

第 10 章

实训一：会员管理系统

本章导读

- ☆ 知识点：掌握 Web 应用中常见基本模块的开发方法。
- ☆ 重点：掌握 MVC 模式在设计 Web 应用中的重要性。
- ☆ 难点：掌握“登录”和“上传照片”模块的设计
- ☆ 关键实践：对“会员管理系统”进行调试。

本章讲述如何用 JSP 技术建立一个简单的会员管理系统，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 Tomcat 5.5；系统采用 MVC 模式实现各个模块；数据库连接操作使用加载 Java 纯驱动程序方式，使得本系统可以采用任何提供 Java 纯驱动程序的数据库，本系统在实际应用中采用的是 SQL Server 2000 数据库。

10.1 系统模块构成

会员管理系统具有如下模块。

- ◎ **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。
- ◎ **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。
- ◎ **上传照片**：如果登录成功，用户可以使用该模块上传照片。
- ◎ **浏览会员**：成功登录的会员可以分页浏览其他会员，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。
- ◎ **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。
- ◎ **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，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。
- ◎ **退出登录**：成功登录的用户可以使用该模块退出登录。

10.2 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 SQL Server 2000 建立一个数据库 ComeHere, 该数据库共有一个表, 表的结构如下。

1. member 表的字段

会员的注册信息存入 member 表中, member 表的主键是 logname, 各字段值的说明如下:

- ◎ logname 存储会员登录名字。
- ◎ password 存储会员登录密码。
- ◎ sex 存储会员性别。
- ◎ age 存储会员的年龄。
- ◎ phone 存储会员的电话。
- ◎ email 存储会员的 E-mail 地址。
- ◎ message 存储会员的简历。
- ◎ pic 存储会员照片的名字。

2. member 表的结构

member 表的详细结构设计如图 10.1 所示。

设计表 "member", 位置是 "ComeHere" 中, "(local)" 上				
	列名	数据类型	长度	允许空
PK	logname	nvar char	50	
	password	char	10	
	sex	char	10	
	age	int	4	
	phone	nvar char	50	✓
	email	nvar char	50	✓
	message	nvar char	50	✓
	pic	nvar char	50	✓

图 10.1 member 表的结构

10.3 系统管理

本系统使用的 Web 服务目录是 chapter10。chapter10 是在 Tomcat 安装目录的 webapps 目录下建立的 Web 服务目录。

在当前 Web 服务目录下建立目录 chapter10\WEB-INF\classes, 然后根据 Servlet 的包名, 在 classes 下再建立相应的子目录, 如 Servlet 类的包名为 myservlet.control, 那么在 classes 下建立子目录 \myservlet\control; 如果 JavaBean 类的包名为 mybean.data, 那么在 classes 下建立子目录 \mybean\data。为了让 Tomcat 服务器启用上述目录, 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 chapter10 中。

所有的页面将包括一个导航条, 该导航条由注册、登录、上传照片、浏览会员、查看会员、留言板、修改密码、修改个人信息组成。为了减少代码的编写, 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 保存在目录 chapter10 中。head.txt 的内容如下:

head.txt

```

<%@ page contentType="text/html;charset=GB2312" %>
<CENTER><FONT size=6 ><P>会 员 管 理 系 统</FONT></CENTER>
<TABLE cellSpacing="1" cellPadding="1" width="500" align="center" border="0" >
  <TR valign="bottom">
    <TD align="center"><A href="register.jsp"><B>会员注册</B></A></TD>
    <TD align="center"><A href="login.jsp"><B>会员登录</B></A></TD>
    <TD align="center"><A href="upload.jsp"><B>上传照片</B></A></TD>
    <TD align="center"><A href="choiceLookType.jsp"><B>浏览会员</B></A></TD>
  </TR>
  <TR valign="bottom">
    <TD align="center"><A href="choiceModifyMess.jsp"><B>修改注册信息</B></A></TD>
    <TD align="center"><A href="modifyPassword.jsp"><B>修改密码</B></A></TD>
    <TD align="center"><A href="helpExitLogin"><B>退出登录</B></A></TD>
    <TD align="center"><A href="index.jsp"><B>返回主页</B></A></TD>
  </TR>
</FONT>
</TABLE>

```

主页 index.jsp 由导航条、一个欢迎语和一幅图片 welcome.jpg 组成, welcome.jpg 保存在目录 chapter10 中。

index.jsp

```

<%@ page contentType="text/html;charset=GB2312" %>
<HTML> <BODY>
<HEAD><%@ include file="head.txt" %></HEAD>
<CENTER> <H1><FONT size=9 color=red>欢迎您的到来</FONT></H1>
<IMG src="welcome.jpg" width=800 height=800 ></IMG>
</CENTER></H1>
</BODY></HTML>

```

用户可以通过在浏览器的地址栏中输入“http://服务器 IP:8080/index.jsp”或“http://服务器 IP:8080/”访问该主页, 主页运行效果如图 10.2 所示。

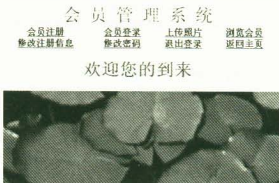


图 10.2 主页 index.jsp

2. JavaBean 与 Servlet 管理

本系统的 JavaBean 类的包名均为 mybean.data; Servlet 类的包名均为 myservlet.control。需要将编译通过的 JavaBean 类和 Servlet 类的字节码文件分别复制到目录 chapter10\WEB-INF\classes\mybean\data 和 chapter10\WEB-INF\classes\myservlet\control 中。

由于 Servlet 类中要使用 JavaBean, 所以为了能顺利编译 Servlet 类和 JavaBean 类, 需要将 JavaBean 类和 Servlet 类分别保存到 D:\myservlet\control\mybean\data 和 D:\myservlet\control 目录中。分别进入上述目录, 编译 JavaBean 类和 Servlet 类。

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。根据本书使用的 Tomcat 安装目录及使用的 Web 服务目录, 将下面的 web.xml 文件保存到 D:\apache-tomcat-5.5.20\webapps\chapter10\WEB-INF 中 (有关 web.xml 文件的配置规定可参见 8.1.2 节)。

web.xml

```
<?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>
    <servlet-name>login</servlet-name>
    <servlet-class>myservlet.control.HandleLogin</servlet-class>
</servlet>
<servlet>
    <servlet-name>upload</servlet-name>
    <servlet-class>myservlet.control.HandleUpload</servlet-class>
</servlet>
<servlet>
    <servlet-name>showImage</servlet-name>
    <servlet-class>myservlet.control.ShowImage</servlet-class>
</servlet>
<servlet>
    <servlet-name>lookRecord</servlet-name>
    <servlet-class>myservlet.control.HandleDatabase</servlet-class>
</servlet>
<servlet>
    <servlet-name>modifyPassword</servlet-name>
    <servlet-class>myservlet.control.HandlePassword</servlet-class>
</servlet>
<servlet>
    <servlet-name>getOldMess</servlet-name>
    <servlet-class>myservlet.control.GetOldMess</servlet-class>
</servlet>
<servlet>
    <servlet-name>modifyOldMess</servlet-name>
    <servlet-class>myservlet.control.HandleModifyMess</servlet-class>
</servlet>
```

```

<servlet>
    <servlet-name>exit</servlet-name>
    <servlet-class>myservlet.control.HandleExit</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>register</servlet-name>
    <url-pattern>/helpRegister</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>login</servlet-name>
    <url-pattern>/helpLogin</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>upload</servlet-name>
    <url-pattern>/helpUpload</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>showImage</servlet-name>
    <url-pattern>/helpShowImage</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>lookRecord</servlet-name>
    <url-pattern>/helpShowMember</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>modifyPassword</servlet-name>
    <url-pattern>/helpModifyPassword</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>getOldMess</servlet-name>
    <url-pattern>/helpGetOldMess</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>modifyOldMess</servlet-name>
    <url-pattern>/helpModifyMess</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>exit</servlet-name>
    <url-pattern>/helpExitLogin</url-pattern>
</servlet-mapping>
</web-app>

```

10.4 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；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；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 的实例用来描述用户注册信息。

Register.java

```
package mybean.data;
public class Register{
    String logname="",password="",sex="",
        email="", phone="", message="";
    String backNews;
    int age;
    public void setLogname(String name) {
        logname=name;
    }
    public String getLogname(){
        return logname;
    }
    public void setAge(int n) {
        age=n;
    }
    public int getAge(){
        return age;
    }
    public void setSex(String s) {
        sex=s;
    }
    public String getSex(){
        return sex;
    }
    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，如注册是否成功等。

Register.jsp (效果如图 10.3 所示)

图 10.3 填写注册信息

```

<%@ page contentType="text/html;charset=GB2312" %>
<HTML><HEAD><%@ include file="head.txt" %></HEAD>
<BODY><FONT size=2>
<CENTER>
<FORM action="helpRegister" name=form>
<BR>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。
<TABLE>
<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>性别:</TD>
<TD><Input type=radio name="sex" checked="o" value="男">男
<Input type=radio name="sex" value="女">女

```

```

</TD>
</TR>
<TR><TD>会员年龄:</td><td><input type="text" name="age" value="0"></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>输入您的个人简介: </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>

```

showRegisterMess.jsp（效果如图 10.4 所示）



图 10.4 显示注册结果

```

<%@ page contentType="text/html; charset=GB2312" %>
<%@ page import="mybean.data.Register"%>
<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>
<FONT size=2>
<TABLE>
<TR><TD>注册的会员名称:</td><td><jsp:getProperty name="register"
property="logname" /></TD></TR>
<TR><TD>注册的性别:</TD> <td><jsp:getProperty name="register" property="sex" /></TD></TR>
<TR><TD>注册的会员年龄:</TD><td><jsp:getProperty name="register"
property="age" /></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>您输入的个人简介: </TD></TR>
  <TR><TD><TextArea name="message" Rows="6" Cols="30">
    <jsp:getProperty name="register" property="message" /></TextArea></TD> </TR>
</TABLE>
</FONT>
</CENTER>
</BODY></HTML>

```

3. 控制器 (Servlet)

Servlet 负责连接数据库, 将用户提交的信息写入到 member 表, 并将用户转发到 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.*;

public class HandleRegister extends HttpServlet {
    public void init(ServletConfig config) throws ServletException{
        super.init(config);
        try { Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver"); }
        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;
        Register reg=new Register();
        request.setAttribute("register",reg);
        String logname=request.getParameter("logname").trim(),
        password=request.getParameter("password").trim(),
        sex=request.getParameter("sex").trim(),
        email=request.getParameter("email").trim(),
        phone=request.getParameter("phone").trim(),
        message=request.getParameter("message");
        int age=Integer.parseInt(request.getParameter("age").trim());
    }
}

```

```
String uri="jdbc:sqlserver://127.0.0.1:1433;DatabaseName=ComeHere";
if(logname==null)
    logname="";
if(password==null)
    password="";
boolean isLD=true;
for(int i=0;i<logname.length();i++) {
    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;
String backNews="";
try{ con=DriverManager.getConnection(uri,"sa","sa");
    String insertCondition="INSERT INTO member VALUES (?, ?, ?, ?, ?, ?, ?)";
    sql=con.prepareStatement(insertCondition);
    if(boo) {
        sql.setString(1,handleString(logname));
        sql.setString(2,handleString(password));
        sql.setString(3,handleString(sex));
        sql.setInt(4,age);
        sql.setString(5,phone);
        sql.setString(6,email);
        sql.setString(7,handleString(message));
        sql.setString(8,"public.jpg");
        int m=sql.executeUpdate();
        if(m!=0) {
            backNews="注册成功";
            reg.setBackNews(backNews);
            reg.setLogname(logname);
            reg.setPassword(handleString(password));
            reg.setAge(age);
            reg.setSex(handleString(sex));
            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";
        reg.setBackNews(backNews);
    }
    con.close();
}
catch(SQLException exp) {
    backNews="该会员名已被使用, 请您更换名字"+exp;
    reg.setBackNews(backNews);
}
```

```

    }
    RequestDispatcher dispatcher= request.getRequestDispatcher("/showRegiterMess.jsp");
    dispatcher.forward(request, response);
}
public void doGet(HttpServletRequest request, HttpServletResponse response)
                                throws ServletException, IOException {
    doPost(request,response);
}
}
}

```

10.5 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：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；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

```

package mybean.data;
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，如登录是否成功等。

login.jsp（效果如图 10.5 所示）

图 10.5 输入登录信息

```

<%@ page contentType="text/html; charset=GB2312" %>
<HTML><HEAD><%@ include file="head.txt" %></HEAD>
<BODY bgcolor=pink><FONT size=2><CENTER>
<P>请您登录
<FORM action="helpLogin" Method="post">
    <BR>登录名称:<Input type=text name="logname">
    <BR>输入密码:<Input type=password name="password">
    <BR><Input type=submit name="g" value="提交">
</FORM></CENTER>
</BODY></HTML>

```

showLoginMess.jsp（效果如图 10.6 所示）

图 10.6 显示登录结果

```

<%@ 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>
<HTML><BODY bgcolor=yellow >

```

```

<CENTER><FONT size=4 color=blue >
    <BR> <jsp:getProperty name="login" property="backNews" />
</FONT>
<% 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" />
<% }
%>
</FONT></CENTER>
</BODY></HTML>

```

3. 控制器 (Servlet)

控制器负责连接数据库, 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.microsoft.sqlserver.jdbc.SQLServerDriver"); }
        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:sqlserver://127.0.0.1:1433;DatabaseName=ComeHere";
    boolean boo=(logname.length()>0)&&(password.length()>0);
    try{ con=DriverManager.getConnection(uri,"sa","sa");
        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);
            }
        }
    }
    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);
}
}
}

```

10.6 上传照片

用户可在该模块上传自己的照片。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；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；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 10.7 所示)



图 10.7 选择上传的文件

```

<%@ page contentType="text/html;charset=GB2312" %>
<HTML><HEAD><%@ include file="head.txt" %></HEAD>
<BODY bgcolor=orange><CENTER>
    <P>文件将被上传到 D:\apache-tomcat-5.5.20\webapps\chapter10\image 中。
    <P>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):
    <BR>
    <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>
</BODY>
</HTML>

```

showUploadMess.jsp (效果如图 10.8 所示)

```

<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="mybean.data.UploadFile"%>
<jsp:useBean id="upFile" type="mybean.data.UploadFile" scope="request" />
<HTML><HEAD><%@ include file="head.txt" %></HEAD>
<HTML><BODY bgcolor=yellow >
<CENTER><FONT size=4 color=blue >

```




图 10.8 显示上传操作信息

```

<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=100 height=100>图像效果
</IMG>
</FONT></CENTER>
</BODY></HTML>

```

3. 控制器 (Servlet)

该 Servlet 负责检查用户是否是登录用户，如果用户没有登录，控制器将把用户定向到登录页面 login.jsp；对于登录的用户，该 Servlet 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，即保存到目录 D:\apache-tomcat-5.5.20\webapps\chapter10\image 中，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。该 Servlet 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.microsoft.sqlserver.jdbc.SQLServerDriver"); }
        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);    //接收上传文件
        }
    }

    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++;
            }
            int position=secondLine.lastIndexOf("\"); //获取第 2 行中目录符号'\'最后出现的位置
            //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称，将该文件存入磁盘
File dir=new File("D:/apache-tomcat-5.5.20/webapps/chaper10/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内容的最后位置，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:sqlserver://127.0.0.1:1433;DatabaseName=ComeHere";
    Connection con=DriverManager.getConnection(uri,"sa","sa");
    Statement sql=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                                        ResultSet.CONCUR_UPDATABLE);
    ResultSet rs=ql.executeQuery("SELECT * FROM members
                                where logname = '"+logname+"'");

    if(rs.next()){
        if(isLetterOrDigit) {
            rs.updateString(8,savedFileName);
            int index=rs.getRow();
            rs.absolute(index);
            rs.updateRow();
            backNews=fileName+"成功上传";
            upFile.setFileName(fileName);
            upFile.setSavedFileName(savedFileName);
            upFile.setBackNews(backNews);
        }
    }
    con.close();
    f1.delete();
}
catch(Exception exp) {
    backNews=""+"exp;
    upFile.setBackNews(backNews);
}
try{ RequestDispatcher
        dispatcher= request.getRequestDispatcher("/showUploadMess.jsp");
        dispatcher.forward(request, response);
}
catch(Exception ee){ }
}
public void doGet(HttpServletRequest request,HttpServletResponse response)
                                throws ServletException,IOException{
    doPost(request,response);
}
}

```

10.7 浏览会员

该模块负责分页显示注册会员的信息,包括会员名、性别、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 **JavaBean** 分别描述会员信息和分页信息;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；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 (JavaBean)

模型有两个 JavaBean。

(1) 会员信息

下列 JavaBean 的实例用来描述会员信息。

MemberInform.java

```
package mybean.data;

public class MemberInform {
    String logname="",sex="",email="",
        phone="",address="",
        message="",pic,backNews="";
    int age;
    public void setLogname(String name) {
        logname=name;
    }
    public String getLogname(){
        return logname;
    }
    public void setAge(int n) {
        age=n;
    }
    public int getAge(){
        return age;
    }
    public void setSex(String s) {
        sex=s;
    }
    public String getSex(){
        return sex;
    }
    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 的实例用来描述记录分页信息。

MemberInform.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、showLookedMember.jsp 和 showAllMember.jsp。choiceLookType.jsp 负责将浏览会员的方式提交给控制器, showLookedMember.jsp 负责显示被查找的会员的信息, showAllMember.jsp 负责分页显示全体会员的信息。

choiceLookType.jsp (效果如图 10.9 所示)

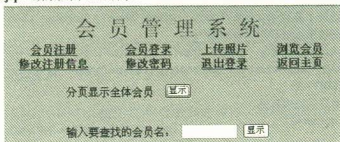


图 10.9 选择浏览方式

```

<%@ page contentType="text/html; charset=GB2312" %>
<HTML><HEAD><%@ include file="head.txt" %></HEAD>
<BODY bgcolor=cyan><CENTER><FONT size=3>
<TABLE>
    <FORM action="helpShowMember" method="post" name="form">
        <BR>分页显示全体会员
        <Input type="hidden" value="1" name="showPage" size=6>
        <Input type="submit" value="显示" name="submit">
    </FORM>
    <FORM action="helpShowMember" method="get" name="form">
        <BR>输入要查找的会员名:
        <Input type="text" name="logname" size=6>
        <Input type="submit" value="显示" name="submit">
    </FORM></CENTER>
</BODY></HTML>

```

showAllMember.jsp (效果如图 10.10 所示)



图 10.10 分页显示会员信息

```

<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="mybean.data.ShowByPage" %>
<jsp:useBean id="show" type="mybean.data.ShowByPage" scope="session"/>
<HTML><%@ 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>性别</TH><TH>年龄</TH><TH>电话</TH>
        <TH>email</TH><TH>简历</TH><TH>照片</TH>
    </TR>
    <jsp:getProperty name="show" property="presentPageResult" />
</TABLE>
<BR>单击“前一页”或“下一页”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>

```

showLookedMember.jsp (效果如图 10.11 所示)

会员管理系统						
会员注册		会员登录		上传照片		浏览会员
修改注册信息		修改密码		退出登录		返回主页
会员名	性别	年龄	电话	email	简历	照片
lixiao	男	33	0234-98765456	sss@yahoo.com.cn	喜欢攀登	

图 10.11 显示某个会员的信息

```

<%@ page contentType="text/html; charset=GB2312" %>
<%@ page import="mybean.data.MemberInform" %>
<jsp:useBean id="inform" type="mybean.data.MemberInform" scope="request" />
<CENTER>
<HTML><BODY bgcolor=pink><%@ include file="head.txt" %></HEAD>
<FONT size=3>
<TABLE border=2>
<TR>
<TH>会员名</TH><TH>性别</TH><TH>年龄</TH><TH>电话</TH>
<TH>email</TH><TH>简历</TH><TH>照片</TH>
</TR>
<TR>
<TD><jsp:getProperty name="inform" property="logname" /></TD>
<TD><jsp:getProperty name="inform" property="sex" /></TD>
<TD><jsp:getProperty name="inform" property="age" /></TD>
<TD><jsp:getProperty name="inform" property="phone" /></TD>
<TD><jsp:getProperty name="inform" property="email" /></TD>
<TD><jsp:getProperty name="inform" property="message" /></TD>
<TD><IMG src=image/<jsp:getProperty name="inform" property="pic" />></IMG></TD>
</TR>
</TABLE>
</CENTER>
</BODY></HTML>

```

3. 控制器 (Servlet)

Servlet 使用 doPost() 方法查询数据库表中的全部记录, 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。

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.microsoft.sqlserver.jdbc.SQLServerDriver"); }
        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:sqlserver://127.0.0.1:1433;DatabaseName= ComeHere";
try{  con=DriverManager.getConnection(uri,"sa","sa");
    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");
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>");
            str.append("<td>"+rowSet.getString(7)+"</td>");
            String s="<img src=image/"+rowSet.getString(8)+" 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:sqlserver://127.0.0.1:1433;DatabaseName= ComeHere";
        try{   con=DriverManager.getConnection(uri,"sa","sa");
            Statement sql=con.createStatement();
            ResultSet rs=sql.executeQuery("SELECT * FROM member
                                                WHERE logname = '"+logname+"'");

            if(rs.next()) {
                inform.setLogname(rs.getString(1));
                inform.setSex(rs.getString(3));
                inform.setAge(rs.getInt(4));
                inform.setPhone(rs.getString(5));
                inform.setEmail(rs.getString(6));
                inform.setMessage(rs.getString(7));
                inform.setPic(rs.getString(8));
                inform.setBackNews("查询到的会员信息: ");
            }
            con.close();
            RequestDispatcher dispatcher=
                request.getRequestDispatcher("/showLookedMember.jsp");
            dispatcher.forward(request, response);
        }
        catch(SQLException exp) {
            inform.setBackNews(""+exp);System.out.println("ok1"+exp);
        }
    }
}
}

```

10.8 修改密码

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；该模块视图部分由两个 JSP 页面构成，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 2 个 JSP 页面: modifyPassword.jsp、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 10.12 所示)

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML><HEAD><%@ include file="head.txt" %></HEAD>
<BODY bgcolor=pink><Font size=2><CENTER>
<P>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>
```

showNewPasswor.jsp (效果如图 10.13 所示)

```
<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="mybean.data.Password"%>
<jsp:useBean id="password" type="mybean.data.Password" scope="request" />
```

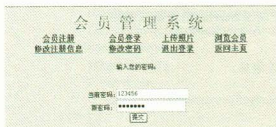


图 10.12 修改密码



图 10.13 显示修改密码结果

```
<HTML><HEAD><%@ include file="head.txt" %></HEAD>
<HTML><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" />
</FONT></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.microsoft.sqlserver.jdbc.SQLServerDriver"); }
        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:sqlserver://127.0.0.1:1433;DatabaseName= ComeHere";
    try{ con=DriverManager.getConnection(uri,"sa","sa");
        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");
    dispatcher.forward(request, response);
}

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

```

10.9 修改注册信息

用户可在该模块修改曾注册的个人信息。该模块的模型 **JavaBean** 描述用户修改的信息; 该模块视图部分由三个 **JSP** 页面构成, 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**, 一个负责读取用户曾注册的信息; 另一个负责修改曾注册的信息, 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**, 一个是在 10.4 节的模型中的 **Register.java**, 另一个是下列 **ModifyMessage.java**, 该类的实例用来描述用户的修改信息。

ModifyMessage.java

```
package mybean.data;
public class ModifyMessage {
    String logname="",newSex="",
        newEmail="",newPhone="",newMessage="",backNews;
    int newAge;
    public void setLogname(String name) {
        logname=name;
    }
    public String getLogname() {
        return logname;
    }
    public void setNewAge(int n) {
        newAge=n;
    }
    public int getNewAge(){
        return newAge;
    }
    public void setNewSex(String s) {
        newSex=s;
    }
    public String getNewSex() {
        return newSex;
    }
    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 3 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, 从中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 负责显示修改反馈信息。当用户访问 choiceModifyMess 页面时, 如果确认要修改自己的注册信息, 将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 10.14 所示)

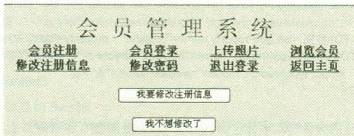


图 10.14 选择修改或放弃修改

```

<%@ page contentType="text/html;charset=GB2312" %>
<CENTER>
<HTML><BODY bgcolor=pink><%@ include file="head.txt" %></HEAD>
<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)

图 10.15 修改信息

```

<%@ 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>性别:<jsp:getProperty name="register" property="sex" /></TD>
        <TD><Input type=radio name="newSex" checked="o" value="男">男
            <Input type=radio name="newSex" value="女">女
        </TD>
    </TR>
    <TR><TD>会员年龄:</TD><TD><Input type=text name="newAge"
        value=<jsp:getProperty name="register" property="age" /></TD></TR>
    <TR><TD>电子邮件:</TD><TD><Input type=text name="newEmail"
        value=<jsp:getProperty name="register" property="email" /></TD></TR>
    <TR><TD>联系电话:</TD><TD><Input type=text name="newPhone"
        value=<jsp:getProperty name="register" property="phone" /></TD></TR>
</TABLE>
<TABLE>
    <TR><TD>输入您的个人简介: </TD></TR>
    <TR>
        <TD><TextArea name="newMessage" Rows="6" Cols="30">
            <jsp:getProperty name="register" property="message"/>
        </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>

```

showModifyMess.jsp (效果如图 10.16 所示)

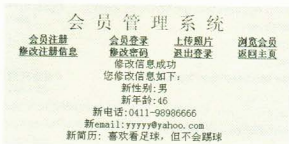


图 10.16 显示修改结果

```

<%@ 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>
<HTML><BODY bgcolor=yellow ><FONT size=4>
<CENTER>
<jsp:getProperty name="modify" property="backNews" />
<BR>您修改信息如下:
<BR>新性别:<jsp:getProperty name="modify" property="newSex" />
<BR>新年龄:<jsp:getProperty name="modify" property="newAge" />
<BR>新电话:<jsp:getProperty name="modify" property="newPhone" />
<BR>新 email:<jsp:getProperty name="modify" property="newEmail" />
<BR>新简历:<jsp:getProperty name="modify" property="newMessage" />
</FONT></CENTER>
</BODY></HTML>

```

3. 控制器 (Servlet)

(1) 查询注册信息的 Servlet

该 Servlet 负责连接数据库, 查询 member 表, 查询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.microsoft.sqlserver.jdbc.SQLServerDriver"); }

```

```

        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();
        Register register=new Register();
        request.setAttribute("register",register);
        String uri="jdbc:sqlserver://127.0.0.1:1433;DatabaseName= ComeHere";
        try{ con=DriverManager.getConnection(uri,"sa","sa");
            Statement sql=con.createStatement();
            ResultSet rs=
            sql.executeQuery("SELECT * FROM member where logname='"+logname+"'");
            if(rs.next()){
                register.setLogname(rs.getString(1));
                register.setSex(rs.getString(3));
                register.setAge(rs.getInt(4));
                register.setPhone(rs.getString(5));
                register.setEmail(rs.getString(6));
                register.setMessage(rs.getString(7));
                register.setBackNews("您原来的注册信息:");
            }
        }
        catch(SQLException exp) { register.setBackNews(""+exp); }
        RequestDispatcher dispatcher= request.getRequestDispatcher("/inputModifyMess.jsp");
        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 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 myervlet.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.microsoft.sqlserver.jdbc.SQLServerDriver"); }
        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 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);
        Login login=(Login)session.getAttribute("login");
        String logname=login.getLogname();
        Connection con;
        PreparedStatement sql;
        ModifyMessage modify=new ModifyMessage();
        request.setAttribute("modify",modify);
        String sex=request.getParameter("newSex").trim(),
```

```

email=request.getParameter("newEmail").trim(),
phone=request.getParameter("newPhone").trim(),
message=request.getParameter("newMessage");
int age=Integer.parseInt(request.getParameter("newAge").trim());
String uri="jdbc:sqlserver://127.0.0.1:1433;DatabaseName=ComeHere";
String backNews="";
try{ con=DriverManager.getConnection(uri,"sa","sa");
    String updateCondition=
        "UPDATE member SET sex=?,age=?,phone=?,email=?,message=?
                                                WHERE logname=?";

    sql=con.prepareStatement(updateCondition);
    sql.setString(1,handleString(sex));
    sql.setInt(2,age);
    sql.setString(3,phone);
    sql.setString(4,handleString(email));
    sql.setString(5,handleString(message));
    sql.setString(6,logname);
    int m=sql.executeUpdate();
    if(m==1) {
        backNews="修改信息成功";
        modify.setBackNews(backNews);
        modify.setLogname(logname);
        modify.setNewAge(age);
        modify.setNewSex(handleString(sex));
        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");
dispatcher.forward(request, response);
}

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

```

10.10 退出登录

该模块只有一个控制器，其中的 Servlet 负责销毁用户的 session 对象，导致登录失效。

HandleExit.java

```

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

public class HandleExit extends HttpServlet {
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        }
        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 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);
        session.invalidate();                                //销毁用户的 session 对象
        response.sendRedirect("index.jsp");                  //返回主页
    }
    public void doGet(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException {

        doPost(request,response);
    }
}

```