

项目 jsp 问卷调查系统

1 绪论

目前基于 internet 的网络问卷调查分析系统用的非常频繁和普遍,才会有那么多的企业和公司在用网络问卷调查法来获取信息。然而每开发一个新的网络问卷调查系统需要耗费大量的时间,很多已有的产品功能过于强大、设置负责、费用过于昂贵等缺点。

本系统主要实现最为普遍的问卷调查,包括问卷管理、设计、统计分析等。为需求者提供网络问卷,让需求者能够快速、方便的实施专业的问卷调研,调查问卷设计,快速回收,跨地域、低成本,投入少量资金即可获得丰富的反馈信息。

2 业务流程

2.1 业务流程图

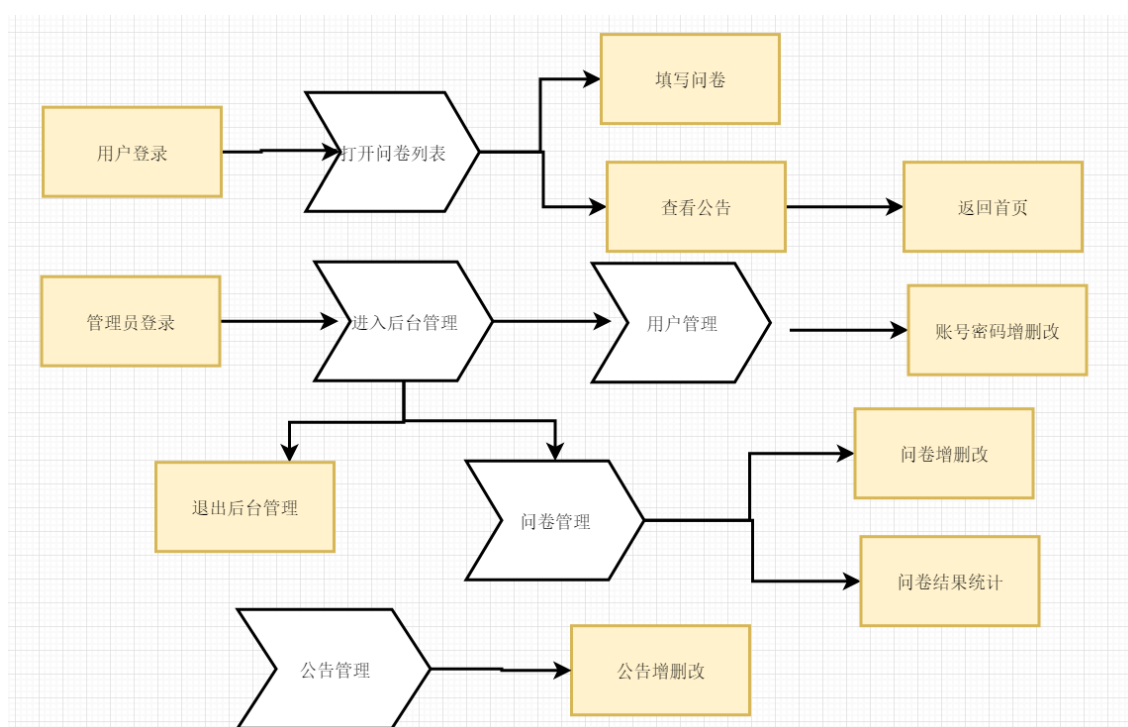


图 2.1 业务流程图

2.2 管理员用户

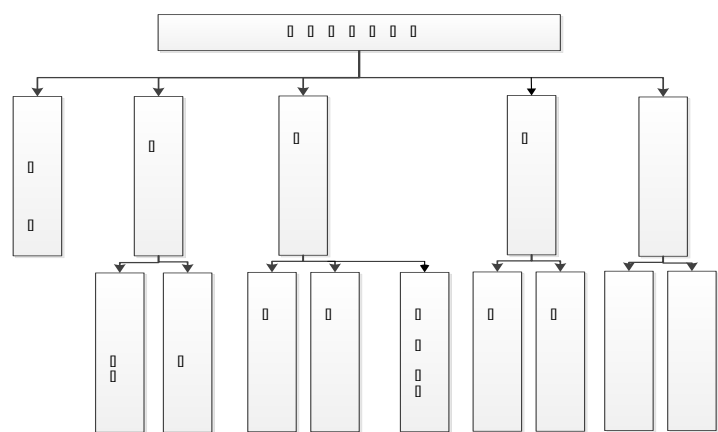


图 2-2 管理员业务

2.3 普通用户

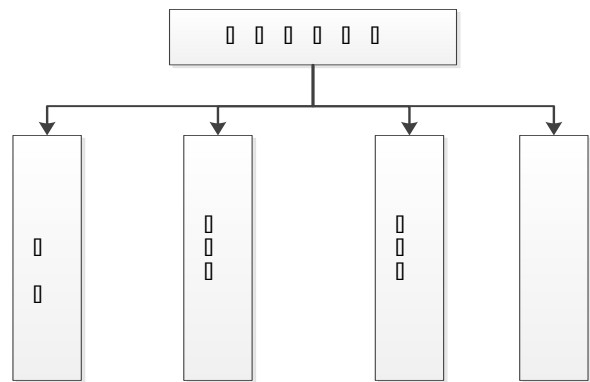


图 2-3 普通用户业务

3 文件结构

3.1 总文件结构

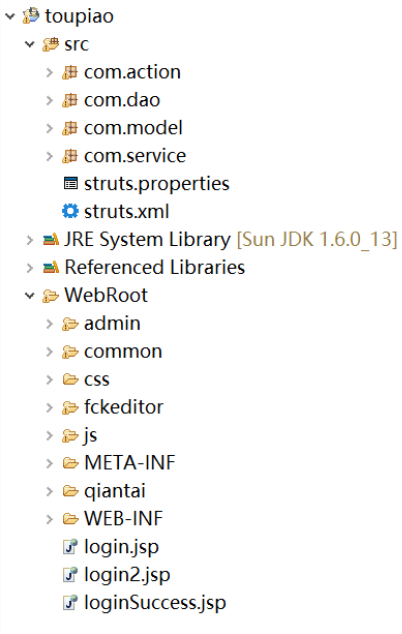


图 3.1

3.2 Java 文件结构

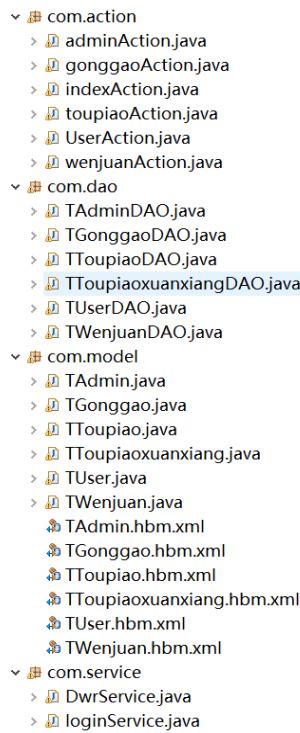


图 3.2

4 数据库的设计

(1)用户信息表主要是记录了注册用户基本信息。

列名	数据类型	长度	允许空	是否主键	说明
user_id	int	4	否	是	用户 ID
user_name	varchar	50	否	否	用户名
user_pw	varchar	50	否	否	用户密码
user_realname	varchar	50	否	否	姓名
user_address	varchar	50	否	否	住址
user_sex	varchar	50	否	否	性别
user_tel	varchar	20	否	否	联系方式
user_email	varchar	20	否	否	电子邮箱

表 4.1 用户信息表(t_user)

(2)问卷信息表主要记录了问卷调查的基本信息。

列名	数据类型	长度	允许空	是否主键	说明
id	int	4	否	是	编号
mingcheng	varchar	50	否	否	问卷名称
shijian	varchar	50	否	否	发布时间

表 4.2 问卷信息表(t_wenjuan)

(3)题目信息表主要是记录了问卷中的题目信息。

列名	数据类型	长度	允许空	是否主键	说明
id	int	4	否	是	编号
biaoti	varchar	50	否	否	题目标题
wenjuan_id	varchar	50	否	否	问卷 ID

表 4.3 题目信息表(t_toupiao)

(4)选项信息表主要是记录了题目的选项信息。

列名	数据类型	长度	允许空	是否主键	说明
id	int	4	否	是	编号
xuanxiangneirong	varchar	50	否	否	选项内容
piaoshu	varchar	50	否	否	票数
toupiao_id	varchar	50	否	否	题目 ID

表 4.4 选项信息表(t_toupiaoxuanxiang)

(5)公告信息表主要是注册管理员发布的公告信息。

列名	数据类型	长度	允许空	是否主键	说明
gonggao_id	Int	4	否	是	编号
gonggao_title	varchar	50	否	否	标题
gonggao_content	varchar	50	否	否	内容
gonggao_data	varchar	50	否	否	发布时间

表 4.5 公告信息表(t_gonggao)

(6)管理员信息表主要记录的管理员的账号信息，包括用户名和密码。

列名	数据类型	长度	允许空	是否主键	说明
----	------	----	-----	------	----

userId	int	4	否	是	编号
userName	varchar	50	否	否	用户名
userPw	varchar	50	否	否	密码

表 4.6 管理员信息表(t_admin)

5 系统功能实现原理

5.1 ssh 框架原理

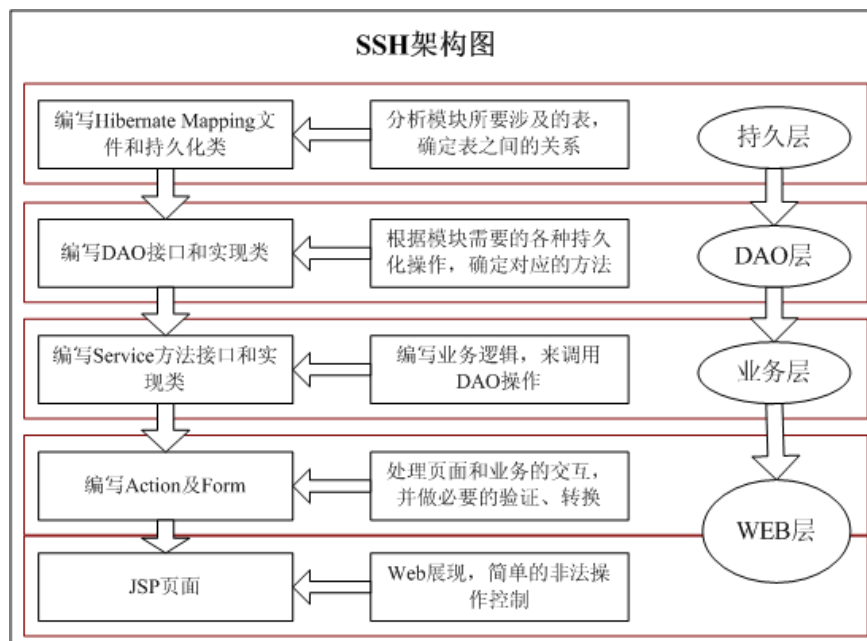


图 5.1

5.1.1 ssh 对象之间的调用流程

Struts—>Spring—>Hibernate

JSP—>Action—>Service—>DAO—>Hibernate

5.1.2 strust 对数据的 crud 操作实例

配置 struts.xml，对数据进行 crud 操作，以 admin 为例子增删改操作：

```
<action name="adminAdd" class="adminAction" method="adminAdd">
</action>
<action name="adminDel" class="adminAction" method="adminDel">
</action>
<action name="adminMana" class="adminAction" method="adminMana">
    <result name="success">/admin/admin/adminMana.jsp</result>
</action>
```

图 5.1.2

5.1.3 Jstl

Jstl 的 `<c:forEach>` 标签根据循环条件遍历数组和集合类中的所有或部分数据，使用 Hibernate 技术访问数据库时，返回的都是数组、`java.util.List` 和 `java.util.Map` 对象，它们封装着从数据库中查询得出的数据，这些数据是 JSP 页面需要的。如果在 JSP 页面中使用 Java 代码来循环遍历所有数据，会使页面非常混乱，不易分析和维护。使用 JSTL 的 `<c:forEach>` 标签循环显示这些数据可以解决 JSP 页面混乱的问题，同时也提高了代码的可维护性。

下面为问卷读取数据关键代码：

`<!-- 显示填写问卷调查网页 -->`

```
<table cellpadding="2" cellspacing="2">
  <tr>
    <td colspan="2">
      ${sta.index+1 }: ${toupiao.title }
    </td>
  </tr>

  <c:forEach items="${toupiao.xuanxiangList}" var="xuanxiang" varStatus="ss">

    <tr>
      <td>
        <input type="radio" name="${toupiao.id}" value="${xuanxiang.id}" style="border:
0px;"/>${xuanxiang.xuanxiangneirong}<br/>
      </td>
    </tr>
  </c:forEach>
```

5.2 数据库的连接原理

使用 Hibernate 连接数据库，对于应用程序来说，所有底层的 JDBC 都被抽象了，Hibernate 会替你照管所有的细节，只需要操作对象，不需要编写 SQL 语句：

5.2.1 为不同的对象建立实体类.

这个类的属性用来与数据库建立联系。下图为对象-管理员的实体类，对应着数据库里的 admin 表

```

public class TAdmin implements java.io.Serializable {

    private Integer userId;
    private String userName;
    private String userPw;

    public TAdmin() {
    }

    public TAdmin(String userName, String userPw) {
        this.userName = userName;
        this.userPw = userPw;
    }

    public String getUserName() {
        return this.userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public String getUserPw() {
        return this.userPw;
    }

    public void setUserPw(String userPw) {
        this.userPw = userPw;
    }
}

```

图 5.2.1

5.2.2 spring 配置建立连接

Java 的对 MySQL 的数据库必须是要通过 JDBC 来建立连接，交给 spring 配置文件——applicationContext.xml 文件来操作，建立实体类与数据库的映射如下图

```

<beans
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www

    <bean id="dataSource"
        class="org.apache.commons.dbcp.BasicDataSource">
        <property name="driverClassName"
            value="com.mysql.jdbc.Driver">
        </property>
        <property name="url"
            value="jdbc:mysql://localhost:3306/db_toupiao?useUnicode=true&

        </property>
        <property name="username" value="root"></property>
        <property name="password" value=""></property>
    </bean>
    <bean id="sessionFactory"
        class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
        <property name="dataSource">
            <ref bean="dataSource" />
        </property>
        <property name="hibernateProperties">
            <props>
                <prop key="hibernate.dialect">
                    org.hibernate.dialect.SQLServerDialect
                </prop>
                <prop key="hibernate.show_sql">true</prop>
            </props>
        </property>
    </bean>

```

图 5.2.2

5.3 dwr 的实现动态登录原理

5.3.1 建立 JS 对象。

```

<script type='text/javascript' src='<%=path %>/dwr/interface/loginService.js'></script>
<script type='text/javascript' src='<%=path %>/dwr/engine.js'></script>
<script type='text/javascript' src='<%=path %>/dwr/util.js'></script>

```

图 5.3.1

5.3.2. js 的使用- LoginService 类

从 Spring 中获取 Java 对象，将目标对象的方法转换成 JS 对象的方法——loginService,

```
<dwr>
  <allow>
    <create creator="spring" javascript="loginService">
      <param name="beanName" value="loginService"></param>
    </create>

    <create creator="spring" javascript="dwrService">
      <param name="beanName" value="dwrService"></param>
    </create>
  </allow>
</dwr>
```

```
public class loginService
{
    private TAdminDAO adminDAO;
    private TUserDAO userDAO;
    public TAdminDAO getAdminDAO()
    {
        return adminDAO;
    }
    public void setAdminDAO(TAdminDAO adminDAO)
    {
        this.adminDAO = adminDAO;
    }
    public TUserDAO getUserDAO()
    {
        return userDAO;
    }
    public void setUserDAO(TUserDAO userDAO)
    {
        this.userDAO = userDAO;
    }
}
```

图 5.3.2

5.5 总体功能模块

1. 描述：系统主页面：左方页面展示了管理员可操作的六大功能，进入相关的管理页面可以链接到子菜单，并且高亮显示，每个管理模块下面都有相应的子菜单。

2. 程序运行效果图



图 5.5 管理员主页面

在每个页面将会对相关用户进行拦截操作，这样可以提高安全性，防止用户不经过登陆页面而进入任何子菜单页面：

```
if(session.getAttribute("user")==null)
{
    out.print("<script>alert(' 请先登录!');window.open('../index.J2EE','_self')</script>");
}
```

5.5.1 问卷信息管理

1. 描述：管理员输入问卷信息后点击录入按钮，如果是没有输入完整的问卷信息，都会给出相应的错误提示，不能录入成功。输入数据都通过 form 表单中定义的方法 onsubmit="return checkForm()"来检查，checkForm() 函数中是各种的校验输入数据的方式。

2. 程序效果图如下图 4.3 所示

问卷信息管理	
名称:	<input type="text"/>
	<input type="button" value="提交"/> <input type="button" value="重置"/>

图 5.5.1 问卷信息添加

5.5.1.2 问卷信息管理

1. 描述：管理员点击左侧的菜单“问卷信息管理”，页面跳转到问卷信息管理界面，调用后台的 action 类查询出所有的问卷信息，并把这些信息封转到数据集合 List 中，绑定到 request 对

象，然后页面跳转到相应的 jsp，显示出问卷信息。

2. 程序效果图如下图所示

问卷信息管理			
序号	名称	发布时间	操作
1	小学期杀我	2019-09-04	删除 问卷预览 添加题目 调查结果
2	沃兹基德问卷平台	2019-09-01	删除 问卷预览 添加题目 调查结果
3	追星	2019-09-01	删除 问卷预览 添加题目 调查结果

添加

图 6.2.1.2 问卷信息管理

问卷信息管理关键代码：

```
public String wenjuanAdd()
{
    TWenjuan wenjuan=new TWenjuan();

    wenjuan.setId(String.valueOf(new Date().getTime()));
    wenjuan.setMingcheng(mingcheng);
    wenjuan.setShijian(new SimpleDateFormat("yyyy-MM-dd").format(new Date()));

    wenjuanDAO.save(wenjuan);
    this.setMessage("添加完毕");
    this.setPath("wenjuanMana.action");
    return "succeed";
}

public String wenjuanMana()
{
    String sql="from TWenjuan";
    List wenjuanList =wenjuanDAO.getHibernateTemplate().find(sql);
    Map request=(Map)ServletActionContext.getContext().get("request");
    request.put("wenjuanList", wenjuanList);
    return ActionSupport.SUCCESS;
}

public String wenjuanDel()
{
    TWenjuan wenjuan=wenjuanDAO.findById(id);
    wenjuanDAO.delete(wenjuan);
    this.setMessage("删除完毕");
    this.setPath("wenjuanMana.action");
    return "succeed";
}
```

5.5.1.3 问卷信息删除

1. 描述：先是点击问卷信息管理，页面跳转到问卷信息管理界面，浏览所有的问卷信息，点击要删除的问卷信息，点击确认按钮，完成问卷删除操作。

2. 程序效果图如下图所示

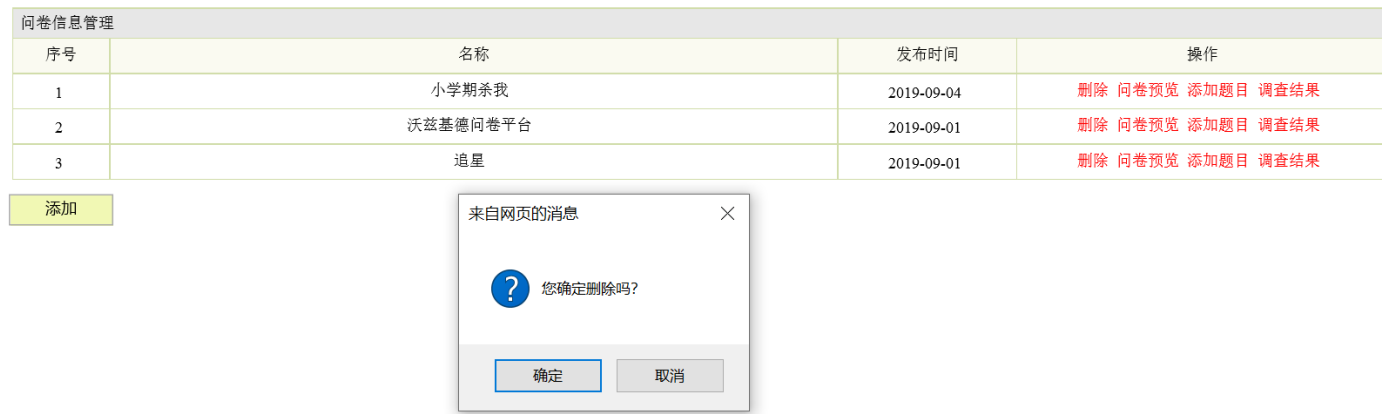


图 5.5.1.3 问卷信息删除

5.5.2 问卷题目管理

5.5.2.1 问卷题目添加

1. 描述：点击问卷管理，显示所有的问卷信息，然后点击“添加题目”超练级，系统自动弹出添加题目的页面，输入题目信息后点击录入按钮，如果是没有输入完整的题目信息，都会给出相应的错误提示，不能录入成功。输入数据都通过 form 表单中定义的方法 onsubmit="return checkForm()"来检查，checkForm() 函数中是各种的校验输入数据的方式。

2. 程序效果图如下图 4.7 所示：

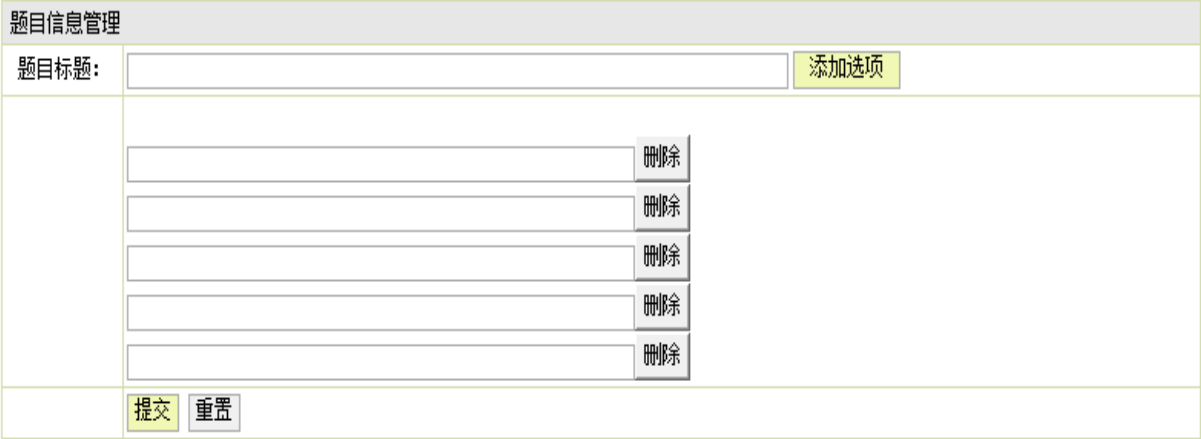


图 5.5.2.1 题目信息受理

5.5.2.2 问卷信息预览

1. 描述：点击问卷预览按钮，系统自动显示改问卷的题目信息。
2. 程序效果图如下图 4.8 所示



图 5.5.2.2 问卷预览界面

问卷题目管理关键代码：

```
public String toupiaoAdd()  
{  
    HttpServletRequest request=ServletActionContext.getRequest();  
  
    String id=String.valueOf(new Date().getTime());  
    String title=request.getParameter("title");  
    String wenjuan_id=request.getParameter("wenjuan_id");  
    TToupiao toupiao=new TToupiao();  
  
    toupiao.setId(id);  
    toupiao.setTitle(title);  
    toupiao.setWenjuan_id(wenjuan_id);  
  
    toupiaoDAO.save(toupiao);  
  
    String[] xuanxiangneirong=request.getParameterValues("xuanxiangneirong");  
    for(int i=0;i<xuanxiangneirong.length;i++)  
    {  
        try  
        {  
            Thread.sleep(100);  
        }  
    }  
}
```

```
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }
        System.out.println(xuanxiangneirong[i]+"%%");
        toupiaoxuanxiangSave(xuanxiangneirong[i], id);//这个地方的id是投票的id
    }

    request.setAttribute("msg", "添加完毕");
    return "msg";
}

public void toupiaoxuanxiangSave(String xuanxiangneirong,String toupiao_id)
{
    TToupiaoxuanxiang toupiaoxuanxiang=new TToupiaoxuanxiang();

    toupiaoxuanxiang.setId(String.valueOf(new Date().getTime()));
    toupiaoxuanxiang.setXuanxiangneirong(xuanxiangneirong);
    toupiaoxuanxiang.setPiaoshu(0);
    toupiaoxuanxiang.setToupiao_id(toupiao_id);

    toupiaoxuanxiangDAO.save(toupiaoxuanxiang);
}
```

5.5.3 调查结果统计

1. 描述：选择一条问卷信息，点击“结果统计”，系统调转到统计页面。
2. 程序效果图如下图所示：

沃兹基德问卷平台

☐ 6分-8分 (1)

☐ 8分以上 (1)

☐ 6分以下 (2)

☐ 棒极了 (2)

☐ 一般般咯 (0)

☐ 我喜欢的 (2)

☐ 不咋样 (1)

图 6.2.3 调查结果统计

统计结果关键代码：

```
String path = request.getContextPath();
List xuanxiangList=(List)request.getAttribute("xuanxiangList");

DefaultCategoryDataset categoryDataset=new DefaultCategoryDataset();
for(int i=0;i<xuanxiangList.size();i++)
{
    TToupiaoxuanxiang toupiaoxuanxiang=(TToupiaoxuanxiang)xuanxiangList.get(i);

categoryDataset.setValue(toupiaoxuanxiang.getPiaoshu(),"",toupiaoxuanxiang.getXuanxian
gneirong()+"("+toupiaoxuanxiang.getPiaoshu()+")" );
}
```

5.5.4 系统公告管理

5.5.4.1 公告发布

1. 描述：输入公告的标题和内容点击发布即可。



公告添加

标题:

内容:

B *I* |

请输入内容

2.程序效果图

图 6.2.4.1 公告发布

3.从<http://java.fckeditor.net>网站上面下载的一个在线编辑器。下载相关文件之后导入到webroot文件夹下面，记事本打开ewebeditor.htm文件，将config.ServerExt = "asp";改为config.ServerExt = "jsp";最后在发布信息页面引入<%@ taglib uri="http://java.fckeditor.net" prefix="FCK"%>。

5.5.4.2 公告管理

1. 描述：管理员点击左侧的菜单“公告管理”，页面跳转到公告管理界面，调用后台的公告

Action 类查询出所有的公告信息，并把这些信息封转到数据集合 List 中，绑定到 request 对象，然后页面跳转到相应的 jsp，显示出公告信息。

2.程序效果图

公告信息管理				
序号	公告标题	发布时间	公告内容	操作
1	小学期杀我	2019-9-4 22:54:22	公告内容	删除
2	英国脱欧关键一周！议会与首相的权力拉锯战	2019-9-1 21:41:45	公告内容	删除
3	三大运营商取消“不限量套餐”：流量收入已成负增长	2019-9-1 21:40:52	公告内容	删除
4	王一博和他的摩托车	2019-9-1 18:36:25	公告内容	删除

添加公告

图 5.5.4.2 公告管理

公告管理关键代码

```
public String gonggaoAdd()
{
    TGonggao gonggao=new TGonggao();
    gonggao.setGonggaoTitle(gonggaoTitle);
    gonggao.setGonggaoContent(gonggaoContent);
    gonggao.setGonggaoData(new Date().toLocaleString());
    gonggaoDAO.save(gonggao);
    this.setMessage("公告添加完毕");
    this.setPath("gonggaoMana.action");
    return "succeed";
}

public String gonggaoMana()
{
    List gonggaoList =gonggaoDAO.findAll();
    Map request=(Map)ServletActionContext.getContext().get("request");
    request.put("gonggaoList", gonggaoList);
    return ActionSupport.SUCCESS;
}

public String gonggaoDel()
{
    TGonggao gonggao=gonggaoDAO.findById(gonggaoId);
    gonggaoDAO.delete(gonggao);
    this.setMessage("公告删除完毕");
    this.setPath("gonggaoMana.action");
    return "succeed";
}
```

```
public String gonggaoDetail()  
{  
    TGonggao gonggao=gonggaoDAO.findById(gonggaoId);  
    Map request=(Map)ServletContext.getContext().get("request");  
    request.put("gonggao", gonggao);  
    return ActionSupport.SUCCESS;  
}
```

5.5.5 系统登陆页面实现

1. 描述：为了保证系统的安全性，要先使用本系统必须先登陆到系统中，用户需要正确的账号和密码登录本系统。

2. 程序运行效果图 示：



图 5.5.5 系统登陆页面设计

3. 在登陆页面输入用户名和密码以，点击提交按钮，跳转到登陆的 service 中，在该 service 中会对用户名，密码进行判断，验证通过进入对应的页面，loginService 关键代码：

```
public String login(String userName,String userPw,int userType)  
{  
    String result="no";  
    if(userType==0)//系统管理员登陆  
    {  
        String sql="from TAdmin where userName=? and userPw=?";
```



```
Object[] con={userName,userPw};
List adminList=adminDAO.getHibernateTemplate().find(sql,con);
if(adminList.size()==0)
{
    result="no";
}
else
{
    WebContext ctx = WebContextFactory.get();
    HttpSession session=ctx.getSession();
    TAdmin admin=(TAdmin)adminList.get(0);
    session.setAttribute("userType", 0);
    session.setAttribute("admin", admin);
    result="yes";
}
}
if(userType==1)//putongyuangong
{
}
}
```

5.5.6 修改个人密码

1. 描述：输入用户名和用户的原有密码，输入新密码以后点击修改按钮即可修改密码成功。
2. 程序效果图 所示：

密码修改	
登录名：	<input type="text" value="a"/>
原密码：	<input type="password"/>
新密码：	<input type="password"/>
确认密码：	<input type="password"/>
<input type="button" value="修改"/> <input type="button" value="重置"/>	

图 5.5.6 修改密码

5.5.7 安全退出系统

1. 描述：点此按钮回到系统的主页面。
2. 关键代码：主要是通过 javascript 语句来实现，`item_word[8][4]="退出系统";`
`item_link[8][4]="javascript:window.open('../index.J2EE','_self')";`

5.5 普通用户模块

5.5.1 系统主页面实现

1. 描述：页面的上方显示的是该网站的 logo 和当前的时间，下方主要部分是信息显示区域，右侧部分是通知公告，管理员发布的信息公告都可以在此点击后链接到相应的信息页面查看具体通知公告，右上方是登陆模块，用户必须输入正确的用户名、密码和验证码后才能进入系统。
2. 程序运行效果图 所示：



图 5.5.1 系统主页面设计

问卷列表显示关键代码：

```
function wenjuanDetailQian(id)
{
    var strUrl = "<%=path %>/wenjuanDetailQian.action?id="+id;
    var ret = window.showModalDialog(strUrl,"", "dialogWidth:800px;
dialogHeight:600px; dialogLeft: status:no;
directories:yes;scrollbars:yes;Resizable=no;");
}
```

```
<input type="radio" name="$ {toupiao.id}" value="$ {xuanxiang.id}" style="border:
0px;"/>$ {xuanxiang.xuanxiangneirong}<br/>
```

5.5.2 用户注册

1. 描述：对于游客，至于浏览的功能，如果想参与问卷调查，首先需要注册一个合法的账号，

然后登陆后，方能使用，点击导航栏菜单“免费注册”，调用 js 的 open() 方法，弹出注册页面，输入相关正确信息后点击注册按钮即可完成注册，如果是没有输入完整的信息，或都会给出相应的错误提示，不能录入成功。输入数据都通过 form 表单中定义的方法 onsubmit="return checkForm()"来检查，checkForm() 函数中是各种的校验输入数据的方式。

2. 程序效果图 所示：

用 户 注 册	
用户名：	<input type="text"/>
密 码：	<input type="password"/>
性别：	<input checked="" type="radio"/> 男 <input type="radio"/> 女
E-mail：	<input type="text"/>
<input type="button" value="确定"/> <input type="button" value="取消"/>	

图 5.5.2 用户注册界面设计

用户注册关键代码

```
public String userReg()  
{  
    TUser user=new TUser();  
    user.setUserName(userName);  
    user.setUserPw(userPw);  
    user.setUserTel(userTel);  
    user.setUserEmail(userEmail);  
    user.setUserSex(userSex);  
    user.setUserDel("no");  
    userDAO.save(user);  
    Map session= ServletActionContext.getContext().getSession();  
    session.put("user", user);  
    return "successAdd";  
}
```

5.5.3 参与问卷调查

1. 描述： 点击投票名称， 页面跳转到具体的投票页面。

2.程序效果图

图 5.5.3 参与问卷界面设计

参与问卷关键代码

```
<form action="<%=path %>/wenjuan_canyu.action" name="form1" method="post">
    <table cellpadding="2" cellspacing="2" align="center">
        <tr align="center"><td align="center" style="font-size:
22px;">${requestScope.wenjuan.mingcheng}</td></tr>
    </table>

    <c:forEach items="${requestScope.wenjuan.toupiaoList}" varStatus="sta"
var="toupiao">
        <table cellpadding="2" cellspacing="2">
            <tr>
                <td colspan="2">
                    ${sta.index+1 }: ${toupiao.title }
                </td>
            </tr>
            <c:forEach items="${toupiao.xuanxiangList}" var="xuanxiang" varStatus="ss">
                <tr>
                    <td>
                        <input type="radio" name="${toupiao.id}" value="${xuanxiang.id}"
style="border: 0px;"/>${xuanxiang.xuanxiangneirong}<br/>
                    </td>
                </tr>
            </c:forEach>
        </table>
    <br/>
</c:forEach>

<table width="99%" border="0" cellpadding="2" cellspacing="1" bgcolor="#FFFFFF">
```

```
align="center" style="margin-top:8px">
    <tr align='center' bgcolor="#FFFFFF" height="22">
        <td width="100%" bgcolor="#FFFFFF" align="left">
            <input type="hidden" name="xxxxx" id="xxxxx" value="">
            <input type="button" value="提交" onclick="check()"/>&nbsp;
        </td>
    </tr>
</table>
</form>
```