**图书馆图书管理系统测试文档**

# 一、测试目的

图书馆管理系统每天要接待的人相当之多，系统的漏洞可能造成图书借阅与归还的过程出现差错。因此我们要测试编写好的图书馆图书管理系统，以确认完成的系统是否符合预期的用户需求，并有相应的错误处理。

# 二、编程环境

操作系统：windows 7

开发工具：eclipse 4.3.0

数据库：MySQL 5.5.16

# 三、测试范围

1.测试读者及管理员注册功能能否成功实现

2.测试读者及管理员的登录功能能否成功实现

3.测试读者及管理员查询书籍功能能否成功实现

4.测试读者预约图书功能能否成功实现

5.测试管理员登记借阅信息功能能否成功实现

6.测试管理员删除借阅信息功能能否成功实现

7.测试图书注销功能能否成功实现

# 

# 四、单元测试

1. 借书

流程图：

结束

搜索图书

找到？

剩余量

>0？

借书成功

开始

借书不成功

Y

N

Y

N

关键代码：

**try**{

String url = "jdbc:mysql://localhost/library?useUnicode=true&characterEncoding=gbk";

String user = "root";

String pwd = "FannyMylove0327";

Class.*forName*("com.mysql.jdbc.Driver");

Connection conn = DriverManager.*getConnection*(url,user,pwd);

Statement statement = conn.createStatement();

ResultSet res = statement.executeQuery("select count from book\_list where bid = '"+tbid.getText().trim()+"'");

**if**(!res.next()){

JOptionPane.*showMessageDialog*(*container*, "未找到相关书籍！");

**return**;

}

**if**(res.getInt("count")<=0){

JOptionPane.*showMessageDialog*(*container*, "该书以全部借出！");

**return**;

}

**int** count = res.getInt("count");

res.close();

res = statement.executeQuery("select DATE\_FORMAT(now(), '%Y-%m-%d')");

res.next();

String dd = res.getString("date\_format(now(), '%Y-%m-%d')");

res.close();

statement.execute("insert into borrow values('"+tuid.getText().trim()+"','"+tbid.getText().trim()+"','"+dd+"',date\_add('"+dd+"',INTERVAL 30 day),'0')");

statement.executeUpdate("update book\_list set count = "+(count-1));

JOptionPane.*showMessageDialog*(*container*, "借书成功！");

conn.close();

}**catch**(Exception ex){

ex.printStackTrace();

}

1. 还书

流程图：

开始

检查书本

损坏？

查看归还期限

交纳赔偿金

交纳赔偿金

超期？

还书成功

结束

N

N

Y

Y

关键代码：

**try**{

String url = "jdbc:mysql://localhost/library?useUnicode=true&characterEncoding=gbk";

String user = "root";

String pwd = "FannyMylove0327";

Class.*forName*("com.mysql.jdbc.Driver");

Connection conn = DriverManager.*getConnection*(url,user,pwd);

Statement statement = conn.createStatement();

ResultSet res = statement.executeQuery("select count from book\_list where bid = '"+tbid.getText().trim()+"'");

res.next();

**int** c = res.getInt("count");

statement.executeUpdate("update book\_list set count = 1+"+c);

java.util.Date current = **new** java.util.Date();

SimpleDateFormat sdf=**new** SimpleDateFormat("yyyy-MM-dd");

String dd=sdf.format(current);

statement.execute("insert into giveback values('"+tuid.getText().trim()+"','"+tbid.getText().trim()+"','"+dd+"')");

res = statement.executeQuery("select due\_time from borrow where uid='"+tuid.getText().trim()+"' and bid = '"+tbid.getText().trim()+"' and borrow\_time < '"+dd+"'");

res.next();

String bb = res.getString("due\_time");

java.util.Date cc = sdf.parse(bb);

res.close();

res = statement.executeQuery("select money from user\_info where uid = '"+tuid.getText().trim()+"'");

res.next();

**int** money = res.getInt("money");

res.close();

res = statement.executeQuery("select price from book\_list where bid = '"+tbid.getText().trim()+"'");

res.next();

**int** price = res.getInt("price");

res.close();

java.util.Date ee = sdf.parse(dd);

**long** time1 = ee.getTime();

**long** time2 = cc.getTime();

**if**(time1>time2){

**long** diff = (time1-time2)/(1000\*60\*60\*24);

**int** c1 = (**int**) (diff\*0.5);

money -= c1;

JOptionPane.*showMessageDialog*(*container*, "扣除超期赔偿金"+c1+"元！");

statement.execute("insert into claim values('"+tuid.getText().trim()+"','"+tbid.getText().trim()+"',"+c1+",'超期','"+dd+"')");

}

**if**(diu.isSelected()){

money -= price;

JOptionPane.*showMessageDialog*(*container*, "扣除丢失赔偿金"+price+"元！");

statement.execute("insert into claim values('"+tuid.getText().trim()+"','"+tbid.getText().trim()+"',"+price+",'丢失','"+dd+"')");

}

**if**(sun.isSelected()){

**int** c2 = (**int**) (price\*0.3);

money -= c2;

JOptionPane.*showMessageDialog*(*container*, "扣除损坏赔偿金"+c2+"元！");

statement.execute("insert into claim values('"+tuid.getText().trim()+"','"+tbid.getText().trim()+"',"+c2+",'损坏','"+dd+"')");

}

statement.executeUpdate("update user\_info set money = "+money+" where uid = '"+tuid.getText().trim()+"'");

JOptionPane.*showMessageDialog*(*container*, "还书成功！");

conn.close();

}**catch**(Exception ex){

ex.printStackTrace();

}

1. 续借

流程图：

开始

选择要续借的书

查询借书信息

续借过？

续借成功

续借不成功

结束

Y

N

将归还期限延后30天

关键代码：

**boolean** flag = **false**;

**int** ok[] = **new** **int**[30];

**int** z = 0;

**try**{

String url = "jdbc:mysql://localhost/library?useUnicode=true&characterEncoding=gbk";

String user = "root";

String pwd = "FannyMylove0327";

Class.*forName*("com.mysql.jdbc.Driver");

Connection conn = DriverManager.*getConnection*(url,user,pwd);

Statement statement = conn.createStatement();

**for**(**int** i = 0;i<30&&jcb[i]!=**null**;i++){

**if**(jcb[i].isSelected()){

**if**(i == 0)

System.*out*.println(111);

flag = **true**;

String tt = jcb[i].getText();

**int** j = 0;

**while**(tt.charAt(j)!=' ')

j++;

String pp = tt.substring(0, j);

**while**(tt.charAt(++j)!='：');

String qq = "'"+tt.substring(++j)+"'";

ResultSet res = statement.executeQuery("select remark from borrow where bid = "+pp+" and uid = "+name+" and due\_time = "+qq);

res.next();

**if**(res.getString("remark").equals("1")){

ok[z++] = i;

**continue**;

}

res.close();

statement.executeUpdate("update borrow set due\_time = date\_add("+qq+",INTERVAL 30 day) where bid="+pp+" and uid="+name);

statement.executeUpdate("update borrow set remark = '1' where bid = "+pp);

}

}

**if**(z>0){

StringBuilder sb = **new** StringBuilder("抱歉！\n");

**while**(z>0)

sb.append(jcb[ok[--z]].getText()+"\n");

sb.append(" 您已续借过！不能再次续借！");

JOptionPane.*showMessageDialog*(container, sb);

}

**else** **if**(flag)

JOptionPane.*showMessageDialog*(container, "恭喜！续借成功！");

**else**

JOptionPane.*showMessageDialog*(container, "请先选中需要续借的书目！");

conn.close();

}**catch**(Exception ex){

ex.printStackTrace();

}

4.查询书籍（查询读者、查询借还书历史、查询赔偿历史情况类似）

流程图：

开始

输入相关信息

查询书籍列表

找到？

查询成功

查询不成功

结束

Y

N

将相关书籍信息显示在界面

关键代码：

**int** cnt = 1;

**try**{

String url = "jdbc:mysql://localhost/library?useUnicode=true&characterEncoding=gbk";

String user = "root";

String pwd = "FannyMylove0327";

Class.*forName*("com.mysql.jdbc.Driver");

Connection conn = DriverManager.*getConnection*(url,user,pwd);

String t1 = "ANY (select name from book\_list)";

String t2 = "ANY (select author from book\_list)";

String t3 = "ANY (select publisher from book\_list)";

String t4 = "ANY (select category from book\_list)";

**if**(!tname.getText().trim().equals(""))

t1 = "'"+tname.getText().trim()+"'";

**if**(!tauthor.getText().trim().equals(""))

t2 = "'"+tauthor.getText().trim()+"'";

**if**(!tpublisher.getText().trim().equals(""))

t3 = "'"+tpublisher.getText().trim()+"'";

**if**(!tcat.getSelectedItem().toString().trim().equals("请选择"))

t4 = "'"+tcat.getSelectedItem().toString().trim()+"'";

Statement statement = conn.createStatement();

ResultSet res = statement.executeQuery("select \* from book\_list where name = "+t1+" and author = "+t2+" and publisher = "+t3+" and category = "+t4);

**if**(!res.next())

JOptionPane.*showMessageDialog*(container, "抱歉！没有搜索到相关书目！");

**else**{

jta.append(" "+(cnt++)+". 书籍号："+res.getString("bid")+" 书名："+res.getString("name")+" 作者："+res.getString("author")+" 出版社："+res.getString("publisher")+" 类别："+res.getString("category")+"\n");

**while**(res.next()){

jta.append(" "+(cnt++)+". 书籍号："+res.getString("bid")+" 书名："+res.getString("name")+" 作者："+res.getString("author")+" 出版社："+res.getString("publisher")+" 类别："+res.getString("category")+"\n");

}

}

res.close();

conn.close();

}**catch**(Exception ex){

ex.printStackTrace();

}

# 五、界面测试

整体界面反映了用户对系统的第一印象，界面是否让用户感到舒适，能否直观的明白如何获取自己需要的数据，需要我们对界面的美观有一定要求。

# 六、功能测试

功能测试就是对产品的各功能进行验证，根据功能测试用例，逐项测试，检查产品是否达到用户要求的功能。

1. 测试范围：

侧重于可以被直接追踪到用例或业务功能和业务规则的所有测试需求。

1. 测试目的：

核实能否正确地接受、处理和检索数据以及业务规则是否正确实施。

确保测试对象的功能正常，其中包括检索、查询、借阅和删除信息等。

1. 测试方法：

利用有效的和无效的数据和操作来执行各个用例、用例流或功能，以核实以下内容：

在使用有效数据时得到预期的结果。

在使用无效数据时显示相应的错误消息或警告消息。

各业务规则都得到了正确的应用。

1. 系统所要完成的功能应当包括：
2. 对用户设置不同操作权限，使系统面向管理员和读者两类群体均正确运行。
3. 界面简洁大方，操作简单，易于用户使用。
4. 能增加、修改、删除读者账户，对读者进行统一管理。
5. 能增加、修改、删除书籍信息，对书籍进行统一管理。
6. 实现多条件查询读者信息、书籍信息的功能。
7. 实现借书、还书、预约图书的业务功能。
8. 能处理续借、超期赔偿、损坏赔偿、丢失赔偿等意外事故。
9. 方便读者查询自己的借还书历史与赔偿记录。
10. 功能模块

读者账号与信息的增删

图书馆管理信息系统

管理员模块

读者模块

书籍信息的增删改

管理借阅信息

读者、书籍信息的查询

赔偿支付

续借

还书

借书

预约图书

借还书历史的查询

1. 功能测试用例
2. 登录模块测试用例

a.注册检测

用户输入：

身份（选择）：管理员、读者，用户名user\_name, 密码123456

预期输出：注册成功。

b.登录检测

输入：用户名user\_name和密码123456。

预期输出：登录成功。

1. 应用模块测试用例（用户成功登录后）

a.查找书籍

用户点击“查找”，按书名大写字母索引查找或在搜索框中输入书名进行查找。页面显示出所有关于所查找书籍的信息。

b.借阅书籍

图书管理员添加被借阅书籍编号到系统。系统自动修改书库的图书记录、读者库信息。页面显示借阅成功。

c.归还书籍：

工作人员查看该书是否超期，若超期，则打印罚款单给读者；最后清除借阅记录，再将该书设置为可借状态。

d.预约书籍：

如果图书馆中有这本书但是书已外借，那么读者就可以通过系统预约图书。

e.注销书籍：

如果图书丢失或旧书淘汰，则将该书从书库中清除。

f.网站用户管理：

管理员可以查看用户名及密码，用户忘记密码后，通过管理员找回密码。

# 七、性能测试

在一定负载情况下，为了保证图书馆图书管理系统的性能（如系统响应时间、吞吐量等）满足读者和管理员的需求，需要对软件进行性能测试。

## （1）压力测试

发现图书馆图书管理系统的性能在什么情况下会退化或失败。

|  |  |
| --- | --- |
| 测试目标： | 确定所指定功能在以下情况下的性能行为：  大批量数据录入 |
| 方法： | 1.根据写好的测试计划对功能进行测试。  2.通过修改数据的循环次数来改变一定时间段里数据输入的数量。对不同的数据量进行测试。记录下数据量和对应的资源占用率，响应时间。  3.在同一台机器上重复测试，记录数据；在不同的机器上进行相同的操作，记录数据。 |
| 完成标准： | 多个用户：在可接受的时间范围内成功地完成测试脚本，没有发生任何故障。 |

## （2）安全性测试

对图书管理系统中的多方面可能出现的危险进行测试。

|  |  |
| --- | --- |
| 测试目标： | 在系统中，确保不同级别的人员只能进行怎样的操作和数据处理。 |
| 方法： | 1. 功能验证： 对注册功能、登录功能、查询书籍功能、预约图书功能等进行测试。 2. 模拟攻击： 用一组特殊的极端方法对功能进行测试。 |
| 完成标准： | 1. 主要验证上述功能是否有效。 2. 验证系统的安全防护能力。 |