|  |
| --- |
| **学士学位论文** |
| **基于Web的个人科研助理系统** |
| |  |  | | --- | --- | | 学 号： | 20141002562 | | 姓　 名： | 王琦 | | 学 科 专 业： | 网络工程 | | 指 导 教 师： | 曾德泽 副教授 | | 培 养 单 位： | 计算机学院 | |
| 二○一八年六月 |

中国地质大学（武汉）学士学位论文原创性声明

本人郑重声明：本人所呈交的学士学位论文《基于Web的个人科研助理系统》，是本人在指导老师的指导下，在中国地质大学（武汉）攻读学士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果，对论文的完成提供过帮助的有关人员已在文中说明并致以谢意。

本人所呈交的学士学位论文没有违反学术道德和学术规范，没有侵权行为，并愿意承担由此而产生的法律责任和法律后果。

学位论文作者签名：

日 期： 年 月 日

**摘要**

高校科研人员作为科学技术创新的骨干力量，随着国家对科学研究的日益重视，高校承担的科研项目数量和经费在明显增加、高校科研人员的数量不断增加以及科研规模在不断扩大，使得科研信息量不断的增长。如果科研工作成果得不到有效的管理，那么科研人员将会花费大量的时间和精力在整理和收集个人科研成果上，这在一定程度会影响科研人员的科研能力和科研成果。本文旨在帮助科研人员合理地个人科研文献并利用文献快速、便捷地生成论文报表。目前有诸多文献管理工具如EndNote、JebRef、Mendeley等文献管理软件管理自己的科研文献，他们允许科研人员处理文献信息检索、收集、管理、引用等环节，但都没有注意到科研人员在填写种类繁多而复杂的申报书时遇到的烦恼：需要根据申报书中论文报表的不同格式，分别取收集数据、填写数据，这样做不仅效率极其底下，并且还很容易出错。因此，为科研人员提供一个既能科学地管理自己科研文献，又能依据科研文献按照用户指定的格式生成论文报表的工具是非常必要的。

如EndNote、JebRef、Mendeley都需要在不同的操作系统上安装基于不同系统的客户端，这样在每次进行系统更新和维护时都需要花费很大的代价，而如今iPhone、Android、各类Pad等智能设备已经从大众休闲娱乐的设备逐渐变为日常办公工作中必不可少的工具，因此开发一款能够跨平台，兼容性和操作性强的系统就显得十分必要。因此文本采用了Web进行系统开发，在满足用户需求的同时，还能为用户提供良好的适配性和交互性，以此来帮助科研人员开展日工常的科研管理工作。

本文通过对科研助理系统的功能和流程分析，提出了系统开发的方案和思路。本文在收集个人文献资料时采用科研管理工作常用的BibTeX格式的文件（该文件可以从很多文献资料库中下载），科研人员通过BibTeX文件可以获取个人文献的关键信息，如文章类型、标题、作者等等。通过BibTeX文件可以生成个人参考文献列表，以及通过科研人员上传的个人资料形成一份科研人员的个人主页，方便科研人员之间相互了解。同时还可以依据BibTeX文件由用户选择期刊或是会议文章，按照指定格式生成论文报表，并依据报表表头进行多关键词的排序。

本文在技术方面，主要以B/S架构作为开发的基础，使用Python Flask搭档MySQL作为Web服务器，使用HTML、JavaScript和CSS配合开发Web前端，并使用JQuery在服务器和客户端之间交换JSON数据，构建了包括注册、登录、上传个人资料和文章信息、生成个人主页、上传和下载PDF、生成论文报表以及搜索科研人员等功能的一个完整的基于Web的个人科研系统。

**关键词：**科研助理；论文报表；B/S；Flask Web服务器；JQuery；BibTeX；

**Abstract**

University researchers as the backbone of science and technology innovation, as the country is becoming more and more attention to scientific research, colleges and universities bear the number of research projects and funds in a significant increase, increasing the number of researchers in colleges and universities and scientific research in expanding scale, makes scientific research information unceasing growth. If the research work can’t get effective management, researchers will spend a lot of time and energy on sorting and collecting personal scientific research achievements, this to a certain extent affect the scientific research ability and scientific research achievements. The purpose of this paper is to help the researchers to produce the paper reports quickly and conveniently. At present, there are many literature management tools such as EndNote, JebRef and Mendeley to manage their own scientific research literature, they allow researchers to process information retrieval, collection, management and reference, but all of them didn’t notice the trouble the researchers had in filling out a variety of complicated declarations: we often have to collect data and fill in the data according to the different formats of the report forms in the declaration, which is not only extremely efficient, but also easy to make mistakes. Therefore, it is very necessary to provide a tool for researchers to manage their own scientific research literatures scientifically, and to generate paper reports in the format specified by the user.

For example, EndNote, JebRef and Mendeley all need to install different system based clients on different operating systems, this will cost a lot each time the system is updated and maintained. Now the iPhone, Android, all kinds of Pad, etc smart devices from public leisure entertainment equipment gradually become an essential tool in daily office work, thus can develop a cross-platform, strong compatibility and interoperability of systems is very necessary. Therefore, the text adopts the Web for system development. While satisfying the user's needs, it can also provide users with good adaptability and interactivity to help researchers conduct routine scientific research and management work.

In this paper, through the analysis of the function and process of the scientific research assistant system, the system development scheme and ideas are proposed. . This article uses BibTeX format files commonly used for scientific research management work when collecting personal literature (this file can be downloaded from many literature databases). The bibTeX file enables researchers to obtain key information about personal documents, such as article type, title, and author and many more. BibTeX files can be used to generate personal reference lists, as well as personal data uploaded by researchers to form a personal page for researchers to facilitate mutual understanding among researchers. At the same time, users can select journals or conference articles according to BibTeX files, generate paper reports according to the specified format, and sort according to the report header.

In terms of technology, this article mainly uses the B/S architecture as the basis for development. It uses Python Flask partner MySQL as a Web server, uses HTML, JavaScript, and CSS to develop a Web front end, and uses JQuery to exchange JSON data between the server and the client. A complete Web-based personal scientific research system including registration, login, uploading of personal data and article information, generating personal homepages, uploading and downloading PDFs, generating paper reports, and searching for scientific personnel has been constructed.

**Key Words**: scientific research assistant; paper report; Flask Web server; B/S; JQuery;BibTeX;

**目录**

[第一章 绪论 1](#_Toc516651762)

[1.1课题的背景和意义 1](#_Toc516651763)

[1.2 论文构成和工作内容 2](#_Toc516651764)

[1.2.1论文构成 2](#_Toc516651765)

[1.2.2工作内容 2](#_Toc516651766)

[第二章 系统分析与设计 4](#_Toc516651767)

[2.1 系统结构的分析与设计 4](#_Toc516651768)

[2.2 系统开发框架分析 5](#_Toc516651769)

[2.2.1 服务器端 5](#_Toc516651770)

[2.2.2 浏览器端 5](#_Toc516651771)

[2.3 功能需求分析 7](#_Toc516651772)

[2.4 系统数据库设计 8](#_Toc516651773)

[2.5 业务流程分析 10](#_Toc516651774)

[第三章 系统实现 12](#_Toc516651775)

[3.1 用户注册、登录、登出模块 12](#_Toc516651776)

[3.1.1 使用Flask-Login实现用户登录、登出 12](#_Toc516651777)

[3.1.2 注册、登录、登出详细实现 13](#_Toc516651778)

[3.2 创建及修改个人信息模块 14](#_Toc516651779)

[3.2.1 创建个人信息 14](#_Toc516651780)

[3.2.2 BibTeX文件的解析 15](#_Toc516651781)

[3.2.3 修改个人信息 16](#_Toc516651782)

[3.3 上传和下载PDF模块 17](#_Toc516651783)

[3.3.1 上传PDF 17](#_Toc516651784)

[3.3.2 下载PDF 18](#_Toc516651785)

[3.4 生成个人主页模块 18](#_Toc516651786)

[3.5 生成论文报表模块 18](#_Toc516651787)

[3.6 搜索科研人员模块 20](#_Toc516651788)

[第四章 系统运行 22](#_Toc516651789)

[第五章 总结与展望 33](#_Toc516651790)

[5.1 全文总结 33](#_Toc516651791)

[5.2 后续展望 33](#_Toc516651792)

[致谢 34](#_Toc516651793)

[参考文献 35](#_Toc516651794)

第一章 绪论

1.1课题的背景和意义

《“十三五”国家科技创新规划》指出科技创新的总体目标是国家科技实力和创新能力大幅跃升[1]，而高等学校是知识创新、知识传播、知识物化的重要基地[2]。据中华人民共和国教育部的数据，截至2017年5月31日，全国高等学校共计2914所，而据首份国家高等教育质量报告《中国高等教育质量报告》指出，中国高校师资队伍持续壮大，层次结构不断优化，从1999年到2014年，普通高校专任教师从42.6万人增加到153.5万人（净增110.9万），全国整体教师队伍发展潜力大、后劲足[3]。

随着国家对科学研究的日益重视，高校承担的科研项目数量和经费明显增加、高校科研人员的数量不断增加以及科研规模不断的扩大，使得科研信息量不断的增长[4]。为了加强科研信息的管理、提高科研信息管理的质量、降低科研管理的成本，目前已经存在一些科研管理系统，例如：易普拉格科技，学术格子等系统，常见的科研信息管理一般包括科研项目管理、科研经费管理、科研成果管理、科研研发管理、科研评审管理等等。这些系统解决了高校科研人员的部分需求，但是根据科研文章生成所需要的论文报表的功能却存在一个空白区。

高校科研人员经常需要填写各种项目申报书、评奖评优的申请表、成果汇报表等Word文档，在这些文档中会依据自己的科研论文（包括期刊和会议文章）来填写表格（以下统称为论文报表），不同申报书会有不同格式的论文报表表头。在填写论文报表时，会依据表头在科研论文中进行信息收集、统计、填写等操作，这样的方式有很多的缺点：效率低、容易出错、填写工作量庞大等。

科研人员常用如EndNote、JebRef、Mendeley等文献管理软件对科技文献进行查阅、整理和分析。EndNote专门用于管理参考文献，也是现在最有名、历史最久的文献管理软件之一,它能够很好地和WOK平台上的数据库相关联，通过Internet将PubMed检索结果直接保存到数据库中或者读入各种格式的Medline检索结果[5]。通过插件方便地在Word中插入所引用的文献，并能生成指定格式的参考书目列表。JabRef是一款以BibTeX为原生格式的开源文献管理软件，基于Java编写，其最大特点是对BibTeX格式的充分兼容，最适合LaTeX用户使用[6]。Mendeley 是一款基于Qt平台开发的跨平台文献管理软件，它不仅实现了较好的网页文献信息抓取和PDF文献信息抓取功能，还提供了一个学术社交平台，可以让科研人员进行更广泛、更便捷的学术交流[6]。常用的这几款文献管理软件均逐步地完善了其在处理文献信息检索、收集、管理、引用等环节的表现，虽然功能各有偏重和特色，也都实现了文献管理软件的基本功能。但是依据科研文献迅速生成论文报表的功能这些软件却都没有涉及到。

为了提高科研人员的工作效率，节约填写表格时间，本文为高校科研人员提供一个能够跨平台以BibTeX格式进行个人文献的收集、管理，同时以指定格式生成参考书目列表，并能快速方便地生成指定格式的论文报表的工具，解决科研人员填写表格时的烦恼，将更多的精力投入到科研工作中。

1.2 论文构成和工作内容

1.2.1论文构成

本文旨在介绍基于Web的科研助理系统的设计与实现的过程。在绪论中主要介绍本系统的背景和实现该系统的意义，以及介绍了本文的主要结构和本人在实现此系统中的具体工作内容。第二章主要从系统分析入手，分别对系统的架构、使用的框架、系统功能分析、数据库设计和系统的功能流程等方面详细地介绍了系统。第三章则按照不同的功能模块对系统的实现进行了详细的介绍。第四章主要展示了系统所有功能运行结果。第五章总结了文章的工作并对未来提供侧展望。最后是感谢支持本人完成本系统的家人、老师和同学们。

1.2.2工作内容

我在本次系统的设计与实现中，首先依照系统背景和研究现状对本系统的可行性和创新点进行了分析，证明了实现本系统的可行性和必要性。然后依照科研人员的需求对系统的架构进行了分析，最终决定采用B/S架构进行开发。并分析了为什么使用BibTeX格式的文件进行科研人员文献的收集和整理工作。按照系统的功能需求，做了以下工作：

基于Web进行开发，设计和实现个人科研助理系统，实现对科研信息的管理。该个人科研助理系统主要以Web应用开发为核心，服务器端使用Python Flask和MySQL进行开发，Web前端使用HTML和JavaScript进行开发，主要实现的功能是：

1）允许科研人员上传个人基本信息，如姓名、工作单位、Email、科研兴趣、教育背景、科研项目等；

2）能够允许科研人员通过BibTeX文件格式，上传个人著作信息，同时能够允许上传PDF文件；

3）能够依据科研人员上传和填写的信息生成科研人员的个人主页；

4）能够按照科研人员指定的格式生成论文报表。

个人科研助理系统是一个大型的工程，以上所示的四个主要功能也只是整个助理系统的一部分，想要达到工业可用水准还需要进一步的开发。不过本系统提供的数据基础和框架，在它的基础上可以进一步扩展功能。

第二章 系统分析与设计

2.1 系统结构的分析与设计

本系统希望能够不受任何操作系统和平台的影响，也不希望用户安装任何多余的客户端而在广域网上就能访问，因此本系统采用B/S（Browser/Server，浏览器/服务器模式）模型基本架构，是一种常见的客户端和服务器进行交互的模型，它是以Web技术为基础的系统平台模型，而不采用传统的C/S（Client/Server）架构。C/S架构适用于规模较小、用户较少的环境下运行，由于其程序开发量大，系统维护困难，客户端需要单独安装在系统等原因，使用Internet使用越来越广泛的B/S架构，它在很大程度上简化了客户端的工作，客户端上只需要有浏览器，彻底解放客户端，即可连接服务器，由Web服务器完成对数据库的访问等工作[7]-[8][9]。

B/S将C/S结构中的服务器部分分解为数据库服务器和Web服务器，从而形成三层体系结构的客户机/服务器体系[10]。下面将分别介绍这三层结构：

第一层是表示层，主要是指Wed浏览器，主要包含系统的显示逻辑，为用户提供友好的界面，能够由Web浏览器向Web服务器提出请求，Web服务器作出响应。

第二层是业务层，主要是指Web服务器接受浏览器的请求，根据用户的请求，进行处理，对数据库进行读写操作，然后Web服务器把结果返回给浏览器。

第三层是数据层，数据库服务器，主要负责数据的存储功能。它能够接受Web服务器对数据库的操作请求，实现对数据的查询、修改、删除等功能，并将结果返回给Web服务器，便于Web服务器进行处理。下图2.1为该系统的分层结构图[11]-[12]。

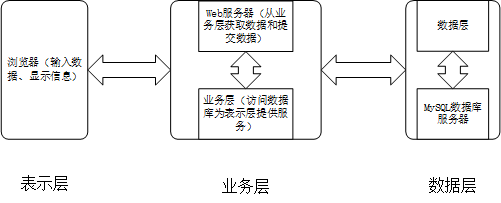


图2.1 系统分层结构图

2.2 系统开发框架分析

2.2.1 服务器端

在开发后台Web服务器时，本系统需要选择一款简单、易开发、依赖少并且能够在多种操作系统下运行的Web服务器框架，在基于Python开发的Web框架里有诸如Flask、Pyramid和Django等的微框架Web服务。选一款适合本次系统开发的框架及其重要。Pyramid和Django虽然都是面向大型的应用，Pyramid关注灵活性，让开发者选择合适的工具来开发项目，也就是说可以自主选择数据库，URL结构，模板风格等等，而Django的目标则是为web应用开发提供一站式解决方案，灵活和自由度不够高。而Flask是一个微框架，主要面向需求简单的小应用，非常适用于小型网站的开发，它并不原生支持数据库访问、Web 表单验证和用户认证等高级功能，这些功能以及其他大多数 Web 程序中需要的核心服务都以扩展的形式实现，因此Flask的自由度和灵活度都足够高，可扩展性也非常强。因此这里本系统开发时选用“轻”、性能更优、入门简单的Flask框架。

数据层则采用关系型数据库MySQL，在 Web 应用方面，MySQL非常适用。由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](https://baike.baidu.com/item/开放源码#_blank)这一特点，一般中小型网站的开发都选择MySQL作为网站数据库[13]。它支持多种操作系统，并为多种语言提供了API，使用标准的 [SQL](https://baike.baidu.com/item/SQL#_blank)数据语言形式，优化的 [SQL](https://baike.baidu.com/item/SQL#_blank)查询算法，有效地提高查询速度。在该系统中Flask Web服务器使用Python语言通过MySQLdb连接MySQL数据库[14]，存储用户信息等数据，方便存储、更新、删除与查询操作，使服务器的读写数据库的操作更加迅速。

2.2.2 浏览器端

为了能向用户提供一个友好的用户界面，本系统在前端的界面设计需要遵循以下的原则，以符合用户的人机交换、逻辑操作、界面美观的整体需求：

（1）简易性

界面的简易性可以帮助科研人员方便使用、了解本系统，减少在使用过程中让科研人员发生错误选择的可能性[15]。

（2）一致性

界面的结构保持清晰且一致，风格与系统内容个人科研助理相一致。本系统中会开发多个不同功能界面，要使各个界面之间的风格与类型相似，保持页面中文字字体与颜色一致，避免一套主题出现多个字体，采用相同的布局方式与对齐方式[16]。

（3）交互性

界面要始终保持能够和用户沟通，不管用户的操作是否正确，都要能够随时提供反馈，告诉用户的行为：状态更改、出现错误或者异常信息等。使用视觉提示或者简单的文字提醒用户其行为是否能够达到预期的效果[17]。

（4）布局合理

设计界面时要求页面布局合理，遵循用户的浏览和操作习惯，避免常用功能事件排列分散，造成用户点击复杂的现象。

前端开发的好坏，会直接影响用户使用和交互的效果，本系统为了能够更加生动，并且能够提供多种交互效果，需要是使用到基本的Web前端的开发技术，其中包括HTML、CSS和JavaScript[18]，其中HTML作为浏览器上显示的内容，CSS是对内容进行生动形象的表现，而JavaScript作为网页的行为，可以被嵌入到HTML文件之中，并能够将当前所触发的事件进行处理或者对网页进行初始化工作[19]。将这三种技术相融合，才能让本系统的前端网页上的内容更加人性化、生动，满足系统的要求和用户使用的需求。

系统需要在浏览器端提交某些用户请求，不希望服务器像传统的Web同步请求框架一样，每次请求都要再次刷新页面，这样会给用户带来非常多的不便，而希望用户在提交某些用户请求时服务器快速响应，并将返回值返回给前端，更新部分网页信息。这就需要用到浏览器和服务器之间使用异步通信机制进行数据通信，即而允许浏览器向服务器获取少量信息而不刷新整个页面[20]。Ajax就是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术，但是Ajax 对浏览器兼容性存在差别，需要解决Ajax对浏览器兼容性问题，这回大大增加开发成本[21]。用jQuery实现Ajax，大大简化了 Ajax 操作，兼容各主流浏览器，其方便灵活的事件设置为无刷新页面增加了友好的效果与交互[22]。如下图2.2所示，为利用JQuery框架进行异步请求的结构[23]。

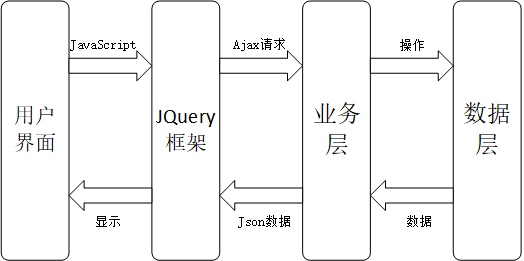


图2.2 JQuery请求框架

2.3 功能需求分析

为了能够依据个人文献并按照指定的要求生成论文报表，首先需要有个人文献数据库，生成论文报表时能够查找个人文献数据库，然后再生成相应的论文报表，因此在系统中必须拥有上传个人文献资料的功能。BibTeX是一个由Oren Patashnik和Leslie Lamport在1985年为LaTex文档处理系统所设计的程序和文件格式[24]，是一种书目数据文件格式，用来列出参考书目项目，它完全基于字符，所以可以被任何程序所读取。BibTeX书目文件名通常以.bib结尾。BibTeX数据库文件由条目列表构成，每个条目对应一个书目项目，条目类型对应于各种类型的书目来源，例如文章，书籍或会议。国内外的大部分文献数据库例如百度学术、Google Scholar、dblp等都支持导出BibTeX文件，应用范围极其广泛，因此用户只需要从其他网站导出自己文献的BibTeX文件，系统即可获取到相关的文献资料，用于生成个人文献列表和论文报表。

按照系统的目标功能，分析出以下的功能模块：

1. 用户注册、登录和登出功能

一个完成的Web系统需要包含注册、登录和登出的功能，未登录不能进行个人信息的编辑，因此科研人员必须进行注册和登录，才能使用本系统中的具体功能，否则只能使用不需要登录的功能。用户想要退出当前用户时可以选择登出。

1. 建立科研人员个人信息并生成个人主页

本系统不仅希望生成论文报表，同时也希望在系统中增加个人信息，生成个人主页，便于自己和其他科研人员了解。科研人员的基本情况包含科研人员的真实姓名、英文名、电子邮件、研究兴趣、工作单位、部门、职位、个人项目和自我介绍等基本信息，将这些信息通过用户录入的方式入进数据库，并提供修改的操作。同时允许上传科研人员的用户头像。科研人员用户可以设置自己主页的URL（要求唯一，不能重复，格式确定），其他浏览者可以通过该URL访问。

1. 导入科研人员个人文献

通过BibTeX格式向系统导入个人文献，其一为初始化个人信息时上传BibTex格式的文件（批量导入），其二为后期添加个人文章（分篇添加），这里又可以分为两种情况：1）直接复制BibTeX格式的文本；2）参考开源软件JabRef，模仿其选择文章类型的方式添加文献，并将其填写的内容转为BibTeX源码。最后依据BibTeX生成个人参考文献列表，显示在个人主页上。

1. 上传和下载科研人员文章的PDF

为了便于让用户方便的在参考文献列表上直接下载文章的PDF文件，本系统提供为科研人员上传和下载文章PDF文件的功能，科研人员可以为自己的主页上传文章PDF，用户本人和其他浏览者随时下载浏览文章内容。

1. 按照指定格式生成论文报表

根据科研人员的个人文献，按照科研人员指定的论文报表表头格式，生成论文报表，显示在用户浏览器的页面上。为了能够方便用户快速方便的生成论文报表表头，本系统需要能够动态的生成表头的列，然后通过拖拽已有的表头选型或者手动编辑的方式，快速生成论文报表表头。此外还需要提供选择期刊文章或者会议文章的功能（也可以两者都选），便于用于只导出某一类文章。还需要能够依据表头的关键字进行排序的功能，能够提供多个关键字排序，排序按照关键字的主次进行排序。当生成好论文报表时，能够提供给用户下载论文报表的功能，报表的格式可以有两种，.docx和.csv，供用户选择。

1. 根据关键字搜索科研人员

本系统应该为用户提供关键字搜索功能，依据科研人员所在的研究单位，列出所有符合搜索条件的科研人员，便于用户查看其他科研人员的个人主页。

2.4 系统数据库设计

依据系统功能分析，建立系统的数据层，采用MySQL数据库，数据库命名为test，建立了用户注册信息、用户基本信息以及文章信息三个实体。

用户注册信息实体用于存放用户的注册信息，其属性有：用户id、电子邮件、真实姓名、电话号码、用户密码，其中电子邮件是主键，作为登录的用户名，用户id唯一。

其E-R图[25]如下图2.3所示：

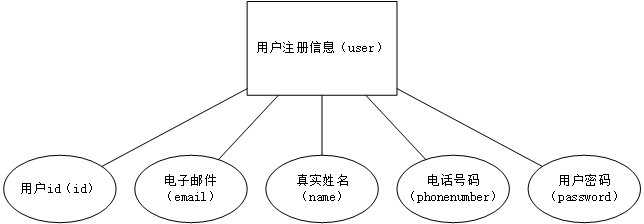


图2.3 用户注册信息E-R图

用户基本信息实体存放科研人员个人主页上的基本信息，其属性有：用户名、真实姓名、英文名字、联系方式、URL域名、研究兴趣、研究机构、部门、职位、自我介绍、头像路径，其中用户名为用户注册信息表的外键，并作为本表的主键。

其E-R图如下图2.4所示：

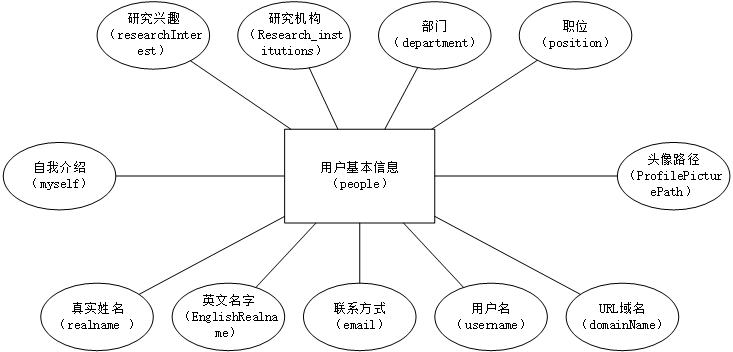


图2.4 用户基本信息E-R图

文章信息实体存放科研人员的文章信息，文章包括期刊和会议两种，其属性包括：用户名、文章BibTeXKey、文章PDF路径，其中用户名为用户注册信息表的外键，而用户名和文章BibTeXKey共同作为文章信息表的主键。文章BibTeXKey为每篇文章对应的BibTeX的key值。而用户的BibTeX文件以及上传的PDF文件全部都保存在文件系统之中，不会写入数据库，数据库只写文件路径，根据路径找到文件处理即可。

其E-R图如下图2.5所示：

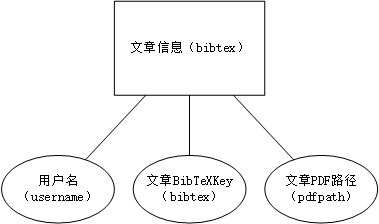


图2.5 文章信息E-R图

这些实体的关系如下：

（1）每个注册的用户都有唯一的用户名，并且只能对应一个用户基本信息；每个用户的基本信息也只能在用户注册信息中找到唯一的对应，因此用户注册信息和用户基本信息为一对一的关系。

（2）每个用户可以发表多篇文章，每篇文章都可能有一个或者多个作者，因此用户注册信息和文章信息之间为多对多的关系。

三个实体之间关系的E-R图如下图2.6所示：

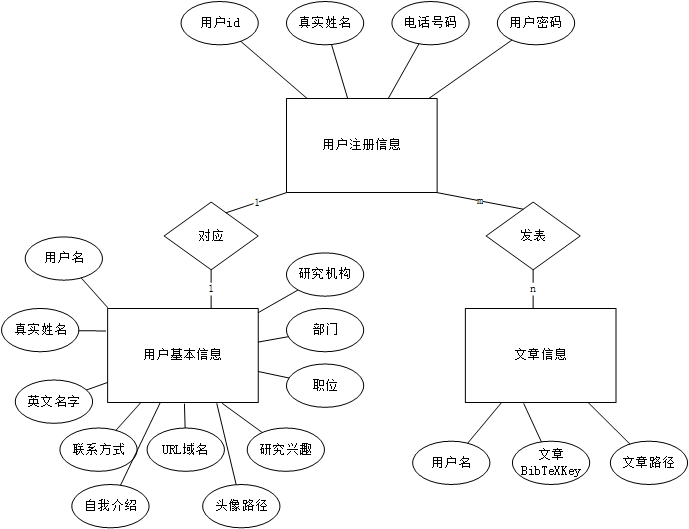


图2.6 个人科研助理系统E-R图

2.5 业务流程分析

通过对本系统具体功能的分析，按照功能之间的先后顺序和并列关系，可以得到本系统的业务流程，业务流程图为用户提供了使用系统的顺序。

业务流程图如下图2.7所示：



图2.7 系统业务流程图

第三章 系统实现

3.1 用户注册、登录、登出模块

根据第二章中的系统功能需求分析，本系统需要完整的注册、登录和登出的功能。依据用户是否登录可以将用户分为两种，第一种为游客用户（未登录），这时用户仅仅可以通过访问URL的方式或者通过搜索的方式查看其它科研人员的个人主页。第二种为登录用户，这时用户可以进行对自己的个人信息、文献信息等进行添加和修改，依据个人的文献信息生成所需要的论文报表，登录用户除拥有自己特有的功能外，还拥有游客用户的功能。此外，用户想要退出登录时可以登出。

3.1.1 使用Flask-Login实现用户登录、登出

Flask-Login[26]-[27]为 Flask Web服务器提供用户会话管理，它可以处理用户的登录和登出，并能长时间保存用户的会话。它能够在会话中存储登登录用户的ID，使用户能够自由地进行登录和登出，并能够限制未登录用户访问某些视图。

使用Flask-Login实现登录登出的流程如下所示：

1. 配置应用：

在Web服务器代码app.py的开头处为应用创建一个LoginManager类对象:

login\_manager = LoginManager()

登录管理(LoginManager)包含让应用和Flask-Login协同工作的代码，比如从一个ID加载用户，当用户登录时跳转到哪个视图等等。

login\_manager对象创建后，进行配置，调用login\_manager的init\_app绑定Flask服务器程序:

login\_manager.init\_app(app)

进行相关参数的设置：

loginmanager.session\_protection = 'strong'

login\_manager.login\_view = 'login'

session\_protection为设置登录用户的安全级别（None、'basic'、'strong'）。login\_view是设置登录时运行视图函数，这里将login\_view设置为'login'。

初始化LoginManger后，在模型中定义回调函数，在验证登录时从数据库加载用户：

@login\_manager.user\_loader

def load\_user(userid):

return User.get(userid)

1. User模型继承UserMixin：

使用flask\_login进行用户的登录和登出管理，需要将User模型继承flask\_login的UserMixin基类：

class User(UserMixin):

def \_\_init\_\_(self, id, username, password):

self.id = id

self.username = username

self.password = password

这里只在User类中定义了三个必须的属性，分别为用户id，用户名以及用户密码，他们分别对应于第二章中数据库设计中user表中三项。

用户密码使用Werkzeug来进行加密。Werkzeug中提供的security模块能够将密码进行散列值计算。分别在注册用户和验证用户阶段调用generate\_password\_hash和check\_password\_hash进行密码的加密和密码的验证。

1. 用户的登录和登出：

用户通过验证时，使用[login\_user](http://www.pythondoc.com/flask-login/#flask.ext.login.login_user)使Flask保存会话，表示已经登录，当想要登出时，使用使用logout\_user，将此次会话登录的User退出。[current\_user](http://www.pythondoc.com/flask-login/#flask.ext.login.current_user)为Flask获取当前会话中登录的用户。

1. 限制登录对象：

需要用户登录才能进行操作的视图可以用[login\_required](http://www.pythondoc.com/flask-login/#flask.ext.login.login_required)装饰，可以限制URL只有登录的用户可以访问。

3.1.2 注册、登录、登出详细实现

在主页面上的页眉右上角设置注册、登录、登出的超链接<a>，点击注册和登录时可以直接跳转至分别对应的页面。

用户注册页面的视图函数为'register'，不需要login\_required。在注册界面需要用户输入用户账号（用email表示）、真实姓名、手机号码、密码和确认密码。这五项内容都为必填选项，少任何一项都会弹出提示框，提醒用户应该输入某一项。密码和确认密码都采用html中input标签，其类型为'password'，输入时密码以实心黑点的方式呈现，避免密码泄露。密码和确认密码需要一致，不一致时会弹出提示框，要求用户重新输入。用户账号必须为系统中未使用的邮箱号，否则会提醒用户名已经使用。

满足上述条件后点击注册按钮，即可向Web服务器发送注册信息，发送请求的url为'registerinfo'。后台收到数据后，首先依据用户名查询数据库，如果用户存在，则向前端返回用户名已经存在的信息；否则，将密码使用generate\_password\_hash进行加密后一同存入数据库的user表中，并向前端返回注册成功的消息，前端接收到注册成功的返回值后，跳转至登录界面。用户id采用获取当前数据库里最大id后再加1的方式，进行插入，以保证id的唯一性。

用户登录界面的视图函数为'login'，输入用户名和密码后，提交至服务器，发送请求的URL仍为'login'。后台视图函数'login'首先判断是否是提交信息的请求，如果是则查询数据库判断用户名是否存在，不存在则直接返回给前端，提示用户名不存在，存在则使用check\_password\_hash判断密码是否相等，返回给前端相应的结果。在登录界面也设置没有账户进行注册的标签<a>。

用户登出使用logout\_user()即可，在前端设置登出链接，服务器收到请求后执行logout\_user()即可退出当前用户。

3.2 创建及修改个人信息模块

3.2.1 创建个人信息

新用户会有创建个人基本信息的页面，方便用户在进入系统时对个人资料进行初始化。创建过程分为五个部分，分别显示在不同的界面，对应的视图函数分别为'pageone'、'pagetwo'、'pagethree'、'pagefour'、'pagefive'。

'pageone'为创建个人学术主页，主要让用户填写真实姓名、英文名字、联系方式、URL域名、研究兴趣和自我介绍，每一项都为必填项，如果不填写会有弹出提醒。其中URL域名为科研人员个人主页的URL，其格式为：'Web服务器的IP地址和端口号'+'/people/'+URL域名，例如Web服务器的IP和port为'127.0.0.1:5000'，URL域名为'w'，那么该用户个人主页的URL为：'127.0.0.1:5000/people/w'，用户自己和其他用户都可以通过该URL访问其个人主页。因此，该URL域名必须唯一。因此在向服务器发送数据后，服务器需要查询用户提交的URL域名是否已经被使用，如果已被占用，则需要重新填写。

'pagetwo'为填写科研人员的工作单位，包括研究机构、部门和职位，同样每一项都为必填项。

'pagethree'为上传个人头像，允许选择本地的图片，并进行裁剪、选择后进行上传，服务器在静态文件夹static下创建一个文件夹名为'pic'的文件，并在该文件下创建以用户名为文件夹名的子文件夹，将其前端上传的头像文件命名为'test.png'后保存在用户对应的子文件夹下。然后将头像的路径保存进数据库people表中。将文件放在静态文件夹static下的原因是static下的文件前端可以直接读取，服务器不必在通过发送文件的方式将头像图片发送至前端。

上传图像的前端使用了插件cropper，它支持图片放大缩小和图片旋转、触摸屏设备、canvas并能够跨浏览器使用。使用cropper将裁剪的图像转为png格式并将其进行base64编码，通过Ajax的方法发送给后台，后台将接收到的信息进行解码，写入服务器的文件系统。

'pagefour'为添加科研人员的文章，文章的信息以BibTeX格式的文件表示，在该界面允许上传BibTeX文件，点击上传按钮后前端会以表单的信息上传文件，后台接收并保存至服务器目录下的bib文件夹下的以用户名命名的子文件夹下，文件名为total.bib。通过解析BibTeX文件，将total.bib文件中包含的期刊（article）和会议（inproceedings）的论文的bibtexkey获取到，与用户名一起作为主键，插入数据库表bibtex中，此次为初始化插入，其对应的PDF路径为空。BibTeX文件的解析将再下一小节介绍。如果前端上传的文件是.bib格式并且能够成功解析插入数据库，则前端会弹出上传成功的提示框，否则会弹出上传失败的提示框。

'pagefive'为创建成功的提醒界面，这时用户可以选择查看通过刚刚信息录入而生成的个人主页。此时的个人主页页面的风格样式与对外显示的个人主页是一样的，不同之处在于此时用户可以编辑、修改这个界面。

至此，个人主页的初始创建过程就已经完成，当用户再次登录系统时，将不会重复上述过程，但是同样可以查看和修改自己的个人主页。

3.2.2 BibTeX文件的解析

从BibTeX中获取数据，下面给出一个BibTeX示例：

**@Book**{abramowitz,

author = "Milton {Abramowitz} and Irene A. {Stegun}",

title = "Handbook of Mathematical Functions with

Formulas, Graphs, and Mathematical Tables",

publisher = "Dover",

year = 1964,

address = "New York City",

edition = "ninth Dover printing, tenth GPO printing"

}

该示例的文章类型为Book，它的bibtexkey值为abramowitz，其属性有author、title、publisher、year、address和edition。

Python的bibtexparser模块可以解析BibTeX文件，该API与json模块类似。解析后的数据作为简单的BibDatabase对象返回，主属性是表示书目来源（如书籍和期刊文章）的条目。下面的示例提供了一种操作BibTeX文件的快速而基本的方法：

import bibtexparser

with open('bibtex.bib') as bibtex\_file:

bibtex\_database = bibtexparser.load(bibtex\_file)

bibtex\_entries = bibtex\_database.entries

使用entries可以获取文件中的BibTeX条目列表，例如@book {...}，@article {...}等。每个条目都是一个带有BibTeX字段值对的简单dict，例如'author'：'Milton {Abramowitz} and Irene A. {Stegun}'，每个条目将总是有以下dict键值（除了其他BibTeX字段）：

ID (BibTeX key)

ENTRYTYPE (条目类型，例如：book等)

利用bibtexparser模块就可以将BibTeX文件中的期刊和会议文章（每一篇文章都是一个条目）的所有属性何其对应的值获取到，为系统提供了极大的便利。

3.2.3 修改个人信息

修改个人信息包括两个部分，一部分为修改个人基础的信息，比如联系方式、研究单位等等，这部分创建一个单独的页面，在加载页面时读取数据库，将对应的信息填写html中的input，用户可以自行修改任何一处，修改完成后，会将全部信息重新发送到服务器端，并将修改的信息更新数据库。第二部分为添加或者删除科研人员的文章信息。

在修改文章信息的界面，通过解析BibTeX，将total.bib中的期刊和会议以(GB/T 7714－2005)参考文献规范的格式显示，每一篇文章都动态的显示在一个div标签中，并在每一个div标签的右上角设置一个删除的标签，点击'X'，就可以隐藏该div，也就是做到了在前端删除的功能。在前端维护一个文章bibtexkey的list，删除了哪个文章，就在list中删除对应文章bibtexkey。最后点击保存时，在将此数组发送至服务器，服务器对比数组之后，删除total.bib中对应的条目，并删除数据库中相应的条目。

在修改文章信息的界面，点击添加文章的选项，即可跳转至增加文章的页面，增加文章有两种方式，一种是直接复制一条BibTeX条目，另一种则是仿照开源软件JabRef添加文章的方式进行添加。下面着重介绍第二种：

点击填写按钮，会有提示框弹出，提醒用户选择增加文献的类型，一共可以分为14中类型，分别为'Article'、'InBook'、'Book'、'Booklet'、'InCollection'、'Conference'、'InProceedings'、'Proceedings'、'Manual'、'MasterThesis'、'PhdThesis'、'TechReport'、'Unpublished'、'Misc'，每一种类型都有不同的可选域和必选域，所有的类型都有相同的基本设置、摘要和评论等选项。例如'Article'类型的文献，其必选域为：'Author'、'Title'、'Journal'、'Year'、'Bibtexkey'、'Note'，可选域为：'Volume'、'Number'、'Pages'、'Month'、'ISSN'、'Note'。前端利用JavaScript根据用户填写的BibTeX信息，生成BibTeX格式的信息，并实时生成BibTeX源码，显示在BibTeX源码区域。不论是采用直接复制的方式，还是采用填写BibTeX的方式，点击提交按钮时，都会向服务器发送BibTeX源码内容，服务器收到后，首先利用文章名进行查询，如果有文章名重复的现象发生，则提醒用户是否确定提交，如果确认提交，则将BibTeX源码追加进total.bib，并更新数据库（向数据库表bibtex增加新条目）；如果不确认提交，则取消此次添加请求。

3.3 上传和下载PDF模块

3.3.1 上传PDF

在3.2小节中所述的增加与删除文献，仅仅是对文献的BibTeX进行操作，并没有涉及到BibTeX中提到的期刊和会议文章PDF的上传，为了便于科研人员本人以及其他用户方便的浏览PDF，在本系统中增加上传文章PDF的功能。在可编辑的个人主页页面，在每篇文章所在div标签的右下角，增加一个上传PDF的标签<a>，在加载可编辑个人主页界面时，首先依据数据库表bibtex判断文章是否上传过PDF，如果已经上传，则显示PDF下图标，表示该文章可下载，否则，在右下角显示上传PDF标签，用户点击即可为对应的文章上传PDF。

点击上传PDF进行页面跳转时在URL里传递参数bibtexkey，便于告知服务器上传的PDF对应于哪个文章。上传PDF时以表单的方式提交文件，服务器收到文件后将文件以bibtexkey命名（唯一），因为文件名作为URL被用户下载时出错，这里讲bibtexkey使用python的URL编码函数urllib.quote，将文件名编码后保存在服务器目录下的pdf文件夹下的以用户名命名的子文件夹下，并将文件路径写入数据库。

3.3.2 下载PDF

当科研人员的某篇文章上传了PDF后，即在个人主页的界面下载，下载时，使用文章作者的用户名和bibtexkey作为URL，例如'127.0.0.1:5000/download/ [123@qq.com/bibtexkey](mailto:123@qq.com/bibtexkey)'，服务器收到请求后，依据URL中的用户名和bibtexkey查看找到文件的路径，并使用send\_from\_directory函数，将文件发送至前端。

3.4 生成个人主页模块

个人主页可以通过两种方式访问，第一种为对所用用户（不需要登录也可以访问）通过在创建个人主页时输入的URL域名进行访问，在这种情况下，用户不可编辑，只能浏览和下载科研人员文章的PDF；第二种为在登录状态下科研人员自己的个人主页，此情况下不需要URL域名，所有用户统一的域名都一样，例如：'127.0.0.1:5000/editpeople'，Web服务器会根据浏览器登录的用户区分不同的个人主页，此界面与第一种界面布局相同，不同点在于此页面可编辑，即修改用户的个人资料。

个人主页界面分为上下两个部分，上面主要显示科研人员的基本介绍、联系方式和头像显示，下半部分则为文章显示部分，解析BibTeX科研人员的期刊和会议以(GB/T 7714－2005)参考文献规范的格式显示，每一篇文章都动态的显示在一个div标签中。

3.5 生成论文报表模块

本模块是本系统中一个非常重要的一部分，其主要作用是在前端能够按照指定的格式生成论文报表，报表以表格的方式呈现，可以使用输入或者手动拖拽的方式在网页上动态的生成表格的表头，希望能够按照表头的格式自动将自己期刊或会议文章的信息填入表格，例如，用户希望将自己发表的所有期刊按照表3.1的方式填写：

表3.1 论文报表表头

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Title | Year, Volume(Number) | Author Order |

表格的内容分别是文章的作者，文章的题目，按照 Year, Volume(Number) 的格式显示期刊的Year、Volume和Number，作者的顺序。下面目标生成的论文报表，假设用户本人为张三，发表过3篇期刊：

表3.2 论文报表

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Title | Year, Volume(Number) | Author Order |
| 张三，李四，王五 | 文章1 | 2018, 20(2) | 1 |
| 李四，张三，王五 | 文章2 | 2017, 10(5) | 2 |
| 李四，王五，张三 | 文章3 | 2016, 15(1) | 3 |

如表3.2所示，动态生成论文报表的功能想要达到这样的效果，生成论文报表还希望能够将报表导出为.docx和.csv格式的文件，方便用户直接下载使用。除此之外，还希望能够选择期刊或者会议，并且能够对表格按照排列条件进行排序。

1. 前端生成报表表头

使用<table> 标签定义 HTML 表格。前端可以动态生成表格的列以及删除表格的列，点击增加列的按钮即可生成出表格的表头，表格的表头设置为可编辑，以便用户对表头的格式进行修改，表头默认的文字均为'Click edit'，改变表头文字的方式可以是直接拖拽已有的标签列，例如Author、Title、Journal、Year、Volume、Number、Pages、Author Order，也可以是手动修改列中属性的格式。记录表格的表头，并将表头保存至JavaScript的数组中，作为向服务器请求的格式方式发送到服务器。

1. 前端选择文章类型以及排序方式

在前端设置input，将其type设置为checkbox选择框，为用户提供选择期刊或是会议文章生成论文报表的功能，默认情况下是期刊和会议都会列入报表。同时，前端还提供了对报表的排序方式，排序方式分为三个关键词排序，主要关键字、次要关键词、第三关键词，关键词为表头列的文字，除此之外，还提供按照Year进行排序，以上面的例子：关键词可以为：Author、Title、Year, Volume(Number)、Author Order、Year，Year会按照年份从大到小进行排序，Author Order会按照作者顺序从小到大排序，其余会按照首字母进行排序。用户可以根据需要选择一个、多个或者不选，对表格进行排序。每次点击提交时，会将（1）中表头列表、选择期刊或论文、排序方式发送至服务器，然后生成对应要求的报表返回给前端。

1. 依据BibTeX生成报表数据

构造Python类LoadBibtex，其构造函数的参数分别是：用户BibTeX文件对用的路径BibtexfilePath、用户请求的报表表头show\_entry，其类型为list、用户本人可能在文章中使用的名字myself，以及文章的类型types。

用户本人可能在文章中使用的名字myself由自定义的Python类EnglishName提供，参数为中文名（用户本人的真实姓名），生成几个可能使用的英文名，例如：用户名为张三，则使用EnglishName的方法getAllNama()将会返回list['san zhang', 'zhang san', 's.zhang', 'san z']，此英文名list主要用户生成作者顺序，查找以上list中任意一个在Author中的顺序。

由于不同的类型的文章有不同的必选域和可选域，例如期刊Article 有下列的属性all\_entry = ['Author', 'Title', 'Journal', 'Year', 'Volume', 'Number', 'Pages', 'Month']，将用户请求的报表表头show\_entry中的元素在all\_entry中查找，如果存在，则就以属性名为键，在BibTeX中对应的值为键值，写入dict，如果不存在，则为用户自定义的格式，这时要进行格式解析，获取格式内属性值，然后将BibTeX中的值按照格式拼接，写入同一个dict中。这样，每篇文章都会生成一个以show\_entry表头为键，解析值为对应取值的dict。为了便于用年份排序，当show\_entry表头没有Year属性时，在dict中增加。最后将每篇文章对应的dict增加至一个list all\_row中，all\_row则为表格中的数据。

Web服务器收到前端的请求以后，首先根据用户选择的文章类型，从BibTeX文件中挑选出用户所需要的文章。然后调用类LoadBibtex生成表格数据，接下来就是进行排序。由于用户请求有排序先后，关键词的先后顺序保存在list order中。由于月重要的排序要排序时要越靠后，因此排序时按照将order倒序后进行排序。将排序后的all\_row返回给前端，然后显示在前端的表格里。

1. 生成.docx和.csv文件并可以下载

将排序后的all\_row以两种格式写入文件分别是.docx和.csv，分别命名为'pubs'+时间戳+'.docx'和'publications'+时间戳+'.csv'，保存至服务器目录下的file文件夹下的以用户名命名的子文件夹下。前端设置导出标签，分别可以导出上述的两个文件，供用户使用。

**3.6 搜索科研**人员模块

在系统首页提供搜索框，用户可根据科研人员的工作单位进行搜索，并将所搜结果显示在搜索框下方。

用户在前端的搜索框内输入例如‘地质大学’这四个关键词，按下回车或者点击搜索，可以将此关键词发送到后台服务器，服务器依据此关键词在数据库people表中的Research\_institutions中进行匹配查询，查询语句使用SQL语句的LIKE[29]操作符，将查询结果返回给前端，对每个用户的信息都放在动态生成的div标签中，该标签上显示了用户的基本信息，点击该标签即可跳转至用户的个人主页，查看用户的详细信息。

第四章 系统运行

本系统的运行主要是用户前端的展示，本章将依据第三章中系统流程分析，展示本系统的所有界面。系统的首页设计较为简单，页面的上侧的右上角主要提供系统的功能选择，包括注册、登录、登出、查看个人主页以及生成论文报表，界面中央提供搜索框。如下图4.1所示，下图中实线框为登录用户使用功能，需要登录后才能在主界面上显示，未登录情况只显示虚线框内容。

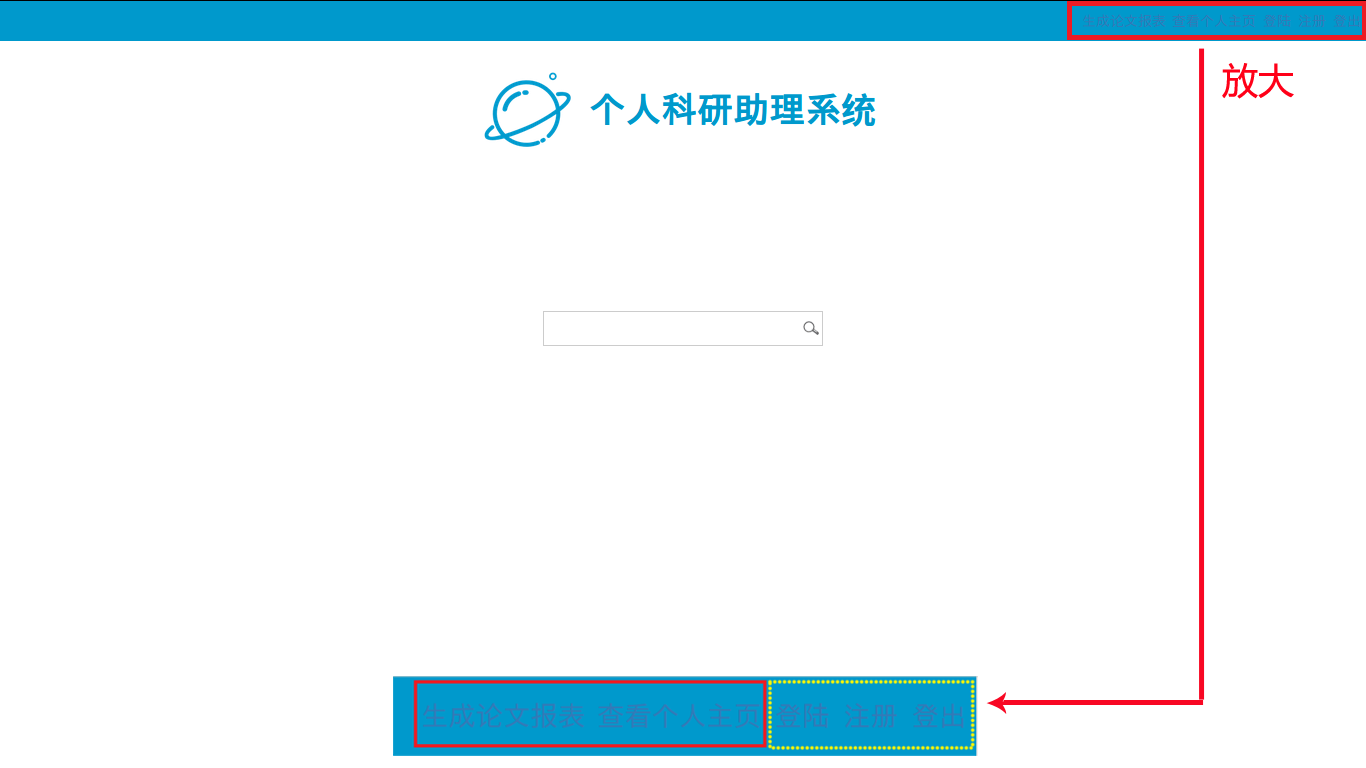


图4.1 系统首页界面



图4.2 用户注册界面

如上图4.2所示，科研人员进入系统需要注册，[此处以科研人员张三为例，用其](mailto:此处填写的用户账号为科研人员的邮箱号码zhangsan@qq.com)邮箱'zhangsan@qq.com'作为用户名，密码'123456'不显示。注册成功后，可跳转至登录界面，如下图4.3所示，用户使用邮箱号进行登录，登陆后可进入系统主界面。登录失败会提醒失败原因。



图4.3 用户登录界面



图4.4 创建个人科研主页界面一

新注册的用户在系统的主页面点击创建个人主页，跳转至初始化个人主页的界面。此过程一共有五个界面，每个界面都有不同的功能。

第一个界面如上图4.4所示，主要填写用户的姓名、英文名、联系方式、研究兴趣、自我介绍、个人主页URL域名等信息，此处仍然以用户张三为例，其URL域名设置为'zhangsan'，如果输入系统已有的URL域名，会提醒用户重新选择。

第二个界面如下图4.5所示，主要填写用户的工作单位、部门以及职位。



图4.5 创建个人科研主页界面二



图4.6 创建个人科研主页界面三

第三个界面如上图4.6所示，从用户本地选择图片作为头像上传到服务器。

第四个界面如下图4.7所示，上传用户文章的BibTeX文件，该BibTeX文件名为：'Zhangsan.bib'，其中包含了若干期刊和会议的文章。选择文件路径后点击上传，如成功，则会提醒上传成功，否则提醒上传失败。



图4.7 创建个人科研主页界面四

第五个界面如下图4.8所示，用户如顺利执行上述的步骤，即可进入如下的创建成功界面，点击查看个人主页即可查看可编辑的个人主页。



图4.8 创建个人科研主页界面五

如下图4.9所示，可编辑的个人主页，这里仅列出了个人文章中的部分，此页面下方还有很多用户的文章信息。可以对用户的信息进行修改，没有上传PDF的文章会在每个文章的右下角显示'上传pdf'的标签。



图4.9 可编辑的个人主页

点击上图4.9中实线框1可进入编辑个人科研主页资料页面，如下图4.10所示，在工作部门出增加信息，点击保存后，即可向服务器提交更新后的信息。



图4.10 个人科研主页资料页面

点击上图4.9中实线框2可进入个人文献修改界面，用户可以删除或者添加文章，其界面如下图4.11所示，点击图中实线框部的'X'，即可删除对应的文章，点击添加图标，即可跳转至增加论文的界面。

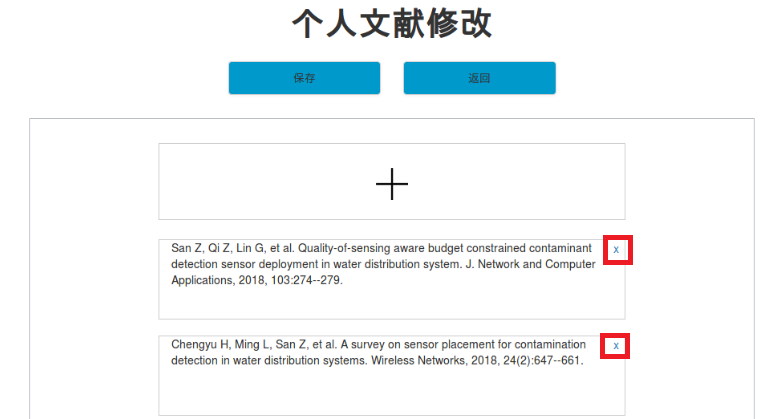


图4.11 个人文献修改页面

添加文章有两种方式，其一为直接复制BibTeX文本，如下图4.12所示，点击提交即可上传，同时也可以选择上传或者不上传对应的PDF：



图4.12 复制BibTeX

当选择填写BibTeX时，首先会弹出选择框选择添加文章的类型，如下4.13所示，这里选择'Article'，然后根据文章的类型，生成不同的必选域和可选域，如下图4.14和4.15所示：

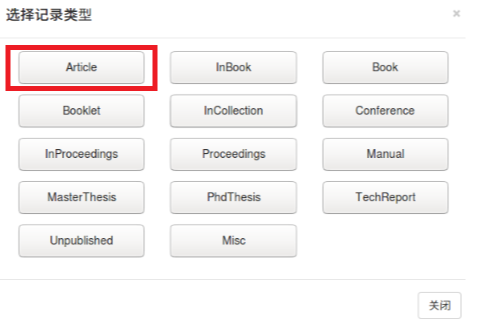


图4.13 选择文章类型

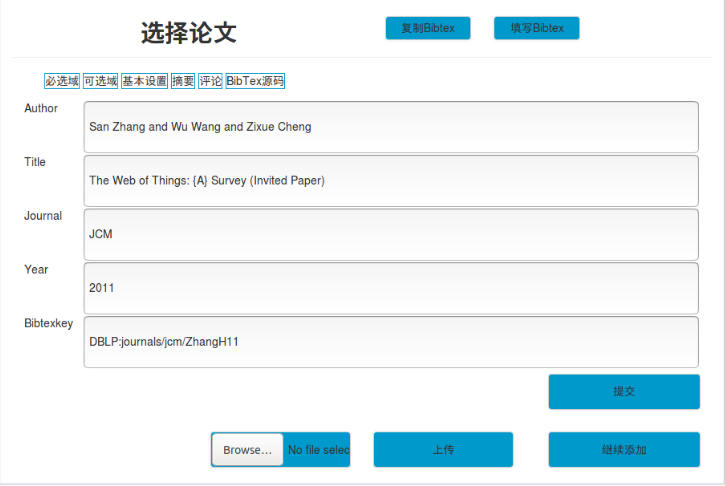


图4.14 'Article'必选域



图4.15 'Article'可选域

依据必选域、可选域、基本设置、摘要、评论生成BibTeX源码，如下图4.16所示。两种方式提交BibTeX源码时如服务器判断疑似重复，则提示如下图4.17所示的提醒框，由用户选择是否继续提交。



图4.16 BibTeX源码

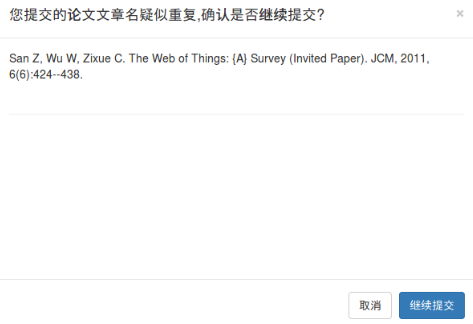


图4.17 提醒文章可能重复

点击上图4.9中上传pdf标签对相应的文章上传PDF文件，下图4.18为修改个人资料、删除文章、增加文章、上传文章PDF后个人主页，实线框处为PDF可下载图标，图4.19为下载PDF文件的提示框。



图4.18 修改后的个人科研主页

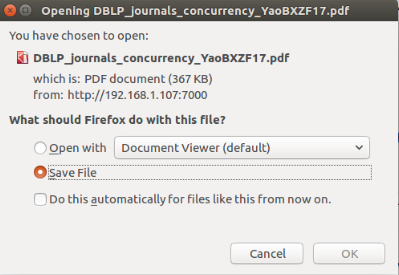


图4.19 下载PDF文件



图4.20 生成论文报表页面

最后一个模块为生成论文报表模块，其页面如上图4.20所示，首先选择期刊，在页面上点击增加列，分别拖拽Author、Title、Journal、Author Order，拖拽并编辑生成Year, Volume(Number)，选择排序方式，主要关键字选择Author Order，次要关键字选择Year，然后可生成论文报表，如下图4.21所示：

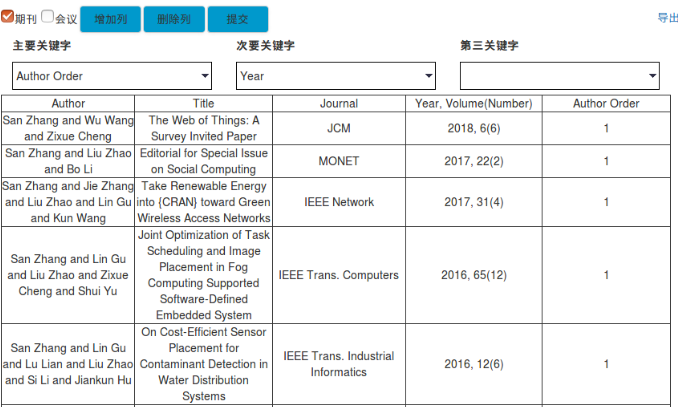


图4.21 论文报表

点击导出按钮，可选择导出.docx和.csv的文件，如下图4.22所示，为导出.docx的下载提示框：

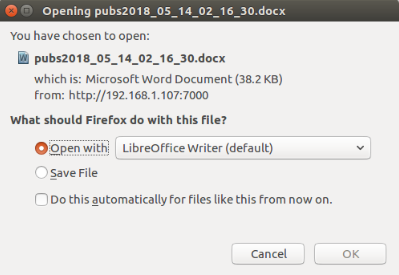


图4.22 导出为.docx

依据科研人员工作单位进行搜索的功能如下，假设系统中属于中国地质大学的科研人员有张三、李四、王五等，其搜索结果如下图4.23所示，点击对应的科研人员即可查看详细的个人主页。



图4.23 搜索结果

第五章 总结与展望

5.1 全文总结

基于Web的个人科研助理系统不依赖于任何操作系统以及软件，仅仅需要一个浏览器就可以随时随地进行用户的个人科研的文章管理、快速生成所需要的论文报表、快捷的查看其他用户的科研信息等等功能，能够充分的提高科研人员的工作效率，节约时间，对高校的科研人员来讲是一个非常必要的个人助理系统。

本系统在原理上比较简单，以常用的B/S架构作为开发的基础，使用Python Flask和MySQL作为服务器，使用HTML、JavaScript和CSS配合开发前端，使用JQuery进行前后台交互的主要方式，构建了包括注册、登录、实现系统功能的完整的B/S架构。在实现系统中，涉及到的算法很少（基本没有），但是工程量比较大，前端+后台的代码量很充足。

本系统基本达到了系统预期的需求，也在开发中增加了部分的细节和功能，也在开发中不断完善系统。

5.2 后续展望

由于个人时间和精力有限，不能独自将系统进行更多的扩展，实现更多有效的功能。

（1）增加依据个人科研主页生成科研人员的个人简历，并将其转换为可导出的PDF格式文件的功能，此功能有利于科研人员快速方便的生成个人简历。

（2）增加依据系统中所有用户上传PDF文件简历倒排索引[30]的功能，此功能有利于用户科研人员可以根据文章的关键词搜索出对应的PDF，找到有用的文章。

本文中没有实现上述功能，但是这些功能同样对于整个系统来说非常重要，后续可以继续完善本系统，提供一个真正行之有效的科研助理系统给高校的科研人员。

致谢

以梦为马，不负韶华，时间如同白驹过隙，转瞬即逝。在人生这趟列车上，我度过了生命中最美好的四年。

首先我要感谢这篇论文的指导老师曾德泽老师，正是在曾德泽老师的耐心指导和帮助下，我才能够完成这次系统的设计。曾老师作为我们的班主任，四年来在学习和生活上对我有太多的帮助，他教我们学习的态度与方法，像指路明灯一样为我们指引方向。记得第一次见面时，他就送给我们四个字：“天道酬勤”，事实也证明，只有努力的去做一件事情，才能在生活中有所收获。通过这四年的努力，我也找到了向下一站前进的方向。

还要感谢任课老师和学工组的各位老师，尤其是刘佳老师和李欢欢老师，在他们的帮助下我成长了许多，不管是学习方面还是为人处事方面都比以前更懂得如何去做到更好。同时还要感谢193142班的同学们，感谢你们在学习与生活上对我的帮助，没有你们，大学四年我不会过得如此开心。

最后我要感谢我的父母，是他们赐予我生命给予我爱，教会我成长去教会我做一个更好的人。在我眼里他们是全世界最好的父母。

感谢所有的相遇，也感谢所有的离别，感谢我选择了地大，也感谢地大选择了我。离别了为了更好的重逢，亲爱的们，我们下一站再见。

参考文献

1. “十三五”国家科技创新规划[J].科技创新与生产力,2016(09):2+121.
2. 刘明军,杨京京,张珺.高校科研管理工作的现状与思考[J].大众科技,2006(03):166-167.
3. 中国高等教育质量报告(摘要)[N].中国教育报,2016-04-08(005).
4. 蒋伟,张弦.高校科研管理系统现状及发展方向[J].科技信息,2013(20):67-68.
5. 谢群.文献管理软件的功能层次划分理论研究——以Endnote为例[J].现代情报,2008(04):113-114+117.
6. 赵飞.常用文献管理软件功能比较[J].现代图书情报技术,2012(03):67-72.
7. 武苍林.B/S与C/S结构的分析与比较[J].电脑学习,1999(05):42-43.
8. 黄文博,燕杨.C/S结构与B/S结构的分析与比较[J].长春师范学院学报,2006(08):56-58.
9. 刘烨.C/S模式与B/S模式的对比分析[J].科技信息(学术研究),2007(30):221+223.
10. 曹风华.基于B/S的科研管理系统设计与实现[J].现代计算机(专业版),2010(04):198-200.
11. 许博.基于.NET三层架构的科研管理系统的开发与应用[J].实验室研究与探索,2013,32(02):188-190+207.
12. 杨利华,李津,何伟.基于Web三层架构的高校科研管理系统的研究与实现[J].北京石油化工学院学报,2012,20(02):37-40.
13. 郑岚.Python访问MySQL数据库[J].电脑编程技巧与维护,2010(06):59-61.
14. 黄传禄.基于Python的MYSQL数据库访问技术[J].现代信息科技,2017,1(04):73-75.
15. 曹华宇,张阿维.基于用户体验的网页界面交互设计原则[J].艺术科技,2015,28(02):206+230.
16. 白文涛,刘正捷.用户界面的需求分析与设计原则[J].大连海事大学学报,2004(04):86-88.
17. 朱燕婷.浅谈网页界面设计的人性化原则[J].信息与电脑(理论版),2009(12):65-66.
18. 杨毅.Web前端开发技术探讨[J].电脑知识与技术,2014,10(23):5458-5459.
19. 于为.HTML中JavaScript的一些应用探讨[J].安顺师范高等专科学校学报(综合版),2004(04):91-94.
20. 于虹.Ajax无刷新技术[J].电脑知识与技术,2009,5(18):4782-4784.
21. 曾顺.精通Javscript+jQuery[M].北京:人民邮电出版社,2007.
22. 李炳练.基于jQuery框架的无刷新技术设计与实现[J].电脑编程技巧与维护,2011(06):5+19.
23. 潘杰,周传生.基于jQuery框架的Web研究与实现[J].沈阳师范大学学报(自然科学版),2015,33(01):96-99.
24. O'rourke J.Art gallery theorems and algorithms[M].Oxford:Oxford University Press,1987：1-3.
25. 王珊,萨师煊.数据库系统概论(第5版).北京:高等教育出版社,2014:217-222.
26. Miguel Grinberg.Flask Web 开发:基于Python的Web应用开发实战[M].(安道,译).北京：人民邮电出版社,2015.
27. Copperwaite M,Leifer C.Learning Flask Framework[M].Packt Publishing Ltd,2015.
28. Fabio Nelli.Simple HTML Tables[M].Apress:2013-06-15,43-60.
29. 程志,张玉彤.浅谈Transact—SQL语句中LIKE的高级应用[J].数字技术与应用,2010(11):120.
30. 王涛,吴洁明.搜索引擎建立倒排索引的算法研究[J].现代计算机(专业版),2008(01):18-20.