目录

[科研成果管理系统实现报告 1](#_Toc58622820)

[一、项目基本信息 2](#_Toc58622821)

[1．项目人员 2](#_Toc58622822)

[2．项目执行情况 2](#_Toc58622823)

[二、系统任务书 3](#_Toc58622824)

[1．系统概述 3](#_Toc58622825)

[2．系统角色 3](#_Toc58622826)

[3. 用户使用流程图 4](#_Toc58622827)

[4. 数据字典 7](#_Toc58622828)

[5 安全性需求 8](#_Toc58622829)

[6. 运行环境需求 9](#_Toc58622830)

[三、概念结构设计 10](#_Toc58622831)

[1．E-R图 10](#_Toc58622832)

[2．遇到的问题及解决方案 10](#_Toc58622833)

[四、系统设计 11](#_Toc58622834)

[1．总体设计 11](#_Toc58622835)

[2．数据结构 12](#_Toc58622836)

[3. 技术选型 14](#_Toc58622837)

[4. 物理设计 15](#_Toc58622838)

[5. 界面设计（可选） 16](#_Toc58622839)

[五、系统测试 19](#_Toc58622840)

[1．系统概述 19](#_Toc58622841)

[2．安装及部署说明 20](#_Toc58622842)

科研成果管理系统实现报告

# 一、项目基本信息

1．项目人员

组长：崔冠宇

组员：邵宁录、吴丝弦、张晨阳、张配天

2．项目执行情况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 任务 | 起始日期 | 结束日期 | 任务负责人  及工作量比例 | 任务参与人  及工作量比例 |
| 任务书撰写 | 2020.10.30 | 2020.11.3 | 崔冠宇 20% | 邵宁录 20%  吴丝弦 20%  张晨阳 20%  张配天 20% |
| ER图设计/绘制 | 2020.11.3 | 2020.11.10 | 张晨阳 100% | —— |
| 前端页面 | 2020.11.6 | 2020.12.10 | 崔冠宇 35% | 邵宁录 5%  吴丝弦 35%  张晨阳 20%  张配天 5% |
| 后端通信 | 2020.11.6 | 2020.12.9 | 张配天 35% | 崔冠宇 10%  邵宁录 35%  吴丝弦 5%  张晨阳 15% |
| 实现报告撰写 | 2020.11.17 | 2020.12.12 | 崔冠宇 40% | 邵宁录 15%  吴丝弦 15%  张晨阳 15%  张配天 15% |

# 二、系统任务书

1．系统概述

###### 1.1 系统名称

本系统名称是科研项目管理系统。

###### 1.2 设计目的

经过实际体验当前系统的录入流程以及采访张晓莹老师的使用感受，我们了解到现有系统存在着许多问题：例如教师录入项目过于复杂、审核管理员无法修改信息只能驳回等。所以我们设计本系统的主要目的是为了简化教师录入科研成果及学院审批的流程。

###### 1.3 主要功能

按照登录账号的身份不同，系统界面和功能略有不同：

普通用户（主要指教师）登录后，主页显示未审核的项目的基本信息，用户可以点击左侧侧边栏中的主页按钮、添加按钮和查询按钮来跳转到相应页面。在添加页面，用户可以填写项目信息表格（仅需填写少部分必须内容，其它部分可以通过查询期刊级别表等辅助表自动填充，也可由管理员在审核时补充填写）并提交。在查询页面，用户可以按条件筛选\*\*自己\*\*已审核通过的项目，也可在此页面导出为 Excel 表格。

管理员登录后，主页显示等待审核的项目，每个条目右边有审核按钮，可直接跳转审核页面。管理员也可以点击主页左侧侧边栏中的主页按钮、添加按钮、查询按钮和用户管理按钮。在添加界面中，管理员也可以添加项目。在查询界面中，管理员可以按条件查询\*\*全院\*\*所有已审核通过的项目，并可导出 Excel 表格。在用户管理界面中，管理员可以创建新账号以提供给新用户使用。在审核页面中，管理员可以审核未审核的项目，可以对项目信息进行修改、补充。

2．系统角色

###### 2.1 角色1：

###### 普通用户（教师）

1. 查询：查询\*\*自己的\*\*已经入库（审核通过的）论文，可以按各种条件筛选查询论文，同时可以导出查询结果
2. 添加：增加新的论文记录（需要提交管理员审核、补充信息）
3. 修改：修改未入库的论文记录
4. 个人信息：修改密码

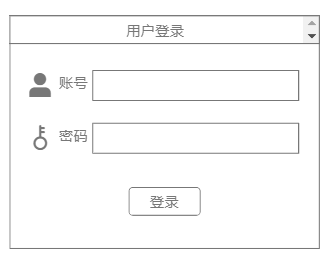
###### 2.2 角色2：

###### 管理员（张老师）

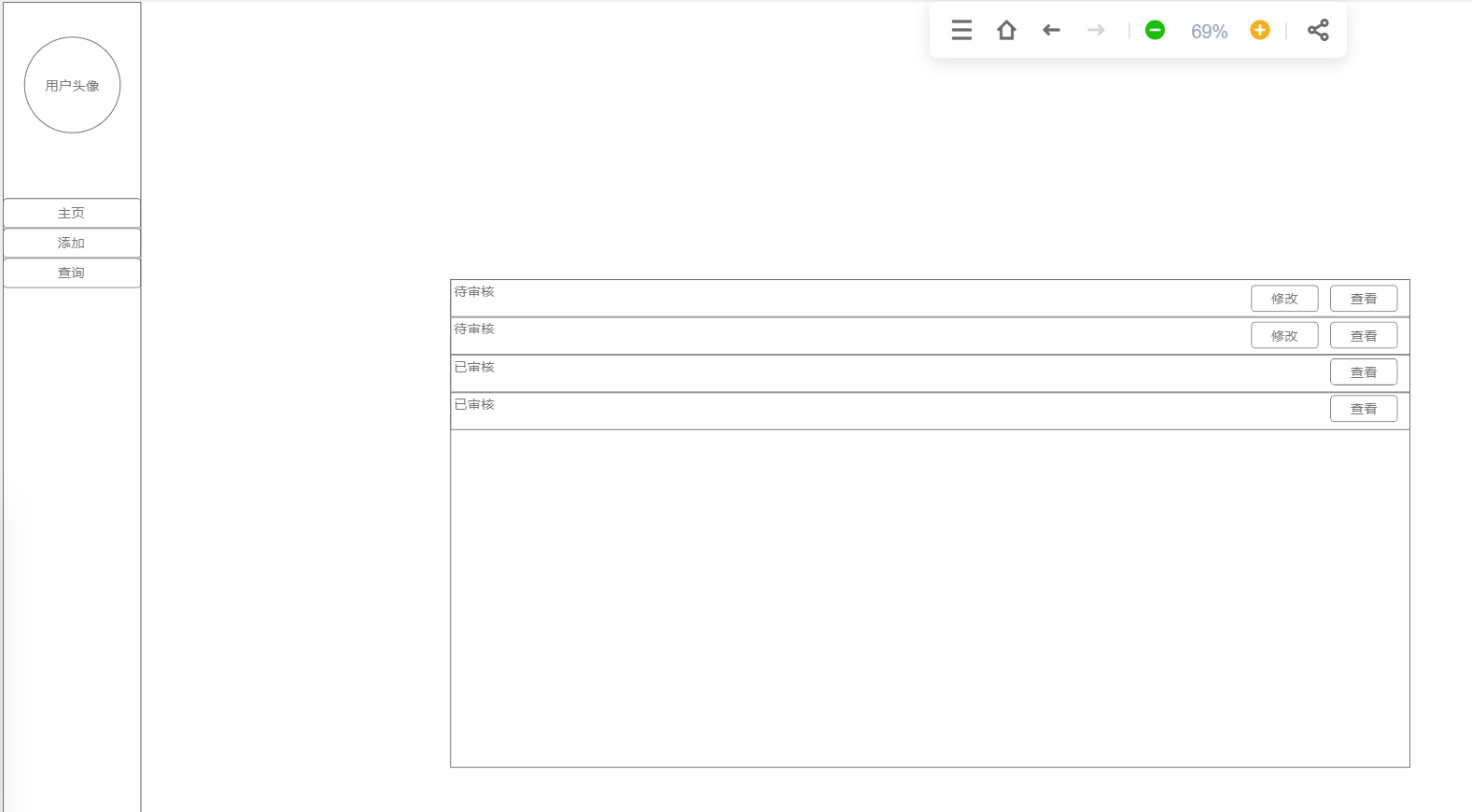
1. 查询：查询已经入库（审核通过的）论文
2. 审核
   1. 审核教师提交的论文记录（可以同时修改、补充内容，也可驳回）
   2. 插入新的会议、期刊库的信息（可选）
3. 修改
   1. 修改已经入库（审核通过的）论文信息
   2. 修改会议、期刊库的信息（可选）
4. 删除
   1. 删除已经入库（审核通过的）论文
   2. 删除会议、期刊库的信息（可选）
5. 用户相关
   1. 创建用户
   2. 删除用户（可选）
   3. 给忘记密码的用户重置密码

## 3. 用户使用流程图

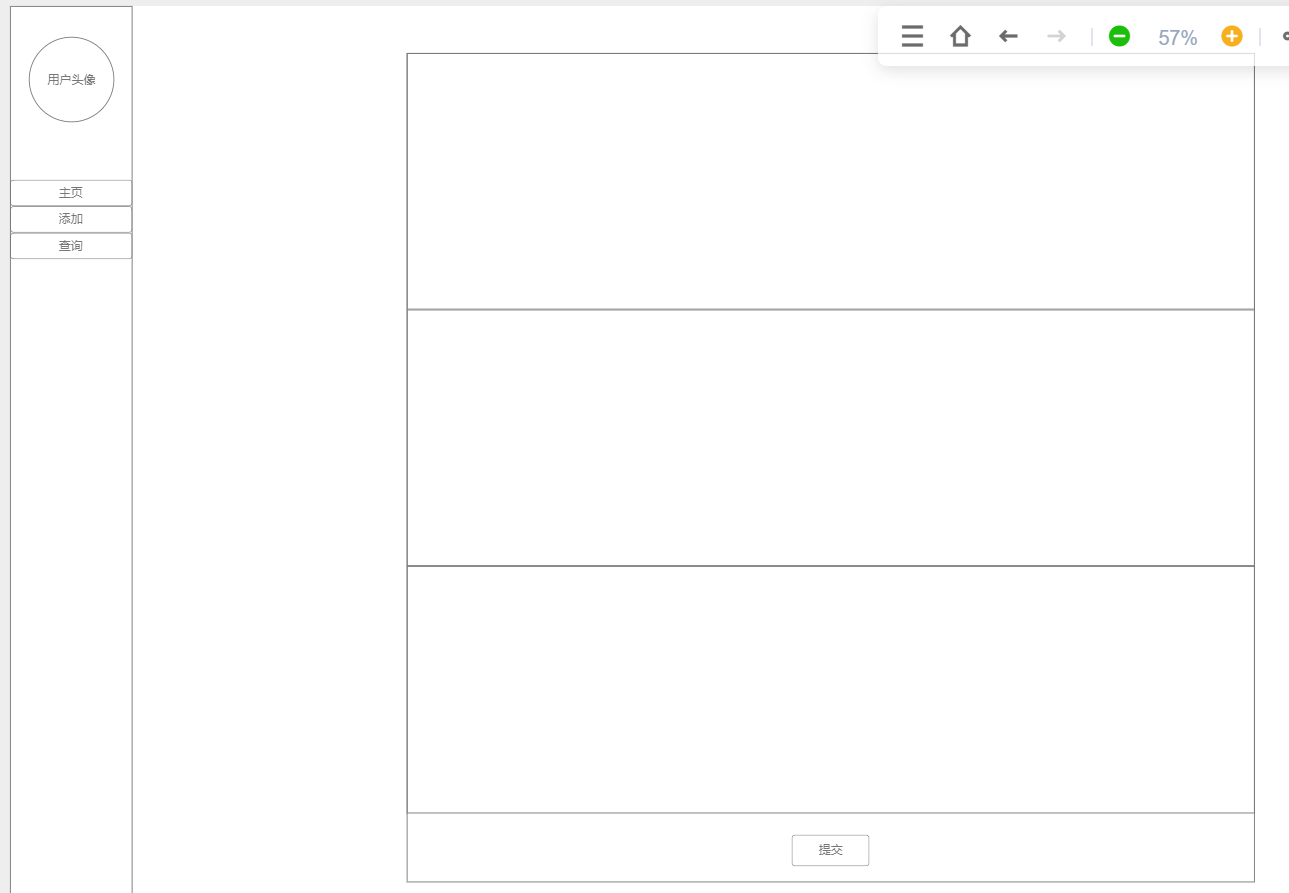
进入系统后，首先出现登录页面：



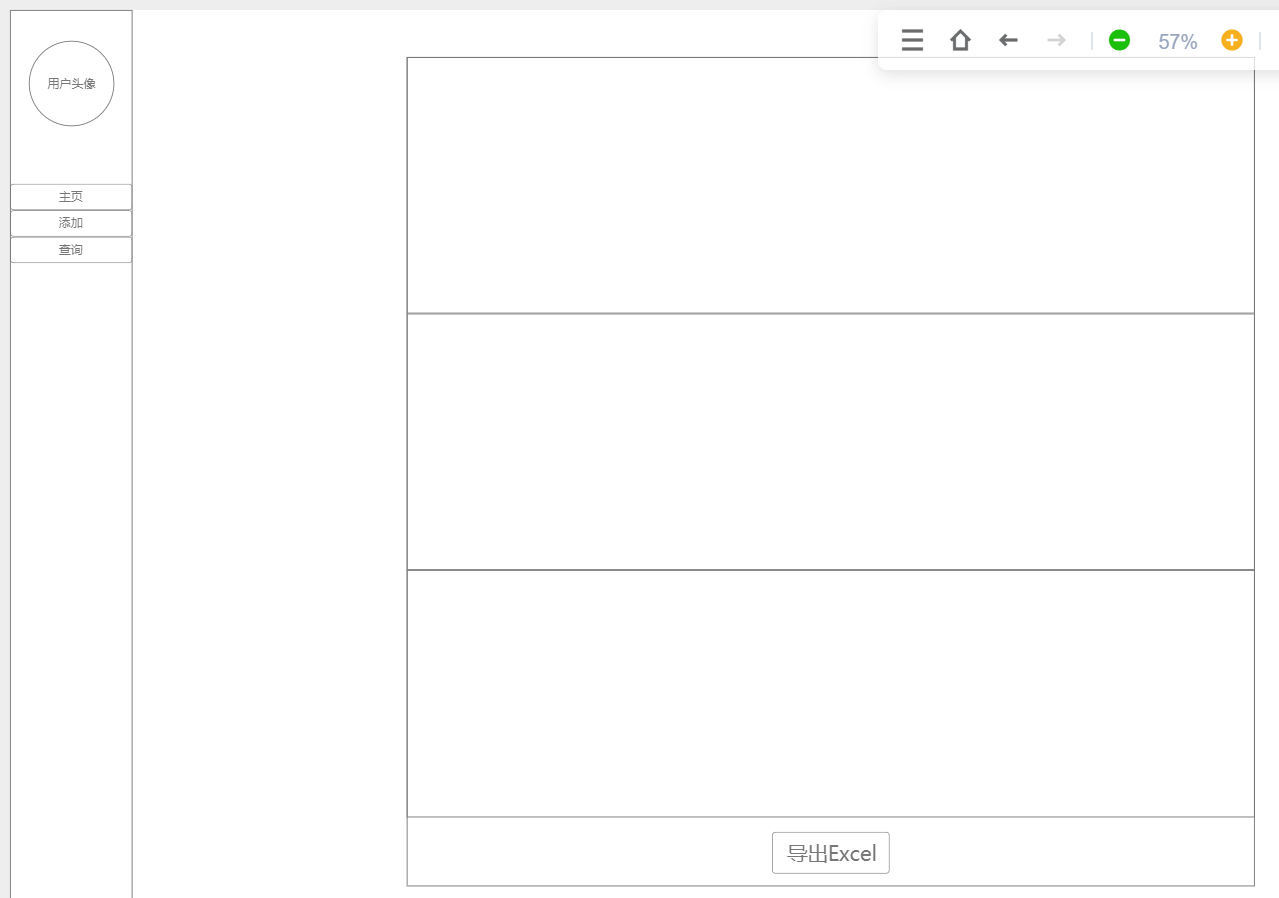
登陆后，跳转到主页面。可以看到主页面左侧功能栏中有三个选项。主页即为当前页面。



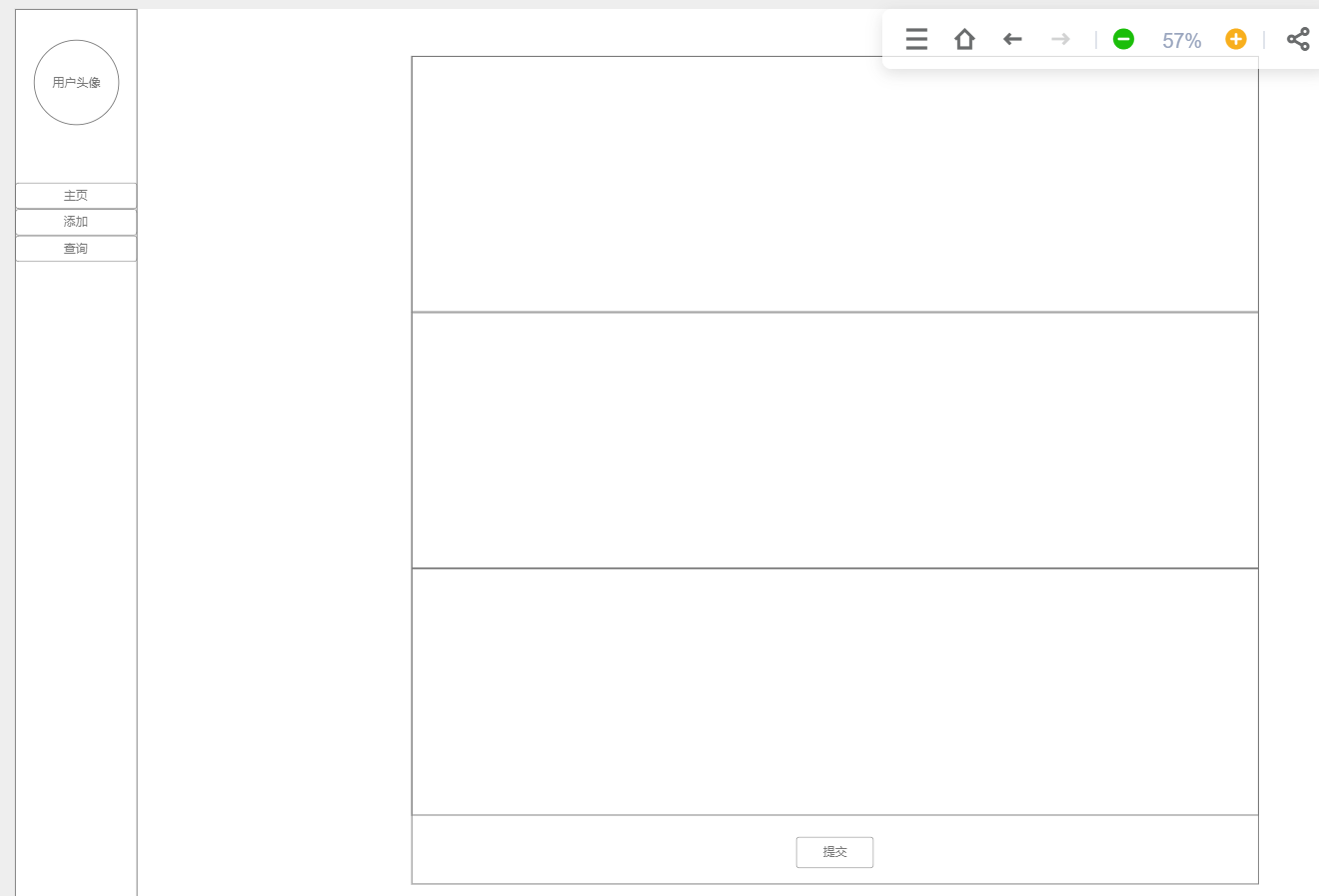
点击待审核论文的“修改”或“查看”，可以跳转到修改页面：



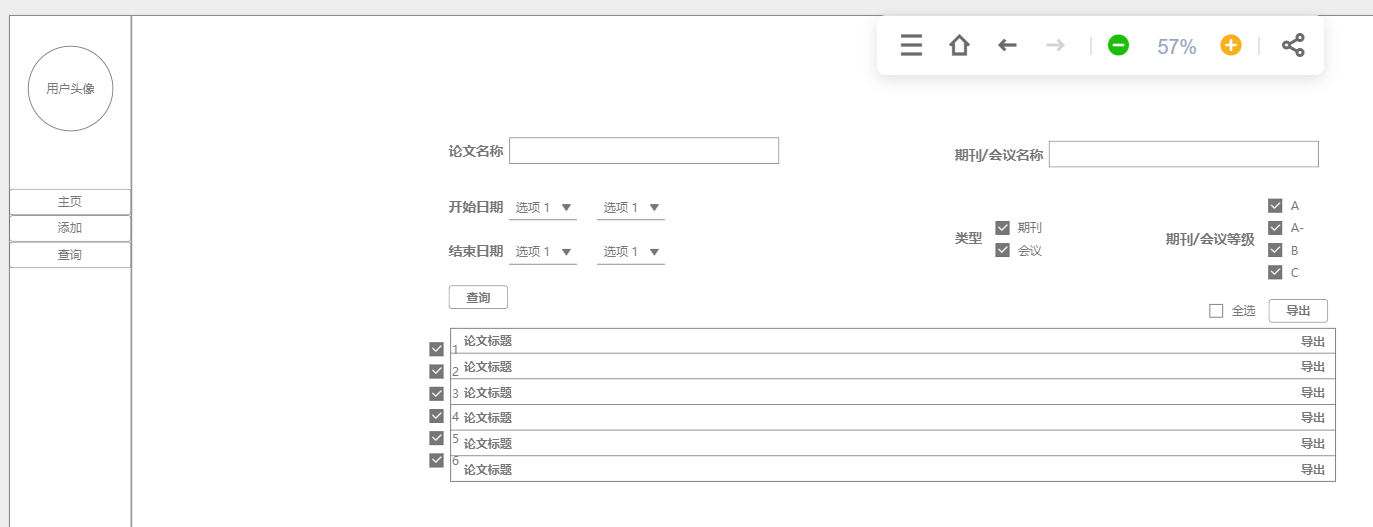
点击已审核论文的“查看”，可以跳转到查看页面，并可以导出论文信息到excel。



点击左侧的“添加”后，跳转到添加页面，在这里进行科研成果的添加。

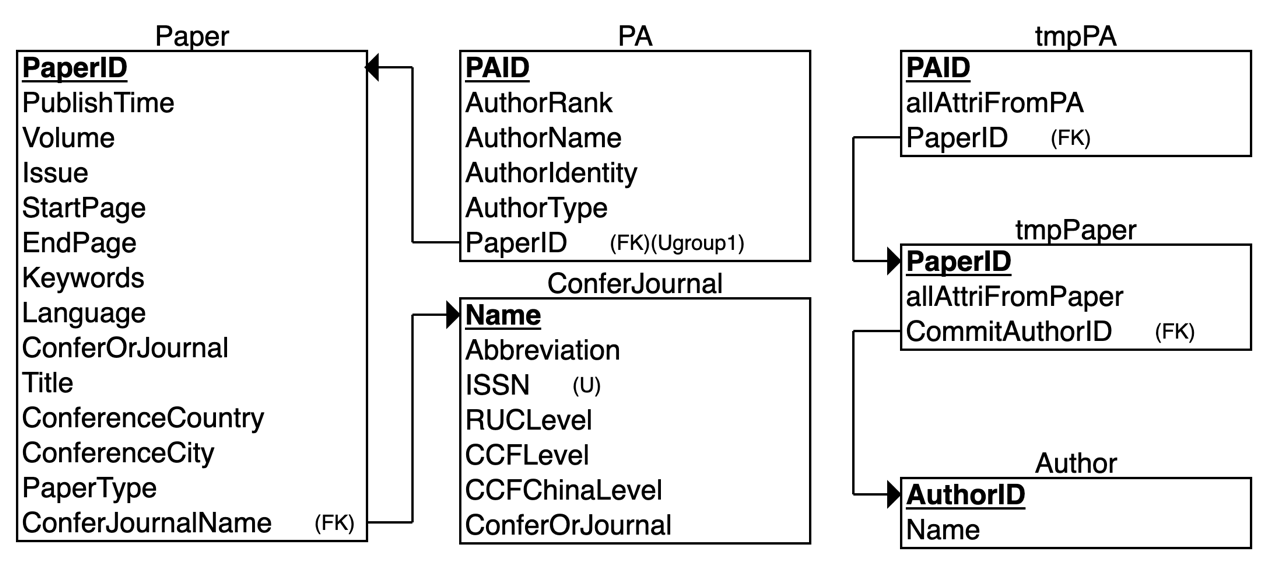


点击左侧的“查询”后，跳转到查询页面，在这里进行科研成果的查询与导出。



## 4. 数据字典

1. 三个表记录已审核的论文信息：
   1. Paper记录论文信息。
   2. Author记录作者信息。
   3. PA记录论文与作者的对应关系。
2. 两个表记录待审核的论文信息：
   1. tmpPaper记录论文信息和提交论文的用户ID。
   2. tmpPA记录论文与作者的对应关系。
3. 辅助表：
   1. ConferJournal记录期刊、会议的具体信息。
4. 关系模式图

****

1. 创建数据表的SQL语句
2. **create** **table** ConferJournal(
3. **Name** **varchar**(512) **primary** **key**,
4. ConferOrJournal **char**(1) not null,  -- C or J
5. Abbreviation **varchar**(50),
6. RUCLevel **char**(2) not null,
7. CCFLevel **char**(2) not null,
8. CCFChinaLevel **char**(2) not null,
9. ISSN **varchar**(50)
10. );
12. **create** **table** Author(
13. AuthorID **varchar**(12) **primary** **key**,
14. **Name** **varchar**(20) not null
15. );
17. **create** **table** Paper(
18. PaperID **bigint** **primary** **key** auto\_increment,
19. Title **varchar**(1024) not null,
20. ConferOrJournal **char**(1) not null,  -- C or J
21. ConferJournalName **varchar**(512) not null,
22. PublishTime **date** not null,
23. Volume **integer**,
24. Issue **integer**,
25. StartPage **integer** not null,
26. EndPage **integer** not null,
27. Keywords **varchar**(100),
28. ConferenceCountry **varchar**(50),
29. ConferenceCity **varchar**(50),
30. PaperType **varchar**(20) not null,
31. Language **char**(1), -- E or C
32. **foreign** **key**(ConferJournalName) **references** ConferJournal(**Name**)
33. );
35. **create** **table** PA(
36. PAID **bigint** **primary** **key** auto\_increment,
37. PaperID **bigint** not null,
38. AuthorName **varchar**(20) not null,
39. AuthorRank **integer** not null,            -- 第几作者
40. AuthorIdentity **varchar**(20) not null,    -- 通讯作者、普通作者
41. AuthorType **varchar**(20) not null,        -- 本院教师 本院学生 其他
42. **foreign** **key**(PaperID) **references** Paper(PaperID)
43. );
45. **create** **table** tmpPaper(
46. PaperID **bigint** **primary** **key** auto\_increment,
47. CommitAuthorID **varchar**(12) not null,
48. Title **varchar**(1024) not null,
49. ConferOrJournal **char**(1) not null,  -- C or J
50. ConferJournalName **varchar**(512) not null,
51. PublishTime **date** not null,
52. Volume **integer**,
53. Issue **integer**,
54. StartPage **integer** not null,
55. EndPage **integer** not null,
56. Keywords **varchar**(100),
57. ConferenceCountry **varchar**(50),
58. ConferenceCity **varchar**(50),
59. PaperType **varchar**(20) not null, -- 正刊 专刊 增刊 长文Oral 长文Poster 短文Oral 短文Poster Demo
60. Language **char**(1), -- E or C
61. **foreign** **key**(CommitAuthorID) **references** Author(AuthorID),
62. **foreign** **key**(ConferJournalName) **references** ConferJournal(**Name**)
63. );
65. **create** **table** tmpPA(
66. PAID **bigint** **primary** **key** auto\_increment,
67. PaperID **bigint** not null,
68. AuthorName **varchar**(20) not null,
69. AuthorRank **integer** not null,            -- 第几作者
70. AuthorIdentity **varchar**(20) not null,    -- 通讯作者、普通作者
71. AuthorType **varchar**(20) not null,        -- 本院教师 本院学生 其他
72. **foreign** **key**(PaperID) **references** tmpPaper(PaperID)
73. );

## 5 安全性需求

###### 潜在的安全性问题

1. 非授权用户对数据库的恶意存取和破坏。例如：非教师或审核者通过非法手段对论文信息或论文内容进行修改和破坏。

2. 数据库中重要或敏感的数据被泄露。例如：教师信息或论文内容的泄露和盗取。

3. 安全环境本身的脆弱性。例如：操作系统安全的脆弱，网络协议安全保障的不足。

###### 系统安全性要求

1. 防止未经过授权的人员访问数据，确保敏感信息没有被不“需要知道”的人员读取到。控制措施：对提出请求的数据库用户进行身份鉴别，防止不可信用户使用系统。

2. 防止未经过授权的人员删除和修改数据。控制措施：在SQL处理层进行自主存取控制和强制存取控制，进一步可以进行推理控制。

3. 监视对数据的访问和更改等使用情况。控制措施：对用户访问行为和系统关键操作进行审计，对异常用户行为进行简单入侵检测。

4. 对论文内容与信息进行数据加密，只有知道解密方法的人员（教师与审核者）才能获得并知道数据真实内容。

## 6. 运行环境需求

1. 硬件平台：可以联系学校的cpu服务器资源

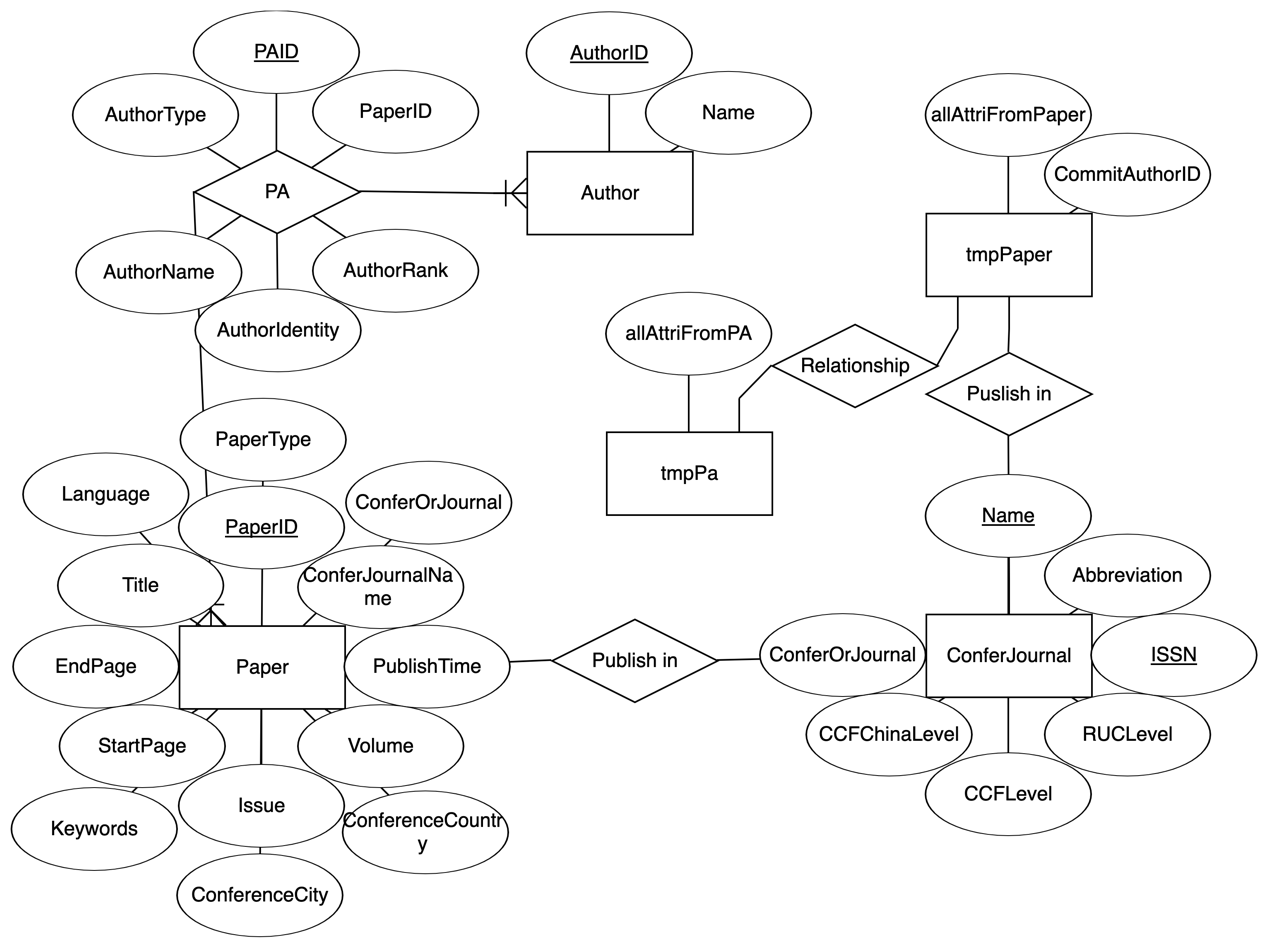
2. 操作系统：跨平台网站应用，bs模式，对用户的操作系统没有特殊要求

3. 客户端：网站

4. 部署方式：python后端+前端框架

# 三、概念结构设计

1．E-R图

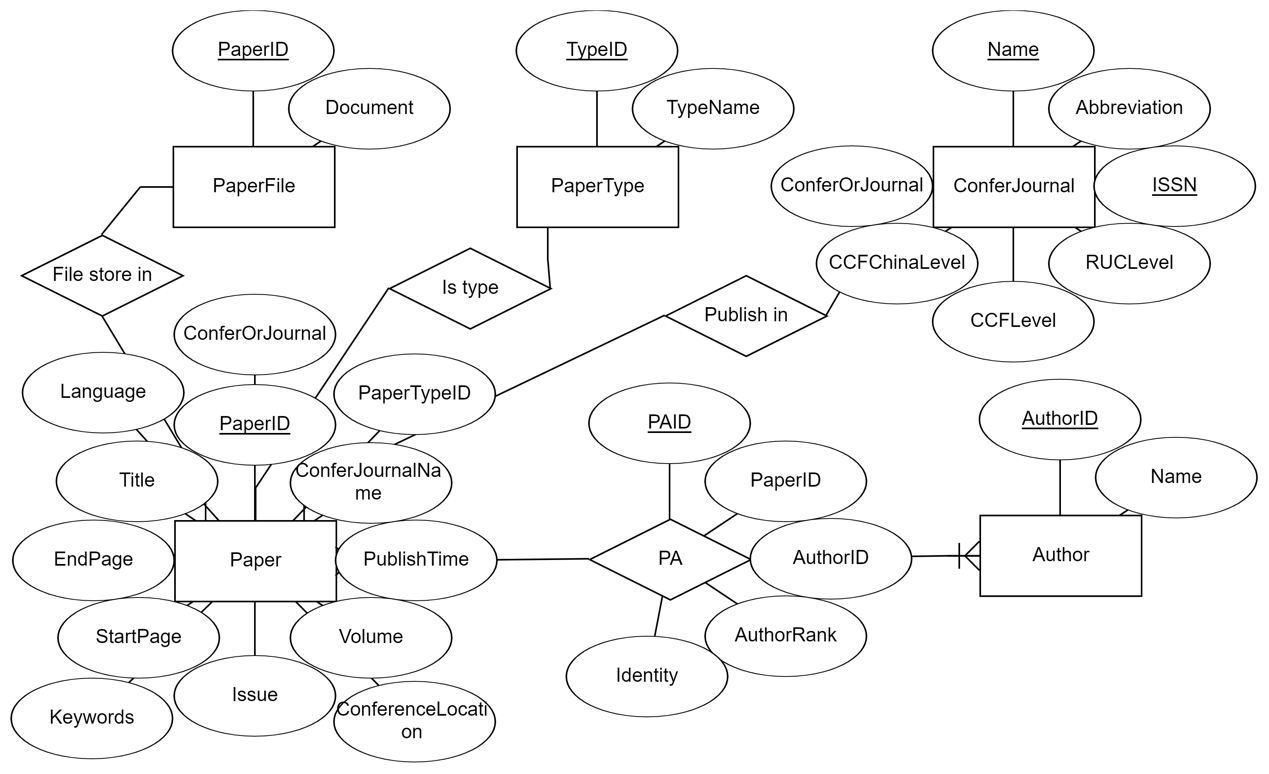


2．遇到的问题及解决方案

2.1 数据冗余与查询效率的权衡

由于期刊和会议论文的具体信息不同，例如期刊有ISSN号，会议有召开国家、城市，因此最初的概念结构设计是将期刊和会议论文分开记录的，具体设计见如下ER图。这种设计方式可以减少数据冗余，但增加了查询的复杂度。

在权衡数据冗余与查询效率后，我们将设计改为上一部分中展示的ER图。最终设计有一定的数据冗余，但查询效率显著高于最初版设计。



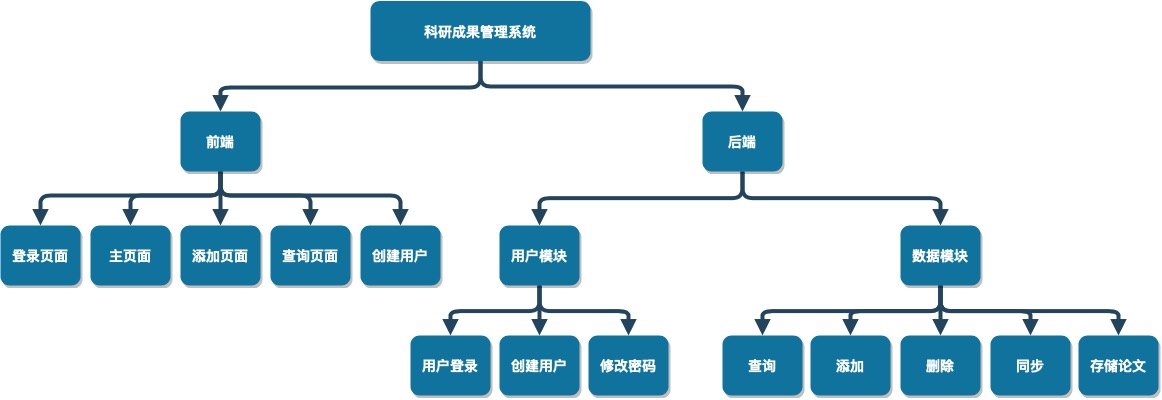
2.2 待审核论文与已审核论文的分区

本系统支持教师上传论文，管理员审核论文的功能，因此一些论文是已经被管理员审核过的论文，另一些是属性未知的待审核论文。已审核的论文几乎不再进行修改，待审核论文通常需要反复修改、删除或是重新上传。

我们将待审核论文和已审核论文进行分区存储。已审核论文的信息存储在Paper表和PA表中，待审核论文的信息存储在tmpPaper和tmpPA表中。这样数据库正式表Paper和PA中的论文信息是管理员审核过的正确信息，临时表是较为混杂的信息。将二者分区可以保证数据库中数据的正确性，并且减少正式表中数据的频繁修改。

# 四、系统设计

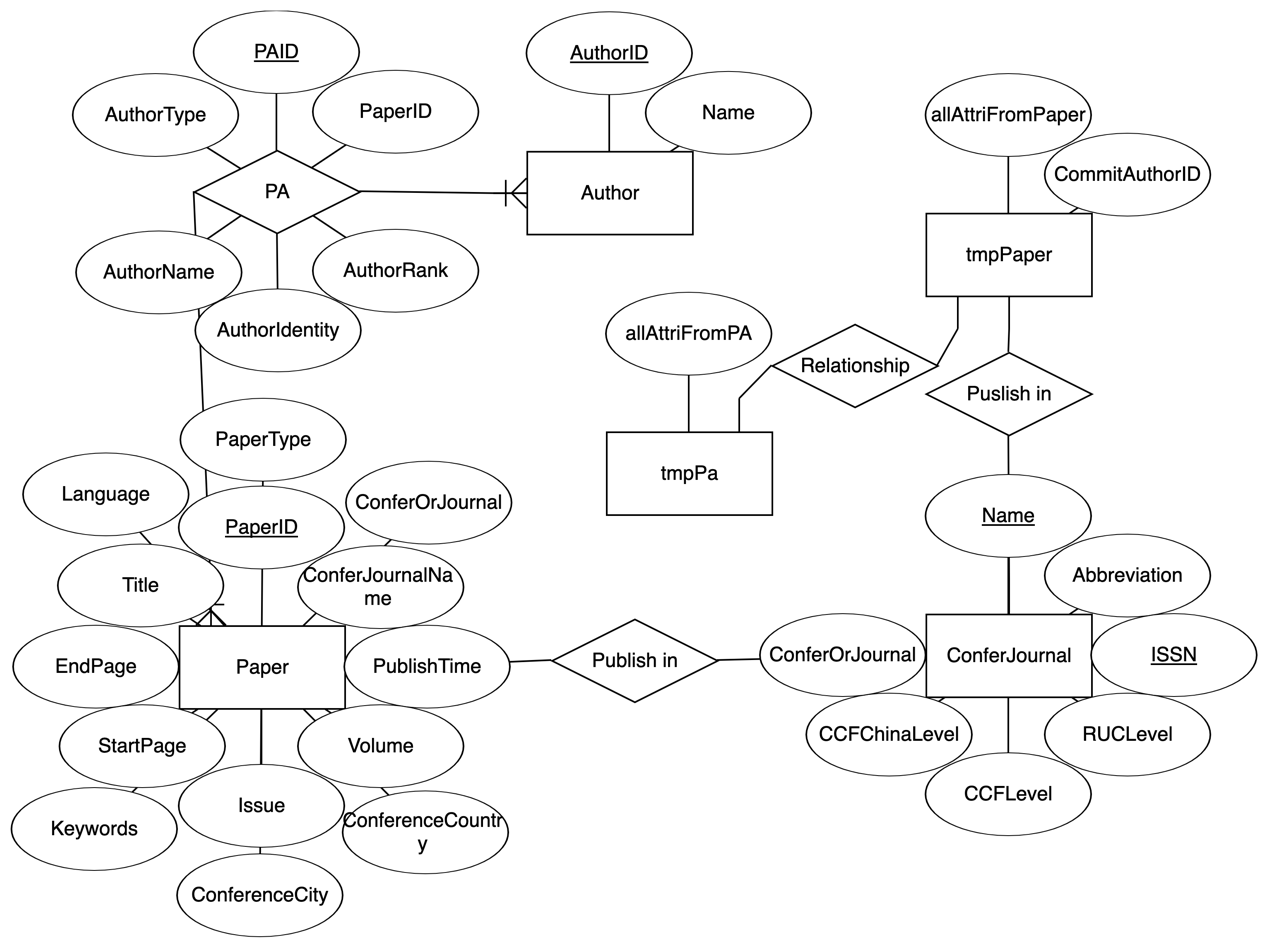
1．总体设计

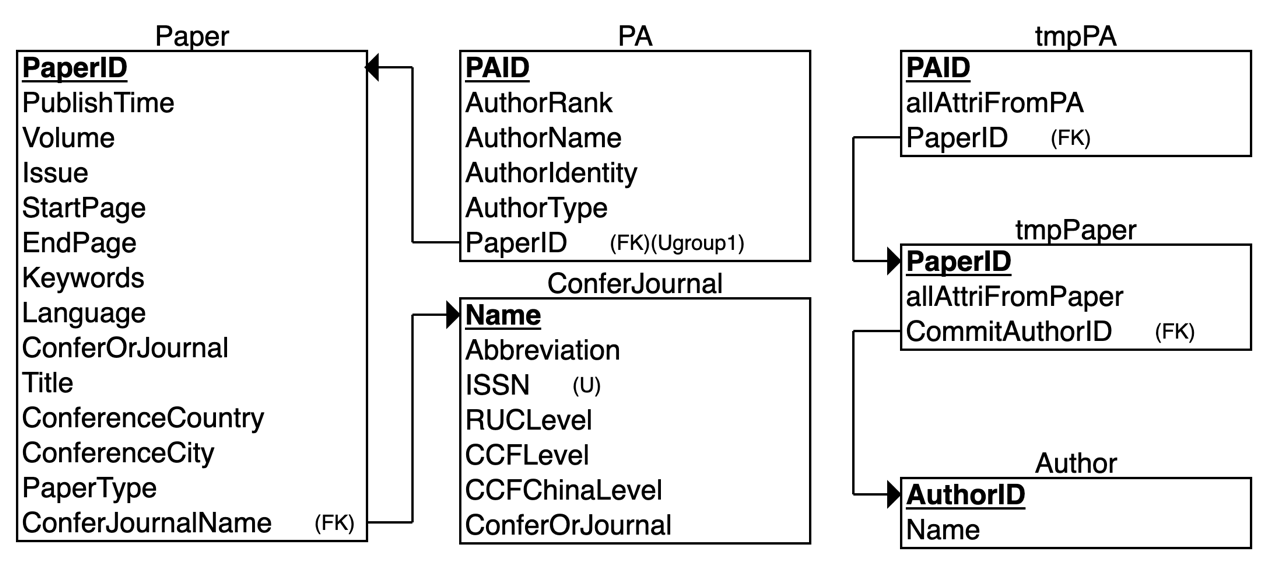


|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | 子模块 | 功能名称 | 具体描述 |
| 前端 | 登录界面 | 用户登录 | 用户输入账号密码登录 |
| “关于本系统”下拉菜单 | 显示系统信息 |
| “友情链接”下拉菜单 | 提供人大官网、信院官网、微人大等链接 |
| 主页面 | 修改/审核 | 普通用户可修改未审核的项目，管理员可审核入库项目 |
| 各功能跳转链接 | 用户可访问不同功能页面 |
| 添加页面 | 删除 | 管理员可以删除数据库中的论文信息 |
| 查询导出页面 | 查询并导出结果 | 用户可在此处按各种条件筛选已入库的项目信息，并导出Excel |
| 详情页面 | 查看项目详情 | 显示项目的各种信息，可导出为Excel |
| （管理员）创建新用户 | 创建用户 | 管理员可在此处创建新用户 |
| 后端 | 用户模块 | 用户登录 | 普通用户(教师)及管理员输入唯一的用户名、设定的密码和正确的验证码登录 |
| 创建新用户 | 管理员有权限创建新的用户,为其分配用户名和密码 |
| 修改密码 | 普通用户(教师)可以修改自己的登录密码 |
| 数据模块 | 查询 | 用户输入需求,查询数据库中的论文、作者信息 |
| 添加 | 用户可以向数据库中添加论文信息 |
| 删除 | 管理员可以删除数据库中的论文信息 |
| 同步 | 管理数据库中数据表之间的同步添加、删除 |
| 论文存储 | 存储论文pdf |

2．数据结构

* 1. E-R图及关系模式图





图中有六个实体：

* 1. Paper记录论文信息。
  2. Author记录作者信息。若作者是本校师生，则ID为职工号或学号；否则ID为负数。
  3. PA记录论文与作者的对应关系。
  4. ConferJournal记录期刊、会议的具体信息。
  5. tmpPaper记录待审核论文信息和提交论文的用户ID。
  6. tmpPA记录待审核论文与作者的对应关系。

两个联系：

1. Publish in：联系Paper、tmpPaper和ConferJournal，通过Paper.ConferJournalName和ConferJournal.Name的自然连接。
2. PA：联系Paper和Author，通过PA.PaperTitle = Paper.Title, PA.AuthorID = Author.AuthorID连接。同时PA.AuthorRank记录该作者是第几作者，PA.Identity记录该作者的特殊身份，包括独立作者、第一作者和通讯作者。
   1. 分析关系模式的范式类型

* 表Paper的主码是Title。其中的函数依赖有
  + 所有属性函数依赖于Title
  + (PublishTime, ConferJournalName, ConferOrJournal) -> (ConferenceCountry, ConferenceCity)。当会议或期刊类型确定、会议名称确定、会议召开时间确定，则会议召开地址确定。
* 表PaperType、ConferJournal、Author中的函数依赖只有所有属性对码的函数依赖。

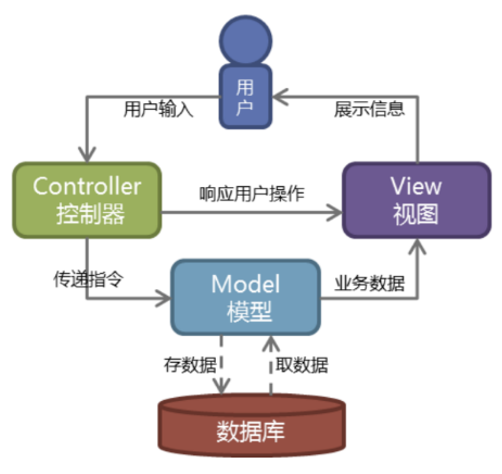
该关系模式中不存在部分函数依赖于码的函数依赖，因此该关系模式属于2NF。因为表Paper中有传递函数依赖Title -> (PublishTime, ConferJournalName, ConferOrJournal), (PublishTime, ConferJournalName, ConferOrJournal) -> ConferenceLocation，所以不属于3NF。因此该关系模式属于2NF。

3. 技术选型

我们拟选择Django作为后端框架；

总体来说, Django是一个基于python的重量级web框架，功能齐全，文档丰富, 为搭建后端提供一站式解决思路。我们从三个方面阐述其优越性:

首先, 在结构方面, Django本身基于MVC模型实现了MTV, 即Model、View和Controller, 三者相互沟通, 并把业务逻辑、模型数据、用户界面分离开来，让开发者将数据与表现解耦；



在功能方面, Django为后端开发提供健全强劲的支持:

1. 自带强大的数据库功能, 仅需一行语句就可以连接外部mysql数据库, 并提供多种数据访问方式；并且提供如用户模型等多种内置的数据模型, 并且提供了方便的api对其进行管理, 大幅减少了工作量。
2. 自带完善的后台功能, 可以访问默认的admin页, 可以管理数据库各项数据以及用户、用户组的各项权限, 对整个后端框架实现完全控制。
3. 支持自定义的优雅的网址, 得益于其MVT开发模式, 程序员不需要用网址向后端传输php语句, 因此网址相较于其他框架更加简洁优美。
4. 具有强大的html模板系统, 程序员可以在html模板中直接访问传入的参数值, 实现前端开发和后端数据的解耦； 同时html模板之间可以继承, 使得重复的工作大幅减少, 支持前端开发的重复利用；
5. 具有完善的缓存系统, 和memcached、redis或者其他的缓存系统联用，提高了页面的加载速度, 让用户的体验度大幅提升；
6. 支持ORM和模板引擎，可以使用jinja等第三方库模板引擎自带的ORM和关系型。

在学习方面, Django为各个模块功能都提供了详细、易于理解的文档介绍； 并且Django拥有活跃的用户社区, 我们在学习的过程中可以广泛地交流讨论。

综上, Django在结构、功能、学习方面都拥有显著的优点, 且符合我们大作业的各项需求, 因此我们决定采用Django作为后端框架。

4. 物理设计

4.1 关系模式存取方法选择

我们在Paper表的主码PaperID上建立了索引。在本系统中，进入论文详情页、修改论文信息、审核论文、导出论文信息都需要搜索PaperID，建立B+树索引可以显著提高搜索效率，从而提升系统性能。

存储论文电子版文档时，本系统将该文件命名为数据库Paper表中的的PaperID，而不是论文标题。在这样的存储方式下，搜索电子版文档的时间从字符串匹配时间改进为了整数比较时间，提高了系统效率，同时不影响任何功能。

4.2 数据库存储结构的确定

在这一部分，我们介绍数据库中数据的存放位置和存储结构。

为了确定数据的存放位置和存储结构，我们需要考虑存取时间、存储空间利用率和维护代价三个方面。这三方面有时相互矛盾，需要权衡后选择折中方案。我们将数据库中的数据分为以下四个部分。

1. 易变部分：

* tmpPaper表和tmpPa表。这两个表存储待审核论文的具体信息和作者信息。由于老师或管理员可能会修改这一部分信息，因此为易变部分。

1. 稳定部分：

* Paper表和PA表。这两个表存储已审核论文的信息，很少再发生改变，因此是稳定的数据。
* ConferJournal表，由于期刊和会议的信息几乎不会发生改变，因此这一部分数据是稳定部分。
* 论文的电子版文档。这个文档几乎不会修改，是稳定的

1. 经常存取部分

* Author表。这个表存储作者的具体信息。添加论文时，若论文作者中包含非本校作者，则会向这个表中添加数据项。由于添加论文在本系统中是频繁操作，所以Author表是易变的。
* tmpPaper表和tmpPa表。老师录入论文时、管理员审核论文、修改论文信息时，都会存取这两个表。
* ConferJournal表的存储频率低，取用频率较高。由于期刊和会议很少新增，所以存储频率低。每次查询论文都需要查找ConferJournal表，因此取用频率高。这个表几乎是只读不写的。

1. 存取频率较低部分

* 论文的电子版文档。这个文档几乎只会存储一次，而且较少读取。

针对以上四部分数据，我们的存储原则是：

1. 将易变部分存储在带宽较大、磨损损失较小的存储设备上。且数据间留有较大的缝隙，以供后续添加、修改数据时，减小数据搬移，从而减小易变数据改变时的额外开销，尽管这会浪费一部分存储空间。
2. 将稳定部分紧密地存储。因为这部分数据很少发生改变，因此紧密地存储不会带来很多的数据修改时的搬移开销，同时节省了存储空间。
3. 将经常存取部分存储在带宽较大、访存速度较快的设备中，必要时可以固定在内存中。这样减少了数据存取的数据传输时间，提升系统性能。存储在内存中，更是有效减少了磁盘访问，极大提高系统性能。
4. 将存取频率较低部分存储在带宽较小、访存速度较慢的设备中。将更多的带宽大、访存速度快的存储设备留给经常存取的数据使用。提高系统的整体效率。

4.3 初始数据的载入

在初始化系统阶段，我们载入了以下数据：

1. 期刊、会议的具体信息写入ConferJournal表
2. 本院教师、学生的姓名、学号写入Author表

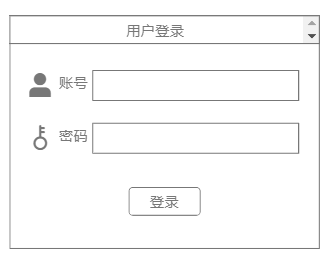
5. 界面设计（可选）

主要采用了Bootstrap这一基于HTML、CSS、JavaScript开发的前端开发框架，使用HTML技术，编程实现应用程序主界面框架。

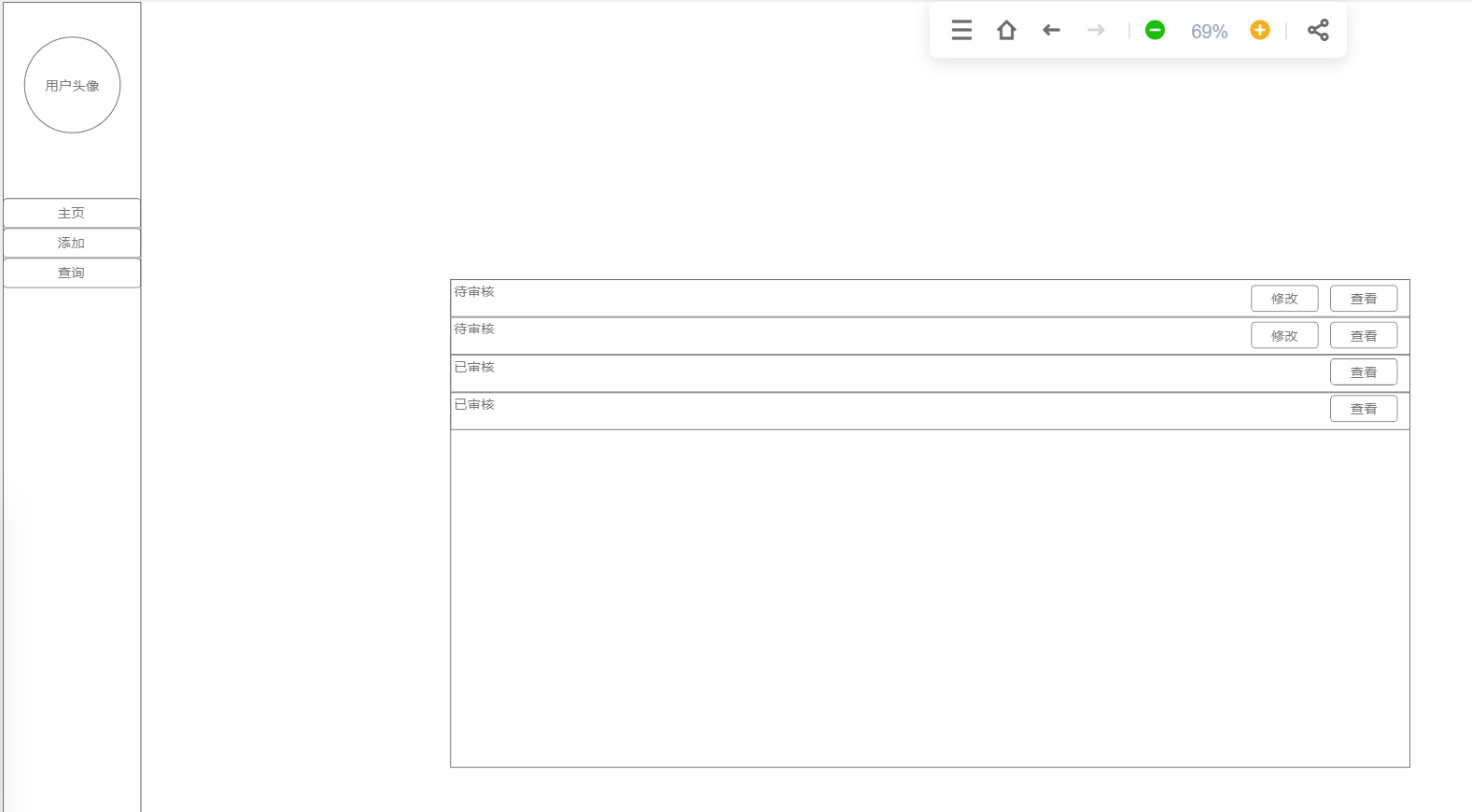
本系统主要的用户界面分别为登录页面，主页面，查询页面，论文添加页面，详情页面（基于论文添加页面，结构大致相同）

这里通过“摹客”设计并展示各页面的大致布局：

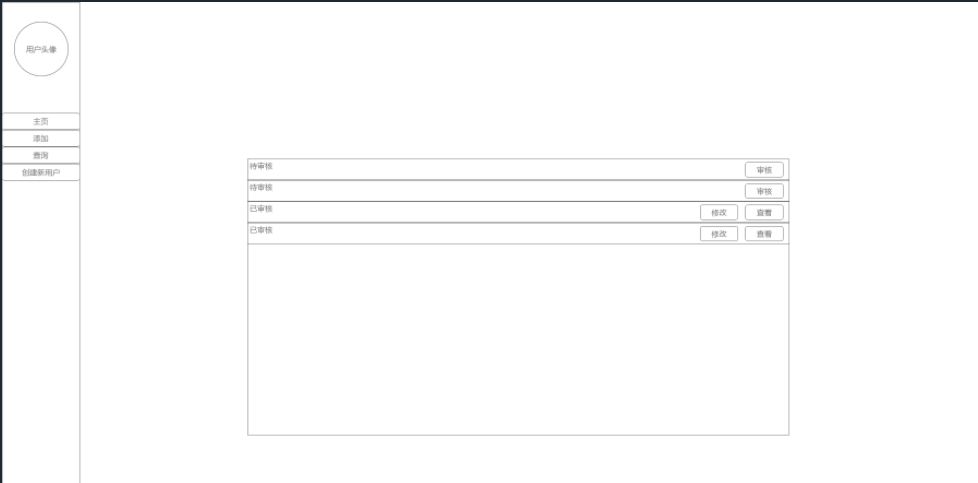
1. 登录页面



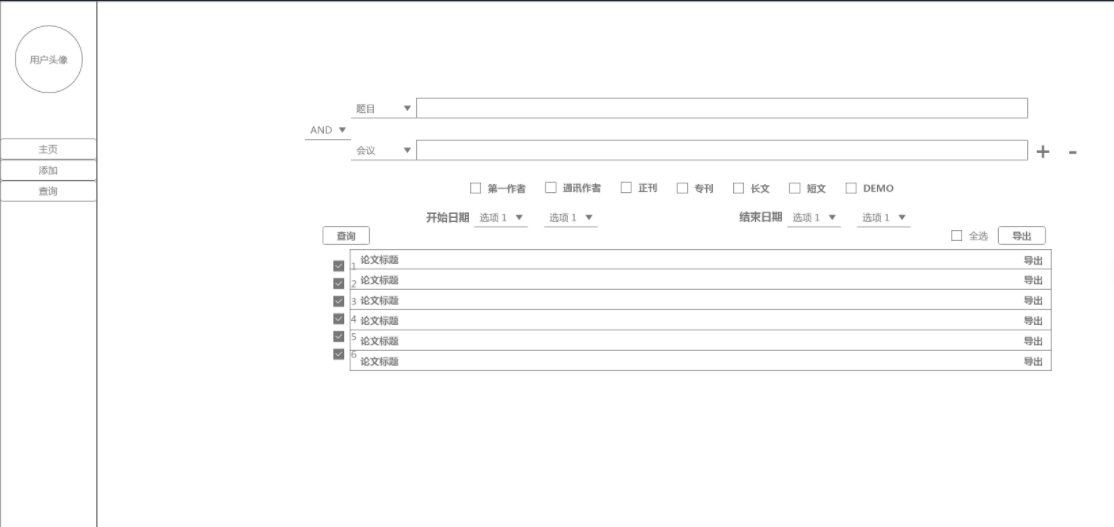
1. 主页面（普通用户）



主页面（管理员）



1. 查询页面（普通用户）



查询页面（管理员）



1. 添加页面



# 五、系统测试

1．系统概述

下面对系统中的核心功能或模块，逐一进行测试。

1）模块：登录界面

测试名称：登录界面外观与功能测试。

测试目的：验证登录界面外观和功能是否正常。

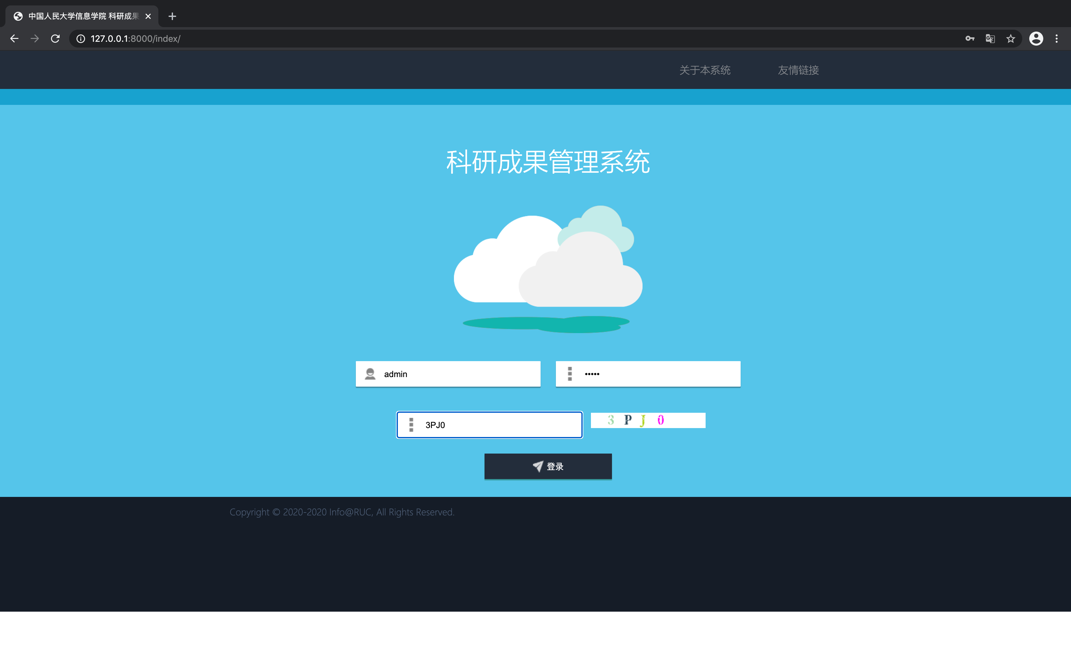
测试：

输入：在index页面输入用户名、密码、验证码，并点击“登录”。

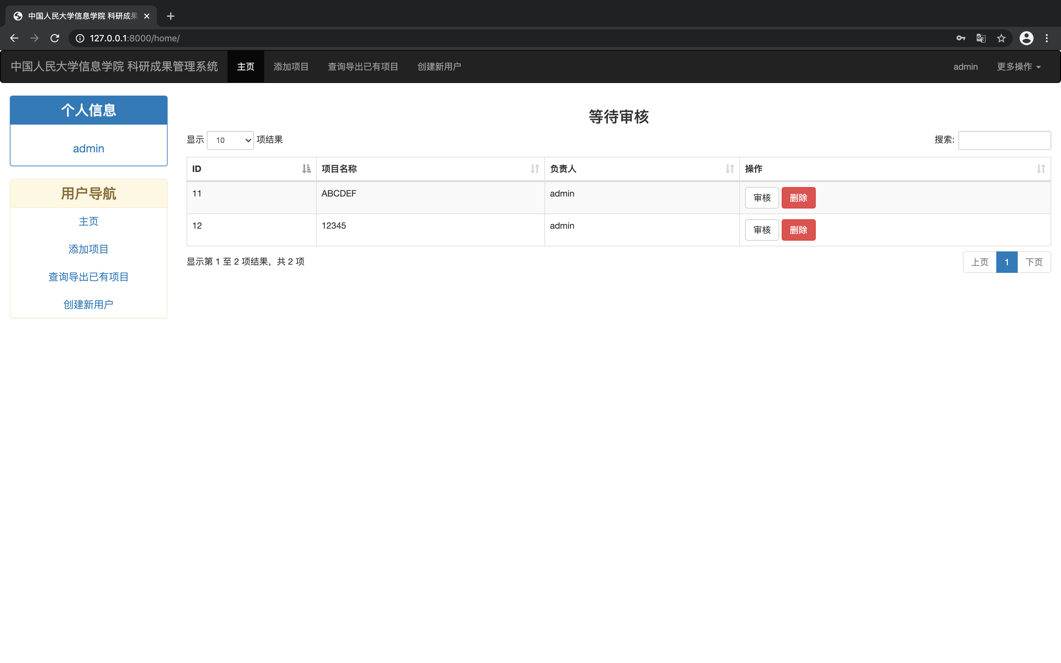
期待输出：登录成功，跳转到主页。

实际输出：登录成功，跳转到主页。

测试截图:



(在登录页面输入信息)



（成功登录至主页）

2）模块：主页面

测试名称：主页面外观与功能测试。

测试目的：验证主页面外观和功能是否正常。

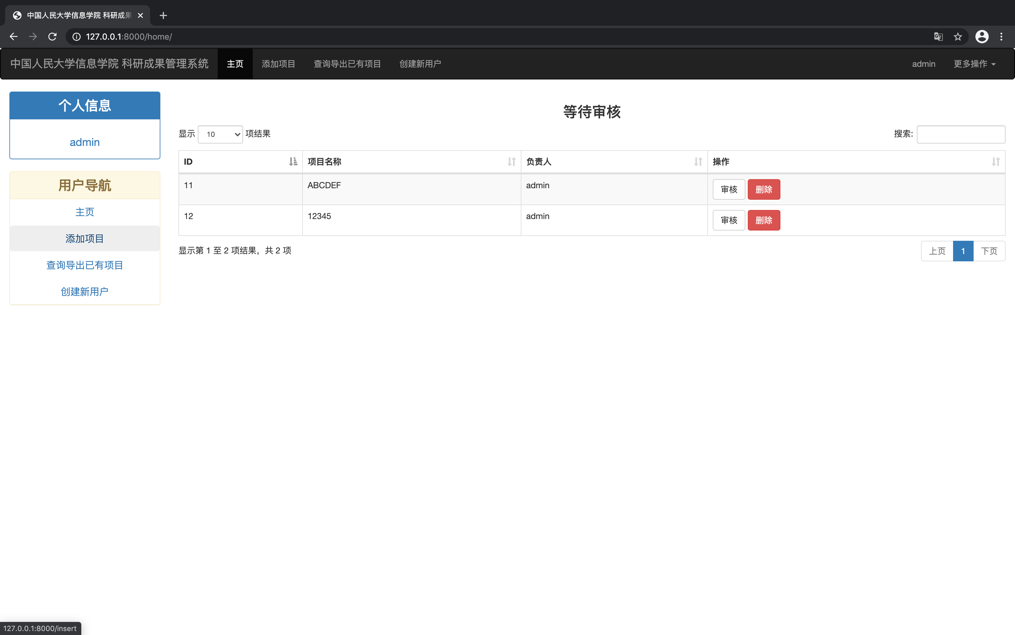
测试：

输入：在home页面点击各种链接。

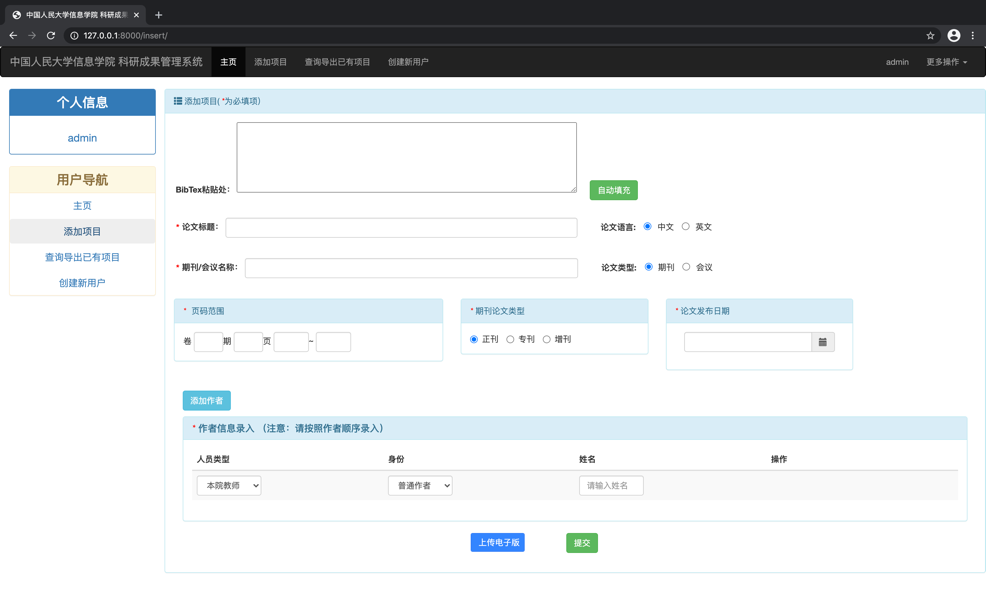
期待输出：正常跳转。

实际输出：正常跳转到各界面。

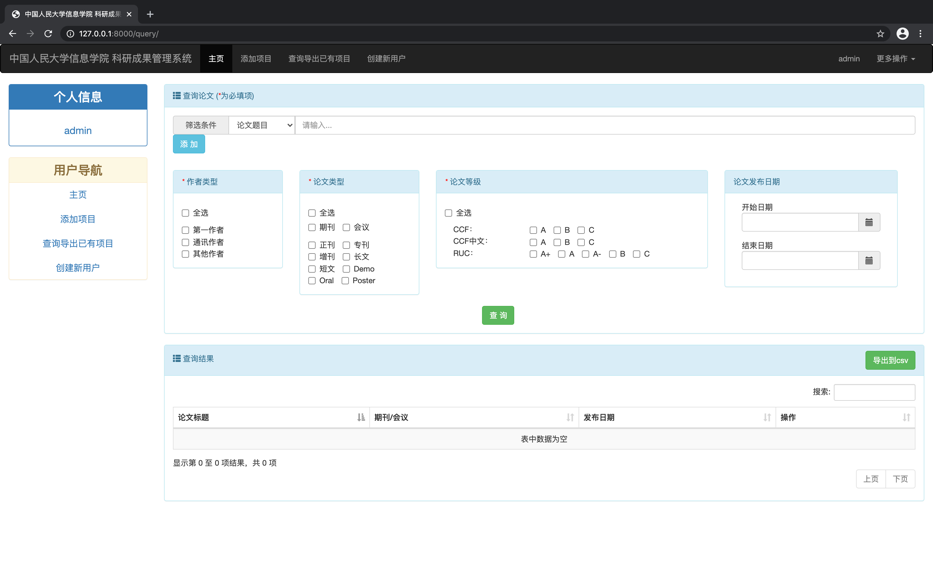
测试截图:



（在主页）



（跳转至添加页面）



（跳转至查询页面）

3）模块：插入页面

测试名称：插入页面外观与功能测试。

测试目的：验证插入页面外观和功能是否正常。

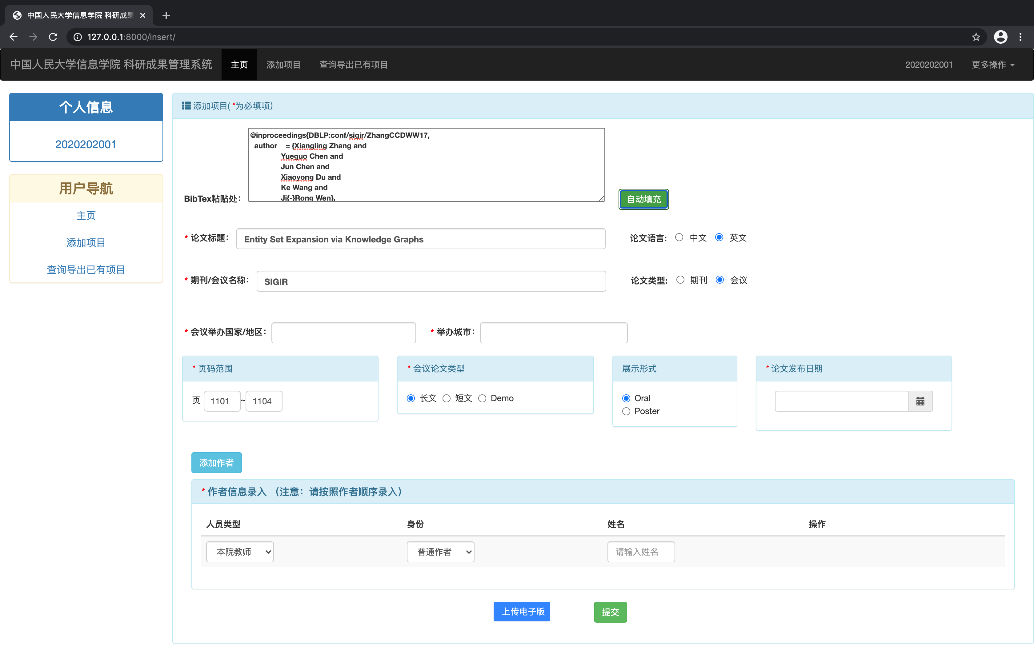
测试：

输入一：在insert页面粘贴bibtex。

期待输出：bibtex内容自动填充到论文信息的相应位置。

实际输出：bibtex内容自动填充正确。

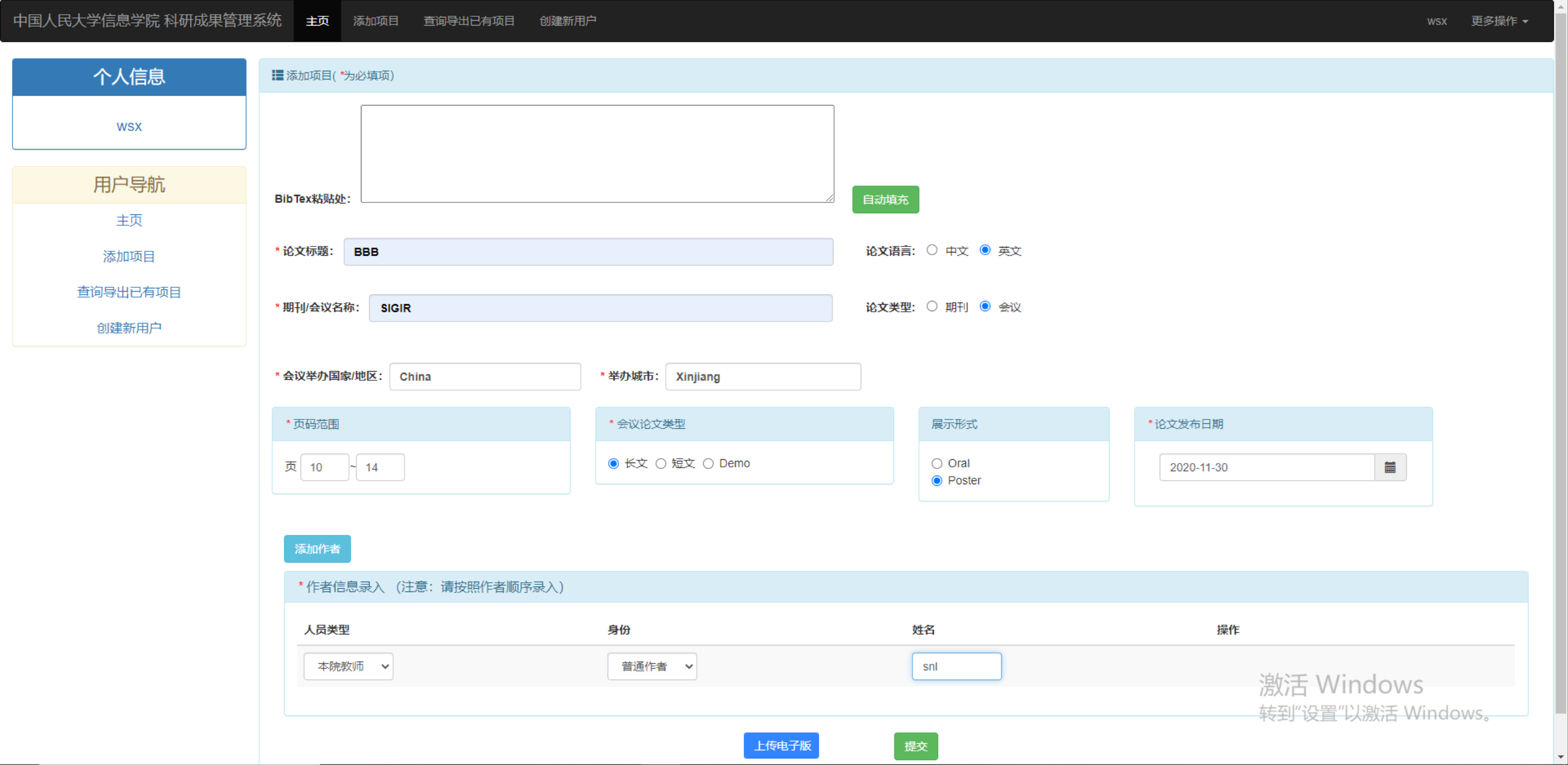
测试截图:



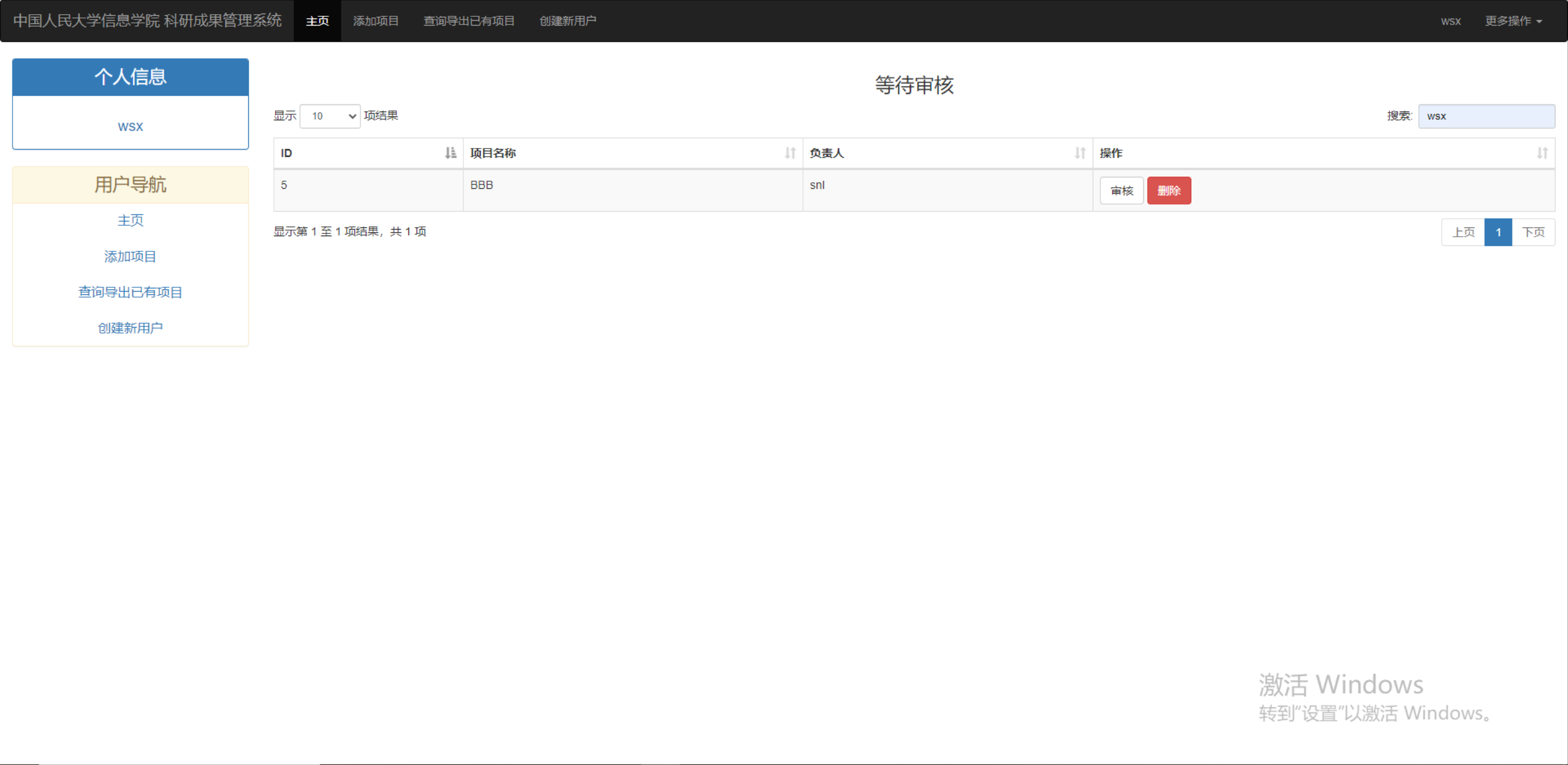
输入二：在insert页面完善内容相关信息，并上传电子档后提交。

期待输出：论文信息及电子档入临时表，进入待审核队列。

实际输出：论文信息及电子档入临时表，进入待审核队列。



（论文信息填写）



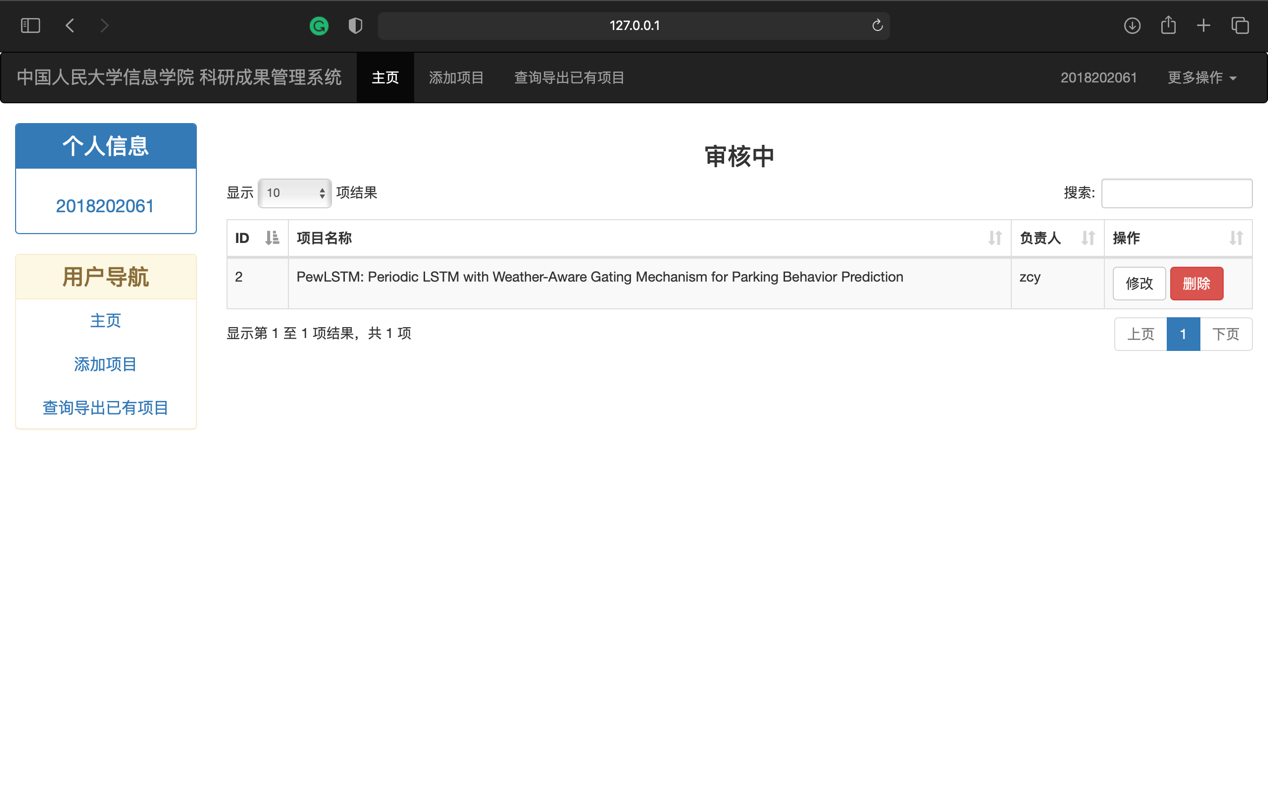
（进入待审核）

4）模块：修改页面

访问权限：数据库中的普通用户，即在本系统中拥有账号的所有本院教师。

功能简介：教师可以通过此页面修改审核中的论文的所有信息。

具体使用：首先，普通用户在首页中可以看到所有的审核中论文，点击右侧操作列的“修改”按钮，进入修改页面。



进入修改页后，页面会自动加载论文信息并显示。该页面中的所有信息都可以直接修改，完成修改页的功能。为了让老师核实上传的电子版论文，我们还在网页下部额外加入了“下载电子版”按钮，可以下载已经上传的论文电子版文件。

5）模块：审核页面(管理员)

测试名称：审核普通用户提交记录外观与功能测试

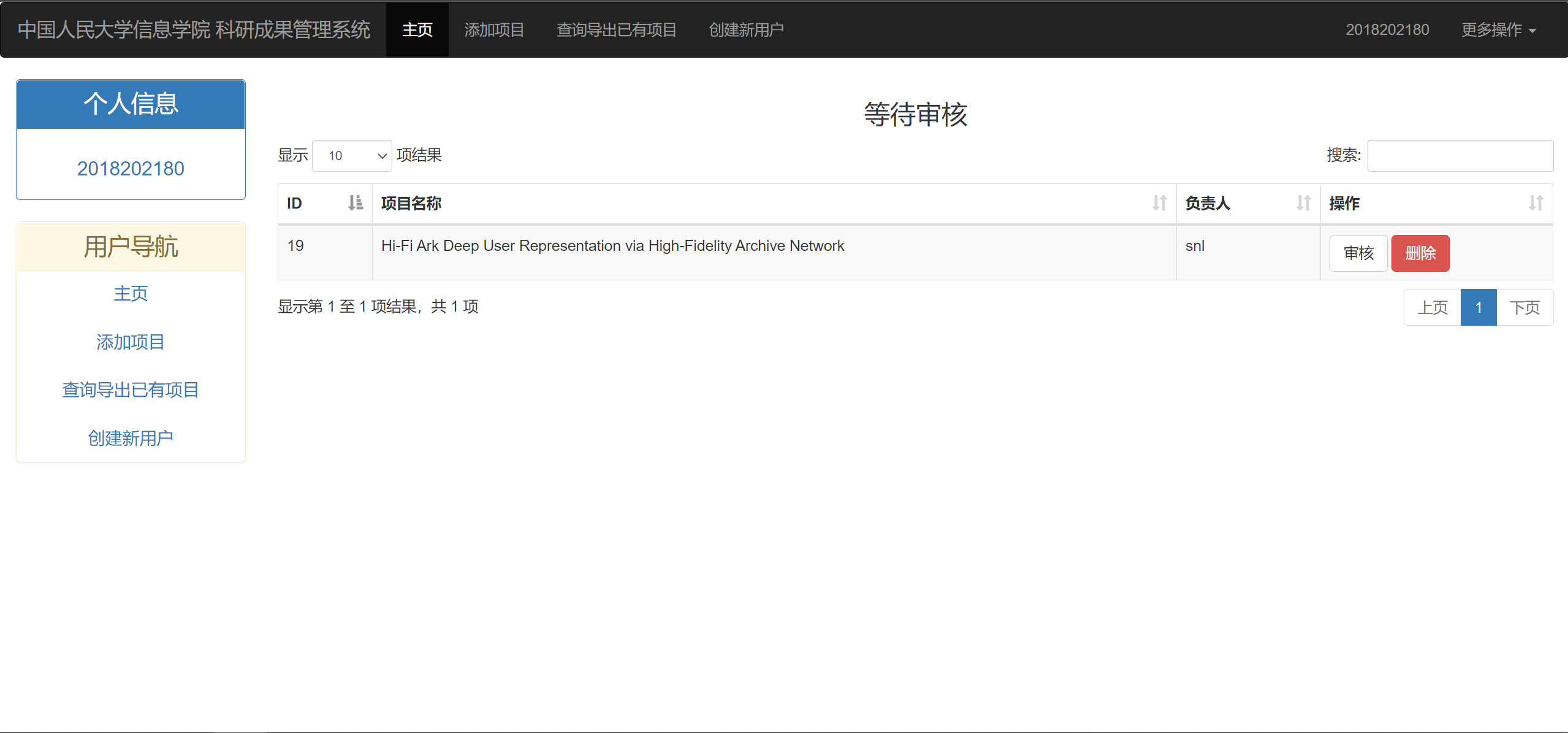
测试目的：验证管理员审核功能是否正常

输入: 普通用户(id为2018202175,姓名为snl)添加一条论文记录《Hi-Fi Ark Deep User Representation via High-Fidelity Archive Network》, 其第一作者为snl, 通讯作者为zpt；

(a)进入主页

预计输出: 显示名为snl的用户的对应提交记录, 显示审核,删除按钮；

实际输出: 正常显示对应提交记录, 按钮显示正常；

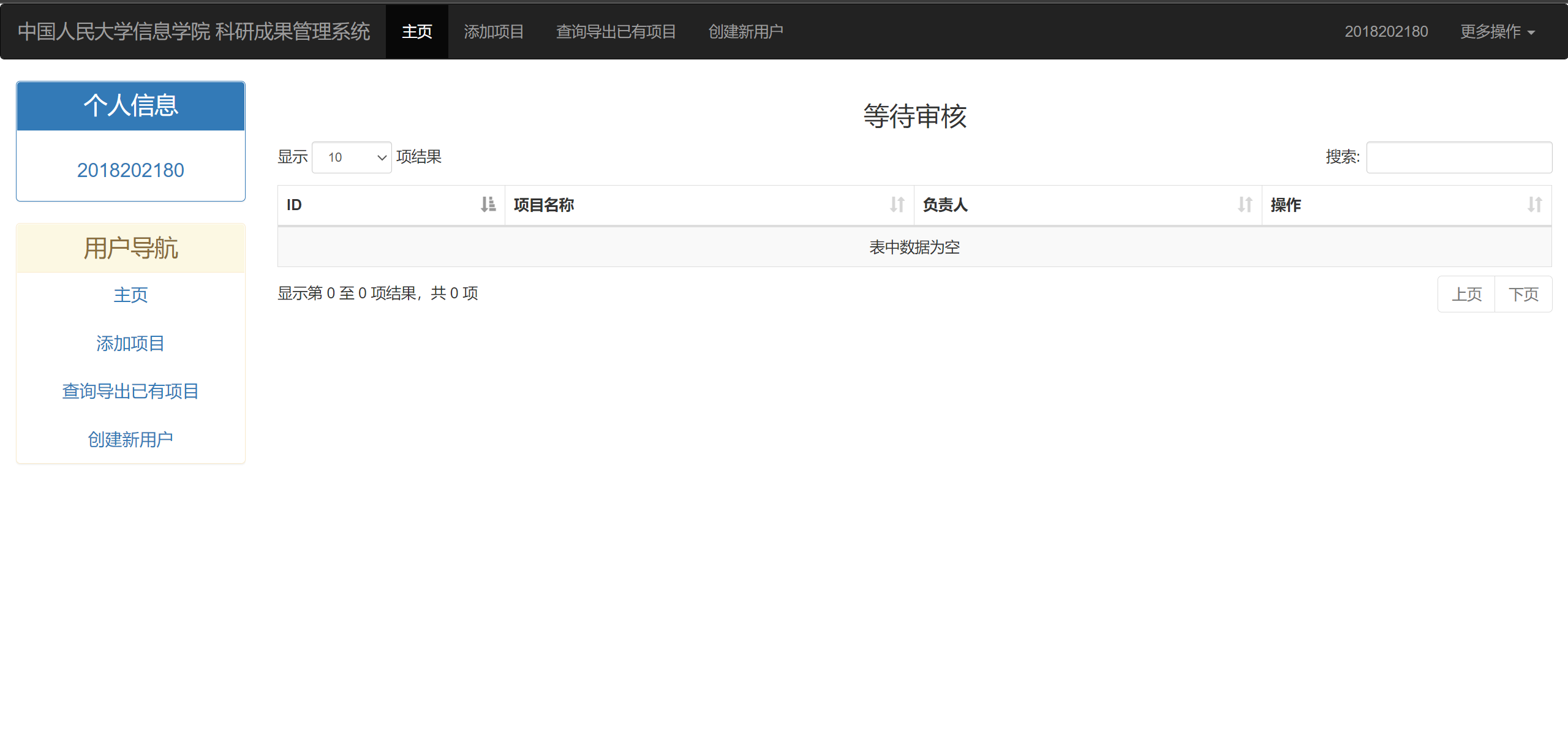


(b)删除

输入: 点击删除按钮；

预计输出: 成功删除对应的提交记录；

实际输出: 成功删除对应的提交记录；

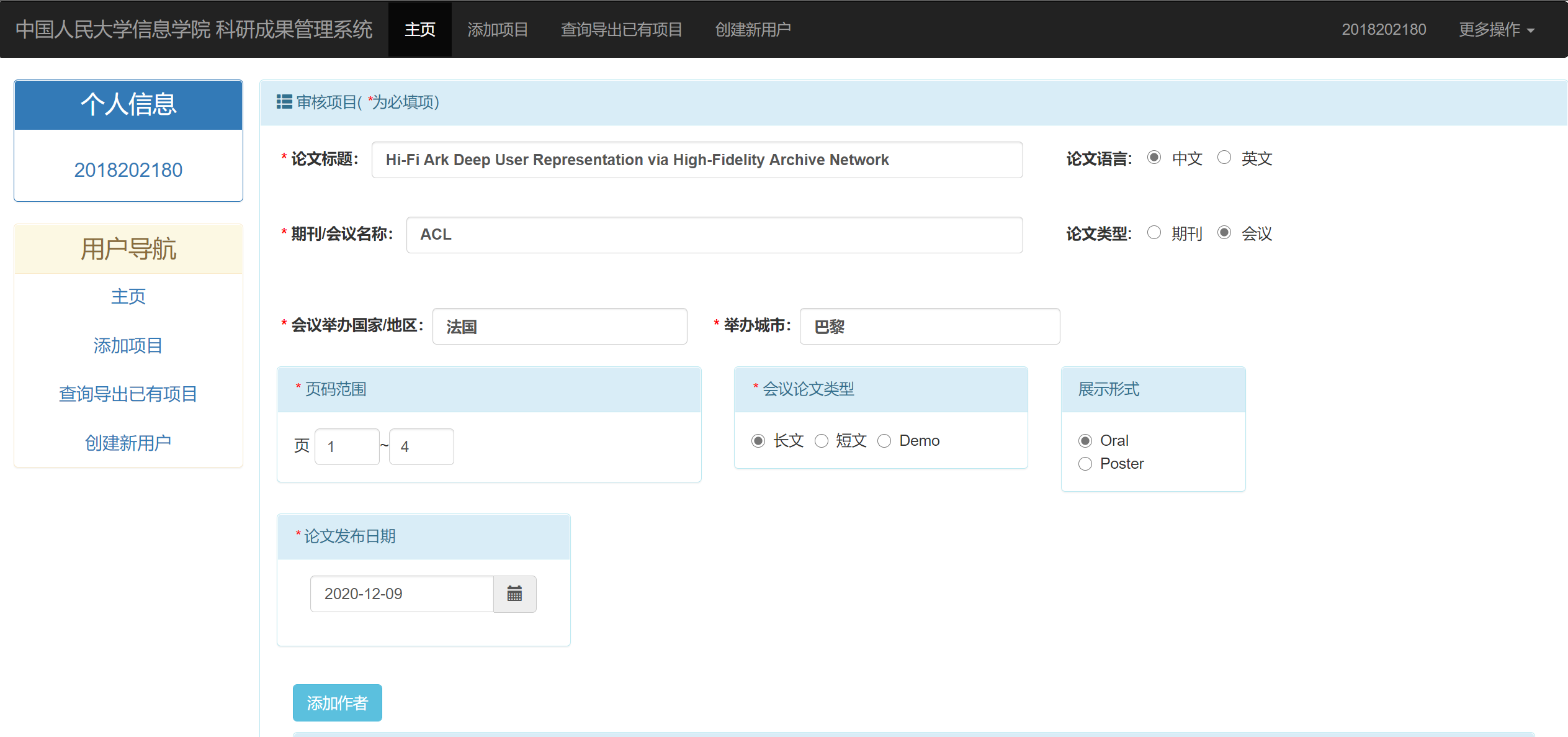


(c)进入审核

输入: 点击审核按钮；

预计输出: 跳转进入审核页面, 显示对应提交信息；

实际输出: 成功跳转, 信息全部对应显示正常；



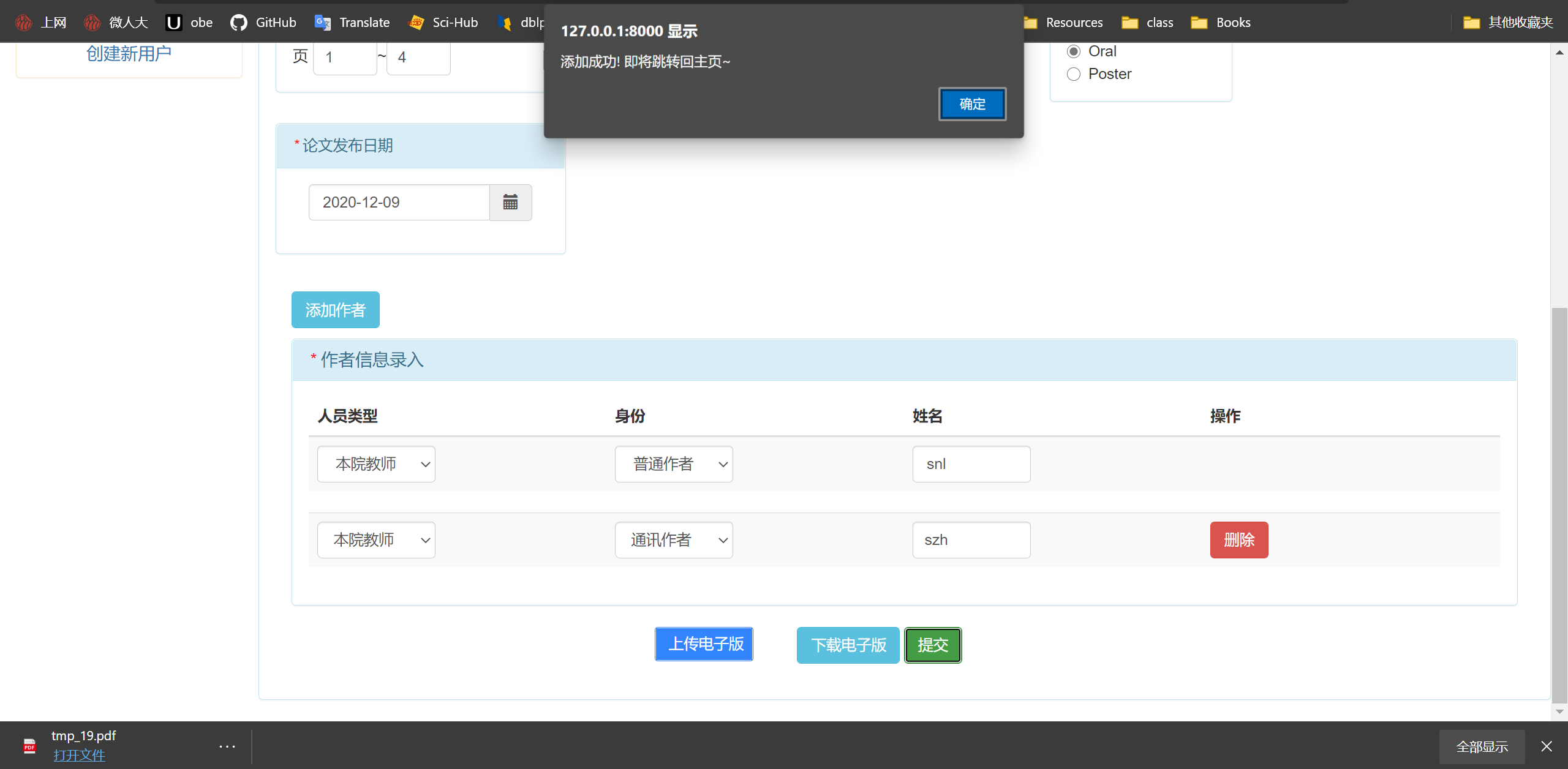


(d)审核

输入: 修改一些信息, 点击下载按钮, 点击上传按钮重新上传, 点击提交按钮；

预计输出: 点击提交后, 修改的内容会覆盖原来内容, 下载对应pdf文件, 新上传的文件会覆盖原有的文件；

实际输出: 各项功能运转正常。



6）模块：查询界面

测试名称：查询结果正确性测试。

测试目的：验证查询功能是否正常。

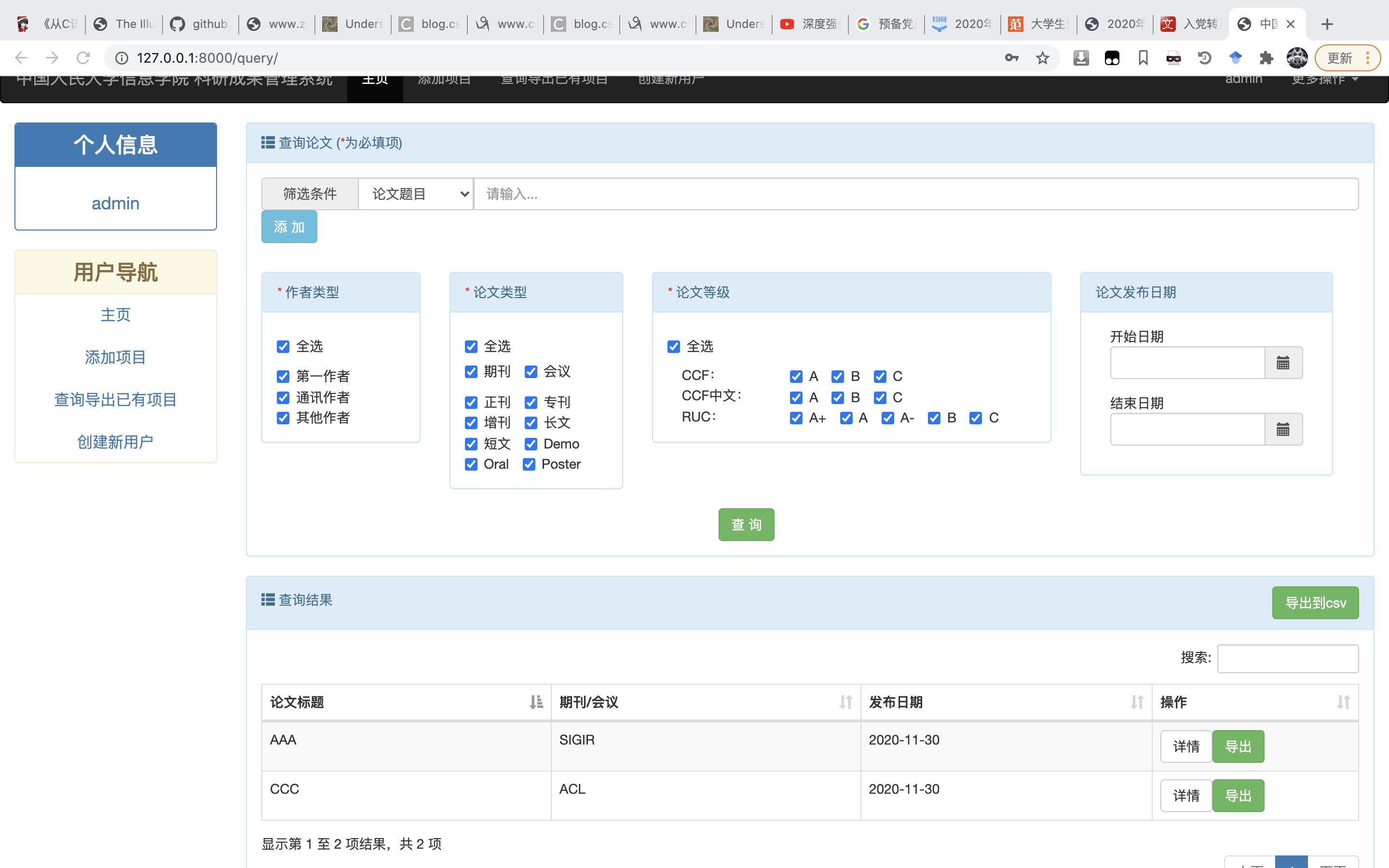
测试：

输入：在查询页面输入想要的筛选条件，并点击“查询”。

期待输出：查询成功，页面显示查询结果。

实际输出：查询成功，页面显示查询结果。

测试截图:



7）模块：详情界面

测试名称：详情页面正确性测试。

测试目的：验证查看详情页功能是否正常。

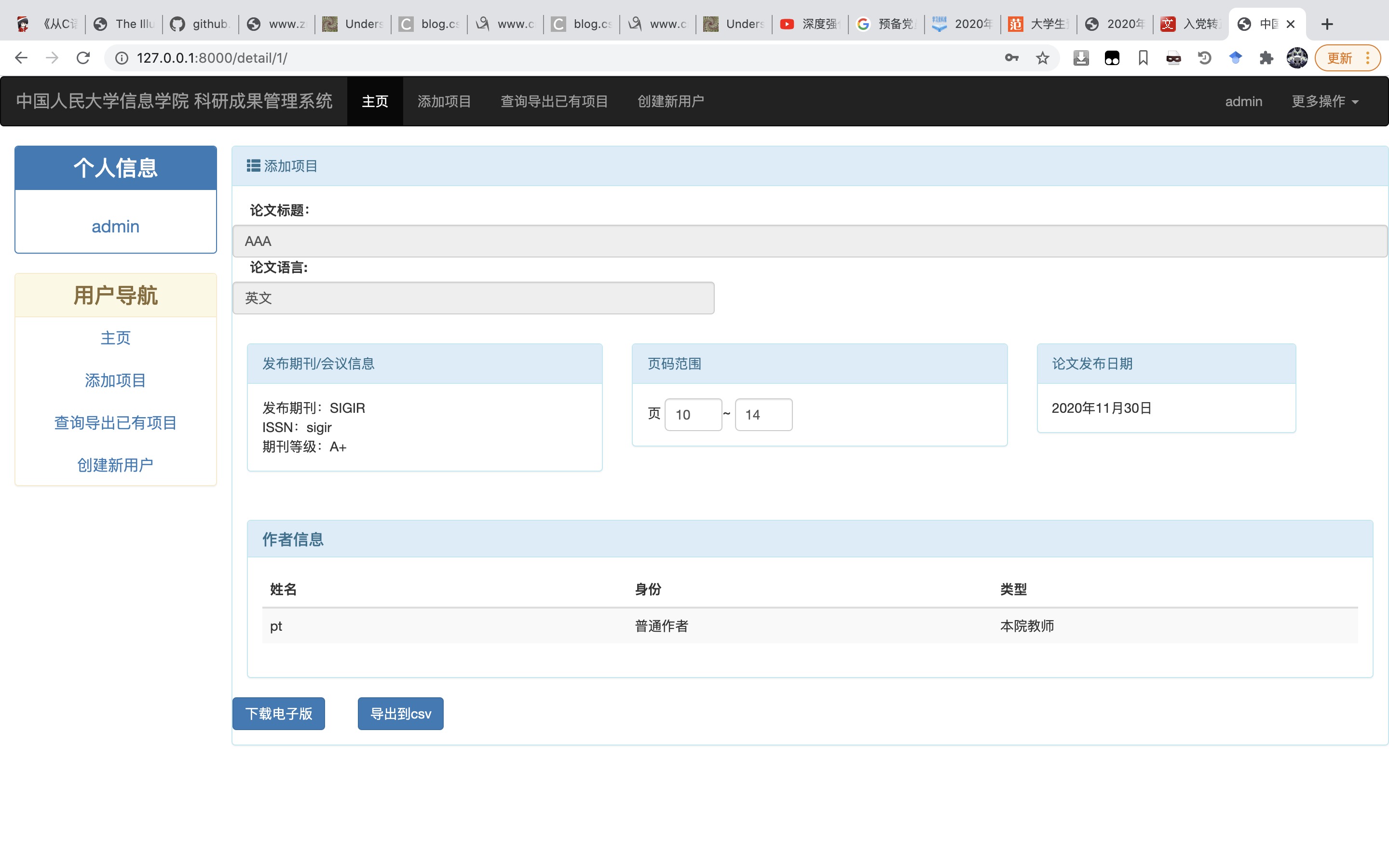
测试：

输入：找到想要查看详情的论文，并点击“详情”。

期待输出：跳转到详情页面成功，页面显示详情内容。

实际输出：跳转到详情页面成功，页面显示详情内容。

测试截图:



8）模块：新建用户页面（管理员）

测试名称：新建用户页面外观与功能测试。

测试目的：验证新建用户页面外观和功能是否正常。

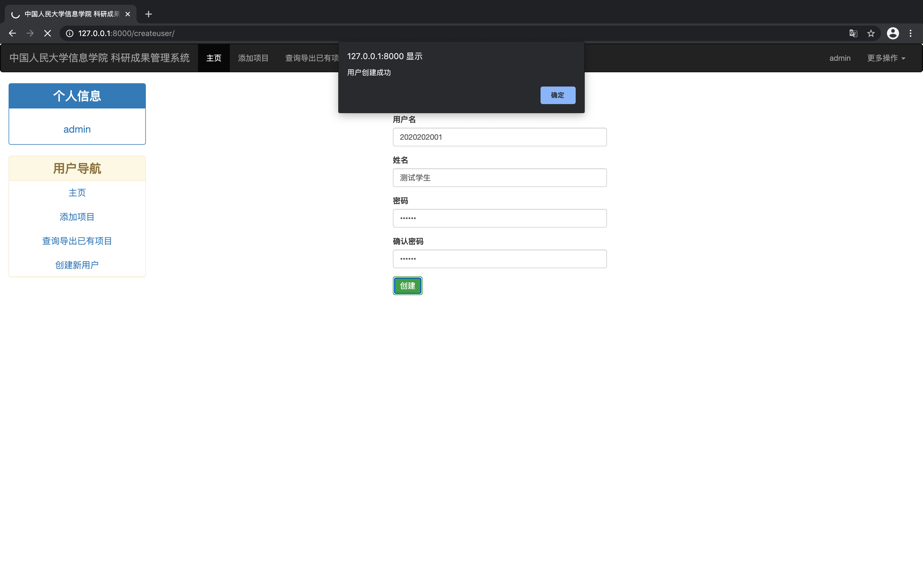
测试：

输入：在create\_user页面输入用户名、密码创建新用户。

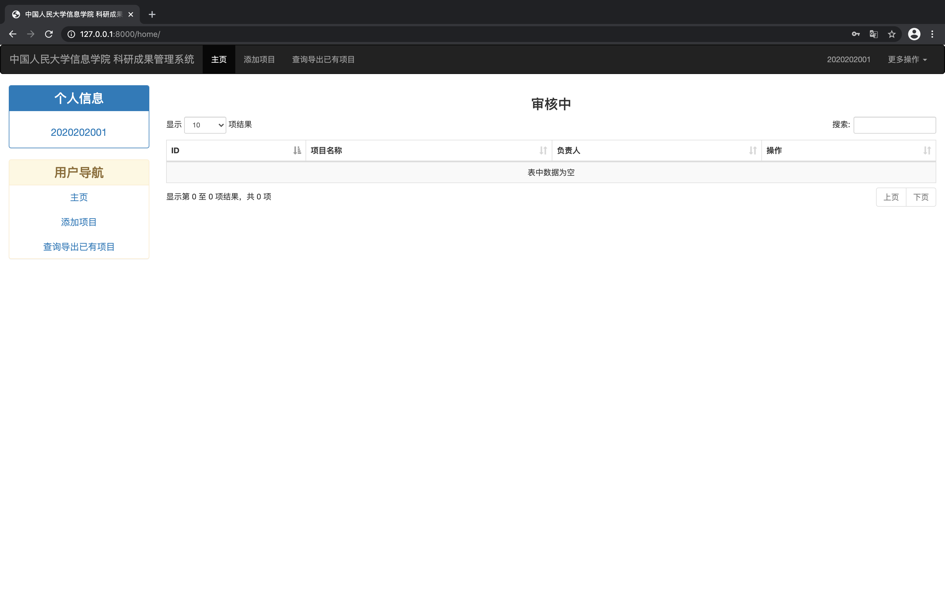
期待输出：新用户创建成功，可以登录。

实际输出：新用户创建成功，可以登录。

测试截图:



（创建用户）



(新用户可以登录)

2．安装及部署说明

Django提供了mod\_wsgi模块可以管理Apache部署, startproject会默认创建wsgi对象, 并预留了充分的接口, 经过简单的配置后即可使用；

由于本项目尚没有以最终成品的方式展现给老师, 如之后有投入生产的需求, 则可以进行后续开发。

附件：系统演示视频链接

https://www.bilibili.com/video/bv1Pr4y1F7q7