



南京邮电大学
Nanjing University of Posts and Telecommunications

在线专家评审系统 的设计与实现

答辩人：陈锦淞（B20031622）

专业：软件工程

学院：计算机学院、软件学院、网络空间安全学院

目录

CONTENT



PART 1 选题背景及意义



PART 2 研究思路与方法



PART 3 成果展示



PART 4 设计研究总结

01

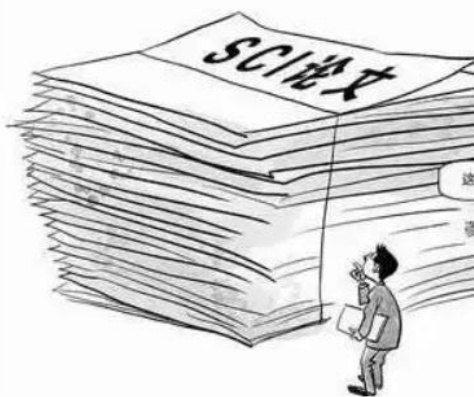
选题背景及意义

- 选题背景
- 研究意义



GRADUATION DEFENSE

选题背景



选题背景一

科研活动是高校发展过程中不可或缺的一部分，各大高校都在鼓励学生和导师去进行科学研究，很多高校都建立了科研相关的信息系统以此来更好的进行项目申报、资源审批、论文评审等一系列的科研工作。

选题背景二

教育科研的信息化就是借助电子设备开发的各种辅助科研活动的信息技术工具手段，打破时间空间的限制，使远程合作成为可能。科研信息化可以总结为数字化、网络化、虚拟化的，它是基于网络架构为平台进行科研工作。





GRADUATION DEFENSE

研究意义



意义

01

基于WEB的小型专家评审系统有利于完善学校一些科研活动的评审管理的科学化和信息化。

02

采用B/S架构进行设计，提高了高校在科研评审工作中的效率。

03

避免出现记录不当而导致的问题，进而确保了评审记录的准确性。

04

提升了科研竞赛活动的举办方和参与方的积极性。

02

研究思路及方法

- 设计语言和编程工具选择
- 使用框架
- 功能设计
- 数据库表设计
- 可行性说明



设计语言和编程工具选择

GRADUATION DEFENSE

小型专家评审系统采用Java语言进行开发设计，Java语言是一门面向对象的编程语言，Java具有简单易使用性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点。

常用的开发工具为ECLIPSE和IDEA两种，由于IDEA在调试、自动完成、反射方面对比ECLIPSE更加方便，所以本系统最终决定使用IDEA进行开发。



使用框架

GRADUATION DEFENSE

系统B/S开发模式，具有良好的跨平台部署能力和使用的便捷性，服务端基于SpringBoot作为系统主框架，利用其容器化、依赖注入、按需配置、广泛集成化等特性降低开发复杂度，提高了系统的可维护性、扩展性和稳定性。

前端部分使用Bootstrap框架设计，Bootstrap是免费的开源CSS框架，旨在快速开发WEB项目和应用程序。它包含了丰富的Web组件，根据这些组件，可以快速的搭建一个漂亮、功能完备的web网页。



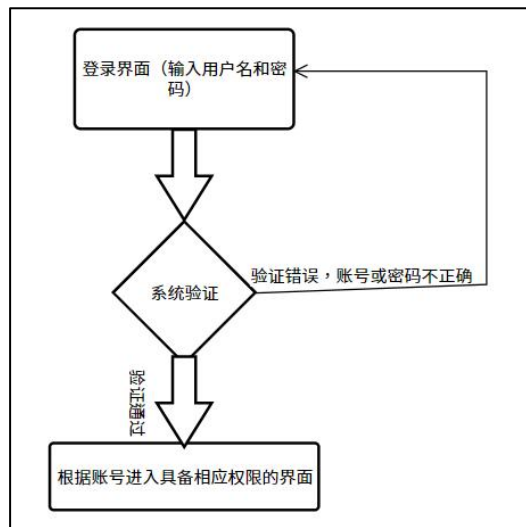
功能设计

GRADUATION DEFENSE

1、登录设计

根据不同账号进行登录，应当进入不同的界面和拥有不同的访问权限，所以登录界面部分的身份验证就是关键，用户输入自己的账号密码，系统验证账号密码是否匹配并且正确，如果正确就进入本账号应具备的权限界面，如果不正确则提示：“用户名或密码错误！”。

同时这里输入账号和密码时会有提示长度限制，本系统设置的是账号最少要输入5个字符，密码最少要输入6个字符，如果输入字符数不够也会给予提示。安全登录这里采用MD5加密方式，这里是通过编写一个MD5Util来实现。登录成功后系统会记录登陆的日志，方便后续的查阅管理。



登录流程图



GRADUATION DEFENSE

```
public static String encode(String password) throws NoSuchAlgorithmException
{
    password = password + SALT;
    char[] charArray = password.toCharArray();
    byte[] byteArray = new byte[charArray.length];
    for (int i = 0; i < charArray.length; i++)
    {
        byteArray[i] = (byte) charArray[i];
    }
    byte[] md5Bytes = getMD5().digest(byteArray);
    StringBuffer hexValue = new StringBuffer();
    for (int i = 0; i < md5Bytes.length; i++)
    {
        int val = ((int) md5Bytes[i]) & 0xff;
        if (val < 16)
        {
            hexValue.append("0");
        }

        hexValue.append(Integer.toHexString(val));
    }
    return hexValue.toString();
}

public static void main(String[] args) throws NoSuchAlgorithmException
{
    System.out.println(MD5Util.encode( password: "xf"));
}
```

MD5Util部分代码

```
@Override
public void logout(HttpServletRequest request, HttpServletResponse response, Authentication authentication) {
    System.out.println("退出成功!");
    String sessionId = request.getSession().getId();
    String u = this.defaultTargetUrl;
    if (StringUtils.isNotBlank(sessionId))
    {
        Log log = logService.getByLogId(sessionId);
        if (null != log)
        {
            long logoutTime = System.currentTimeMillis();
            log.setLogoutTime(logoutTime);
            long l = logoutTime - log.getLoginTime();

            long day = l / (24 * 60 * 60 * 1000);
            long hour = (l / (60 * 60 * 1000)) - day * 24;
            long min = ((l / (60 * 1000)) - day * 24 * 60 - hour * 60);
            long s = (l / 1000 - day * 24 * 60 * 60 - hour * 60 * 60 - min * 60);

            String onlineTime = "";
            if (day > 0)
            {
                onlineTime += day + "天 ";
            }
            if (hour > 0)
            {
                onlineTime += hour + "小时 ";
            }
            if (min > 0)
            {
                onlineTime += min + "分 ";
            }
        }
    }
}
```

日志记录部分代码



2、用户角色权限设计

根据需求分析以及用户角色权限的相关设计，系统设置管理员进行总的权限分配，系统通过设计不同级别和父子节点的管理来进行具体的分配，通过用户角色直接多对多的关系进行增删操作，实现了用户和角色权限对于绑定，从而可以进行权限的分配和取消。

menuId	menuName	fileUrl	menuPic	accessUrl	menuCode	parentMenuCode
1	系统管理	/system/	fa fa-cog	(NULL)	10	0
2	用户管理	/system/user	fa fa-user	/system/user	1001	10
3	组织管理	/system/org	fa fa-building	/system/org	1002	10
4	权限管理	/system/auth	fa fa-users	/system/auth	1003	10
5	系统参数	/system/paramType	fa fa-desktop	/system/paramType	1004	10
6	日志管理	/system/log	fa fa-desktop	/system/log	1005	10
7	项目管理	/work/	fa fa-film	/work/	01	0
8	项目提报	/work/proFile	fa fa-suitcase	/work/proFile	0101	01
9	专家评审	/work/expert	fa fa-user-md	/work/expert	0102	01

权限节点示意图



GRADUATION DEFENSE

```
public class MyAccessDecisionManager implements AccessDecisionManager
{
    @Override
    public void decide(Authentication authentication, Object object, Collection<ConfigAttribute> configAttributes)
        throws AccessDeniedException, InsufficientAuthenticationException
    {
        if (null == configAttributes || configAttributes.size() <= 0)
        {
            return;
        }

        Iterator<ConfigAttribute> ite = configAttributes.iterator();

        while (ite.hasNext())
        {
            ConfigAttribute ca = ite.next();
            String needRole = ca.getAttribute();
            // ga 为用户所被赋予的权限, needRole 为访问相应的资源应该具有的权限.
            for (GrantedAuthority ga : authentication.getAuthorities())
            {
                if (needRole.trim().equals(ga.getAuthority().trim()))
                {
                    return;
                }
            }
        }
        throw new AccessDeniedException("该用户无权限");
    }
}
```

权限获取设置

```
@Component
public class MyFilterSecurityInterceptor extends AbstractSecurityInterceptor implements Filter
{
    /**
     * 引用权限数据
     */
    @Autowired
    private FilterInvocationSecurityMetadataSource securityMetadataSource;

    /**
     * 设置决策管理器
     *
     * @param myAccessDecisionManager
     */
    @Autowired
    public void setMyAccessDecisionManager(MyAccessDecisionManager myAccessDecisionManager)
    {
        super.setAccessDecisionManager(myAccessDecisionManager);
    }

    @Override
    public void init(FilterConfig filterConfig) throws ServletException
    {
        // TODO Auto-generated method stub
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException
    {
    }
}
```

权限拦截器部分代码



3、爬虫修正专家研究方向

评审专家在成登录系统后，可以点击头像进行密码的修改，也可以进行研究方向的修正，这里可以通过网络爬虫来获取更多的方向，专家可根据自己的方向来修正。爬虫部分的组件存放在porm.xml文件中。

```
<!--WebMagic 爬虫-->
<dependency>
  <groupId>us.codecraft</groupId>
  <artifactId>webmagic-core</artifactId>
  <version>0.7.3</version>
</dependency>
<dependency>
  <groupId>us.codecraft</groupId>
  <artifactId>webmagic-extension</artifactId>
  <version>0.7.3</version>
</dependency>
<dependency>
  <groupId>us.codecraft</groupId>
  <artifactId>webmagic-selenium</artifactId>
  <version>0.6.1</version>
</dependency>

<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.3.1</version>
</dependency>
</dependencies>
```

爬虫组件



爬虫主要分为四部分：

1、创建Pagecessor，用来处理网页的数据。

```
public class CsdnBlogProcessor implements PageProcessor {  
    private Site site = Site  
        .me()  
        .setDomain("csdn.net")  
        .setSleepTime(3000)  
        .setUserAgent(  
            "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_2) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.65 Safari/537.31"  
        );  
    @Override  
    public boolean process(ResultItems resultItems, Task task) {  
        // 文章标题  
        String title = resultItems.get("title");  
        // 文章内容  
        String context = resultItems.get("context");  
        System.out.println("title:" + title);  
        System.out.println("context:" + context);  
    }  
}
```

2、创建pipeline, 存储处理好的网页数据。

```
public class CsdnPipeline implements Pipeline {  
    @Override  
    public void process(ResultItems resultItems, Task task) {  
        // 文章标题  
        String title = resultItems.get("title");  
        // 文章内容  
        String context = resultItems.get("context");  
        System.out.println("title:" + title);  
        System.out.println("context:" + context);  
    }  
}
```

3、利用Scheduler调取爬取任务，这里我设置的是51job的网址，由于求职网址方向较为全面。

```
@Override  
public Request poll(Task task) {  
    //Request request = new Request("http://my.oschina.net/flashword/blog/180623");  
    Request request = new Request( url: "https://nj.58.com/job.shtml?utm_source=market&spm=u-2d2yxv86y3v43nkddh1.B0PCPZ_BT&PGTID=0d100"  
    );  
    return request;  
}
```

4、进行爬虫测试。



GRADUATION DEFENSE

专家进入评审页面后，系统会根据他的研究方向来显示相应的已提交的对应方向的论文，专家这里只会显示项目名称、项目方向、论文名称、申报时间、评分分数和论文下载按钮。全程都是匿名状态无法查看申请者的信息，同时这里还加了一个定时器任务，在每天凌晨一点自动处理专家分数差距相差过大的减信誉分，这样可以防止恶意打分的情况出现，如果评分低于最终的平均分20在，则信誉分减少5分，低于平均分50分，则信誉分减少10分，如果信誉分低于60分的话，管理员这边会暂时使此账号停用，专家需自行联系管理员进行申请修改权限。

```
@Scheduled(cron = "0 0 1 1/1 * ?")
//@Scheduled(cron = "0 0/1 * * * ?")
private void configureTasks() {
    try {
        List<Expert> expertAVG = expertService.getExpertAVG(new Expert());
        for (Expert expert:expertAVG){
            float avg = expert.getAvg();
            List<Expert> expertList = expertService.getExpertByFileIdList(expert);
            for (Expert ex: expertList) {
                /**低于平均分20 减信誉分5, 低于平均分50 减信誉分10*/
                float scoreNum = ex.getScoreNum();
                User user = new User();
                user.setUserId(ex.getUserId());
                if((avg - scoreNum) >20){
                    user.setCreditScore(5);
                    userService.updateUserCreditScore(user);
                }else if((avg - scoreNum) > 50){
                    user.setCreditScore(10);
                    userService.updateUserCreditScore(user);
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

信誉分代码设计



数据库表设计

GRADUATION DEFENSE

数据库本系统使用的是MySQL关系型数据库，数据库版本为MySQL5.5。成本低的要求，并且其体积小，速度快，可以免费使用，易于在校生活和独立开发系统的人员的学习和使用。

系统总体表为：文件上传管理信息表（pro_file）、评分表（pro_score）、系统参数表（sys_param）、系统参数对应表（sys_paramtype）、在线记录表（tbl_log）、用户权限对应表（tbl_auth）、菜单管理表（tbl_menu）、用户角色创建模块表（tbl_u_gp）、用户表（tbl_user）、时间记录表（work_weekly）。



GRADUATION DEFENSE

可行性说明

可行性说明

可行性一

本课题设计完成后可用于小型的评审活动，系统经过测试，满足稳定性和高性能、安全性、易使用和易部署、便于升级扩展和维护这几方面的需求。

可行性二

本系统基于平台多用户、高并发访问的应用需求，系统在整体设计时应采用负载均衡技术，确保平台具备高可靠性，这里暂时无法测试太大的数据量，但从设计之初也考虑了在系统稳定的前提之下保证较快的响应速度，用最少的硬件需求来保证最大的系统稳定程度。设计过程中采用了目前较为常用的IDEA工具和个人设计系统使用较多的MySQL数据库，避免了因为工具的原因导致的运行出错，加上IDEA在运行部署方面较为方便，省去了不少人为操作，也使系统在编写调试方面更为快捷。

03

成果展示

- 登录界面
- 管理员界面
- 专家界面
- 申请者界面



登录界面

GRADUATION DEFENSE

在线专家评审系统

ONLINE EXPERT REVIEW SYSTEM

您已退出平台!

用户名

密码

登 录

打开浏览器输入网址即可进行登录界面

在线专家评审系统

ONLINE EXPERT REVIEW SYSTEM

1

最少要输入 5 个字符

密码

最少要输入 6 个字符

登 录

这里设置的账号最少为5个字符，密码最少为6个字符。如果输入错误则会提示用户名或密码错误！



管理员界面

GRADUATION DEFENSE



登录管理员账号后可进行系统管理



可查看所有用户信息



管理员界面

GRADUATION DEFENSE

+ 新增用户

用户名

必填

姓名

必填

密码

如果不填则为默认密码或原密码

确认密码

Email

必填

联系电话

必填

所属用户组

申请者组

账号状态

已激活

工号

性别

男

职称

所属院校

方向多个方向","分隔

备注

管理员可进行账号的新增和修改

+ 新增权限

用户组名称

必填

备注

选择权限

项目管理 (PC端)

项目提报 (PC端)

专家评审 (PC端)

系统管理 (PC端)

用户管理 (PC端)

组织管理 (PC端)

权限管理 (PC端)

系统参数 (PC端)

日志管理 (PC端)

关闭

提交

管理员可以通过勾选所需的权限进行管理用户组



专家界面

GRADUATION DEFENSE

修订方向

研究方向: 多个方向","分隔

it

获取

网络, 通讯, 电子, it, 软件, 测试, 爬虫, 研发, 软件工程, 游戏设计, 网页设计, 网页策划, 无线电工, 机械工程, JAVA, C++, C, .Net

关闭 提交

研究方向修订



申请者界面

GRADUATION DEFENSE

+ 提交申报

项目名称: *

文件: *.doc, *.docx

选择文件...

选择...

研究方向: 多个方向", "分隔

获取

软件设计、物联网、硬件运维

关闭

提交

论文提交

评分列表__平均分: 80

专家1号

分数:
80

评语:
DASDA

关闭

结果查看



04

设计研究总结



设计总结

GRADUATION DEFENSE

本文设计并实现了一个小型专家匿名评审系统，为管理员、评审专家、申请者提供全面的在线服务，既方便了参与人员快速完成在线评审的工作，同时又保证了评审过程的安全性和公平性。

整个项目是我独立设计开发，整个需求分析、后端代码、前端代码和数据库设计都是作者本人全程参与，测试也是在本机自主测试。编写过程中借鉴了一部分开源的评审系统源码，并进行学习修改，使得代码运行符合本系统需要。同时系统内部使用了较多现在比较流行的框架，使得后续可以在类似系统开发中继续使用，由于设备人员有限，所以只进行了少量数据的测试，在大数据测试方面还显得较为薄弱，希望后续有机会进一步改进程序，使得系统能上架使用。



南京邮电大学
Nanjing University of Posts and Telecommunications

感谢聆听 批评指正

答辩人：陈锦淞