学 号 2011061119

密 级

哈尔滨工程大学学士学位论文

基于BS架构的网络资料管理和推荐系统设计与实现

院（系）名　称：计算机学院

专　业　名　称：计算机科学与技术专业

学　生　姓　名：李日朗

指　导　教　师：申林山 讲师

哈尔滨工程大学

2015年5月

**基于BS架构的网络资料管理和推荐系统设计与实现 李日朗 哈尔滨工程大学**

学 号 2011061119

密 级

基于BS架构的网络资料管理和推荐系统设计与实现

Design and Implementation of the network data management and recommendation Systems

**学生姓名：**李日朗

**所在学院：**计算机学院

**所在专业：**计算机科学与技术专业

**指导教师：**申林山

**职称：**讲师

**所在单位：**哈尔滨工程大学

**论文提交日期：**2015年6月

**论文答辩日期：**2015年6月

**学位授予单位：**哈尔滨工程大学

摘 要

在信息大爆炸时代这个大背景下，现在社会上有很多很优秀的人才，他们做出了极为优异的作品，但他们的才华却被海量的数据无情的埋没了。本论文拟设计并实现一个网络资料管理和推荐系统让广大的作者们有一个平台展示他们的才华从而闪耀出无限的光彩。

在大量阅读和软件设计与实现以及推荐算法相关的文章以后，本文将按照软件工程中软件的实现思路来设计并实现本系统。论文中以需求分析、概要设计、详细设计、系统测试的顺序设计并实现一个完整可用的系统，该系统中实现了管理员信息管理、作者信息管理、游客信息管理、作者作品管理以及系统管理用基础功能等三十余种功能，同时设计了一个负反馈的推荐算法运用到系统的实现之中。

本文所设计的系统已经实现了大部分的功能点并进行了相关的测试解决了大部分系统问题，在实际的运行过程中本系统取得了较好的效果。

关键词： BS架构；网络资料；管理和推荐系统

**ABSTRACT**

In the background of information explosion, there are lots of people who are talent.They make excellent works, but their ability were buried.This paper will design and implementation of the network data management and recommendation system,This system allow that authors will have a platform to demonstrate their works.

Writer read lots of literature about design and implementation of the software systems and articles which are related about recommendation algorithm.This paper will design and implement this system according to the software engineering software implementation ideas. This paper will be written in the order of demand analysis, outline design, detailed design, system testing to design and implement a complete and available system.The system realizes a series of functions such as the administrator information management, the author information management,the visitor information management,the work management and system management with basis functions.There are more than 30 kinds of functions in this system.At the same time, this paper design a negative feedback of the recommendation algorithm and used the algorithm in the system.

In this paper, most of the functions which were designed in this paper are achieved.Writer do the related test to solve the most of problem in the system.This system made good effect in the course of the actual operation.

**Keywords:** BS architecture; Network data; Management and recommendation system

目 录

第1章 绪论1

1.1 论文的背景目的意义1

1.1.1 论文背景1

1.1.2 论文目的1

1.1.3 论文意义1

1.2 国内外研究状况1

1.2.1 国内研究现状1

1.2.2 国外研究现状2

1.3 论文的研究方法3

1.4 论文的内容及整体内容介绍3

第2章 需求分析4

2.1 任务概述4

2.1.1 目标4

2.1.2 用户的特点4

2.1.3 假定和约束4

2.2 需求规定4

2.2.1 对功能的规定4

2.2.2 对性能的规定5

2.2.3 输入输出要求5

2.2.4 数据管理能力要求6

2.3 运行环境规定6

2.3.1 支持软件6

2.3.2 采用结构和设计模式6

2.4 本章小结6

第3章 概要设计8

3.1 系统功能概述8

3.1.1 系统结构8

3.1.2 功能点概述9

3.2 操作及处理流程11

3.2.1 用户共有操作11

3.2.3 管理员操作14

3.2.4 作者游客共有操作16

3.2.4 作者特有操作17

3.2.5 游客信息管理18

3.3 系统数据结构设计19

3.3.1 逻辑结构设计要点19

3.3.2 操作与属性关系22

3.3.3 数据结构与程序的关系25

3.4 本章小结26

第4章 详细设计27

4.1 数据存取接口27

4.2 注册登录注销33

4.3 信息管理34

4.4 作品管理37

4.5 数据服务40

4.6 自适应推荐算法40

4.6.1 推荐算法大致流程40

4.6.2 公式介绍42

4.7 本章小结44

第5章 系统测试46

5.1 测试46

5.1.1 管理员部分功能测试46

5.1.2 作者部分功能测试52

5.1.3 游客部分功能测试54

5.2 本章小结55

结论56

参考文献57

攻读学士学位期间发表的论文和取得的科研成果58

致谢59

第1章 绪论

本章将会从论文的背景、目的与意义、论文的国内外研究现状、论文中使用的研究方法、论文的整体内容介绍这四个部分来编写绪论，以期望让读者有一个更加清晰的脉络把握本文的行文脉络。

1.1 论文的背景目的意义

1.1.1 论文背景

现今社会上越来越多的资料充斥在网络这个巨大无比的空间之中，人们进入了信息大爆炸的时代。在现在这个时代，随着网络上各种应用的不断成熟，人们几乎可以在网络上找到他们所需要的所有信息，因此人们越来越依赖于计算机网络。网络也在不知不觉间成了很多人展示自己的平台，不少人越来越习惯于在他们的博客或者各种论坛上面展示他们的作品。就像李白说过的“天生我才必有用”，世界上有几十亿人口，大部分的人在各种各样的方面上有着他们独特的闪光点。对于一个人来说，通常情况下他并不缺少独特的才能，缺少的是能够发现他才能的人。

1.1.2论文目的

在这个社会上有很多很优秀的作品，它们的作者大都在某些方向上很有才华。但由于现在社会处于信息爆炸时代，大量信息被无情的埋没了，常常发生千里马常有而伯乐难求的尴尬情况。因此，给广大作者一个展示自己作品并通过它使得伯乐找到自己的展示平台越发的重要起来。基于此，提出网络资料管理和推荐系统用于面向广大的作者，为他们提供一个展示自己作品的平台；同时面向社会上各类企业，帮助他们更方便的发掘各类有特殊才能的人才。从而使得人才和公司之间呈现出的双赢态势。

1.1.3论文意义

通过本篇论文提出的软件的各个设计层面的资料，软件的开发人员能够方便快捷的构建出网络资料管理和推荐系统，从而为广大的作者们服务，为他们提供平台展示作品。让其他的人们有一个更加好的角度来了解他们。

1.2 国内外研究状况

1.2.1 国内研究现状

本论文的题目可以拆开为两个部分进行理解，第一层，论文目的在于构建一个软件系统，第二层，该系统能够实现资料的管理和推荐功能。对于第一层，国内的软件开发领域有大量的成果，比如说在2007年由周凤林硕士发表的《电子商务下的柔性推荐系统》中就提出了一个推荐系统的实现方案。她提出的柔性推荐系统FRS(Flexible Recommendation System)这个系统包括数据库、模型库、控制引擎、推荐策略、网站系统组成。在FRS中,推荐策略和模型可以被加载、卸载。这个系统可以依据商业应用的不同来制定互不相同的推荐规则从而生成适合于本应用的控制引擎，并以此为用户进行个性化推荐。FRS不和具体的算法或应用产生耦合，拥有极强的可拓展性；这个系统能够按照不同推荐策略来产生与之相对应的推荐功能，拥有很多的柔性[1]。在对于第二层而言，目前的国内系统研究领域中还没有一个和本系统功能完全一致的成熟系统。人们无法找到拥有相同功能的系统为他们服务。国内的研究主要倾向于推荐方法的研究，而不倾向于系统的构建，也就是说大部分成果都没有办法形成一个完整的系统以供人们使用。比如说在2013年由贾大文博士发表的博士论文《社会网络资源在线共享与推荐方法研究》一文中，贾大文博士认为一方面，网络上数据是海量的，每天都会有大量新用户加入网络同时每天都会上传大量新的数据。与此同时无序是数据资源的特点，大部分资源没有结构化的数据规范,比如说视频、音频、文本和图片等,这些数据大都是多语义的而规模却是海量的,没有办法应用现在已经完全成熟的检索技术来对这些数据进行排序，所以他提出了一个基于系统使用者共同偏好的推荐算法[2]。读者从文章的描述中可以清楚的看出一个推荐方法的产生原因以及实现方法。但它也仅仅只是一个模块而已，只是系统的一小个部分。读者无法推过它构建出整个系统。而一些涉及到系统实现的研究成果中，他们的倾向性大都和本文的研究方向相差甚远。人们亦无法通过这些系统构建研究的文章中得到本系统的实现方式，最多做一些结构上的参考。因此国内的研究其实在网络资料管理和推荐系统的实现方面上有一定的空缺。

1.2.2 国外研究现状

和国内的情况有些相类似，国外的研究工作者们也将大多数的精力放在了推荐方法或者管理方法的研究上面，比如说在2003年由M Kärkkäinen,T Ala-Risku,K [Främling](http://scholar.glgoo.com/citations?user=f8U3FDcAAAAJ&hl=zh-CN&oi=sra)等人联名发表的文章”a solution to supply network information management problems?”一文中他们为供应网络管理提供了一个行之有效的方法，正如在文章中他所说的“The main reasons for this are the increasing product customisation and increasingly complex supply networks”[3]，在数据量急速增大的时候，网络资料管理也会面临一个极大的挑战，而且这个问题确是本系统构建过程中不可避免的问题，本系统必须设法解决它。可以参考这个方法来设计网络资料管理和推荐系统中的管理模块。再比如说在2008年由Walter F E, Battiston S, Schweitzer F等人联名发表的文章”A model of a trust-based recommendation system on a social network”中他们提出了”a trust-based recommendation system on a social network”[4]这给笔者研究推荐给了一个很好的启示，笔者通过可信任关系来更好的构建系统的推荐模块，同时为了减少不可信任的行为带给系统的不良影响，系统可以采用负反馈的形式来强化推荐结果的科学性。国外的研究领域有很多很漂亮的成果，可惜的是笔者并没有找到一篇针对该系统或者拥有相近功能的系统进行构建的文章。

1.3 论文的研究方法

文献研究法是根据本文的研究目的以及网络资料管理和推荐系统这个课题在网络上查阅大量的文献资料，从而更加全面的了解这篇文章应该写什么，明确系统应该怎么设计又应该怎么实现等问题，从而对课题有一个全面的准确的了解。

实证研究法为了更好的研究怎么的设计才能达到一个更加好的效果，本人在撰写该论文的同时依据该论文中提出的各种理论构建了网络资料管理和推荐系统。在不断的完善系统功能的过程中，发现系统的问题，及时纠正，理论指导实践，实践又反馈理论，让文章的不断的修改过程中呈现螺旋状前进的态势。

1.4 论文的内容及整体内容介绍

第一章绪论。该部分内容交代了论文的背景、目的和意义、国内外的研究现状、论文的研究方法、论文的总体内容结构等。

第二章需求分析。在这一个章节中交代了本系统的需求，为论文的阅读者提供一个明确的方向，知道本文的系统在什么情况下使用。

第三章概要设计。在这一个章节中交代了系统的概要设计，明确系统功能划分问题以及模块之间的相互联系。

第四章详细设计。在这一个章节中交代了系统的详细设计，从程序级的层面上交代类与类之间的关系。

第五章系统测试。在这一个章节中交代了系统的测试过程，从功能描述、输入和输出三个方面来交代测试过程。

第2章 需求分析

本章将会按照任务概述、需求规定、运行环境规定等的顺序来编制网络资料管理和推荐系统的需求分析。在任务概述中将对本系统的目标、用户特点、假定和约束，在需求规定中从功能规定、性能规定、输入输出要求、数据管理能力要求这四个方面叙述，在运行环境规定中将从系统需要的设备、支持软件、采用架构和设计模式方面来阐述。

2.1 任务概述

2.1.1 目标

为了给广大的创作爱好者提供一个展示自己的平台，特此建立个人作品展示系统用以展示作者的各类作品。吸引各类社会群体（如公司）发现作者个人价值，以帮助各位作者有一个更加宽广的发展空间。

2.1.2 用户的特点

用户特性：登录时间不确定，呈现随机性，其中部分用户会在系统中发表自己的各类作品。

管理员特性：定期对系统进行维护。

游客特性：访问呈现随机性，对作品的查找和查看呈现随机性。

2.1.3 假定和约束

第一：大部分系统的使用者懂得中文，同时使用者懂得使用浏览器浏览页面并且能够根据界面上的一系列提示来完成使用者所需要的操作。

第二：对于系统的发布者，拥有发布系统所需要的一系列软件、硬件以及一定量的系统管理人员。比如说系统发布者至少需要拥有一台可以作为服务器的计算机，支持系统存储数据的数据库。

2.2 需求规定

2.2.1 对功能的规定

该系统主要提供三类人使用分别为游客、用户和管理员。

游客：要自动对游客信息进行收集，根据他们的习惯向他们推荐不同的作品。同时游客还应该能够在系统中搜索现有作品，并对这些作品进行查看，如果感觉作品比较好，可以对作品进行评分。

用户：在系统中登录以后便赋予操作人员用户角色，可以对自己的信息进行管理。每个用户都可以将自己的作品上传到系统中，查找自己已经上传的作品，对某些不太满意的作品做出修改，对已经不需要的作品进行删除操作。查看别人的作品，对喜欢的作品评分，对某些喜爱的关注的用户添加关注。

管理员：在系统中登录以后赋予操作者管理员身份，管理员可以对系统进行数据备份、数据恢复，还可以对他们自己的信息进行管理。当系统中有新的作品时，管理员可以方便的看到，对不符合国家法律等要求的作品可以给予删除，并对作者给予适当反馈。当管理员团队需要扩充时，一级管理员可以添加二级管理员信息，当某个二级管理员辞职时，一级管理员可以对他进行权限删除。当某个管理员公出等情况时，可以设立临时管理员。当管理员回来以后收回临时管理员权限。

2.2.2 对性能的规定

为了让该系统方便易用且对未来的延续性较好，要求系统做到以下特性。

时效性：对于一般的系统功能而言，系统的反应时间不能超过3秒。对于需要大量后台操作而导致响应时间过长的功能应该给予用户适当的提示。

健壮性：在企业级（4G内存）服务器上运行时应可以容纳100个以上的客户端浏览器同时访问，服务器端程序能连续工作半年以上。在普通笔记本作为服务器时，应可以容纳5个以上的客户端浏览器访问[5]。

安全性：对于不同权限的用户给予不同的界面，减少系统内信息的泄漏，对于没有权限的用户不能让他们在系统中得到对应信息，对于有权限的用户给予权限内的服务。

正确性：系统在正常运行过程中故障率以及错误率应小于1%。

可靠性：在服务器的数据因不可抗拒力而产生错误时，系统应要有备份数据保障系统数据的可靠。

易用性：所有懂得中文并懂得使用浏览器浏览网页的用户都可以根据界面上的提示进行操作得到对应的结果。

交互性：用户能够及时的得到系统的反馈信息，并进行对应的业务。

可扩展性：预留部分接口，为未来系统的扩展提供便利。

2.2.3 输入输出要求

输入要求：对于那些需要输入的业务。系统应该提供对应的输入接口，用户可以通过浏览器中的输入框、选择框、按钮等控件来进行数据的输入。

输出要求：对于那些需要输出的业务。系统应该为那些业务提供对应的输出界面，将应该由用户了解的信息以一定形式组织在页面之中。

2.2.4 数据管理能力要求

要求能够将系统运行必要的所有数据以一定的组织方式通过对应的数据库接口存放到指定的数据库之中，在系统需要这些数据的时候通过预留的数据库接口正确读取这些数据以保证系统的正确运行。

2.3 运行环境规定

2.3.1支持软件

Tomcat软件:Tomcat是Java应用开发领域中最出名的开源服务器发布软件。它简单易用且服务器稳定性极好，不但可以用做个人学习，也可以把它用于商业产品的发布[5]。

MySQL数据库：该数据库是由MySQL AB公司开发开源数据库，它是本世纪最流行的数据库管理系统。本论文中使用MySQL 5.5作为数据库管理系统，他和以前的版本相比有较大的改进，它对触发器、存储过程和视图有了比以前版本更好的支持[6]。

2.3.2 采用结构和设计模式

B/S架构：即浏览器和服务器结构，这种结构是对C/S架构（客户端/服务器结构）的改进。在客户所持有的客户端上，应用程序不需要安装软件只需要拥有浏览器即可，在服务端时应用程序的业务得到处理并发送到客户的浏览器上[7]。

MVC设计模式：这个设计模式要求应用将模型、视图和控制器分开，从而降低模块和模块之间的耦合关系，提高应用的可复用性和拓展性。所以它并不是使用Java编程时特有的概念，是所有面向对象程序设计共同遵从的设计理念。使用MVC设计模式时程序被分为了三个部分：模型、视图和控制器，它们都只处理属于自己的任务[8]。

2.4 本章小结

本章讲述了系统的需求，从系统的任务概述、需求规定和运行环境规定这三个部分详细阐述了要设计实现的系统的需求。在任务概述中，笔者明确了系统的目标、分析了系统的使用者以及他们的特征然后还设定了一系列的假定缩小系统的访问者，方便系统边界的确定。在需求规定部分中，笔者明确了系统的主要功能、性能要求、输入输出要求、数据管理能力要求以及故障处理要求。在系统的运行环境规定部分中笔者指定了本系统所运行的环境，同时还明确了运行系统所需要的软件。

第3章 概要设计

本章将按照系统功能概述、操作及处理流程和系统数据结构设计的顺序来阐述本系统的概要设计。在系统功能概述中阐述系统结构和功能点概述，在操作及处理流程部分中将从用户共有操作、管理员操作、作者游客共有操作、作者特有操作和游客信息管理这五个方面来叙述系统中各种用户的操作以及操作流程，在系统数据结构设计中将从逻辑结构设计要点、操作与属性关系和数据结构与程序的关系这三个部分来叙述。

3.1 系统功能概述

3.1.1 系统结构

系统结构：系统分为系统功能和基础功能两个部分。其中系统功能为数据管理，基础功能部分分为管理员管理、用户管理、用户作品管理、游客信息管理等四个部分。如图3.1所示

数据管理部分主要是数据备份和数据恢复，数据备份功能在系统空闲时由管理员触发，从而保存系统数据库中的信息。而当系统出现问题，数据遭到破坏的时候，管理员则通过数据恢复功能对数据进行恢复使得系统恢复到正常工作的状态下。管理员管理部分分为二级管理员添加、二级管理员删除、临时管理员添加、临时管理员权限收回、管理员登录、管理员注销、管理员信息修改、管理员密码修改、管理员信息查询、添加密保、修改密保、找回密码、新作品提醒、不合法作品删除等功能，以期望管理员能够更方便的使用系统。用户管理部分分为用户注册、用户登录、用户信息查询、用户信息修改、密码修改、添加密保、修改密保、密码找回、注销登录、添加关注等功能，让作者能更加正常的使用系统不用记特殊的操作。用户作品管理部分分为作品上传、作品删除、作品修改、作品展示、作品查询等功能，让用户更方便的管理作品。游客信息管理分为游客信息采集和作品推荐两个部分，让游客对系统有个更加好的直观感受。

与此同时，系统还可以根据操作的共性和异性对功能点进行分类[10]。可以分类为管理员和作者共有的操作、管理员操作、作者游客共有操作、作者特有操作、游客操作等部分，本章将以这个顺序进行叙述。共有的操作包括：注册、登录系统、注销登录、用户密码取回、用户信息管理等功能。管理员操作包括：管理员系统管理、管理员作品操作。作者特有操作包括：作者作品管理。游客操作包括：游客信息采集和作品推荐。作者游客共有操作包括：作者查找、作品操作和展示。



图3.1功能分解图

3.1.2 功能点概述

具体情况如表3.1所示。

表3.1 功能点说明

|  |  |  |
| --- | --- | --- |
| 功能点名称 | 功能所属模块 | 功能点简要说明 |
| 二级管理员添加 | 管理员管理 | 为系统添加一名二级管理员 |
| 二级管理员删除 | 管理员管理 | 将系统中的某个二级管理员删除 |
| 临时管理员添加 | 管理员管理 | 为系统添加一名临时管理员 |
| 临时管理员权限收回 | 管理员管理 | 将系统中某个临时管理员的权限收回 |
| 管理员登录 | 管理员管理 | 管理员登录到系统中 |
| 管理员注销 | 管理员管理 | 管理员从系统中退出 |
| 管理员信息修改 | 管理员管理 | 管理员修改个人信息 |
| 管理员密码修改 | 管理员管理 | 管理员修改密码 |
| 管理员信息查询 | 管理员管理 | 管理员查询在权限范围内查询管理员的信息，可查到自身信息 |
| 添加密保 | 管理员管理 | 添加一个密保 |
| 修改密保 | 管理员管理 | 修改管理员现有的密保 |
| 找回密码 | 管理员管理 | 找回该账户的密码 |
| 新作品提醒 | 管理员管理 | 提醒管理员新上传到系统中的作品，让管理员对其测评 |
| 不合法作品删除 | 管理员管理 | 对于测评结果为不合法作品进行删除。 |
| 用户注册 | 用户管理 | 作者注册一个可以登录到系统中的账号 |
| 用户登录 | 用户管理 | 作者通过输入ID和密码登录系统 |
| 用户信息查询 | 用户管理 | 作者查询自己和别的作者的信息 |
| 用户信息修改 | 用户管理 | 作者修改自身信息 |
| 密码修改 | 用户管理 | 作者修改密码 |
| 添加密保 | 用户管理 | 添加一个密保 |
| 修改密保 | 用户管理 | 修改已有密保 |
| 密码找回 | 用户管理 | 作者找回该账号的密码 |
| 注销登录 | 用户管理 | 作者登出系统 |
| 添加关注 | 用户管理 | 作者将他觉得好奇或优秀的人予以关注 |
| 作品上传 | 用户作品管理 | 作者上传作品到系统数据库中 |
| 作品删除 | 用户作品管理 | 作者从系统中删除他们的作品 |
| 作品修改 | 用户作品管理 | 作者修改系统中他们私人的作品 |
| 作品展示 | 用户作品管理 | 系统展示系统中存在的合法作品 |
| 作品查询 | 用户作品管理 | 用户查询系统中的合法作品 |
| 作品推荐 | 游客信息管理 | 系统为游客推荐游客可能感兴趣的作品 |
| 游客信息采集 | 游客信息管理 | 系统采集游客的信息 |
| 数据备份 | 数据管理 | 备份数据库中的数据 |
| 数据恢复 | 数据管理 | 用备份数据恢复系统中的数据 |

3.2 操作及处理流程

3.2.1 用户共有操作

密码取回：当有用户忘记了他的密码可以通过密码找回功能找回他的密码。系统会先要求用户输入他的编号，系统查询该用户是否有设置密保，如果有则让用户输入密保信息并验证密保，如果密保验证通过系统自动返回用户密码，反之提醒用户他们输入的密保信息错误。用户如没有设置密保或者密保验证失败可以通过联系管理员来取回密码。如图3.2所示。



图3.2 密码找回功能流程图

注销：用户想要退出系统的时候，系统清理用户的登录状态。如图3.3所示。



图3.3 注销功能流程图

登录系统：用户进入系统登录页面后输入ID、密码并选择访问人员类别。系统根据访问人员类别跳转到不同类别的访客的登录处理程序中，判断访问人员ID和密码是否相匹配，如果相匹配则登录成功，否则登录失败。当用户选择匿名登录的时候，可以直接进入系统之中，系统记录游客的部分信息。登录后用户统一进入本系统的主页中。具体流程如图3.4所示。



图3.4 登录系统流程图

用户个人信息管理：首先判断系统用户是否已经成功登录到系统中，如果用户没有登录则给予用户提示要求用户先登录系统。用户登录系统以后，读取他的个人信息。用户可以选择进行信息查看、信息修改、密保修改、密码修改等操作。当用户选择进行信息查看时，系统将用户的个人信息展示给用户，用户可以选择修改个人信息，也可以直接结束。当用户选择修改个人信息时，系统会要求用户输入更新后的信息，并把它们保存到系统中。当用户选择密保修改操作时，系统先判断用户是否已经有密保，如果没有密保则让用户输入密保信息以添加新的密保，否则要求输入旧密保信息和新密保信息，当旧密保信息验证通过后更新密保信息，否则不更新。当用户选择密码修改操作时，系统要求用户输入旧密码和新密码，先验证旧密码，如果验证通过则更新密码，否则不更新。具体流程如图3.5所示。



图3.5 用户个人信息管理流程图

3.2.3 管理员操作

管理员系统操作：首先系统判断该管理员是否已经成功登录到系统中，如果没有登录就给管理员提示要求管理员先登录系统。登录后系统读取管理员信息，不同类别管理员可以看到的操作不同，一级管理员可以看到二级管理员添加、临时管理员添加、查看管理员信息、删除管理员、系统备份、系统还原等操作。二级管理员看不到二级管理员添加操作。临时管理员看不到二级管理员添加操作、临时管理员添加操作和管理员删除操作。管理员选择添加二级或临时管理员时，系统要求输入管理员信息保存到数据库中，当管理员选择查看管理员信息操作时，用户输入管理员类别，系统将会列出所有符合条件的管理员列表，管理员可以选择他所想要查看的管理员，查看完毕后，管理员可以选择是否删除管理员，如删除就使用删除管理员功能。当管理员选择删除管理员操作时，先验证用户权限是否满足，如满足则删除选中的管理员，否则予以拒绝。当管理员选择系统备份操作时，系统自动备份数据库内容。当管理员选择系统还原时，系统自动还原数据库数据。具体流程如图3.6所示。



图3.6 管理员系统操作流程图

管理员作品管理：首先系统判断该管理员是否已经成功登录到系统中，如果没有登录就给管理员提示要求管理员先登录系统。登录以后系统显示出那些还未验证是否符合国家法律法规的作品的列表。管理员通过选择随意作品来查看该作品的内容，如果管理员觉得该作品符合国家的法律法规则评定此作品为合法。系统自动给予该作品100分的初始分数。如果管理员认为该作品不符合国家的法律法规则评定为该作品不合法，系统自动删除该作品并给予作者反馈信息。具体流程如图3.7所示。



图3.7 管理员新作品管理流程图

3.2.4 作者游客共有操作

作品查找及展示：用户可以通过输入关键字或者作者编号来查询作品，系统根据用户的输入来查找并显示作品列表，用户选择作品后，系统展示该作品内容。用户查看完作品以后可以选择是否给该作品评分，如果对该作品评分，则系统根据用户选择的分数通过一定的算法计算出作品的新分数并对作品的评分予以更新。具体流程如图3.8所示。



图3.8 作品查找及展示流程图

作者查找：首先判断是否已经登录，如果未登录则用户可以通过输入作者编号来查询作者，如果已经登录则用户可以选择“我关注的人”或者输入作者编号来查询作者，系统给出对应的作者列表。用户选择作者后系统读出该作者的信息列出他的信息以及作品信息。用户可以选择进行查看作品操作或者选择关注/取消关注该作者。如果用户选择查看作品操作时，系统展示该作品。当用户选择关注/取消关注该作者时，系统更新作者间的关系。具体流程如图3.9所示。



图3.9 作者查找流程图

3.2.4 作者特有操作

作者注册：用户输入作者信息后系统将作者的信息添加到系统中并将一个唯一化的编号给予作者，作者就可以通过该编号以及他的密码来登录系统。如图3.10所示。



图3.10 作者注册流程图

作者作品管理：首先判断作者是否已经登录到系统中，若未登录则给予作者提示要求先登录系统。当登录系统以后作者可以选择以下操作：作品上传操作、作品修改操作和作品删除操作。当作者选择作品上传操作时系统要求作者输入作品名称、作品类别、作品相对应的文件，然后将这些数据存储到数据库中。当作者选择作品修改的时候，系统列出作者的个人作品，作者选择作品后输入作品名称、作品类别、作品文件，然后系统更新数据库。当作者选择作品删除时，系统列出作者的个人作品，作者选择作品后系统对该作品予以删除。具体流程如图3.11所示。



图3.11 作者作品管理流程图

3.2.5 游客信息管理

作品推荐：用户以匿名方式登录到系统之中的时候，系统记录该用户的IP地址以及它在系统中查找过的作品信息。当该访客第二次访问系统时，系统读取数据库中对应的数据将该访客可能感兴趣的作品列出，并根据评分高低排序，以保证优质作品排列在前。流程如图3.12。



图3.12 作品推荐流程图

3.3 系统数据结构设计

3.3.1逻辑结构设计要点

为了简便的构建本系统数据之间里关系，所以采用关系数据模型来构建。关系可以描述现实社会中实体之间的联系[11]。基于此设计如下表存储系统中所需要用到的信息,这些表需要符合第一范式[12]。

Administrator：存储管理员的信息。如表3.2所示。

表3.2 管理员信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 程序中标识符 | 数据类型 | 数据库中标识符 | 数据库中数据类型 | 描述 | 约束 |
| 1 | aid | int | aid | int(5) | 管理员编号，自增长 | 主键 |
| 2 | type | String | type | char(1) | 管理员类别，”1”为一级管理员,”2”为二级管理员,”3”为临时管理员 | -- |
| 3 | password | String | password | varchar(16) | 管理员密码，8到16位 | -- |
| 4 | contact | String | contact | char(11) | 管理员联系方式，11位手机号 | -- |

AdministratorPasswordProtection:存储管理员密保信息。如表3.3所示。

表3.3 管理员密保表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 程序中标识符 | 数据类型 | 数据库中标识符 | 数据库中数据类型 | 描述 | 约束 |
| 1 | aid | int | aid | int(5) | 管理员编号 | 外键，受administrator表约束 |
| 2 | question1 | String | question1 | varchar(100) | 密保问题1 | -- |
| 3 | answer1 | String | answer1 | varchar(100) | 密保答案1 | -- |
| 4 | question2 | String | question2 | varchar(100) | 密保问题2 | -- |
| 5 | answer2 | String | answer2 | varchar(100) | 密保答案2 | -- |
| 6 | question3 | String | question3 | varchar(100) | 密保问题3 | -- |
| 7 | answer3 | String | answer3 | varchar(100) | 密保回答3 | -- |

Author：存储作者信息。如表3.4所示

表3.4 作者信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 程序中标识符 | 数据类型 | 数据库中标识符 | 数据库中数据类型 | 描述 | 约束 |
| 1 | id | int | id | int(5) | 作者编号，自增长 | 主键 |
| 2 | name | String | name | char(10) | 作者姓名 | -- |
| 3 | password | String | password | varchar(16) | 作者密码，8到16位 | -- |
| 4 | contact | String | contact | char(11) | 联系方式，手机号 | -- |
| 5 | resume | String | resume | blob | 简历文件，二进制流。 | -- |
| 6 | public\_information | Boolean | public\_information | tinyint(1) | 是否公开个人信息 | -- |

Authorandauthor:存储作者间的关注信息。如表3.5所示。

表3.5 作者关注信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 程序中标识符 | 数据类型 | 数据库中标识符 | 数据库中数据类型 | 描述 | 约束 |
| 1 | fid | int | fid | int(5) | 关注动作主动者的编号 | 外键受Author表约束 |
| 2 | sid | int | sid | int(5) | 关注动作被动作的编号 | 外键受Author表约束 |

AuthorPasswordProtection:存储作者密保信息。如表3.6所示。

表3.6 作者密保信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 程序中标识符 | 数据类型 | 数据库中标识符 | 数据库中数据类型 | 描述 | 约束 |
| 1 | id | int | id | int(5) | 作者编号 | 外键，受Author表约束 |
| 2 | question1 | String | question1 | varchar(100) | 密保问题1 | -- |
| 3 | answer1 | String | answer1 | varchar(100) | 密保答案1 | -- |
| 4 | question2 | String | question2 | varchar(100) | 密保问题2 | -- |
| 5 | answer2 | String | answer2 | varchar(100) | 密保答案2 | -- |
| 6 | question3 | String | question3 | varchar(100) | 密保问题3 | -- |
| 7 | answer3 | String | answer3 | varchar(100) | 密保回答3 | -- |

Visitor:存储游客信息。如表3.7所示。

表3.7 游客信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 程序中标识符 | 数据类型 | 数据库中标识符 | 数据库中数据类型 | 描述 | 约束 |
| 1 | ip | String | ip | char(15) | 访客IP地址 | -- |
| 2 | keyword | String | keyword | varchar(200) | 关键字 | -- |

Works:用于存储作品信息。如表3.8所示。

表3.8 作品信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 程序中标识符 | 数据类型 | 数据库中标识符 | 数据库中数据类型 | 描述 | 约束 |
| 1 | wid | int | wid | int(5) | 作品编号 | 主键 |
| 2 | name | String | name | varchar(50) | 作品名称 | -- |
| 3 | type | String | type | varchar(200) | 作品类别”picture”指图片”audio”指音频”video”指视频”document”指文档 | -- |
| 4 | count | int | count | int(11) | 作品评分，0-100分，0分指未测评的新作品。 | -- |
| 5 | id | int | id | int(5) | 作者编号 | 外键，受Author表约束 |
| 6 | utime | Date | utime | date | 上传时间 | -- |
| 7 | ctime | Date | ctime | date | 修改时间 | -- |
| 8 | -- | -- | file | longblob | 作品文件 | -- |

3.3.2 操作与属性关系

操作对管理员信息的影响如表3.9所示。

表3.9 操作对管理员信息影响表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 操作名称 | 操作描述 | aid | type | password | contact |
| findById(type,obj) | 通过编号查找记录 | ※ | ※ | ※ | ※ |
| doInsert(type,obj) | 插入记录 | ※ | ※ | ※ | ※ |
| doDelete(type,obj) | 删除记录 | ※ | ※ | ※ | ※ |
| doUpdate(type,obj) | 更新记录 | -- | -- | ※ | ※ |
| findByKeyword(type,obj) | 通过关键字查询记录 | ※ | ※ | ※ | ※ |
| findByAuthorId(id) | 通过作者编号查询记录 | -- | -- | -- | -- |
| getFile(obj) | 从数据库获取文件 | -- | -- | -- | -- |
| setFile(file) | 添加文件到数据库 | -- | -- | -- | -- |
| findByFid(fid) | 通过第一个作者编号查询记录 | -- | -- | -- | -- |

操作对管理员密保信息的影响如表3.10所示。

表3.10 操作对管理员密保信息影响表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 操作名称 | 操作描述 | aid | question1 | answer1 | qusetion2 | answer2 | quwetion3 | answer3 |
| findById(type,obj) | 通过编号查找记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ |
| doInsert(type,obj) | 插入记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ |
| doDelete(type,obj) | 删除记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ |
| doUpdate(type,obj) | 更新记录 | -- | ※ | ※ | ※ | ※ | ※ | ※ |
| findByKeyword(type,obj) | 通过关键字查询记录 | -- | -- | -- | -- | -- | -- | -- |
| findByAuthorId(id) | 通过作者编号查询记录 | -- | -- | -- | -- | -- | -- | -- |
| getFile(obj) | 从数据库获取文件 | -- | -- | -- | -- | -- | -- | -- |
| setFile(file) | 添加文件到数据库 | -- | -- | -- | -- | -- | -- | -- |
| findByFid(fid) | 通过第一个作者编号查询记录 | -- | -- | -- | -- | -- | -- | -- |

操作对作者信息的影响如表3.11所示。

表3.11 操作对作者信息影响表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 操作名称 | 操作描述 | id | name | password | contact | resume | public\_information |
| findById(type,obj) | 通过编号查找记录 | ※ | ※ | ※ | ※ | ※ | ※ |
| doInsert(type,obj) | 插入记录 | ※ | ※ | ※ | ※ | ※ | ※ |
| doDelete(type,obj) | 删除记录 | ※ | ※ | ※ | ※ | ※ | ※ |
| doUpdate(type,obj) | 更新记录 | -- | ※ | ※ | ※ | ※ | ※ |
| findByKeyword(type,obj) | 通过关键字查询记录 | ※ | ※ | ※ | ※ | ※ | ※ |
| findByAuthorId(id) | 通过作者编号查询记录 | -- | -- | -- | -- | -- | -- |
| getFile(obj) | 从数据库获取文件 | -- | -- | -- | -- | -- | -- |
| setFile(file) | 添加文件到数据库 | -- | -- | -- | -- | -- | -- |
| findByFid(fid) | 通过第一个作者编号查询记录 | -- | -- | -- | -- | -- | -- |

操作对作者关注信息的影响如表3.12所示。

表3.12 操作对作者关注信息影响表

|  |  |  |  |
| --- | --- | --- | --- |
| 操作名称 | 操作描述 | fid | sid |
| findById(type,obj) | 通过编号查找记录 | ※ | ※ |
| doInsert(type,obj) | 插入记录 | ※ | ※ |
| doDelete(type,obj) | 删除记录 | ※ | ※ |
| doUpdate(type,obj) | 更新记录 | -- | -- |
| findByKeyword(type,obj) | 通过关键字查询记录 | -- | -- |
| findByAuthorId(id) | 通过作者编号查询记录 | -- | -- |
| getFile(obj) | 从数据库获取文件 | -- | -- |
| setFile(file) | 添加文件到数据库 | -- | -- |
| findByFid(fid) | 通过第一个作者编号查询记录 | ※ | ※ |

操作对作者密保信息的影响如表3.13所示。

表3.13 操作对作者密保信息影响表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 操作名称 | 操作描述 | id | question1 | answer1 | qusetion2 | answer2 | quwetion3 | answer3 |
| findById(type,obj) | 通过编号查找记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ |
| doInsert(type,obj) | 插入记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ |
| doDelete(type,obj) | 删除记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ |
| doUpdate(type,obj) | 更新记录 | -- | ※ | ※ | ※ | ※ | ※ | ※ |
| findByKeyword(type,obj) | 通过关键字查询记录 | -- | -- | -- | -- | -- | -- | -- |
| findByAuthorId(id) | 通过作者编号查询记录 | -- | -- | -- | -- | -- | -- | -- |
| getFile(obj) | 从数据库获取文件 | -- | -- | -- | -- | -- | -- | -- |
| setFile(file) | 添加文件到数据库 | -- | -- | -- | -- | -- | -- | -- |
| findByFid(fid) | 通过第一个作者编号查询记录 | -- | -- | -- | -- | -- | -- | -- |

操作对游客信息的影响如表3.14所示。

表3.14 操作对游客信息影响表

|  |  |  |  |
| --- | --- | --- | --- |
| 操作名称 | 操作描述 | ip | keyword |
| findById(type,obj) | 通过编号查找记录 | ※ | ※ |
| doInsert(type,obj) | 插入记录 | ※ | ※ |
| doDelete(type,obj) | 删除记录 | ※ | ※ |
| doUpdate(type,obj) | 更新记录 | ※ | ※ |
| findByKeyword(type,obj) | 通过关键字查询记录 | -- | -- |
| findByAuthorId(id) | 通过作者编号查询记录 | -- | -- |
| getFile(obj) | 从数据库获取文件 | -- | -- |
| setFile(file) | 添加文件到数据库 | -- | -- |
| findByFid(fid) | 通过第一个作者编号查询记录 | -- | -- |

操作对作者作品信息的影响如表3.15中所示。

表3.15 操作对作品信息影响表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 操作名称 | 操作描述 | wid | name | type | count | id | utime | ctime | file |
| findById(type,obj) | 通过编号查找记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ | -- |
| doInsert(type,obj) | 插入记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ | -- |
| doDelete(type,obj) | 删除记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ | ※ |
| doUpdate(type,obj) | 更新记录 | -- | ※ | ※ | ※ | -- | -- | ※ | -- |
| findByKeyword(type,obj) | 通过关键字查询记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ | -- |
| findByAuthorId(id) | 通过作者编号查询记录 | ※ | ※ | ※ | ※ | ※ | ※ | ※ | -- |
| getFile(obj) | 从数据库获取文件 | ※ | ※ | -- | -- | -- | -- | -- | ※ |
| setFile(file,obj) | 添加文件到数据库 | ※ | -- | -- | -- | -- | -- | -- | ※ |
| findByFid(fid) | 通过第一个作者编号查询记录 | -- | -- | -- | -- | -- | -- | -- | -- |

注：※：表示操作对该表的指定数据项产生影响。

3.3.3数据结构与程序的关系

操作方法参数解释：

Type: 1,Administrator; 2,AdministratorPasswordProtection; 3,Author; 4,AuthorPassword Protection; 5,Log; 6,Visitor; 7,Works; 8Authorandauthor；

obj:对应vo；

id:作者编号；

file:需要存储的作品；

fid:第一个作者编号；

当系统需要对数据进行操作的时候，调用对应的操作将参数传给对应方法调用程序中对应的数据结构收集数据然后通过数据库接口对数据库中对应的数据进行操作。

3.4 本章小结

在本章中对系统进行了概要设计，明确了系统的边界。将系统划分了教为清晰的结构层次，划分了各个功能点同时明确了这些功能点的功能。设计了大部分业务的大概处理流程，设计了数据的逻辑结构和物理结构，还明确了系统各个模块之间的接口问题。总之本章的内容描绘出了系统的轮廓为下一步的详细设计提供了便利。

第4章 详细设计

本章从数据存取接口、注册登录注销、信息管理、作品管理和数据服务这五个方面来叙述本系统的详细设计。在每个部分中都解释了一系列的类，分别从程序描述、功能、输入项、输出项、逻辑流程这五个方面来详细描述该类，使得读者对本文中程序级设计有一个更加清晰的认识。

4.1 数据存取接口

共性定义：

\* type == 1 Administrator

\* type == 2 AdministratorPasswordProtection

\* type == 3 Author

\* type == 4 AuthorPasswordProtection

\* type == 6 Visitor

\* type == 7 Works

\* type == 8 Authorandauthor

1、 factory文件getProxy()设计说明

程序描述：该方法为静态方法，用于获取下一层代理。

功能：输入：无；处理：新建并返回下一层代理；输出：无。

流程逻辑：新建并且返回PWSS\_DAO\_proxy()。

2、 proxy文件findById(int type,Object obj)设计说明

程序描述：通过type选定vo，将obj转换为对应的vo实例通过vo实例中的编号来查找记录。

功能：输入：type,obj ；处理：调用impl文件中的findById(int type,Object obj)；输出：下一层返回值。

输入项：Type:见共性定义；Obj:所要查询的记录对应的VO的实例。

输出项：下一层的返回值。

流程逻辑：调用impl文件中的findById(int type,Object obj)，将本方法中的参数原样传递过去，返回impl文件中的findById(int type,Object obj)的返回值。

3、 proxy文件doInsert(int type,Object obj)设计说明

程序描述：通过type选定vo，将obj转换为对应的vo实例，根据vo实例中向数据库中插入记录。

功能：输入：type,obj ；处理：先调用impl文件中的findById(int type,Object obj)，如果返回值为true则调用impl文件中的doInsert(int type,Object obj)。否则返回0；输出：下一层返回值或0。

输入项：Type:见共性定义；Obj:所要查询的记录对应的VO的实例。

输出项：下一层返回值或0。

流程逻辑：先调用impl文件中的findById(int type,Object obj)，将本方法中的参数原样传递过去,如果返回值为true调用impl文件中的doInsert(int type,Object obj)，将本方法中的参数原样传递过去，返回impl文件中的doInsert(int type,Object obj)的返回值。否则返回0.

4、 proxy文件doDelete(int type,Object obj)设计说明

程序描述：通过参数type选定vo，将参数obj转换成type对应的vo实例，然后以vo实例为依据删除数据库中数据。

功能：输入：type,obj；处理：先调用impl文件中的findById(int type,Object obj)，如果返回值为true则调用impl文件中的doDelete(int type,Object obj)。否则返回false；输出：下一层返回值或false。

输入项：Type:见共性定义；Obj:所要查询的记录对应的VO的实例。

输出项：下一层返回值或false。

流程逻辑：先调用impl文件中的findById(int type,Object obj)，将本方法中的参数原样传递过去,如果返回值为true调用impl文件中的doDelete(int type,Object obj)，将本方法中的参数原样传递过去，返回impl文件中的doDelete(int type,Object obj)的返回值。否则返回false

5、 proxy文件doUpdate(int type,Object obj)设计说明

程序描述：通过type选定vo，将obj转换为对应的vo实例，根据vo实例更新数据库中数据。

功能：输入：type,obj；处理：先调用impl文件中的findById(int type,Object obj)，如果返回值为true则调用impl文件中的doUpdate(int type,Object obj)。否则返回false；输出：下一层返回值或false

输入项：Type:见共性定义。Obj:所要查询的记录对应的VO的实例。

输出项：下一层返回值或false

流程逻辑：先调用impl文件中的findById(int type,Object obj)，将本方法中的参数原样传递过去,如果返回值为true调用impl文件中的doUpdate(int type,Object obj)，将本方法中的参数原样传递过去，返回impl文件中的doUpdate(int type,Object obj)的返回值。否则返回false

6、 proxy文件findByKeyword(int type,String keyword)设计说明

程序描述：通过参数type选定vo，通过参数keyword在数据库中依据vo实例从对应的数据库表中查询数据。

功能：输入：type,obj；处理：则调用impl文件中的findByKeyword(int type,String keyword)。输出：下一层返回值

输入项：Type:见共性定义。Keyword：关键字。

输出项：下一层的返回值。

流程逻辑：调用impl文件中的findByKeyword(int type,String keyword)，将本方法中的参数传递过去，返回impl文件中的findByKeyword(int type,String keyword)的返回值。

7、 proxy文件findByAuthorId(int id)设计说明

程序描述：通过作者的id来查询数据库中的works表信息。

功能：输入：作者编号；处理：则调用impl文件中的findByAuthorId(int id)。否则返回null。输出：下一层返回值

输入项：Id:author中的id.

输出项：下一层返回值

流程逻辑：调用impl文件中的findByAuthorId(int id)，将本方法中的参数原样传递过去，返回impl文件中的findByAuthorId(int id)的返回值。

8、 proxy文件getFile(Object obj)设计说明

程序描述：通过作者的id来查询数据库中的works表信息。

功能：输入：works的实例；处理：先调用impl文件中的findById(int type,Object obj)，如果返回值为true则调用impl文件中的getFile(Object obj)。输出：下一层返回值

输入项：works的实例

输出项：下一层返回值

流程逻辑：先调用impl文件中的findById(int type,Object obj)，将本方法中的参数原样传递过去,type=7,如果返回值为true调用impl文件中的getFile(Object obj)，将本方法中的参数原样传递过去，返回impl文件中的getFile(Object obj)的返回值。

9、 proxy文件setFile(File file,Works works)设计说明

程序描述：将file文件转存到works实例所对应的数据库记录中

功能：输入：file文件、works实例处理：先调用impl文件中的findById(int type,Object obj)，如果返回值为true则调用impl文件中的setFile(File file,Works works)。输出：下一层返回值

输入项：file文件、works实例

输出项：下一层返回值

流程逻辑：先调用impl文件中的findById(int type,Object obj)，将本方法中的参数原样传递过去type=7,如果返回值为true调用impl文件中的setFile(File file,Works works)，将本方法中的参数传递过去，返回impl文件中的setFile(File file,Works works)的返回值。

10、 proxy文件findByFid(int fid)设计说明

程序描述：通过第一个作者的编号来查找作者间的关注关系。

功能：输入：file文件、works实例；处理：调用impl文件中的findByFid(int fid)；输出：下一层返回值

输入项：file文件、works实例

输出项：下一层返回值

流程逻辑：调用impl文件中的findByFid(int fid)，将本方法中的参数原样传递过去，返回impl文件中的findByFid(int fid)的返回值。

11、 impl文件findById(int type,Object obj)设计说明

程序描述：通过type选定vo，将obj转换为对应的vo实例通过vo实例中的编号来查找记录。

功能：输入：type,obj；处理：通过type选定vo，将obj转换为对应的vo实例通过vo实例中的编号来查找记录并更新vo实例。输出：vo实例或空

输入项：Type:见共性定义。Obj:所要查询的记录对应的VO的实例。

输出项：如果数据库中不存在该记录则返回空，否则返回vo实例

流程逻辑：先根据switch(type)来选择对应的逻辑。将obj转化为对应的vo实例，编写sql语句，将vo中需要的属性传给sql语句，调用数据库查询语句，更新vo实例。返回vo实例。

12、 impl文件doInsert(int type,Object obj)设计说明

程序描述：通过type选定vo，将obj转换为对应的vo实例，根据vo实例中向数据库中插入记录。

功能：输入：type,obj；处理：通过type选定vo，将obj转换为对应的vo实例根据vo实例中向数据库中插入记录。输出：0/1/编号

输入项：Type:见共性定义。Obj:所要查询的记录对应的VO的实例。

输出项：当数据插入失败时返回0，插入成功时为1，当vo类型为administrator/author/works中一个时返回编号。

流程逻辑：先根据switch(type)来选择对应的逻辑。将obj转化为对应的vo实例，编写sql语句，将vo中需要的属性传给sql语句，调用数据库更新语句，更新数据库中数据。返回0/1/编号。

13、 impl文件doDelete(int type,Object obj)设计说明

程序描述：通过type选定vo，将obj转换为对应的vo实例，通过vo实例中的编号来删除记录。

功能：输入：type,obj；处理：通过type选定vo，将obj转换为对应的vo实例通过vo实例中的编号来删除记录。输出：true/false

输入项：Type:见共性定义。Obj:所要查询的记录对应的VO的实例。

输出项：当操作删除成功是返回true反之false.

流程逻辑：先根据switch(type)来选择对应的逻辑。将obj转化为对应的vo实例，编写sql语句，将vo中需要的属性传给sql语句，调用数据库更新语句，删除数据库中的记录。当操作删除成功是返回true反之false。

14、 impl文件doUpdate(int type,Object obj)设计说明

程序描述：通过type选定vo，将obj转换为对应的vo实例通过vo实例来更新数据库中记录。

功能：输入：type,obj；处理：通过type选定vo，将obj转换为对应的vo实例通过vo实例中的编号来查找并更新数据库记录。输出：true/false

输入项：Type:见共性定义。Obj:所要查询的记录对应的VO的实例。

输出项：当数据更新成功是返回true反之false.

流程逻辑：先根据switch(type)来选择对应的逻辑。将obj转化为对应的vo实例，编写sql语句，将vo中需要的属性传给sql语句，调用数据库更新语句，更新数据库中数据。当数据更新成功是返回true反之false.

15、 impl文件findByKeyword(int type,String keyword)设计说明

程序描述：通过type选定vo通过keyword来查找数据库中的记录更新vo实例。

功能：输入：type,keyword；处理：通过type选定vo，通过keyword查找数据库中记录，把结果中的每一条记录存到一系列vo实例中，把vo实例添加到一个list中返回list。输出：list

输入项：Type:见共性定义。Obj:所要查询的记录对应的VO的实例。

输出项：list，其中包含由type指定的vo的实例

流程逻辑：先根据switch(type)来选择对应的逻辑。编写sql语句，通过keyword查询数据库，将返回集中的每一条记录存放到vo实例中，将vo实例添加到list中返回list。

16、 impl文件findByAuthorId(int id)设计说明

程序描述：根据works的id字段中的值来查询数据库中works表中的数据。

功能：输入：type,obj；处理：将id传到数据库中查询works表，读取对应数据并返回。输出：list

输入项：Works表中的id字段

输出项：list，其中包含一系列works的实例

流程逻辑：编写sql语句，将id属性传给sql语句，调用数据库查询语句，更新works实例添加到list中。返回list。

17、 impl文件getFile(Object obj)设计说明

程序描述：将obj转换为works实例，通过works实例中的作品编号来查找记录，从而得到works对应的file。

功能：输入：obj；处理：将obj转换为works实例，通过works实例中的作品编号来查找记录，从而得到works对应的file。输出：更新后的works实例

输入项：Obj:works的实例。

输出项：更新后的works实例

流程逻辑：将obj转化为works实例，编写sql语句，将works中编号传给sql语句，调用数据库查询语句，获取works中的file，文件形式存储。返回更新后的works实例

18、 impl文件setFile(File file,Works works)设计说明

程序描述：将file对应的文件存到数据库works表对应的记录中。

功能：输入：file，works；处理：通过type选定vo，将obj转换为对应的vo实例通过vo实例中的编号来查找记录并更新vo实例。输出：vo实例

输入项：File：所要存储的文件；Works:对应的作品记录。

输出项：当获取file成功时返回true，否则返回false。

流程逻辑：编写sql语句，将works实例中的编号和file的内容传给sql语句，调用数据库更新语句，更新数据库数据。当数据更新成功返回true更新失败则返回false。

19、 impl文件findByFid(int fid)设计说明

程序描述：通过第一个作者的编号来查询作者间的关系。

功能：输入：fid；处理：通过第一个作者的编号来查询作者间的关系。输出：list

输入项：Fid:第一个作者编号。

输出项：List：包含一系列authorandauthor实例的集合

流程逻辑：编写sql语句，将fid传给sql语句，调用数据库查询语句，将得到的返回集的每一条记录存到一个authorandauthor实例中，将所有的实例添加到list集合中，返回list集合。

4.2注册登录注销

1、 register设计说明

程序描述：作者注册

功能：输入：作者姓名、密码、联系方式；处理：将作者信息添加到数据库中后返回一个用于唯一标识该作者的编号。 输出：作者编号

输入项：作者姓名：10个字符以内；密码：8到16位；联系方式：11位手机号

输出项：唯一标识该作者的编号

流程逻辑：由一个JSP页面收集用户输入的作者姓名、密码、联系方式等信息[13]，然后调用author\_register\_servlet，将这些信息传给该servlet后，通过调用PWSS\_DAO\_factory.getProxy().doInsert(3, author)将信息存到数据库中，返回作者的编号。

2、 login设计说明

程序描述：登录系统，包括administrator\_login,author\_login,visitor\_login。分别对应管理员登录、作者登录和访客登录。

功能：输入：对于administrator\_login和author\_login时系统要求用户输入用户账号编号和帐户密码，然后选择访客类型。对于visitor\_login只需要选择访客类型；处理：验证用户输入的信息和数据库中存储的信息是否一致，如果一致则登录成功向session中对应的账号信息。当用户为管理员或作者时删除之前session中的访客账号。输出：对应提示

输入项：administrator\_login、author\_login：用户编号、密码并选择对应的访客类型visitor\_login：访客类型。

输出项：提示成功或提示失败。

流程逻辑：由一个JSP页面收集用户输入的用户编号、密码、访客类型等信息，然后根据访客类型调用author\_login\_servlet、administrator\_login、visitor\_login中的一个，将这些信息传给该servlet后，通过调用PWSS\_DAO\_factory.getProxy().findById (type,obj)来验证用户输入信息和数据库中存储的数据是否一致，若一直登录成功，反之登录失败。其中type的值为1（管理员），3（作者），6（访客），obj为对应vo的实例（需要用上个页面传过来的参数对其初始化）。

3、 logout设计说明

程序描述：注销，包括administrator\_logout,author\_ logout,visitor\_ logout，它们分别对应管理员注销功能、作者注销功能和访客注销功能。

功能：输入：无；处理： 将session中的账号信息清除。输出：对应提示

输入项：无

输出项：提示成功或提示失败。

流程逻辑：将session中的账户登录信息清空。

4.3 信息管理

1、 information\_show设计说明

程序描述：展示用户信息，包括管理员和作者。

功能：输入：无；处理：将对应的用户的信息展示出来。输出：将用户信息以页面形式组织并输出。

输入项：无

输出项：将用户信息以页面形式组织并输出。

流程逻辑：通过一个JSP页面收集用户想要查看的用户编号以及类别，转到servlet中调用PWSS\_DAO\_factory.getProxy().doInsert(type, obj)获得用户信息，将用户信息以页面形式组织反馈给查询的用户。其中type：1（管理员）3（用户）。Obj为对应的vo的实例,用用户输入的编号为它们初始化。

2、 information\_modify设计说明

程序描述：修改用户个人信息，包括管理员和作者。

功能：输入：用户姓名、用户联系方式、是否公开信息。处理：将输入的信息更新到对应的数据库数据记录中；输出：提示成功或失败

输入项：用户姓名：10个字符以内；联系方式：11位手机号；是否公开信息：单选框，boolean类型。

输出项：提示成功或失败

流程逻辑：读取session中的账户信息，作为初始值提供给用户，用户对这些信息进行修改后，系统将这些信息提交到author\_information\_servlet或administrator\_information\_serblet中（根据用户类型选择）。调用PWSS\_DAO\_factory.getProxy().doUpdate(type,obj)来更新数据库中数据。

3、 information\_search设计说明

程序描述：用户信息查询，包括管理员和作者。

功能：输入：用户关系、用户类别、管理员种类；处理：根据输入到对应的数据库中读取数据；输出：用户列表

输入项：在作者中，作者关系为关注对象或者作者的编号。在管理员中，管理员种类为“1”，“2”，“3”分别代表一级管理员、二级管理员、临时管理员。

输出项：用户列表

流程逻辑：先检查登录状态，如果未登录只能通过输入作者编号来查询。如果为作者可以选择关注的人或者作者编号来查询。如果是用户为管理员则通过管理员的级别来进行查询。系统将信息传到administrator\_search\_servlet或author\_search\_servlet中调用PWSS\_DAO\_factory.getProxy().findByKeyword(type,keyword)查询数据库中的数据，并把list返回给jsp页面

4、 password\_modify设计说明

程序描述：密码修改，包括管理员和作者。

功能：输入：旧密码、新密码；处理：验证用户的旧密码和系统数据库中所提前存储的数据是否一致，如果密码一致就将用户的新密码更新到系统数据库中，反之则更新失败。输出：成功或失败提示

输入项：旧密码：8到16位；新密码：输入两次，8到16位

输出项：成功或失败提示

流程逻辑：先检查登录状态，如果未登录提示先登录系统。登录后用一个JSP页面收集用户的旧密码和新密码。发送到对应的administrator\_password\_modify\_servlet或author\_password\_modify\_servlet中。调用PWSS\_DAO\_factory.getProxy().findById(type,obj)获得数据库中对应用户的信息，验证旧密码和数据库中数据是否一致，如果不一致则给予失败提示，否则将新的密码通过调用PWSS\_DAO\_factory.getProxy().doUpdate(type,obj)更新数据库数据，给予成功提示。Type:1(管理员)，3（作者）。Obj通过session中的登录信息转换成的vo来初始化。

5、 password\_retrieve设计说明

程序描述：密码取回，包括管理员和作者。

功能：输入：密保回答，要取回密码的用户编号；处理：先验证该用户有没有密保信息，如果没有密保信息就提示用户使用别的渠道取回密码。如果有密保验证密保后告诉用户密码。 输出：成功或失败提示

输入项：用户编号：用户用于登录系统的ID号。密保问题回答：50个字符以内。

输出项：成功或失败提示

流程逻辑：先由一个JSP页面收集用户想要取回的账户的编号，然后到数据库中查询该用户是否已经设置密保（调用PWSS\_DAO\_factory.getProxy().findById(type,obj)），如果有密保则将密保问题以一个JSP页面形式输出给用户，用户在该页面上输入密保回答。系统验证密保回答和数据库中是否一致，如果一致则返回用户密码，（调用PWSS\_DAO\_factory.getProxy().findById(type,obj)）否则返回密保回答错误。

Type:1(管理员帐户)，2（管理员密保），3（作者账号），4（作者密保）

6、 password\_protection\_add设计说明

程序描述：密保添加，包括管理员和作者。

功能：输入：选择系统预设的密保问题，根据提示输入密保问题的回答；处理：将密保问题和密保回答存储到数据库中；输出：成功或失败提示

输入项：密保问题，预设7个密保问题，用户只能选择。密保问题回答：50个字符以内。

输出项：成功或失败提示

流程逻辑：先由一个JSP页面收集用户设置的密保问题和密保答案信息，将这些信息传到对应的servlet中（管理员：administrator\_password\_protection\_add\_servlet,作者：author\_password\_protection\_add\_servlet），将这些数据存储到数据库对应的表中（调用PWSS\_DAO\_factory.getProxy().findById(type,obj)）。Type:1(管理员帐户)，2（管理员密保），3（作者账号），4（作者密保）。

7、 password\_protection\_modify设计说明

程序描述：密保修改，包括管理员和作者。

功能：输入：选择密保问题，输入密保回答，输入旧密保回答；处理：将旧的密保回答和数据库中数据想匹配如果验证通过将新密保问题和新密保回答更新到数据库中，否则失败；输出：成功或失败提示

输入项：新密保问题，预设7个密保问题，用户只能选择。新密保问题回答：50个字符以内。旧密保问题回答：50个字符以内。

输出项：成功或失败提示

流程逻辑：先由一个JSP页面输出用户的旧密保回答要求用户输入就密保答案和新密保问题、新密保答案等数据，将这些信息传到对应的servlet中（管理员：administrator\_password\_protection\_modify\_servlet,作者：author\_password\_protection\_modify\_servlet），验证旧密保答案和数据库中密保答案是否相匹配，如果匹配将这些数据存储到数据库对应的表中（调用PWSS\_DAO\_factory.getProxy().findById(type,obj)），否则失败。Type:1(管理员帐户)，2（管理员密保），3（作者账号），4（作者密保）。

4.4 作品管理

1、 works\_upload设计说明

程序描述：作品上传

功能：输入：作品名、作品类别、作品文件；处理：将作品存到数据库中；输出：成功或失败提示

输入项：作品名：50字以内；作品类型：选单形式分为”picture”图片、”audio”音频、”video”视频、”document”文档；作品文件：文件本身。

输出项：成功或失败提示

流程逻辑：验证作者是否已经登录，如未登录提示先登录系统。由一个JSP页面收集用户输入的作品名、作品类别以及作者选择的文件，将这些数据传到works\_upload\_servlet中调用PWSS\_DAO\_factory.getProxy().doInsert(8,works)将作品信息存到数据库中在调用PWSS\_DAO\_factory.getProxy().setFile(file,works)将作品文件的内容存储数据库中。

2、 works\_modify设计说明

程序描述：作品修改

功能：输入：作品名、作品类别、作品文件；处理：将作品更新数据库中；输出：成功或失败提示

输入项：作品名：50字以内；作品类型：选单形式分为”picture”图片、”audio”音频、”video”视频、”document”文档；作品文件：文件本身。

输出项：成功或失败提示

流程逻辑：验证作者是否已经登录，如未登录提示先登录系统。先有一个页面列出该作者的所有作品，由作者选择要修改的作品，由一个JSP页面收集用户输入的作品名、作品类别以及作者选择的文件[14]，将这些数据传到works\_modify\_servlet中调用PWSS\_DAO\_factory.getProxy().doUpdate (8,works)将作品信息更新到数据库中再调用PWSS\_DAO\_factory.getProxy().setFile(file,works)将数据库的作品文件进行更新。

3、 works\_delete设计说明

程序描述：作品删除

功能：输入：要删除的作编；处理：将作品删除；输出：成功或失败提示

输入项：在列表中选择其中一个个人作品。

输出项：成功或失败提示

流程逻辑：验证作者是否已经登录，如未登录提示先登录系统。先有一个页面列出该作者的所有作品，由作者选择要删除的作品，将数据传到works\_delete\_servlet中调用PWSS\_DAO\_factory.getProxy().doDelete (8,works)将数据库中对应作品信息删除。

4、 works\_search设计说明

程序描述：作品查找

功能：输入：作品的作者编号或者关键字；处理：从系统数据库取得作品列表然后将列表返回给用户；输出：作品列表

输入项：作品的作者编号：作者ID；关键字:50字符以内

输出项：作品列表

流程逻辑：有一个JSP页面收集用户想要查找的作品的信息[15]，将它们传输到works\_search\_servlet中，在这个servlet中根据用户输入的是作者编号还是关键字分别调用PWSS\_DAO\_factory.getProxy().findByKeyword(8,keyword)或者PWSS\_DAO\_factory.getProxy().findByAuthor(id)到数据库中查找作品信息，以list形式返回给用户。

5、 works\_exhibition设计说明

程序描述：作品展示

功能：输入：在列表中选择作品；处理：出数据库中取出作品信息和作品文件，展示给用户；输出：作品内容

输入项：在列表中选择作品

输出项：作品内容

流程逻辑：用户从列表中选择一个作品，系统将这个作品的编号发给works\_exhibition\_servlet，调用PWSS\_DAO\_factory.getProxy().findById(8,works)将作品信息和作品文件从数据库中取出，然后根据作品的类别跳转到不同的作品展示页。分为图片展示、音频展示、视频展示、文档展示

6、 works\_count\_midify设计说明

程序描述：作品评分修改

功能：输入：用户选择作品进行评分；处理：系统通过算法计算出作品的新得分，更新到数据库中；输出：成功或失败提示

输入项：在选单中选择1-100中的一个数字给作品评分。

输出项：成功或失败提示

流程逻辑：在展示页面上作品进行评分。将评分传到works\_count\_midify\_servlet中，通过算法算出作品的新得分，调用PWSS\_DAO\_factory.getProxy().doUpdate (8,works)更新数据。

7、 new\_works设计说明

程序描述：新作品审核

功能：输入：评价该作品是否合法；处理：对合法作品评定为合法，对不合法作品评定为不合法；输出：成功或失败提示

输入项：合法/不合法按钮

输出项：成功或失败提示

流程逻辑：在一个列表中选择新作品，管理员查看这个作品，然后评定这个作品是否合法。如果该作品合法就点击合法按钮，如果该作品不合法就点击不合法按钮。对于合法的系统给予100分的初始评分并调用PWSS\_DAO\_factory.getProxy().doUpdate (8,works)更新数据。对于不合法作品调用PWSS\_DAO\_factory.getProxy().doDelete (8,works)删除数据并给予作者反馈。

8、 works\_recommend设计说明

程序描述：作品推荐

功能：输入：访客上一次查找的关键字；处理：到数据库中查找和该关键字相关的作输出：作品列表

输入项：访客上一次查找的关键字

输出项：作品列表

流程逻辑：访客第一次访问系统是记录访客的IP信息以及查找过的作品的关键字。访客后续再有访问系统的时候，系统读取访客信息得到关键字。（调用PWSS\_DAO\_factory.getProxy().findById(6,visitor)）,使用该访客信息中的关键字查询该访客可能感兴趣的作品（调用PWSS\_DAO\_factory.getProxy().findByKeyword(6,keyword)），以list形式返回。

4.5数据服务

1、 backup设计说明

程序描述：数据备份

功能：输入：无；处理：将数据按照文件的形式进行备份；输出：备份文件

输入项：无

输出项：数据库数据备份文件

流程逻辑：首先检查管理员的登录状态，如果管理员未登录系统要求管理员先登录系统。管理员选择该功能后系统自动将数据库的存储文件备份到一个预设的目录之下。

2、 restore设计说明

程序描述：数据还原

功能：输入：数据库备份后的数据文件；处理：用数据库备份后的数据文件中的数据来还原数据库数据；输出：无

输入项：数据库数据备份文件

输出项：无

流程逻辑：首先检查管理员的登录状态，如果管理员未登录系统要求管理员先登录系统。管理员选择该功能后系统自动使用将数据库备份后的数据文件中的数据对数据库进行还原。

4.6 自适应推荐算法

4.6.1 推荐算法大致流程

正如上一节中所说的那样，本文中采用的方法使用了上述几种方法部分进行了有序的组合，以求达到目的，该推荐算法的大致思路如下。

对每一个元素进行一个初始化评分，使用用户相似以及用户对内容的要求对某些元素进行一个评分，使用元素评分生成一个元素排名表，将元素类别分开，使用用户的偏好提取其中部分元素生成推荐集。用户查看某些元素后对用户偏向进行反馈，用户评分后对元素评分进行反馈，然后影响用户权重，更新相似集对推荐集的影响。这样多个方法相结合，且每个方法都采用了负反馈的方式进行设计，可以在很大程度上解决问题。

详细请查看图4.1和图4.2所示。



图4.1 预处理流程图



图4.2 推荐算法流程图

4.6.2 公式介绍

1、 现有公式

(1) 平均数公式见式(4-1)

|  |  |
| --- | --- |
|  |  |

为均值，为第i个元素值，n为元素数量。

(2) 标准差公式见式(4-2)

|  |  |
| --- | --- |
|  |  |

为标准差，为第i个元素值，为均值，n为元素数量。

2、 预处理

(1) 元素评分初始化：元素评分初始化过程中，为了使评分较为公平，我们采用善意推断的方式，将元素的初始化评分定位为88.3%所以元素初始化评分的值的计算方法如公式(4-3)所示。

|  |  |
| --- | --- |
|  |  |

其中iscore为元素初始评分，为元素评分均值，为元素评分标准差。

(2) 用户评分权重初始化：用户评分权重初始化过程中，为了使评分较为公平，我们采用善意推断的方式，将元素的初始化评分定位为88.3%所以元素初始化评分的值的计算方法如公式(4-4)所示。

|  |  |
| --- | --- |
|  |  |

其中iweight为用户评分初始权重，为用户评分权重均值，为用户评分权重标准差。

(3) 用户偏好处理：用户偏好初始值设置如公式(4-5)所示

|  |  |
| --- | --- |
|  |  |

其中为偏好维数i偏好初始值，为偏好维数即元素类别数。

3、用户相似处理

(1) 用户相似度计算：假设有两个用户，分别记为甲和乙，对用户甲和用户乙查看过的所有元素进行处理，用户相似度采用公式(4-6)、公式(4-7)和公式(4-8)计算，元素属于两个用户的∪集。

|  |  |
| --- | --- |
|  |  |

us为用户相似度，为不同用户在第i个元素的相似值，n为用户甲和用户乙查看过的作品的并集的元素个数

|  |  |
| --- | --- |
|  |  |

为不同用户在第i个元素的相似值，为用户在第i个元素的评分距离。

|  |  |
| --- | --- |
|  |  |

为用户在第i个元素的评分距离，为第一个用户对于第i个元素的评分，为第二个用户对于第i个元素的评分

(2) 用户相似度影响度更新：用户操作对用户相似集在推荐中起到的作用的影响如公式（4-9）所示

|  |  |
| --- | --- |
|  |  |

其中nse为相似度影响度，ose为相似度原影响度，d为相似度原影响度影响值任意设定，si为用户在推荐集中选择的元素中属于相似集部分元素占总元素个数的比例.

4、用户搜索内容关键字影响处理

用户搜索内容关键字影响度更新:用户操作对用户搜索内容关键字在推荐中起到的作用的影响如公式（4-10）所示

|  |  |
| --- | --- |
|  |  |

usi为用户搜索内容关键字影响度，osi为用户搜索内容关键字原始影响度，e为关键字对原影响度影响值任意设定，us为用户在推荐集中选择的元素中与关键字内容相同的元素个数，all为推荐集总个数。

5、总评分

总评分计算:元素的总评分可能会高于100，从相似度影响、内容关键字影响和作品评分三个方面来考虑，具体公式如(4-11)所示

|  |  |
| --- | --- |
|  |  |

其中ascore为总评分，se为相似度影响度，us为用户相似度，score为元素评分，usi为用户搜索内容关键字影响度，scount为用户对该内容搜索次数。

6、用户偏好处理

对于用户偏好的处理使用到公式(4-12)、(4-13)和(4-14).

|  |  |
| --- | --- |
|  |  |

其中为新偏好维数i偏好值归一化处理前值，topv为本次推荐中用户选择的元素类别为i的个数占总推荐集比例，oopv为原偏好维数i偏好值，b为新操作影响系数。

|  |  |
| --- | --- |
|  |  |

其中为新偏好维数i偏好值，为新偏好维数i偏好值归一化处理前值，为归一化系数。

|  |  |
| --- | --- |
|  |  |

其中为归一化系数，为新偏好维数i偏好值归一化处理前值。

7、推荐集生成

从使用总评分排序后的元素中选择出N个元素，其中偏好维数i的元素个数为偏好维数i偏好值\*N取下限值，使用偏好值最高的维数的元素补齐剩下的元素。

8、元素评分更新

元素新评分生成：当有用户对某个元素提交评分时，系统先通过公(4-15)和(4-16)算用户本次评分的权重然后在通过公式(4-17)算出元素的新评分。

|  |  |
| --- | --- |
|  |  |

其中为用户评分对均值偏移，和为正态分布中用户评分到该元素评分均值之间的概率，uscore为用户评分，u为该元素元素评分均值。

|  |  |
| --- | --- |
|  |  |

其中tweight用户本次评分权重，为用户评分对均值偏移，oweight为用户原评分权重，a为评分靠近度影响任意设定。

|  |  |
| --- | --- |
|  |  |

其中nscore为元素新评分,tweight为用户本次评分权重,uscore为用户权重, oscore为元素原评分。

9、用户权重处理

用户权重更新如公式(4-18)和(4-19)所示

|  |  |
| --- | --- |
|  |  |

wcw为用户权重修改权重，oweight为用户原权重，tweight为用户本次评分权重。

|  |  |
| --- | --- |
|  |  |

nweight为用户新权重，wcw为用户权重修改权重，oweight为用户原权重，tweight为用户本次评分权重。

4.7 本章小结

本章中对系统中涉及到的各个功能模块中的功能点以及数据库接口的程序进行了详细设计，分别从程序描述、功能、输入项、输出项和流程逻辑这五个方面将实现这些功能点过程中会遇到的各种细节进行了说明。程序实现人员可以根据这些描述将系统代码的框架全部搭建起来，剩余的只要将各个功能点的输入输出组织成适当的页面就可以了。

第5章 系统测试

本章将介绍系统的测试。在本章中对系统的部分功能进行了测试，被测的功能点将从描述、输入和输出这个三个方面并配合该功能点的截图进行说明，以期望让读者了解系统的测试过程。

5.1 测试

5.1.1 管理员部分功能测试

1、登录测试

描述：登录界面如图5.1所示，在这个界面上登录系统，需要输入ID、密码并选择访客类型。通过不同的访客类型跳转到不同的后台处理程序中验证密码是否正确。

输入：ID、密码、访客类型。

部分输出：管理员登录后界面如图5.2所示。



图5.1 登录界面

图5.2 管理员主界面

2、 管理员信息查看测试

描述：管理员信息查看界面如图5.3所示，在这个界面上点击个人信息中的信息查看。

输入：无。

部分输出：管理员信息（ID和联系方式）。

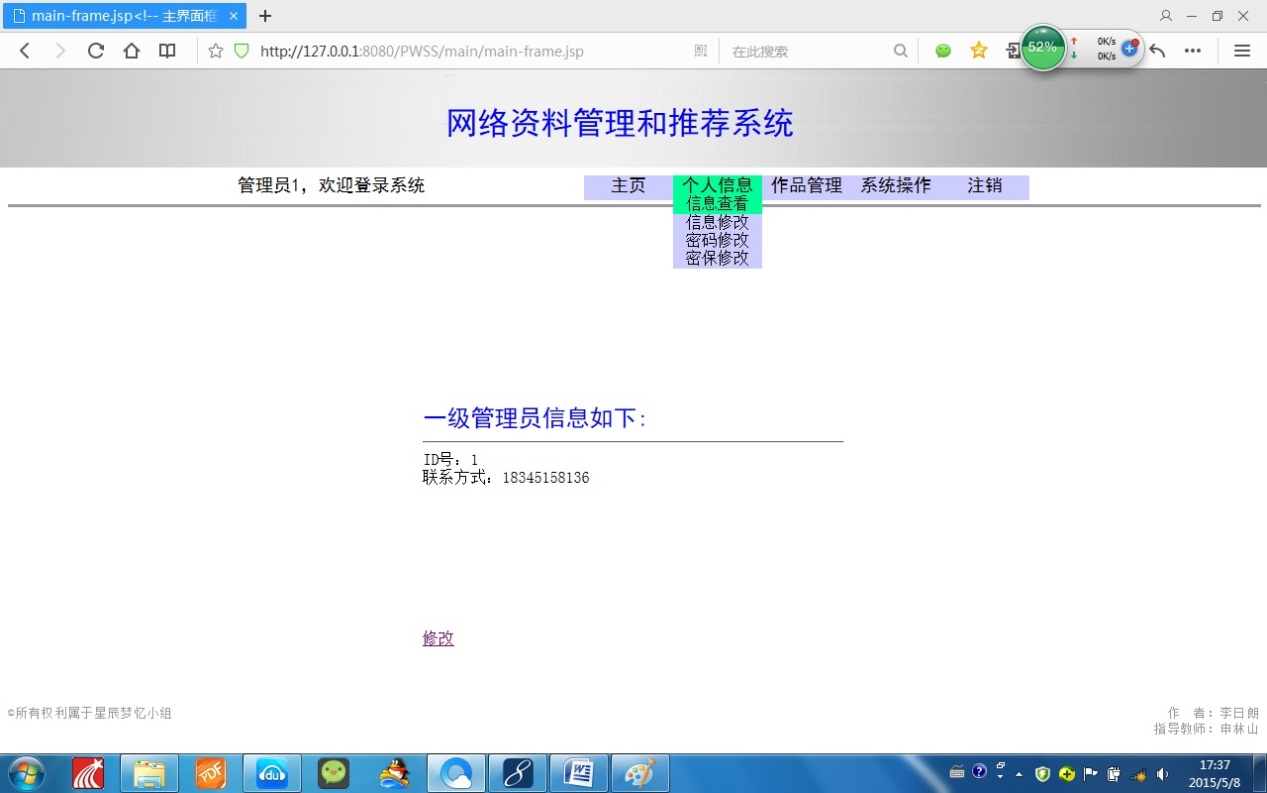


图5.3 管理员信息查看界面

3、 管理员信息修改测试

描述：管理员信息修改界面如图5.4所示，在这个界面上点击个人信息中的信息修改。

输入：联系方式。

部分输出：修改成功。



图5.4 管理员信息修改界面

4、 管理员密码修改测试

描述：管理员密码修改界面如图5.5所示，在这个界面中可以修改密码，要求输入旧密码，新密码，两个新密码要求相同。

输入：旧密码，新密码。

部分输出：修改成功或失败（旧密码正确且两次新密码相同时成功否则失败）。

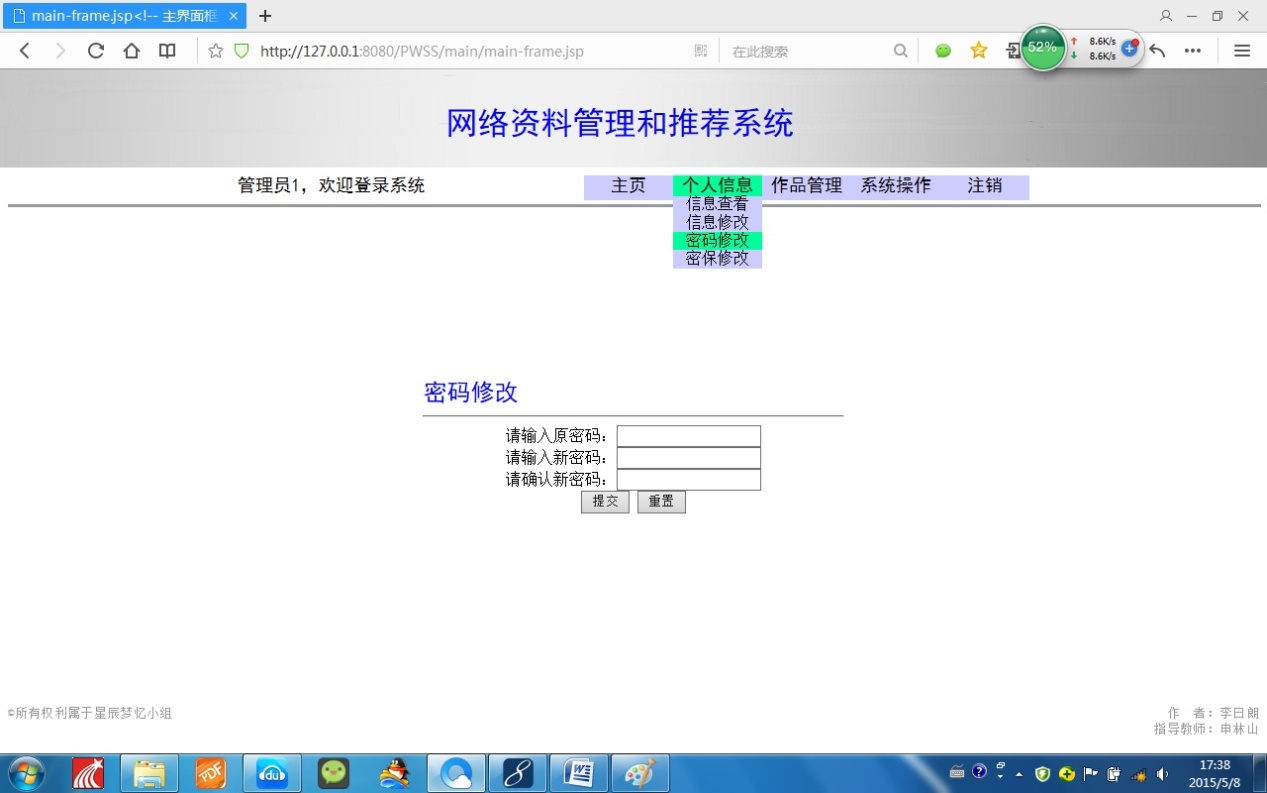


图5.5 管理员密码修改界面

5、 管理员密保修改测试

描述：管理员密保修改界面如图5.6所示，在这个界面中可以修改密保，需要先保证旧的密保答案正确才能修改密保问题和答案。

输入：旧密保答案，新密保问题，新密保答案。

部分输出：修改成功或失败（旧密保正确且新密保有效时成功否则失败）。



图5.6 管理员密保修改界面

6、 管理员新作品查找测试

描述：管理员新作品查找界面如图5.7所示，在这个界面中管理员可以任意点其中一行就会自动打开该作品。

输入：无。

部分输出：跳转到指定作品。



图5.7 管理员新作品查找界面

7、 管理员新作品测评测试

描述：管理员新作品测评界面如图5.8所示，在这个页面中管理员可以看到作品内容，并且对其进行测评。

输入：选择合法或不合法。

部分输出：合法则给予100分不合法则删除。



图5.8 管理员新作品测评界面

8、 管理员二级管理员添加测试

描述：管理员二级管理员添加界面如图5.9所示，在这个界面中可以通过输入一个手机号码来添加一个二级管理员。

输入：二级管理员手机号码。

部分输出：添加成功。



图5.9 管理员二级管理员添加界面

9、 管理员临时管理员添加测试

描述：管理员临时管理员添加界面如图5.10所示，在这个界面中可以通过输入一个手机号码来添加一个临时管理员。

输入：临时管理员手机号码。

部分输出：添加成功。



图5.10 管理员临时管理员添加

10、 管理员查找测试

描述：管理员查找界面如图5.11所示，在这个界面中可以选择管理员级别来查找管理员。

输入：管理员级别

部分输出：管理员列表。



图5.11 管理员查找界面

11、 数据备份测试

描述：数据备份界面如图5.12所示，选择系统操作中的数据备份实现系统数据的备份。

输入：无

部分输出：备份结果。

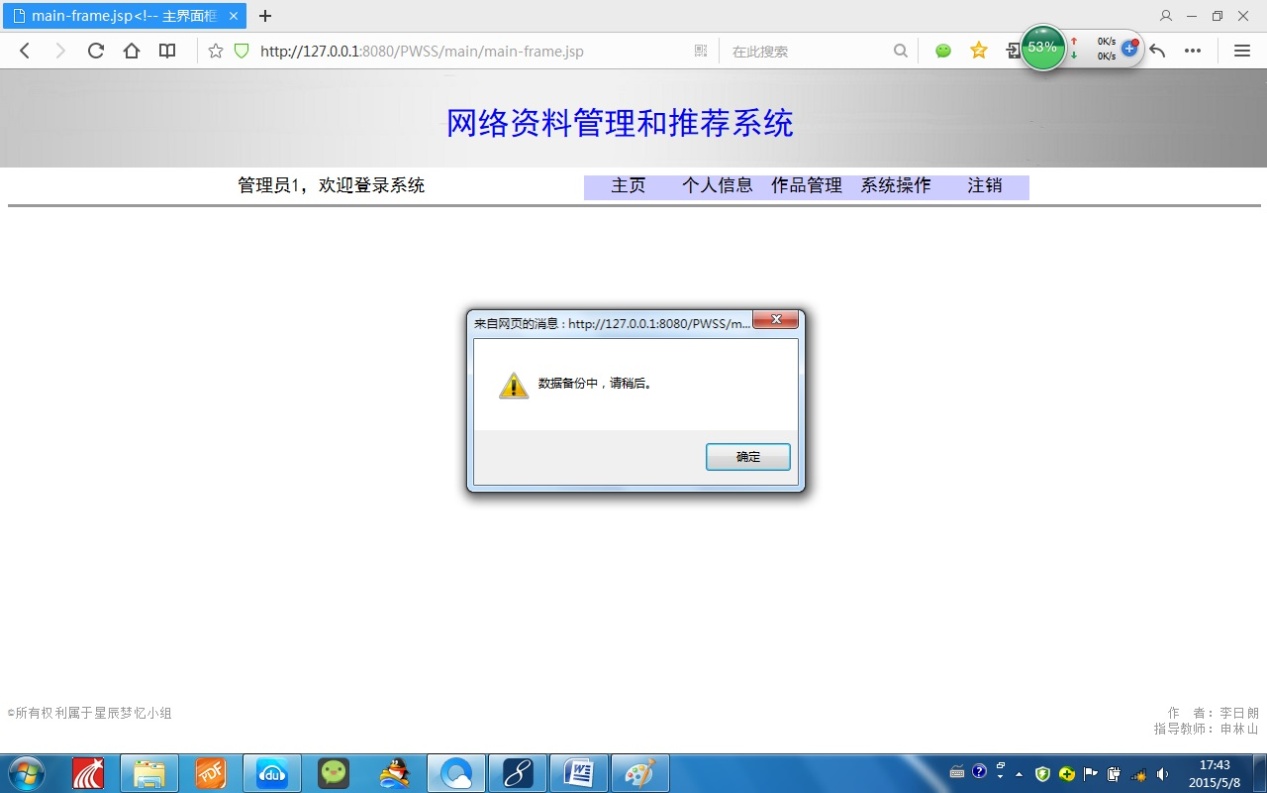


图5.12 数据备份界面

12、 数据恢复测试

描述：数据还原界面如图5.13所示，选择系统操作中的数据还原实现系统数据的还原。

输入：无

部分输出：还原结果。

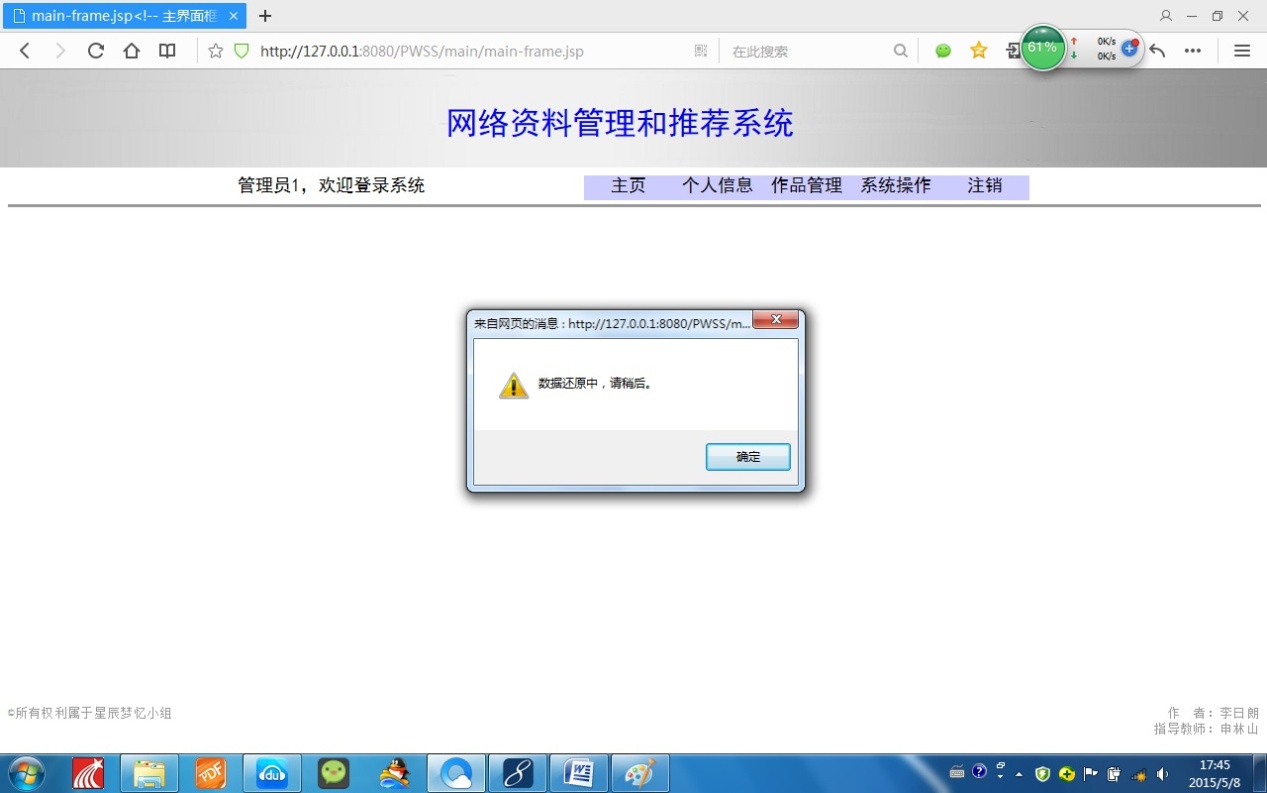


图5.13 数据恢复界面

13、 注销测试

描述：注销界面如图5.14所示。

输入：无

部分输出：用户注销



图5.14 注销界面

5.1.2 作者部分功能测试

1、作者关注的人测试

描述：作者关注的人界面如图5.15所示。这个界面中可以查看到作者关注的人通过点击其中任意一行可以打开详细信息。

输入：无

部分输出：关注的人列表



图5.15 作者关注的人界面

2、 作者信息展示测试

描述：作者信息展示界面如图5.16和图5.17所示。这个界面中可以查看到作者的信息以及作者的作品信息。



图5.16 作者信息展示界面一



图5.17 作者信息展示界面二

3、 作品上传测试

描述：作品上传界面如图5.18和图5.19所示。在这两个页面中将信息填写完整就可以实现作品上传。

输入：作品名、作品类型、文件。

部分输出：文件上传成果或失败。

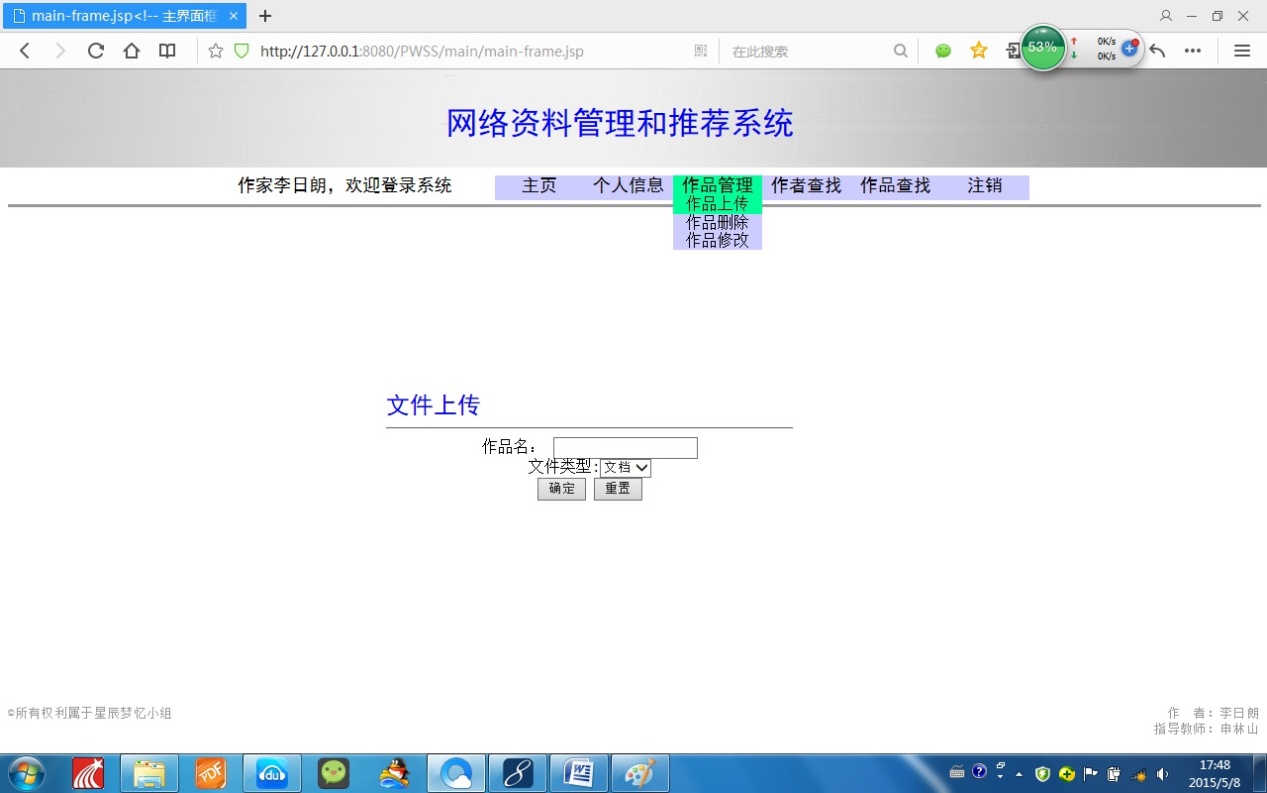


图5.18 作品上传界面一



图5.19 作品上传界面二

4、 个人作品测试

描述：个人作品界面如图5.20所示。这个页面列出了该作者的所有作品，可以通过点击任意一行来打开它，显示作品内容。

输入：选择任一作品。

部分输出：打开作品。



图5.20 个人作品界面

5、 作品展示和评分测试

描述：作品展示和评分界面如下图5.21中所示。在这个页面中可以对作品内容进行查看也可以对作品进行评分。

输入：评分。

部分输出：修改作品分数。



图5.21 作品展示和评分界面

5.1.3 游客部分功能测试

1、 访客作品查找测试

描述：访客作品查找界面如下图5.22中所示。在这个页面中访客可以对作品查找。

输入：关键字或作者ID

部分输出：作品列表。



图5.22 访客作品查找界面

2、 访客作品推荐测试

描述：访客作品推荐界面如下图5.23中所示。系统会根据访客上一次访问系统时采取的动作对访客进行作品推荐。

输入：无

部分输出：作品列表和访问者IP。



图5.23 访客作品推荐界面

5.2本章小结

在本章中展示了本论文中设计的系统实现以后的部分功能的测试，从功能的描述、功能的输入、功能的输出等方面对系统测试进行描述。本章的测试包括登录测试、管理员信息查看测试、管理员信息修改测试、管理员密码修改测试、管理员密保修改测试、管理员新作品查找测试、管理员新作品测评测试、管理员二级管理员添加测试、管理员临时管理员添加测试、管理员查找测试、数据备份测试、数据恢复测试、注销测试、作者关注的人测试、作者信息展示测试、作品上传测试、个人作品测试、作品展示和评分测试、访客作品查找测试、访客作品推荐测试等20项功能点的测试。本次测试没有包含网络资料管理和推荐系统的全部功能，测试只截取了系统中部分代表性的功能点的测试结果。

结论

网络上数据太过巨大，无数优秀作品被海量信息埋没，作者们缺乏一个统一的平台给他们展示自己的才华。本文针对这个现状，在参考国内外相关系统和现有技术的基础下，采用 BS架构、DAO数据存取模式和MVC设计模式，设计并实现了一个用于作者展示作品的平台，与此同时网络资料管理和推荐系统为管理员、作者和游客用户提供了各种各样的服务让他们能够更好的管理自己的信息，发现所需要的信息的网络资料管理和推荐系统。

通过本文中的设计和实现，取得了以下几方面成果：

(1)在数据管理模块中实现了数据备份和数据恢复功能。

(2)在管理员管理模块中实现了二级管理员添加、二级管理员删除、临时管理员添加、临时管理员权限收回、管理员登录、管理员注销、管理员信息修改、管理员密码修改、管理员信息查询、添加密保、修改密保、找回密码、新作品提醒、不合法作品删除等功能。对于数据输入部分采取了表单验证与控制。

(3)在用户管理模块中实现了用户注册、用户登录、用户信息查询、用户信息修改、密码修改、添加密保、修改密保、密码找回、注销登录、添加关注等功能。对于数据输入部分采取了表单验证与控制。

(4)在用户作品管理模块中实现了作品上传、作品删除、作品修改、作品展示、作品查询等功能。对于数据输入部分采取了表单验证与控制。

(5)在游客信息管理模块中实现了游客信息采集和作品推荐两个功能。

(6)提出了一个基于负反馈的自适应推荐算法。

由于学生学位论文完成时间的限制以及开发网站类软件经验不足，本系统存在着一些问题。当系统测试以后，可以看出本系统的设计和实现具有可行性。但是存在着一些不足，例如当系统运行一段时间以后，系统中的数据量将增大到一定程度，此时数据备份和数据恢复功能需要较长时间才能完成，速度大约是600M/分钟，当用户上传大文件时（100M以上）会出现响应偏慢的情况，诸如此类的问题。在未来的时间里将会对系统的设计和实现进一步完善，使得系统拥有更加完善的功能和更加优秀的性能。

参考文献

[1] 周凤林.电子商务下的柔性推荐系统[D].武汉:华中科技大学,2007.

[2] 赵基宏.基于MYECLIPSE开发工具的移动营业厅收费系统的设计与实现[D].四川:电子科技大学,2012。

[3] K01rkk01inen M, Ala-Risku T, Fr01mling K. The Product Centric Approach: A Solution to Supply Network Information Management Problems[J]. Computers in Industry, 2003, 52(3):147–159.

[4] Walter F E, Battiston S, Schweitzer F. A model of a trust-based recommendation system on a social network[J]. Autonomous Agents and Multi-Agent Systems, 2008, 16(1):57-74.

[5] 张海藩.软件工程导论（第三版）[M].北京:清华大学出版社,2003:35-37.

[6] 李刚.轻量级Java EE 企业应用实战（第4版）--Struts 2+Spring 4+Hibernate整合开发[M].北京:电子工业出版社,2014:7-7.

[7] 刘斌.精通Java Web 整合开发:JSP+AJAX+Struts+Hibernate(第2版)[M].北京:电子工业出版社,2011:12-12.

[8] 陈倩.BS架构智能巡检系统管理平台模块的设计与实现[D].北京:北京邮电大学,2014.

[9] 田娟,徐钊.基于J2EE的MVC设计模式的分析与思考[J].计算机与现代化,2010,10:54-58.

[10] (美)Roger S. Pressman著,郑人杰,马素霞等译.软件工程:实践者的研究方法[M].北京:机械工业出版社,2014:29-30.

[11] 王珊,萨师煊.数据库系统概论[M].北京:高等教育出版社,2006:49-49.

[12] 陈红.数据库系统原理教程[M].北京:清华大学出版社,2000:62-62.

[13] (美)霍尔(Hall,M.),(美)布朗(Brown,L.),(美)蔡金(Chaikin,Y)著,胡书敏等译.Servlet与JSP核心编程（第2卷 第2版）[M].北京:清华大学出版社,2009:63-65.

[14] 梁建武.JSP程序设计实用教程[M].北京:中国水利水电出版社,2013:15-16.

[15] Mart Hall,Larry Brown,And Yaakov Chaikin.Core Servlet and JavaServer Pages Volume 2:Advanced Technologies[M].Person Education,Inc,2008.27-28.

# 攻读学士学位期间发表的论文和取得的科研成果

致谢

非常感谢申林山讲师，申林山讲师对笔者的论文编写工作给予了详细的指导。从一开始选定论文题目，到论文相关收集资料，再到论文的编写和数次的修改，申林山讲师对笔者进行了无私帮助和细致辅导，让笔者每当想起总热泪盈眶。为了给笔者的论文编写工作进行指导，申林山讲师放弃了他的个人时间。申老师这种无私奉献的精神让笔者倾佩，在此笔者向申老师表达最真挚的谢意。

同时笔者对所有任课老师以及在日常学习中帮助过笔者的同学表示感谢。是他们陪笔者走过了这四年的学士学位的学习生涯，在这1400多个日夜中，他们给了笔者鼓励让笔者拥有了走下去的勇气。在此笔者向他们表达真挚的谢意，希望所有给予笔者指导的老师培养出更多优秀的人才，希望同学们理想成真。