****

软件开发实训课程设计

说明书

题 目： **图书推荐购买平台**

学 院： 计算机与信息安全学院

专 业： 计算机科学与技术

学生姓名： 陶俊荣

学 号： 1400310227

指导教师： 梁 海

2017年12月24日

目 录

[1 详细设计 1](#_Toc502047280)

[1.1 系统总体流程设计 1](#_Toc502047281)

[1.2 业务流程图 2](#_Toc502047282)

[2 系统实现 5](#_Toc502047283)

[2.1 登录模块 5](#_Toc502047284)

[2.1.1 界面 5](#_Toc502047285)

[2.1.2功能 5](#_Toc502047286)

[2.1.3代码及说明 5](#_Toc502047287)

[2.2 用户提交留言模块 6](#_Toc502047288)

[2.2.1界面 6](#_Toc502047289)

[2.2.2功能 7](#_Toc502047290)

[2.2.3代码及说明 7](#_Toc502047291)

[2.3 用户查看历史留言模块 7](#_Toc502047292)

[2.3.1界面 7](#_Toc502047293)

[2.3.2功能 8](#_Toc502047294)

[2.3.3代码及说明 8](#_Toc502047295)

[2.4 管理员审核留言模块 9](#_Toc502047296)

[2.4.1界面 9](#_Toc502047297)

[2.4.2功能 10](#_Toc502047298)

[2.4.3代码及说明 10](#_Toc502047299)

[3 功能测试 11](#_Toc502047300)

[3.1 测试用例 11](#_Toc502047301)

[3.2 功能测试 11](#_Toc502047302)

[3.3 测试结果分析 13](#_Toc502047303)

[4 总结 14](#_Toc502047304)

# 1 详细设计

## 1.1 系统总体流程设计

本图书推荐购买平台分为管理员功能模块和用户功能模块，管理员功能模块主要功能有：注销账户，修改密码，推荐列表管理,导出图书列表；用户功能模块主要功能有：注销账户，修改密码，提交推荐图书；系统总体流程图如图1.1所示：

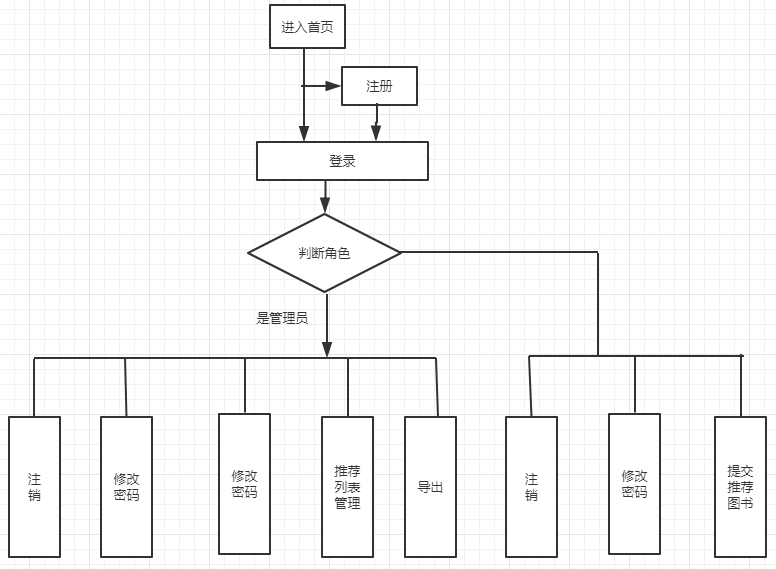


图1. 1 系统总流程图

## 1.2 业务流程图

（1）用户功能模块中:

1:判断用户是否登录，如果未登录，自动跳转到登录页面

2：已登录用户点击注销账户链接，进行账户的注销，后端自动使得页面跳转到系统的首页

3:修改密码，用户输入旧密码，两次新密码，在旧密码对了，以及两次密码输入符合输入的格式要求的前提之下，用户的密码发生改变

4.提交推荐购买的图书:输入并提交推荐图书的信息,判断用户输入的格式是否正确，否则不能进行提交

5.：入库图书列表查看和入库图书模糊查询

（4）管理员功能模块中:

1:判断用户是否登录，如果未登录，自动跳转到登录页面

2：已登录用户点击注销账户链接，进行账户的注销，后端自动使得页面跳转到系统的首页

3:修改密码，用户输入旧密码，两次新密码，在旧密码对了，以及两次密码输入符合输入的格式要求的前提之下，用户的密码发生改变

4.推荐图书列表管理（同意推荐，删除推荐信息）和入库图书列表管理（查询，添加）

# 2 系统实现

## 2.1 登录模块

### 2.1.1 界面

登录界面，本系统设计的登录界面简易并且易操作，，用户登录界面图如图2.1所示：



图2. 1 用户登录界面图

### 2.1.2功能

系统的登录模块，若输入密码错误，会出现相应的提示；原始预设值的账号密码单击后将清空输入框中的内容；为了网站的安全性，登录推荐购买平台要求输入验证码，当用户帐号密码跟验证码都正确之后，就可以进入系统首页，当然系统会根据用户的帐号信息，来判断出用户的角色

### 2.1.3代码及说明

在该功能模块中，主要是进行用户的登录验证的验证，代码设计简易，逻辑清晰，后台先判断是否存在此账号，接着判断密码是否正确。登陆接口主要代码如下：

Dao层接口

public User valdateUser(String userName, String userPass) {

String sql = "select \* from user where name=? and password=? ";

User u = (User) find(sql, new Object[] { userName, userPass },

new BeanHandler(User.class));

return u;

}

Controller层

String name=request.getParameter("name");

String password=request.getParameter("password");

request.setAttribute("name", name);

request.setAttribute("password", password);

if(StringUtil.isEmpty(name)||StringUtil.isEmpty(password)){

request.setAttribute("message", "不能为空");

request.getRequestDispatcher("login.jsp").forward(request, response);

return;

}

Connection con=null;

UserDao dao=new UserDao();

User u=dao.valdateUser(name, password);

if(u==null){

request.setAttribute("message", "用户名或者密码错误");

request.getRequestDispatcher("login.jsp").forward(request, response);

return;

}else if("1".equals(u.getRole())){

System.out.println("管理员进行登录");

HttpSession session=request.getSession();

session.setAttribute("currentUser",name);

session.setAttribute("userid",u.getUserid());

// �ͻ�����ת

response.sendRedirect("admin\_main.jsp");

}else{

System.out.println("普通用户进行登录");

HttpSession session=request.getSession();

session.setAttribute("currentUser",name);

// �ͻ�����ת

response.sendRedirect("user\_main.jsp");

}

## 2.3用户注册页面

### 2.2.1界面

用户提交留言界面，界面中的设计了一个文本框和两个按钮：“提交”、“匿名提交”，底部显示站内留言；需要用户输入留言内容后，点击其中一个按钮完成提交；用户提交留言界面图如图2.2所示：

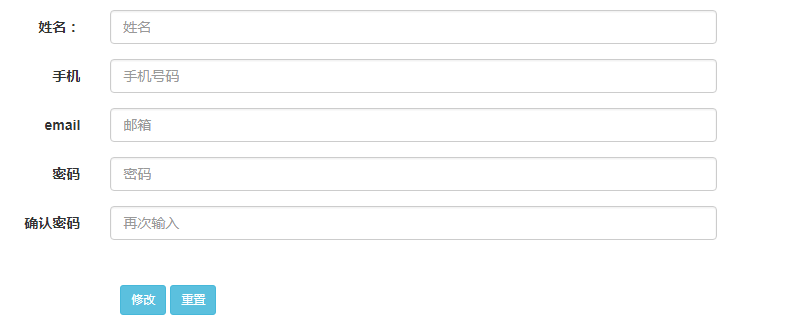


图2. 2 用户提交注册信息图

### 2.2.2功能

这里只是普通用户注册模块:用户点击注册按钮，进入注册界面，用户填写注册信息，系统判断注册信息是否符合格式要求，如果符合要求就允许用户进行注册

### 2.2.3代码及说明

首先得获取注册的表单参数，然后调用dao层接口将一条记录插入User表

dao接口

public void addUser(String name, String password, String email, String phone) {

String role = "0";// 普通用户

String userid = StringUtil.getId();

String sql = "insert into user (name,password,email,phone,role)"

+ " values (?,?,?,?,?)";

update(sql, new Object[] { name, password, email, phone, role });

}

## 2.3 管理员登录首页

### 2.3.1界面

用户查看历史留言界面，界面中显示了该用户个人的全部历史留言，并按时间排序，每条留言后有删除按钮，点击按钮后删除对应的留言；右上角有返回主页的链接；用户查看历史留言界面图如图2.3所示：

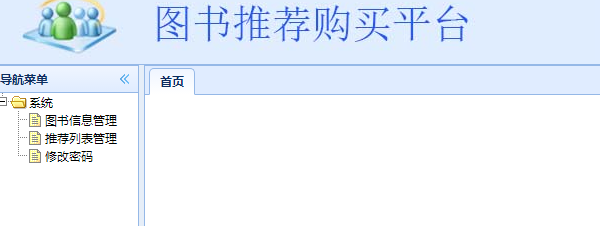


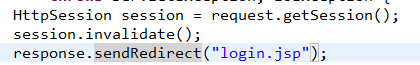
图2. 3 管理员登录首页

### 2.3.2功能

1.首页只是提供操作的功能选择：用户可以选择图书列表管理（这些图书信息已经入库）,推荐列表管理，管理密码，注销账户

### 2.3.3代码及说明

里面采用了easyui的树形菜单，通过获取json数据来产生菜单，json数据根据用户角色来生成，用户注销账户代码如下：



## 2.4 管理员推荐列表模块

### 2.4.1界面

管理员审核留言界面，该界面显示所有待审核的图书推荐列表，单击“审核通过”按钮则该留言通过审核；点击“删除”按钮则该留言不符合审核要求，对其进行删除操作。管理员审核留言界面图如图2.4所示：

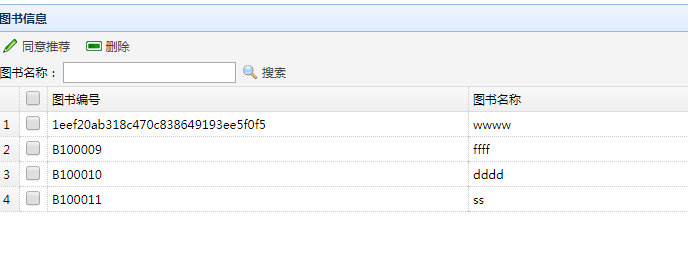


图2. 4 审核推荐图书界面

### 2.4.2功能

管理员先在复选框选择同意推荐的图书，然后点击同意推荐按钮,进行同意推荐的操作，只有管理员同意之后,这些图书信息就会在图书信息列表里面出现。同时管理员也可以选择认为不可以同意推荐的图书

### 2.4.3代码及说明

在该功能模块中，主要是进行推荐图书的审核，后台先查询出待审核的推荐图书的信息列表，然后管理员进行审核，审核通过，则修改推荐图书的审核状态；审核不通过，则删除该图书信息。审核图书信息的接口主要代码如下：

Dao层接口:

public int bookAgree(Connection con, String ids[]) throws Exception {

for (String id : ids) {

String sql = "update book set state='1' where book\_id = ?";

update(sql, new Object[] {id});

}

return ids.length;

}

Controller层:

String book\_ids=request.getParameter("delIds");

System.out.println("ids:"+book\_ids);

String ids[]=book\_ids.split(",");

Connection con=null;

AdviseDao dao=new AdviseDao();

try{

con=DbUtil.getConnection();

JSONObject result=new JSONObject();

int delNums=dao.bookAgree(con,ids);

## 2.5普通用户提交推荐图书

### 2.5.1界面

:用户提交推荐图书界面：用户填写要推荐的图书信息，前端会自动判断用户提交的信息是否符合输入的格式要求，然后点击保存按钮就实现提交



### 2.5.2功能

用户提交推荐图书，推荐的图书进行管理员系统中的管理推荐里面界面。

### 2.5.3代码及说明:

public void updateBook(String book\_id, String book\_name, String author,String publish,String content,String reason) {

String sql = "update book set book\_name=?,author=?,publish=?,content=?,reason=? where book\_id = ?";

update(sql, new Object[]{book\_name,author,publish,content,reason,book\_id});

}

public void addBook(Connection con,String book\_name, String author,String publish,String content,String reason) throws Exception {

//生成id

int count=this.bookCount(con, new Book());

count++;

System.out.println("count:"+count);

StringBuilder str=new StringBuilder("B1000");

str.append(count);

String book\_id=str.toString();

//生成state:

String state="0";

book\_id.concat(String.valueOf(count));

String sql = "insert into book (book\_id,book\_name,author,publish,content,reason,state)"+

" values (?,?,?,?,?,?,?)";

update(sql, new Object[]{book\_id,book\_name,author,publish,content,reason,state});

}

# 3 功能测试

## 3.1 测试用例

在经过了对图书推荐平台一系列开发设计之后，为了检验图书推荐平台的整体功能是否达到预期的设计目标，各个功能模块是否得到预期运行结果，先对图书推荐平台进行测试工作，尽可能地发现并排除软件中潜藏的错误，从而提高软件的质量。

针对用户功能模块，本次软件测试工作的测试用例设计如下：

（1） 用户登录模块。测试能否进入到用户登录界面，测试账号和密码为别为“admin 、123456”、“tao、123456”。两个分别属于不同的角色，进入之后进入的是不同的首页，功能菜单不一样

（2）提交推荐信息，不按照格式输入，看看能否可以提交。

（3）图书推荐列表审核模块。测试管理员是否能审核推荐图书，推荐的图书能否进入。

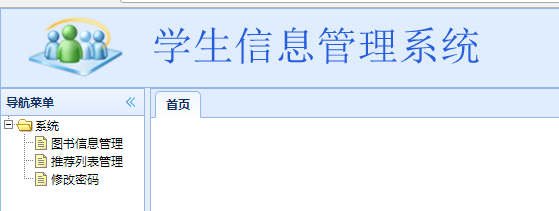
## 3.2 功能测试

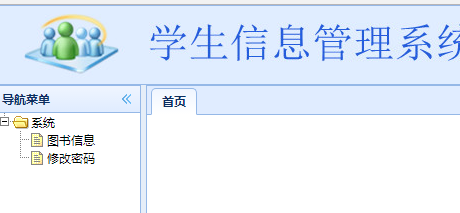
首先针对用户功能模块的主要功能进行测试。

（1）登录模块：该项测试中，在验证码输入正确的情况下，输入账号和密码为“admin 、123456”时，进入管理员操作界面，输入账号和密码为“tao、123456”时，进入普通会员操作界面：



图3. 1 登录模块测试结果图

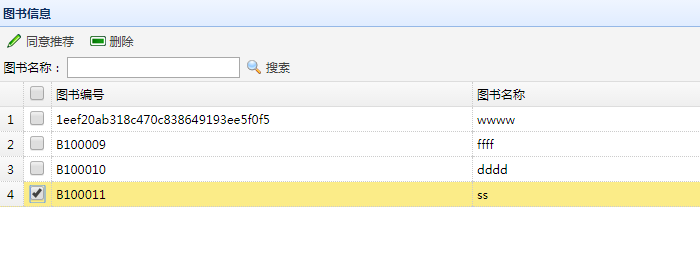




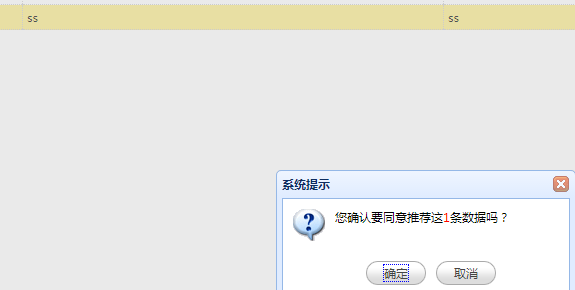
（2）提交推荐信息测试：防止用户进行非空提交



（3）审核推荐列表：在管理员操作界面，可以进行推荐列表的进行删除，同意推荐的操作，同意推荐的图书将不会在推荐列表出现，并且它会出现在图书列表里面，此时该图书的status属性已经发生改变



点击同意推荐



## 3.3 测试结果分析

（1）登录模块功能。经测试，该功能实现了用户登录系统的功能，登录验证准确无误，正确的账号密码能登录到相应的主界面，错误的账号密码登录失败，达到了预期效果。

（2）提交推荐图书功能。经测试，该功能能够成功实现推荐图书提交，这与系统对提交推荐图书信息的要求的情况一致，基本达到了预期效果。

（3）审核推荐图书功能。经测试，管理员能完全实现对用户提交的推荐列表进行审核或者删除，与预期结果一致。

# 4 总结

通过本次课程设计，本人对软件的认识不仅仅只是编码，软件是以文档和程序的形式存在，又涉及软件开发和实施过程的配置与组织，它包含从设计到编码，以及测试到维护等一系列过程，建筑着软件从五到有的过程。同时软件开发也存在各种难度：进度难以控制，估计软件开发的工作量困难，成本高，软件质量难以保证

本图书推荐购买平台，在看了课程设计要求之后进行设计编码，对课程的功能要求进行了需求分析，可行性分析（里面包含：进行技术的可行性分析，经济的可行性分析，社会的可行性分析），系统的运行环境是；

操作系统:Win7/Win8/Win10，数据库管理系统:MySQL5.7，服务器:tomcat.随后便是系统的设计：用户登录时自动进行对用户角色的分析，根据角色，进入不同的管理界面，进行不同的操作。

在系统完成之后，经过多方面的测试，证明系统能够满足用户的需求：服务器后端实现仅限JAVA EE和Node.js，数据库不限，web前端页面任选框架实现。根据选题设计与实现RESTful API接口。注册、登录、注销、修改密码、图书推荐信息提交、图书推荐列表管理和导出。对于已经接受的推荐意见，后台提供设置入库状态和索书号；前台页面提供查询。

在实现Web端各项功能的同时，我掌握了在后台数据库MySQL中建立数据库表和表单的存储过程；学会了使用WebStorm开发工具；以及掌握了在easyui中利用JS、.ejs和路由设置接口技术的基础知识。通过本次课程设计，我对软件开发的过程有了进一步的认识，提高了自己的动手实践能力，收获颇丰。我认识到在开始一定要提前做好规划，在开发之前一定要明确用户的需求，在做好需求的前提下，再加上详细的设计与计划，同时确保系统功能符合用户需求，这样才可以避免在系统开发过程中出现混乱，并且在对程序进行调试时也能减少附带的错误。

总之，在本次课程设计过程中，我查找了很多书籍和有关资料，并主动和同学讨论在程序中遇到的问题，这使我的编程能力有了不少的提高。由于完成课程设计时间比较紧，加上自身的技术能力有限，所以基于留言板Web端中还有一些不尽如人意的地方，比如系统功能还不是很齐全，系统稳定性比较弱，数据库还不十分安全。同时，由于自身水平有限，系统中还存在少量的不足，系统功能上还是有需要改进的地方，还可以开发更多的功能在界面显示上，也可以做的更简单化，让用户使用起来更加简洁方便，这些不足都将在今后中继续完善。