**北 京 林 业 大 学**

**19 学年— 20 学年第1 学期 Java高级技术 实习报告书**

专 业：计算机科学与技术 班 级：

姓 名： 学 号：

实习地点： 辅导教师：

实习题目： 　　研究资源共享平台

实习环境： 　 JDK11+SSM框架+MySQL5.1+tomcat9.0

实习内容：

# （一）系统需求分析与概要设计

## 1. 系统用户角色和功能分析设计

该系统有两类用户，一类是管理员用户，另一类是普通用户，管理员账号由系统指定，普通用户可以通过注册获得账号

* 1. **管理员用户功能**

管理员用户的功能主要是对注册用户，用户上传的资源和资源领域进行统计，该角色的用例图如图1.1.1所示。

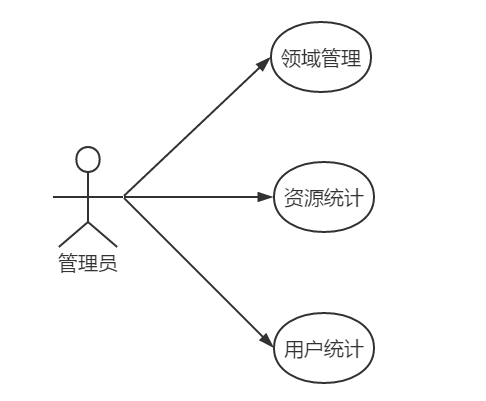


图1.1.1 管理员用户用例图

* 1. **普通用户功能**

注册用户的功能主要是搜索资源，上传资源，查看自己上传的资源以及删除资源。该角色的用例图如图1.1.2所示。

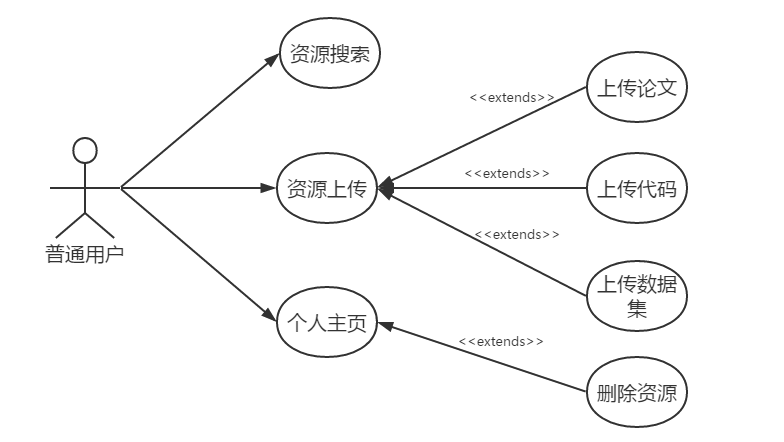


图1.1.2 普通用户功能

## 数据库设计

* 1. **概念结构设计**

图1.2.1展示了整个软件系统的E-R模型设计，图中反应了各个实体的属性和实体之间的联系。用户可以上传论文、代码或数据集，每个论文、代码或数据集都有对应领域，都可以被不同的用户多次评论。

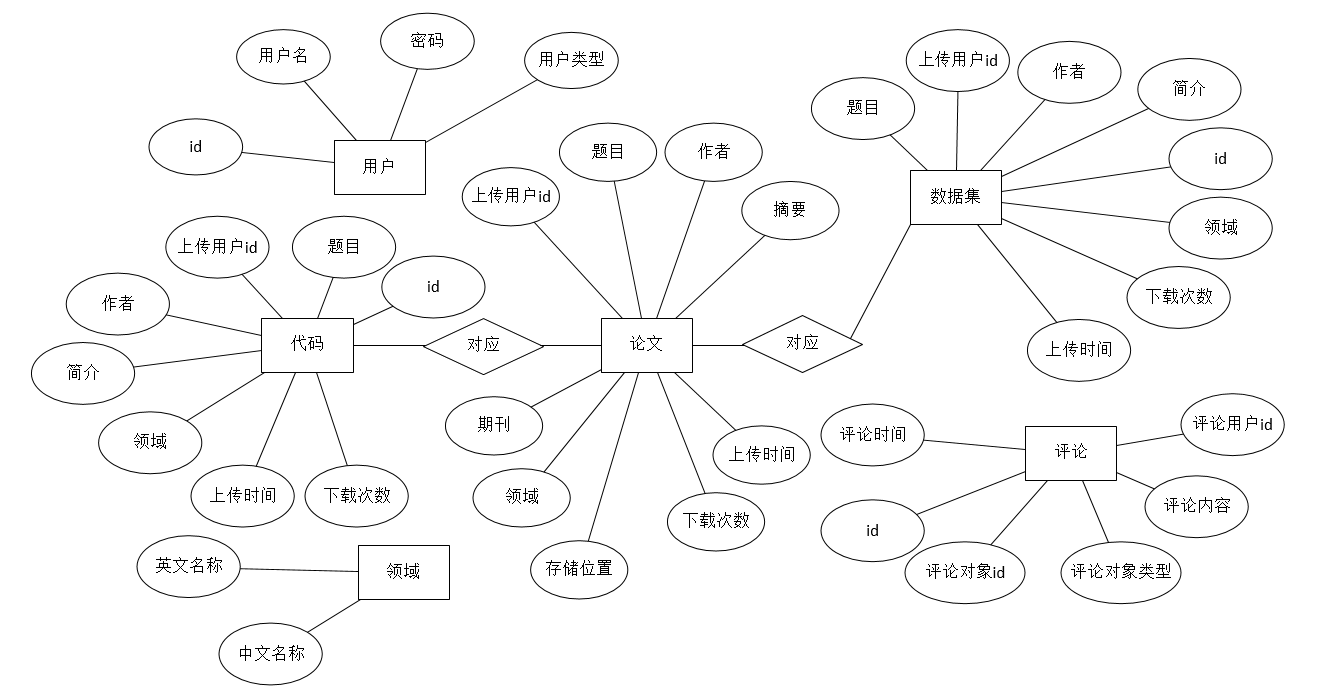


图1.2.1 E-R图

* 1. **数据表结构及作用**

以下介绍用户表，论文表，代码表，数据集表，评论表、领域表和资源对应关系表。

* + 1. **用户表user**

表2.2.1 用户表user

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 含义 |
| id | INT | 主键 | 该表的自增主键 |
| name | VARCHAR(30) | 无 | 用户名 |
| pwd | VARCHAR(30) | 无 | 密码 |
| Clazz | INT | 无 | 用户类型 |

* + 1. **论文表paper**

表2.2.2 论文表paper

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 含义 |
| id | INT | 主键 | 该数据表的自增主键 |
| userid | INT | 外键 | 上传该论文的用户id |
| titel | VARCHAR(80) | 无 | 论文题目 |
| author | VARCHAR(30) | 无 | 作者 |
| abstarct | TEXT | 无 | 摘要 |
| journal | VARCHAR(50) | 无 | 论文所属期刊 |
| field | VARCHAR(50) | 无 | 所属领域名称 |
| url | VARCHAR(256) | 无 | 存储在服务器上的位置 |
| downTimes | INT | 无 | 被下载次数 |
| uploadTime | DATE | 无 | 上传时间 |

* + 1. **代码表code**

表2.2.3 代码表code

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 含义 |
| id | INT | 主键 | 该数据表的自增主键 |
| userid | INT | 外键 | 上传该代码的用户id |
| name | VARCHAR(80) | 无 | 代码名称 |
| author | VARCHAR(30) | 无 | 作者 |
| intro | TEXT | 无 | 简介 |
| field | VARCHAR(50) | 无 | 所属领域名称 |
| url | VARCHAR(256) | 无 | 存储在服务器上的位置 |
| downTimes | INT | 无 | 被下载次数 |
| uploadTime | DATE | 无 | 上传时间 |

* + 1. **数据集表dataset**

表2.2.4 数据集表dataset

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 含义 |
| id | INT | 主键 | 该数据表的自增主键 |
| userid | INT | 外键 | 上传该数据集的用户id |
| name | VARCHAR(80) | 无 | 数据集名称 |
| author | VARCHAR(30) | 无 | 作者 |
| intro | TEXT | 无 | 简介 |
| field | VARCHAR(50) | 无 | 所属领域名称 |
| url | VARCHAR(256) | 无 | 存储在服务器上的位置 |
| downTimes | INT | 无 | 被下载次数 |
| uploadTime | DATE | 无 | 上传时间 |

* + 1. **评论表comment**

表2.2.5 评论表comment

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 含义 |
| id | INT | 主键 | 该数据表的自增主键 |
| userid | INT | 外键 | 发出该评论的用户id |
| clazz | INT | 无 | 评论对象的类型 |
| otherid | INT | 无 | 评论对象的id |
| comment | TEXT | 无 | 评论内容 |
| com\_time | DATA | 无 | 评论时间 |

* + 1. **领域表field**

表2.2.6 领域表field

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 约束 | 含义 |
| fieldCN | VARCHAR(50) | 主键 | 领域的中文名称 |
| fieldEN | VARCAHR(100) | 无 | 邻域的英文名称 |

* + 1. **资源对应关系表papercode和paperdata**

对应关系表是指论文和代码的对应关系papercode、论文和数据集的对应关系paperdata，这两表是联系表，只有自增主键、论文id和代码id(或数据集id)三个字段。

## 系统架构设计

该系统使用SpringMVC框架提供的MVC架构，在这个系统中，该架构分为四层，自底向上分别是模型层，服务层，控制层和视图层，如图3.1所示。

模型层使用了MyBatis的generator生成，得到途中模型层的dao层，mapper层和po类，之后根据需要补充代码和pojo类

服务层的作用是实现控制层所需要的功能，如数据访问和计算等功能，是模型层对控制层透明，为控制层提供简化的接口，从而减少控制层的代码，使整个系统的每一层只与和它相邻的层有关，符合迪米特法则。

控制层使用SpringMVC提供的Request mapping注解，每一个函数分别对应一个url地址和jsp页面。

视图层全部为jsp页面，使用了layui前端框架。

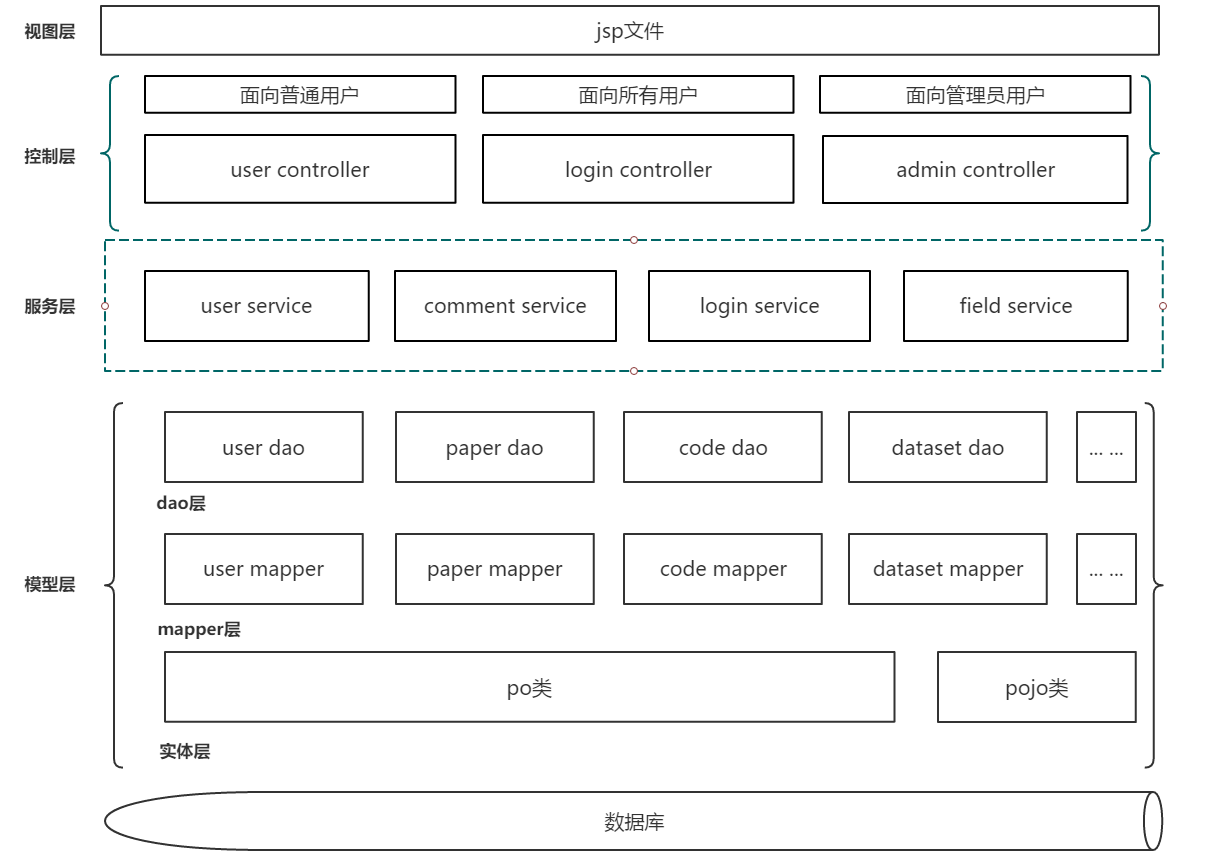


图3.1 系统架构设计图

## 系统功能模块设计

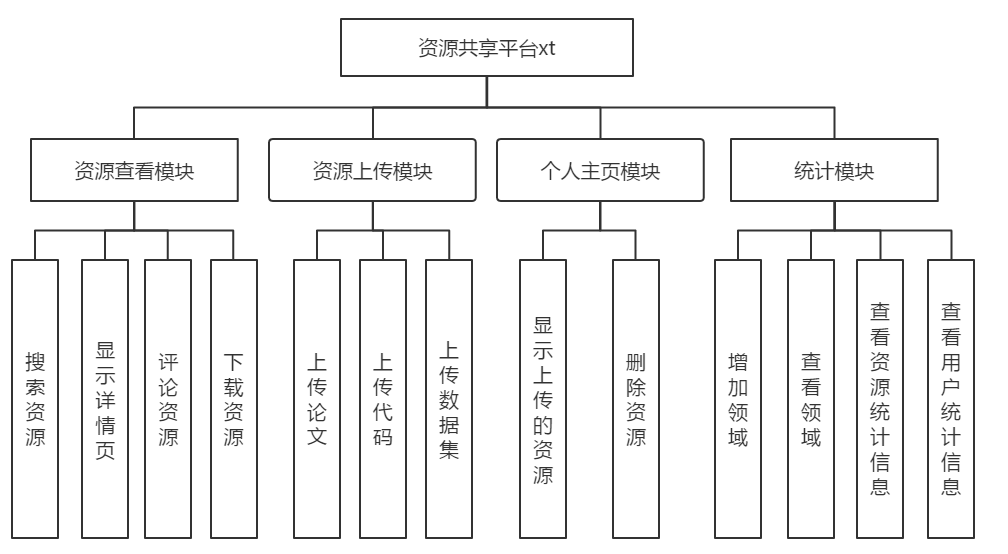


图4 系统功能模块设计图

# （二）详细设计与关键技术

## 1. 关键模块-资源搜索模块

普通用户在登录后会直接跳转到资源搜索功能，该功能呈现了资源搜索的筛选条件和所有资源，用户可以通过输入筛选条件的方式减少列表项，从而方便查询。资源搜索模块将给用户呈现一个符合筛选条件的资源列表。该模块的顺序图如图2.1所示。

为例图示简介，以下的图片将控制层，服务层和模型层统称为后台程序。

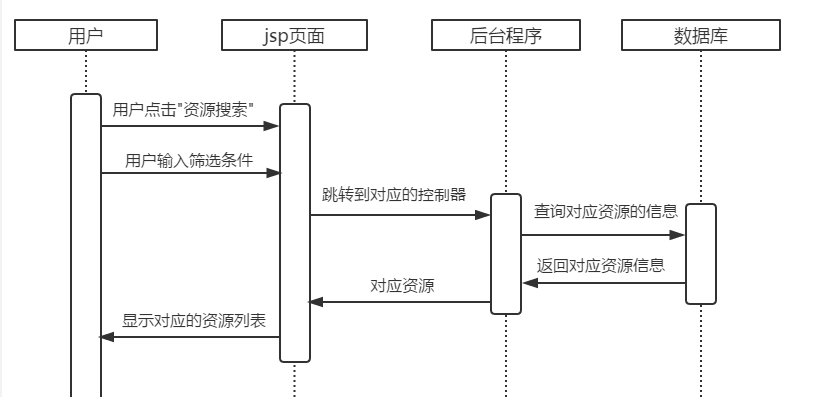


图2.1 资源搜索模块顺序图

## 关键模块-资源详情页模块

得到资源搜索结果列表后，用户可以点击对应的资源跳转到资源详情页。资源详情页显示资源的基本信息：题目，作者，摘要，上传时间等，显示该资源对应的资源，显示该资源的所有评论，提供下载按钮和发布评论表单。该模块的顺序图如图2.2所示

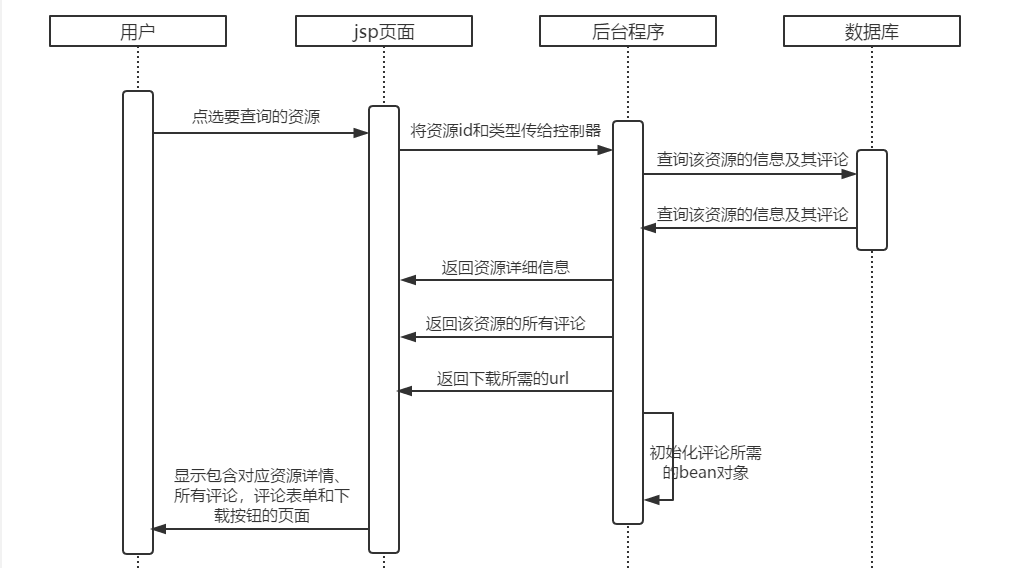


图2.2 资源详情页模块顺序图

## 关键模块-资源上传模块

用户可以上传资源，包括论文，代码，数据集，其中上传代码和数据集时可以指定当前用户上传的论文与之对应。图2.3是资源上传模块的顺序图（以上传代码为例）。

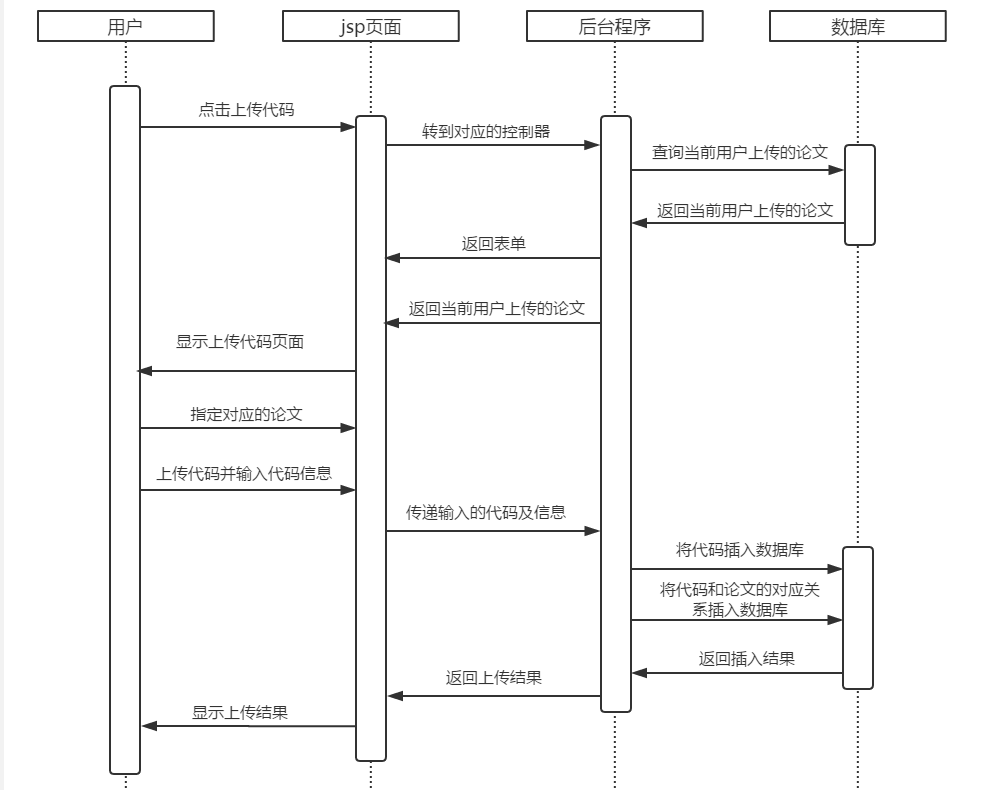


图2.3 资源上传模块的顺序图（以上传代码为例）

# （三）系统部署与发布

## 1. 系统部署环境配置

本系统使用的JDK版本为JDK11。使用了Spring，MyBatis，SpringMVC框架，数据库使用了MySQL5.1，服务器使用了Tomcat 9.0.24.

## 系统部署说明

系统使用的数据库名称为codeshare，数据库的连接属性可以在src/applicationContext.xml中的20行至34行配置。项目的启动页为http://localhost:2008/计算机171\_171002614\_董泽宇/user/login(已经在WebContent/WEB-INF中配置好了)。项目中用到了上传和下载功能，上传下载产生的论文文件存储在服务器的D:/myupload/，代码文件存储在服务器的D:/codeRepo/，数据集文件存储在服务器中的D:/dataRepo/。

## 3. 系统部署结果展示

登录页面：



图3.3.1 登录界面

普通用户界面：

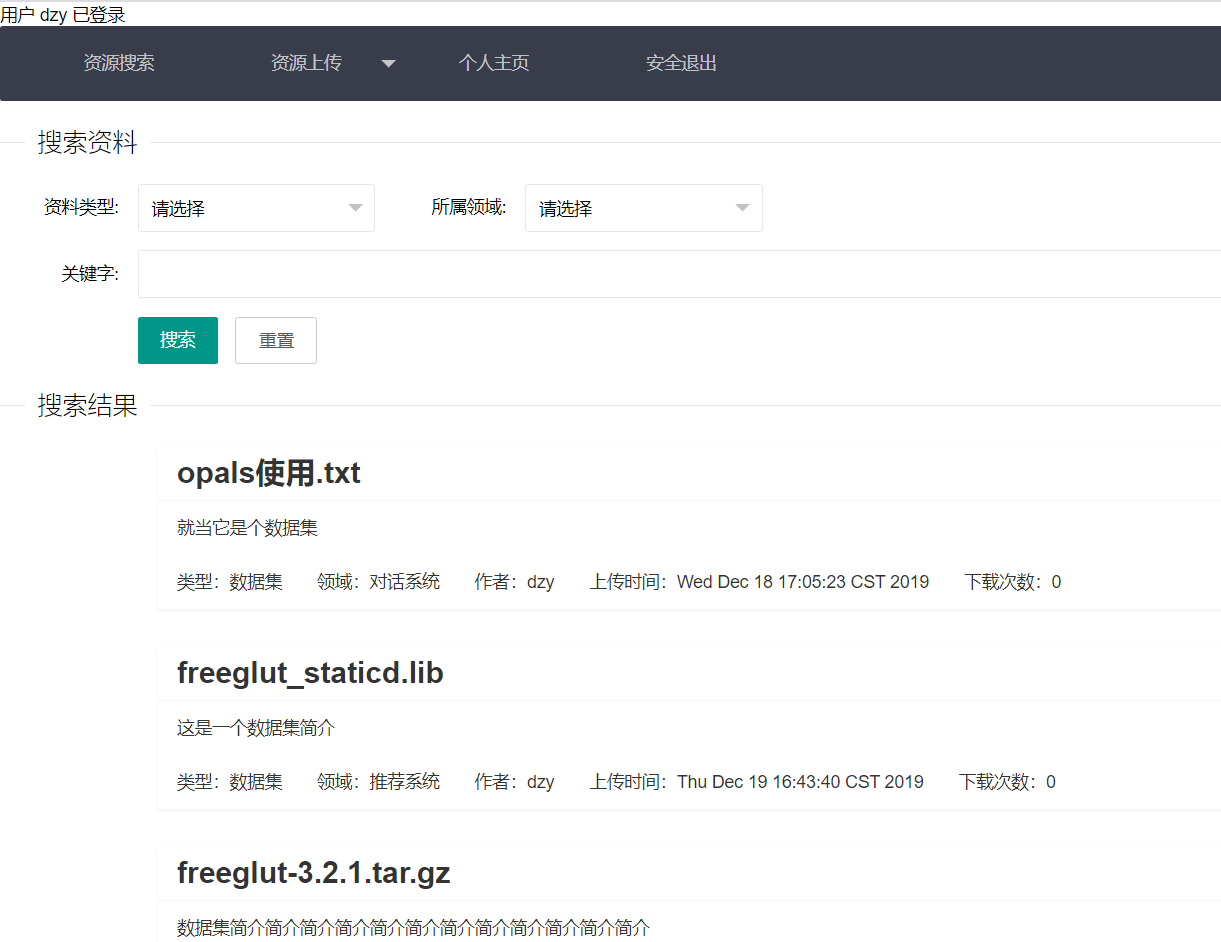


图3.3.2 普通用户界面

管理员界面：

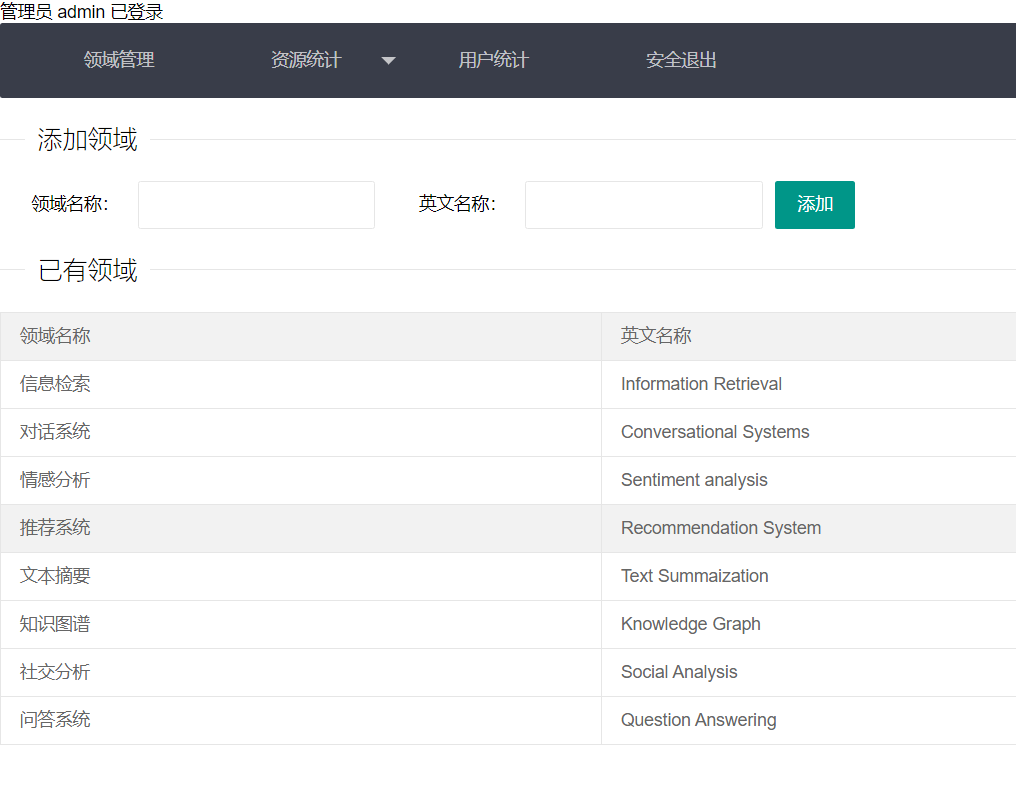


图3.3.3 管理员界面

# 结论分析

1. **问题与解决方法**

遇到的问题大多是有关框架的，比如不熟悉SpringMVC的使用，不会制作复杂的页面。解决方法只有查找资料和尝试。通过一段时间的学习，我自认为把SpringMVC思想贯彻到了这个系统中，页面页做得较为丰富，能够将导航栏，表单，按钮，列表，面板等呈现在同一个页面中，并且跳转到不同的页面。

1. **收获和体会**

收获：对MVC框架有了更深的理解，对前端页面有了进一步的了解，提高了学习能力，增强了独立解决代码问题的能力。

体会：MVC框架通过分层分块的方式降低耦合，通过注解机制和代理机制减少代码量，其中的设计思路值得进一步的学习。感受到了框架带来的便利，接触到了更多更全面的后端知识，同时也感受到前端页面的繁杂，了解到自己知识上的欠缺。