

# **文献检索系统 V1.0**

**付德航 谭洁 余锦帆**

**二零二五年三月**

# 目录

<b>1 引言</b>	<b>3</b>
1.1 设计目的 . . . . .	3
1.2 背景 . . . . .	3
<b>2 项目概述</b>	<b>3</b>
2.1 项目概述 . . . . .	3
2.1.1 项目背景 . . . . .	3
2.1.2 项目意义 . . . . .	3
2.1.3 系统前景 . . . . .	4
2.1.4 系统功能概述 . . . . .	4
2.2 用户特点 . . . . .	4
<b>3 可行性研究</b>	<b>4</b>
3.1 技术可行性 . . . . .	4
3.2 经济可行性 . . . . .	4
3.3 操作可行性 . . . . .	5
<b>4 需求分析</b>	<b>5</b>
4.1 建立用例模型 . . . . .	5
4.1.1 业务参与者 . . . . .	5
4.1.2 业务需求用例 . . . . .	5
4.2 性能需求 . . . . .	13
4.2.1 时间特性 . . . . .	13
4.2.2 适应性 . . . . .	13
4.3 其他需求 . . . . .	13
4.3.1 安全性 . . . . .	13
4.3.2 可用性 . . . . .	14
4.3.3 可维护性 . . . . .	14
4.3.4 可扩展性 . . . . .	14
<b>5 概要设计</b>	<b>14</b>
5.1 体系结构架构设计 . . . . .	14
5.2 功能模块概要设计 . . . . .	16
5.2.1 前台系统功能模块 . . . . .	16
5.2.2 后台系统功能模块 . . . . .	17
<b>6 详细设计</b>	<b>18</b>
6.1 数据库物理设计 . . . . .	18
6.1.1 静态数据 . . . . .	18
6.1.2 动态数据 . . . . .	18
6.1.3 表设计 . . . . .	18
6.1.4 数据库逻辑模型概要设计 . . . . .	19
6.2 接口文档设计 . . . . .	19

6.3 UI 界面设计 . . . . .	28
6.3.1 页头 . . . . .	28
6.3.2 首页 . . . . .	28
6.3.3 搜索页 . . . . .	29
6.3.4 详情页 . . . . .	29
6.3.5 用户页面 . . . . .	30
<b>7 测试</b>	<b>31</b>
<b>8 附录</b>	<b>34</b>
8.1 论文的获取 . . . . .	34
8.2 论文的文本提取 . . . . .	35
8.2.1 使用 Grobid 处理 pdf 论文 . . . . .	35
8.2.2 批量处理结果 . . . . .	36
8.2.3 解析文件 . . . . .	36
8.2.4 数据清洗 . . . . .	37
8.3 索引建立以及检索 . . . . .	37
8.3.1 原始文本处理 . . . . .	37
8.3.2 索引建立 . . . . .	37
8.3.3 检索 . . . . .	38

# 1 引言

## 1.1 设计目的

为了应对学术研究中信息检索的复杂性和效率问题，尤其是在学术文献数量庞大、种类繁多的情况下。该系统旨在提供一个平台，使学生能够快速、准确地检索到所需文献，从而提高研究工作的效率，节省宝贵的学习时间。此外，随着教育数字化转型的深入，学生对于文献管理与检索的需求不断增长，传统的图书馆检索方式已无法满足学生对于即时、精准获取信息的期望。因此，一个高效、用户友好的文献检索系统不仅能够辅助学生在学术探索和知识发现的过程，还能促进学术交流与合作，支持跨学科的资源整合，进而为学生提供一个全面、便捷的学术研究环境。通过集成先进的搜索算法、自然语言处理技术，以及推荐系统，该检索系统能够智能地响应学生的查询需求，提供高度相关的文献结果，同时支持用户自定义检索条件，以适应不同学术领域的特定要求。最终，这样的系统将极大地丰富学生的学习体验，加强学术研究的深度与广度，为高等教育机构的教学与研究工作带来显著的价值。

## 1.2 背景

当前学术界对于文献管理与检索的需求日益增长，特别是在学术出版业的蓬勃发展和在线数据库的普及，学生在撰写论文或进行课题研究时，需要查阅大量文献，传统的检索方式已经不能满足快速、精确获取信息的需求。此外，数字化资源的日益丰富，使得学生面对海量信息时，更加需要一个能够整合并有效利用这些资源的工具。因此，一个专为学生设计的文献检索系统不仅能够通过优化检索流程、提供个性化服务，进而支持学术活动，帮助学生更好地管理和利用学术资源，还能促进学术成果的传播和交流，加速知识的更新与创新。通过引入如大数据分析、人工智能等先进技术，该系统能够智能识别研究趋势，预测学术需求，提供前瞻性的信息支持。同时，系统还能根据用户的检索历史和偏好，智能推荐相关文献，从而提升研究工作的针对性和深度。此外，系统的设计注重用户交互体验，确保界面直观易用、操作简便，以降低学习成本，使得学生能够快速上手，专注于学术研究本身。综上所述，一个先进的文献检索系统对于提升学术研究的质量和效率、促进学术界知识共享与合作具有重要意义，是适应数字化时代学术工作的必要工具。

# 2 项目概述

## 2.1 项目概述

### 2.1.1 项目背景

随着 Internet 的迅猛发展和 Web 信息的增加，从海量级的网络信息资源中快速准确地获取信息成为了一个迫切的需求。学术论文作为学术研究的重要成果，其数量也在快速增长。传统的图书馆检索和文献查阅方式已经无法满足研究人员对学术论文的高效获取需求。因此，研发文献检索系统成为了解决这一问题的关键。此外，各类文献的数字化趋势日益明显，尤其是电子期刊等网络文献的增多，为文献检索系统的研发提供了丰富的数据资源和物理条件。同时，高校扩招和公共图书馆借阅高峰等问题也促使论文搜索系统的研发更加迫切，以缓解馆藏资源不足和借阅压力。

### 2.1.2 项目意义

提高学术科研效率。文献检索系统通过关键词、作者、期刊等多种方式提供快速的文献定位功能，帮助研究人员迅速找到自己感兴趣的论文，节省大量的时间和精力。同时，系统允许研究人员

根据自己的研究需求进行筛选和过滤，只获取与自己研究主题相关的高质量论文，提高研究的准确性和深度。

拓展学术视野。文献检索系统涵盖了不同国家、不同学科的相关文献，通过检索，研究人员可以了解不同学者的观点和研究成果，拓宽学术视野，促进学术交流和碰撞。

### 2.1.3 系统前景

技术的更新使文献检索系统能够实现更加智能化的检索和推荐功能，提高检索的准确性和效率。随着数字化趋势的加速，越来越多的学术论文将被数字化并纳入论文搜索系统中，使得系统的数据资源更加丰富和全面。随着技术的不断进步和用户需求的不断变化，论文搜索系统将不断优化用户体验，提供更加个性化、便捷的服务。例如，设置在线客服、提供多种检索方式等。

### 2.1.4 系统功能概述

用户：注册；登录和退出登录；查看和修改个人信息；根据文献名称、作者、发表时间等检索项进行检索；搜索建议；随便看看；查看历史记录；收藏文献；上传论文；转换文献格式

管理员：注册；登录和退出登录；用户信息查看；审核用户上传论文；管理员上传论文

## 2.2 用户特点

本产品的用户包括高校学生和专业科研人员，这两类人群的需求和特点略有区别。专业科研人员通常对特定领域或主题的文献有深入需求，具备较高的信息素养和检索技能，能够灵活运用高级检索功能和专业术语进行检索，他们更注重检索结果的准确性和全面性。大学生主要为了完成作业、论文或研究项目而检索文献，需求相对广泛但可能不够深入，由于检索技能参差不齐，部分学生可能更倾向于使用简单、直观的检索界面和工具。

## 3 可行性研究

### 3.1 技术可行性

随着互联网的迅猛发展和 Web 信息的增加，从海量级的网络信息资源中快速准确地获取信息成为了一个迫切的需求。同时，各类文献的数字化趋势日益明显，尤其是电子期刊等网络文献的增多，为文献检索系统的研发提供了丰富的数据资源和物理条件。因此，从技术角度来看，开发这样一个系统是完全可行的。

### 3.2 经济可行性

随着教育数字化转型的深入，学生对于文献管理与检索的需求不断增长，传统的图书馆检索方式已无法满足学生对于即时、精准获取信息的期望。因此，一个高效、用户友好的文献检索系统不仅能够辅助学生在学术探索和知识发现的过程，还能促进学术交流与合作，支持跨学科的资源整合，进而为学生提供一个全面、便捷的学术研究环境。这将极大地丰富学生的学习体验，加强学术研究的深度与广度，为高等教育机构的教学与研究工作带来显著的价值。因此，从经济角度来看，投资开发这样一个系统是有价值的。

### 3.3 操作可行性

该系统的设计注重用户交互体验，确保界面直观易用、操作简便，以降低学习成本，使得学生能够快速上手，专注于学术研究本身。此外，系统还允许研究人员根据自己的研究需求进行筛选和过滤，只获取与自己研究主题相关的高质量论文，提高研究的准确。

## 4 需求分析

### 4.1 建立用例模型

#### 4.1.1 业务参与者

由需求描述的分析和，可以画出环境图如图 1 所示，由此可以确定两类参与者：用户和管理员

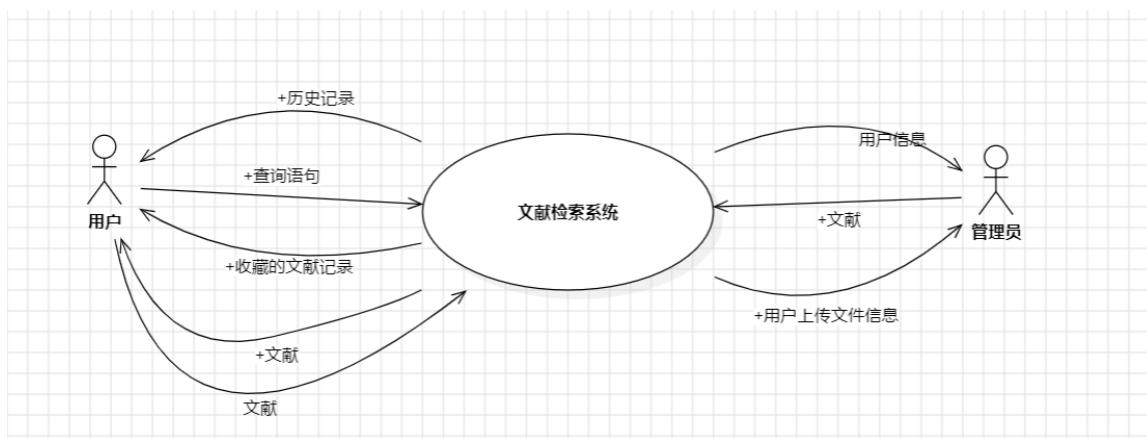


图 1：文献检索系统的环境图

#### 4.1.2 业务需求用例

根据所得到的环境图 1，我们可以确定系统主要的输入输出，并结合未显示的功能需求，得到表 1 所示的用例列表。

参与者	用例	说明
用户	检索	按照不同的检索项检索文献
	随便看看	给用户随机推荐一些文献
	输入提示	当用户输入一个关键词的时候，给出提示选项
	查看详情	让用户在实际看到文献之前，将文献的一些信息先展示出来
	查看历史记录	查看个人浏览的记录
	日期筛选	根据日期筛选得到的结果
	注册、登录	创建和登录个人账号
	文件格式转换	将用户上传的 pdf 文档，转化为 word 文档返回
	上传文献	用户上传自己的文献
	查看和更改个人信息	用户可以查看和改变自己的个人信息
管理员	收藏	用户可以收藏那些自己需要的文档
	登录	管理员账号的登录
	审核文献	审核用户上传的文献
	上传文献	管理员自己也可以上传文献
	查看用户信息	查看已经有的用户的信息

表 1: 文献检索系统的用例列表

接下来，对每个用例进行完整的描述，包括使用用例名称、参与者、前置条件、后置条件、一个主动事件流、零到多个备选事件流，具体见下面各表：

用例名称	检索
参与者	用户
前置条件	用户选择好检索项和输入完毕检索文本
后置条件	如果用例执行成功，那么系统展示出来检索得到的结果，如果没有成功系统给出提示，并且系统保持不变
主事件流	用户点击选项或者用户输入文本时，此用例开始
	用户选择所要检索的领域
	用户输入所要检索的内容
	点击搜索按钮或者按“enter”键
	系统发起检索，并处理结果
	系统展示搜索结果
备选事件流	如果用户没有输入任何文本，那么系统提示“输入不能为空”，此用例结束
	如果前端请求后端服务器失败，系统提示“出现错误，请稍后重试”，此用例结束
	如果检索到的内容为空，则给出空的提示

表 2: 检索的规格说明

用例名称	随便看看
参与者	用户
前置条件	用户打开检索系统主页或点击“换一批”按钮
后置条件	系统展示随机抽取出的三条论文连接
主事件流	用户进入检索系统主页或点击“换一批”按钮，用例开始，系统展示随便看看的显示结果
备选事件流	

表 3: 随便看看的规格说明

用例名称	输入提示
参与者	用户
前置条件	用户开始输入文本
后置条件	如果用例执行成功，那么会在输入框下方出现提示选项 如果没有成功提示选项会保持加载状态
主事件流	用户开始输入文本时，此用例开始 用户选择提示框中的选项进行搜索或仍使用原输入文本进行搜索
备选事件流	如果用户点击输入框后，没有输入任何文本，那么提示框会显示加载状态，不会给出提示选项

表 4: 输入提示的规格说明

用例名称	查看详情
参与者	用户
前置条件	用户点击随便看看中的论文链接链接或点击检索结果中的论文链接
后置条件	跳转到详情页
主事件流	用户点击论文链接时，此用例开始 用户查看详情页中的论文概述 用户点击查看论文原文 用户点击下载论文
备选事件流	

表 5: 查看详情的规格说明

用例名称	检索查看历史记录
参与者	用户
前置条件	用户点击“History”键
后置条件	侧边弹出历史记录页面
主事件流	用户点击“History”键时，此用例开始
	用户查看近一周浏览记录
	用户删除浏览记录
	用户点击浏览记录连接跳转至详情页
备选事件流	如果用户没有任何浏览记录，那么系统显示空状态页面，此用例结束

表 6: 查看历史记录的规格说明

用例名称	日期筛选
参与者	用户
前置条件	用户在搜索页面左边栏中点击按钮筛选日期
后置条件	系统展示筛选出的所有结果
主事件流	系统默认日期为“Any time”，用户点击其他不同按钮时，此用例开始，系统展示筛选出的所有结果
备选事件流	

表 7: 日期筛选的规格说明

用例名称	注册、登录
参与者	用户
前置条件	用户点击检索系统上侧导航栏中的头像进行注册或登录
后置事件	如果没有注册，用户需要输入用户名和密码进行注册，并保证账号的唯一性
	如果注册了，用户输入用户名和密码进行登录
主事件流	用户点击检索系统上侧导航栏中的头像时，此用例开始
	如果没有注册，用户设置用户名和密码进行注册
	如果注册成功了，用户输入用户名和密码进行登录，登陆成功进入系统主页
备选事件流	如果用户没有注册登录，那么系统不会保留用户的历史记录
	如果用户的用户名或密码输入错误，系统会弹出错误提示
	如果用户忘记密码，点击“忘记密码”按钮后进入修改密码页面

表 8: 注册、登录的规格说明

用例名称	文件格式转换
参与者	用户
前置条件	用户在个人空间中点击“文件格式转换”按钮，然后上传 PDF 格式文件
后置条件	系统将文件转换成 word 文档返回给用户
主事件流	用户进入个人空间点击“文件格式转换”按钮，用例开始 用户在指定区域上传 PDF 格式文件 系统处理文件成 word 文档返回给用户
备选事件流	

表 9: 文件格式转换的规格说明

用例名称	上传文献
参与者	用户
前置条件	用户进入个人空间点击“上传文献”按钮并上传文献
后置条件	系统将文献保存在用户个人空间的“收藏”中
主事件流	用户进入个人空间点击“上传文献”按钮，用例开始 用户在指定区域上传文献（PDF 格式） 上传成功后文献将展示在用户个人空间的“收藏”中
备选事件流	

表 10: 上传文献的规格说明

用例名称	查看和更改个人信息
参与者	用户
前置条件	用户已登录
后置条件	如果用例执行成功，系统展示用户的个人信息，并允许用户进行更改 如果没有成功系统给出提示，并且系统保持不变
主事件流	用户点击“我的账户”或“个人信息”选项时，此用例开始 系统展示当前用户的个人信息 用户选择要更改的信息字段并输入新的信息 用户点击“保存”按钮或按“enter”键提交更改 系统验证新信息的有效性 如果信息有效，系统更新用户信息并提示“信息更新成功”；如果无效，系统提示“请输入有效的信息”，此用例结束
备选事件流	如果用户未登录，系统提示“请先登录”，此用例结束

表 11: 查看和更改个人信息的规格说明

用例名称	收藏
参与者	用户
前置条件	用户已登录且正在浏览文献页面
后置条件	如果用例执行成功，那么系统将文献添加到用户的收藏列表中 如果没有成功系统给出提示，并且系统保持不变
主事件流	用户在文献详情页面点击“收藏”按钮或图标时，此用例开始 系统检查用户是否已登录，若未登录则跳转到登录页面 如果用户已登录，系统将该文献添加到用户的收藏列表中 系统提示“文献已添加到收藏列表”
备选事件流	如果添加失败（如网络问题），系统提示“收藏失败，请稍后再试”，此用例结束

表 12: 收藏的规格说明

用例名称	管理员登录注册
参与者	管理员
前置条件	无（对于注册）；管理员已拥有账号（对于登录）
后置条件	如果用例执行成功，那么管理员将进入管理界面； 如果没有成功系统给出提示，并且系统保持不变
主事件流	管理员选择“登录”或“注册”选项时，此用例开始
主事件流（登录）	管理员输入用户名和密码，点击“登录”按钮或按“enter”键 系统验证用户名和密码的正确性 如果验证通过，系统进入管理界面；如果失败，系统提示“用户名或密码错误”，此用例结束
主事件流（注册）	管理员输入必要的注册信息（如用户名、密码、邮箱等），点击“注册”按钮或按“enter”键 系统验证信息的完整性和有效性 如果信息有效，系统创建管理员账号并自动登录进入管理界面；如果无效，系统提示“请输入有效的信息”，此用例结束
备选事件流	如果网络连接有问题或其他技术故障导致操作失败，系统提示“操作失败，请稍后再试”，此用例结束

表 13: 管理员登录注册的规格说明

用例名称	审核文献
参与者	管理员
前置条件	存在未被审核的上传文件
后置条件	如果用例执行成功，那么文献根据管理员决策决定是否进入系统数据库； 如果没有成功系统给出提示，并且系统保持不变
主事件流	管理员选择“审核”选项时，此用例开始 系统展示待审核文献目录 管理员选择要审核的文献 系统展示管理员选择的文献具体内容 管理员评判是否通过审核，如果审核通过则将文献收录入系统数据库； 如果审核未通过则发送消息提示上传该文献的用户审核未通过
备选事件流	如果网络连接有问题或其他技术故障导致操作失败，系统提示“操作失败，请稍后再试”，此用例结束

表 14: 管理员审核文献的规格说明

用例名称	上传文献
参与者	管理员
前置条件	管理员已登录
后置条件	如果用例执行成功，那么上传的文献进入系统数据库； 如果没有成功系统给出提示，并且系统保持不变
主事件流	管理员选择“上传文献”选项时，此用例开始 管理员选择要上传的文献 系统验证上传的文献格式是否符合要求 如果验证通过，上传文献进入系统数据库 如果失败，系统提示“文献格式错误”，此用例结束
备选事件流	如果网络连接有问题或其他技术故障导致操作失败，系统提示“操作失败，请稍后再试”，此用例结束

表 15: 管理员上传文献的规格说明

用例名称	查看用户信息
参与者	管理员
前置条件	管理员已登录
后置条件	显示用户信息
主事件流	管理员选择“查看用户信息”选项时，此用例开始
	系统展示所有已注册用户 ID
	管理员选择要查看的用户 ID
	系统展示管理员选择的用户详情信息
备选事件流	如果网络连接有问题或其他技术故障导致操作失败，系统提示“操作失败，请稍后再试”，此用例结束

表 16: 管理员查看用户信息的规格说明

在得到上述用例规格说明后，我们进一步得到文献检索系统的用例图，如图2以及图3：

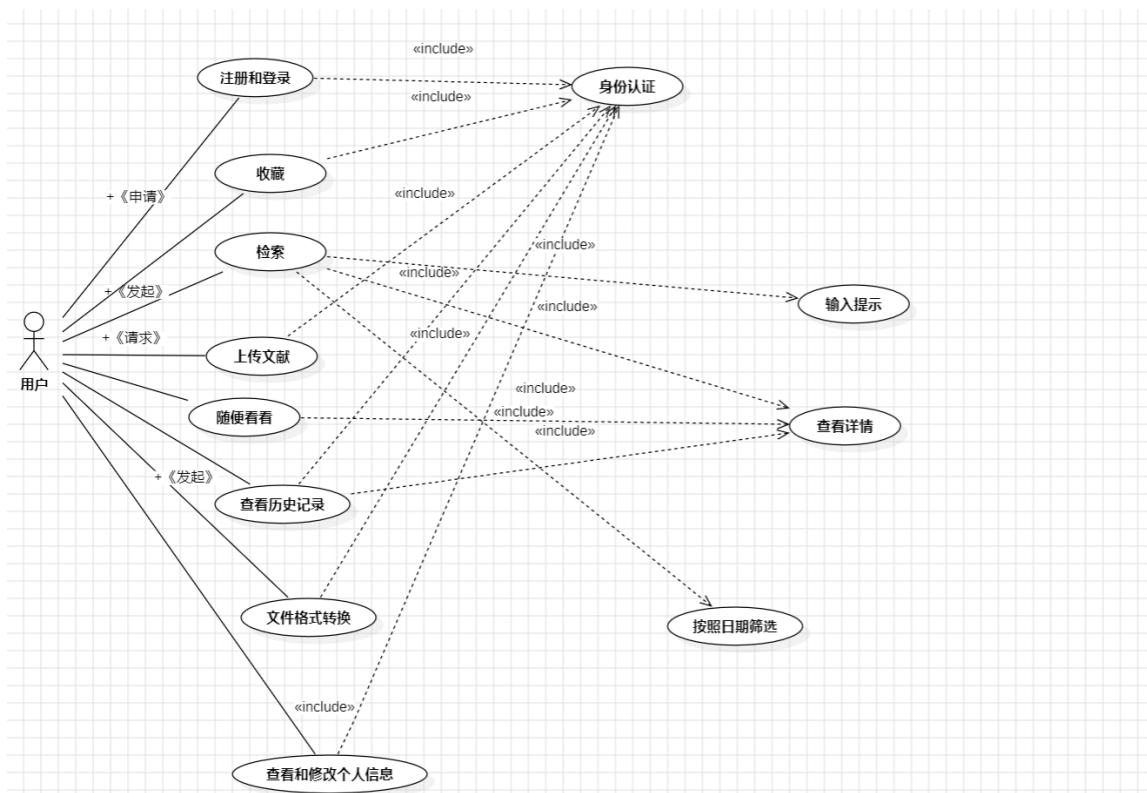


图 2: 用户用例图

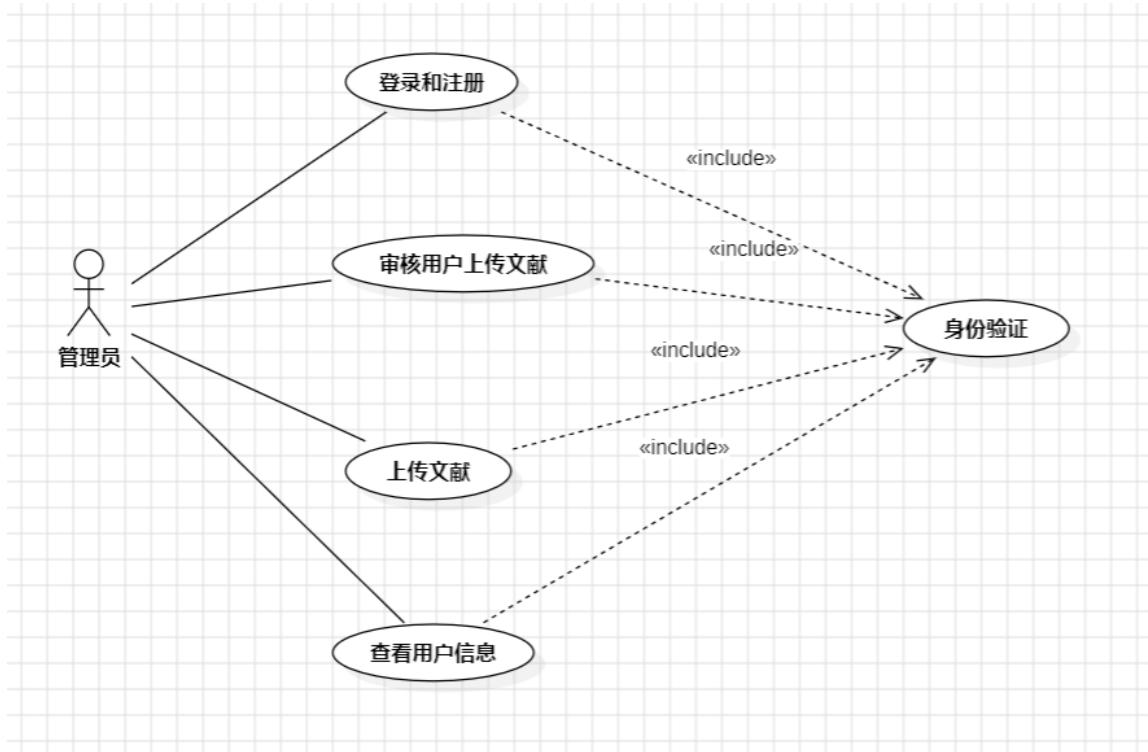


图 3: 管理员用例图

## 4.2 性能需求

### 4.2.1 时间特性

- 响应时间：系统应能在合理的时间内响应用户的查询请求，通常应在几秒内返回结果。
- 数据处理速度：系统应能快速处理和检索大量论文数据，确保检索结果的实时性。

### 4.2.2 适应性

- 负载均衡：系统应具备负载均衡机制，以确保在高负载情况下仍能保持稳定的性能。
- 并发处理能力：系统应能支持一定数量的并发用户同时进行检索操作，而不显著影响性能。

## 4.3 其他需求

### 4.3.1 安全性

- 数据加密：用户密码、敏感数据应进行加密存储，传输过程中应使用 HTTPS 等加密协议。
- 访问控制：系统应实施严格的访问控制策略，确保只有授权用户和管理员能访问特定资源。
- 防止 SQL 注入：系统应具备防止 SQL 注入攻击的能力，确保数据库安全。
- 防止跨站脚本攻击 (XSS)：系统应具备防止 XSS 攻击的能力，确保用户输入的安全性。

#### 4.3.2 可用性

1. 用户界面友好：系统应提供直观、易用的用户界面，方便用户进行检索操作。
2. 错误处理：系统应提供清晰的错误提示信息，帮助用户和管理员快速定位和解决问题。
3. 帮助文档：系统应提供详细的帮助文档和用户指南，帮助用户和管理员快速上手。

#### 4.3.3 可维护性

1. 模块化设计：系统应采用模块化设计，便于后续的功能扩展和维护。
2. 代码可读性：代码应具备良好的可读性和注释，便于开发人员理解和维护。

#### 4.3.4 可扩展性

1. 水平扩展：系统应支持水平扩展，即通过增加服务器数量来提升系统性能。
2. 垂直扩展：系统应支持垂直扩展，即通过升级硬件（如 CPU、内存）来提升系统性能。
3. 插件机制：系统应提供插件机制，便于第三方开发者扩展系统功能。

## 5 概要设计

### 5.1 体系结构架构设计

文件检索系统的体系结构架构采用分层设计的方法来组织系统的不同功能模块，包括数据访问层、用户界面层、业务逻辑层、接口层和访问层，涵盖用户身份管理、检索服务、接口管理、数据库管理等方面的内容，这种设计理念具有可扩展性、灵活性和高可用性等优点。

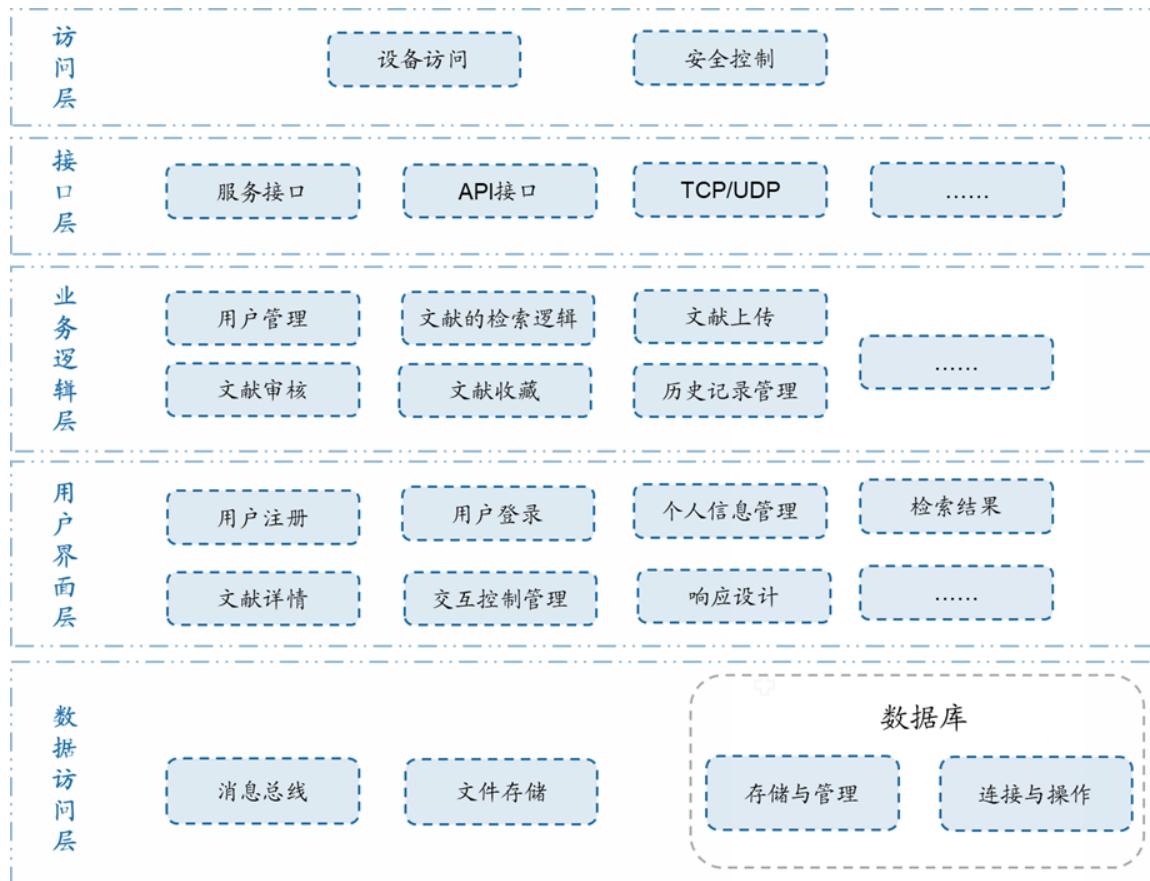


图 4: 体系结构架构概要设计

同时，文件检索系统采用了客户端-服务器模式。这种架构将系统的前端用户界面（客户端）与后端数据处理和存储（服务器）分离开来。在这种模式下，客户端负责向用户提供交互界面，使用户能够输入查询条件并查看检索结果。而服务器则承担了处理用户请求、执行文件检索算法以及管理文件索引等核心任务。通过这种分工，系统不仅能够提高响应速度和处理能力，还能确保数据的安全性和一致性。此外，客户端-服务器模式还具有良好的扩展性和灵活性，使得系统能够轻松应对不同规模和复杂度的应用需求。

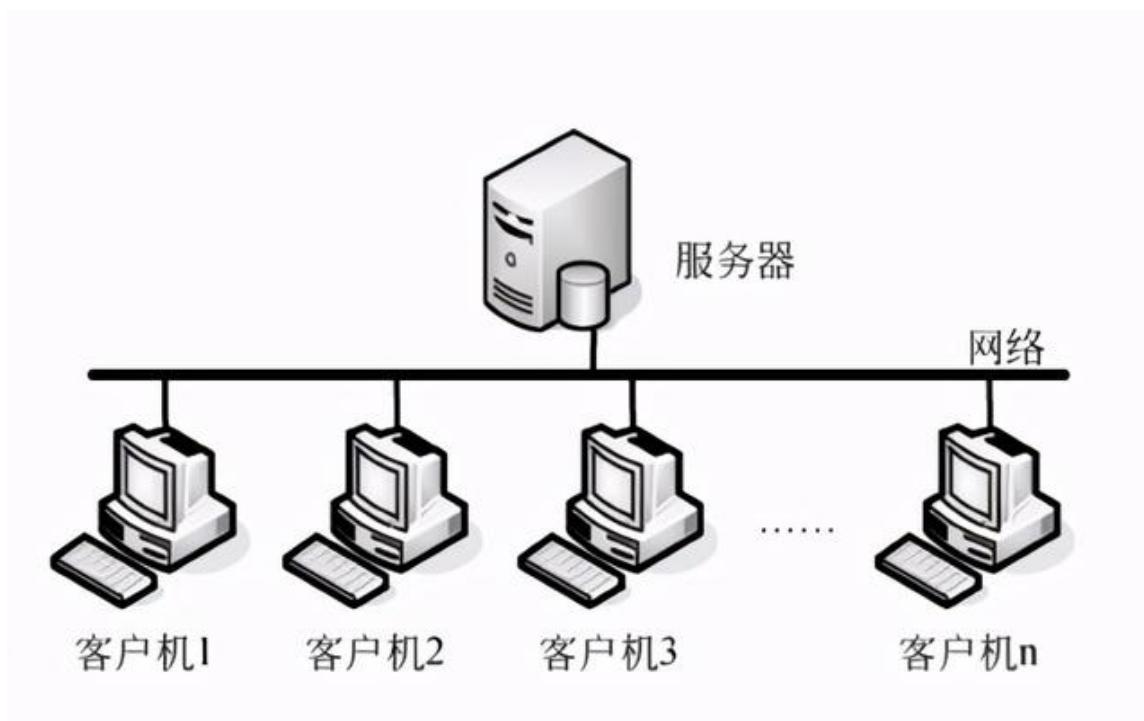


图 5: C/S 体系结构

## 5.2 功能模块概要设计

该平台涵盖前台和后台系统，包括用户登录注册、查看和更改个人信息、检索文献、收藏文献、查看历史纪录等功能。通过合理的分层设计，实现了系统的可扩展性、灵活性和高可用性。

### 5.2.1 前台系统功能模块

前台系统是用户直接交互的界面，包括注册登录、查看和更改个人信息、检索文献、查看历史记录等功能模块。

(1) 注册、登录：用户可以通过点击检索系统上侧导航栏中的头像进行注册或登录。注册时需要设置唯一的用户名和密码，而登录则需要输入已注册的用户名和密码。这一步骤确保了用户数据的安全和个人化服务。

(2) 查看和更改个人信息：已登录用户可以查看和修改自己的个人信息，如姓名、邮箱等，以保持账户信息的准确性和最新性。

(3) 随便看看：用户可以通过点击“随便看看”按钮或“换一批”按钮，随机查看系统抽取出的三条论文链接。这个模块提供了一个轻松浏览最新或热门研究的方式，增加用户的探索兴趣。

(4) 输入提示：当用户开始输入文本时，系统会在输入框下方出现提示选项。如果用例执行成功，那么会在输入框下方出现提示选项；如果没有成功提示选项会保持加载状态。此模块旨在帮助用户快速找到他们感兴趣的研究领域或具体论文。

(5) 查看详情：用户可以通过点击随便看看中的论文链接或检索结果中的论文链接来跳转到详情页。详情页提供论文的概述、查看原文和下载选项，使用户能够深入了解选定的论文内容。

(6) 日期筛选：用户可以根据需要选择不同的日期范围进行搜索，系统会根据选择展示相应的结果。这允许用户更精确地查找特定时间范围内的文献资料。

(7) 检索查看历史记录：用户可以点击“历史”键查看近一周的浏览记录，包括删除记录和通过记录链接跳转至详情页的选项。这有助于用户跟踪他们的搜索历史和重新访问之前感兴趣的内容。

(8) 上传文献：用户可以在个人空间点击“上传文献”按钮并上传文献(PDF格式)，上传成功后文献将展示在此用户个人空间的“收藏”中。待管理员审核通过后，文献会被纳入系统数据库中供其他用户访问使用。

(9) 收藏：已登录用户在浏览文献页面时可以点击“收藏”按钮或图标将文献添加到个人的收藏列表中，方便以后查阅。

### 5.2.2 后台系统功能模块

后台系统是管理员进行管理的界面，包括系统初始化、用户管理、审核文献、上传文献等功能模块。

(1) 系统初始化模块：管理员可以进行系统的初始化操作，如设置网站名称、添加初始管理员账号等。

(2) 管理员登录注册：为管理员提供登录和注册的功能，确保只有授权的管理员才能访问后台管理界面进行操作。

(3) 用户管理模块：管理员可以查看所有用户的信息，包括用户名、手机号、邮箱等，管理员还可以对用户进行分类管理，如设置普通用户和管理员用户。

(4) 审核文献：管理员可以对待审核的上传文件进行审查，决定是否将其纳入系统数据库。这是保证平台内容质量和合法性的关键步骤。

(5) 上传文献：管理员还可以直接上传文献到系统中，这些文献经过验证后会被添加到系统数据库中供用户检索和使用。

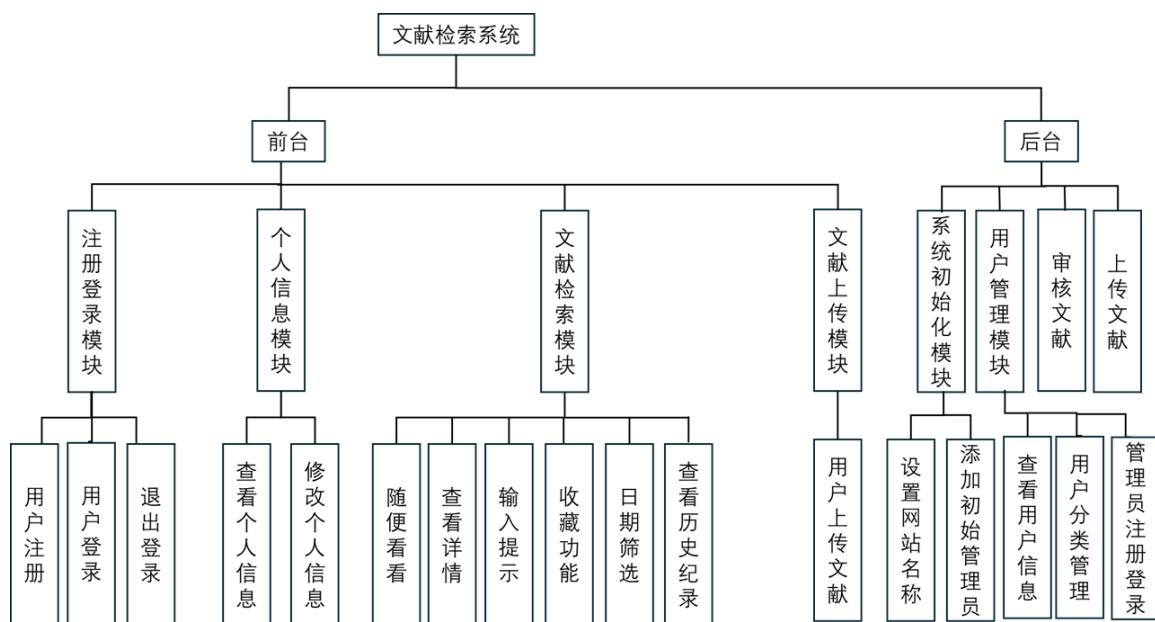


图 6: 功能模块概要设计

以上是文献检索系统的前台和后台主要功能模块的设计说明，每个模块都旨在提高用户体验、增强系统的功能性和安全性。

## 6 详细设计

在详细设计之前，我们首先要对论文文件进行获取和处理，具体方法参见附录。

### 6.1 数据库物理设计

#### 6.1.1 静态数据

用户信息、文献信息

#### 6.1.2 动态数据

用户查询记录、用户收藏消息、文献更新消息、系统通知和消息

#### 6.1.3 表设计

##### 1. 历史记录表

RecordID	记录 ID<PK>	VARCHAR2	255
UserID	用户 ID	VARCHAR2	32
Title	文章标题	VARCHAR2	255
Date	记录日期	DATETIME	

表 17: 历史记录表

##### 2. 用户信息表

UserID	用户 ID<PK>	VARCHAR2	255
UserName	用户名	VARCHAR2	32
RecordID	密码 (MD5)	VARCHAR2	255
Email	邮箱	VARCHAR2	32
Avatar	头像	IMAGE	
IsAdmin	是否为管理员	BOOL	

表 18: 用户信息表

##### 3. 收藏表

CollectID	收藏 ID<PK>	VARCHAR2	255
UserID	用户 ID	VARCHAR2	255
Title	标题	VARCHAR2	32
Domain	领域	VARCHAR2	32

表 19: 收藏表

##### 4. 待审核文件表

DocID	待审核文件 ID<PK>	VARCHAR2	255
UserID	用户 ID	VARCHAR2	255
UploadTime	上传时间	DATETIME	
State	状态	VARCHAR2	32
DocName	文件名	VARCHAR2	32

表 20: 待审核文件表

#### 6.1.4 数据库逻辑模型概要设计

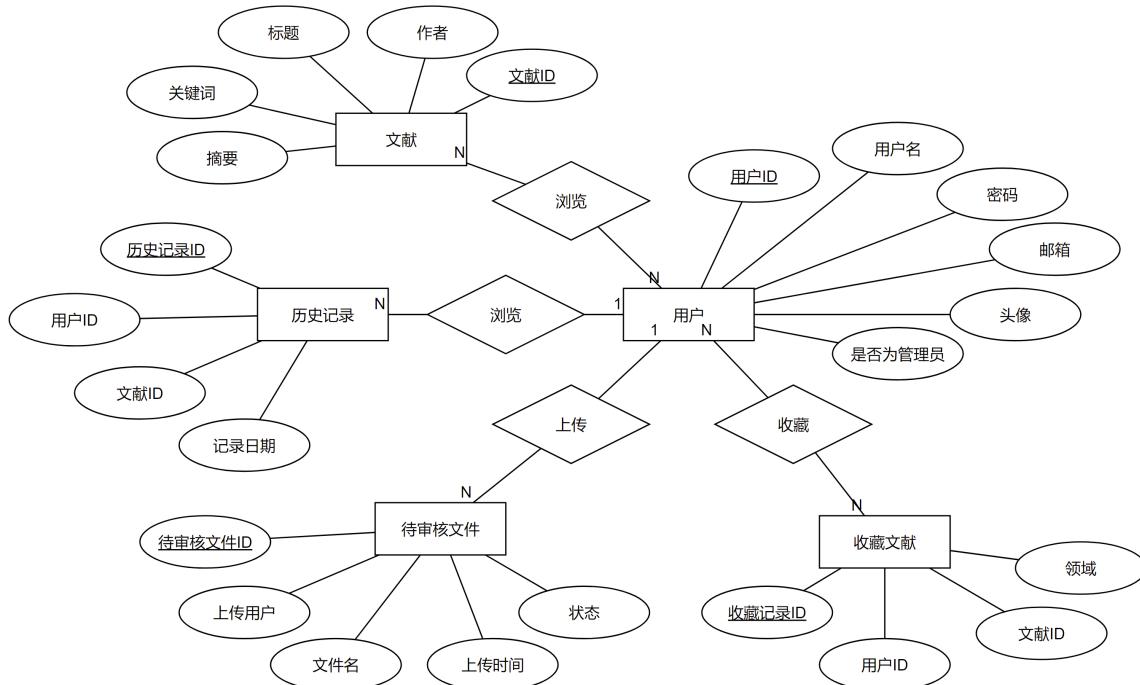


图 7: ER 图

**实体:** 用户, 历史记录, 待审核文件, 收藏文件

**联系:** 见 ER 图

**属性:**

1. 用户 (用户 ID+ 用户名 + 密码 + 邮箱 + 头像 + 是否为管理员)
2. 文献 (文献 ID+ 作者 + 标题 + 关键词 + 摘要)
3. 历史记录 (历史记录 ID+ 用户 ID+ 文献 ID+ 记录时间)
4. 待审核文件 (待审核文件 ID+ 上传用户 + 文件名 + 上传时间 + 状态)
5. 收藏文献 (收藏记录 ID+ 用户 ID+ 文献 ID+ 领域)

## 6.2 接口文档设计

由于需要前后端分离进行开发，所以我们必须做好前后端发送请求的文档，包括请求方式、请求头、请求参数名等，具体如下：

接口名	改变待审核文章状态
参与者	管理员
请求类型	POST
请求路径	/admin/change_upload_record_state
请求头	token
请求参数	id(待审核文档 id, integer) state(请求改变的状态, string)
响应数据	code(响应码 0 代表成功, integer) msg(返回提示, string) data(返回数据, 字典或列表)

表 21: 改变待审核文章状态接口文档

接口名	删除一条待审核记录
参与者	管理员
请求类型	DELETE
请求路径	/admin/delete_upload_record
请求头	token
请求参数	id(待审核文档 id, integer)
响应数据	code(响应码 0 代表成功, integer) msg(返回提示, string) data(返回数据, 空)

表 22: 删除一条待审核记录接口文档

接口名	获取全部待审核的记录
参与者	管理员
请求类型	GET
请求路径	/admin/get_upload_record
请求头	token
请求参数	page(当前页码, integer) pageSize(每页大小, integer)
响应数据	code(响应码 0 代表成功, integer) msg(返回提示, string) data(返回数据, 字典 (包含总数量 (total) 和具体数据 (records)) )

表 23: 获取全部待审核的记录接口文档

接口名	获取用户信息
参与者	管理员
请求类型	GET
请求路径	/admin/get_user_info
请求头	token
请求参数	page(当前页码, integer)
	pageSize(每页大小, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 字典 (包含总数量 (total) 和具体数据 (records)) )

表 24: 获取用户信息接口文档

接口名	获取搜索结果
参与者	用户
请求类型	GET
请求路径	/search/get_search_result
请求头	无
请求参数	dateChoice(日期选项, integer)
	currentPage(当前页码, integer)
	field(检索领域, string)
	queryText(检索文本, string)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 列表 (文档对象))

表 25: 获取搜索结果接口文档

接口名	获取输入建议
参与者	用户
请求类型	GET
请求路径	/search/get_suggest
请求头	无
请求参数	field(检索领域, string)
	queryText(检索文本, string)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 列表 (string))

表 26: 获取输入建议接口文档

接口名	随便看看
参与者	用户
请求类型	GET
请求路径	/search/random_look
请求头	无
请求参数	无
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 列表 (文档对象) )

表 27: 随便看看接口文档

接口名	收藏一篇文献
参与者	用户
请求类型	POST
请求路径	/user/add_collection
请求头	token
请求参数	id(收藏记录 id, integer)
	title(收藏文档标题, string)
	userId(用户 id, integer)
	domain(用户的哪一个分类, string)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 空)

表 28: 收藏一篇文献接口文档

接口名	增加一个收藏分类
参与者	用户
请求类型	POST
请求路径	/user/add_folder
请求头	token
请求参数	userId(用户 id, integer)
	newDomain(用户新增分类, string)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 空)

表 29: 增加一个收藏分类接口文档

接口名	添加浏览记录
参与者	用户
请求类型	POST
请求路径	/user/add_history
请求头	token
请求参数	id(收藏记录 id, integer)
	date(日期, string(date-time))
	userId(用户 id, integer)
	title(收藏文章标题, string)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 空)

表 30: 添加浏览记录接口文档

接口名	解析文章
参与者	用户
请求类型	POST
请求路径	/user/analysis
请求头	token
请求参数	file(上传文件, file)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, string)

表 31: 解析文章接口文档

接口名	取消收藏一篇文章
参与者	用户
请求类型	DELETE
请求路径	/user/cancel_collected
请求头	token
请求参数	id(收藏记录 id, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 空)

表 32: 取消收藏一篇文章接口文档

接口名	更换头像
参与者	用户
请求类型	POST
请求路径	/user/change_avatar
请求头	token
请求参数	file(上传头像, file)
	userId(用户 id, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(头像访问链接, string)

表 33: 更换头像接口文档

接口名	修改密码
参与者	用户
请求类型	POST
请求路径	/user/change_password
请求头	token
请求参数	password(新密码, string)
	userId(用户 id, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 空)

表 34: 修改密码接口文档

接口名	修改用户名
参与者	用户
请求类型	POST
请求路径	/user/change_user_name
请求头	token
请求参数	userName(新用户名, string)
	userId(用户 id, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 空)

表 35: 修改用户名接口文档

接口名	删除某一分类的全部收藏
参与者	用户
请求类型	DELETE
请求路径	/user/delete_folder
请求头	token
请求参数	domain(分类, string) userId(用户 id, integer)
响应数据	code(响应码 0 代表成功, integer) msg(返回提示, string) data(返回数据, 空)

表 36: 删除某一分类的全部收藏接口文档

接口名	删除浏览记录
参与者	用户
请求类型	DELETE
请求路径	/user/delete_history
请求头	token
请求参数	historyIds(历史记录 ids, array)
响应数据	code(响应码 0 代表成功, integer) msg(返回提示, string) data(返回数据, 空)

表 37: 删除浏览记录接口文档

接口名	发送邮件
参与者	用户
请求类型	POST
请求路径	/user/email
请求头	无
请求参数	target(目标邮箱, string) isRegister(是否是注册, boolean)
响应数据	code(响应码 0 代表成功, integer) msg(返回提示, string) data(返回数据, 空)

表 38: 发送邮件接口文档

接口名	获取全部收藏分类
参与者	用户
请求类型	GET
请求路径	/user/get_all_folder
请求头	token
请求参数	userId(用户 id, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 列表 (string) )

表 39: 获取全部收藏分类接口文档

接口名	获取某一分类下收藏的文章
参与者	用户
请求类型	GET
请求路径	/user/get_collection
请求头	token
请求参数	domain(分类, string)
	page(当前页码, integer)
	pageSize(每页大小, integer)
	userId(用户 id, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 字典 (总数量 (total) 和全部数据 (records)) )

表 40: 获取全部收藏分类接口文档

接口名	获取浏览记录
参与者	用户
请求类型	GET
请求路径	/user/get_history
请求头	token
请求参数	userId(用户 id, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 列表 (历史记录对象) )

表 41: 获取浏览记录接口文档

接口名	获取上传记录
参与者	用户
请求类型	GET
请求路径	/user/get_upload_record
请求头	token
请求参数	userId(用户 id, integer)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 列表 (上传记录对象) )

表 42: 获取上传记录接口文档

接口名	登录
参与者	用户
请求类型	POST
请求路径	/user/login
请求头	token
请求参数	name(用户名或邮箱, string)
	password(密码, string)
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 用户对象)

表 43: 登录接口文档

接口名	注册
参与者	用户
请求类型	POST
请求路径	/user/register
请求头	无
请求参数	用户对象
响应数据	code(响应码 0 代表成功, integer)
	msg(返回提示, string)
	data(返回数据, 空)

表 44: 注册接口文档

接口名	上传文献
参与者	用户
请求类型	POST
请求路径	/user/upload
请求头	token
请求参数	userId(用户 id, integer) file(上传的文件, file)
响应数据	code(响应码 0 代表成功, integer) msg(返回提示, string) data(返回数据, 空)

表 45: 上传文献接口文档

接口名	验证验证码
参与者	用户
请求类型	POST
请求路径	/user/verify_code
请求头	无
请求参数	target(邮箱, string) verificationCode(验证码, string)
响应数据	code(响应码 0 代表成功, integer) msg(返回提示, string) data(返回数据, 空)

表 46: 验证验证码接口文档

### 6.3 UI 界面设计

前端页面我们一共分为了页头、首页、搜索页、详情页、用户界面这五个主要的部分，具体如下

#### 6.3.1 页头

页头中包含了用户、历史记录、回到首页三个按钮，以及我们的项目名和项目图标，如图8，这里点击用户可以实现登录（跳转到用户详情页），而历史记录则会展示出用户的历史记录（如果已经登录），如果在其他的页面（非首页），点击 home 可以回到首页。



图 8: 页头设计

#### 6.3.2 首页

首页中包含了检索和随便看看，这里右侧可以实现选择不同的领域，输入框中为检索的文本，如图9。

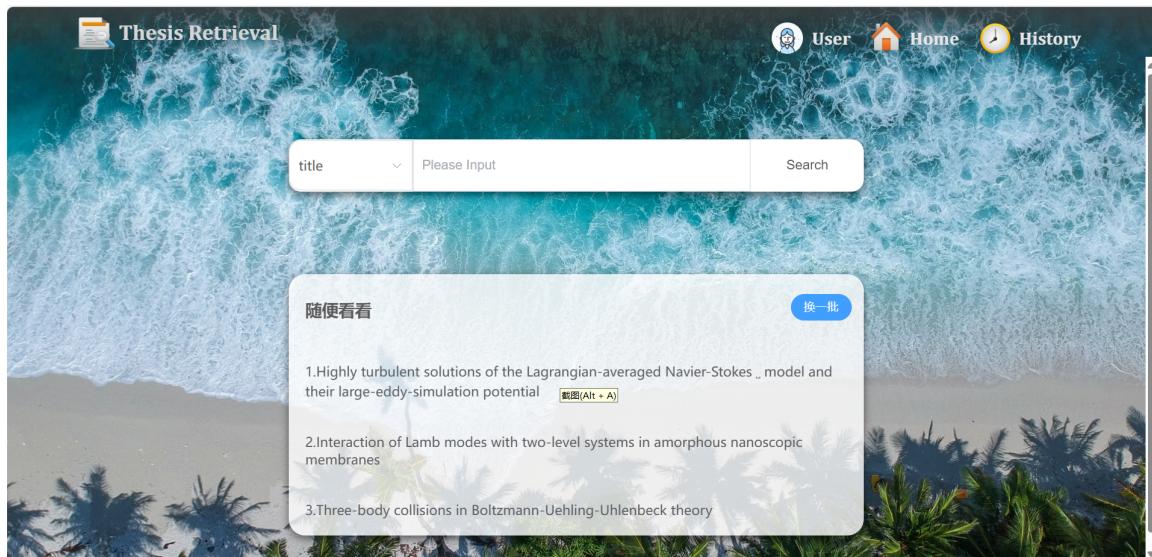


图 9: 首页设计

### 6.3.3 搜索页

搜索页中点击可以跳转到详情页，在右侧选择筛选的日期可以进行筛选，如图10。

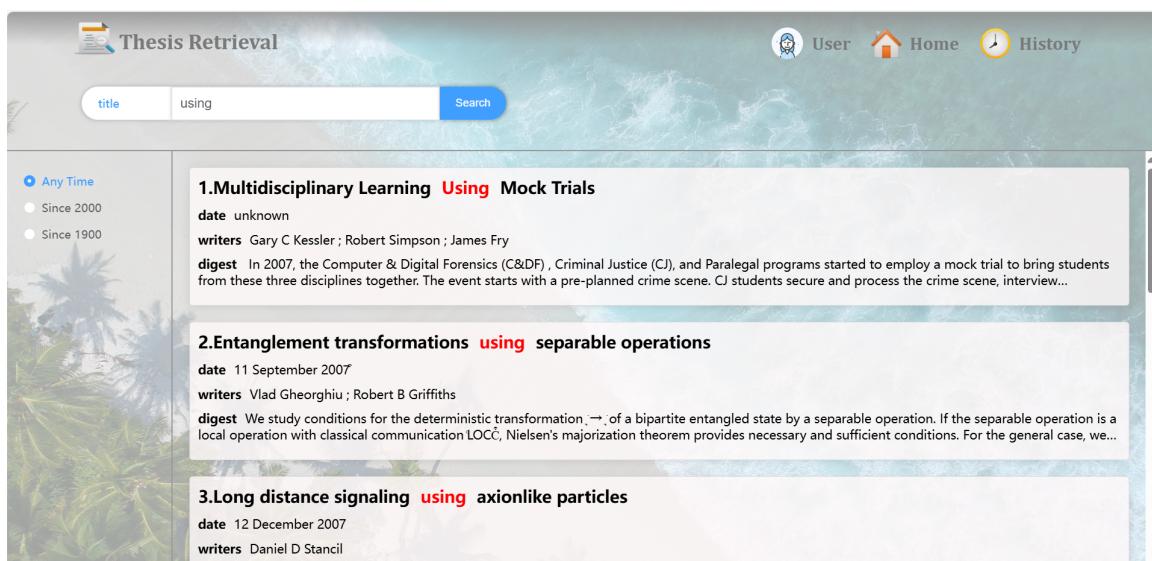


图 10: 搜索页设计

### 6.3.4 详情页

详情页中可以看到一篇文章的具体信息，并且可以点击看到原文，另外可以进行收藏（如果已经登录的情况下），如图11。

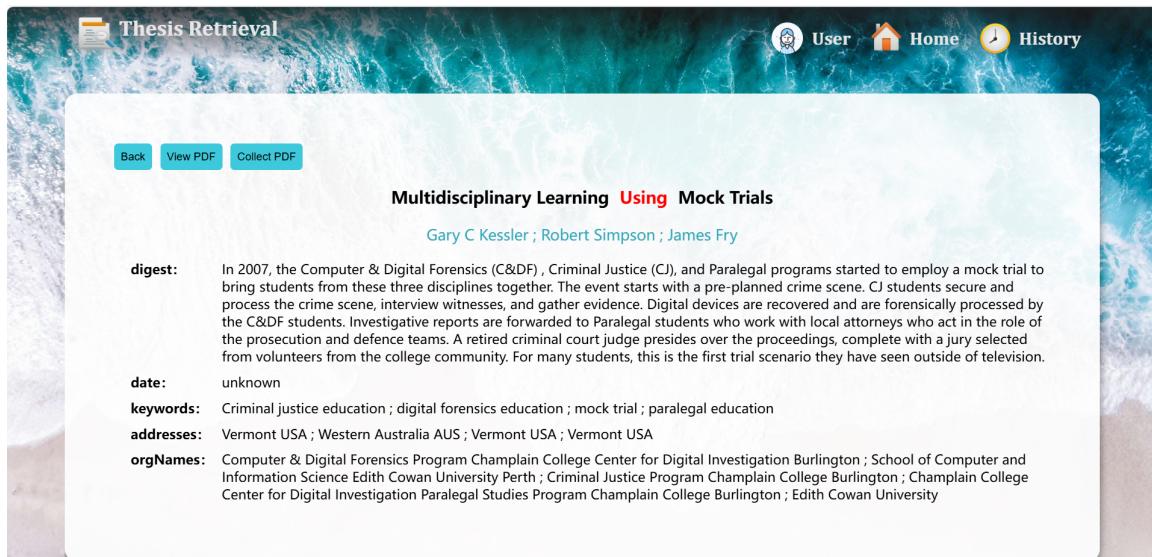


图 11: 详情页设计

### 6.3.5 用户页面

用户页面分为管理员页面和普通用户页面，其点击左侧的选项，就会在右边显示出不同的功能，普通用户如图12、管理员如图13。

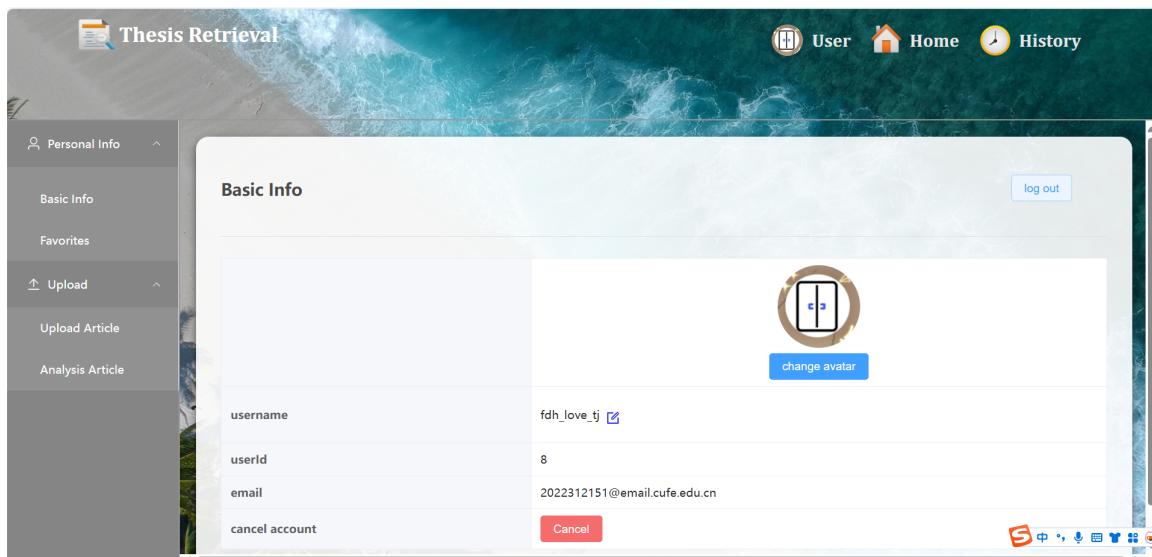


图 12: 用户页设计

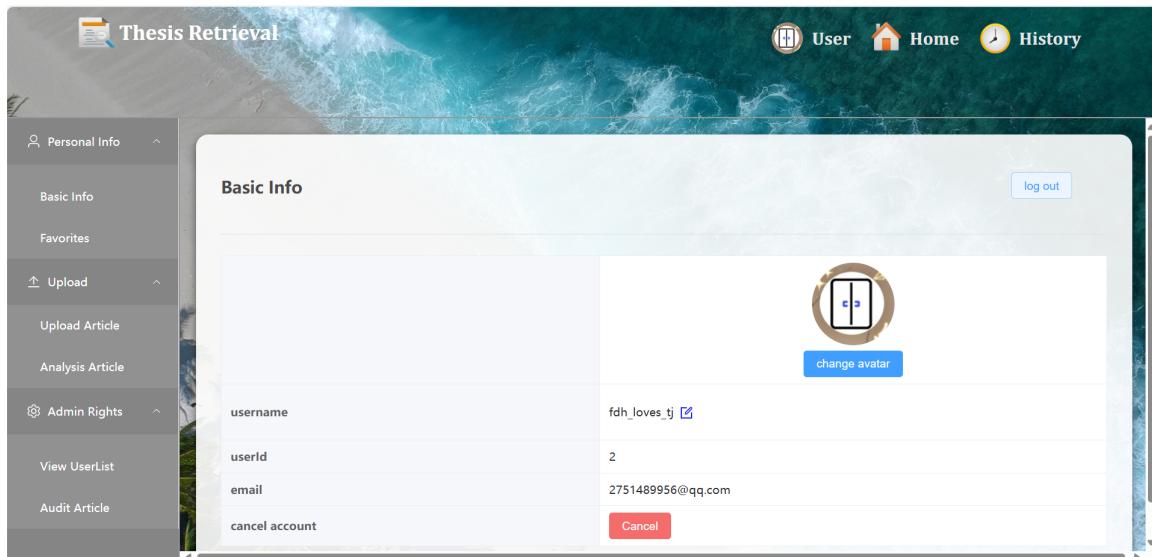


图 13: 管理员页设计

## 7 测试

测试用例及结果如下：

测试接口	测试用例	校验点	预期结果	实际结果
上传文献	1. 上传 pdf 文件且大小在 15MB 以下 2. 上传其他类型文件 3. 上传大于 15MB 的文件	上传文件的校验	1. 上传成功 2. 提示“只能上传 pdf 文件” 3. 提示“文件过大”	预期结果均实现
验证验证码	1. 输入正确的邮箱和验证码 2. 输入不存在的邮箱 3. 输入错误的验证码	验证验证码功能	1. 验证通过 2. 提示“邮箱不存在” 3. 提示“验证码错误”	预期结果均实现
修改待审核文章状态	1. 点击查看按钮 2. 点击通过按钮 3. 点击未通过按钮	修改待审核文章状态的校验	1. 查看文章原文 2. 文章状态由待审核转变为已通过 3. 文章状态由待审核转变为未通过	预期结果均实现
删除一条待审核的记录	1. 点击通过按钮以改变文章状态 2. 点击未通过按钮以改变文章状态	删除一条待审核记录的校验	1. 文章从待审核列表中删除 2. 文章从待审核列表中删除	预期结果均实现
获取全部待审核的记录	点击菜单栏中的“查看待审核列表”	查看全部待审核记录的校验	分页显示全部待审核的记录	预期结果均实现

测试接口	测试用例	校验点	预期结果	实际结果
获取用户信息	点击菜单栏中的“用户列表”页	获取用户信息的校验	分页显示全部用户信息	预期结果均实现
获取搜索结果	1. 未输入检索文本点击搜索按钮 2. 输入检索文本后点击搜索按钮 3. 通过时间筛选进行搜索 4. 通过选择检索领域进行搜索 5. 点击分页组件中的其他页 6. 搜索数据库中不存在的论文	获取搜索结果的校验	1. 提示“请在输入后搜索” 2. 分页显示搜索结果列表 3. 分页显示时间符合及未标明时间的搜索结果 4. 分页显示该检索领域下的论文列表 5. 显示点击页的搜索结果 6. 显示空状态页面	预期结果均实现
获取输入建议	1. 点击输入框后未输入文本 2. 点击输入框后输入文本	获取输入建议的校验	1. 输入框下方弹出正在加载中的建议模块 2. 输入框下方弹出与输入文本相关的输入建议	预期结果均实现
随便看看	点击更换按钮并查看任意论文	验证随便看看功能	成功跳转至详情页	预期结果均实现
收藏一篇文献	1. 未登录时收藏 2. 登录后再收藏	收藏一篇文献的校验	1. 提示“先登录再收藏” 2. 收藏成功，默认收藏到“所有”文件夹	预期结果均实现
增加一个收藏分类	1. 增加一个新分类 2. 增加已有的分类	增加一个收藏分类的校验	1. 收藏成功 2. 提示“该分类已存在，请重新输入”	预期结果均实现
添加浏览记录	1. 未登录时查看记录 2. 登录后查看记录	添加浏览记录的校验	1. 提示“登录后才能查看” 2. 添加成功，在“随便看看”或搜索页或浏览记录中都有记录更新	预期结果均实现
解析文章	1. 上传 pdf 文件且大小在 15MB 以下 2. 上传其他类型文件 3. 上传大于 15MB 的文件	验证解析文章功能	1. 解析成功，返回 word 文档 2. 提示“只能上传 pdf 文件” 3. 提示“文件过大”	预期结果均实现
取消收藏一篇文章	点击取消收藏	取消收藏一篇文章的校验	取消成功	预期结果均实现

测试接口	测试用例	校验点	预期结果	实际结果
获取全部收藏分类	1. 已登录时获取全部收藏分类 2. 未登录时尝试获取全部收藏分类 3. 返回的收藏分类列表是否按预期显示所有分类	获取全部分类收藏的校验	1. 成功获取全部收藏分类 2. 提示“先登录后再获取” 3. 收藏分类列表按照预期显示	预期结果均实现
获取某一分类下收藏的文章	1. 已登录且指定分类存在时获取该分类下的文章 2. 已登录但指定分类不存在时获取文章 3. 验证返回的文章是否正确属于某一分类。	获取某一分类下收藏的文章的校验	1. 成功获取指定分类下的文章列表 2. 提示“分类不存在” 3. 文章属于该分类	预期结果均实现
获取浏览记录	1. 已登录时获取浏览记录 2. 未登录时获取浏览记录 3. 返回的浏览记录是否包含正确的文章标题和日期	获取浏览记录的校验	1. 成功获取浏览记录 2. 提示登录后才能查看 3. 浏览记录包含正确的文章标题和日期	预期结果均实现
获取上传记录	1. 已登录时获取上传记录 2. 未登录时获取上传记录 3. 返回的上传记录是否包含正确的文件名和上传时间	获取上传记录的校验	1. 成功获取上传记录 2. 提示登录后才能查看 3. 上传记录包含正确的文件名和上传日期	预期结果均实现
登录	1. 输入正确的邮箱和验证码进行登录 2. 输入错误的邮箱和验证码尝试登录 3. 输入不存在的邮箱和任意验证码进行登录	登录的校验	1. 成功登录并跳转到用户主页 2. 提示登录失败，请检查邮箱或验证码是否输入正确 3. 提示邮箱不存在	预期结果均实现
注册	1. 使用有效的邮箱和验证码进行注册 2. 使用已存在的邮箱进行注册 3. 使用无效的邮箱格式进行注册 4. 填写信息不完整	注册的校验	1. 成功注册新用户并自动登录到用户主页 2. 提示邮箱已被注册 3. 提示邮箱格式不正确 4. 提示请完整填写信息	预期结果均实现
更换头像	1. 已登录时更换头像 2. 未登陆时等换头像	更换头像的校验	1. 成功更换头像 2. 提示登录后才能更换头像	预期结果均实现

测试接口	测试用例	校验点	预期结果	实际结果
修改密码	1. 输入正确的原密码进行修改密码 2. 输入错误的原密码进行修改密码 3. 修改后的新密码与原密码相同	修改密码的校验	1. 成功修改密码 2. 提示输入的原密码错误 3. 提示修改后密码不能与原密码相同	预期结果均实现
修改用户名	1. 输入未被占用的用户名进行更改 2. 输入已被占用的用户名进行更改 3. 输入含违规字符的名称	修改用户名的校验	1. 成功修改用户名 2. 提示该用户名已被占用 3. 提示该用户名含有违规字符	预期结果均实现
删除某一分类的全部收藏	点击删除收藏	删除某一分类的全部收藏的校验	成功删除	预期结果均实现
删除浏览记录	1. 选择单条浏览记录删除 2. 选择多条浏览记录删除 3. 选择所有浏览记录删除	删除浏览记录的校验	1. 成功删除单条浏览记录 2. 成功删除多条浏览记录 3. 成功删除所有浏览记录	预期结果均实现
发送邮件	1. 用户输入邮箱进行登录 2. 用户输入邮箱进行注册 3. 用户输入不存在的邮箱	发送邮件的校验	1. 系统发送登录验证码邮件 2. 系统发送注册验证码邮件 3. 提示邮箱不存在	预期结果均实现

表 47: 测试接口用例表

## 8 附录

### 8.1 论文的获取

我们从下面图14的这个公开数据集中，利用里面的 doi，下载论文的原始 pdf 格式的文件。

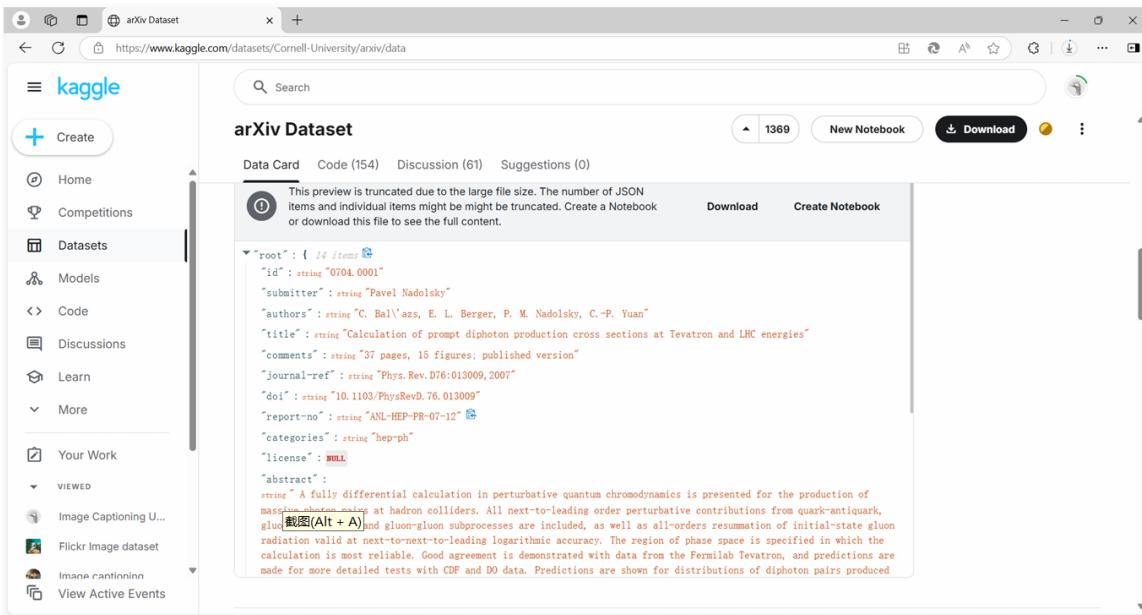


图 14: 论文数据来源

最终我们得到了 4000 余篇原始论文文件。

## 8.2 论文的文本提取

### 8.2.1 使用 Grobid 处理 pdf 论文

我们在虚拟机中使用 Grobid 处理 pdf 论文，由于在 Grobid 中处理 pdf 文件较慢，我们编写了批量处理脚本，对 pdf 论文进行批量处理。

```
#!/bin/bash
# 目标文件夹
input_folder="/home/dehang-fu/oriPDFs"
# 输出的文件夹，用于存放处理后的XML文件
output_folder="/media/sf_shareWithUbuntu/oriPDFsXML"

# 确保输出文件夹存在
mkdir -p "$output_folder"

# 遍历目标文件夹中的所有PDF文件
for pdf_file in "$input_folder"/*.pdf; do
    # 构建输入文件的完整路径
    input_file="$pdf_file"
    # 构建输出文件的完整路径
    output_file="$output_folder/${basename $pdf_file}.xml"

    echo "Processing $input_file -> $output_file"
    # 执行curl命令并将输出重定向到对应的XML文件
    curl -v -H "Accept: application/xml" --form input=@$input_file localhost:8070/api/processFulltextDocument > "$output_file"

    # 检查curl命令是否成功执行
    if [ $? -eq 0 ]; then
        echo "Successfully processed $pdf_file"
    else
        echo "Failed to process $pdf_file"
    fi
done
```

图 15: 批量处理脚本

### 8.2.2 批量处理结果

使用脚本对 pdf 论文进行处理后，得到 xml 文件格式，文件格式如下图所示。

```

<textClass>
  <keywords>
    <term>Criminal justice education</term>
    <term>digital forensics education</term>
    <term>mock trial</term>
    <term>paralegal education</term>
  </keywords>
  <textClass>
  <abstract>
  <div>
    In 2007, the Computer & Digital Forensics (C&DF)
    <p>
      , Criminal Justice (CJ), and Paralegal programs started to employ a mock trial to bring students from these three disciplines together. The event starts with a pre-planned crime scene. CJ students secure and process the crime scene, interview witnesses, and gather evidence. Digital devices are recovered and are forensically processed by the C&DF students. Investigative reports are forwarded to Paralegal students who work with local attorneys who act in the role of the prosecution and defence teams. A retired criminal court judge presides over the proceedings, complete with a jury selected from volunteers from the college community. For many students, this is the first trial scenario they have seen outside of television.
    </p>
  </div>
  </abstract>
  </profileDesc>
  <teilHeader>
  <text xml:lang="en">
    <body>
      <div>
        <head n="1">Introduction</head>
        <p>
          Champlain College started an undergraduate degree program in Computer & Digital Forensics (C&DF) in 2003. Recognizing that digital forensics is a multidisciplinary field of study, the curriculum provides students with a good grounding in computer technology, networking, and criminal justice in addition to fundamental computer forensics and digital investigation courses
          <ref type="bibl" target="#fo1">Kessler and Schirling, 2006</ref>
          . Digital forensics education requires a high degree of hands-on, interactive activities, which are enhanced by courses where C&DF students take courses with peers in other disciplines, such as Criminal Justice (CJ) and information technology programs.
        </p>
        <p>
          It is common in the public sector for the criminal investigator to identify potentially relevant digital devices and turn those exhibits over to the computer forensics team, so that the investigator's next contact with the digital part of the case
        </p>
      </div>
    </body>
  </text>
  </teilHeader>
</textClass>

```

图 16: xml 文件格式

### 8.2.3 解析文件

为方便后续索引建立操作，我们使用 python 将 xml 文件解析并保存为 json 格式。比起 xml 文件，json 格式在进行查询处理和索引建立更具有灵活性。

```

{
  "writers": [
    "Robert Simpson",
    "James Fry"
  ],
  "addresses": [
    "Vermont\tUSA",
    "Western Australia AUS",
    "Vermont\tUSA",
    "Vermont\tUSA",
    ""
  ],
  "orgNames": [
    "Computer & Digital Forensics Program Champlain College Center for School of Computer and Information Science\tEdith Cowan University",
    "Criminal Justice Program Champlain College Burlington",
    "Champlain College Center for Digital Investigation\tParalegal Stud",
    "Edith Cowan University"
  ],
  "keywords": [
    "Criminal justice education",
    "digital forensics education",
    "mock trial",
    "paralegal education"
  ],
  "fulltext": "Introduction\nChamplain College started an undergraduate d"
}

```

图 17: 解析文件为 json 格式

### 8.2.4 数据清洗

考虑到会有解析错误、同一篇文章下载多次的情况，所以我们对数据进行了清洗，最终得到 3000 余篇文章。

## 8.3 索引建立以及检索

### 8.3.1 原始文本处理

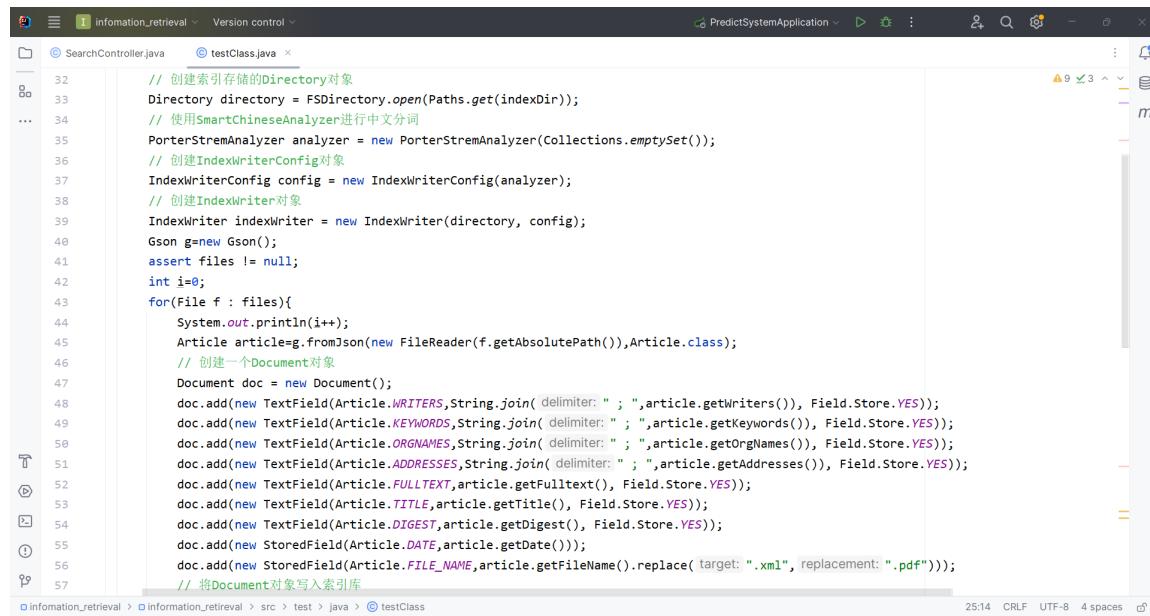
为了提高我们的检索的准确性和泛化能力，我们需要在建立索引之前，对文本进行一定的处理，比如将文本转化为小写字母，将单词归一化，将单词复数等形式变为原型表示等等，具体而言，我们将单词变为小写表示以后，使用 porter 算法提取单词的词根。效果如图18所示：

```
in 2007 the comput digit forens c df crimin justic cj and paraleg program start to emploi a mock
In 2007, the Computer & Digital Forensics (C&DF)
,Criminal Justice (CJ), and Paralegal programs started to employ a mock trial to bring students
```

图 18: 文本处理结果 (上面为处理后文本，下面为原始文本)

### 8.3.2 索引建立

由于我们需要对不同的内容进行检索，所以我们在建立索引的时候，也需要分不同的内容建立索引，比如作者、全文、摘要等，我们使用的为 Luncene，作为我们检索的工具，建立索引的过程如下图19所示：



The screenshot shows an IDE interface with the following details:

- Title Bar:** PredictSystemApplication
- File Explorer:** Shows two files: SearchController.java and testClass.java.
- Code Editor:** Displays Java code for creating an IndexWriter and adding documents. The code uses various Apache Lucene classes like Directory, PorterStemAnalyzer, IndexWriterConfig, and Document.
- Status Bar:** Shows the file path infomation\_retrieval/src/test/java/testClass.java, and system information like 25:14, CRLF, UTF-8, 4 spaces.

```

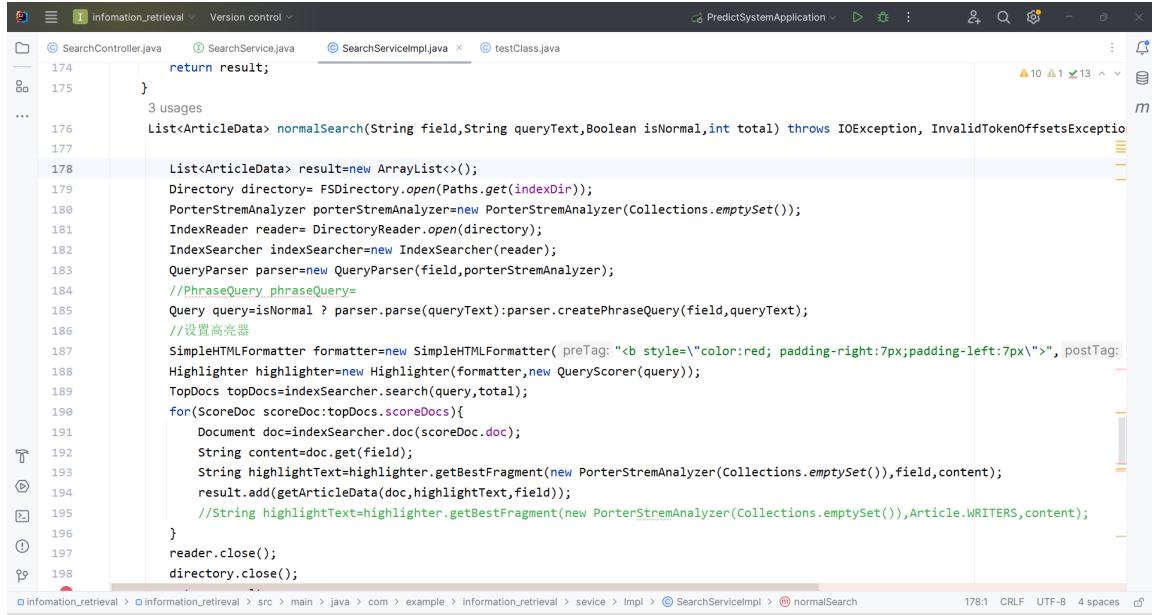
32     // 创建索引存储的Directory对象
33     Directory directory = FSDirectory.open(Paths.get(indexDir));
34     // 使用SmartChineseAnalyzer进行中文分词
35     PorterStemAnalyzer analyzer = new PorterStemAnalyzer(Collections.emptySet());
36     // 创建IndexWriterConfig对象
37     IndexWriterConfig config = new IndexWriterConfig(analyzer);
38     // 创建IndexWriter对象
39     IndexWriter indexWriter = new IndexWriter(directory, config);
40     Gson g=new Gson();
41     assert files != null;
42     int i=0;
43     for(File f : files){
44         System.out.println(i++);
45         Article article=g.fromJson(new FileReader(f.getAbsolutePath()),Article.class);
46         // 创建一个Document对象
47         Document doc = new Document();
48         doc.add(new TextField(Article.WRITERS, String.join(" ", article.getWriters()), Field.Store.YES));
49         doc.add(new TextField(Article.KEYWORDS, String.join(" ", article.getKeywords()), Field.Store.YES));
50         doc.add(new TextField(Article.ORGNAMES, String.join(" ", article.getOrgNames()), Field.Store.YES));
51         doc.add(new TextField(Article.ADDRESSES, String.join(" ", article.getAddresses()), Field.Store.YES));
52         doc.add(new TextField(Article.FULLTEXT, article.getFulltext(), Field.Store.YES));
53         doc.add(new TextField(Article.TITLE, article.getTitle(), Field.Store.YES));
54         doc.add(new TextField(Article.DIGEST, article.getDigest(), Field.Store.YES));
55         doc.add(new StoredField(Article.DATE, article.getDate()));
56         doc.add(new StoredField(Article.FILE_NAME, article.getFileName().replace(target, ".pdf")));
57     // 将Document对象写入索引库
    }
  
```

图 19: 索引创建

从图中可以看到，我们为每一个不同的领域分别建立起了索引，这样便可以通过指定检索不同的领域进行不同的检索。

### 8.3.3 检索

我们提供了多种检索功能，比如模糊匹配、短语查询等，其实现都依托于所建立的索引以及 Lucene 所提供的功能，以短语查询为例，如图20：



```

174     return result;
175 }
...
176     3 usages
177
178     List<ArticleData> result=new ArrayList<>();
179     Directory directory= FSDirectory.open(Paths.get(indexDir));
180     PorterStemAnalyzer porterStemAnalyzer=new PorterStemAnalyzer(Collections.emptySet());
181     IndexReader reader= DirectoryReader.open(directory);
182     IndexSearcher indexSearcher=new IndexSearcher(reader);
183     QueryParser parser=new QueryParser(field,porterStemAnalyzer);
184     //PhraseQuery phraseQuery=
185     Query query=isNormal ? parser.parse(queryText):parser.createPhraseQuery(field,queryText);
186     //设置高亮器
187     SimpleHTMLFormatter formatter=new SimpleHTMLFormatter( preTag: "<b style=\"color:red; padding-right:7px;padding-left:7px\>", postTag:
188     Highlighter highlighter=new Highlighter(formatter,new QueryScorer(query));
189     TopDocs topDocs=indexSearcher.search(query,total);
190     for(ScoreDoc scoreDoc:topDocs.scoreDocs){
191         Document doc=indexSearcher.doc(scoreDoc.doc);
192         String content=doc.get(field);
193         String highlightText=highlighter.getBestFragment(new PorterStemAnalyzer(Collections.emptySet()),field,content);
194         result.add(getArticleData(doc,highlightText,field));
195         //String highlightText=highlighter.getBestFragment(new PorterStemAnalyzer(Collections.emptySet()),Article.WRITERS,content);
196     }
197     reader.close();
198     directory.close();

```

图 20: 查询示例