## “投票系统”程序技术说明文档

姓名： 陈铭海 学号： 20162180198 班别： 软件5班

**一．设计方案**（功能组成、界面组成、算法、数据结构等等）

1. **投票系统的功能组成**

投票系统功能有：选择投票和个人操作及设置，投票设置，投票结果分析，投票操作保障。详细功能如下：

(0):选择投票和个人操作及设置

* 列出所有正在进行的投票活动的简略信息，供投票人选择：（也应可查看已有结果的投票活动情况）列出信息有：投票活动的名称、发起人、投票人数、投票时间等
* 点击选择某投票活动后，列出该次投票活动的详细内容介绍，可选择进入投票
* 进行投票操作：要先输入验证信息（如系统自动检测该机IP地址是否合法，检测投票人是否合法，检测验证码是否合法等），如无错误才可以进行投票，按投票规则填写投票提交表，提交完成投票
* 设置及记录服务器地址：即让客户端机器能顺利连接到服务器端，并且该客户端IP地址合法等
* 个人资料管理：是投票人的个人资料登记、密码设置等，可参考论坛式的注册、登陆、管理模式

(1):投票设置

* 设置候选人资料：

填写个人信息（包括姓名、性别、年龄、民族、出生年月日、政治面貌、家庭住址、联系方式等……）

编码：由系统自动生成，投票前后需一致

推荐意见：事迹介绍、或者是个人介绍等

* 设置投票时间：设置该次投票的开始及结束时间，投票人只能在投票有效期间投票
* 设置投票类型、投票规则：如该次投票是单选、还是多选、（选多少人）、还是评分制（最低分、最高分多少）、是否可投弃权票、是否可投反对票、多选最终选出多少人、评分制最终选出多少人等
* 设置合法投票者：设置投票机器的IP、投票人的名单等

(2):投票结果分析

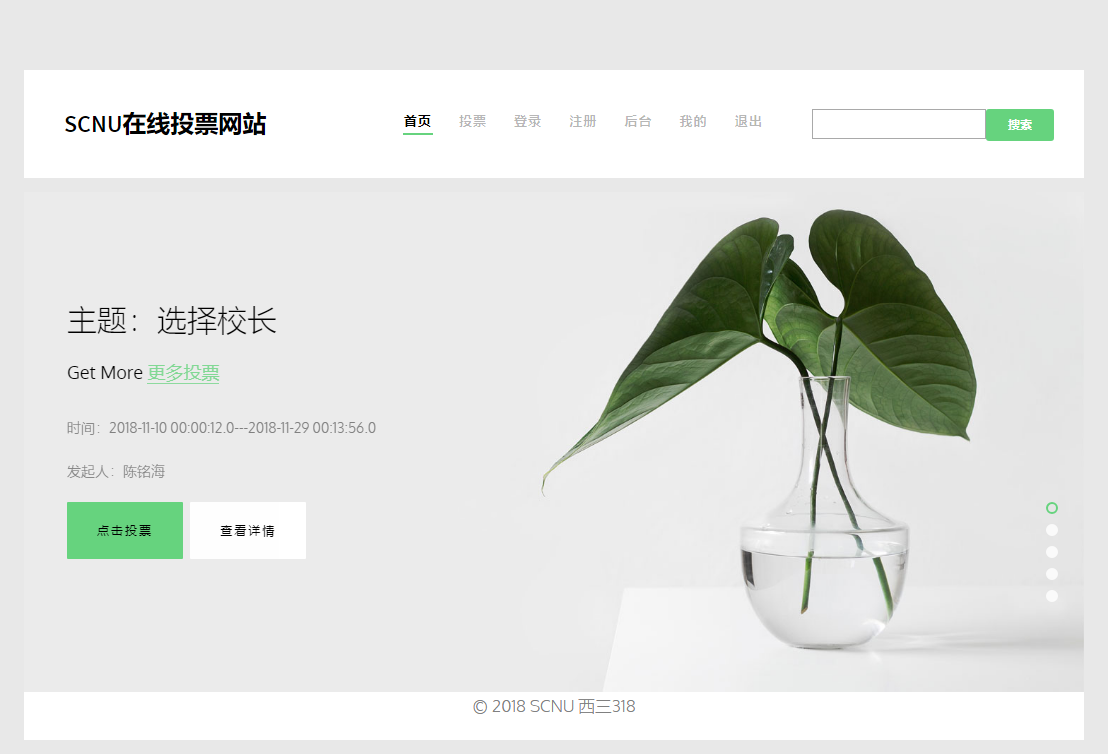
* 排名结果：单选的结果、多选的结果（按得票数排列，胜出人显著显示）、评分制结果（按分数排列，胜出人显著显示）
* 投票的统计信息：（需要做到实时变化以及最终结果显示）包括投票的剩余时间、投票的人数情况等

(3):投票操作保障

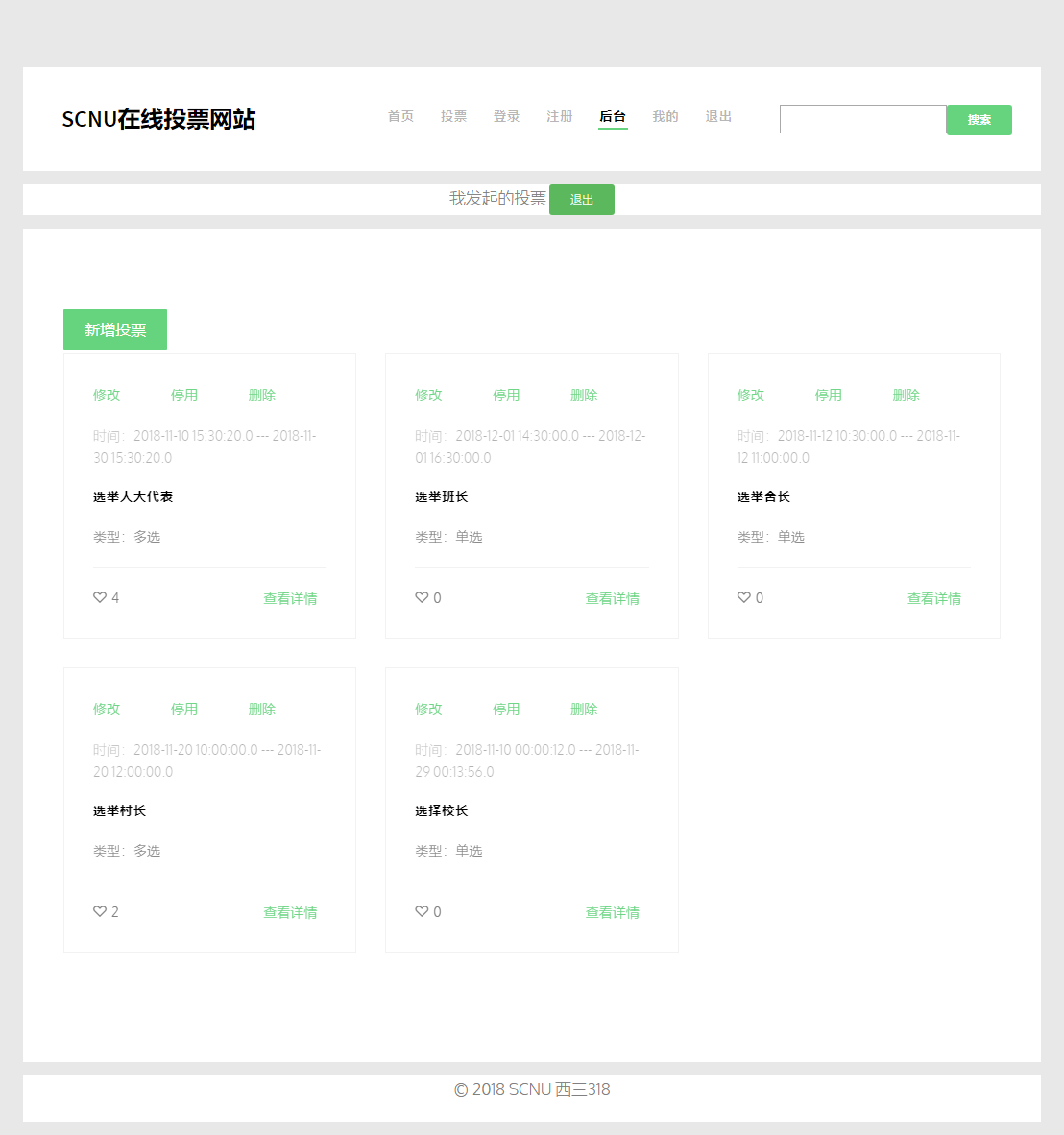
* IP验证：验证投票人的机器IP地址是否合法
* 时间验证：投票时间的控制，时间到即结束该次投票活动（对局域网,可不用考虑延时问题; 但如果是基于internet的投票,要考虑: 即客户端投票时,还在有效投票时间内,但数据传到服务器端, 已经过了有效投票时间, 这时应该如何计算? 如果要使得系统设计得更合理, 希望能实现按投票当时的时间,而不是按数据到达时间, 又要防止客户端在时间上欺骗, 应该如何设计?）
* 投票人验证：投票人是本系统用户，但要验证其是否享有对某次投票活动的投票权利，并且验证其帐号、密码的正确性，不可多次投票
* 投票对象验证：所投的人是否存在于候选人列表中，或是否符合本次投票活动规则（因为某些投票活动可另填自己认为可以的候选人），如不符合是否当弃权处理
* 验证码验证：防止利用软件连续投票，或自动投票

1. **投票系统的界面组成**

(1).投票系统前台界面(浏览器截图)：



(2).投票系统后台界面(浏览器截图)：



1. **投票系统的算法、数据结构**

(1).投票系统前端：

前端主要用到了盒子模型，使用到的数据结构主要是数组，集合来用于存储从数据库查询到的投票列表以及候选人信息。

(2).投票系统后台：

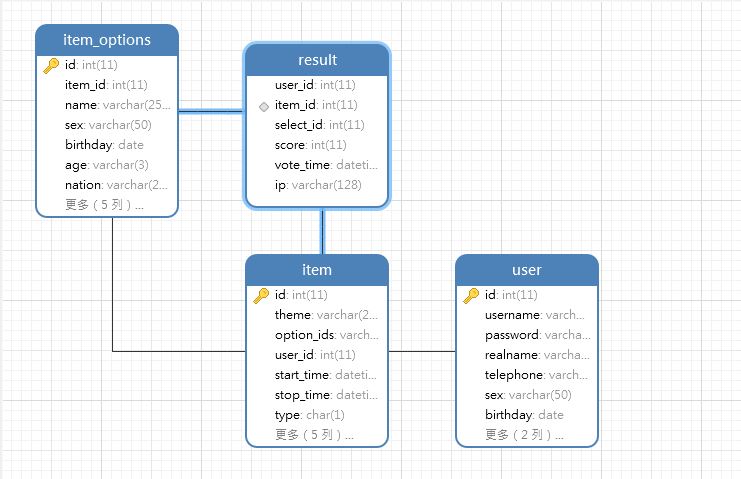
后台使用到的算法主要是查找算法，例如从数组或集合中查找到该用户，或者查找到用户点击的投票选项。数据结构使用了Map，ArrayList来存储提示信息和用户session。

**二．开发过程**（如何设计、如何实现、如何检查系统功能）

1. **投票系统的设计**

1.数据库设计

根据实验要求，抽取出四个数据库表，分别是用户表(user)，投票活动表(item)，候选人信息表(item\_options)，投票结果表(result)。其中投票活动表和候选人表是一对多关系，投票活动表和投票结果表也是一对多关系，用户表和投票活动表是一对一关系。四个表的ER图如下所示：



2.系统架构设计

本投票系统使用MVC架构，使用JSP+HTML+JQuery+CSS作为视图层，Servlet作为控制器，JDBC+MySQL作为数据模型层。架构图如下：

数据访问层

控制器

数据库

用户

视图

3.页面设计

使用盒子模型，分为上中下三个盒子，其中数据显示主要在中间区域。设计如下：

1. **投票系统的实现**

1.实现技术

利用Java语言，使用JavaWeb技术体系中的：JSP+Servlet+JDBC+MySQL技术，来开发一个web端的投票系统。

2.数据访问层实现(请看db包和dao包下的所有文件。这里只贴DB.java)

public class DB {  
 public static final String *DB\_DRIVER* = "com.mysql.jdbc.Driver";  
 public static final String *DB\_URL* = "jdbc:mysql://127.0.0.1:3306/db\_vote?characterEncoding=utf-8";  
 public static final String *USER* = "root";  
 public static final String *PWD* = "123456";  
 private Connection con = null;  
 private Statement stmt = null;  
 private PreparedStatement pstmt = null;  
  
 public DB() {  
 createConnection();  
 try {  
 stmt = con.createStatement();  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 }  
 public DB(String preparedSql) {  
 createConnection();  
 try {  
 pstmt = con.prepareStatement(preparedSql,Statement.*RETURN\_GENERATED\_KEYS*);  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 }  
 private void createConnection() {  
 try {  
 Class.*forName*(*DB\_DRIVER*);  
 con = DriverManager.*getConnection*(*DB\_URL*, *USER*, *PWD*);  
 } catch (ClassNotFoundException ex) {  
 System.*err*.println("Error: 类不存在!" + ex.getMessage());  
 } catch (SQLException ex) {  
 System.*err*.println("Error: 连接数据库失败!" + ex.getMessage());  
 }  
 }  
 public ResultSet select(String sql) {  
 ResultSet rs = null;  
 try {  
 rs = stmt.executeQuery(sql);  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 return rs;  
 }  
 //preparedStatement的查询方法  
 public ResultSet select() {  
 ResultSet rs = null;  
 try {  
 rs = pstmt.executeQuery();  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 return rs;  
 }  
 public int update(String sql) {  
 int result = 0;  
 try {  
 result = stmt.executeUpdate(sql);  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 return result;  
 }  
 //preparedStatement的更新  
 public int update() {  
 int result = 0;  
 try {  
 result = pstmt.executeUpdate();  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 return result;  
 }  
 public int getInsertId(){  
 int autoInckey = -1;  
 ResultSet rs = null; // 获取结果  
 try {  
 rs = pstmt.getGeneratedKeys();  
 if (rs.next()){  
 autoInckey = rs.getInt(1);  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
  
 return autoInckey;  
 }  
 //以下方法为使用动态SQL语句方式时，设置prestmt的参数的方法  
 //其他类型的参数对应的方法,请自行补充  
 public void setString(int index, String value) {  
 try {  
 pstmt.setString(index, value);  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 }  
 public void setInt(int index, int value) {  
 try {  
 pstmt.setInt(index, value);  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 }  
 public void setLong(int index, long value) {  
 try {  
 pstmt.setLong(index, value);  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 }  
 public void setDouble(int index, double value) {  
 try {  
 pstmt.setDouble(index, value);  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 }  
 public void close() {  
 try {  
 if (stmt != null) {  
 stmt.close();  
 }  
 if (pstmt != null) {  
 pstmt.close();  
 }  
 if (con != null) {  
 con.close();  
 }  
 } catch (SQLException ex) {  
 System.*err*.println("Error: " + ex.getMessage());  
 }  
 }  
}

3.业务逻辑层实现(请看service包下的所有(14个)文件。这里只贴首页的控制器代 码IndexServlet.java以及点击投票的代码IndexVoteIdServlet.java)

（1）.IndexServlet.java

@WebServlet({"/index","/index/vote/list","/index/vote/search"})  
public class IndexServlet extends HttpServlet {  
  
 private static final long *serialVersionUID* = 7209195686057464382L;  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 //由于get传递参数要乱码，所以弄成post，这个问题之后在解决  
 //System.out.println("搜索投票");  
 request.setCharacterEncoding("UTF-8");  
 ItemDao itemDao = new ItemDao();  
 String path = request.getRequestURI();  
 List itemList;  
 if (path.indexOf("/search") > -1){  
 String content = request.getParameter("content").trim();  
 itemList = itemDao.getSearchItemList(content);  
 request.setAttribute("itemList",itemList);  
 request.setAttribute("active",2);  
 request.getRequestDispatcher("/WEB-INF/view/vote.jsp").forward(request,response);  
 }  
 }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 HttpSession session = request.getSession();  
 request.setCharacterEncoding("UTF-8");  
 ItemDao itemDao = new ItemDao();  
 String path = request.getRequestURI();  
 List itemList;  
 if (path.indexOf("vote/list") > -1){  
 session.setAttribute("active",2);  
 itemList = itemDao.getAllItemList();  
 request.setAttribute("itemList",itemList);  
 request.getRequestDispatcher("/WEB-INF/view/vote.jsp").forward(request,response);  
 }else {  
 session.setAttribute("active",1);  
 itemList = itemDao.getLimitItemList(5);//获取最新五条  
 request.setAttribute("itemList",itemList);  
 request.getRequestDispatcher("/WEB-INF/view/index.jsp").forward(request,response);  
 }  
 }  
}

（2）.IndexVoteIdServlet.java

@WebServlet(name = "IndexVoteIdServlet",urlPatterns = "/index/vote")  
public class IndexVoteIdServlet extends HttpServlet {  
   
 private static final long *serialVersionUID* = -6818273018879095397L;  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 HttpSession session = request.getSession();  
 int id = Integer.*parseInt*(request.getParameter("id"));  
 String[] selectIds = request.getParameterValues("answer");  
 String ip = this.getRemortIP(request);  
 ResultDao resultDao = new ResultDao();  
 User user = (User)session.getAttribute("user");  
 int success = resultDao.add(user.getId(),id,selectIds,ip);  
 if (success > 0){  
 session.setAttribute("msg","投票成功");  
 response.sendRedirect(request.getContextPath() + "/index/vote/list");  
 }else{  
 session.setAttribute("msg","投票失败，请不要用同一ip投票");  
 response.sendRedirect(request.getContextPath() + "/index/vote?id=" + id);  
 }  
 }  
  
 @SuppressWarnings("rawtypes")  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 HttpSession session = request.getSession();  
 session.setAttribute("active",2);  
 int id = Integer.*parseInt*(request.getParameter("id"));  
 ItemDao itemDao = new ItemDao();  
 OptionDao optionDao = new OptionDao();  
 Item item = itemDao.getItemById(id);  
 if (itemDao.getVoteStatus(item.getStartTime(),item.getStopTime()) != 2){  
 //判断是否可以投票，即投票是否结束，防攻击  
 response.sendRedirect(request.getContextPath() + "/index/vote/list");  
 return;  
 }  
 List optionList = optionDao.getOptionList(item.getId(),item.getAllVoteCount());  
 request.setAttribute("item",item);  
 request.setAttribute("optionList",optionList);  
 String isWaiver="不可以弃权";  
 String isOppose="不可以投反对票";  
 System.*out*.println(item.getIsWaiver());  
 System.*out*.println(item.getIsOppose());  
 if( "1".equals(item.getIsWaiver()) ){  
 isWaiver="可以弃权";  
 }  
 if( "1".equals(item.getIsOppose()) ){  
 isOppose="可以投反对票";  
 }  
 if( "1".equals(item.getType()) ) {  
 request.setAttribute("type","(单选，"+isWaiver+"，"+isOppose+")");  
 }else if( "2".equals(item.getType()) ){  
 request.setAttribute("type","(多选，可选人数:"+item.getNumber()+"，"+isWaiver+"，"+isOppose+")");  
 }else {  
 request.setAttribute("type","(评分制，可选人数:"+item.getNumber()+"，"+isWaiver+"，"+isOppose+")");  
 }  
 request.getRequestDispatcher("/WEB-INF/view/info.jsp").forward(request,response);  
 }  
 */\*获取用户的ip，防止多次投票\*/* private String getRemortIP(HttpServletRequest request) {  
 if (request.getHeader("x-forwarded-for") == null) {  
 return request.getRemoteAddr();  
 }  
 return request.getHeader("x-forwarded-for");  
 }  
}

4.表现层实现(请查看../WebContent/WEB-INF/view下的所有(14个)文件,这里只贴添 加投票的页面代码：addvote.jsp)

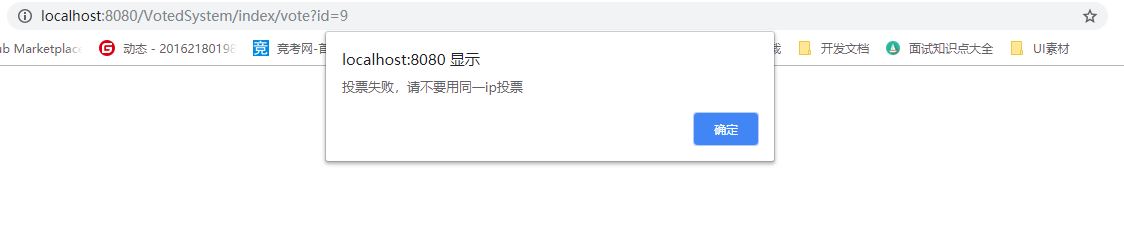
<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<%@**include** file="head.jsp"%>  
<div class="container-wrap">  
 <footer>  
 <div class="row copyright" >  
 <div class="col-md-12 text-center">  
 <big>新增投票</big>  
 </div>  
 </div>  
 </footer>  
</div><!-- END container-wrap -->  
<div class="container-wrap">  
 <div id="fh5co-contact">  
 <div class="row">  
 <div class="col-md-2 col-md-push-1 animate-box">  
 </div>  
 <div class="col-md-6 col-md-push-1 animate-box">  
 <form action="**${**pageContext.request.contextPath**}**/admin/vote/add" method="post" id="vote\_add\_form">  
 <div class="row">  
 <div class="col-md-12">  
 <div class="form-group">  
 <label for="" class="col-sm-3 control-label">主题</label>  
 <div class="col-sm-9">  
 <input type="text" name="theme" class="form-control" placeholder="">  
 </div>  
 </div>  
 </div>  
 <div class="col-md-12">  
 <div class="form-group">  
 <label for="" class="col-sm-3 control-label">发起人</label>  
 <div class="col-sm-9">  
 <input type="text" class="form-control" value="**${**user.realName**}**" placeholder="" disabled>  
 </div>  
 </div>  
 </div>  
 <div class="col-md-12">  
 <div class="form-group">  
 <label for="" class="col-sm-3 control-label">发起时间</label>  
 <div class="col-sm-9">  
 <input type="text" name="start\_time" class="form-control" placeholder="如2018-10-30 15:07:02">  
 </div>  
 </div>  
 </div>  
 <div class="col-md-12">  
 <div class="form-group">  
 <label for="" class="col-sm-3 control-label">结束时间</label>  
 <div class="col-sm-9">  
 <input type="text" name="stop\_time" class="form-control" placeholder="如2018-10-30 15:07:02">  
 </div>  
 </div>  
 </div>  
 <div class="col-md-12">  
 <div class="form-group">  
 <label for="" class="col-sm-3 control-label">投票类型</label>  
 <div class="col-sm-9">  
 <select name="type" id="" class="form-control">  
 <option value="1">单选</option>  
 <option value="2">多选</option>  
 <option value="3">评分制</option>  
 </select>  
 </div>  
 </div>  
 </div>  
   
 <div class="col-md-12">  
 <div class="form-group">  
 <label for="" class="col-sm-3 control-label">投票规则</label>  
 <div class="col-sm-9">  
 <select name="is\_waiver" class="form-control">  
 <option value ="0" selected="selected">不能弃权</option>  
 <option value ="1">可以弃权</option>  
 </select>  
 <select name="is\_oppose" class="form-control">  
 <option value ="0" selected="selected">不能反对</option>  
 <option value ="1">可以反对</option>  
 </select>  
 <input type="text" name="number" class="form-control" placeholder="多选或评分制的可选人数">  
 </div>  
 </div>  
 </div>  
   
 <div class="col-md-12">  
 <div class="form-group">  
 <label for="" class="col-sm-3 control-label">候选人名单</label>  
 <div class="col-sm-9">  
 <textarea rows="5" cols="7" name="option\_content" class="form-control" placeholder="候选人名之间用按回车键（换行）隔开,如:&#13;&#10;陈铭海&#13;&#10;李世民"></textarea>  
 </div>  
 </div>  
 </div>  
 <div class="col-md-12">  
 <div class="form-group">  
 <label for="" class="col-sm-4 control-label"></label>  
 <input type="button" id="submit1" name="select" value="提 交" class="btn btn-primary btn-modify">  
 <a href="**${**pageContext.request.contextPath**}**/admin"><input type="button" value="返 回 " class="btn btn-primary btn-modify"></a>  
 </div>  
 </div>  
 </div>  
 </form>  
 </div>  
 </div>  
 </div>  
</div><!-- END container-wrap -->  
<%@**include** file="foot.jsp"%>  
<script type="text/javascript">  
 $('#submit1').click(**function** () {  
 **var** date = /^(\d{4})-(\d{2})-(\d{2})\s+(\d{1,2}):(\d{1,2}):(\d{1,2})$/;  
 **var** theme = $('input[name=theme]').val();  
 **var** start\_time = $('input[name=start\_time]').val();  
 **var** stop\_time = $('input[name=stop\_time]').val();  
 **var** number = $('input[name=number]').val();  
 **var** option\_content = $('textarea').val();  
 **var** type= $('select[name=type]').val();  
   
 **if**(!theme || (theme.length > 255)){  
 alert('主题不能为空且长度不能超过255个字符');  
 **return**;  
 }  
 **if**(!start\_time){  
 alert('发起时间不能为空');  
 **return**;  
 }  
 **if**(!date.test(start\_time)){  
 alert('发起时间要符合日期格式');  
 **return**;  
 }  
 **if**(!stop\_time){  
 alert('结束时间不能为空');  
 **return**;  
 }  
 **if**(!date.test(stop\_time)){  
 alert('结束时间要符合日期格式');  
 **return**;  
 }  
 **if**(type !="1" && !number){  
 alert('多选或评分制类型的投票的可选人数不能为空');  
 **return**;  
 }  
 **if**(!option\_content){  
 alert('选项不能为空');  
 **return**;  
 }  
 **var** start\_date = **new** Date(start\_time);  
 **var** stop\_date = **new** Date(stop\_time);  
 **if** (stop\_date.getTime() < start\_date.getTime()){  
 alert('结束时间要比开始时间大');  
 **return**;  
 }  
 document.getElementById('vote\_add\_form').submit();  
 });  
</script>

1. **投票系统的测试**

1.黑盒测试(经过测试，发现功能基本完成)

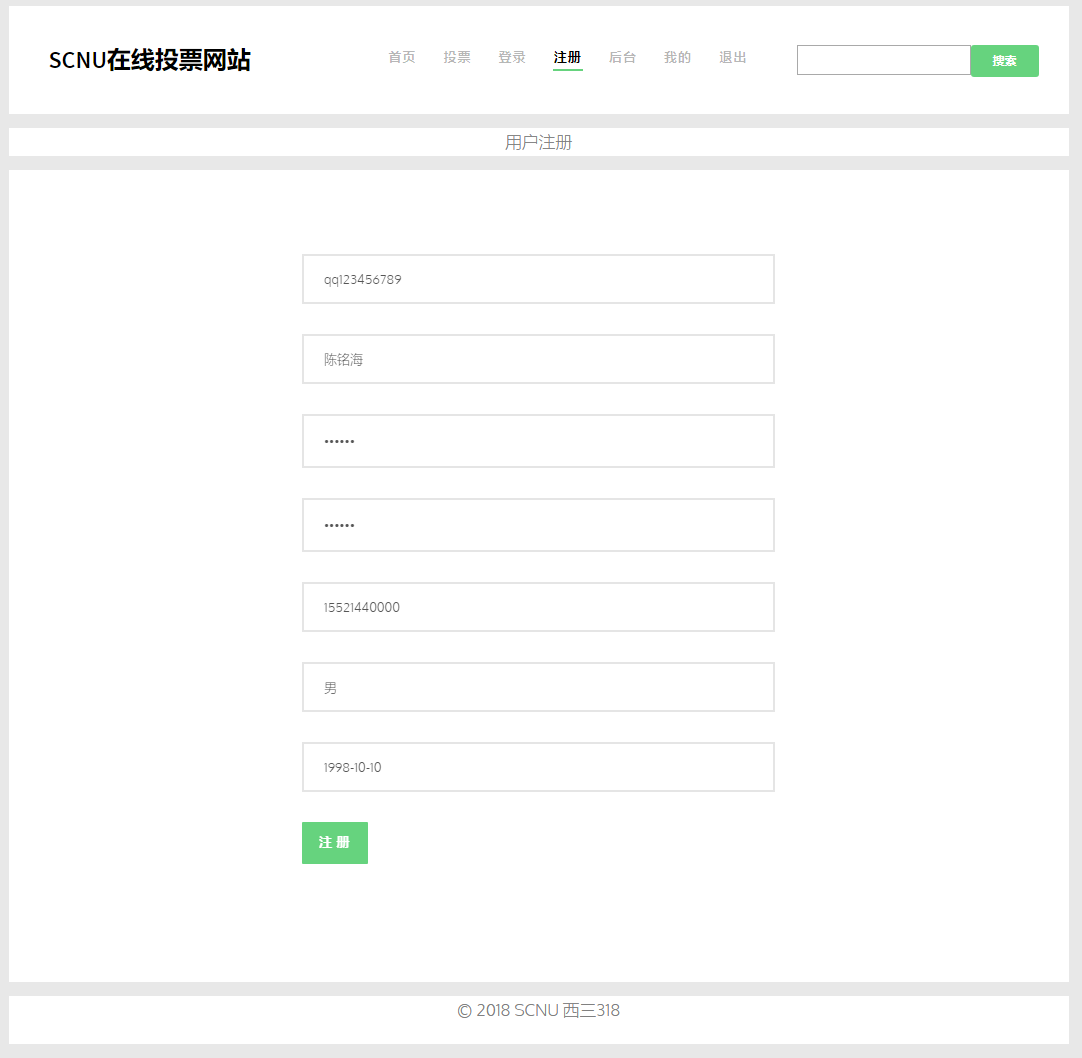
(1).安全机制(防止同一ip重复投票，未登录不能投票，非管理员不能进入后台)







(3).用户注册



(4).用户注册成功



(5).用户登录



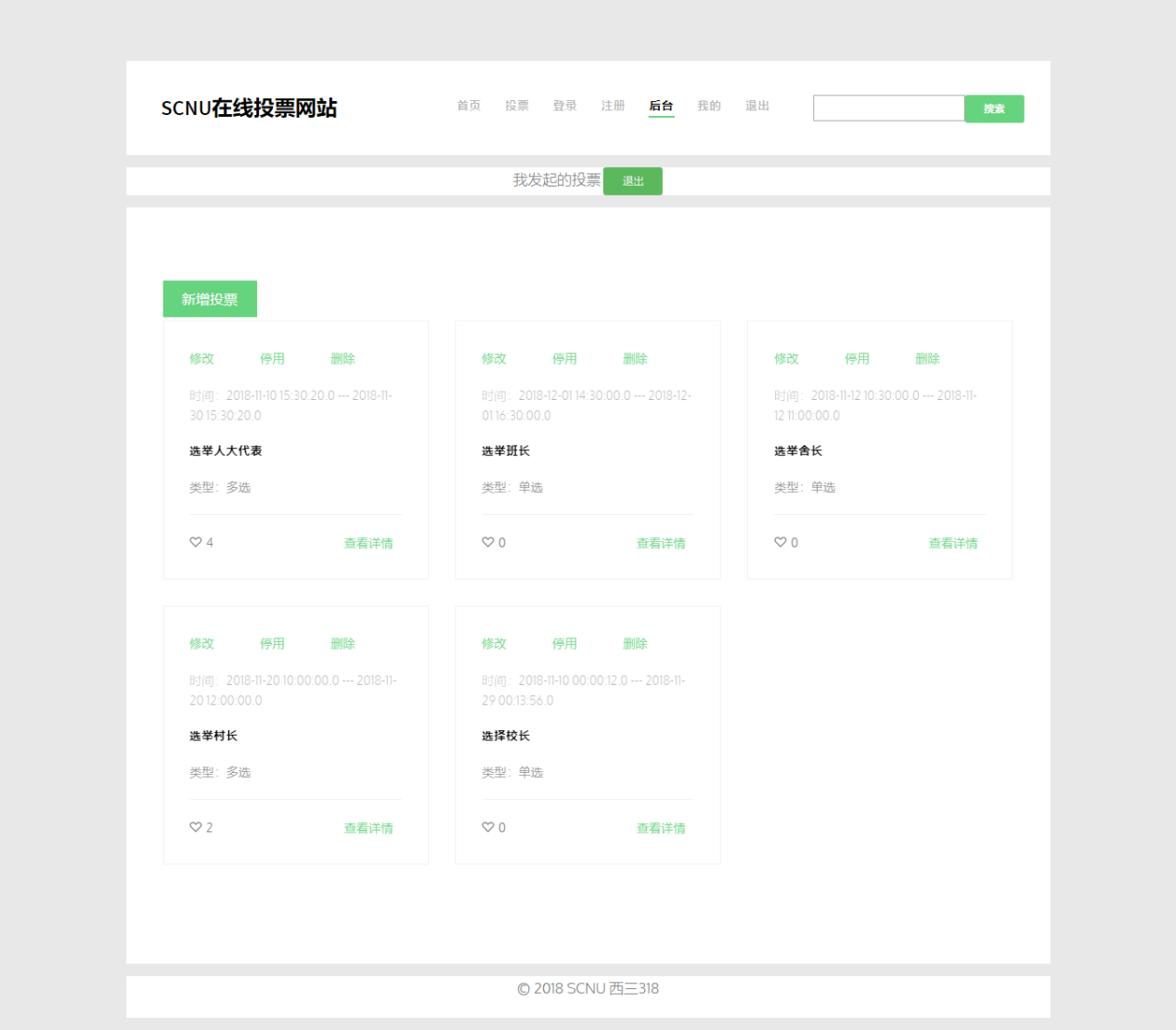
(6).用户登录成功后，查看所有投票活动



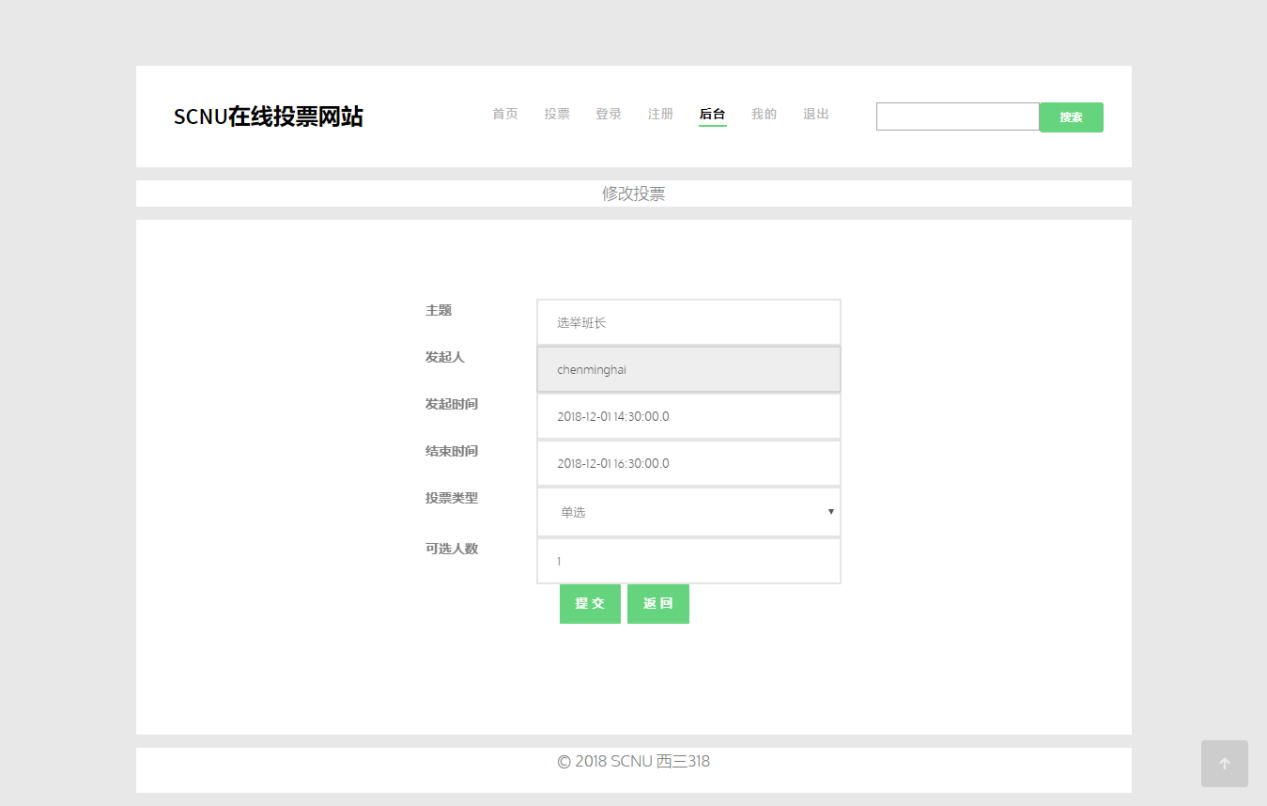
(7).查看所有投票活动后，点击进行投票



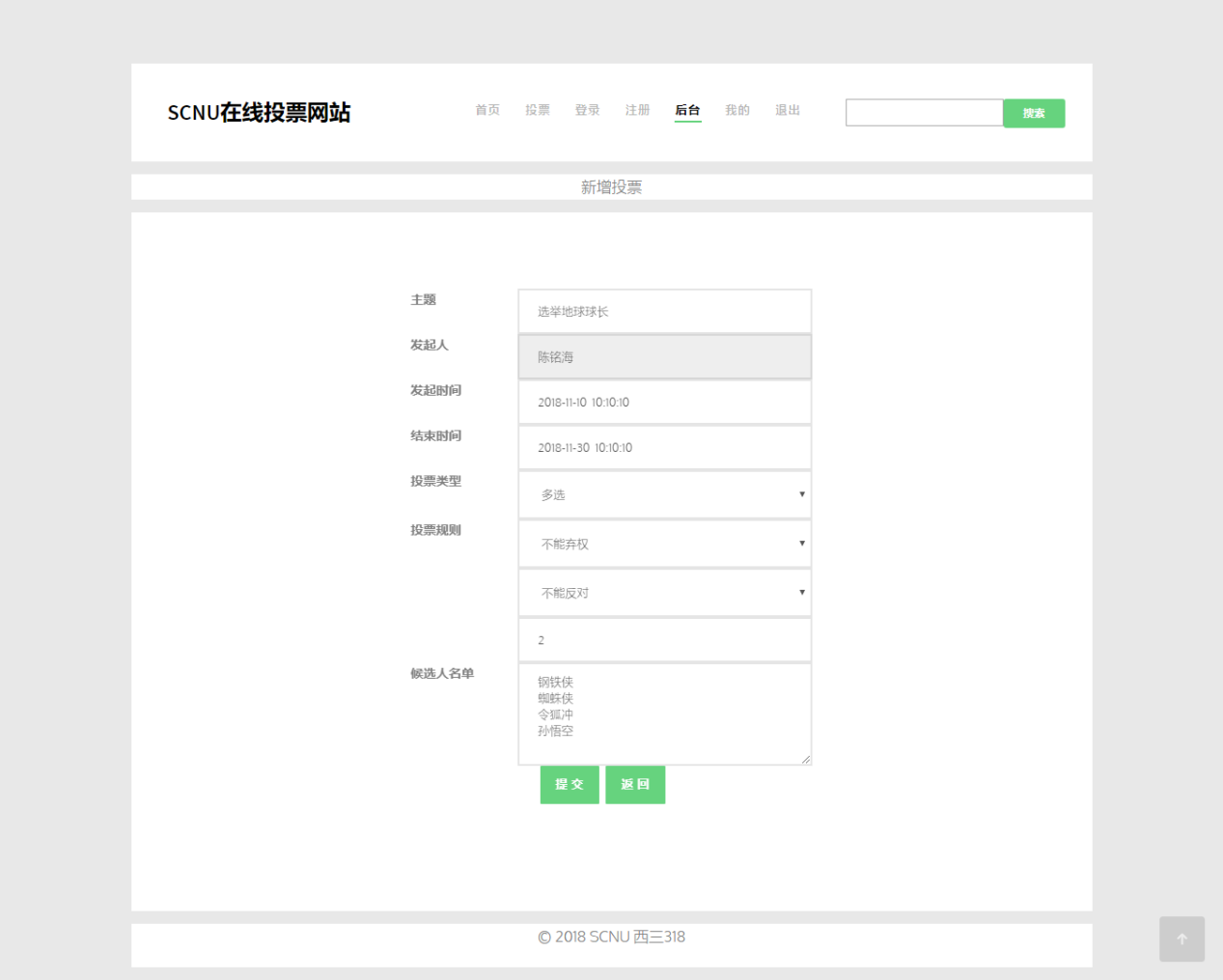
(8).管理员进入后台



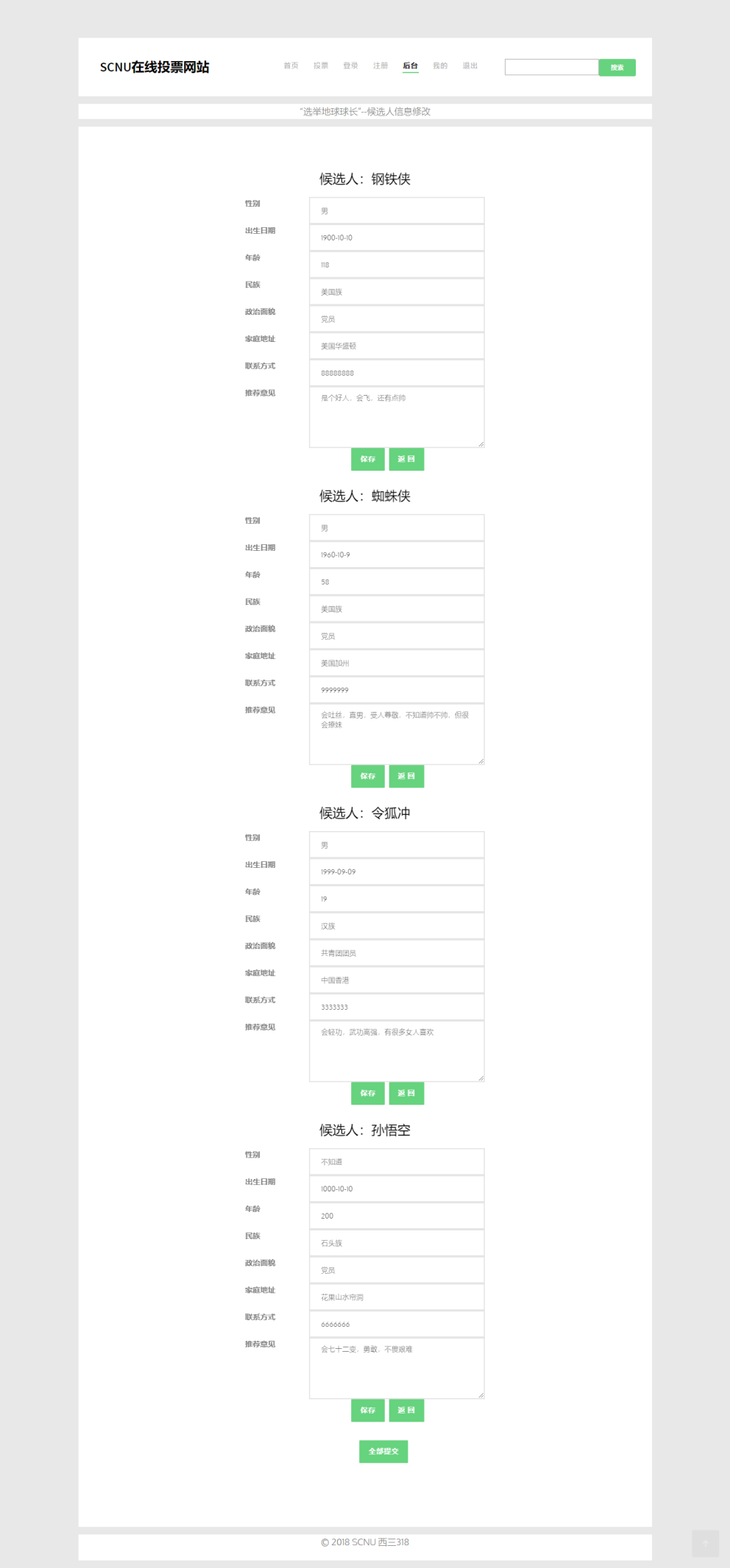
(9).管理员进入后台后，可以修改投票活动



(10).管理员进入后台后，可以添加投票活动



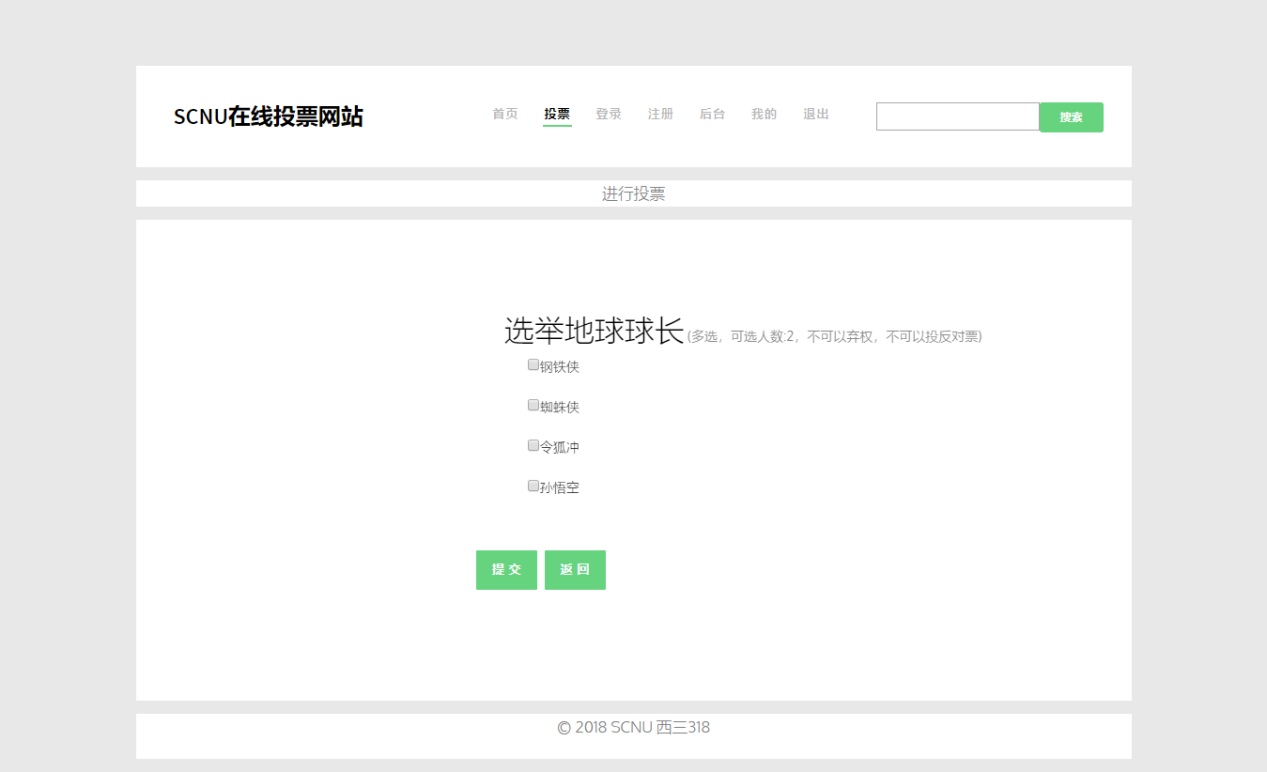
(11).添加投票活动后，添加该活动候选人信息



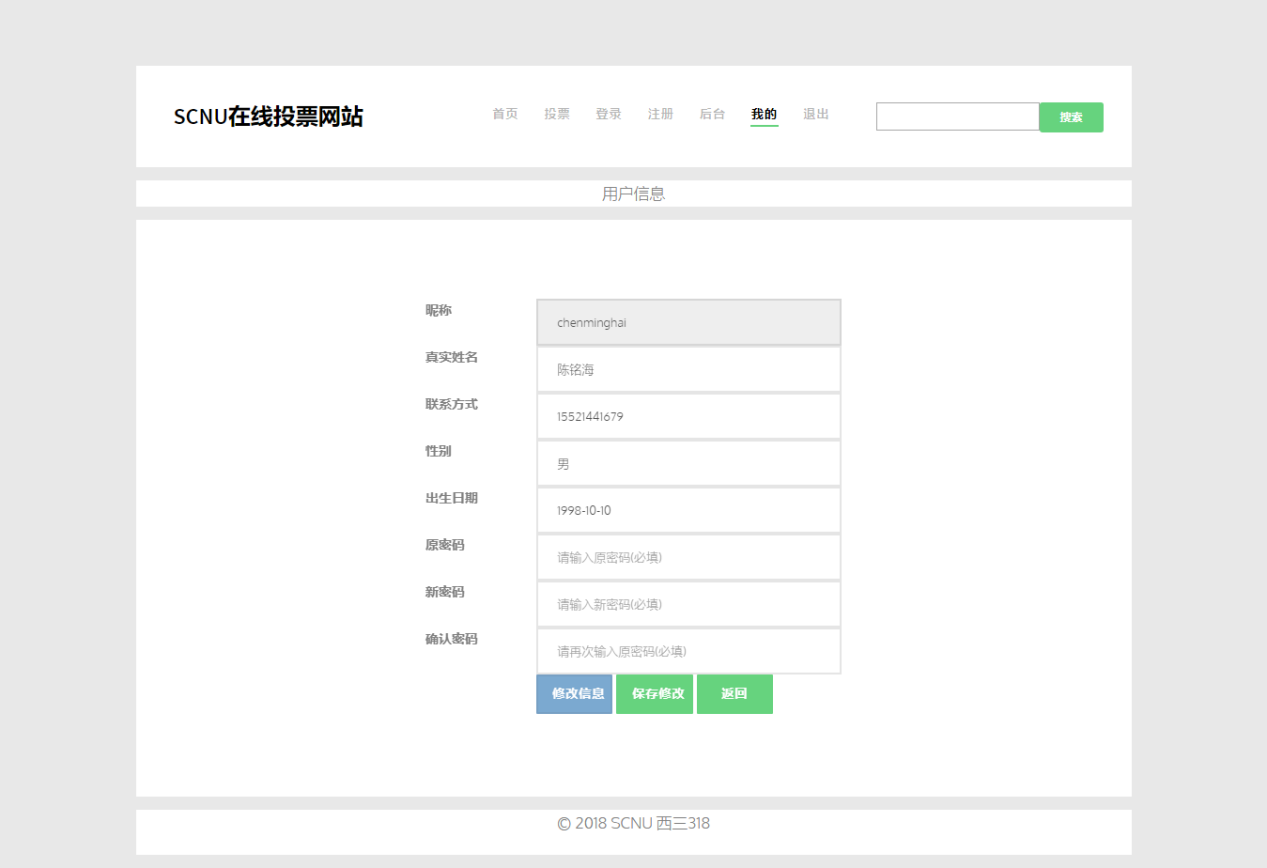
(12).添加活动候选人信息后，保存成功



(13).给新添投票活动投票



(14).查看和修改用户信息



2.系统测试

经过各个系统软件的集合，测试结果显示本软件暂时没有发现bug

**三．技术讨论**（存在问题、改进方向）

1. **存在问题**

1.没有实现评分制的投票类型

解决方案：由于在数据库设计阶段有预留该字段，因此可以一个输入评分框给用户输入来解决。

2.没有对用户登录次数进行限制

解决方案：可以在后台对用户每次登录，在session里添加一次，再用ajax和在后台进行测试。

3.当访问人数多时，可能会卡

解决方案：可以使用缓存来缓存所有投票活动，或者或者前几天的投票活动，来加快访问速度。

1. **改进方向**

1.可以加上安全机制

例如使用 Spring Security框架来加入安全机制。

2.可以继续开发手机APP或微信小程序这些手机端应用

3.可以使用Redis作为缓存加快访问速度