A href="helpExitLogin"><font size=2>退出登录</font></A></td>
    <td><A href="index.jsp"><font size=2>返回主页</font></A></td>
  </tr>
</Font>
</table>
```

主页 index.jsp 由导航条、一句欢迎标语和一幅图片 welcome.jsp 保存在 ch10 中。用户可以通过在浏览器的地址栏中输入 “http://服务器 IP: 8080/” 访问该主页, 主页运行效果如图所示:



index.jsp:

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML>
<HEAD><% @ include file="head.txt" %></HEAD>
<body bgcolor=cyan>
<CENTER> <h1><font Size=4 color=red>欢迎您来这里做广告</font></h1>
</img>
</img>
</img>
</CENTER>
</BODY>
</HTML>
```

2. Javabeen 与 servlet 管理

本系统的 Javabeen 类的包名均为 mybean.data；Servlet 类的包名均为 myservlet.control。由于 Servlet 类中要使用 Javabeen，所以为了能顺利地编译 Servlet 类，首先将 Javabeen 类和 Servlet 类分别保存到 ch10\src\myservlet\control\mybean\data 和 ch10\src\myservlet\control 目录中，分别进入上述目录，编译 Javabeen 类和 Servlet 类。然后将编译通过的 Javabeen 类和 Servlet 类的字节码分别复制到 ch10\WEB-INF\classes\ mybean\data 和 ch10\WEB-INF\classes\myservlet\control 目录中。

3. 配置文件管理

本系统的 Servlet 类的包名均为 myservlet.control，需要配置 Web 服务目录的 web.xml 文件，根据使用的 Tomcat 安装目录及使用的 Web 服务目录，需将下面的 web.xml 文件保存到 webapps\ch10\WEB-INF 中。

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<web-app>
<servlet>
    <servlet-name>register</servlet-name>
    <servlet-class>myservlet.control.HandleRegister</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>register</servlet-name>
    <url-pattern>/helpRegister</url-pattern>
</servlet-mapping>
```

```
<servlet>
    <servlet-name>login</servlet-name>
    <servlet-class>myservlet.control.HandleLogin</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>login</servlet-name>
    <url-pattern>/helpLogin</url-pattern>
</servlet-mapping>
<servlet>
    <servlet-name>upload</servlet-name>
    <servlet-class>myservlet.control.HandleUpload</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>upload</servlet-name>
    <url-pattern>/helpUpload</url-pattern>
</servlet-mapping>
<servlet>
    <servlet-name>lookRecord</servlet-name>
    <servlet-class>myservlet.control.HandleDatabase</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>lookRecord</servlet-name>
    <url-pattern>/helpShowMember</url-pattern>
</servlet-mapping>
<servlet>
    <servlet-name>modifyPassword</servlet-name>
    <servlet-class>myservlet.control.HandlePassword</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>modifyPassword</servlet-name>
    <url-pattern>/helpModifyPassword</url-pattern>
</servlet-mapping>
<servlet>
    <servlet-name>getOldMess</servlet-name>
```

```
<servlet-class>myservlet.control.GetOldMess</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>getOldMess</servlet-name>
  <url-pattern>/helpGetOldMess</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>modifyOldMess</servlet-name>
  <servlet-class>myservlet.control.HandleModifyMess</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>modifyOldMess</servlet-name>
  <url-pattern>/helpModifyMess</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>exit</servlet-name>
  <servlet-class>myservlet.control.HandleExit</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>exit</servlet-name>
  <url-pattern>/helpExitLogin</url-pattern>
</servlet-mapping>
<resource-ref>
  <description>DB Connection</description>
  • <res-ref-name>jdbc/mysql</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
</resource-ref>
</web-app>
```

四、会员注册

当新会员注册时，该模块要求用户必须输入会员名、密码信息，否则不允许注册。用户的注册信息被存入数据库的 member 表中。该模块的模型 Javabean 描述用户的注册信息；该模块视图部分由两个 JSP 页面构成，

一个 JSP 页面负责提交用户的注册信息到控制器, 另一个 JSP 页面负责显示注册是否成功的信息; 该模块的控制器 servlet 负责将视图提交的信息写入数据库的 member 表中, 并负责更新视图。

1.模型(Javabean)

下列 Javabean 用来描述用户注册信息。

Register. java

```
package mybean.data;

public class Register {

    String logname="",password="",advertiseTitle="",
        email="", phone="", message="";

    String backNews;

    public void setLogname(String name)
    {   logname=name;
    }

    public String getLogname()
    {   return logname;
    }

    public void setAdvertiseTitle(String s)
    {   advertiseTitle=s;
    }

    public String getAdvertiseTitle()
    {   return advertiseTitle;
    }

    public void setPassword(String pw)
    {   password=pw;
    }

    public String getPassword()
    {   return password;
    }

    public void setEmail(String em)
    {   email=em;
    }

    public String getEmail()
    {   return email;
    }
}
```

```

    }
    public void setPhone(String ph)
    {   phone=ph;
    }
    public String getPhone()
    {   return phone;
    }
    public String getMessage()
    {   return message;
    }
    public void setMessage(String m)
    {   message=m;
    }
    public String getBackNews()
    { return backNews;
    }
    public void setBackNews(String s)
    { backNews=s;
    }
}

```

2. 视图(JSP 页面)

本模块视图有两个 JSP 页面 Register.jsp 和 showRegisterMess.jsp。Register.jsp 页面负责提供输入注册信息界面；showRegisterMess.jsp 负责显示注册反馈信息，比如注册是否成功等。

register.jsp（效果如图所示）

```

<%@ page contentType="text/html;charset=GB2312" %>
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<BODY bgcolor=cyan><CENTER>
<FORM action="helpRegister" name=form>
<table>
    输入您的信息，会员名字必须由字母和数字组成，带*号项必须填写。
    <tr><td>会员名称:</td><td><Input type=text name="logname" >*</td></tr>
    <tr><td>设置密码:</td><td><Input type=password name="password">*</td></tr>
    <tr><td>广告标题:</td><td><Input type=text  name="advertiseTitle">*</td></tr>

```

```

<tr><td>电子邮件:</td><td><Input type=text name="email"></td></tr>
<tr><td>联系电话:</td><td><Input type=text name="phone"></td></tr>
</table>
<table>
<tr><td>输入您的广告词: </td></tr>
<tr>
<td><TextArea name="message" Rows="6" Cols="30"></TextArea></td>
</tr>
<tr><td><Input type=submit name="g" value="提交"></td> </tr>
</table>
</Form></CENTER>
</Body>
</HTML>

```

网 络 广 告 系 统

[会员注册](#) [会员登录](#) [上传广告照片](#) [浏览会员的广告](#) [修改注册信息](#) [修改密码](#) [退出登录](#) [返回主页](#)

输入您的信息，会员名字必须由字母和数字组成，带*号项必须填写。

会员名称: *
 设置密码: *
 广告标题: *
 电子邮件:
 联系电话:
 输入您的广告词:

showRegisterMess.jsp, 效果图如下图所示

```

<% @ page contentType="text/html;charset=GB2312" %>
<% @ page import="mybean.data.*"%>
<jsp:useBean id="register" type="mybean.data.Register" scope="request" />
<HTML><HEAD><% @ include file="head.txt" %></HEAD>

```

```
<HTML><BODY bgcolor=cyan >
<CENTER>
  <Font size=4 color=blue >
    <BR> <jsp:getProperty name="register" property="backNews" />
  </Font>
<table>
  <tr><td>注册的会员名称:</td><td><jsp:getProperty name="register" property="logname" /></td></tr>
  <tr><td> 注 册 的 广 告 标 题 :</td><td><jsp:getProperty name="register"
property="advertiseTitle"/></td></tr>
  <tr><td>注册的电子邮件:</td><td><jsp:getProperty name="register" property="email" /></td></tr>
  <tr><td>注册的联系电话:</td><td><jsp:getProperty name="register" property="phone" /></td></tr>
</table>
<table><tr><td>您的广告词: </td></tr>
  <tr><td><TextArea name="message" Rows="6" Cols="30">
    <jsp:getProperty name="register" property="message" />
  </TextArea>
  </td>
</tr>
</table>
</CENTER></BODY></HTML>
```

网 络 广 告 系 统

[会员注册](#) [会员登录](#) [上传广告照片](#) [浏览会员的广告](#) [修改注册信息](#) [修改密码](#) [退出登录](#) [返回主页](#)

注册成功

注册的会员名称: xxiaaaa
 注册的广告标题: erqerqwr
 注册的电子邮件:
 注册的联系电话:
 您的广告词:

3. 控制器(servlet)

控制器 servlet 对象的名字是 register(见前文给出的 web. xml 配置文件)。控制

器 register，负责连接数据库，将用户提交的信息写入到 member 表，并将用户 showRegisterMess.jsp 页面查看注册反馈信息。

HandleRegister.java

```
package myservlet.control;
import mybean.data.*;
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.sql.DataSource;
//import javax.naming.*;
import javax.naming.Context;
import javax.naming.InitialContext;
//import javax.naming.NamingException;

public class HandleRegister extends HttpServlet
{
    public void init(ServletConfig config) throws ServletException
    {
        super.init(config);
    }

    public String handleString(String s)
    {
        try{ byte bb[]=s.getBytes("iso-8859-1");
            s=new String(bb);
        }
        catch(Exception ee){ }
        return s;
    }

    public void doPost(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException
    {
        DataSource ds = null;
        Context env = null;
        Connection con = null;
        PreparedStatement sql;           //PreparedStatement 继承于 Statement
        Register reg=new Register();     //定义一个新的 register 对象。
```



```
request.setAttribute("register",reg);           //给 request 增加了一个属性

String logname=request.getParameter("logname").trim(),
password=request.getParameter("password").trim(),
advertiseTitle=request.getParameter("advertiseTitle").trim(),
email=request.getParameter("email").trim(),
phone=request.getParameter("phone").trim(),
message=request.getParameter("message");    //从 request 中提取用户提交的信息

if(logname==null)
    logname="";
if(password==null)
    password="";
boolean isLD=true;
for(int i=0;i<logname.length();i++)           //for 循环用来判断 logname 中组成是不是字母和数字。这也是为什么上面 logname 为"NULL",就给赋值为 “ ” 的原因，null 代表指针所连接的值不存在，不存在是不能求长度的，会抛出无法解析的错误。
{
    char c=logname.charAt(i);
    if(!((c<='z'&&c>='a')||(c<='Z'&&c>='A')||(c<='9'&&c>='0'))
        isLD=false;
}
boolean boo=logname.length()>0&&password.length()>0&&isLD;    //定义 boo,当 logname 和 password 不为空，则 boo 为 1.

String backNews="";

try{
    System.out.println(System.getProperty("java.naming.factory.initial"));

    try{
        env = (Context) new InitialContext().lookup("java:comp/env");
        ds = (DataSource)env.lookup("jdbc/mysql");
        con = ds.getConnection();
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }

    String insertCondition="INSERT INTO member VALUES (?, ?, ?, ?, ?, ?);"
```



```
        sql=con.prepareStatement(insertCondition);

        System.out.println(logname);

        System.out.println(password);

        System.out.println(advertiseTitle);

        if(boo)

        { sql.setString(1,handleString(logname));

          sql.setString(2,handleString(password));

          sql.setString(3,handleString(advertiseTitle));

          sql.setString(4,phone);

          sql.setString(5,email);

          sql.setString(6,handleString(message));

          sql.setString(7,"public.jpg");

          System.out.println(insertCondition);

          System.out.println("boo"+boo);

          int m=sql.executeUpdate();

          if(m!=0)

          {   backNews="注册成功";

              reg.setBackNews(backNews);

              reg.setLogname(logname);

              reg.setPassword(handleString(password));

              reg.setAdvertiseTitle(handleString(advertiseTitle));

              reg.setEmail(handleString(email));

              reg.setPhone(phone);

              reg.setMessage(handleString(message));

          }

        }

        else

        {   backNews="信息填写不完整或名字中有非法字符";           //当 logname 信息不

对

              reg.setBackNews(backNews);

        }

        con.close();

    }
```

```
catch(SQLException exp)                                //发生错误则设置
backNews
{   backNews="该会员名已被使用, 请您更换名字"+exp;
    reg.setBackNews(backNews);
}

RequestDispatcher dispatcher= request.getRequestDispatcher("showRegisterMess.jsp");//转发
dispatcher.forward(request, response);
}

public void doGet(HttpServletRequest request,HttpServletResponse response)
                throws ServletException,IOException
{   doPost(request,response);                            //当采用 get 方式提交时, 则还是要交个 doPost
方法处理
}
}
```

五、会员登录

用户可在该模块输入自己的会员名和密码, 系统将对会员名和密码进行验证, 如果输入用户名或密码有错误, 将提示用户输入的用户名或密码不正确。

该模块的模型 **JavaBean** 描述用户登录的信息; 该模块视图部分由两个 **JSP** 页面构成, 一个 **JSP** 页面负责提交用户的登录信息到控制器, 另一个 **JSP** 页面负责显示登录是否成功的信息; 该模块的控制器 **servlet** 负责验证会员名和密码是否正确, 并负责更新视图。

1. 模型(JavaBean)

下列 **JavaBean** 的实例用来描述用户登录信息。

Login. java

```
public class Login
{   String   logname,
        password,
        backNews="";

    boolean success=false;

    public void setLogname(String name)
    {   logname=name;
    }
}
```

```
public String getLogname()
{
    return logname;
}

public void setPassword(String pw)
{
    password=pw;
}

public String getPassword()
{
    return password;
}

public String getBackNews()
{
    return backNews;
}

public void setBackNews(String s)
{
    backNews=s;
}

public void setSuccess(boolean b)
{
    success=b;
}

public boolean getSuccess()
{
    return success;
}
}
```

2. 视图(JSP 页面)

本模块视图有两个 JSP 页面 login.jsp 和 showLoginMess.jsp。login.jsp 页面负责提供输入登录信息界面；showLoginMess.jsp 负责显示登录反馈信息，比如登录是否成功等。

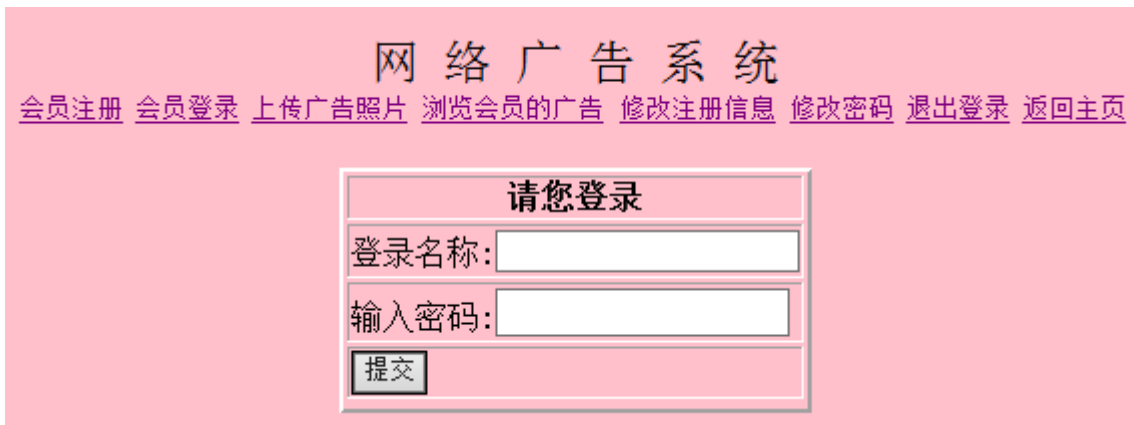
login.jsp(效果如图所示)

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<BODY bgcolor=pink>
<CENTER>
<FORM action="helpLogin" Method="post">
<table border=2>
```

```

<tr><th>请您登录</th></tr>
<tr><td>登录名称:<input type="text" name="logname"></td></tr>
<tr><td>输入密码:<input type="password" name="password"></td></tr>
<tr><td><input type="submit" name="g" value="提交"></td></tr>
</table>
</Form>
</CENTER>
</BODY></HTML>

```



showLoginMess.jsp

```

<% @ page contentType="text/html;charset=GB2312" %>
<% @ page import="mybean.data.Login"%>
<jsp:useBean id="login" type="mybean.data.Login" scope="session" />
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<BODY bgcolor=yellow >
<CENTER>
    <jsp:getProperty name="login" property="backNews" />
    <% if(login.getSuccess()==true)
    {
    %>    <BR>登录会员名称:<jsp:getProperty name="login" property="logname" />
    <% }
    else
    {
    %>    <BR>登录会员名称:<jsp:getProperty name="login" property="logname" />
        <BR>登录会员密码:<jsp:getProperty name="login" property="password" />
    <% }
    %>
</CENTER>
</BODY></HTML>

```

3. 控制器(servlet)

该 servlet 对象的名字是 login(见前文给出的 web.xml 配置文件)。控制器 login 负责连接数据库, 查询 member 表, 验证用户输入的会员名和密码是否在 member 表中, 并将用户转发到 showRegisterMess.jsp 页面查看登录反馈信息。

HandleLogin. java

```
package myservlet.control;
import mybean.data.*;
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HandleLogin extends HttpServlet
{
    public void init(ServletConfig config) throws ServletException
    {
        super.init(config);
        try {
            Class.forName("com.mysql.jdbc.Driver");
        }
        catch(Exception e){}
    }
    public String handleString(String s)
    {
        try {
            byte bb[]=s.getBytes("iso-8859-1");
            s=new String(bb);
        }
        catch(Exception ee){}
        return s;
    }
    public void doPost(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException
    {
        Connection con;
        PreparedStatement sql;
        Login loginBean=null;
        String backNews="";
        HttpSession session=request.getSession(true);
        try {
            loginBean=(Login)session.getAttribute("login");
            if(loginBean==null)
            {
                loginBean=new Login();
                session.setAttribute("login",loginBean);
            }
        }
        catch(Exception ee)
        {
            loginBean=new Login();
            session.setAttribute("login",loginBean);
        }
        String logname=request.getParameter("logname").trim(),
        password=request.getParameter("password").trim();
```




```
boolean ok=loginBean.getSuccess();
logname=handleString(logname);
password=handleString(password);
if(ok==true&&logname.equals(loginBean.getLogname()))
{   backNews=logname+"已经登录了";
    loginBean.setBackNews(backNews);
}
else
{   String
uri="jdbc:mysql://localhost:3306/advertisement?useUnicode=true&characterEncoding=utf-8";
    boolean boo=(logname.length()>0)&&(password.length()>0);
    try{ con=DriverManager.getConnection(uri,"root","123456");
        String condition="select * from member where logname=? and password=?";
        sql=con.prepareStatement(condition);
        if(boo)
        {   sql.setString(1,logname);
            sql.setString(2,password);
            ResultSet rs=sql.executeQuery();
            boolean m=rs.next();
            if(m==true)
            {   backNews="登录成功";
                loginBean.setBackNews(backNews);
                loginBean.setSuccess(true);
                loginBean.setLogname(logname);
            }
            else
            {   backNews="您输入的用户名不存在，或密码不般配";
                loginBean.setBackNews(backNews);
                loginBean.setSuccess(false);
                loginBean.setLogname(logname);
                loginBean.setPassword(password);
            }
        }
        con.close();
    }
    catch(SQLException exp)
    {   backNews="" +exp;
```

```

        loginBean.setBackNews(backNews);
        loginBean.setSuccess(false);
    }
}

RequestDispatcher dispatcher=request.getRequestDispatcher("showLoginMess.jsp");
dispatcher.forward(request, response);
}

public void doGet(HttpServletRequest request,HttpServletResponse response)
    throws ServletException,IOException
{
    doPost(request,response);
}
}

```

六、上传产品照片

用户可在该模块上传广告产品的照片。如果 member 中已经存有一幅照片，新上传的照片将替换原有的照片。用户在注册时，注册模块给会员的照片是默认的一幅图像 public.jpg。该模块的模型 Javabean 描述用户上传的照片文件的有关信息；该模块视图部分由两个 JSP 页面构成，一个 JSP 页面负责提交图像文件到控制器，另一个 JSP 页面负责显示上传操作是否成功的信息；该模块的控制器负责将图像文件上传到服务器、将图像文件的名称写入数据库的 member 表中，必要时还需删除用户曾上传的图像文件；该 servlet 还负责更新视图，使用户能看到上传操作的结果。另外，控制器能阻止未登录用户上传照片。

1. 模型(Javabean)

下列 Javabean 的实例用来描述上传文件的有关信息。

UploadFile. Java

```

package mybean.data;
public class UploadFile
{
    String fileName,savedFileName,
        backNews="";
    public void setFileName(String name)
    {
        fileName=name;
    }
    public String getFileName()
    {
        return fileName;
    }
    public void setSavedFileName(String name)
    {
        savedFileName=name;
    }
}

```

```

public String getSavedFileName()
{
    return savedFileName;
}

public String getBackNews()
{
    return backNews;
}

public void setBackNews(String s)
{
    backNews=s;
}
}

```

2. 视图(JSP 页面)

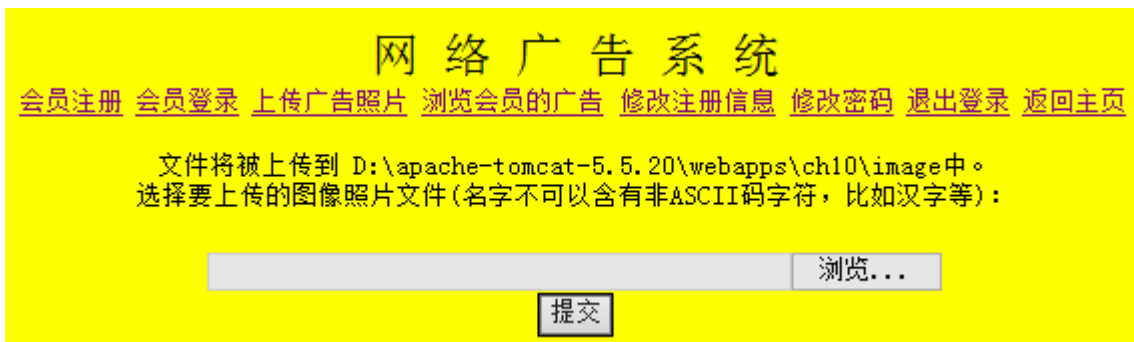
本模块视图有两个 JSP 页面 upload.jsp 和 showUploadMess.jsp。upload.jsp 页面负责提供上传文件的表单，showUploadMess.jsp 负责显示上传文件的反馈信息。

Upload.jsp(效果如图所示)

```

<% @ page contentType="text/html;charset=GB2312" %>
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<BODY bgcolor=yellow><Font size=2><CENTER>
<BR>文件将被上传到 D:\apache-tomcat-5.5.20\webapps\ch10\image中。
<BR>选择要上传的图像照片文件(名字不可以含有非ASCII码字符，比如汉字等):
<FORM action="helpUpload" method="post" ENCTYPE="multipart/form-data">
    <INPUT type=FILE name="fileName" size="40">
    <BR> <INPUT type="submit" name="g" value="提交">
</FORM></CENTER>
</font> </BODY></HTML>

```



showUploadMess. jsp(效果如图所示)

```

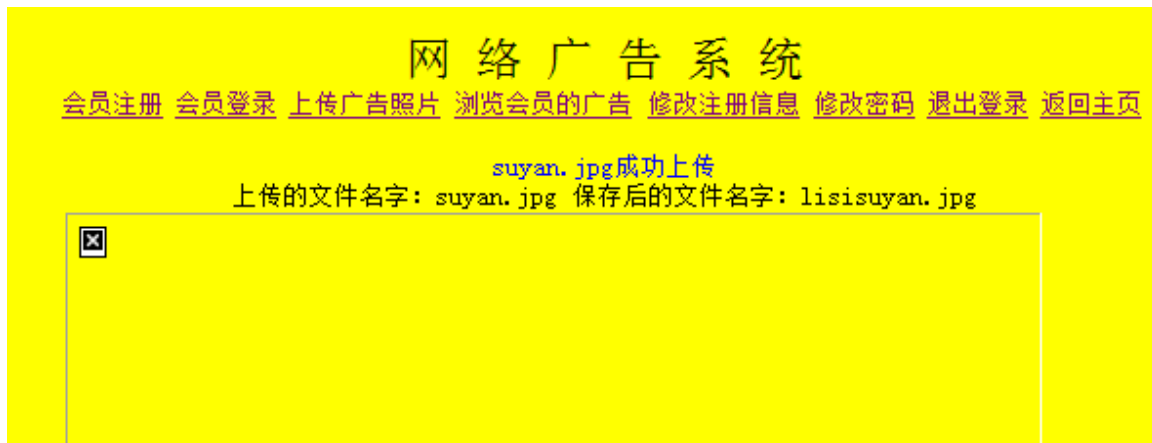
<% @ page contentType="text/html;charset=GB2312" %>
<% @ page import="mybean.data.*"%>
<jsp:useBean id="upFile" type="mybean.data.UploadFile" scope="request" />
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<HTML><BODY bgcolor=yellow >
<CENTER><Font size=2 color=blue >
    <BR> <jsp:getProperty name="upFile" property="backNews" />

```

```

</Font>
<BR><font size=2>上传的文件名字: <jsp:getProperty name="upFile" property="fileName" />
      保存后的文件名字: <jsp:getProperty name="upFile" property="savedFileName" />
<BR> <img src=image/<jsp:getProperty name="upFile" property="savedFileName" />
      width=500 height=500>图像效果
</img>
</FONT></CENTER>
</BODY></HTML>

```



3. 控制器(servlet)

该 servlet 对象的名字是 upload（见前文给出的 web.xml 配置文件）。upload 控制器负责检查用户是否是登录用户，如果用户没有登录，upload 控制器将把用户定向到登录页面 login.jsp；对于登录的用户，upload 控制器负责把用户提交的图像文件保存到当前 Web 服务目录的特定子目录 image 中，即保存到如下目录中：

\ch10\image

保存的图像文件名字是在用户上传的文件名字前面添加上用户的会员名。upload 控制器同时负责将保存的图像文件名存入 member 表，然后将用户转发到 showUploadMess.jsp 页面查看上传操作的反馈信息。

HandleUpload. Java

```

package myservlet.control;
import mybean.data.*;
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HandleUpload extends HttpServlet
{
    public void init(ServletConfig config) throws ServletException
    {
        super.init(config);
        try {
            Class.forName("com.mysql.jdbc.Driver");

```



```
    }  
    catch(Exception e){ }  
}  
  
public void doPost(HttpServletRequest request,HttpServletResponse response)  
    throws ServletException,IOException  
{ HttpSession session=request.getSession(true);  
  Login login=(Login)session.getAttribute("login"); //获取用户登录时的Javabean  
  boolean ok=true;  
  if(login==null)  
  { ok=false;  
    response.sendRedirect("login.jsp"); //重定向到登录页面  
  }  
  if(ok==true)  
  { String logname=login.getLogname();  
    uploadFileMethod(request,response,logname); //接受上传文件  
  }  
}  
  
public void uploadFileMethod(HttpServletRequest request,HttpServletResponse response,  
String logname) throws ServletException,IOException  
{ UploadFile upFile=new UploadFile();  
  String backNews="";  
  try{ HttpSession session=request.getSession(true);  
    request.setAttribute("upFile",upFile);  
    String tempFileName=(String)session.getId();  
    File f1=new File(tempFileName);  
    FileOutputStream o=new FileOutputStream(f1);  
    InputStream in=request.getInputStream();  
    byte b[]=new byte[10000];  
    int n;  
    while( (n=in.read(b))!=-1)  
    { o.write(b,0,n);  
    }  
    o.close();  
    in.close();  
    RandomAccessFile random=new RandomAccessFile(f1,"r");  
    int second=1; //读出f1的第2行，析取出上传文件的名字：  
    String secondLine=null;  
    while(second<=2)  
    { secondLine=random.readLine();  
      second++;  
    }  
    //获取第2行中目录符号\最后出现的位置：  
    int position=secondLine.lastIndexOf("\\");  
    //截取文件名:
```



```
String fileName=secondLine.substring(position+1,secondLine.length()-1);
byte cc[]=fileName.getBytes("ISO-8859-1");
fileName=new String(cc);
fileName=fileName.replaceAll(" ", "");
//文件是否由字母或数字组成判断名字
String checkedStr=fileName.substring(0,fileName.indexOf("."));
boolean isLetterOrDigit=true;
for(int i=0;i<checkedStr.length();i++)
{ char c=checkedStr.charAt(i);
  if(!((c<='z'&&c>='a')||(c<='Z'&&c>='A')||(c<='9'&&c>='0'))
    { isLetterOrDigit=false;
      break;
    }
}
if(isLetterOrDigit==false)
{ response.sendRedirect("upload.jsp"); //重定向到upload.jsp页面
}
```

//保存文件名是上传文件名加会员名为前缀:

```
String savedFileName=logname.concat(fileName);
random.seek(0);
long forthEndPosition=0; //获取第4行回车符号的位置
int forth=1;
while((n=random.readByte())!=-1&&(forth<=4))
{ if(n=='\n')
  { forthEndPosition=random.getFilePointer();
    forth++;
  }
}
//根据客户上传文件的名字, 将该文件存入磁盘:
File dir=new File("D:/javawork/workspace/adver/WebRoot/image");
dir.mkdir();
//首先删除用户曾上传过的图像文件:
File file[]=dir.listFiles();
for(int k=0;k<file.length;k++)
{ if(file[k].getName().startsWith(logname))
  file[k].delete();
}
File savingFile= new File(dir,savedFileName); //需要新保存的上传文件
RandomAccessFile random2=new RandomAccessFile(savingFile,"rw");
random.seek(random.length());
long endPosition=random.getFilePointer();
long mark=endPosition;
int j=1;
//确定出文件f1中包含客户上传的文件的最后位置, 即倒数第6行:
```



```
        while((mark>=0)&&(j<=6))
        {
            mark--;
            random.seek(mark);
            n=random.readByte();
            if(n=='\n')
            {
                endPosition=random.getFilePointer();
                j++;
            }
        }
        random.seek(forthEndPosition);
        long startPoint=random.getFilePointer();
        while(startPoint<endPosition-1)
        {
            n=random.readByte();
            random2.write(n);
            startPoint=random.getFilePointer();
        }
        random2.close();
        random.close();
        String
uri="jdbc:mysql://localhost:3306/advertisement?useUnicode=true&characterEncoding=utf-8";
        Connection con=DriverManager.getConnection(uri,"root","123456");
        Statement sql=con.createStatement();
        ResultSet rs=sql.executeQuery("SELECT * FROM member where logname = '"+logname+"'");
        if(rs.next())
        {
            if(isLetterOrDigit)
            {
                int mm=sql.executeUpdate("UPDATE member SET pic= '"+savedFileName+"' where
logname = '"+logname+"'");
                if(mm!=0)
                {
                    backNews=fileName+"成功上传";
                    upFile.setFileName(fileName);
                    upFile.setSavedFileName(savedFileName);
                    upFile.setBackNews(backNews);
                }
            }
        }
        con.close();
        fl.delete();
    }
    catch(Exception exp)
    {
        backNews=""+exp;
        upFile.setBackNews(backNews);
    }
    try{
        RequestDispatcher dispatcher=
request.getRequestDispatcher("showUploadMess.jsp");//转发
```



```

        dispatcher.forward(request, response);
    }
    catch(Exception ee){ }
}
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    doPost(request, response);
}
}

```

七、浏览会员广告

该模块负责分页显示注册会员的信息，包括会员名、广告标题、广告照片等，同时提供查找功能，即用户可以查找某个会员的信息。该模块的模型 Javabeen 分别描述会员信息和分页信息；该模块视图部分由三个 JSP 页面构成，一个 JSP 页面负责提交用户浏览会员信息的方式：分页浏览全部会员信息、浏览特定会员信息，另外两个 JSP 页面分别负责分页显示全体会员信息和显示特定会员信息；该模块的控制器 servlet 使用 doPost 方法查询数据库表中的全部记录，并对记录进行分页处理，使用 doGet 方法查询数据库表中特定的记录。另外，控制器能阻止未登录用户浏览和查询会员信息。

1. 模型(Javabeen)

模型有两个 Javabeen。

(1)会员信息

下列 Javabeen 描述会员信息。

MemberInform.java

```

package mybean.data;
public class MemberInform
{
    String logname="",advertiseTitle="",email="",
        phone="", message="",pic,backNews="";
    public void setLogname(String name)
    {
        logname=name;
    }
    public String getLogname()
    {
        return logname;
    }
    public void setAdvertiseTitle(String s)
    {
        advertiseTitle=s;
    }
}

```



```
public String getAdvertiseTitle()
{ return advertiseTitle;
}

public void setEmail(String em)
{ email=em;
}

public String getEmail()
{ return email;
}

public void setPhone(String ph)
{ phone=ph;
}

public String getPhone()
{ return phone;
}

public String getMessage()
{ return message;
}

public void setMessage(String m)
{ message=m;
}

public String getPic()
{ return pic;
}

public void setPic(String s)
{ pic=s;
}

public String getBackNews()
{ return backNews;
}

public void setBackNews(String s)
{ backNews=s;
}
}
```

(2)分页信息

下列 Javabean 描述记录分页信息。

ShowByPage.java

```
package mybean.data;
import com.sun.rowset.*;
public class ShowByPage
{   CachedRowSetImpl rowSet=null;           //存储表中全部记录的行集对象
    int pageSize=10;                         //每页显示的记录数
}
```

```

int pageAllCount=0;           //分页后的总页数
int showPage=1 ;             //当前显示页
StringBuffer presentPageResult; //显示当前页内容
public void setRowSet(CachedRowSetImpl set)
{
    rowSet=set;
}
public CachedRowSetImpl getRowSet()
{
    return rowSet;
}
public void setPageSize(int size)
{
    pageSize=size;
}
public int getPageSize()
{
    return pageSize;
}
public int getPageAllCount()
{
    return pageAllCount;
}
public void setPageAllCount(int n)
{
    pageAllCount=n;
}
public void setShowPage(int n)
{
    showPage=n;
}
public int getShowPage()
{
    return showPage;
}
public void setPresentPageResult(StringBuffer p)
{
    presentPageResult=p;
}
public StringBuffer getPresentPageResult()
{
    return presentPageResult;
}
}

```

2. 视图(JSP 页面)

本模块视图有三个 JSP 页面 choiceLookType.jsp、showAllMember.jsp 和 showLookedMember.jsp。choiceLookType.jsp 负责将浏览会员的方式提交给控制器，showAllMember.jsp 负责分页显示全体会员的信息，showLookedMember.jsp 负责显示被查找的会员的信息。

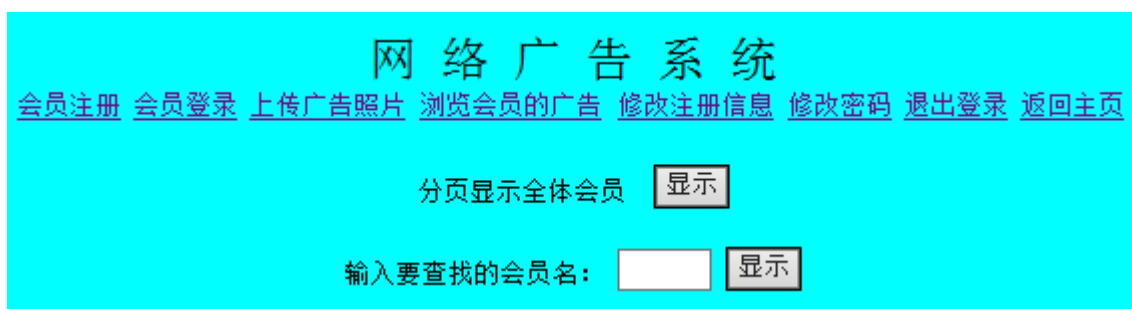
choiceLookType.jsp(效果如图所示)

```
<% @ page contentType="text/html;charset=GB2312" %>
```

```

<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<BODY bgcolor=cyan><center><Font size=2>
<center>
  <FORM action="helpShowMember" method="post" name="form">
    分页显示全体会员
    <INPUT type="hidden" value="1" name="showPage" size=5>
    <INPUT type="submit" value="显示" name="submit">
  </Form>
  <FORM action="helpShowMember" method="get" name="form">
    输入要查找的会员名:
    <INPUT type="text" name="logname" size=5>
    <INPUT type="submit" value="显示" name="submit">
  </FORM>
</center>
</BODY>
</HTML>

```



showAllMember.jsp (效果如图所示)

```

<% @ page contentType="text/html;charset=GB2312" %>
<% @ page import="mybean.data.ShowByPage" %>
<jsp:useBean id="show" type="mybean.data.ShowByPage" scope="session"/>
<HTML><head><% @ include file="head.txt" %></HEAD>
<Body ><center>
<P>显示会员信息。
<BR>每页最多显示<jsp:getProperty name="show" property="pageSize" />条信息。
<BR>当前显示第<Font color=blue>
  <jsp:getProperty name="show" property="showPage" />
</Font>页,共有
<Font color=blue><jsp:getProperty name="show" property="pageAllCount" />
</Font>页。
<BR>当前显示的内容是:
<table border=2>
<tr>
  <th>会员名</th><th>广告标题</th><th>电话</th>
  <th>email</th><th>广告词</th><th>广告照片</th>
</tr>

```



```
<jsp:getProperty name="show" property="currentPageResult" />
</table>
<BR>单击“前一页”或“后一页”按钮查看信息
<Table>
  <tr><td><FORM action="helpShowMember" method=post>
    <Input type=hidden name="showPage" value="<%=show.getShowPage()-1 %>" >
    <Input type=submit name="g" value="前一页">
  </FORM>
  </td>
  <td><FORM action="helpShowMember" method=post>
    <Input type=hidden name="showPage" value="<%=show.getShowPage()+1 %>" >
    <Input type=submit name="g" value="后一页">
  </Form>
  </td>
  <td> <FORM action="helpShowMember" method=post>
    输入页码: <Input type=text name="showPage" size=5 >
    <Input type=submit name="g" value="提交">
  </FORM>
  </td>
</tr>
</Table>
</Center>
</BODY></HTML>
```

网络广告系统

[会员注册](#)
[会员登录](#)
[上传广告照片](#)
[浏览会员的广告](#)
[修改注册信息](#)
[修改密码](#)
[退出登录](#)
[返回主页](#)

显示会员信息。
 每页最多显示3条信息。
 当前显示第 1 页, 共有 2 页。
 当前显示的内容是:

会员名	广告标题	电话	email	广告词	广告照片
fdasfaf	saffa				
lisi	zhangsan				
xxiaaaa	erqerqwr				

showLookedMember.jsp (效果如图所示)

```

<% @ page contentType="text/html;charset=GB2312" %>
<% @ page import="mybean.data.MemberInform" %>
<HTML>
<head><% @ include file="head.txt" %>
<jsp:useBean id="inform" type="mybean.data.MemberInform" scope="request" /></HEAD>
<BODY bgcolor=pink>
<Center>
<table border=2>
  <tr>
    <th>会员名</th><th>广告标题</th><th>电话</th>
    <th>email</th><th>广告词</th><th>广告照片</th>
  </tr>
  <tr>
    <td><jsp:getProperty name="inform" property="logname" /></td>
    <td><jsp:getProperty name="inform" property="advertiseTitle" /></td>
    <td><jsp:getProperty name="inform" property="phone" /></td>
    <td><jsp:getProperty name="inform" property="email" /></td>
    <td>a<jsp:getProperty name="inform" property="message" /></td>
    <td><img src=image/<jsp:getProperty name="inform" property="pic" /> width=50 height=50 >
      </img></td>
  </tr>

```

```
</table>
</Center>
</html>
```

网 络 广 告 系 统					
会员注册	会员登录	上传广告照片	浏览会员的广告	修改注册信息	修改密码
退出登录	返回主页				
会员名	广告标题	电话	email	广告词	广告照片
lisi	zhangsan		a		<input type="checkbox"/>

3. 控制器(servlet)

该模块控制器的名字是 lookRecord（见前文给出的 web.xml 配置文件）。lookRecord 使用 doPost 方法查询数据库表中的全部记录，并对记录进行分页处理，使用 doGet 方法查询数据库表中特定的记录。另外，控制器 lookRecord 能阻止未登录用户浏览和查询会员信息。

HandleDatabase.java

```
package myservlet.control;
import mybean.data.*;
import com.sun.rowset.*;
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HandleDatabase extends HttpServlet
{
    CachedRowSetImpl rowSet=null;

    public void init(ServletConfig config) throws ServletException
    {
        super.init(config);
        try {
            Class.forName("com.mysql.jdbc.Driver");
        }
        catch(Exception e){ }
    }

    public void doPost(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException
    {
        HttpSession session=request.getSession(true);
        Login login=(Login)session.getAttribute("login");//获取用户登录时的Javabean
        boolean ok=true;
        if(login==null)
        {
            ok=false;
            response.sendRedirect("login.jsp"); //重定向到登录页面
        }
        if(ok==true)
```




```
{ continueDoPost(request,response);
}

}

public void continueDoPost(HttpServletRequest request,HttpServletResponse response)
    throws ServletException,IOException
{
    HttpSession session=request.getSession(true);
    Connection con=null;
    StringBuffer presentPageResult=new StringBuffer();
    ShowByPage showBean=null;
    try{ showBean=(ShowByPage)session.getAttribute("show");
        if(showBean==null)
        { showBean=new ShowByPage(); //创建Javabean对象
          session.setAttribute("show",showBean);
        }
    }
    catch(Exception exp)
    { showBean=new ShowByPage();
      session.setAttribute("show",showBean);
    }

    showBean.setPageSize(3); //每页显示3条记录
    int showPage=Integer.parseInt(request.getParameter("showPage"));
    if(showPage>showBean.getPageAllCount())
        showPage=1;
    if(showPage<=0)
        showPage=showBean.getPageAllCount();
    showBean.setShowPage(showPage);
    int pageSize=showBean.getPageSize();
    String
uri="jdbc:mysql://localhost:3306/advertisement?useUnicode=true&characterEncoding=utf-8";
    try{ con=DriverManager.getConnection(uri,"root","123456");
        Statement sql=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                                           ResultSet.CONCUR_READ_ONLY);

        ResultSet rs=sql.executeQuery("SELECT * FROM member");
        rowSet=new CachedRowSetImpl(); //创建行集对象
        rowSet.populate(rs);
        con.close(); //关闭连接
        showBean.setRowSet(rowSet); //数据存储在showBean中
        rowSet.last();
        int m=rowSet.getRow(); //总行数
        int n=pageSize;
        int pageAllCount=((m%n)==0)?(m/n):(m/n+1);
        showBean.setPageAllCount(pageAllCount);//数据存储在showBean中
        presentPageResult=show(showPage,pageSize,rowSet);
        showBean.setPresentPageResult(presentPageResult);
    }
```



```
    }  
    catch(SQLException exp){}  
    RequestDispatcher dispatcher=  
request.getRequestDispatcher("showAllMember.jsp");//转发  
    dispatcher.forward(request, response);  
}  
  
public StringBuffer show(int page,int pageSize,CachedRowSetImpl rowSet)  
{    StringBuffer str=new StringBuffer();  
    try{ rowSet.absolute((page-1)*pageSize+1);  
        for(int i=1;i<=pageSize;i++)  
        { str.append("<tr>");  
            str.append("<td>"+rowSet.getString(1)+"</td>");  
            str.append("<td>"+rowSet.getString(3)+"</td>");  
            str.append("<td>"+rowSet.getString(4)+"</td>");  
            str.append("<td>"+rowSet.getString(5)+"</td>");  
            str.append("<td>"+rowSet.getString(6)+"</td>");  
            String s="<img src=image/"+rowSet.getString(7)+" width=100 height=100/>";  
            str.append("<td>"+s+"</td>");  
            str.append("</tr>");  
            rowSet.next();  
        }  
    }  
    catch(SQLException exp){}  
    return str;  
}  
  
public void doGet(HttpServletRequest request,HttpServletResponse response)  
    throws ServletException,IOException  
{    HttpSession session=request.getSession(true);  
    Login login=(Login)session.getAttribute("login");//获取用户登录时的JavaBean  
    boolean ok=true;  
    if(login==null)  
    { ok=false;  
        response.sendRedirect("login.jsp");           //重定向到登录页面  
    }  
    if(ok==true)  
    {    continueDoGet(request,response);  
    }  
}  
  
public void continueDoGet(HttpServletRequest request,HttpServletResponse response)  
    throws ServletException,IOException  
{    MemberInform inform=new MemberInform();  
    request.setAttribute("inform",inform);  
    String logname=request.getParameter("logname");  
    Connection con=null;
```

```
String
uri="jdbc:mysql://localhost:3306/advertisement?useUnicode=true&characterEncoding=utf-8";
try{ con=DriverManager.getConnection(uri,"root","123456");
    Statement sql=con.createStatement();
    ResultSet rs=
sql.executeQuery("SELECT * FROM member where logname = '"+logname+"'");
    if(rs.next())
    {   inform.setLogname(rs.getString(1));
        inform.setAdvertiseTitle(rs.getString(3));
        inform.setPhone(rs.getString(4));
        inform.setEmail(rs.getString(5));
        inform.setMessage(rs.getString(6));
        inform.setPic(rs.getString(7));
        inform.setBackNews("查询到的会员信息: ");
    }
    con.close();
    RequestDispatcher dispatcher=
request.getRequestDispatcher("showLookedMember.jsp");//转发
    dispatcher.forward(request, response);
}
catch(SQLException exp)
{   inform.setBackNews(""+exp);System.out.println("ok1"+exp);
}
}
```

八、修改密码

已经登录的用户可在该模块修改密码。该模块的模型 **Javabean** 描述密码的有关信息；该模块视图部分有两个 **JSP** 页面，一个 **JSP** 页面负责提交用户的新旧密码到控制器，另一个 **JSP** 页面负责显示修改是否成功的信息。该模块的控制器 **servlet** 负责修改密码。

1. 模型(Javabean)

下列 **Javabean** 的实例用来描述修改密码有关信息。

Password.java

```
package mybean.data;
public class Password
{   String   oldPassword,newPassword,backNews="";
    public void setNewPassword(String pw)
    {   newPassword=pw;
    }
```

```

public String getnewPassword()
{
    return newPassword;
}

public void setOldPassword(String pw)
{
    oldPassword=pw;
}

public String getOldPassword()
{
    return oldPassword;
}

public String getBackNews()
{
    return backNews;
}

public void setBackNews(String s)
{
    backNews=s;
}
}

```

2. 视图(JSP 页面)

本模块视图有两个 JSP 页面 modifyPassword.jsp 和 showNewPasswor.jsp。modifyPassword.jsp 页面负责提供输入密码界面，showNewPasswor.jsp 负责显示修改密码的反馈信息。

modifyPassword.jsp(效果如图所示)

```

<%@ page contentType="text/html;charset=GB2312" %>
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<BODY bgcolor=pink><Font size=2><CENTER>
<BR>输入您的密码:
<FORM action="helpModifyPassword" Method="post">
<BR>当前密码:<Input type=text name="oldPassword">
<BR>新密码: <Input type=password name="newPassword">
<BR><Input type=submit name="g" value="提交">
</Form></CENTER>
</BODY></HTML>

```

网络广告系统

[会员注册](#) [会员登录](#) [上传广告照片](#) [浏览会员的广告](#) [修改注册信息](#) [修改密码](#) [退出登录](#) [返回主页](#)

输入您的密码:

当前密码:

新密码:

showNewPasswor.jsp(效果如图所示)

```
<% @ page contentType="text/html;charset=GB2312" %>
<% @ page import="mybean.data.Password" %>
<jsp:useBean id="password" type="mybean.data.Password" scope="request" />
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<BODY bgcolor=yellow >
<CENTER>
    <jsp:getProperty name="password" property="backNews" />
    <BR>您的新密码: <jsp:getProperty name="password" property="newPassword" />
    <BR>您的旧密码: <jsp:getProperty name="password" property="oldPassword" />
</CENTER>
</BODY></HTML>
```

3. 控制器(servlet)

控制器负责连接数据库，根据当前用户注册的会员名修改 member 表中该会员的 password 字段的值，并转发修改信息到 showModifyMess.jsp 页面。另外，控制器能阻止未登录用户浏览进行修改密码操作。

HandlePassword.java

```
package myservlet.control;
import mybean.data.*;
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HandlePassword extends HttpServlet
{
    public void init(ServletConfig config) throws ServletException
    {
        super.init(config);
        try { Class.forName("com.mysql.jdbc.Driver");
        }
        catch(Exception e){}
    }
    public void doPost(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException
    {
        HttpSession session=request.getSession(true);
        Login login=(Login)session.getAttribute("login"); //获取用户登录时的Javabean
        boolean ok=true;
        if(login==null)
        { ok=false;
            response.sendRedirect("login.jsp"); //重定向到登录页面
        }
        if(ok==true)
```



```
{    continueWork(request,response);
}

}

public void    continueWork(HttpServletRequest request,HttpServletResponse response)
                throws ServletException,IOException

{    HttpSession session=request.getSession(true);
    Login login=(Login)session.getAttribute("login");
    Connection con=null;
    String logname=login.getLogname();
    Password passwordBean=new Password();
    request.setAttribute("password",passwordBean);
    String oldPassword=request.getParameter("oldPassword");
    String newPassword=request.getParameter("newPassword");
    String
uri="jdbc:mysql://localhost:3306/advertisement?useUnicode=true&characterEncoding=utf-8";
    try{ con=DriverManager.getConnection(uri,"root","123456");
        Statement sql=con.createStatement();
        ResultSet rs=
        sql.executeQuery("SELECT * FROM member where logname='"+
logname+"And password='"+oldPassword+"'");
        if(rs.next())
        { String updateString="UPDATE member SET password='"+
newPassword+" where logname='"+logname+"'";
            int m=sql.executeUpdate(updateString);
            if(m==1)
            { passwordBean.setBackNews("密码更新成功");
                passwordBean.setOldPassword(oldPassword);
                passwordBean.setNewPassword(newPassword);
            }
            else
            { passwordBean.setBackNews("密码更新失败");
            }
        }
        else
        { passwordBean.setBackNews("密码更新失败");
        }
    }
    catch(SQLException exp)
    {    passwordBean.setBackNews("密码更新失败"+exp);
    }

    RequestDispatcher dispatcher=
request.getRequestDispatcher("showNewPassword.jsp");//转发
    dispatcher.forward(request, response);
}
```

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    doPost(request, response);
}
}
```

九、修改注册信息

用户可在该模块修改已经注册成功的个人信息。该模块的模型 **Javabean** 描述用户修改的信息；该模块视图部分有三个 **JSP** 页面，第一个 **JSP** 页面负责确认是否进行修改，第二个 **JSP** 页面负责提交用户的修改信息到控制器，最后一个 **JSP** 页面负责显示修改是否成功的信息。该模块的控制器有两个 **servlet**，一个负责读取用户旧的注册信息；另一个负责修改旧的注册信息，两个 **servlet** 都能阻止未登录用户使用该模块。

1. 模型（javabean）

模型中需要两个 **JavaBean**，一个是第 4 部分中的 **Register.java**，另一个是下面的 **ModifyMessage.java**，该类用实例来描述用户的修改信息。

ModifyMessage.java

```
package mybean.data;
public class ModifyMessage
{
    String logname="",newAdvertiseTitle="",newEmail="",newPhone="",
        newMessage="",backNews;
    public void setLogname(String name)
    {
        logname=name;
    }
    public String getLogname()
    {
        return logname;
    }
    public void setNewAdvertiseTitle(String s)
    {
        newAdvertiseTitle=s;
    }
    public String getNewAdvertiseTitle()
    {
        return newAdvertiseTitle;
    }
    public void setNewEmail(String em)
    {
        newEmail=em;
    }
    public String getNewEmail()
    {
        return newEmail;
    }
}
```

```

    }

    public void setNewPhone(String ph)
    {   newPhone=ph;
    }

    public String getNewPhone()
    {   return newPhone;
    }

    public String getNewMessage()
    {   return newMessage;
    }

    public void setNewMessage(String m)
    {   newMessage=m;
    }

    public String getBackNews()
    {   return backNews;
    }

    public void setBackNews(String s)
    {   backNews=s;
    }

}

```

2. 视图(JSP 页面)

本模块视图有三个 JSP 页面 choiceModifyMess.jsp、inputNewMess.jsp 和 showModifyMess.jsp。用户选择修改信息视图后，首先进入 choiceModifyMess.jsp 页面，在该页面确认是否开始修改操作；inputNewMess.jsp 页面负责提供修改信息界面，showModifyMess.jsp 页面负责显示修改反馈信息。当用户访问 choiceModifyMess.jsp 页面时，如果确认要修改自己的注册信息，将首先被定向到控制器中的一个 servlet。如果是登录的用户，该 servlet 将查询该用户曾注册的信息，并将该登录用户重新定向到 inputNewMess.jsp 页面；如果用户没有登录，将被定向到登录页面 login.jsp。

choiceModifyMess.jsp（效果如图所示）

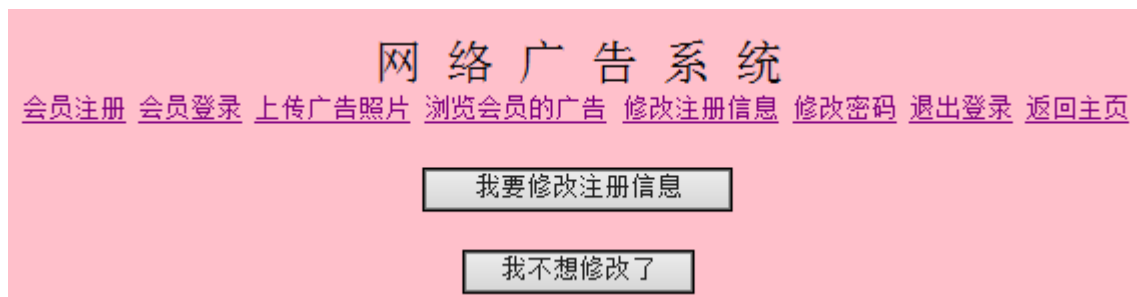
```

<%@ page contentType="text/html;charset=GB2312" %>
<HTML>
<BODY bgcolor=pink>
<%@ include file="head.txt" %>
<Center>
    <FORM action="helpGetOldMess" name=form>
        <INPUT type=submit value="我要修改注册信息">
    </FORM>
    <FORM action="index.jsp" name=form>
        <INPUT type=submit value="我不想修改了">
    </FORM>

```



```
</Center>
</BODY>
</HTML>
```



inputModifyMess.jsp(效果如图 10-15 所示)

```
<% @ page contentType="text/html;charset=GB2312" %>
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<% @ page import="mybean.data.Register"%>
<jsp:useBean id="register" type="mybean.data.Register" scope="request" />
<HTML><BODY bgcolor=pink><CENTER>
<P>
<Font color=blue size=4>
    以下是您(<jsp:getProperty name="register" property="logname"/>)曾注册的信息,
    您可以修改这些信息。
</Font>
<Font size=2>
<FORM action="helpModifyMess" name=form>
<table>
    <tr><td>广告标题:</td><td><Input type=text name="newAdvertiseTitle"
        value=<jsp:getProperty name="register" property="advertiseTitle" /> </td></tr>
    <tr><td>联系电话:</td><td><Input type=text name="newPhone"
        value=<jsp:getProperty name="register" property="phone" /> </td></tr>
    <tr><td>电子邮件:</td><td><Input type=text name="newEmail"
        value=<jsp:getProperty name="register" property="email" /></td></tr>
</table>
<table>
    <tr><td>广告词: </td></tr>
    <tr>
        <td><TextArea name="newMessage" Rows="6" Cols="30">
            <jsp:getProperty name="register" property="message"/>
        </TextArea></td>
    </tr>
    <tr><td><Input type=submit name="g" value="提交修改"></td></tr>
    <tr><td><Input type=reset value="重置"></td></tr>
</table><Font></CENTER>
</BODY></HTML>
```

网 络 广 告 系 统

[会员注册](#) [会员登录](#) [上传广告照片](#) [浏览会员的广告](#) [修改注册信息](#) [修改密码](#) [退出登录](#) [返回主页](#)

以下是您(lisi)曾注册的信息， 您可以修改这些信息。

广告标题:

联系电话:

电子邮件:

广告词:

showModifyMess.jsp(效果如图所示)

```
<% @ page contentType="text/html;charset=GB2312" %>
<% @ page import="mybean.data.ModifyMessage"%>
<jsp:useBean id="modify" type="mybean.data.ModifyMessage" scope="request" />
<HTML><HEAD><% @ include file="head.txt" %></HEAD>
<BODY bgcolor=yellow >
<CENTER>
<jsp:getProperty name="modify" property="backNews" />
<BR>您修改信息如下:
  <BR>新广告标题:<jsp:getProperty name="modify" property="newAdvertiseTitle" />
  <BR>新电话:<jsp:getProperty name="modify" property="newPhone" />
  <BR>新email:<jsp:getProperty name="modify" property="newEmail" />
  <BR>新广告词:<jsp:getProperty name="modify" property="newMessage" />
</CENTER>
</BODY>
</HTML>
```

网 络 广 告 系 统

[会员注册](#) [会员登录](#) [上传广告照片](#) [浏览会员的广告](#) [修改注册信息](#) [修改密码](#) [退出登录](#) [返回主页](#)

修改信息成功
您修改信息如下:
新广告标题:zhangsan
新电话:fdsafaf
新email:fdasfsaf
新广告词: fddgsagasg

3. 控制器(servlet)

(1)查询注册信息的 servlet

该 servlet 对象的名字是 `getOldMess`(见前文给出的 `web.xml` 配置文件)。
`getOldMess` 负责连接数据库, 查询 `member` 表, 得到当前登录会员的注册信息 (不包括密码和照片)。然后将用户转发到 `inputModifyMess.jsp` 页面。

`GetOldMess.java`

```
package myservlet.control;
import mybean.data.*;
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class GetOldMess extends HttpServlet
{
    public void init(ServletConfig config) throws ServletException
    {
        super.init(config);
        try { Class.forName("com.mysql.jdbc.Driver");
        }
        catch(Exception e){ }
    }

    public void doPost(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException
    {
        HttpSession session=request.getSession(true);
        Login login=(Login)session.getAttribute("login");//获取用户登录时的Javabea
        boolean ok=true;
        if(login==null)
        { ok=false;
            response.sendRedirect("login.jsp");           //重定向到登录页面
        }
        if(ok==true)
        { continueWork(request,response);
        }
    }

    public void continueWork(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException
    {
        HttpSession session=request.getSession(true);
        Login login=(Login)session.getAttribute("login");
        Connection con=null;
        String logname=login.getLogname();
        Register register=new Register();
        request.setAttribute("register",register);
        String
uri="jdbc:mysql://localhost:3306/advertisement?useUnicode=true&characterEncoding=utf-8";
        try{ con=DriverManager.getConnection(uri,"root","123456");
```

```

        Statement sql=con.createStatement();
        ResultSet rs=
        sql.executeQuery("SELECT * FROM member where logname='"+logname+"'");
        if(rs.next())
        {
            register.setLogname(rs.getString(1));
            register.setAdvertiseTitle(rs.getString(3));
            register.setPhone(rs.getString(4));
            register.setEmail(rs.getString(5));
            register.setMessage(rs.getString(6));
            register.setBackNews("您原来的注册信息:");
        }
    }
    catch(SQLException exp)
    {
        register.setBackNews(""+exp);
    }

    RequestDispatcher dispatcher=
    request.getRequestDispatcher("inputModifyMess.jsp");//转发
    dispatcher.forward(request, response);
}

public void doGet(HttpServletRequest request,HttpServletResponse response)
    throws ServletException,IOException
{
    doPost(request,response);
}
}

```

(2)负责修改注册信息的 servlet

该 servlet 对象的名字是 modifyOldMess（见前文给出的 web.xml 配置文件）。modifyOldMess 负责连接数据库，将用户提交的新的信息写入到 member 表，并将用户转发到 showModifyMess.jsp 页面查看修改反馈信息。

HandleModifyMess.java

```

package myservlet.control;
import mybean.data.*;
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HandleModifyMess extends HttpServlet
{
    public void init(ServletConfig config) throws ServletException
    {
        super.init(config);
        try {
            Class.forName("com.mysql.jdbc.Driver");
        }
        catch(Exception e){ }
    }
    public String handleString(String s)

```



```
{    try{ byte bb[]=s.getBytes("iso-8859-1");
        s=new String(bb);
    }
    catch(Exception ee){}
    return s;
}

public void doPost(HttpServletRequest request,HttpServletResponse response)
    throws ServletException,IOException
{    HttpSession session=request.getSession(true);
    Login login=(Login)session.getAttribute("login");//获取用户登录时的Javabea
    boolean ok=true;
    if(login==null)
    { ok=false;
        response.sendRedirect("login.jsp");           //重定向到登录页面
    }
    if(ok==true)
    {    continueDoPost(request,response);
    }
}

public void    continueDoPost(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException,IOException
{    HttpSession session=request.getSession(true);
    Login login=(Login)session.getAttribute("login");
    String logname=login.getLogname();
    Connection con;
    PreparedStatement sql;
    ModifyMessage modify=new ModifyMessage();
    request.setAttribute("modify",modify);
    String advertiseTitle=request.getParameter("newAdvertiseTitle").trim(),
    email=request.getParameter("newEmail").trim(),
    phone=request.getParameter("newPhone").trim(),
    message=request.getParameter("newMessage");
    String
uri="jdbc:mysql://localhost:3306/advertisement?useUnicode=true&characterEncoding=utf-8";
    String backNews="";
    try{ con=DriverManager.getConnection(uri,"root","123456");
        String updateCondition=
"UPDATE member SET advertiseTitle=?,phone=?,email=?,message=? WHERE logname=?";
        sql=con.prepareStatement(updateCondition);
        sql.setString(1,handleString(advertiseTitle));
        sql.setString(2,phone);
        sql.setString(3,handleString(email));
        sql.setString(4,handleString(message));
```



```
sql.setString(5,logname);
int m=sql.executeUpdate();
if(m==1)
{ backNews="修改信息成功";
  modify.setBackNews(backNews);
  modify.setLogname(logname);
  modify.setNewAdvertiseTitle (handleString(advertiseTitle));
  modify.setNewEmail(handleString(email));
  modify.setNewPhone(phone);
  modify.setNewMessage(handleString(message));
}
else
{ backNews="信息填写不完整或信息中有非法字符";
  modify.setBackNews(backNews);
}
con.close();
}
catch(SQLException exp)
{ modify.setBackNews(""+exp);
}

RequestDispatcher dispatcher=
request.getRequestDispatcher("showModifyMess.jsp");//转发
dispatcher.forward(request, response);
}

public void doGet(HttpServletRequest request,HttpServletResponse response)
throws ServletException,IOException
{ doPost(request,response);
}
}
```

十、退出登录

该模块只有一个名字为 exit 的 servlet 控制器，exit 负责销毁用户的 session 对象，导致登录失效。

控制器(servlet)

HandleExit.java

案例 1 图书借阅管理应用网站系统

一个现代化的图书借阅管理系统在正常运营中总是面对大量的读者信息、书籍信息以及两者相互作用产生的借书信息、还书信息等。面对图书借阅管理数以万计的图书、纷繁复杂的读者信息、频繁更替的借书信息，传统的直接方法不但管理起来及其不方便，而且极容易出现漏洞，从而造成损失。因此一个智能化、系统化、信息化的图书借阅管理信息系统是十分必要的。充分利用计算机的功能实现对读者管理、书籍管理、借阅管理等自动化控制，将会使图书管理工作大大减轻，方便友好的图书用户界面，简便的操作，完善的数据库管理，将会使得图书借阅管理系统极大限度的应用于现代化的图书借阅中，成为图书管理人员的得力助手。

一、系统分析

现在流行的图书借阅网站不仅要有漂亮的网页，更要有严谨的规划。每一个细小的环节都很重要。运用相关应用网站开发技术把数据库和动态网页相关联。使得图书借阅管理系统中的数据能及时得到更新，保存，从而方便读者对图书相关信息的查询，续借。同时也大大提高了管理员的工作效率。

该图书借阅管理系统是以已有图书为主体，以强大的计算机软硬件技术为后台支撑的，运用数据库对图书进行更新，保存等管理。该系统具有以下功能：

1. 支持信息存储：此系统应该支持所有的图书信息数据；
2. 完善的权限设置：基于角色的权限设置，同时系统具有完善的信息审核机制，来保证系统的安全性，防范攻击；
3. 易用性：通过简易的浏览器界面操作，非专业人员也可完成图书的查询、借阅、续借等的全部工作，不需要具备 C# 语言等专业知识；
4. 易维护性：采用当前流行的 B/S 模式，系统支持多用户同时本地操作。支持不同的 Web 应用开发语言；
5. 支持独立的网页设计。

作为一个图书借阅管理系统的设计目的，就是要能正确、及时、方便的传递信息给用户。从用户角度来讲，就是为用户借阅图书提供方便，要尽量使用户使用起来觉得方便，因此：

1. 使用的浏览软件应尽量简单，界面友好；
2. 对图书进行组织，归类，使读者在大量的图书信息面前能清楚的选择自己所需要的图书信息。

综合性的图书借阅管理系统。应该具有以下功能：

1. 用户权限管理：实现了对管理员的添加、用户权限的设置、用户权限的修改和对用户的口令进行更改；

2. 书架设置管理：实现了对书架的增加、修改和删除的功能；
3. 读者类型管理：实现了对读者类型的增加、删除和修改的功能；
4. 读者档案管理：实现了对读者信息的增加、删除和修改的功能；
5. 图书类型管理：实现了对图书类型信息的增加、修改和删除的功能；
6. 图书档案管理：实现了对图书档案信息的增加、删除和修改的功能；

数据流图，E-R 图，功能模块图（设计报告中必须有，此处略）。

二、系统数据表设计

为实现系统功能，本系统设计了 9 个表：用户表、用户权限表、图书借阅信息表、书架信息表、图书信息表、图书类型信息表、读者信息表、读者类型表和管理人员信息表。

1. 用户表 (tb_user)：用户表用来存放用户编号、用户名称、用户密码和用户类型，当用户进入系统时，需要输入用户名称和用户密码，选择用户的类型，此时将调用此表。如果用户名称，用户密码和用户类型同时正确，则用户就可以进入该系统进行操作了，否则将进入不了该系统。

表 1 用户表

列名	数据类型	长度	允许空	主键否	功能描述
userId	int	4	否	是	用户编号
userName	varchar	50	是	否	用户名称
userPwd	varchar	50	是	否	用户密码
userType	varchar	50	是	否	用户类型

2. 用户权限设置表 (tb_admSet)：当用户进入本系统后，需要对该用户进行权限设置，如果此用户无哪个模块的权限，则此用户对该模块就不能进行访问了，当对用户进行权限设置时系统将调用数据库中的用户权限表。用户的类型分为管理员和读者两种，其中管理员可以具有访问所有页面的权限，但读者仅具有图书续借、图书借阅查询和图书档案查询的权限。

表 2 用户权限设置表

列名	数据类型	长度	允许空	主键否	功能描述
userName	varchar	80	否	否	用户名称
userManager	bit	1	是	否	管理员设置
bookcaseManage	bit	1	是	否	书架设置
readerType	bit	1	是	否	读者类型
readerInfo	bit	1	是	否	读者档案
bookType	bit	1	是	否	图书类型

bookInfoManage	bit	1	是	否	图书档案
bookBorrow	bit	1	是	否	图书借阅
bookRenewal	bit	1	是	否	图书续借
bookReturn	bit	1	是	否	图书归还
bookBorrowSearch	bit	1	是	否	图书借阅查询
bookInfoSearch	bit	1	是	否	图书档案查询

3. 图书借阅信息表 (tb_bookBorrow): 当读者需要借书时, 管理员输入读者的相关信息, 进行搜索, 显示出读者相关信息后, 再输入图书的条形码进行搜索, 然后单击借书的链接, 将调用图书信息表, 将图书信息表中图书的条形码, 图书名称和读者信息表中读者条形码和读者名称添加到图书借阅信息表中。此时图书借阅信息表中将增添一条新的图书相关记录。

表 3 图书借阅信息表

列名	数据类型	长度	允许空	主键否	功能描述
BookBarcode	varchar	50	否	否	图书条形码
BookName	varchar	20	是	否	图书名称

续表 4-3

BorrowTime	datetime	8	是	否	借书时间
ReturnTime	datetime	8	是	否	还书时间
ReaderBarcode	varchar	50	是	否	读者条形码
ReaderName	varchar	20	是	否	读者名称
IsRetrun	bit	1	是	否	是否归还

4. 书架信息表 (tb_bookcase): 当管理员进入系统后, 需要对书架信息进行添加、删除、修改时, 将调用书架信息表。

表 4 书架信息表

列名	数据类型	长度	允许空	主键否	功能描述
BookId	int	4	否	是	书架编号
BookcaseName	varchar	80	是	否	书架名称

5. 图书信息表 (tb_bookInfo): 当用户进入系统后进行的很多操作都会调用图书信息表, 此表中保存了图书的相关信息。

表 5 图书信息表

列名	数据类型	长度	允许空	主键否	功能描述
BookBarcode	varchar	100	否	是	图书条形码
BookName	varchar	100	是	否	图书名称
BookType	int	4	是	否	图书类型
BookCase	int	4	是	否	图书书架
BookConCern	varchar	100	是	否	出版社
Author	varchar	80	是	否	作者

Price	money	8	是	否	图书定价
BorrowSum	int	4	是	否	借阅次数

6. 图书类型信息表 (tb_bookType): 当管理员进入该图书借阅管理系统后, 如果需要添加图书类别时, 在图书类型添加页面填写完图书类别和可借天数的信息后, 单击保存按钮, 此条图书类型和可借阅的天数将保存到此表中。如果要删除和修改图书类别的信息, 只要单击所属图书类型表后的修改和删除链接, 则此表中的此条信息将会有相应的变化或被删除掉。

表 6 图书类型信息表

列名	数据类型	长度	允许空	主键否	功能描述
TypeID	int	4	否	否	类型编号
TypeName	varchar	50	是	否	类型名称
BorrowDay	int	4	是	否	可借天数

7. 读者信息表 (tb_readerInfo): 此表中保存了所有读者的相关信息, 当管理员进入该系统后单击用户档案的链接, 系统将会调用该读者信息表, 在此页面显示所有读者的相关信息, 当管理员添加读者信息时, 单击添加读者信息的链接, 在弹出的页面中输入相关的信息后, 单击保存按钮, 此条读者信息将会插到此读者信息表中。当管理员修改或删除读者的信息时, 只要单击所属该读者信息后的修改和删除链接即可, 则表中的此条读者信息将会有相应的变化或删除掉。

表 7 读者信息表

列名	数据类型	长度	允许空	是否主键	功能描述
ReaderBarcode	varchar	50	否	是	读者条形码
ReaderName	varchar	50	是	否	读者姓名
Sex	char	10	是	否	性别
ReaderType	varchar	50	是	否	读者类型
CertificateType	varchar	50	是	否	证件类型
Certificate	varchar	50	是	否	证件号码
Tel	varchar	50	是	否	联系电话
Email	varchar	50	是	否	邮箱
Remark	varchar	500	是	否	备注

8. 读者类型表 (tb_readerType): 当管理员进入该系统后, 单击读者类型。进入该页面后管理员要添加读者类型时, 单击添加读者类型的链接, 填写好类型名称和可借数量, 单击保存按钮, 此条读者类型信息将插入到该读者类型表中。当管理员修改或删除读者类型信息时, 只要单击所属该读者类型信息后的修改和删除链接即可, 则表中的此条读者类型信息将会有相应的变化或删除掉。

表 8 读者类型表

列名	数据类型	长度	允许空	是否主键	功能描述
Id	int	4	否	是	类型号
Type	varchar	50	是	否	读者类型名称
Num	varchar	50	是	否	可借数量

9. 管理员信息表 (tb_managerInfo): 此表中保存了管理员的相关信息, 当管理员进入该系统后, 添加管理员信息时, 单击添加管理员的链接, 在弹出的页面中输入相关的信息后, 单击保存按钮, 此条管理员信息将会插到此管理员信息表中。当管理员修改或删除管理员的信息时, 只要单击所属该管理员信息后的修改和删除链接即可, 则表中的此条管理员信息将会有相应的变化或删除掉。

表 9 管理员信息表

列名	数据类型	长度	允许空	是否主键	功能描述
ManagerBarcode	varchar	50	否	是	管理员条形码
ManagerName	varchar	50	是	否	管理员姓名
Sex	char	10	是	否	性别
CertificateType	varchar	50	是	否	证件类型
Certificate	varchar	50	是	否	证件号码
Tel	varchar	50	是	否	联系电话
Email	varchar	50	是	否	邮箱
Remark	varchar	500	是	否	备注

三、各功能模块设计与实现（参考）

1. 登录页面



图 1 登录页面

2.系统首页

用户通过“登录”模块的验证后，可以登录到图书借阅管理系统的首页，首页主要包括信息栏、导航栏、借阅排行榜和版权信息等 4 个部分。系统首页页面的运行效果如图 2 所示。



图 2 系统首页页面

3.管理员设置页面

管理员设置页面主要包括添加用户、用户权限设置、删除用户和更改口令 4 个功能。此页面中将显示用户名称和用户拥有的权限。管理员设置页面的运行效果如图 3 所示。

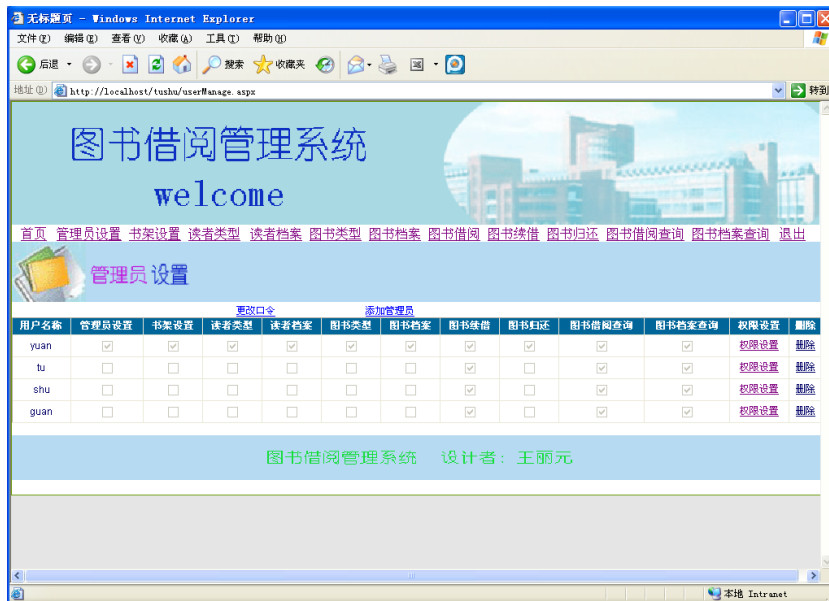


图 3 管理员设置页面

4 书架设置页面

书架设置模块主要包括添加、删除和修改 3 个功能。此页面将显示书架名称。单击“添加书架信息”的链接将弹出添加书架信息页面，在 TextBox 中添加好书架名称，单击保存，将完成添加书架操作。若想对书架信息进行修改或删除，只要单击书架信息所属行的修改或删除链接即可。书架设置页面的运行效果如图所示。



图 4 书架设置页面

5 图书借阅页面

管理员进入图书借阅页面后输入读者条形码单击查找读者按钮，将显示出该读者的信息，此时调用了读者信息表。然后输入读者想借的图书的条形码，将显示图

书信息，然后单击图书信息中“借阅”链接按钮实现借阅功能。此图书借阅页面运行效果如图 5 所示。



图 5 图书借阅页面

6 图书续借页面

进入到图书续借页面后，在此页面中根据管理员输入的读者条形码，显示读者信息和读者借阅未还的图书信息。输入读者需要续借图书的条形码，将显示此书信息。单击此书信息中“续借”链接按钮实现图书续借功能。图书续借页面运行效果如图 6 所示。



图 6 图书续借页面

7. 图书归还页面

1. 图书归还页面介绍

进入图书归还页面。在此页面中根据管理员输入的读者条形码，显示读者信息和读者借阅未还的图书信息。输入读者需要归还图书的条形码。将显示此书信息。单击此书信息中的“归还”链接按钮实现图书归还功能，如果读者归还的图书过期的话，将弹出罚款对话框，并显示出罚款金额。图书归还页面运行效果如图 7 所示。



图 7 图书归还页面

8. 图书借阅查询

本系统可以根据图书条形码、图书名称关键字对图书的借阅信息进行查询。页面的运行结果如图。图书借阅查询页面的运行结果如图 8 所示。



图 8 图书借阅查询页面

9 图书档案查询

当读者想要查询需要借阅的图书时，可以在此页面中根据图书条形码、书名、图书类别、出版社名称、作者来查看图书的详细资料。图书档案查询页面的运行结果如图 9 所示。

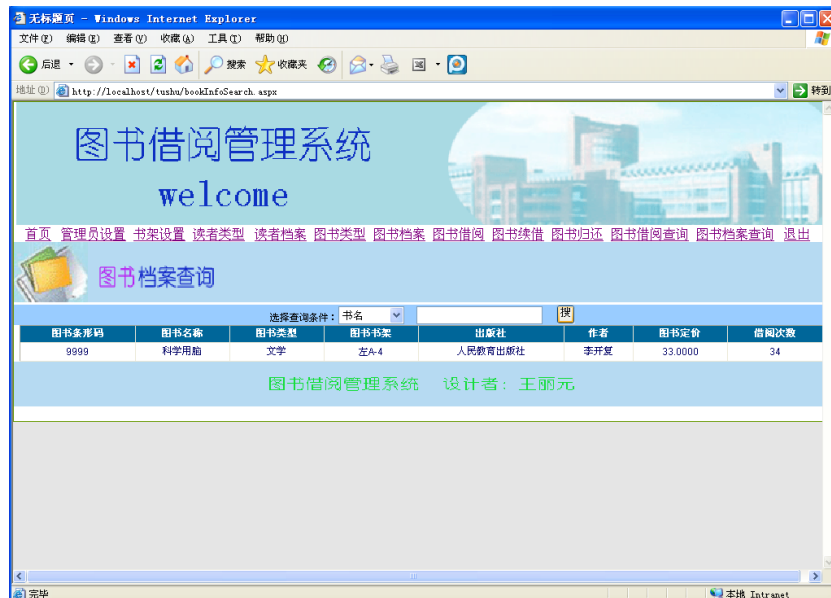


图 9 图书档案查询页面

四、利用 ADO 访问数据库

在 Web 动态交互过程中，经常需要根据用户输入的查询条件来查询数据库，从数据库中取出结果并以一定的格式显示给用户。有时也需要从用户那里收集信息，并按一定的格式保存到数据库里。本系统采用 SQL2000 数据库，下面介绍本系统数据库的连接方式。

1. 创建数据库链接(Connection)

ASP.NET 文件中如果要访问数据，必须首先创建与数据库的连接，本系统数据库连接如下：

```
SqlConnection Conn = new SqlConnection("Data Source=local;Initial Catalog=tikuguanli;Integrated Security=True");
```

这条语句打开链接，Conn 表示与数据源建立连接的对象，这里以与本系统的数据库连接为例，Source=local 连接本地数据源，Initial Catalog 选择要打开的数据库名。

2. 创建 Command 实例

创建数据库连接后，可以通过 SqlCommand 对象提供的方法传递对数据库操作的命令，并返回命令执行的结果。操作命令的类型可以使 SQL 语句，也可以是存储过程。例如下面是本系统中管理员登陆时所用到的 Command 对象：

```
SqlCommand cmd2 = new SqlCommand("select * from admin where admin_id =" +
```


Textbox_id.Text.ToString() + " and admin_pwd =" + Textbox_pwd.Text.ToString() + """, conn);

cmd2 是 Command 对象。括号里面的是 SQL 语句，用来查询数据库中等于管理员所输入的用户名和密码的信息；如果存在，则用户名和密码正确，用户进入管理员界面；否则用户名或密码错误，不能登录系统。

(3) 打开数据库连接

SQL2000 中的打开数据库连接的语句是：conn.Open();

(4) 执行操作

打开数据库后，就可以执行之前的 SQL 语句，对数据库进行操作了。

(5) 关闭连接

在执行完数据库操作后，应当及时关闭数据库。其关闭连接的语句是：
conn.Close();

以上几个步骤中创建连接与打开连接缺一不可，因为链接对象的创建与打开是两回事，只有打开了才真正可以使用。

