软件测试课程设计

书店管理系统

2024年 12 月 16 日

**目 录**

1 书店管理系统介绍........................................3

2 软件需求分析(软件规格) .................................4

3 展开软件架构设计................. ......................5

4开发测试计划(静态评审、单元测试与集成测试).............. ........................................6

4.1静态评审...........................................6

4.2单元测试...........................................17

4.3集成测试...........................................25

5 系统功能测试............ ..............................29

6 性能测试............................... ...............43 7工作对应及贡献表 ............... ......................49

1.软件项目介绍：

本书店管理系统包括用户注册与登录、书籍信息管理、借阅管理、用户留言、用户权限管理等功能。是一个基于JSP+Servlets实现的书店管理系统。

系统分为管理员、读者两个登录角色，一共是8个功能模块，包括书籍管理、用户管理、借阅管理、图书详情、书架管理、找回密码、用户注册、用户登录这几个模块

书籍管理：包括添加书籍、删除书籍、修改书籍信息、查询书籍等功能。

用户管理：可以对读者或管理员信息的修改和删除，增加用户，修改用户管理权限。完成用户登录，注册和退出功能。

借阅管理：记录读者的借书行为，包括借书时间、图书信息等，以及处理读者的还书行为，更新图书库存状态。

图书详情：读者可以查看图书详情，在书底下留下评论

书架管理：读者可以添加图书到自己的书架中

找回密码：允许读者或管理员通过账号找回密码。

用户注册：允许用户在登录界面注册账号

用户登录：运行用户在登录界面登录自己的账号

2.软件需求分析：

作为书店的经营者，我们委托软件开发公司开发一套书店管理系统，需求如下：

系统必须支持管理员对图书的增删改查操作。

系统必须支持管理员对用户的增删改查操作。

需要有读者注册、登录和退出功能，支持权限管理。

系统应记录每本书的借阅情况。

应支持用户查找登录密码。

应支持读者在书籍底下留言

应支持读者收藏书籍到自己的书架当中

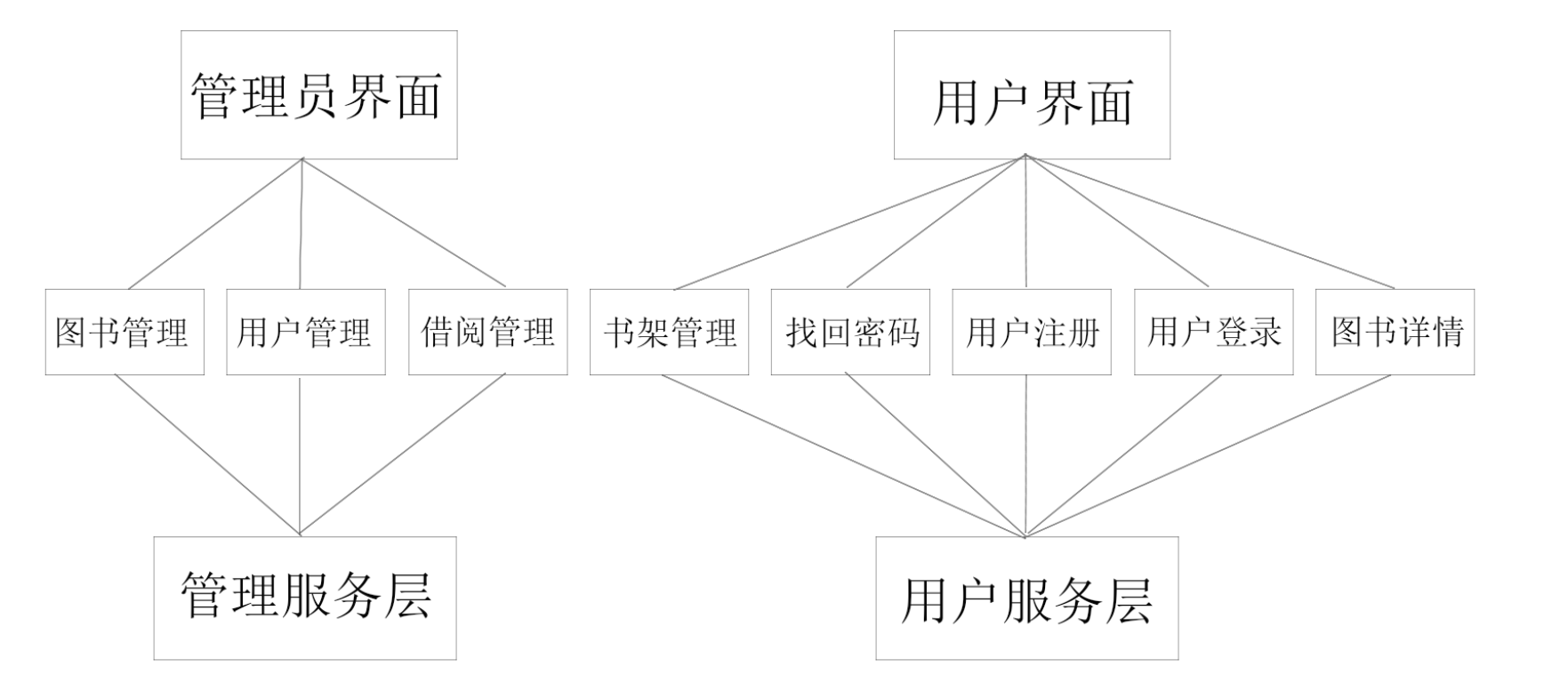
应支持读者通过书名查找书籍

安装条件：win7以上可用、8GB内存、硬盘空间1GB、JDK8以上、TomCat8以上

效能：支持200人以上同时操作

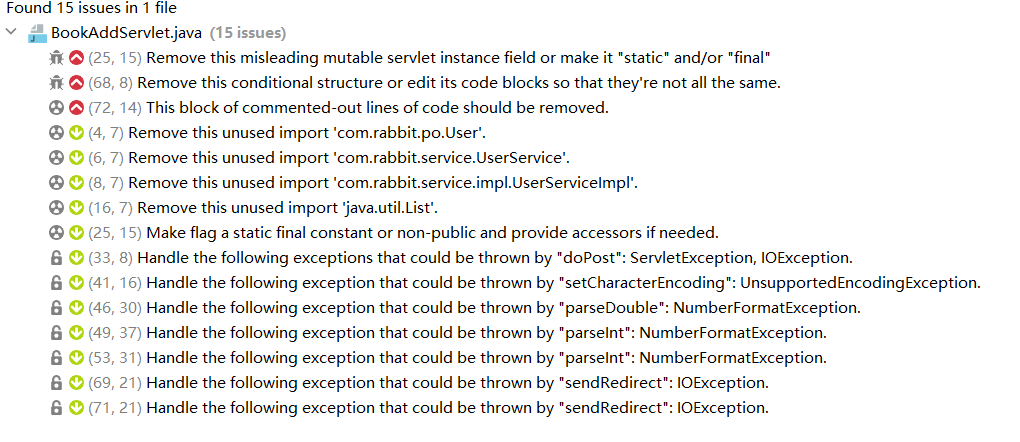
UI：简洁易懂

3.展开软件架构设计



4.开发测试计划（静态评审、单元测试和集成测试）

（1）静态评审：

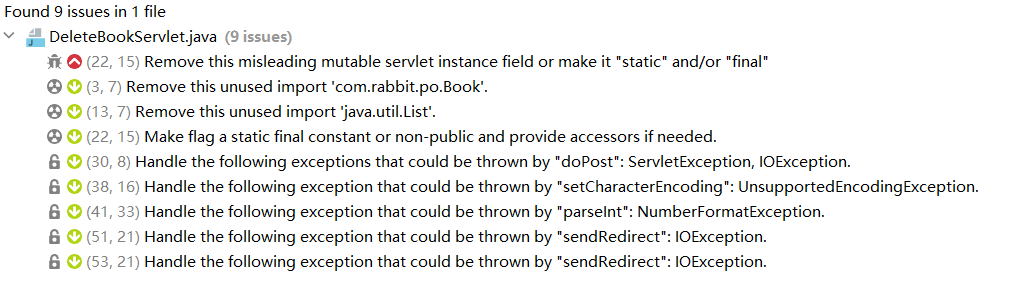


1. **(25, 15) Remove this misleading mutable servlet instance field or make it "static" and/or "final"**
   * **问题**: 第25行有一个可变且可能引起误解的 Servlet 实例字段。建议移除它，或者将其声明为 static 和/或 final。
2. **(68,8) Remove this conditional structure or edit its code blocks so that they're not all the same.**
   * **问题**: 第68行的条件结构中的代码块都是相同的。建议移除这个条件结构，或者修改代码块以确保它们不相同。
3. **(72,14) This block of commented-out lines of code should be removed.**
   * **问题**: 第72行有一段被注释掉的代码块。建议移除它。
4. **(25, 15) Make flag a static final constant or non-public and provide accessors if needed.**
   * **问题**: 第25行的 flag 应该被声明为 static final 常量，或者如果不是公共的，提供访问器方法。
5. **(33,8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第33行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

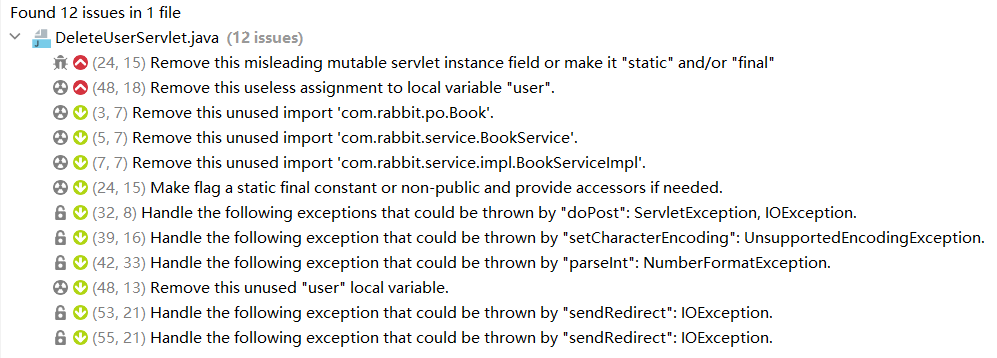
图片包含 文本

描述已自动生成

1. **(3, 7) Remove this unused import 'com.rabbit.dao.UserDao'.**
   * **问题**: 第3行有一个未使用的导入声明。建议移除它。
2. **(25, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第25行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。
3. **(32, 16) Handle the following exception that could be thrown by "setCharacterEncoding": UnsupportedEncodingException.**
   * **问题**: 第32行的 setCharacterEncoding 方法可能会抛出 UnsupportedEncodingException，需要处理这个异常。
4. **(37, 60) Handle the following exceptions that could be thrown by "forward": ServletException, IOException.**
   * **问题**: 第37行的 forward 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。



1. **(22, 15) Remove this misleading mutable servlet instance field or make it "static" and/or "final"**
   * **问题**: 第22行有一个可变且可能引起误解的 Servlet 实例字段。建议移除它，或者将其声明为 static 和/或 final。
2. **(22, 15) Make flag a static final constant or non-public and provide accessors if needed.**
   * **问题**: 第22行的 flag 应该被声明为 static final 常量，或者如果不是公共的，提供访问器方法。
3. **(30, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第30行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

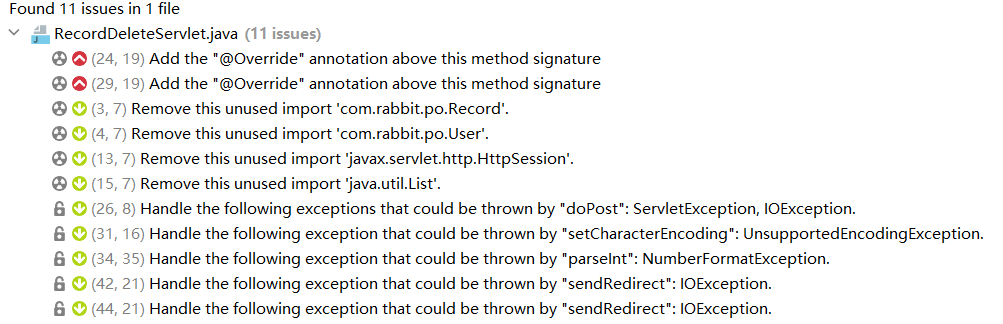


1. **(24, 15) Remove this misleading mutable servlet instance field or make it "static" and/or "final"**
   * **问题**: 第24行有一个可变且可能引起误解的 Servlet 实例字段。建议移除它，或者将其声明为 static 和/或 final。
2. **(48, 18) Remove this useless assignment to local variable "user".**
   * **问题**: 第48行对局部变量 "user" 的赋值没有实际用途。建议移除这个赋值。
3. **(24, 15) Make flag a static final constant or non-public and provide accessors if needed.**
   * **问题**: 第24行的 flag 应该被声明为 static final 常量，或者如果不是公共的，提供访问器方法。
4. **(32, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第32行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。
5. **(48, 13) Remove this unused "user" local variable.**
   * **问题**: 第48行声明了一个未使用的局部变量 "user"。建议移除它。
6. **(53, 21) Handle the following exception that could be thrown by "sendRedirect": IOException.**
   * **问题**: 第53行的 sendRedirect 方法可能会抛出 IOException，需要处理这个异常。

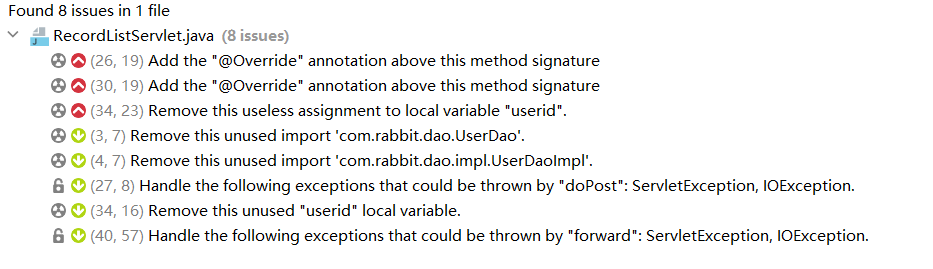
文本

低可信度描述已自动生成

1. **(21, 15) Remove this misleading mutable servlet instance field or make it "static" and/or "final"**
   * **问题**: 第21行有一个可变且可能引起误解的 Servlet 实例字段。建议移除它，或者将其声明为 static 和/或 final。
2. **(39, 10) This block of commented-out lines of code should be removed.**
   * **问题**: 第39行有一段被注释掉的代码块。建议移除它。
3. **(65, 14) This block of commented-out lines of code should be removed.**
   * **问题**: 第65行有一段被注释掉的代码块。建议移除它。
4. **(21, 15) Make flag a static final constant or non-public and provide accessors if needed.**
   * **问题**: 第21行的 flag 应该被声明为 static final 常量，或者如果不是公共的，提供访问器方法。
5. **(60, 21) Handle the following exception that could be thrown by "sendRedirect": IOException.**
   * **问题**: 第60行的 sendRedirect 方法可能会抛出 IOException，需要处理这个异常。



1. **(24, 19) Add the "@Override" annotation above this method signature**
   * **问题**: 第24行的方法签名缺少 @Override 注解注解，这有助于明确表示该方法是重写父类或接口中的方法。
   * **改进**: 在方法签名上方添加 @Override 注解注解。
2. **(29, 19) Add the "@Override" annotation above this method signature**
   * **问题**: 第29行的方法签名同样缺少 @Override 注解注解。
   * **改进**: 在方法签名上方添加 @Override 注解注解。
3. **(3, 7) Remove this unused import 'com.rabbit.po.Record'.**
   * **问题**: 第3行有一个未使用的导入声明。
   * **改进**: 移除这个未使用的导入。
4. **(26, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第26行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。



1. **(26, 19) Add the "@Override" annotation above this method signature**
   * **问题**: 第26行的方法签名缺少 @Override 注解，这有助于明确表示该方法是重写父类或接口中的方法。
   * **改进**: 在方法签名上方添加 @Override 注解。
2. **(30, 19) Add the "@Override" annotation above this method signature**
   * **问题**: 第30行的方法签名同样缺少 @Override 注解。
   * **改进**: 在方法签名上方添加 @Override 注解。
3. **(34, 23) Remove this useless assignment to local variable "userid".**
   * **问题**: 第34行对局部变量 "userid" 的赋值没有实际用途。
   * **改进**: 移除这个无用的赋值。
4. **(3, 7) Remove this unused import 'com.rabbit.dao.UserDao'.**
   * **问题**: 第3行有一个未使用的导入声明。
   * **改进**: 移除这个未使用的导入。
5. **(27, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第27行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。
6. **(34, 16) Remove this unused "userid" local variable.**
   * **问题**: 第34行声明了一个未使用的局部变量 "userid"。
   * **改进**: 移除这个未使用的局部变量。
7. **(40, 57) Handle the following exceptions that could be thrown by "forward": ServletException, IOException.**
   * **问题**: 第40行的 forward 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

文本

描述已自动生成

1. **(25, 19) Add the "@Override" annotation above this method signature**
   * **问题**: 第25行的方法签名缺少 @Override 注解，这有助于明确表示该方法是重写父类或接口中的方法。
   * **改进**: 在方法签名上方添加 @Override 注解。
2. **(29, 19) Add the "@Override" annotation above this method signature**
   * **问题**: 第29行的方法签名同样缺少 @Override 注解。
   * **改进**: 在方法签名上方添加 @Override 注解。
3. **(5, 7) Remove this unused import 'com.rabbit.po.Book'.**
   * **问题**: 第5行有一个未使用的导入声明。
   * **改进**: 移除这个未使用的导入。
4. **(26, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第26行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

文本

描述已自动生成

1. **(56, 11) Remove this expression which always evaluates to "true" [+3 locations]**
   * **问题**: 第56行有一个表达式，它总是评估为 true。这可能是一个逻辑错误或者冗余代码。
   * **改进**: 移除这个总是为 true 的表达式，检查代码逻辑是否正确。
2. **(4, 7) Remove this unused import 'com.rabbit.service.UserService'.**
   * **问题**: 第4行有一个未使用的导入声明。
   * **改进**: 移除这个未使用的导入。
3. **(24, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第24行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

图片包含 表格

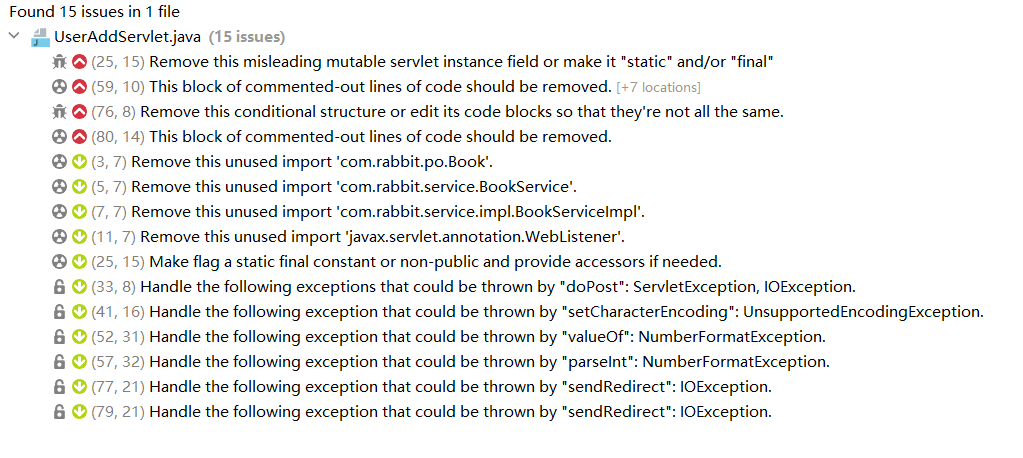
描述已自动生成

1. **(22, 15) Remove this misleading mutable servlet instance field or make it "static" and/or "final"**
   * **问题**: 第222行有一个可变且可能引起误解的 Servlet 实例字段。建议移除它，或者将其声明为 static 和/或 final。
2. **(52, 18) Remove this useless assignment to local variable "book".**
   * **问题**: 第52行对局部变量 "book" 的赋值没有实际用途。
   * **改进**: 移除这个无用的赋值。
3. **(13, 7) Remove this unused import 'java.util.List'.**
   * **问题**: 第13行有一个未使用的导入声明。建议移除它。
4. **(22, 15) Make flag a static final constant or non-public and provide accessors if needed.**
   * **问题**: 第22行的 flag 应该被声明为 static final 常量，或者如果不是公共的，提供访问器方法。
5. **(30, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第30行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

图片包含 文本

描述已自动生成

1. **(24, 15) Remove this misleading mutable servlet instance field or make it "static" and/or "final"**
   * **问题**: 第24行有一个可变且可能引起误解的 Servlet 实例字段。建议移除它，或者将其声明为 static 和/或 final。
2. **(3, 7) Remove this unused import 'com.rabbit.po.Book'.**
   * **问题**: 第3行有一个未使用的导入声明。建议移除它。
3. **(24, 15) Make flag a static final constant or non-public and provide accessors if needed.**
   * **问题**: 第24行的 flag 应该被声明为 static final 常量，或者如果不是公共的，提供访问器方法。
4. **(32, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第32行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

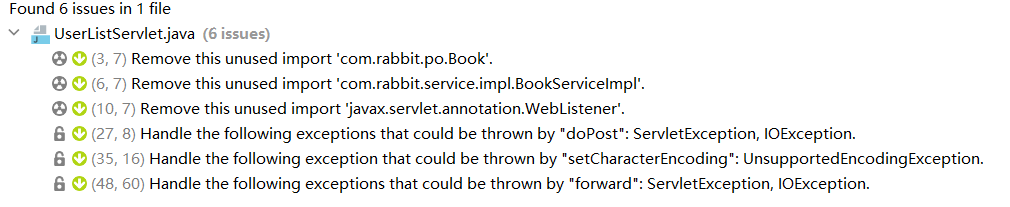


1. **(25, 15) Remove this misleading mutable servlet instance field or make it "static" and/or "final"**
   * **问题**: 第25行有一个可变且可能引起误解的 Servlet 实例字段。建议移除它，或者将其声明为 static 和/或 final。
2. **(59, 10) This block of commented-out lines of code should be removed. [+7 locations]**
   * **问题**: 第59行开始有一段被注释掉的代码块，建议移除。这个问题在7个不同的位置被标记。
3. **(76, 8) Remove this conditional structure or edit its code blocks so that they're not all the same.**
   * **问题**: 第76行的条件结构中的代码块都是相同的。建议移除这个条件结构，或者修改代码块以确保它们不相同。
4. **(80, 14) This block of commented-out lines of code should be removed.**
   * **问题**: 第80行有一段被注释掉的代码块。建议移除它。
5. **(25, 15) Make flag a static final constant or non-public and provide accessors if needed.**
   * **问题**: 第25行的 flag 应该被声明为 static final 常量，或者如果不是公共的，提供访问器方法。
6. **(33, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第33行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

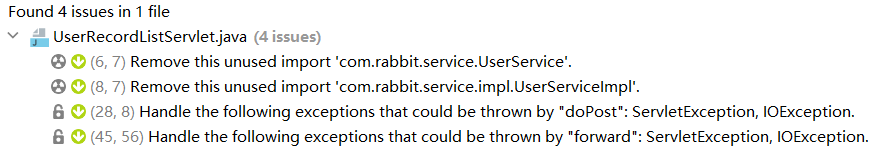
文本

描述已自动生成

1. **(17, 15) Remove this misleading mutable servlet instance field or make it "static" and/or "final"**
   * **问题**: 第17行有一个可变且可能引起误解的 Servlet 实例字段。建议移除它，或者将其声明为 static 和/或 final。
2. **(17, 15) Make flag a static final constant or non-public and provide accessors if needed.**
   * **问题**: 第17行的 flag 应该被声明为 static final 常量，或者如果不是公共的，提供访问器方法。
3. **(25, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第25行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。
4. **(32, 16) Handle the following exception that could be thrown by "setCharacterEncoding": UnsupportedEncodingException.**
   * **问题**: 第32行的 setCharacterEncoding 方法可能会抛出 UnsupportedEncodingException，需要处理这个异常。
5. **(39, 26) Use isEmpty() to check whether the collection is empty or not.**
   * **问题**: 第39行应该使用 isEmpty() 方法来检查集合是否为空，而不是通过其他方式。
6. **(43, 54) Handle the following exceptions that could be thrown by "forward": ServletException, IOException.**
   * **问题**: 第43行的 forward 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。



1. **(3, 7) Remove this unused import 'com.rabbit.po.Book'.**
   * **问题**: 第3行有一个未使用的导入声明。
   * **改进**: 移除这个未使用的导入。
2. **(27, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第27行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。



1. **(6, 7) Remove this unused import 'com.rabbit.service.UserService'.**
   * **问题**: 第6行有一个未使用的导入声明。
   * **改进**: 移除这个未使用的导入。
2. **(28, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第28行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

文本

描述已自动生成

1. **(4, 7) Remove this unused import 'com.rabbit.po.Record'.**
   * **问题**: 第4行有一个未使用的导入声明。
   * **改进**: 移除这个未使用的导入。
2. **(29, 8) Handle the following exceptions that could be thrown by "doPost": ServletException, IOException.**
   * **问题**: 第29行的 doPost 方法可能会抛出 ServletException 和 IOException，需要处理这些异常。

（2）单元测试

用户登录模块：

LoginServletTest 单元测试分析

一、测试概述

本次单元测试针对LoginServlet的doPost方法进行了全面的功能性测试，涵盖了管理员登录，用户登录，登陆失败等多种情况，旨在确保doPost方法的正确性和安全性。

二、测试用例执行情况

（一）testDoPost\_Admin

测试目的：验证管理员用户登录是否成功。

测试代码：

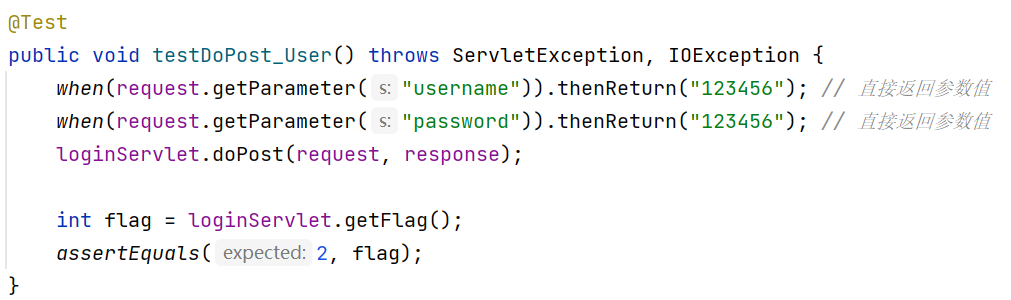


测试分析：模拟HTTP请求参数，用户名和密码都设置为"admin"。通过assertEquals(1, flag);断言检查LoginServlet的getFlag()方法返回值是否为1，表示管理员登录成功。

（二）testDoPost\_User

测试目的：验证普通用户登录是否成功。

测试代码：

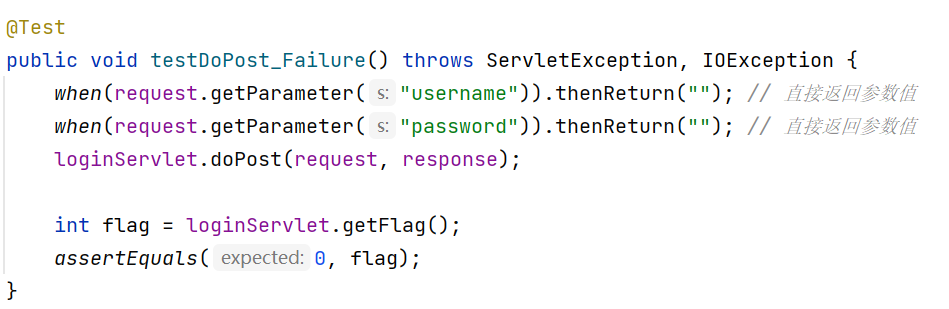


测试分析：模拟HTTP请求参数，用户名和密码都设置为"123456"。通过assertEquals(2, flag);断言检查LoginServlet的getFlag()方法返回值是否为2，表示普通用户登录成功。

（三）testDoPost\_Failure

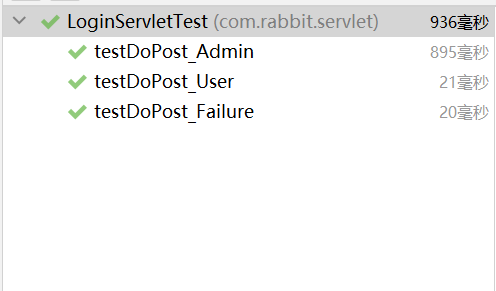
测试目的：验证无效登录是否被正确处理。

测试代码：



测试分析：模拟HTTP请求参数，用户名和密码都设置为空字符串。通过assertEquals(0, flag);断言检查LoginServlet的getFlag()方法返回值是否为0，表示登录失败。

测试结果：



图书管理功能模块：

BookServletTest 单元测试分析

一、测试概述

BookServletTest 类旨在测试与书籍相关的三个Servlet：BookAddServlet、UpdateBookServlet 和 DeleteBookServlet。这些Servlet分别处理书籍的添加、更新和删除操作。测试类使用JUnit框架和Mockito库来模拟HTTP请求和响应，以验证Servlet的行为。

二、测试用例执行情况

（一）**addTest**

测试目的：验证BookAddServlet是否能够成功添加书籍。

测试代码：

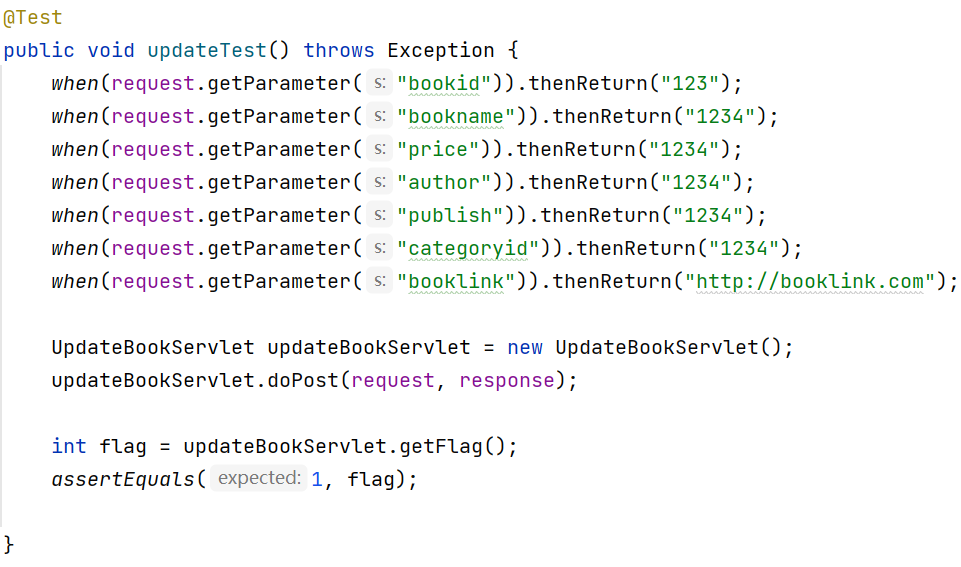


测试分析：模拟HTTP请求参数，包括书籍的ID、名称、价格、作者、出版社、分类ID和链接。通过assertEquals(1, flag);断言检查BookAddServlet的getFlag()方法返回值是否为1，表示添加成功。

（二）**updateTest**

测试目的：验证UpdateBookServlet是否能够成功更新书籍信息。

测试代码：



测试分析：模拟HTTP请求参数，包括更新后的书籍信息。通过assertEquals(1, flag);断言检查UpdateBookServlet的getFlag()方法返回值是否为1，表示更新成功。

（三）**deleteTest**

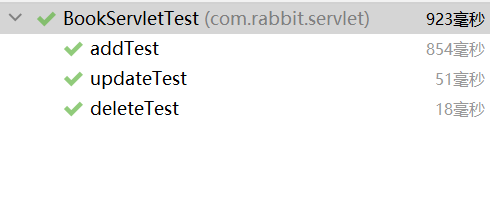
测试目的：验证DeleteBookServlet是否能够成功删除书籍。

测试代码：



测试分析：模拟HTTP请求参数，包括书籍的ID和其他信息。通过assertEquals(1, flag);断言检查DeleteBookServlet的getFlag()方法返回值是否为1，表示删除成功。

测试结果：



用户管理模块：

UserServletTest 单元测试分析

一、测试概述

UserServletTest 类是一个JUnit测试类，旨在测试用户相关的Servlet，包括UserAddServlet、UpdateUserServlet和DeleteUserServlet。这些Servlet分别处理用户的添加、更新和删除操作。测试类使用Mockito框架来模拟HttpServletRequest和HttpServletResponse对象，以验证Servlet的行为。

二、测试用例执行情况

（一）**addTest**

测试目的：验证用户添加操作是否成功。

测试代码：



测试分析：模拟HTTP请求参数，包括用户ID、用户名、密码、电子邮件、电话和是否为管理员。通过assertEquals(1, flag);断言检查UserAddServlet的getFlag()方法返回值是否为1，表示添加成功。

（二）**addWrongTest**

测试目的：验证添加已有用户是否失败

测试代码：

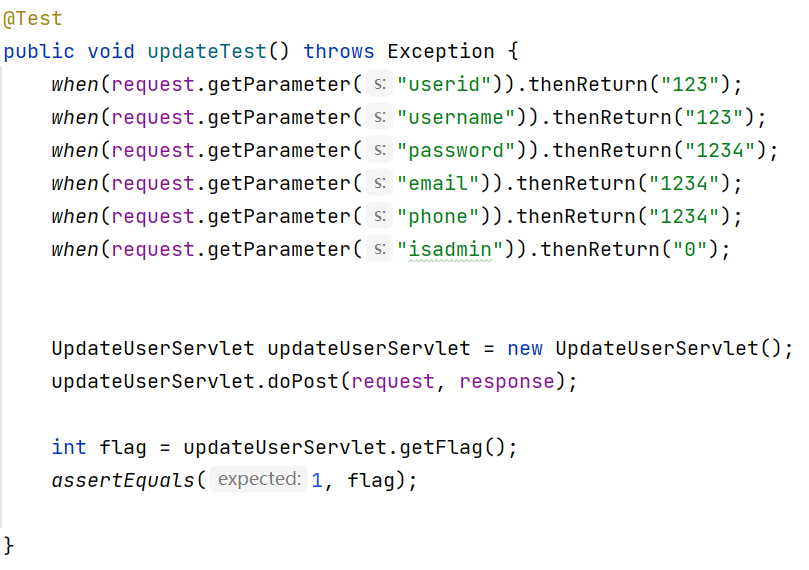


测试分析：模拟HTTP请求参数，包括用户ID、用户名、密码、电子邮件、电话和是否为管理员。通过assertEquals(1, flag);断言检查UserAddServlet的getFlag()方法返回值是否为0，表示添加失败。

（三）**updateTest**

测试目的：验证用户信息更新操作是否成功。

测试代码：



测试分析：模拟HTTP请求参数，包括用户ID、用户名、密码、电子邮件、电话和是否为管理员。通过assertEquals(1, flag);断言检查UpdateUserServlet的getFlag()方法返回值是否为1，表示更新成功。

（四）**deleteTest**

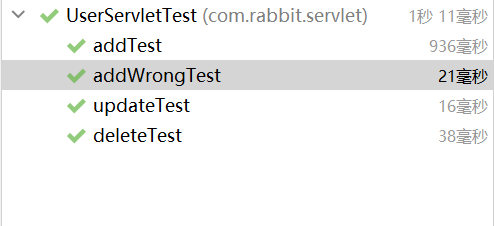
测试目的：验证用户删除操作是否成功。

测试代码：



测试分析：模拟HTTP请求参数，仅包括用户ID。通过assertEquals(1, flag);断言检查DeleteUserServlet的getFlag()方法返回值是否为1，表示删除成功。

测试结果：



（3）集成测试：

UserServletTest 集成测试分析

一、测试概述

本次集成测试针对UserServlet的用户管理功能进行了全面测试，包括正常添加用户、添加用户重复、更新用户数据、删除用户等多个场景，旨在用户管理功能的正确性、一致性和安全性。

二、测试用例执行情况

（一）**addTest**

测试目的：验证用户添加操作是否成功。

测试代码：



测试分析：模拟HTTP请求参数，包括用户ID、用户名、密码、电子邮件、电话和是否为管理员。通过assertEquals(1, flag);断言检查UserAddServlet的getFlag()方法返回值是否为1，表示添加成功。

（二）**addWrongTest**

测试目的：验证添加已有用户是否失败

测试代码：

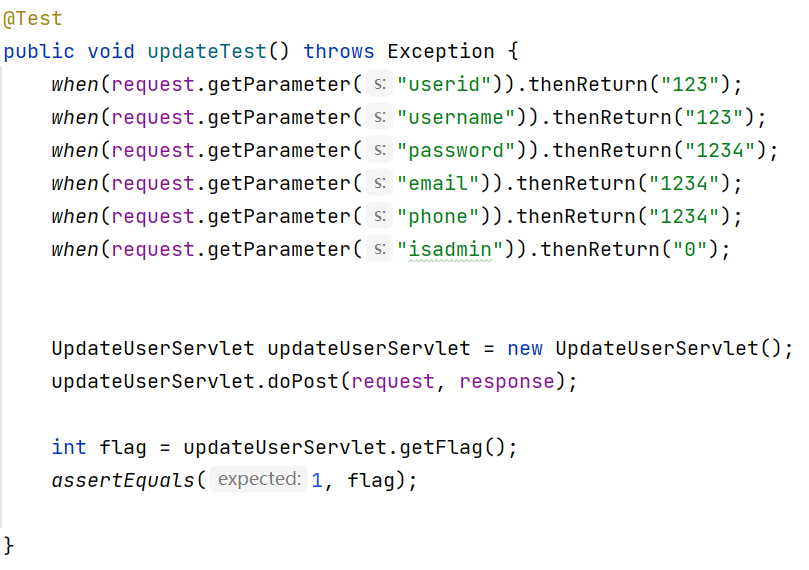


测试分析：模拟HTTP请求参数，包括用户ID、用户名、密码、电子邮件、电话和是否为管理员。通过assertEquals(1, flag);断言检查UserAddServlet的getFlag()方法返回值是否为0，表示添加失败。

（三）**updateTest**

测试目的：验证用户信息更新操作是否成功。

测试代码：



测试分析：模拟HTTP请求参数，包括用户ID、用户名、密码、电子邮件、电话和是否为管理员。通过assertEquals(1, flag);断言检查UpdateUserServlet的getFlag()方法返回值是否为1，表示更新成功。

（四）**deleteTest**

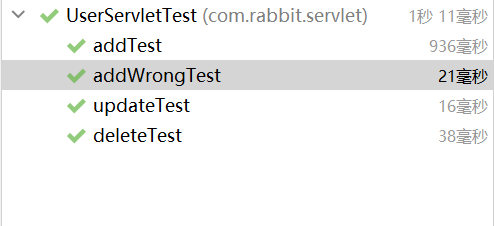
测试目的：验证用户删除操作是否成功。

测试代码：



测试分析：模拟HTTP请求参数，仅包括用户ID。通过assertEquals(1, flag);断言检查DeleteUserServlet的getFlag()方法返回值是否为1，表示删除成功。

测试结果：

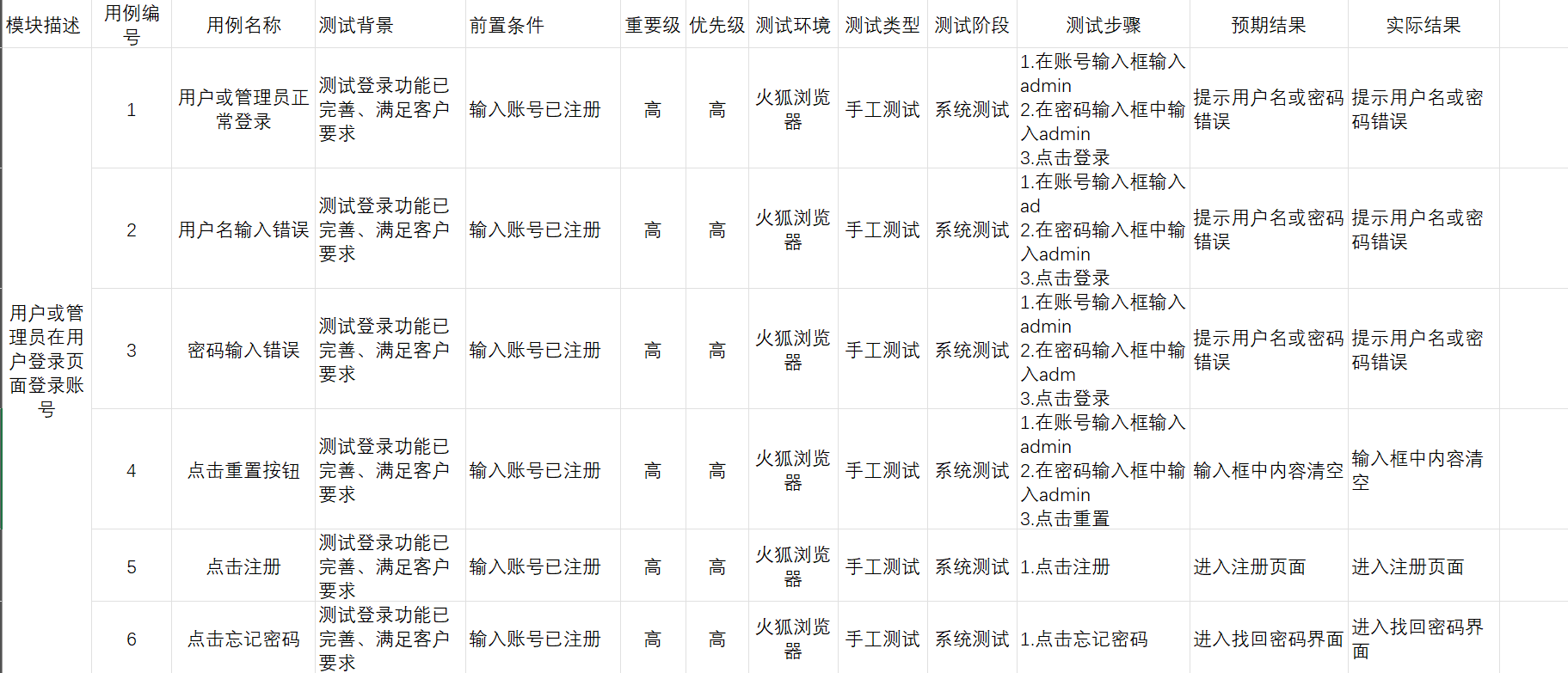


5.系统功能测试

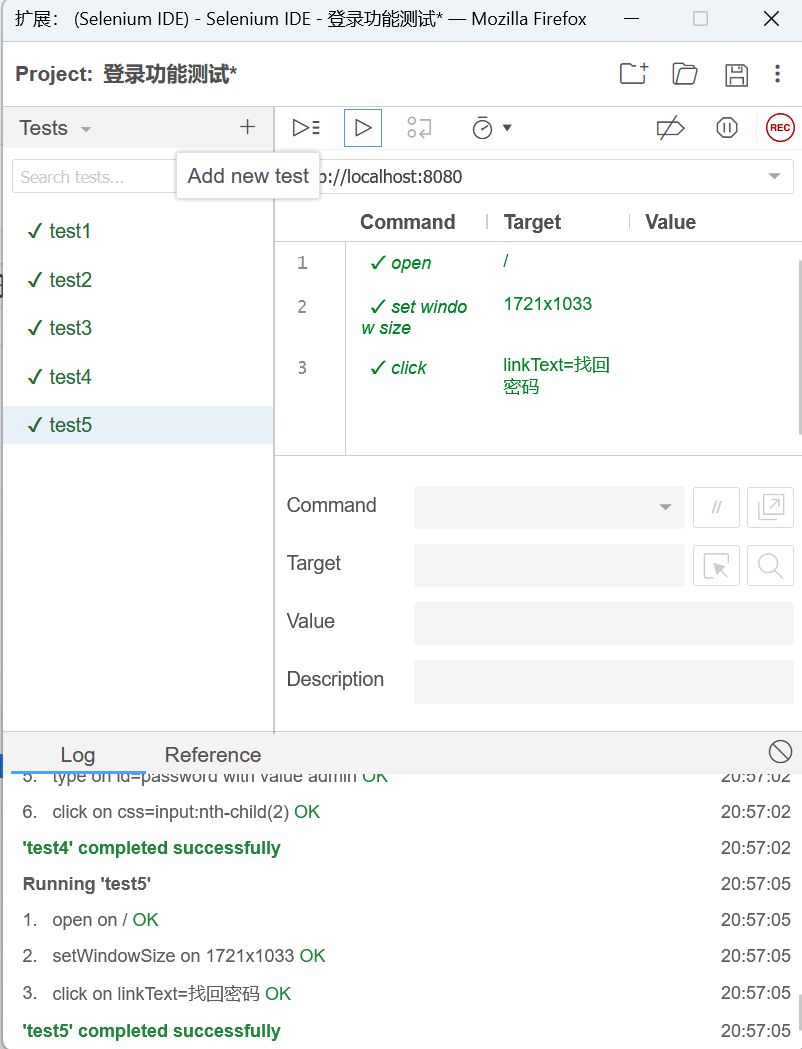
功能测试：

编写测试用例，使用Selenium IDE进行自动化测试

1.用户登录功能：

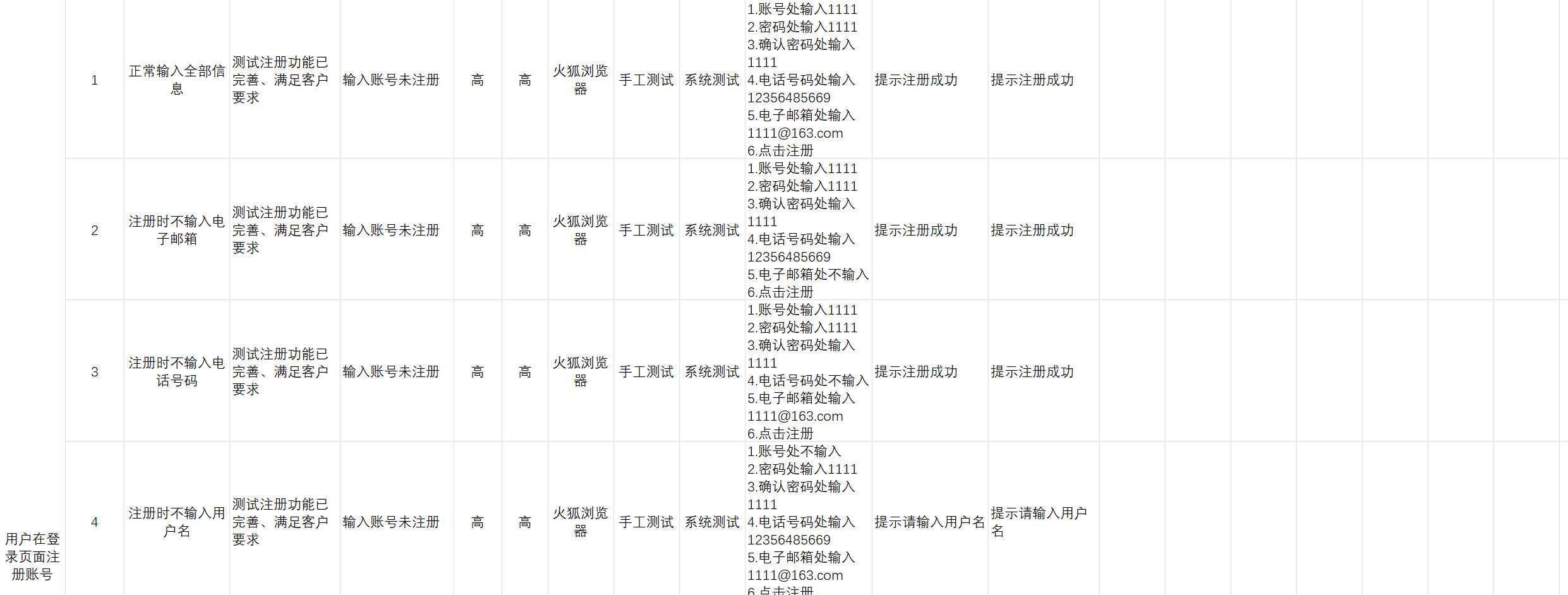
测试用例

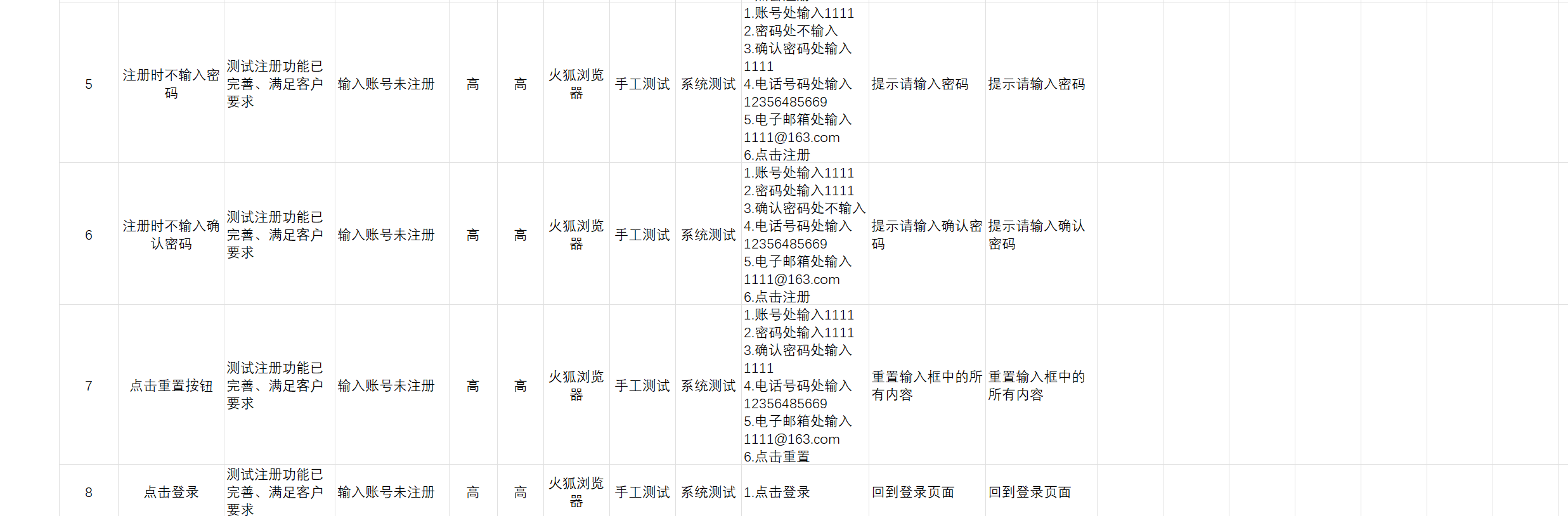
测试用户登录的功能是否正常，注册和忘记密码是否能使用



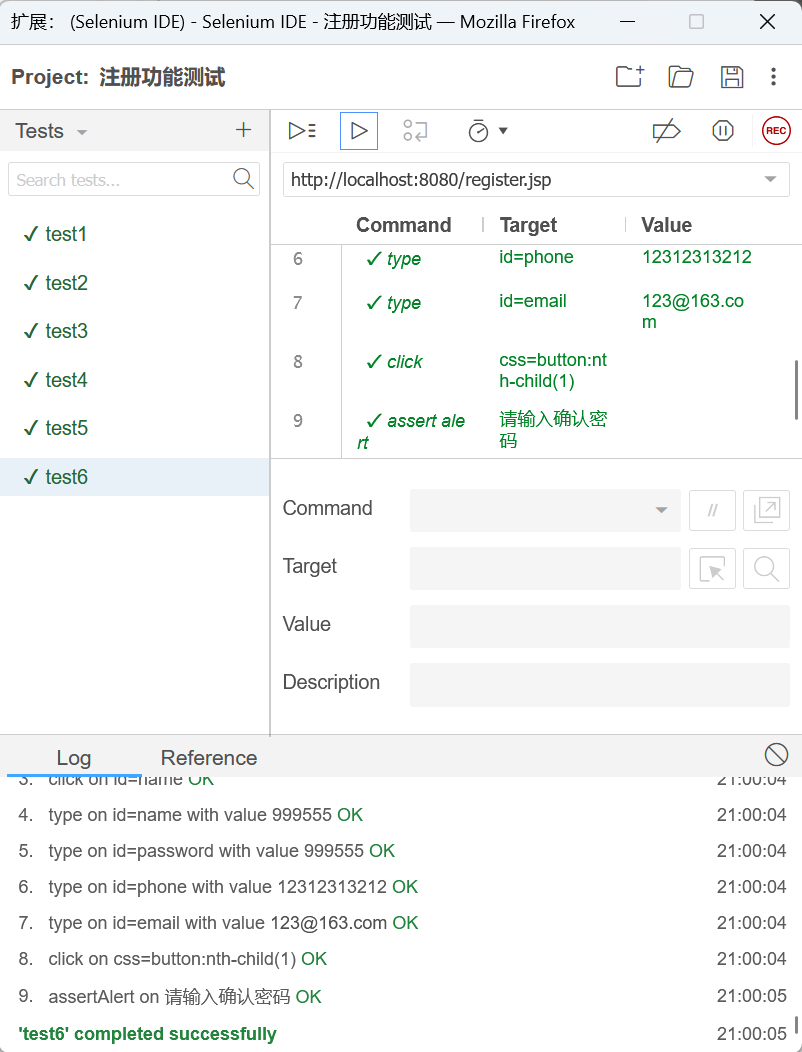
2.注册功能

测试用例：





测试注册功能是否正常使用：

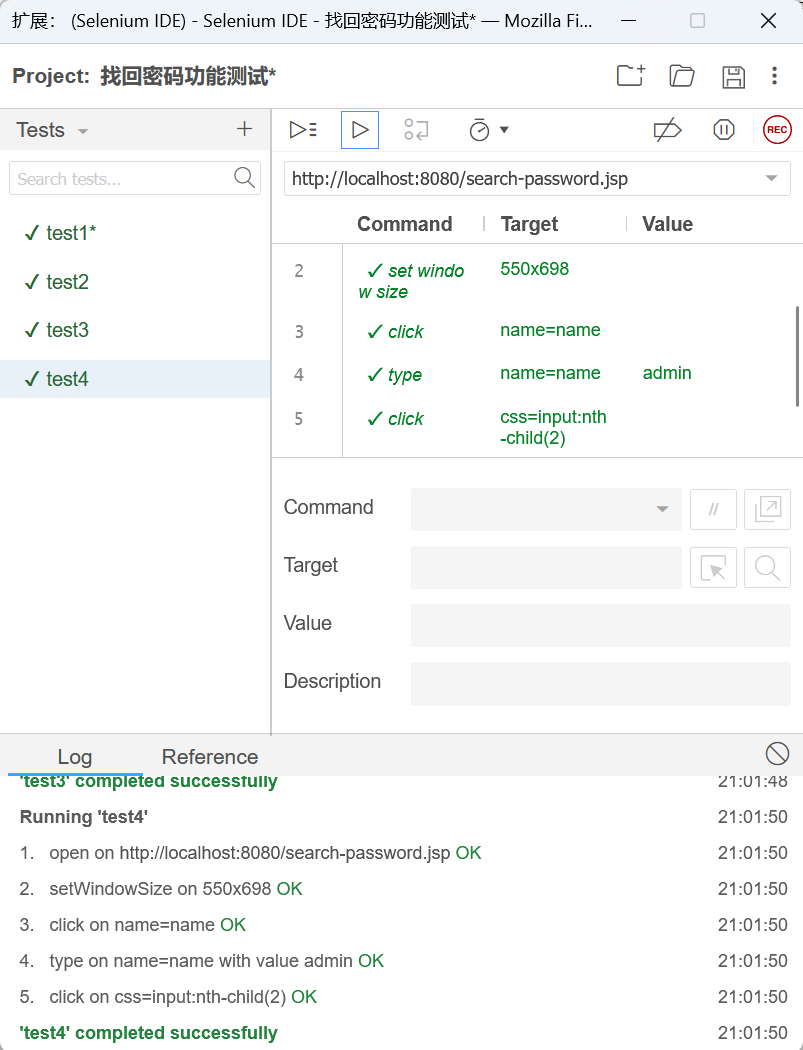


3.忘记密码功能

测试用例：

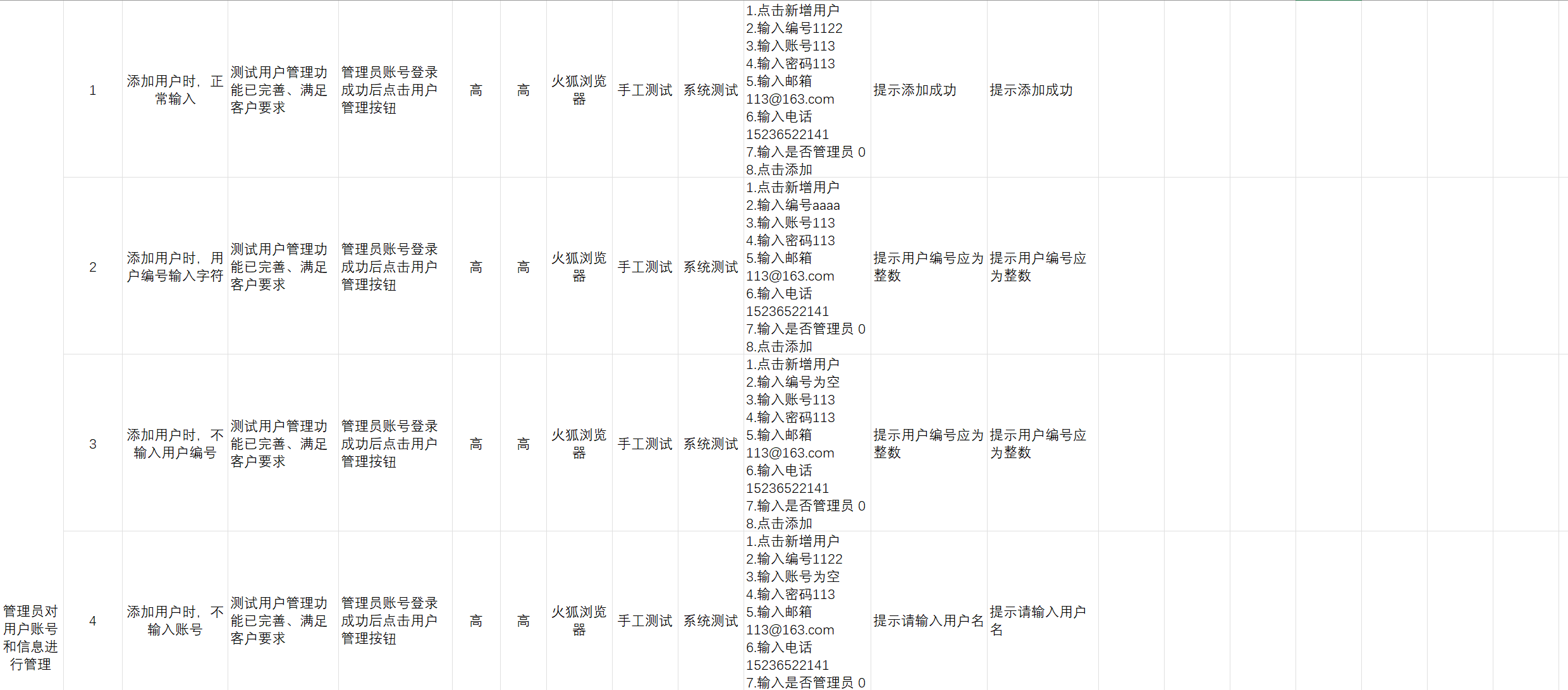


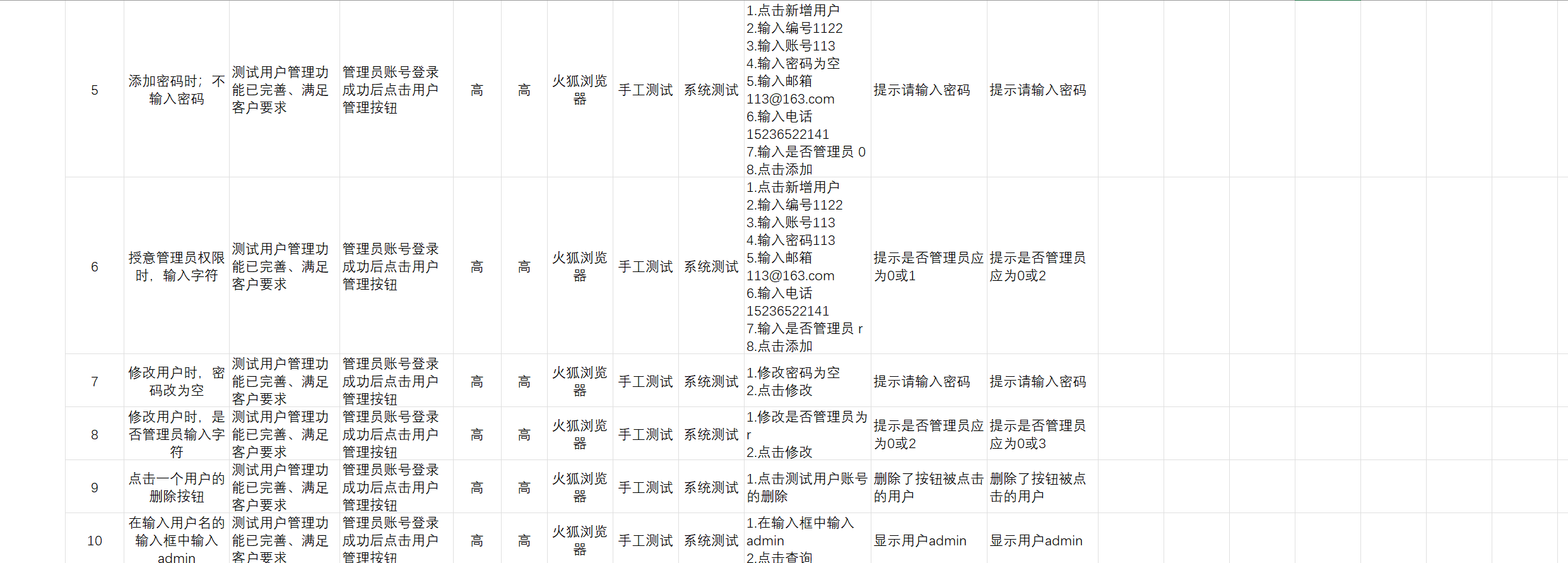
测试忘记密码的功能是否完整：



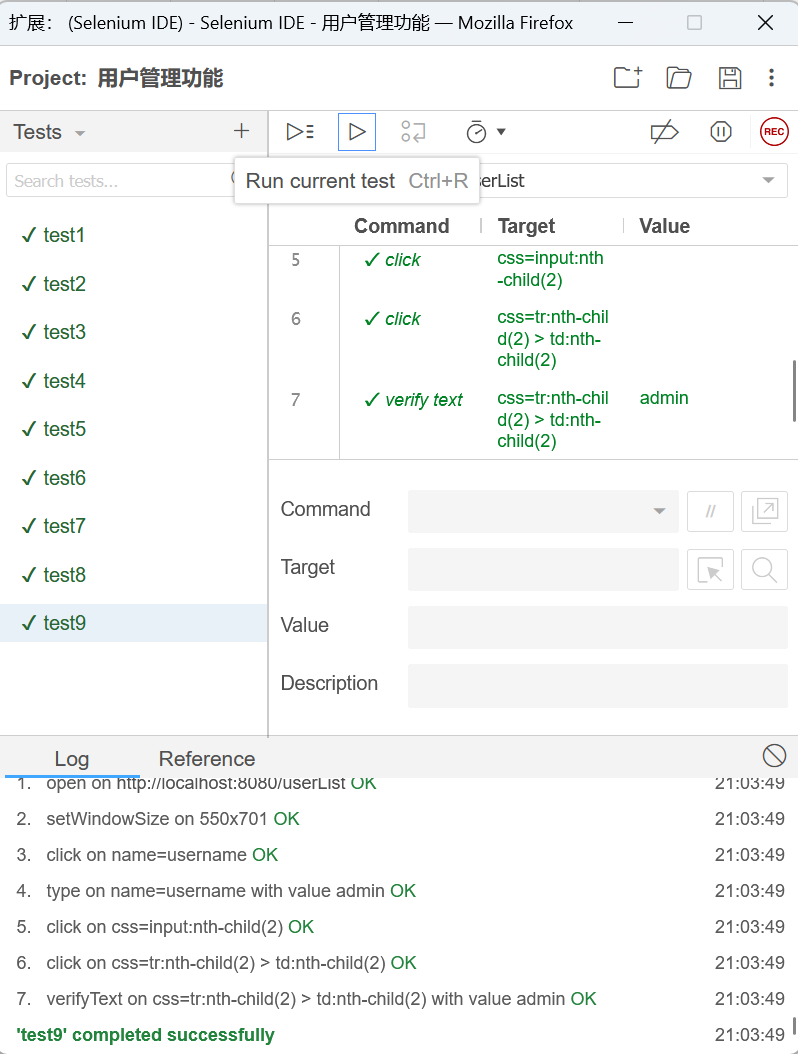
4.用户管理功能：

测试用例：



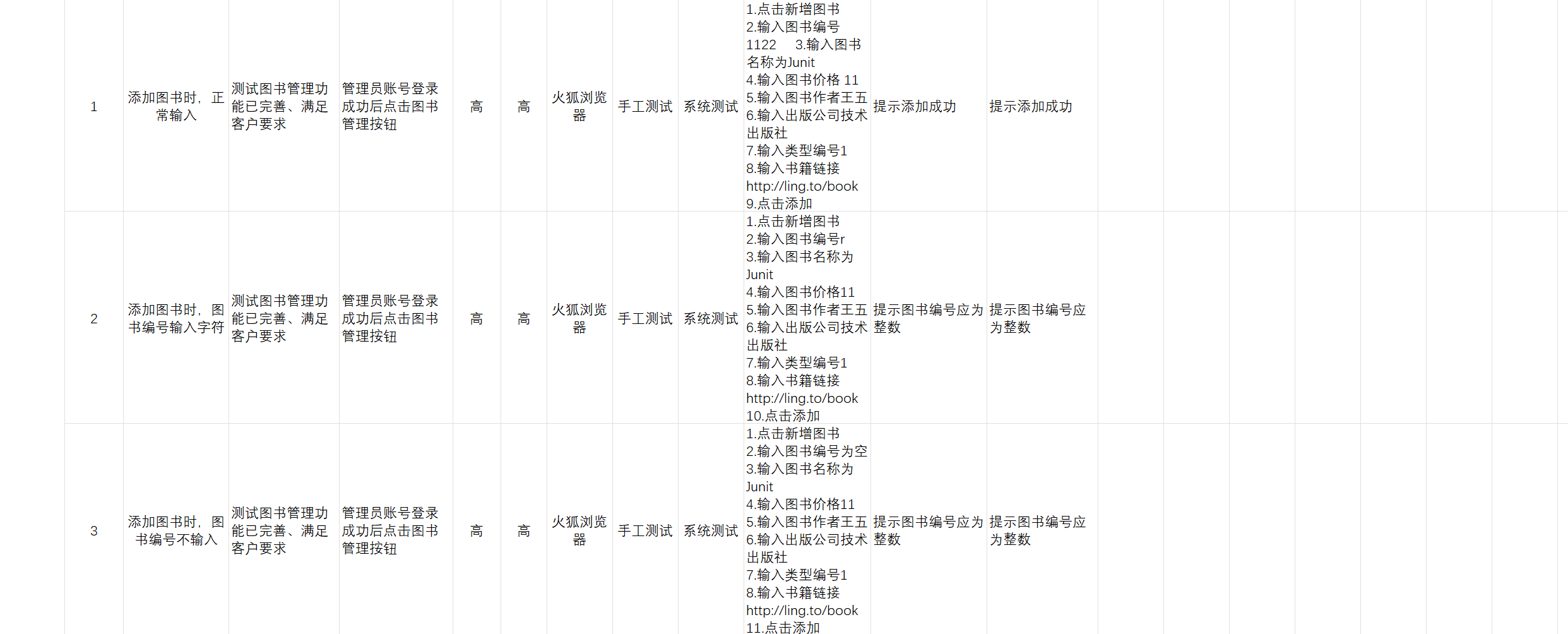


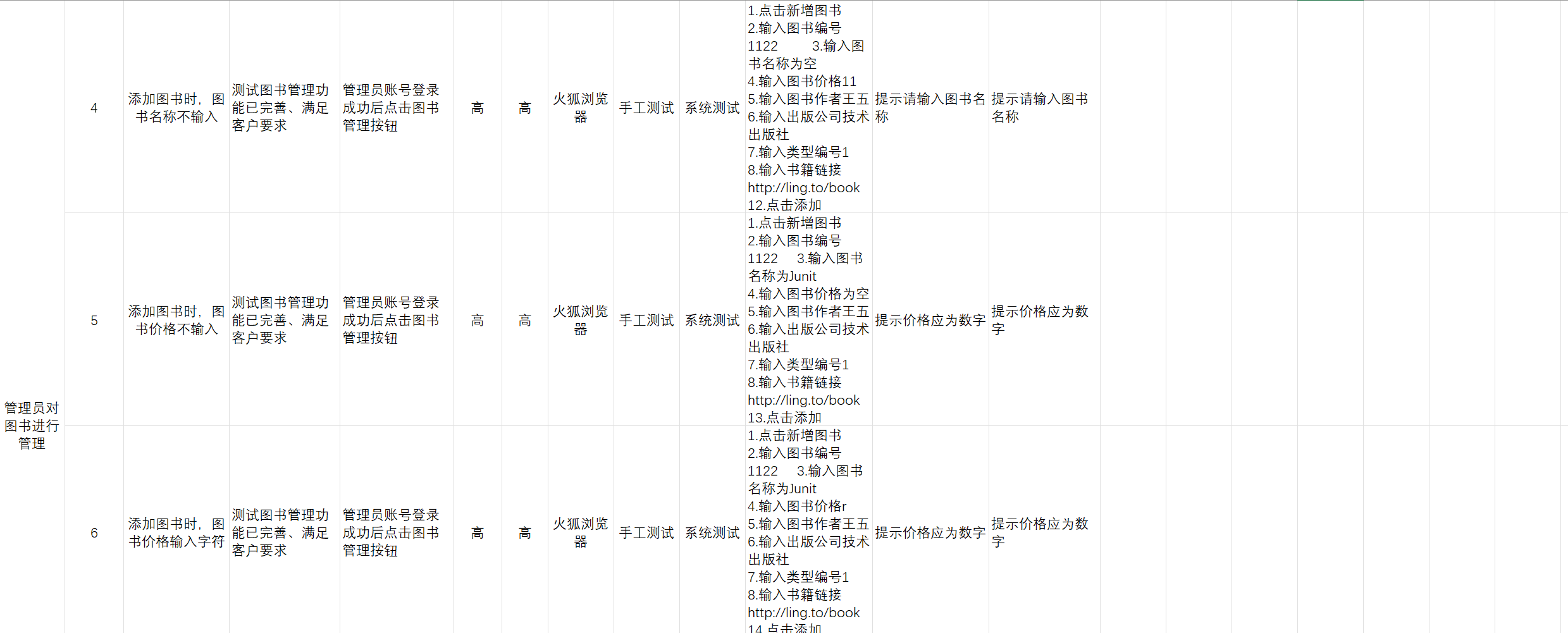
测试用户管理功能的添加用户，修改用户等功能是否完善：

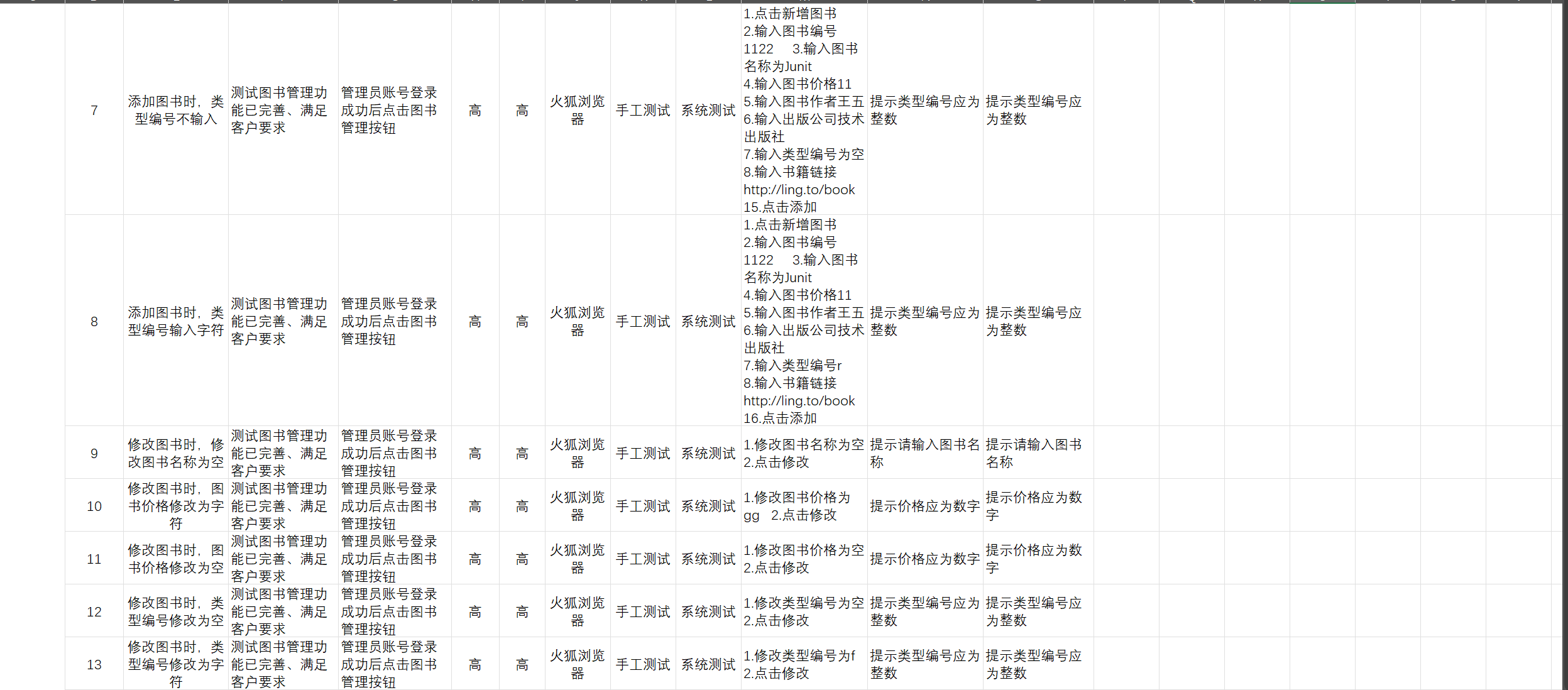


5.图书管理功能：

测试用例：

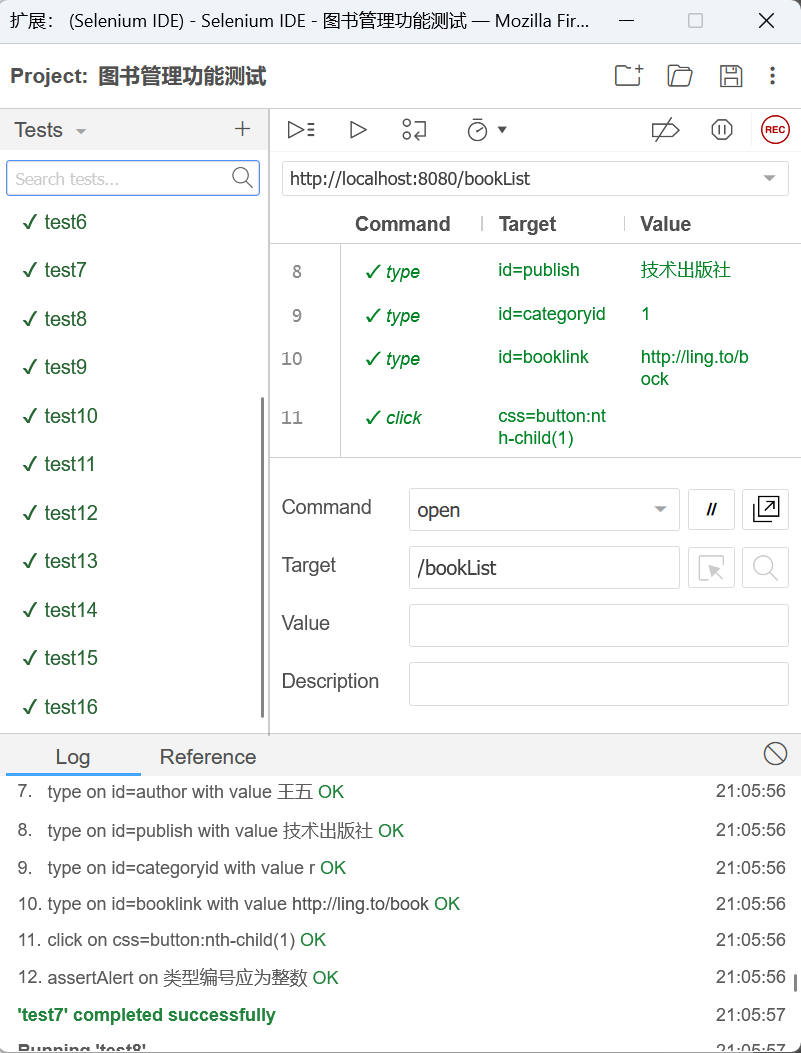






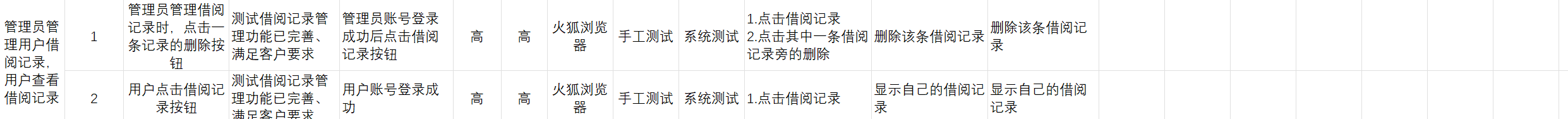


测试图书管理功能的添加图书，修改图书等功能是否完善

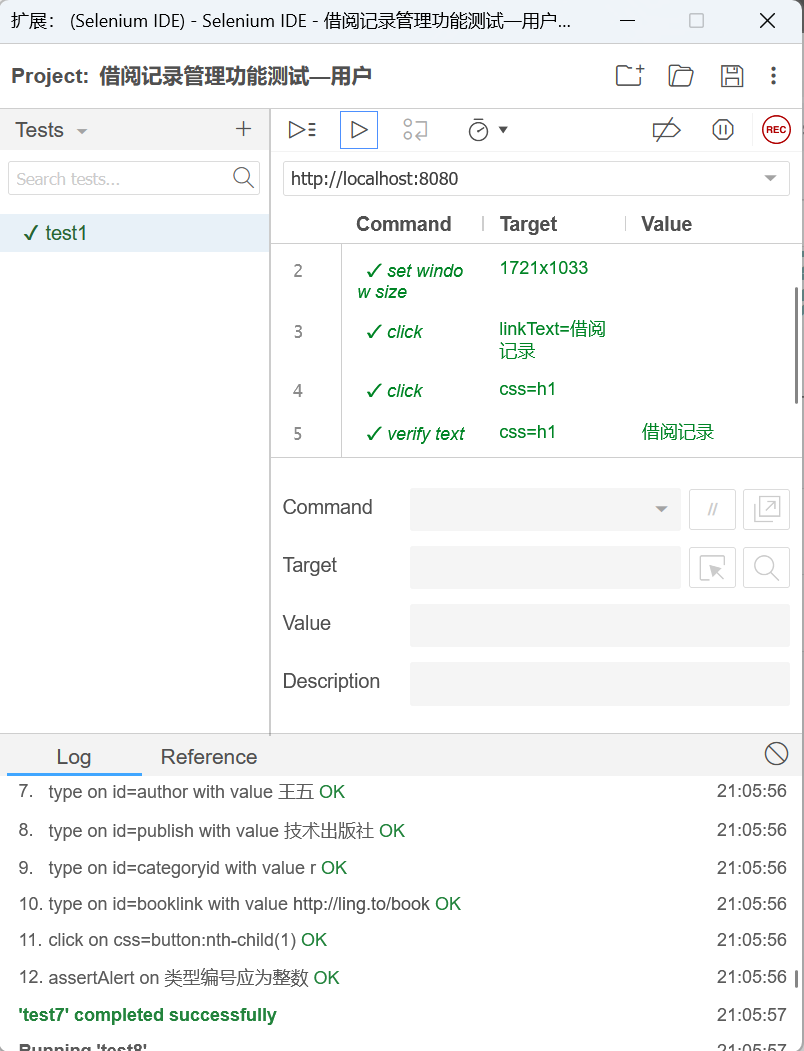
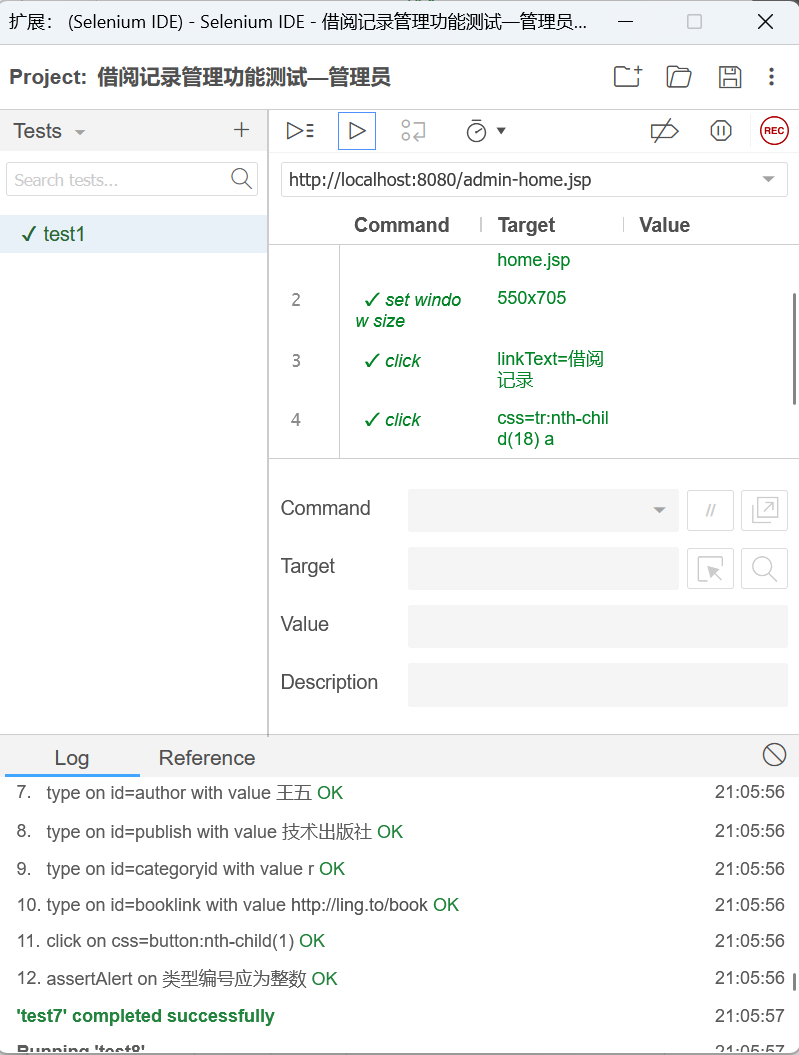


6.借阅记录管理功能：

测试用例



测试管理员对借阅记录的管理和用户查看借阅记录是否完善：

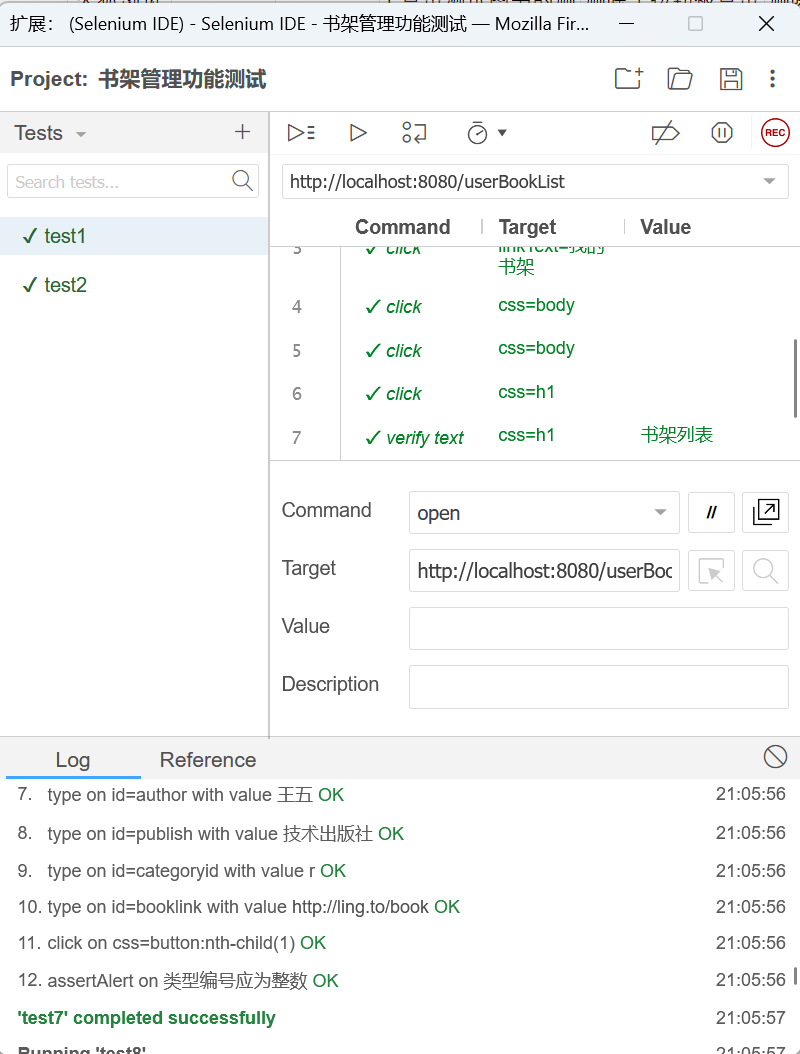


7.书架管理功能：

测试用例：

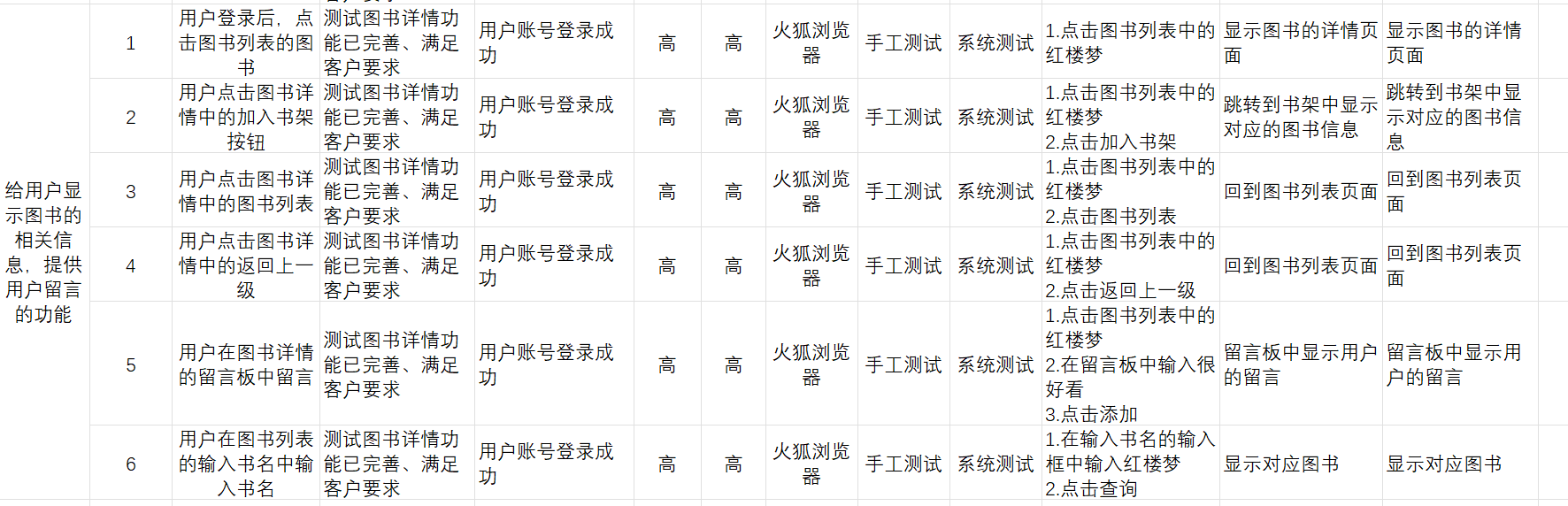


测试用户对书架的管理是否能正常使用：

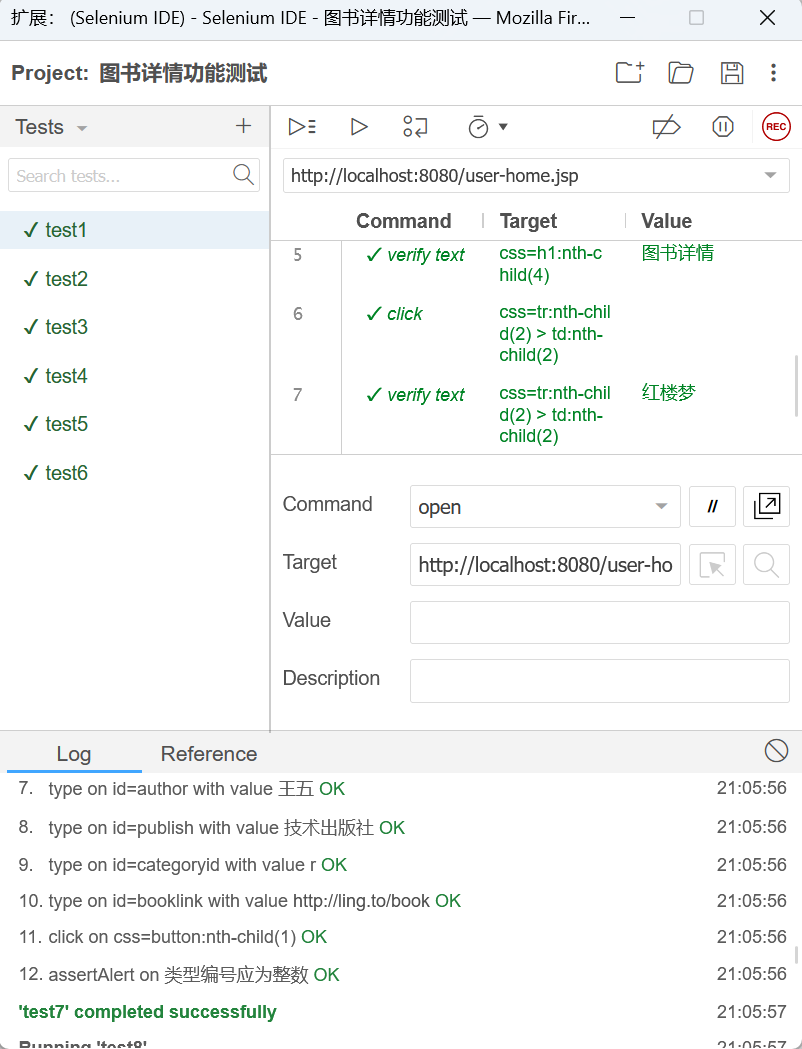


8.图书详情功能：

测试用例：

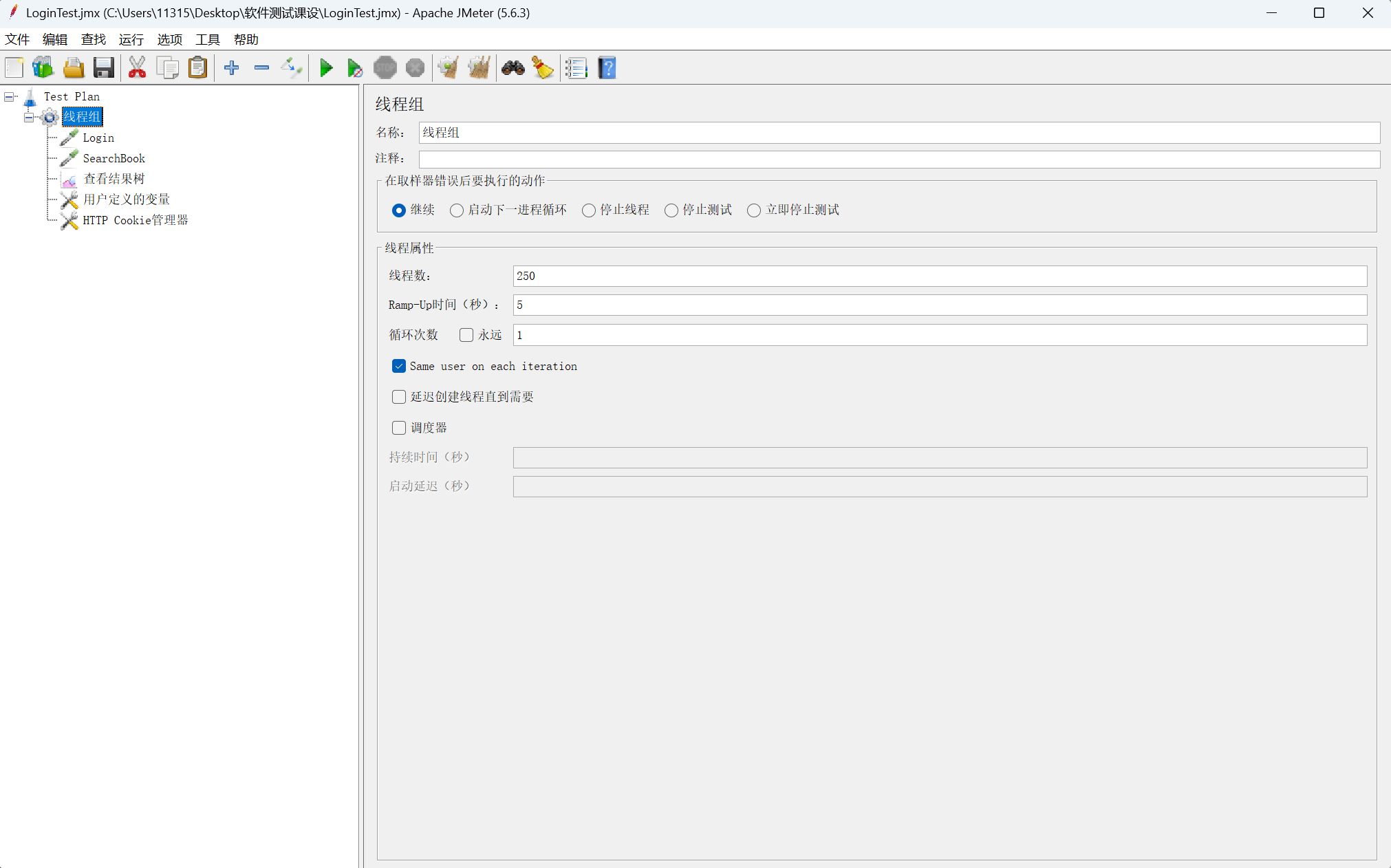


测试图书的显示信息是否正常，用户能否留言等功能

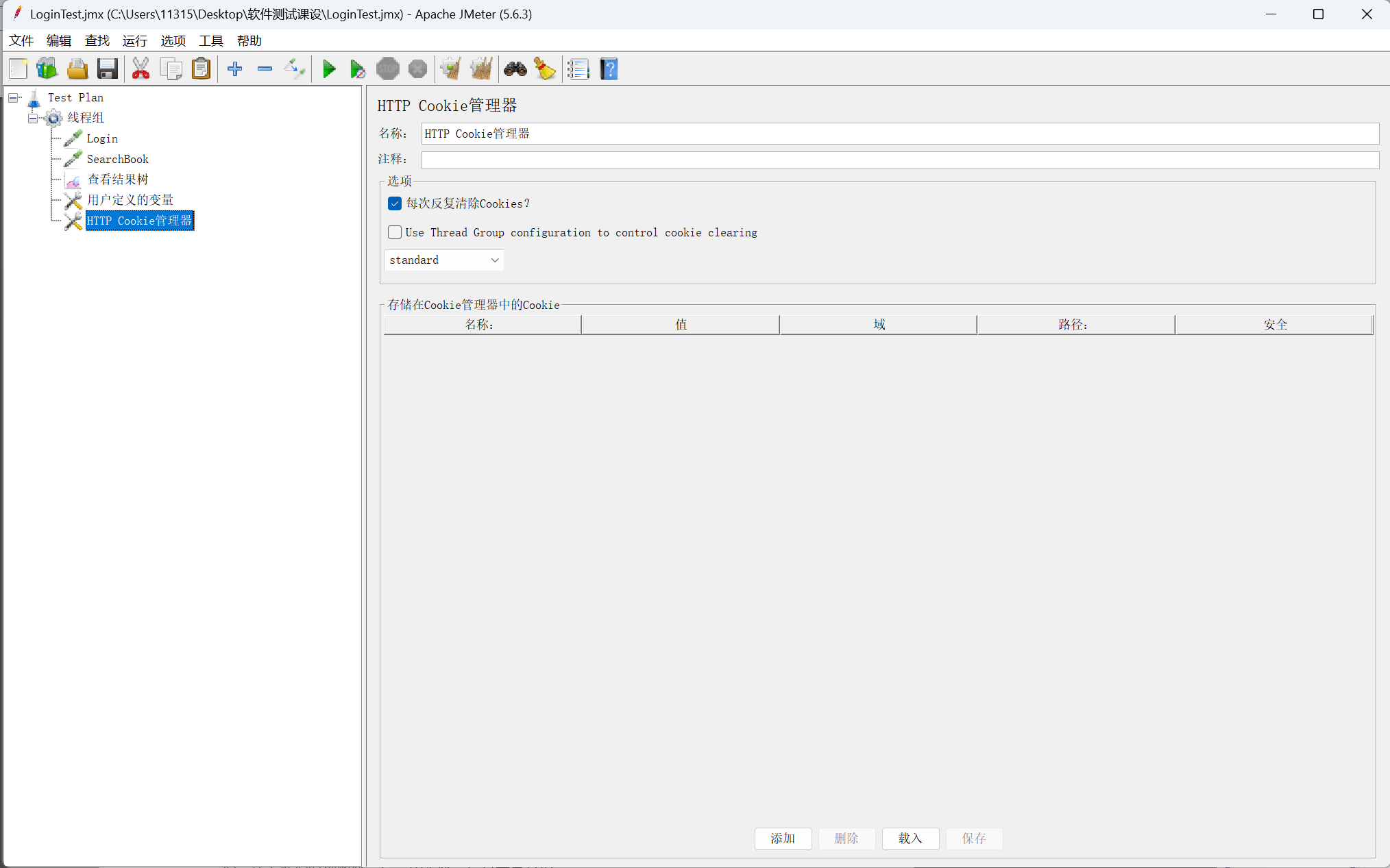


6.性能测试

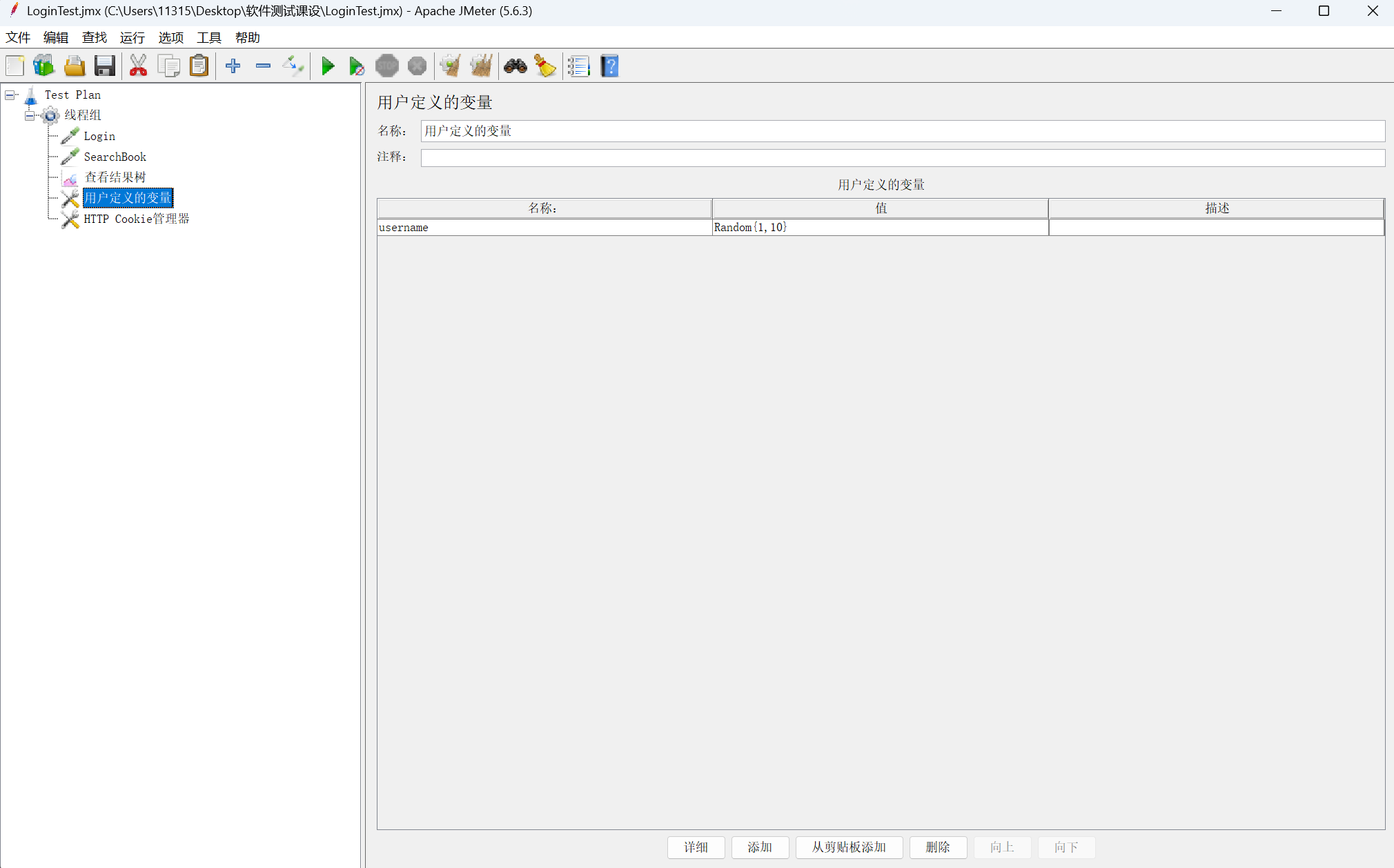
1. 设置线程组，为了模拟高峰期，我们选择让250组用户在5秒内上线，也就是1秒上线50个用户。



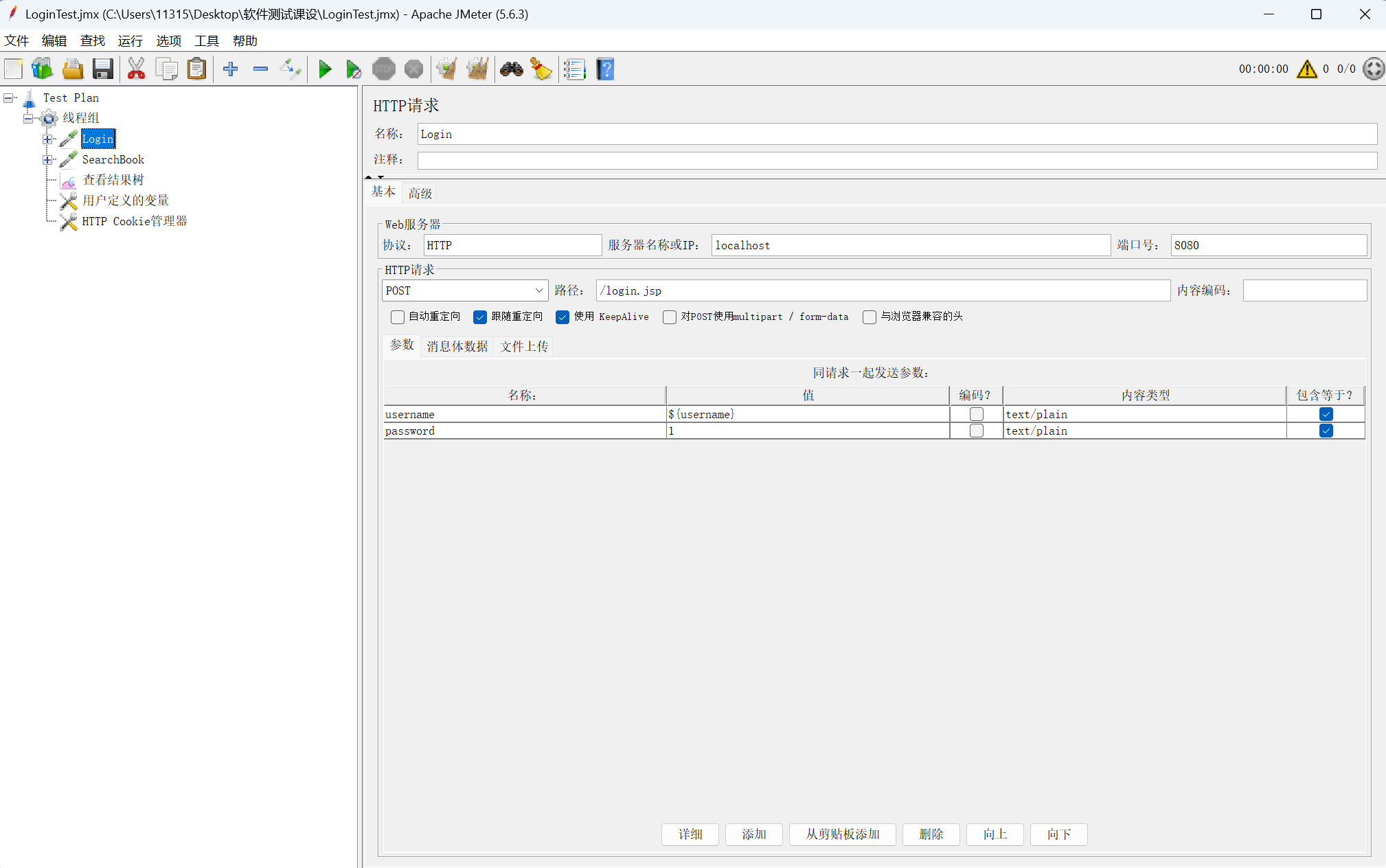
1. 设定每轮测试时都清除cookie，模拟真实用户的独立会话行为，防止影响下一次测试的结果。用户在关闭浏览器后重新打开访问网站，或者在不同的设备上登录同一个账号，这些情况下通常会开始新的会话，之前的 Cookie 信息不会被保留。



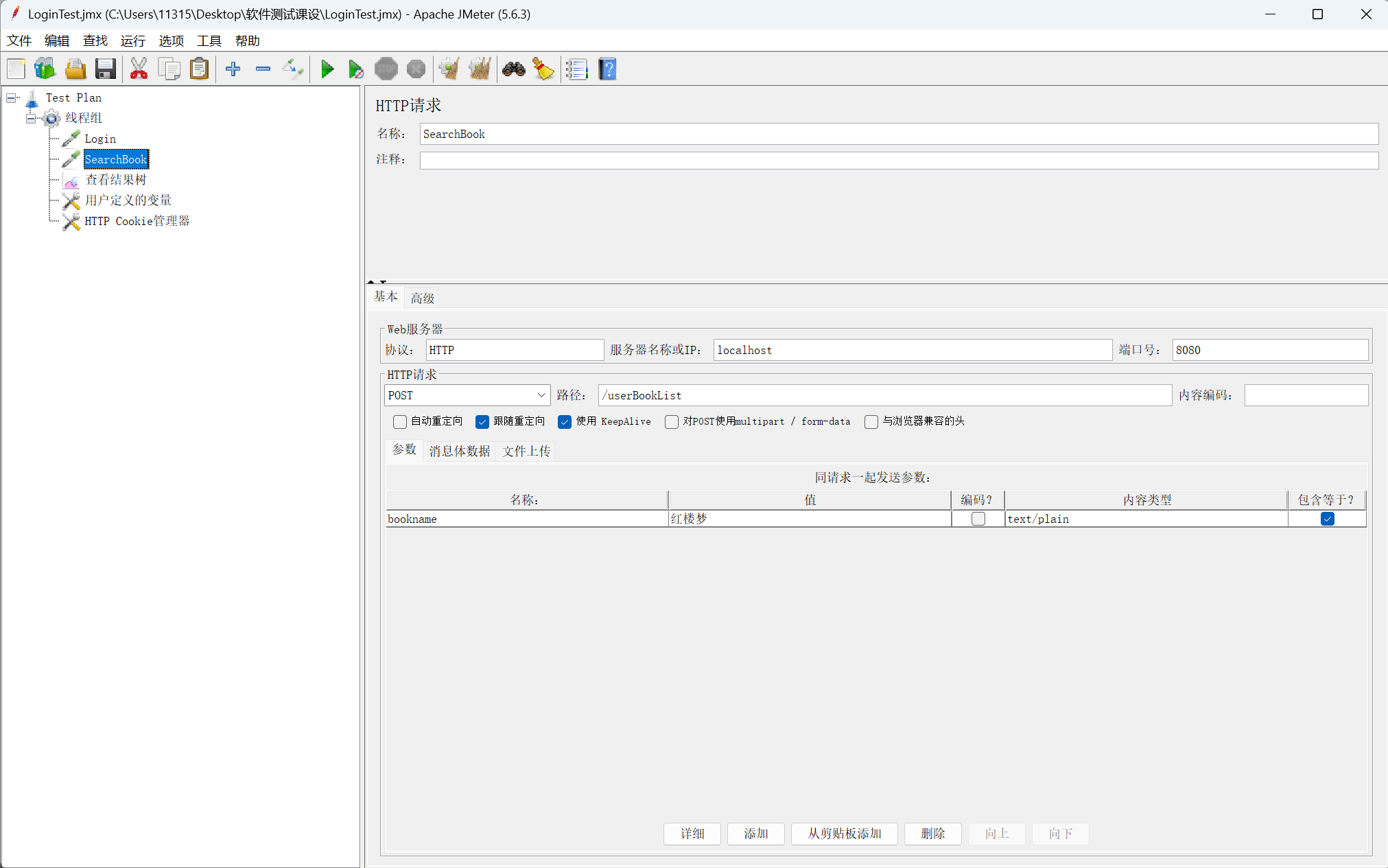
1. 为线程组设置测试用户登录的测试数据，username用Random随机1-10的数字



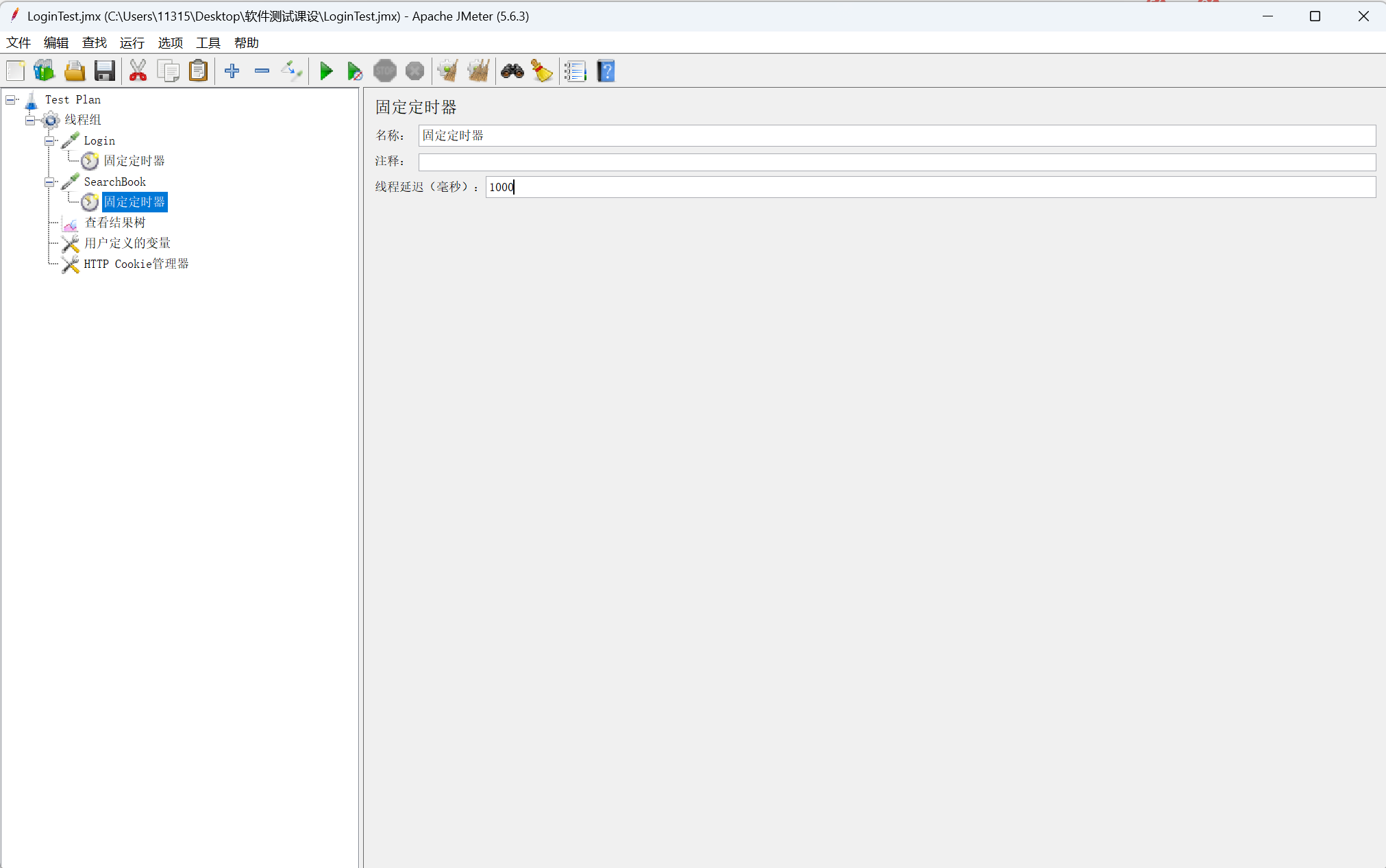
1. 设置登录，参数为username和password， 密码设置恒定为1



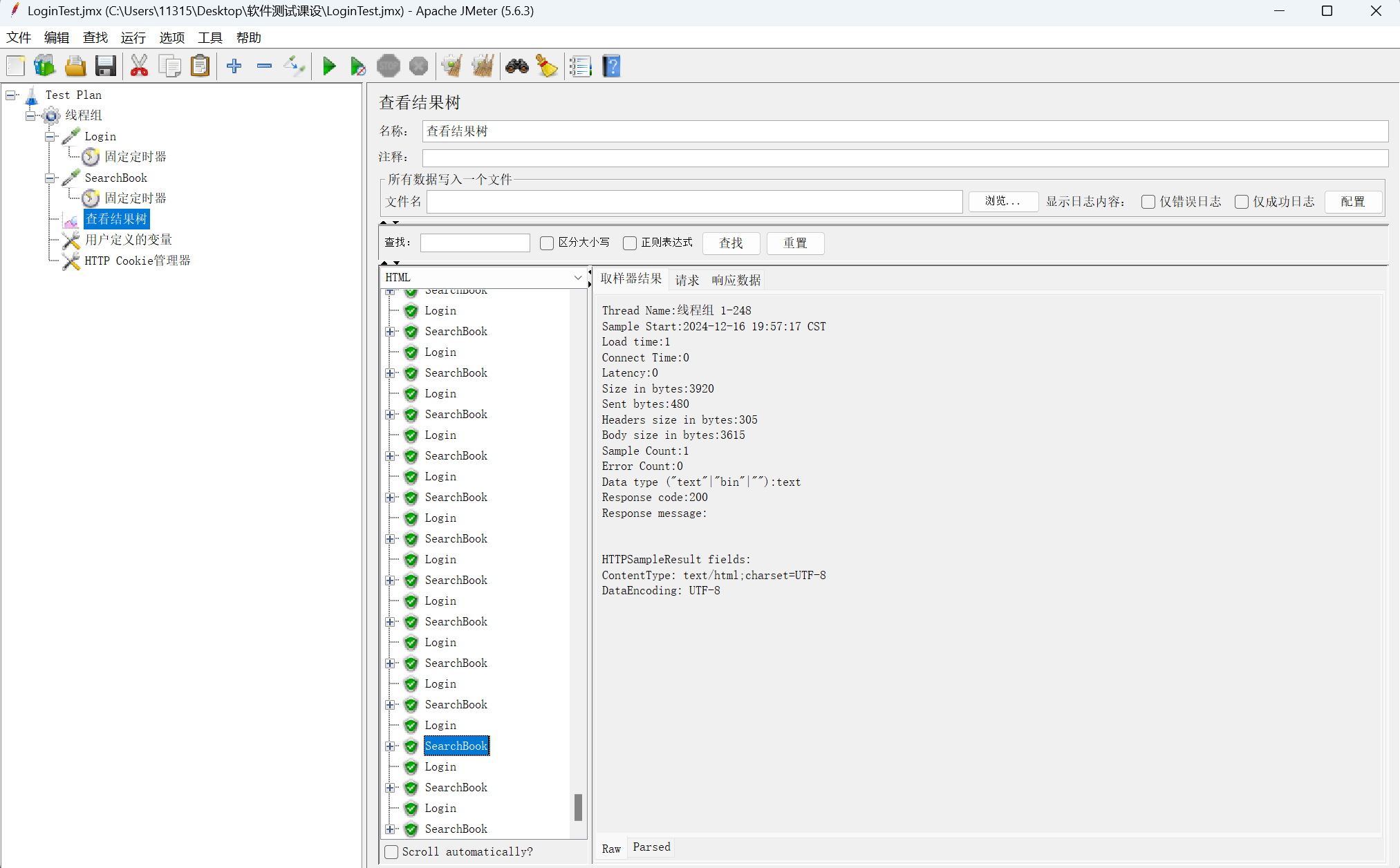
1. 设置查找图书，均查找红楼梦



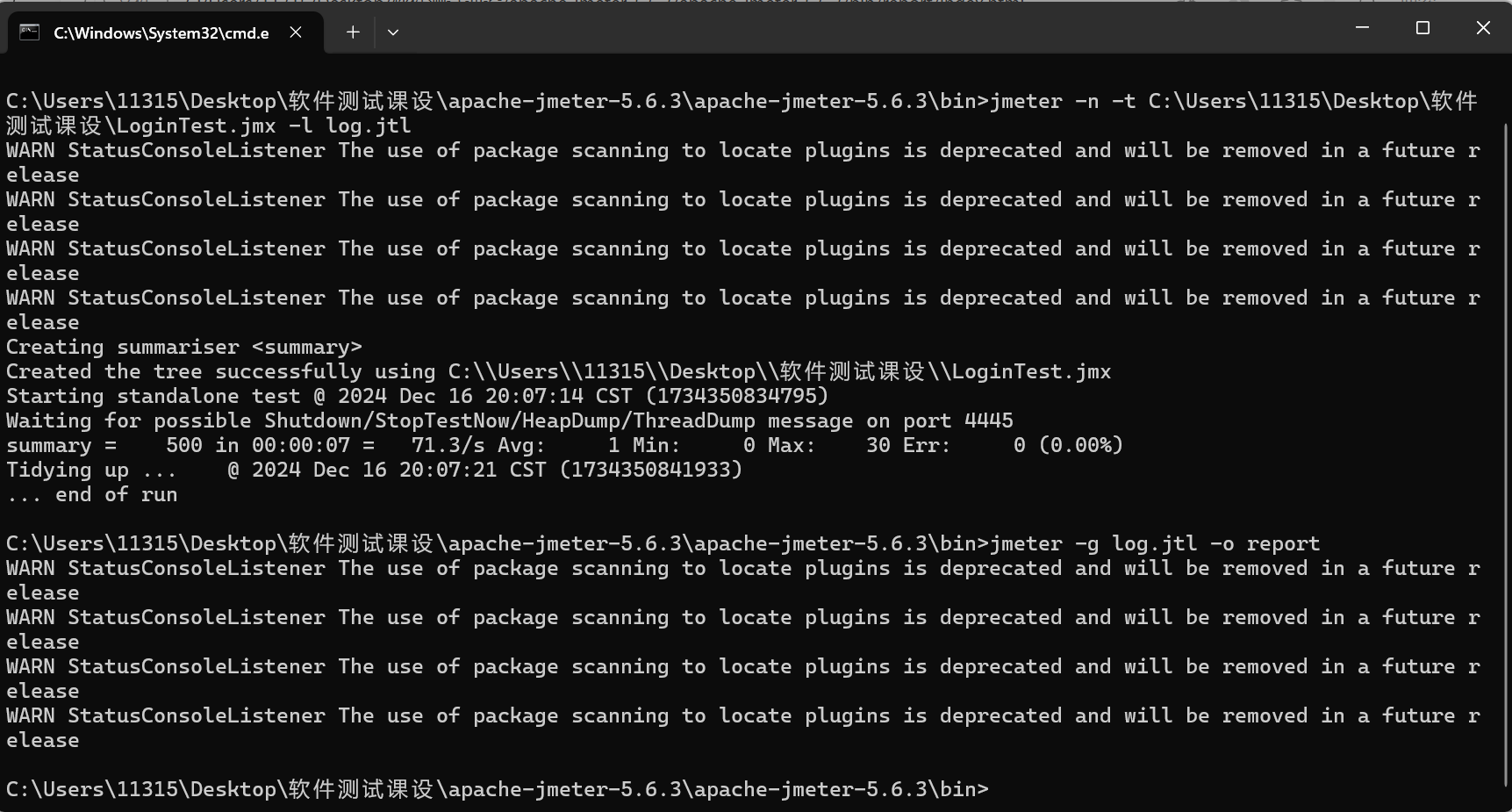
1. 在各个操作之间插入定时器，模拟更真实的操作。



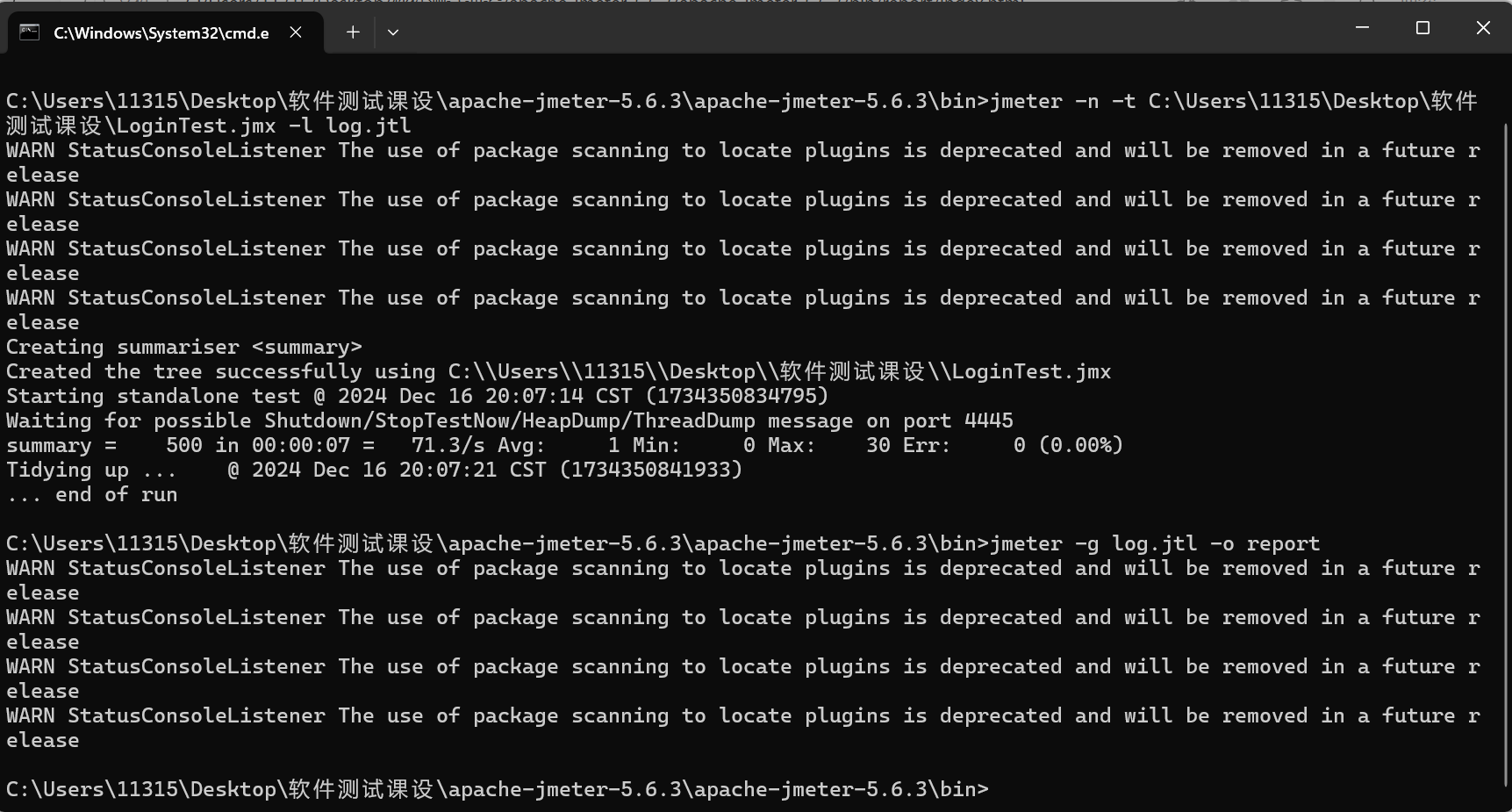
7. 设置查看结果树，查看运行后的结果



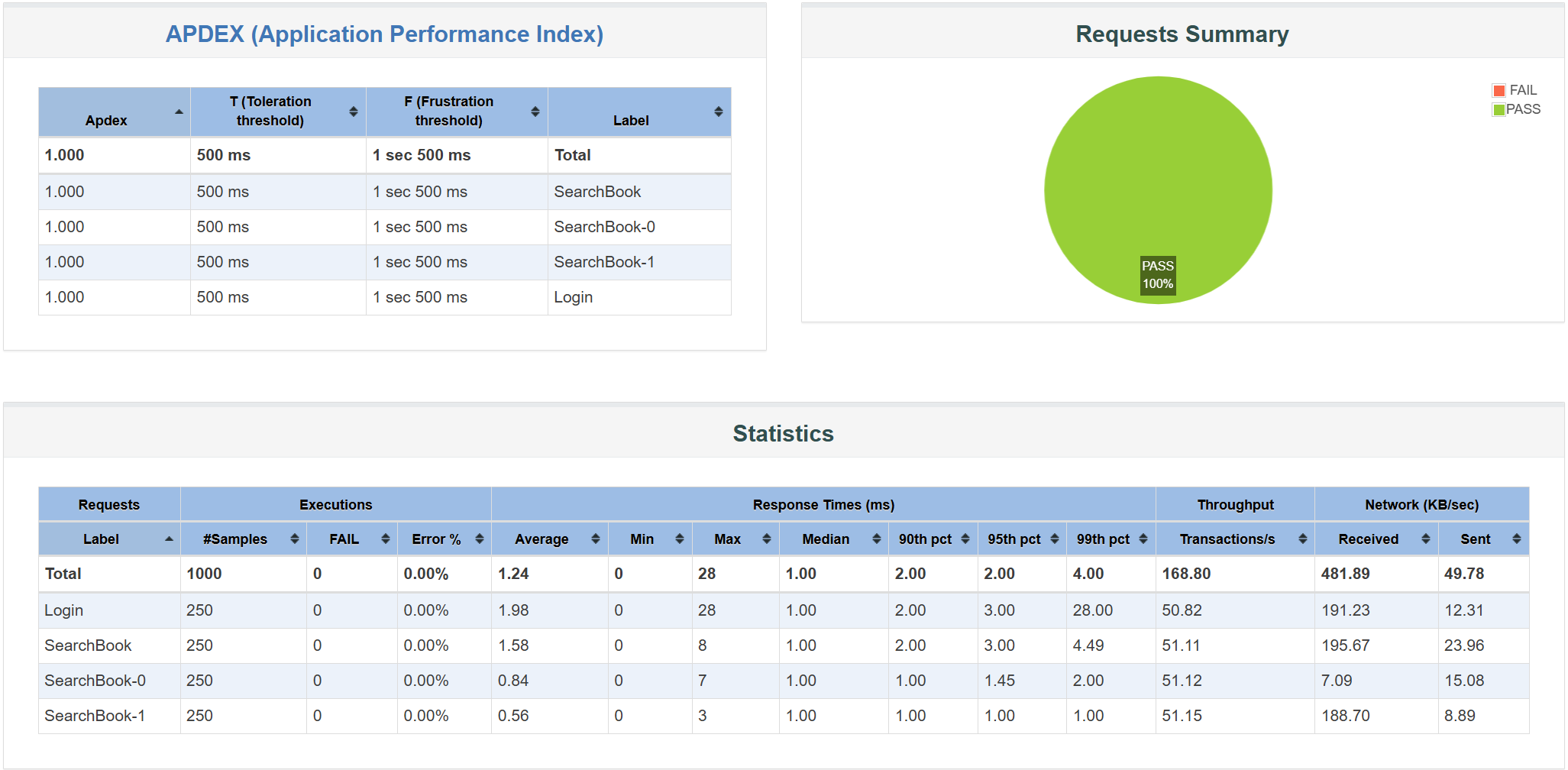
8. 在jmeter的bin目录中打开cmd,在命令行输入jmeter -n -t C:\Users\11315\Desktop\软件测试课设\LoginTest.jmx -l log.jtl，以命令行模式运行实施性能测试。



9. 在命令行输入jmeter -g log.jtl -o report1，就会产生report1目录，里面的index.html 打开就是报告。

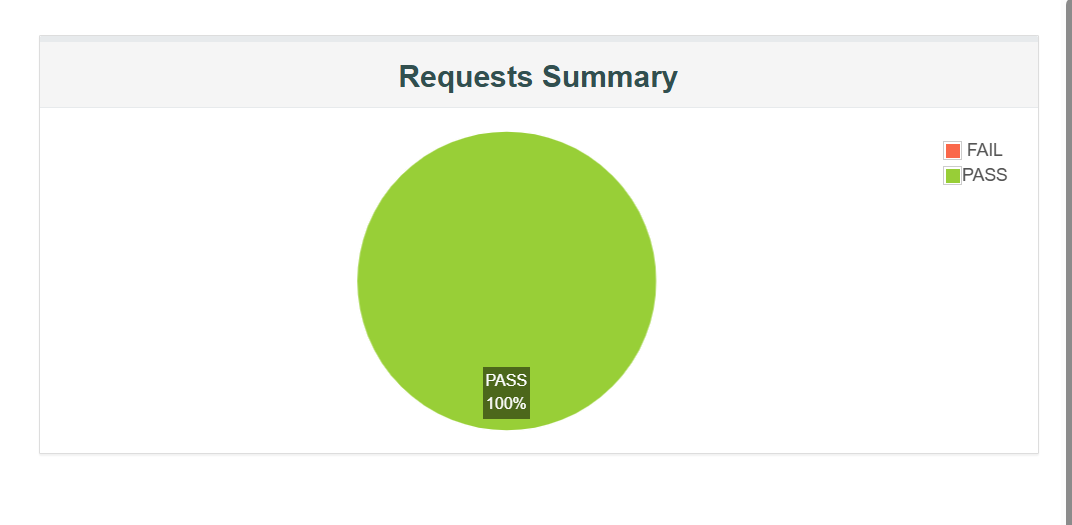


10. 用浏览器打开report下的index.xml

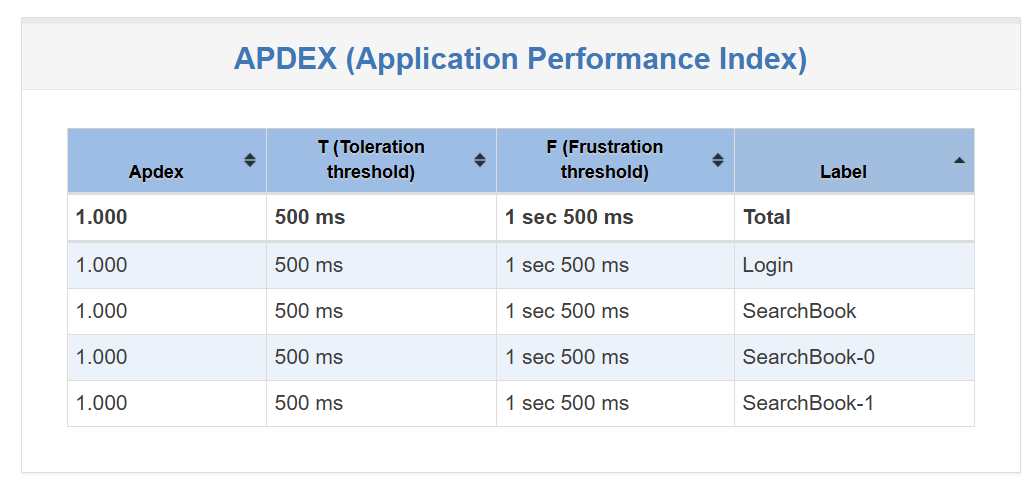


**结果分析：**

在所有请求中，错误率均为0.00%，表明系统在高并发情况下能够稳定处理请求，没有出现崩溃或错误，具有高并发处理能力。



对书店管理系统而言，用户查找图书，需要系统较低的响应时间，所以我们设定总体的用户容忍响应时间为500ms，最大容忍1500ms，测试结果Apdex（满意度）值越大表现越好，Apdex为1表明用户体验很满意。



在1000个sample中，平均响应时间为1.24毫秒，表明系统响应速度较快。



代码档：

