

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： 17-1

姓 名： 王植琴 学 号： 171002506

实习地点： 信息中心 N08 机房 辅导教师： 田萱

实习题目： 研究资源共享平台

实习环境： Windows10 IDEA NavicatForMySQL

实习内容：

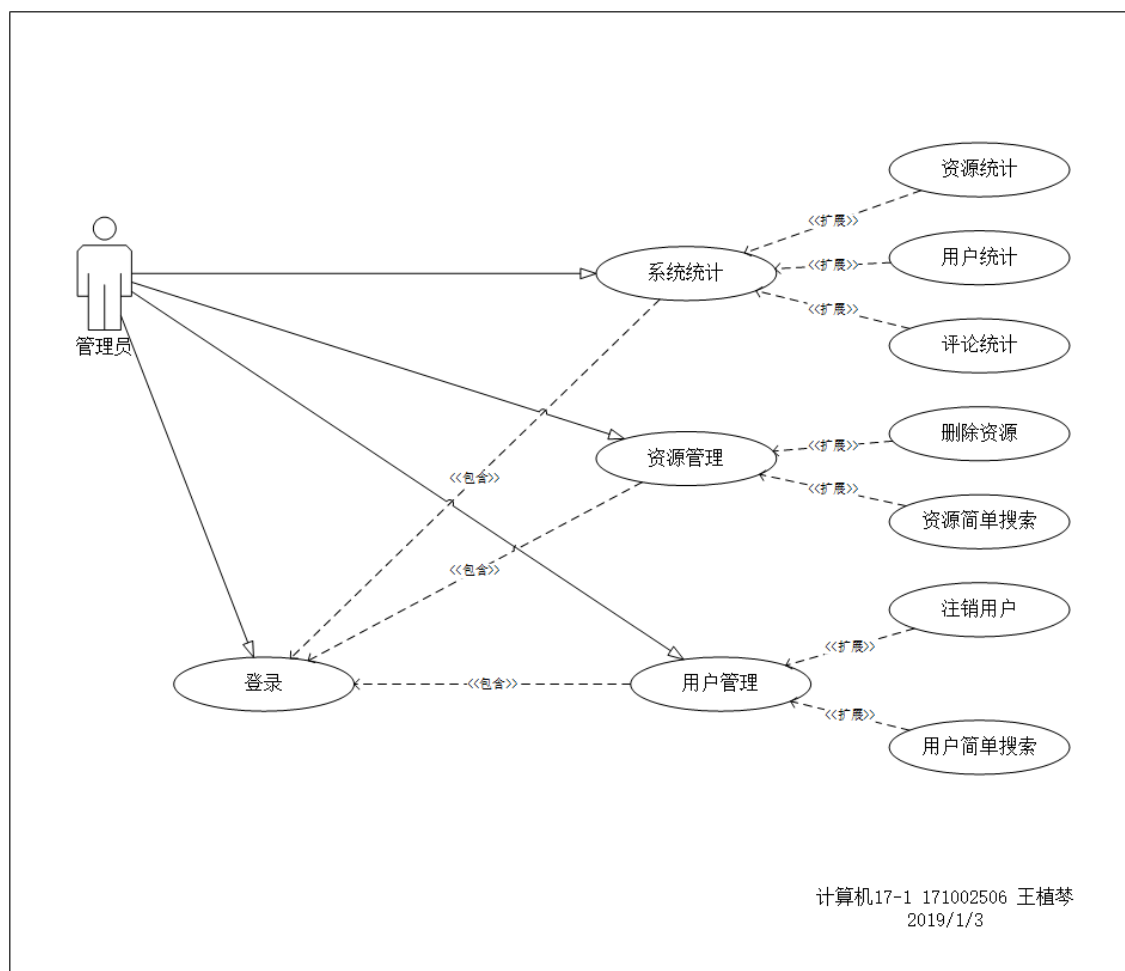
（一）系统需求分析与概要设计

1. 系统用户角色和功能分析设计

1) 该系统具体用户分为两类：

- （1） 系统管理员（admin）（侧重于对用户和平台所有资源的管理）
- （2） 普通用户（generaluser）（侧重于使用平台和对个人资源的管理）

2) 对系统管理员，主要负责管理平台全部资源和维护用户信息，其行为包括：



① 登录成功页面。

登录成功进入管理员主页面，页面上方显示出登陆用户名并一直延续到后面相关页面。

- 左侧导航栏展示资源管理（包括资源删除功能和资源统计功能）和用户信息管理等两个选项，可分别进入对应的管理页面。
- 右侧主页面按照模块分别显示资源统计情况，统计图，版本信息，用户信息，平台简介。

若登陆失败，则提示管理员登录失败等错误信息，返回登录页面重新登陆。

② 删除资源管理页面：

以 LayUI 动态表格形式展示所有资源基本信息，包括(资源 ID，资源类型，资源名称，资源作者，资源领域，资源来源，资源简介，上传用户，上传时间)等，右侧设置详情按钮和删除按钮。

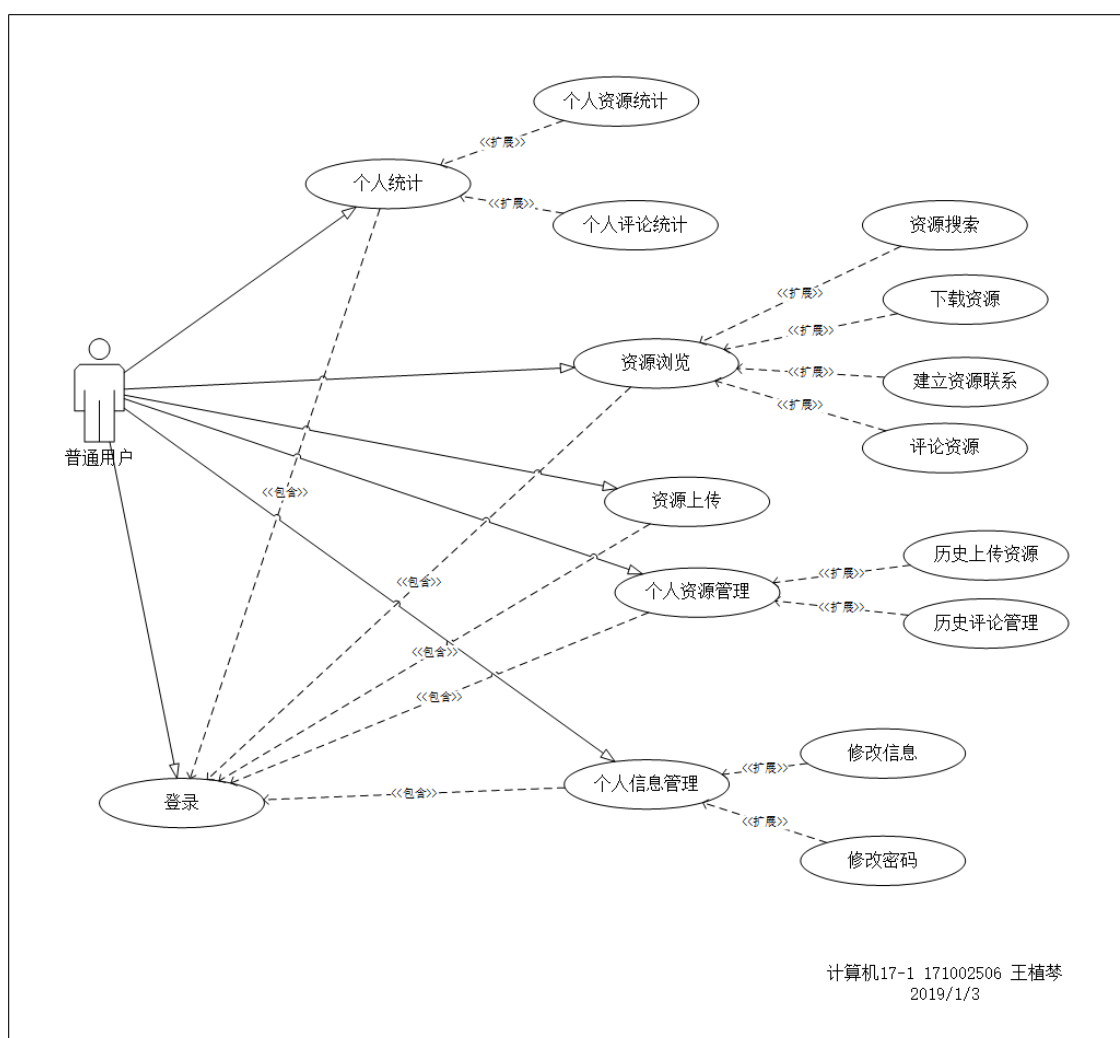
- a) 点击详情按钮可进入该资源详情页面，由于是管理员身份，资源系详情页面只显示资源详情信息、资源关联情况和资源评论情况。
- b) 点击删除按钮，弹出确认删除对话框，需要再次确认之后，再进行资源的删除。

③ 用户管理页面：

以 LayUI 动态表格形式展示所有用户基本信息，包括(用户名，用户密码（经过 SHA256 加密），用户类型，姓名，联系方式)，右侧设置删除用户按钮。

- a) 点击删除用户按钮，弹出确认删除对话框，需要再次确认之后，再进行用户的删除

3) 对普通平台用户，主要行为包括：



① 登录成功页面

登录成功进入普通用户主页面，页面上方显示出登陆用户名并一直延续到后面相关页面。

- a) 左侧导航栏展示资源中心（包括浏览资源：其中有全部资源浏览和分类浏览功能入口，上传资源入口）、个人中心（包括历史上传页入口和历史评论入口）和用户设置（包括基本资料和修改密码功能入口）。
- b) 右侧主页面按照模块分别显示功能快捷方式、个人资源统计概览和统计图、版本信息、用户信息以及平台简介。

若登陆失败，则提示用户登录失败错误信息，返回登录页面重新登陆。

② 浏览资源页面（分为全部资源页面，论文资源页面，代码资源页面和数据及资源页面）

所有资源页面均以 LayUI 动态表格形式展示所有资源基本信息，包括(资源 ID，资源类型，资源名称，资源作者，资源领域，资源来源，资源简介，上传用户，上传时间)等，右侧设置详情按钮。

点击详情按钮进入资源详情页面，页面包括资源详情信息、资源关联建立与查看区域、资源评论与查看评论区域。

上方设置多条件搜索区，可以进行多条件（包括上传用户，资源名，资源种类和资源领域*动态生成）模糊搜索。

③ 资源详情页面

资源详情页面的目的是展示资源的详细信息，提供资源的下载功能，进行资源关联的查看和添加，进行资源评论的查看和添加。

a) 资源关联添加

根据资源的类型显示添加资源的类型，如果该资源为论文资源，则在添加资源关联模块只显示登录用户上传过的非论文资源，反之，如果论文资源为非论文（代码资源或数据集资源），则在添加资源关联模块只显示该用户上传过的论文资源，即只允许用户将非论文资源与论文资源进行关联

b) 资源评论

资源评论区设置评论提交和评论显示两部分，用户可以提交对该资源的评论，也可以在下方浏览其他用户和自己对于该资源的评论。

④ 上传资源页面

上传资源界面基于 LayUI-Form 设计，提示用户填写资源名称，资源类型，资源作者，资源来源，资源领域（请求后台数据动态生成下拉框）和资源摘要，下方为上传资源按钮，其中有些内容通过 filter 设置了必填项（在未填写情况下弹出框提醒用户必填项不能为空），最下方是立即提交按钮和重置按钮。

⑤ 历史上传页面

历史上传页面以 LayUI 动态表格形式展示有关该用户上传的资源基本信息，包括(资源 ID，资源类型，资源名称，资源作者，资源领域，资源来源，资源简介，上传用户，上传时间)等，右侧设置详情按钮。

上方设置多条件搜索区，可以进行多条件（包括资源名，资源种类）模糊搜索。

⑥ 历史评论页面

历史评论页面以 LayUI 动态表格形式展示有关该用户对资源的历史评论信息，包括(评论 id、评论资源名、资源类型、评论时间和评论类型)，右侧设置详情按钮。

点击详情按钮跳转到评论资源的详情页面。

上方设置多条件搜索区，可以进行多条件（包括评论资源名，评论资源种类）模糊搜索

⑦ 基本资料页面

基本资料页面显示登陆用户的用户身份，用户 ID，姓名，性别和联系方式，只允许修改姓名，性别和联系方式，其余均为不可更改项。

⑧ 修改密码页面

修改密码页面要求用户填写旧密码以确认身份，旧密码正确的前提下验证新密码两次输入是否相同，如果相同则修改用户密码。

2. 数据库设计

该系统的具体数据库设计如下（下划线为主码，波浪线为外码）：

- 1) 用户表 User (用户 ID，用户密码，用户类型，用户姓名，用户性别，用户联系方式)
- 2) 资源表 resource(资源 ID，资源类型)

- 3) 论文资源表 essay(论文 ID, 论文名, 论文作者, 论文来源, 论文简介, 论文物理存储位置, 论文上传时间, 论文上传用户 ID, 领域 ID)
- 4) 代码资源表 code(代码 ID, 代码名, 代码作者, 代码来源, 代码简介, 代码物理存储位置, 代码上传时间, 代码上传用户 ID, 领域 ID)
- 5) 数据集资源表 code(数据集 ID, 数据集名, 数据集作者, 数据集来源, 数据集简介, 数据集物理存储位置, 数据集上传时间, 数据集上传用户 ID, 领域 ID)
- 6) 论文数据集关联表 essay2dataset (论文 ID, 数据集 ID, 资源关联建立用户, 资源关联建立时间)
- 7) 论文代码关联表 essay2code (论文 ID, 代码 ID, 资源关联建立用户, 资源关联建立时间)
- 8) 领域表 area (领域 ID, 领域名)
- 9) 评论表 comment (评论 ID, 评论用户 ID, 评论资源 ID, 评论时间, 评论内容)

该系统提供的数据库结构 E-R 模型如下图所示:

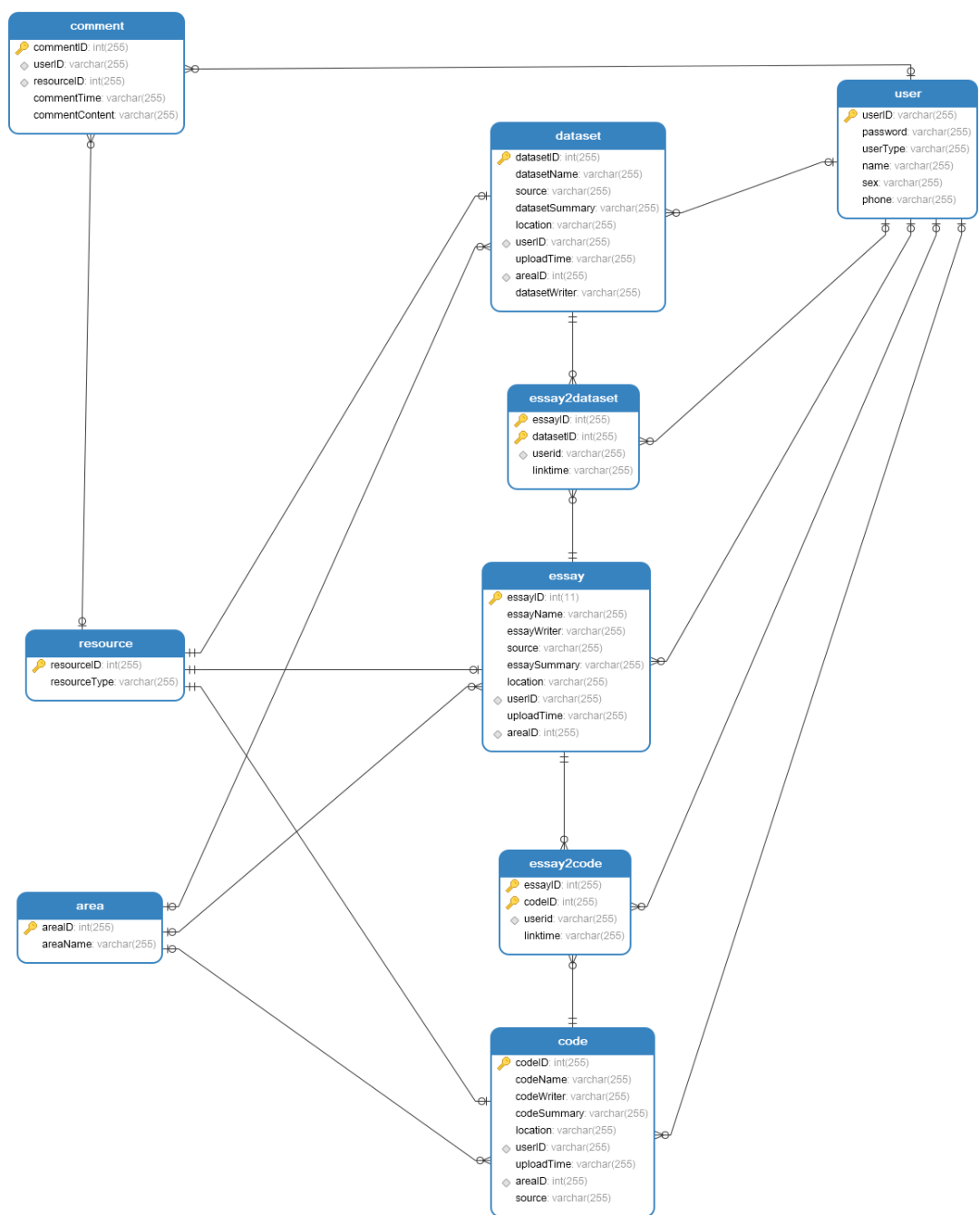


图 1：研究生课程论文选读系统的 E-R 模型

数据字典定义如下

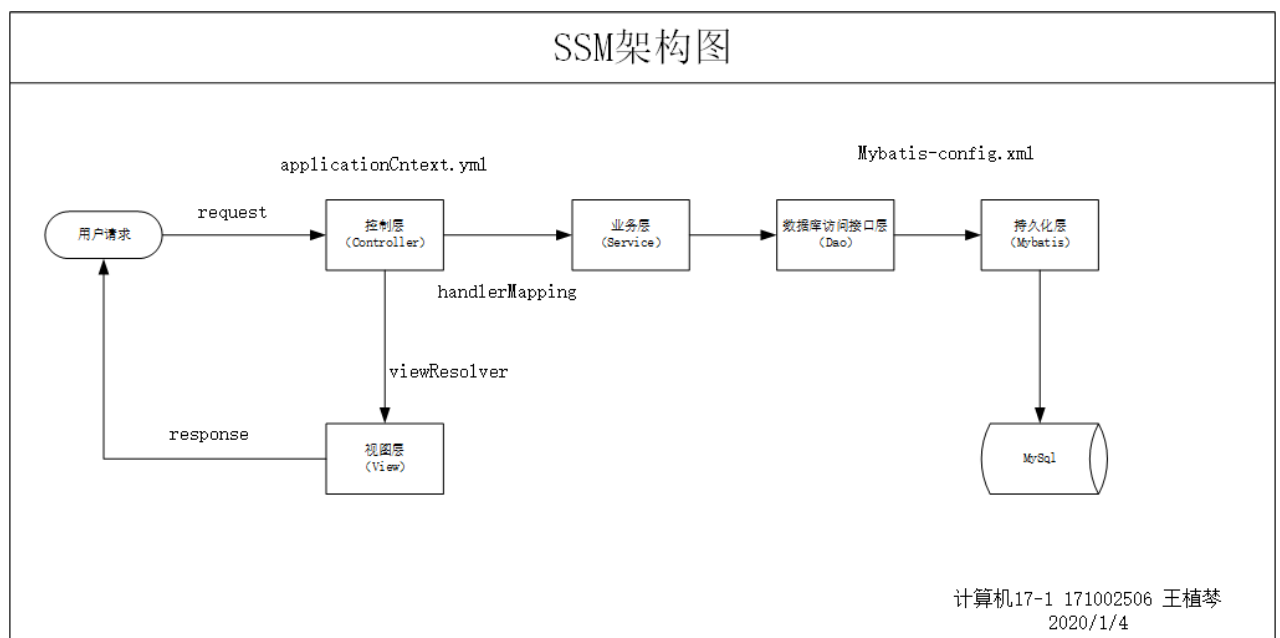
表名	字段名	数据类型	允许为空
Area 表			
area	arealD	int(255)	NO
area	areaName	varchar(255)	YES
code 表			
code	codeID	int(255)	NO
code	codeName	varchar(255)	YES

code	codeWriter	varchar(255)	YES
code	codeSummary	varchar(255)	YES
code	location	varchar(255)	YES
code	userID	varchar(255)	YES
code	uploadTime	varchar(255)	YES
code	areaID	int(255)	YES
code	source	varchar(255)	YES
comment 表			
comment	commentID	int(255)	NO
comment	userID	varchar(255)	YES
comment	resourceID	int(255)	YES
comment	commentTime	varchar(255)	YES
comment	commentContent	varchar(255)	YES
Dataset 表			
dataset	datasetID	int(255)	NO
dataset	datasetName	varchar(255)	YES
dataset	source	varchar(255)	YES
dataset	datasetSummary	varchar(255)	YES
dataset	location	varchar(255)	YES
dataset	userID	varchar(255)	YES
dataset	uploadTime	varchar(255)	YES
dataset	areaID	int(255)	YES
dataset	datasetWriter	varchar(255)	YES
Essay 表			
essay	essayID	int(11)	NO
essay	essayName	varchar(255)	YES
essay	essayWriter	varchar(255)	YES
essay	source	varchar(255)	YES
essay	essaySummary	varchar(255)	YES
essay	location	varchar(255)	YES
essay	userID	varchar(255)	YES
essay	uploadTime	varchar(255)	YES
essay	areaID	int(255)	YES
Essay2code 表			
essay2code	essayID	int(255)	NO
essay2code	codeID	int(255)	NO
essay2code	userid	varchar(255)	YES
essay2code	linktime	varchar(255)	YES
Essay2dataset 表			
essay2dataset	essayID	int(255)	NO
essay2dataset	datasetID	int(255)	NO
essay2dataset	userid	varchar(255)	YES
essay2dataset	linktime	varchar(255)	YES

Resource 表			
resource	resourceID	int(255)	NO
resource	resourceType	varchar(255)	YES
User 表			
user	userID	varchar(255)	NO
user	password	varchar(255)	YES
user	userType	varchar(255)	YES
user	name	varchar(255)	YES
user	sex	varchar(255)	YES
user	phone	varchar(255)	YES

3. 系统架构设计

本系统基于 SpringBoot 创建 SSM (Spring + Spring MVC + MyBatis) 框架，清晰的区分了模型层、视图层和控制层。既满足了系统当前的需求，也保证了系统的稳定运行，AOP 编程也保证了对系统进一步扩展以及维护的可行性。系统架构设计说明如图 3.1 所示：



(1) 模型层 (Model)

完成业务逻辑、数据持久化工作，通过 MyBatis 来实现数据库的操作，通过

javabeans 存放实体对象（Entity 类）通过 service 进行业务逻辑的运算。

① mybatis-config.xml 文件配置

```
<configuration>
    <!-- 驼峰命名法 -->
    <settings>
        <setting name="mapUnderscoreToCamelCase" value="true"/>
    </settings>
    <!-- 将实体中的所有类配置别名, 引用实体类时可以使用实体类的别名来代替实体类 -->
    <typeAliases>
        <package name="com.bjfu.wzs.resource-sharingplatform.bean" />
    </typeAliases>
    <!-- 将所有 mapper 映射文件引入 -->
    <mappers>
        <package name="com.bjfu.wzs.resource-sharingplatform.dao" />
    </mappers>
</configuration>
```

(2) 视图层 (View)

负责展示用户界面，是整个MVC架构中唯一显示给用户看，同时接收用户的输入，显示处理结果，主要由 存放在resources/templates文件夹下HTML文件利用Thymleaf来显示视图，选取了LayUI框架作为系统的前端框架， Static目录下的css、js、images目录下存放了各种资源文件。

(3) 控制层 (Control)

主要负责视图层和模型层之间的连接，根据用户的输入来控制用户界面数据的显示。它由Spring托管的ServletDispatcher来实现控制，从视图层接收用户请求，传递给模型层，调用相应bean对象和通过Mybatis来访问数据库，然后Controller把处理结果存入ModelAndView，经过视图解析器传递到相应页面，协同完成请求。本系统的Controller功能如表 3.6 所示。

Controller名称	功能
AdminManageController	进行管理员相关操作的控制
CommentController	进行评论和建立关联的控制
DownloadController	进行下载相关操作的控制
GeneralController	进行相关页面的映射控制

LoginController	进行登陆注册相关操作的控制
PersonManagementController	进行个人信息和个人资源管理操作的控制
StatisticController	进行资源统计功能相关操作的控制
ResouController	进行资源搜索显示等相关功能操作的控制
UploadController	进行上传资源相关操作的控制

(4) 系统配置设置

由于系统基于 SpringBoot 创建项目，整体 Maven 依赖引入位于 Pom.xml。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.2.2.RELEASE</version>
    <relativePath/> <!-- Lookup parent from repository -->
  </parent>
  <groupId>com.bjfu.wzs</groupId>
  <artifactId>resourcesharingplatform</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>resourcesharingplatform</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
      <exclusions>
```

```

        <exclusion>
            <groupId>org.junit.vintage</groupId>
            <artifactId>junit-vintage-engine</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<!-- MySQL 的 JDBC 驱动包，用 JDBC 连接 MySQL 数据库时必须使用该 jar 包-->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.4</version>
    <scope>runtime</scope>
</dependency>
<!-- 支持 jdbc 连接-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
<!-- 官方说明: MyBatis Spring-Boot-Starter will help you use MyBatis with Spring
Boot-->
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>1.3.3</version>
</dependency>
<!-- 对 Thymeleaf 模板引擎的支持，包括和 Spring 的集成-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<!-- devtools 热部署-->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <optional>true</optional>
    <scope>true</scope>
</dependency>
<dependency>
    <groupId>com.vaadin.external.google</groupId>
    <artifactId>android-json</artifactId>
    <version>0.0.20131108.vaadin1</version>
    <scope>compile</scope>
</dependency>
<dependency>

```

```

        <groupId>org.json</groupId>
        <artifactId>json</artifactId>
        <version>20180813</version><!-- 注意: 20160810 版本不支持 JSONArray-->
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
        <plugin>
            <groupId>org.mybatis.generator</groupId>
            <artifactId>mybatis-generator-maven-plugin</artifactId>
            <version>1.3.2</version>
        </plugin>
    </plugins>
</build>
</project>

```

Spring 的配置信息位于 applicationContext.yml 和 applicationContext.properties 中

#连接mysql 数据库

spring:

datasource:

username: jeedesign

password: 123

url:

jdbc:mysql://39.97.252.94:3306/jeedesign?useSSL=false&useUnicode=true&characterEncoding=UTF-8

username: root

password: 981204

url:

jdbc:mysql://localhost:3306/resourcesharing?useSSL=false&useUnicode=true&characterEncoding=UTF-8

driver-class-name: com.mysql.jdbc.Driver

整合 mybatis

mybatis:

config-location: classpath:mybatis/mybatis-config.xml

mapper-locations: classpath:mybatis/mapper/*.xml

```

## 开放 80 端口
server.port=8080

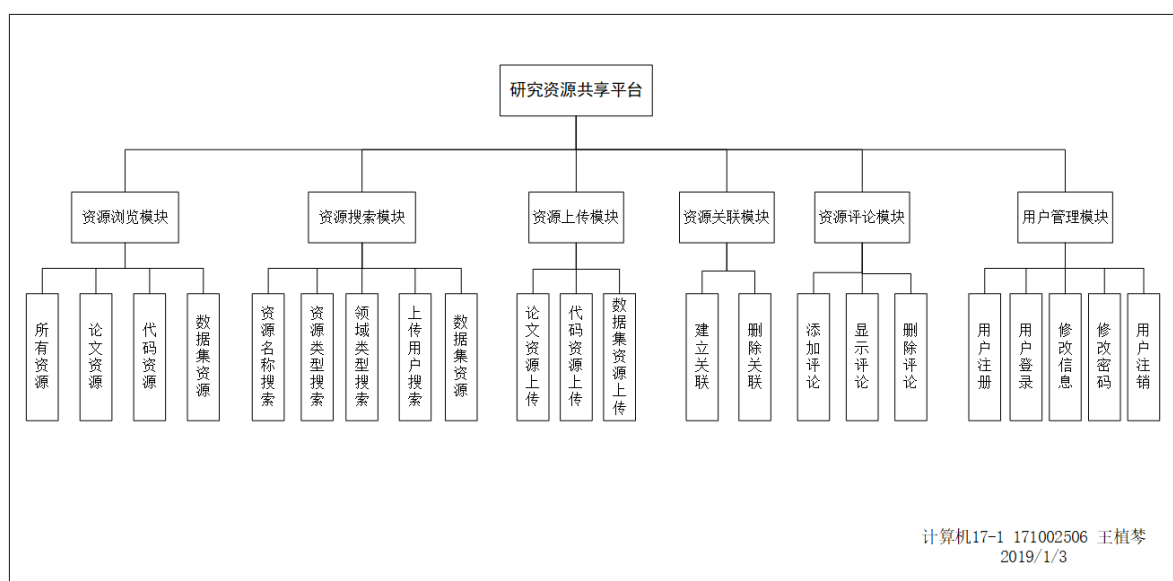
server.servlet.context-path=/

## 日志记录级别
logging.level.bjfu.wzs=trace
logging.level.bjfu.wzs.resourcesharingplatform.dao=debug

# 上传文件总的最大值
spring.servlet.multipart.max-request-size=10MB
# 单个文件的最大值
spring.servlet.multipart.max-file-size=10MB
# 是否支持批量上传 (默认值 true)
spring.servlet.multipart.enabled=true
# 上传文件的临时目录 (一般情况下不用特意修改)
spring.servlet.multipart.location=
# 文件大小阈值, 当大于这个阈值时将写入到磁盘, 否则存在内存中, (默认值 0 一般情况下不用特意修改)
spring.servlet.multipart.file-size-threshold=0
# 判断是否要延迟解析文件 (相当于懒加载, 一般情况下不用特意修改)
spring.servlet.multipart.resolve-lazily=false

```

4. 系统功能模块设计



(1) 资源浏览模块

用户登录系统后, 可进行所有资源浏览或者按类别浏览论文资源浏览或代码浏览或数

数据集浏览。

（2）资源搜索模块

用户登录系统后，可进行资源搜索。资源搜索分为按资源名称搜索、按资源类型搜索、按资源领域搜索和按上传用户搜索。。

（3）资源上传模块

资源上传分为按类型上传。共有论文上传，数据集上传，代码上传。

（4）资源关联模块

资源关联分为建立关联和删除关联。当删除数据集或代码时会自动删除与之建立的关联论文。

（5）资源评论模块

该模块分为建立评论，浏览评论，删除评论。当删除某资源时会自动删除与之对应的评论。

（6）用户管理模块

该模块分为用户登录，用户注册，用户注销，修改信息，修改密码。用户注销功能由管理员进行。

（二）详细设计与关键技术

1. 关键模块 1：资源关联

资源关联功能位于资源详细信息页面，提供给用户添加资源关联和显示相关联资源的功能。

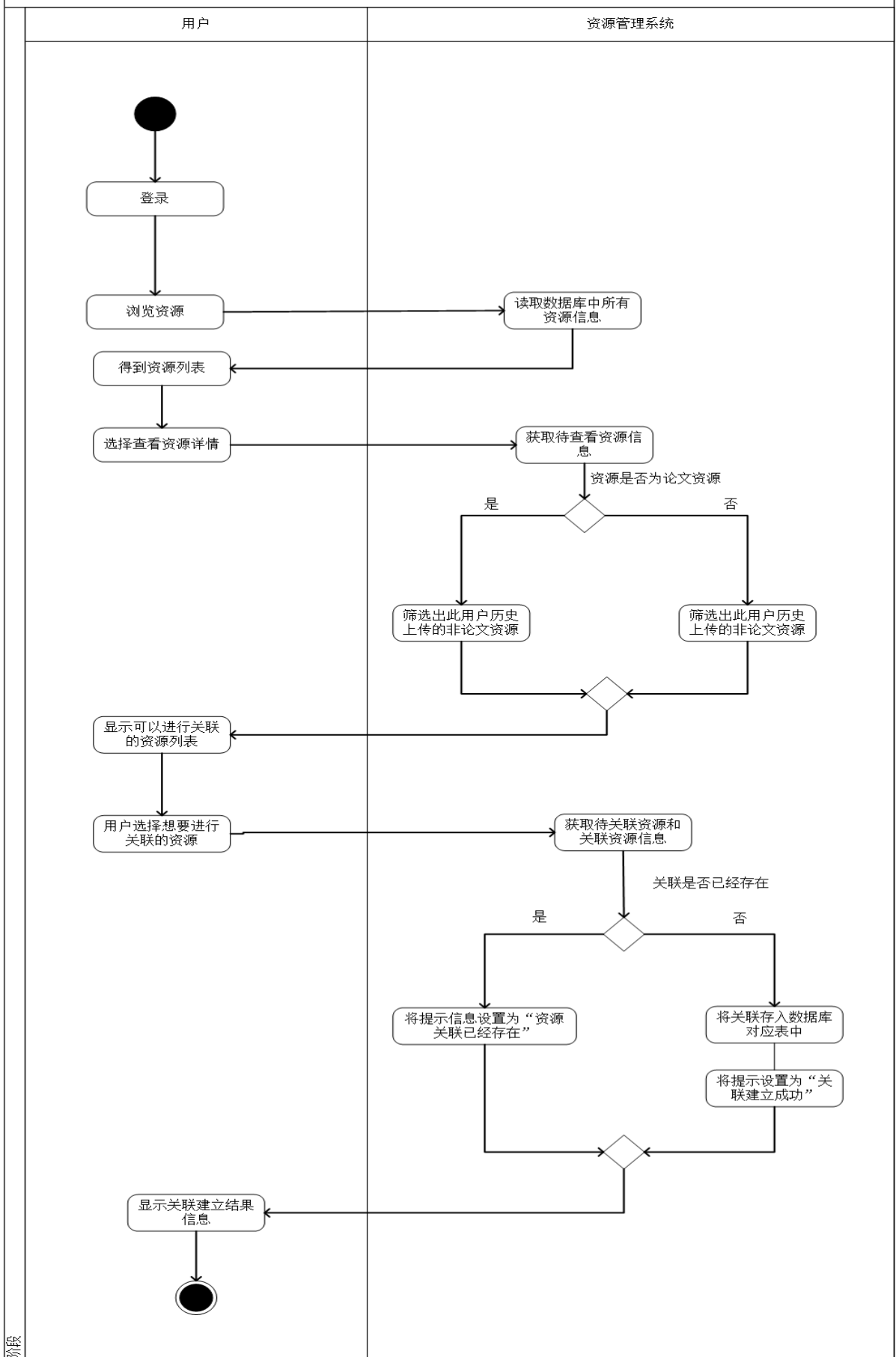
对于添加资源关联功能，出于系统的安全性和道德性考虑，只允许用户将自己上传的资源与其它资源进行关联。所以，添加资源关联分两种情况，一是为用户的论文资源添加代码资源或数据集资源，二是把用户的数据集资源或代码资源添加到其它论文资源下。

具体的业务流程如下：

- （a）判断详情页资源类型是论文资源还是非论文资源
- （b）如果是论文资源，则在添加资源下拉框中仅显示该用户之前上传的数据集和代码资源以供选择。
- （c）如果是非论文资源，则在添加资源下拉框中仅显示该用户之前上传的论文资源以供选择。
- （d）用户选择完想要关联的资源后点击建立关联按钮，前端通过 ajax 将待关联资源和关联资源的相关信息传送到服务器后台，进行资源关联的处理。
- （e）服务器后台 Controller 接收到两个资源后，调用 Service 处理业务流程，首先根据资源类型到 essay2code 表或 essay2dataset 表中查询资源关联是否已经存在，若关联已经存在则提示用户关联已存在无需重复建立关联，若关联不存在则将两个资源关联存入相关表中，并提示用户资源关联建立成功。

资源关联功能活动图如下所示：

资源关联功能活动图



2. 关键模块 2：资源上传

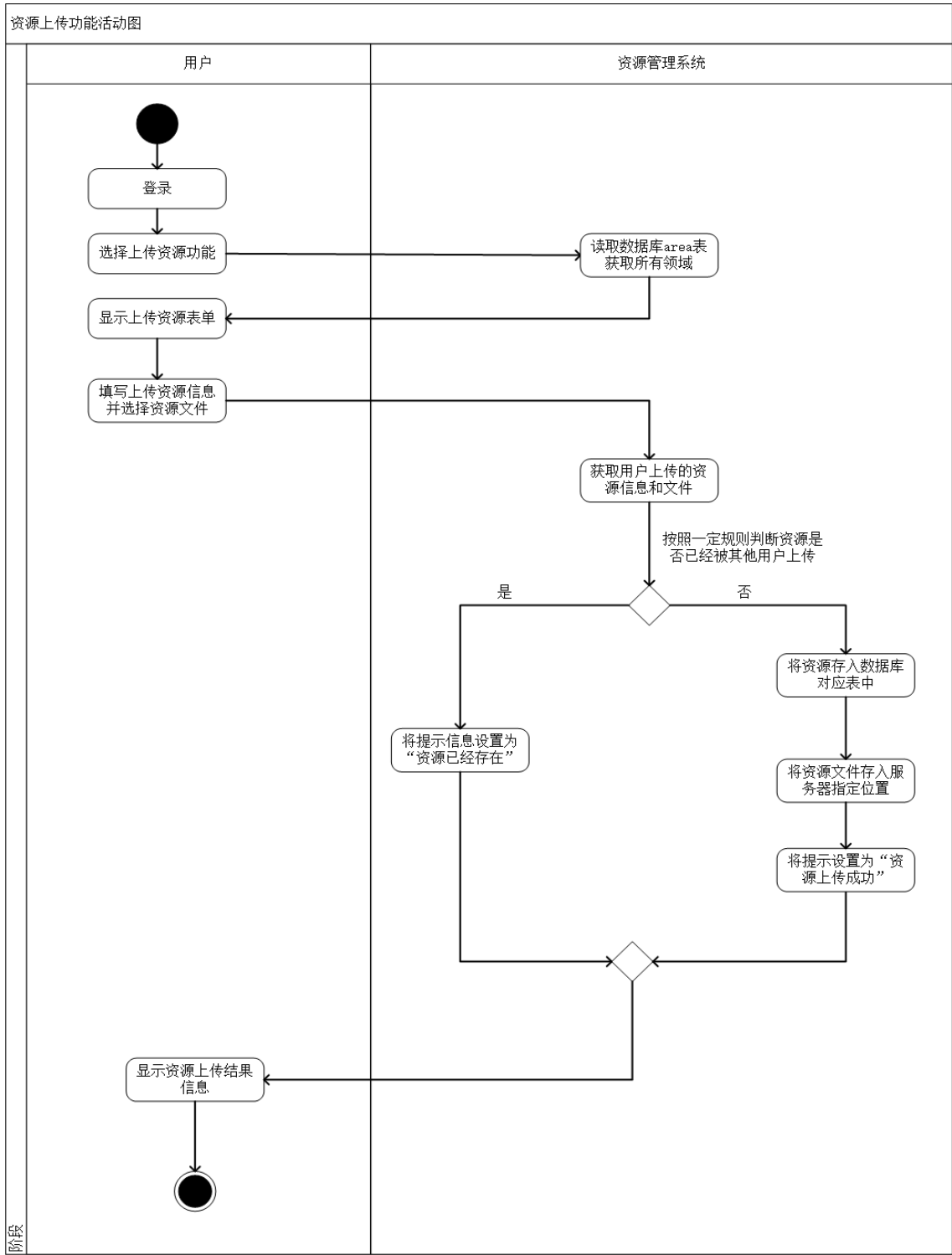
资源上传模块包括资源信息上传处理和资源文件上传两个子功能，是提供给普通用户的功能模块。

普通用户登录系统后可以在左侧导航栏资源中心模块选择上传资源，进入上传资源界面。填写资源信息并提交资源文件后数据通过 AJAX 异步传输到后台 UploadController 中，Controller 调用 UploadService 进行具体的业务处理。

具体的业务流程如下：

- (a) 获取用户上传的资源信息和资源文件。
- (b) 访问资源数据库，根据资源名称、资源作者和资源类型进行查重操作。
- (c) 如果资源重复，则提示用户资源已经存在，无法上传。
- (d) 如果资源不存在，则接收用户提交的资源信息和资源文件，存入到 resource 表和对应的 essay 或 code 或 dataset 表中，并且将资源文件保存在服务器指定位置。
- (e) 前端响应用户上传资源情况。

资源关联功能活动图如下所示：



3. 关键模块 3：资源删除

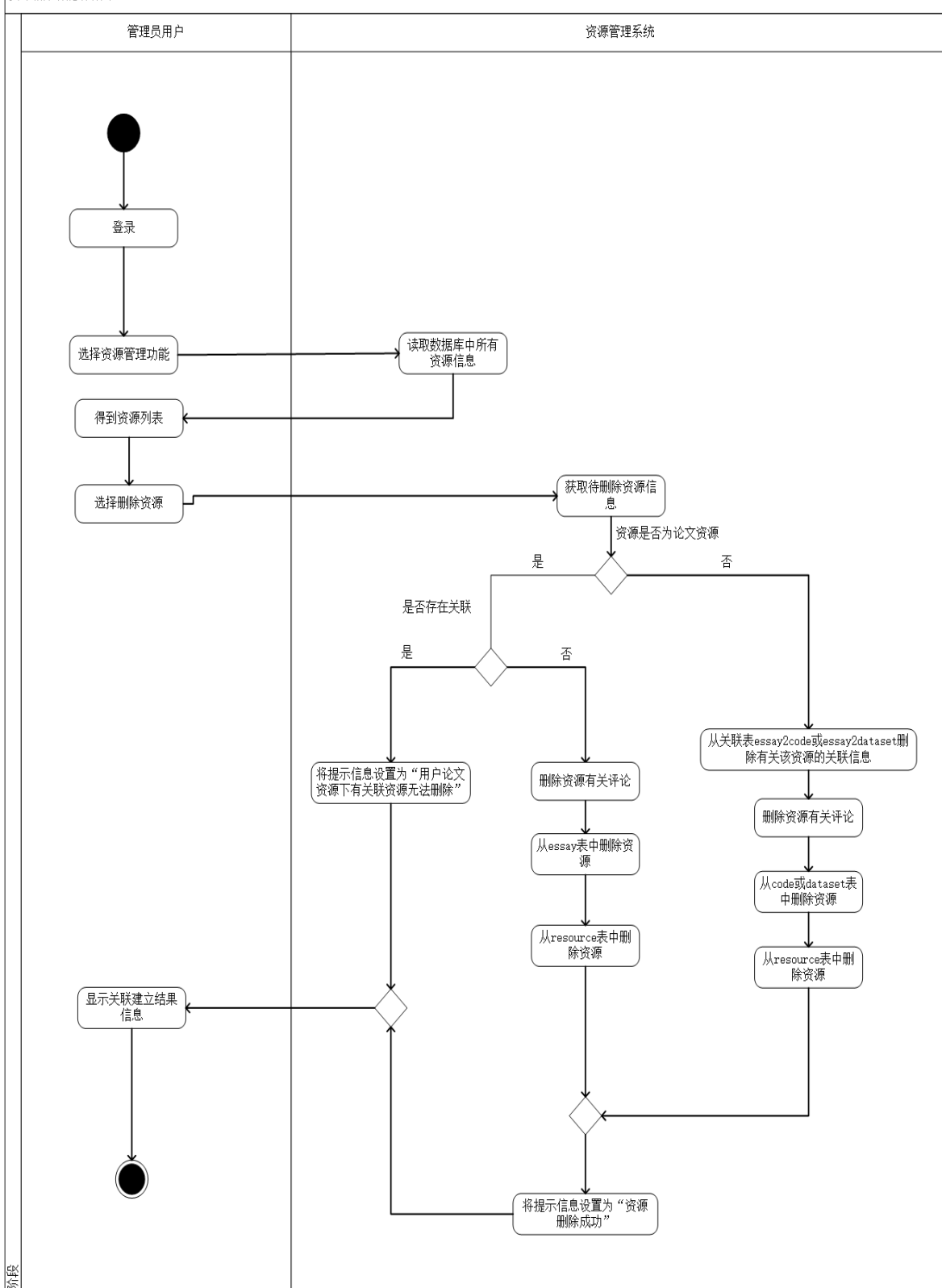
资源删除功能是只提供给管理员的功能模块，管理员登录后在左侧导航栏资源管理模块选择资源删除，页面基于 LayUI 动态表格显示平台所有资源，上方提供资源检索功能模块，可以进行多条件（包括上传用户，资源名，资源种类和资源领域*动态生成）模糊搜索符合条件的资源，右侧设置资源详情按钮和删除资源按钮。点击资源详情页面进入管理员资源详情页面，由于管理员身份的功能限制，详情页面只显示与该资源相关联的资源 and 评论而不允许评论和建立资源关联。点击删除资源按钮，提示确认框要求管理员确认删除操作，若确认，则将该条资源的信息通过 ajax 传输到后台 AdminManageController，在 Controller 里调用 ResourceService 进行删除资源的业务流程处理。

在删除操作时，进行的具体业务流程如下：

- (a) 判断要删除的资源为论文或非论文资源
- (b) 若是论文资源，则判断论文资源是否存在与其它资源的关联，如果存在资源关联则提示用户资源无法删除，要求用户先删除关联才可删除该论文。如果论文无相关联资源，则进行删除操作。删除操作按以下流程进行：先删除该论文资源下的用户评论（comment 表），再删除 essay 表中的论文信息，最后删除 resource 表中的该论文信息。
- (c) 若是非论文资源，则直接允许用户进行删除操作，删除操作按以下流程进行：首先在 Comment 表中删除用户对于该资源的所有评论，然后在 essay2code 表或 essay2dataset 表中删除该资源与论文的关联，再删除 code 表或 dataset 表中的资源信息，最后删除 resource 表中的该资源信息。

资源关联功能活动图如下所示：

资源删除功能活动图



4.关键模块 4： 系统安全性

出于对系统安全性和程序道德性的考量，本平台做出以下优化：

1) 设置非法访问拦截器

SpringMVC 中的 Interceptor 拦截器是相当重要和相当有用的，它的主要作用是拦截用户的请求并进行相应的处理。本系统采用实现 Spring 的 HandlerInterceptor 接口然后继承实现 HandlerInterceptor 接口的类 HandlerInterceptorAdapter，通过它来进行权限验证和判断用户是否登录，进而阻止用户非法访问路径。

2) 用户密码加密

对于注册到平台的用户的密码采取 SHA-256 加密处理，保证了每个用户的隐私。

3) PreparedStatement 防止 SQL 注入

本系统使用 Mybatis 统一管理 SQL，对数据库进行存取操作。所有 SQL 均用 PreparedStatement，有效防止了 SQL 注入

4) 程序业务逻辑中安全性处理

在程序业务逻辑实现过程中，首先保证所有操作均在确认用户身份之后进行，在某些涉及到用户资源访问权限的地方设置判断，例如添加资源关联时只允许用户将自己资源

（三）系统部署与发布

1. 系统部署环境配置

1) Web 服务器：Apache Tomcat/9.0.29

Tomcat 是一个轻量级应用服务器，是 Apache 软件基金会（Apache Software Foundation）的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。Tomcat 性能稳定、技术先进，运行时占用的系统资源小，而且免费并在不断改进完善中，深受部分软件开发者和广大程序员的喜爱，成为目前比较流行的 Web 应用服务器。

2) 数据库服务器：MySQL

MySQL 是一种关系型数据库管理系统，关联数据库在不同的表中存储数据，而不是放在一个大仓库中，可提高数据操作的速度和灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。由于其体积小、速度快、总体成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

3) 运行系统环境：Windows/Linux

2. 系统部署说明

本系统已经部署在本地服务器和阿里云服务器中，部署端口号为 8080。

由于本系统基于 SpringBoot 开发部署，导出的程序 jar 包仅要求服务器主机操作系统含有 Java 运行环境即可。

1) 启动项目

Windows 操作系统下在控制台 CMD 输入：`java -jar resourcesharing1.0.1.jar`

Linux 操作系统下在控制台 terminal 输入：`java -jar resourcesharing1.0.1.jar`

2) 浏览器访问

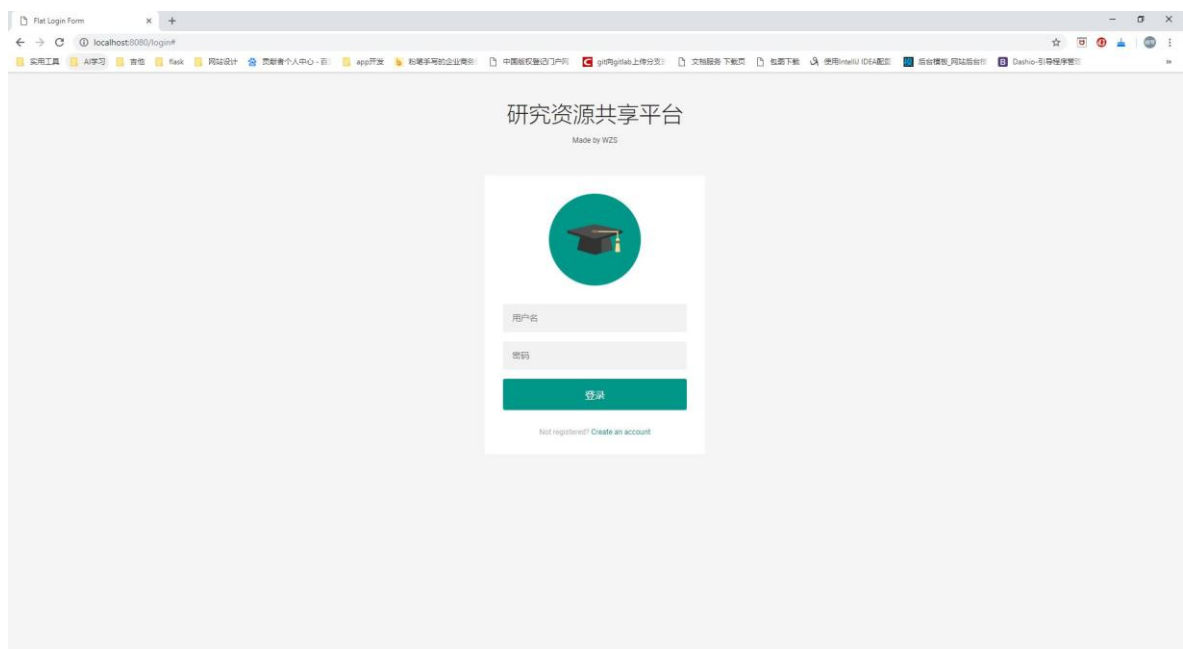
若访问本地服务器下的项目在地址栏输入：`http://localhost:8080/login`

若访问阿里云服务器下的项目在地址栏输入：`http://39.97.252.94:8080/login`

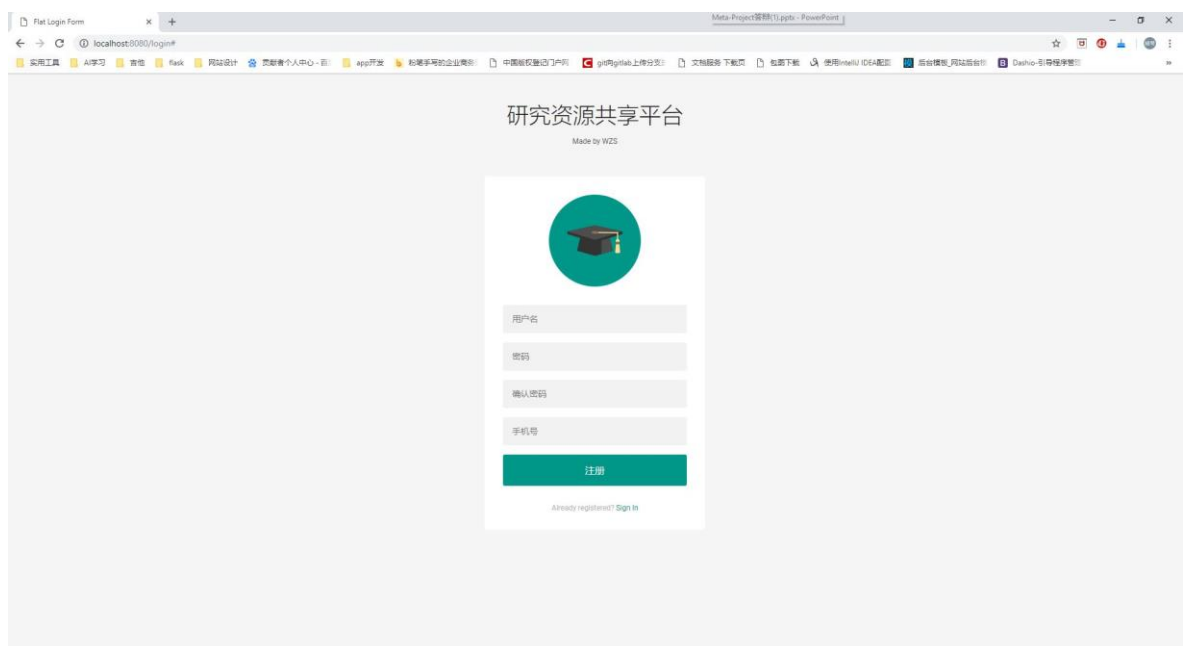
3. 系统部署结果展示

在系统的运行过程中，通过对系统的整体页面布局和按钮、图片的位置大小进行了细微调整，使系统界面更加友好、美观；在系统退出和界面跳转间添加了提示信息，使系统的提示功能得到进一步完善，增强了系统的用户导向性。系统运行结果表明：本系统操作流畅，页面设计简洁美观，体系结构完整，功能模块合理。

登录页面



注册页面

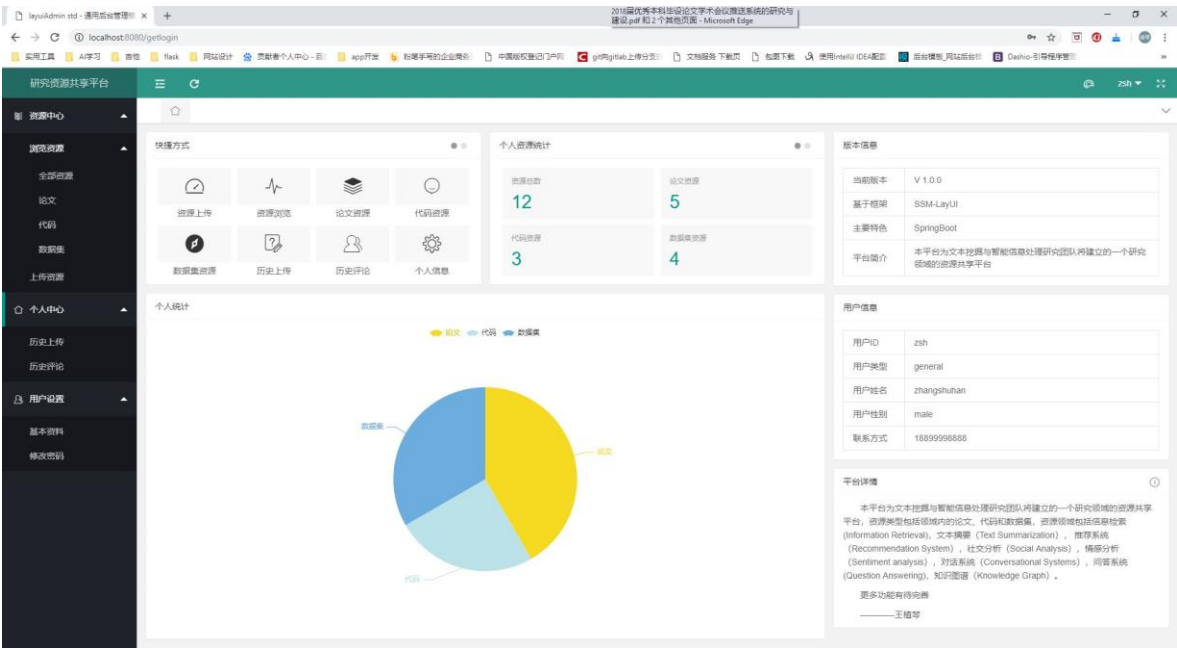


普通用户主页面

页面上方显示出登陆用户名并一直延续到后面相关页面。

左侧导航栏展示资源中心（包括浏览资源：其中有全部资源浏览和分类浏览功能入口，上传资源入口）、个人中心（包括历史上传页入口和历史评论入口）和用户设置（包括基本资料和修改密码功能入口）。

右侧主页面按照模块分别显示功能快捷方式、个人资源统计概览和统计图、版本信息、用户信息以及平台简介。



浏览资源页面（分为全部资源页面，论文资源页面，代码资源页面和数据及资源页面）

所有资源页面均以 LayUI 动态表格形式展示所有资源基本信息，包括(资源 ID，资源类型，资源名称，资源作者，资源领域，资源来源，资源简介，上传用户，上传时间)等，右侧设置详情按钮

点击详情按钮进入资源详情页面，页面包括资源详情信息、资源关联建立与查看区域、资源评论与查看评论区域。

上方设置多条件搜索区，可以进行多条件（包括上传用户，资源名，资源种类和资源领域*动态生成）模糊搜索。

资源id	资源类型	资源名称	资源作者	资源来源	资源简介	资源上传用户	资源上传时间	资源领域	
1	论文	A	A	A	A	zsh	2019-12-21 00:2...	信息检索	查看详情
2	论文	B	B	B	B	zsh	2019-12-21 00:2...	文本摘要	查看详情
3	数据集	dataset1	ssss	dataset1	dataset1	zsh	2019-12-21 00:2...	对话系统	查看详情
4	代码	code1	code1	1	code1	zsh	2019-12-21 00:2...	对话系统	查看详情
5	论文	C	C	C	C	zsh	2019-12-21 00:2...	信息检索	查看详情
8	论文	第一次论文...	王德等	王德等	这是我的第一次论文提交测试	zsh	2019-12-21 00:2...	文本摘要	查看详情
9	代码	改中文	王德等	王德等	改中文后第一次测试	zsh	2019-12-21 00:3...	文本摘要	查看详情
10	数据集	第三次	王德等	王德等	第三次测试	zsh	2019-12-21 00:4...	信息检索	查看详情
11	数据集	myc	myc	123	123	zsh	2019-12-21 15:4...	信息检索	查看详情
12	论文	测试上传啊	王德等啊	王德等啊	王德等测试上传文件	zsh	2019-12-23 17:5...	推荐系统	查看详情
13	代码	上传测试	王德等	王德等	王德等	zsh	2019-12-23 18:1...	文本摘要	查看详情
14	数据集	222	22	22	111	zsh	2020-01-04 00:4...	文本摘要	查看详情

资源详情页面

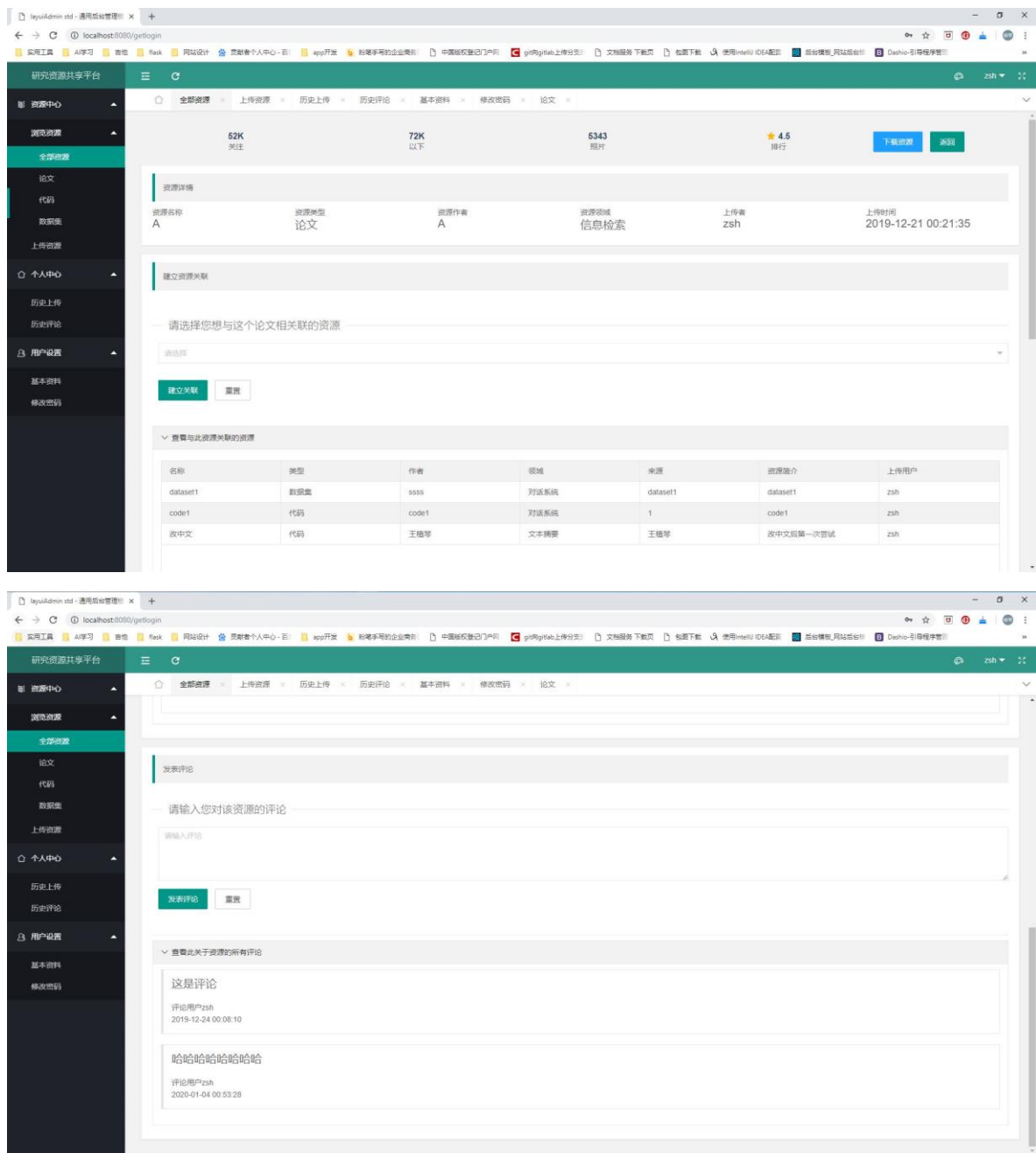
资源详情页面的目的是展示资源的详细信息，提供资源的下载功能，进行资源关联的查看和添加，进行资源评论的查看和添加。

资源关联添加

根据资源的类型显示添加资源的类型，如果该资源为论文资源，则在添加资源关联模块只显示登录用户上传过的非论文资源，反之，如果论文资源为非论文（代码资源或数据集资源），则在添加资源关联模块只显示该用户上传过的论文资源，即只允许用户将非论文资源与论文资源进行关联

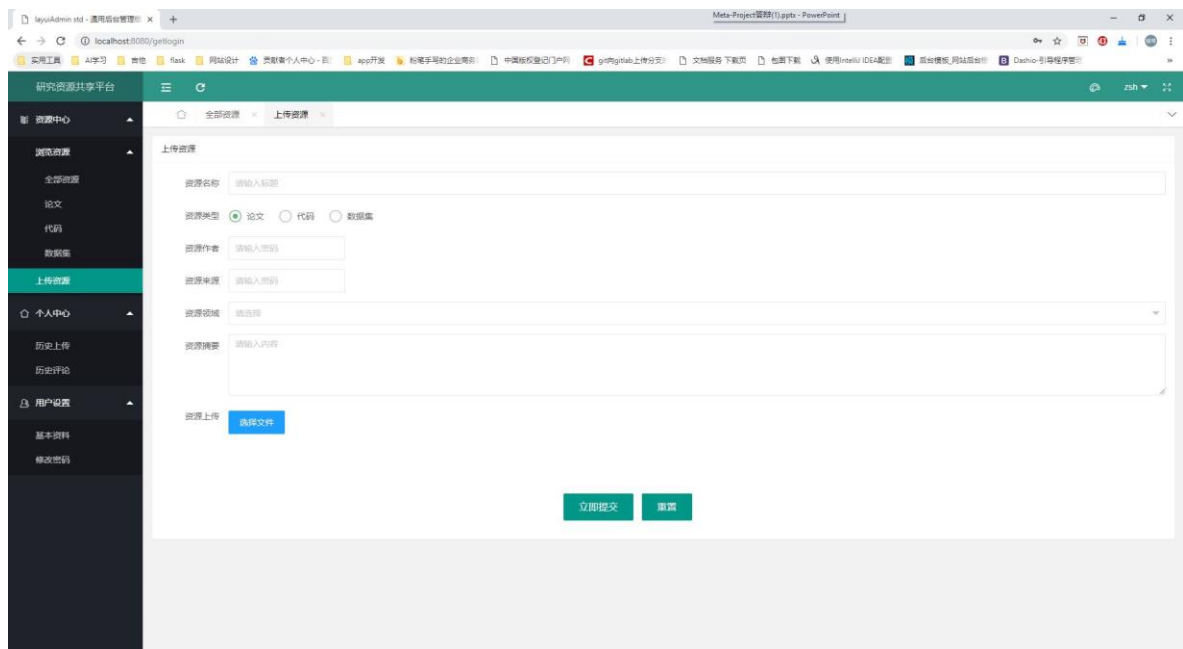
资源评论

资源评论区设置评论提交和评论显示两部分，用户可以提交对该资源的评论，也可以在下方浏览其他用户和自己对于该资源的评论。



资源上传页面

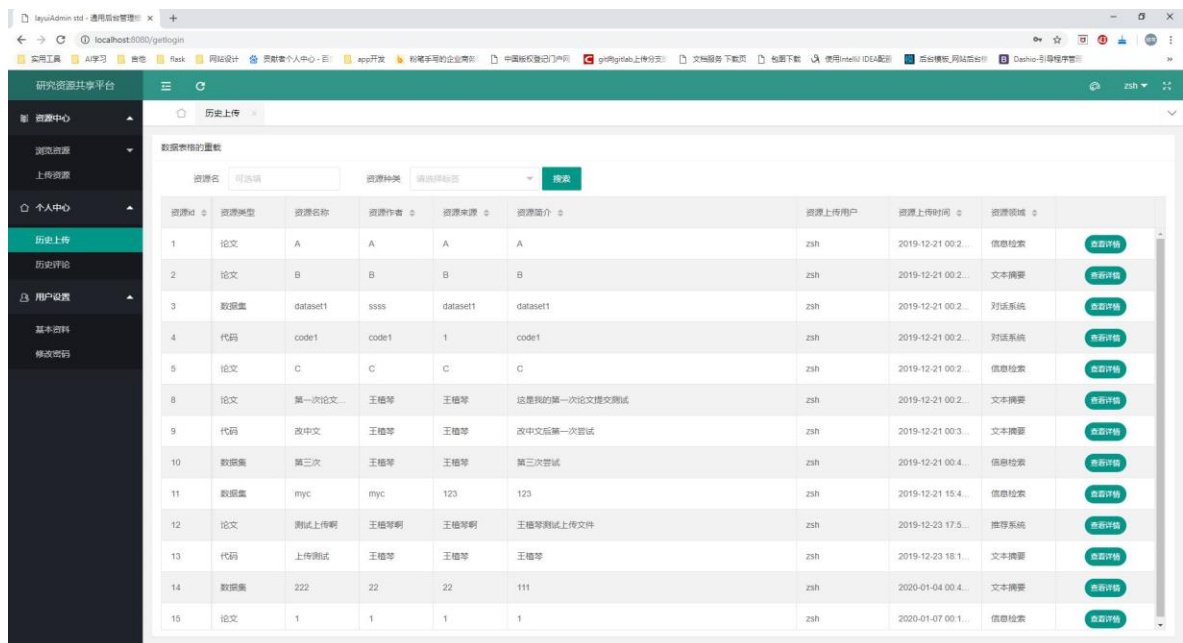
上传资源界面基于 LayUI-Form 设计，提示用户填写资源名称，资源类型，资源作者，资源来源，资源领域（请求后台数据动态生成下拉框）和资源摘要，下方为上传资源按钮，其中有些内容通过 filter 设置了必填项（在未填写情况下弹出框提醒用户必填项不能为空），最下方是立即提交按钮和重置按钮。



历史上传页面

历史上传页面以 LayUI 动态表格形式展示有关该用户上传的资源基本信息，包括(资源 ID，资源类型，资源名称，资源作者，资源领域，资源来源，资源简介，上传用户，上传时间)等，右侧设置详情按钮。

上方设置多条件搜索区，可以进行多条件（包括资源名，资源种类）模糊搜索。



历史评论页面

历史评论页面以 LayUI 动态表格形式展示有关该用户对资源的历史评论信息，包

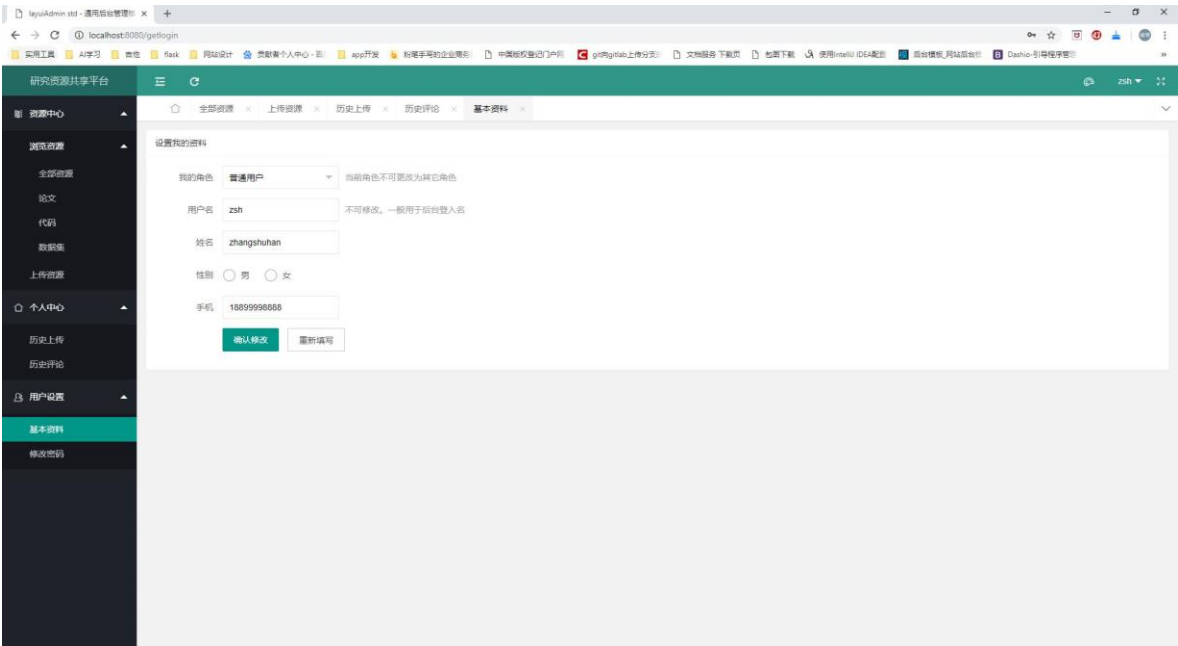
括(评论 id、评论资源名、资源类型、评论时间和评论类型)，右侧设置详情按钮。

点击详情按钮跳转到评论资源的详情页面。

上方设置多条件搜索区，可以进行多条件（包括评论资源名，评论资源种类）模糊搜索

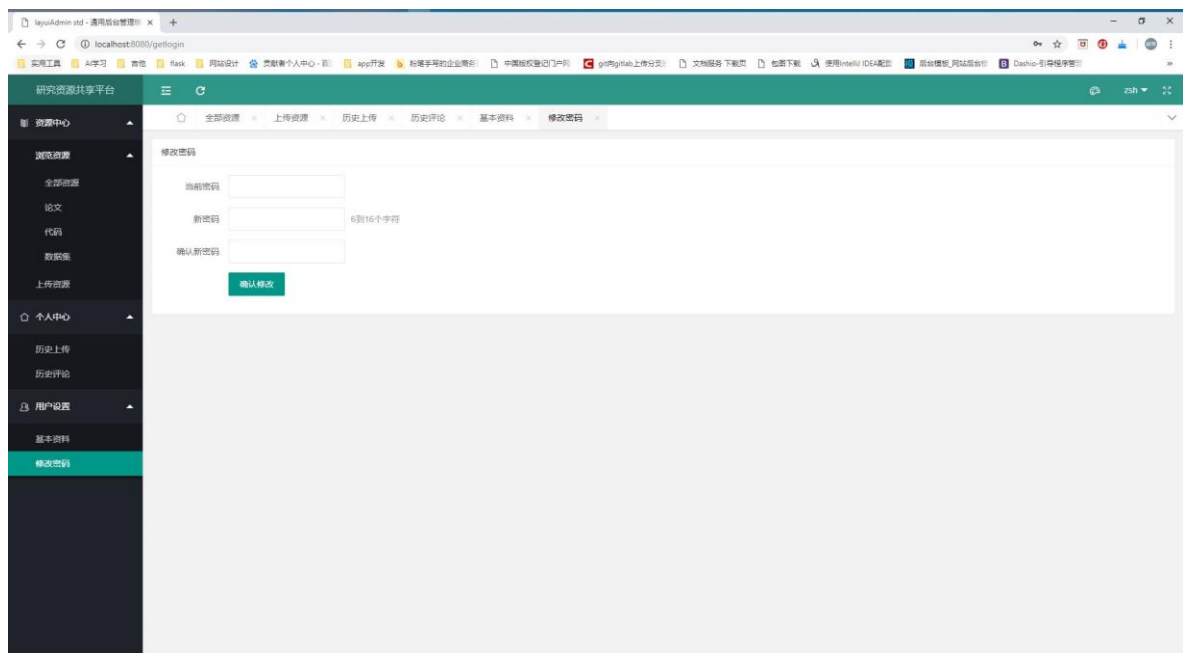
基本资料页面

基本资料页面显示登陆用户的用户身份，用户 ID，姓名，性别和联系方式，只允许修改姓名，性别和联系方式，其余均为不可更改项。



修改密码页面

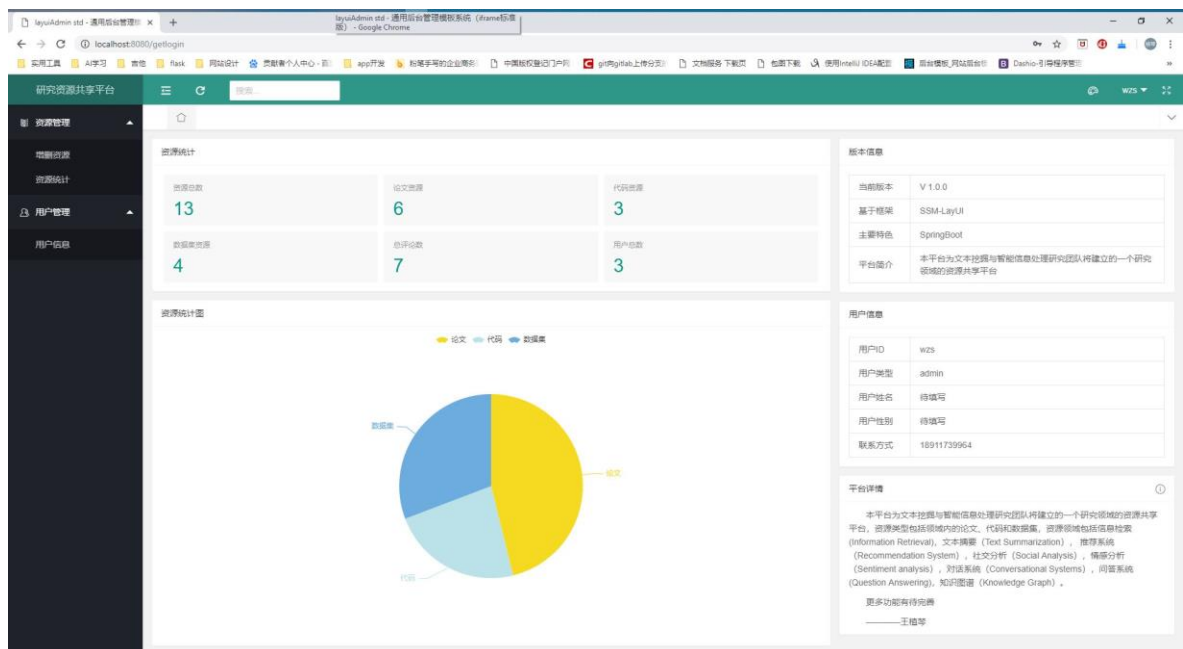
修改密码页面要求用户填写旧密码以确认身份，旧密码正确的前提下验证新密码两次输入是否相同，如果相同则修改用户密码。



管理员主页面

页面上方显示出登陆用户名并一直延续到后面相关页面。

- 左侧导航栏展示资源管理（包括资源删除功能和资源统计功能）和用户信息管理两个选项，可分别进入对应的管理页面。
- 右侧主页面按照模块分别显示资源统计情况，统计图，版本信息，用户信息，平台简介。



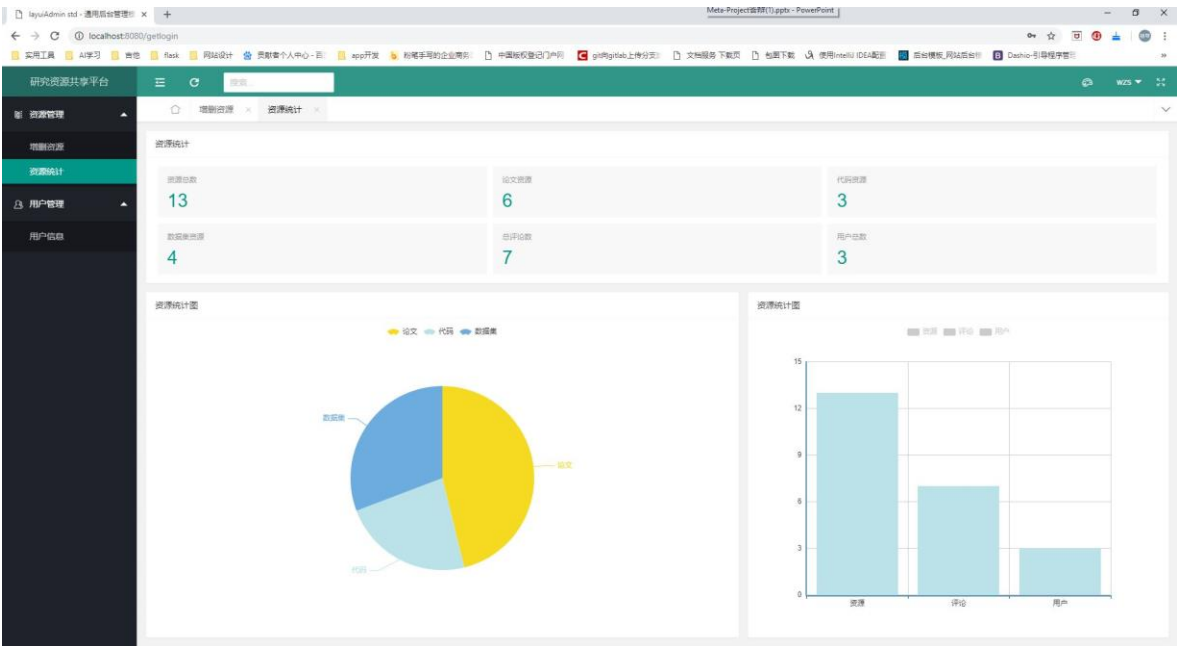
删除资源管理页面

以 LayUI 动态表格形式展示所有资源基本信息，包括(资源 ID，资源类型，资源名称，资源作者，资源领域，资源来源，资源简介，上传用户，上传时间)等，右侧设置详情按钮和删除按钮。

- a) 点击详情按钮可进入该资源详情页面，由于是管理员身份，资源系详情页面只显示资源详情信息、资源关联情况和资源评论情况。
- b) 点击删除按钮，弹出确认删除对话框，需要再次确认之后，再进行资源的删除。

资源ID	资源类型	资源名称	资源作者	资源来源	资源简介	资源上传用户	资源上传时间	资源领域	操作
1	论文	A	A	A	A	zsh	2019-12-21 00:2...	信息检索	详情 删除
2	论文	B	B	B	B	zsh	2019-12-21 00:2...	文本摘要	详情 删除
3	数据集	dataset1	ssss	dataset1	dataset1	zsh	2019-12-21 00:2...	对话系统	详情 删除
4	代码	code1	code1	1	code1	zsh	2019-12-21 00:2...	对话系统	详情 删除
5	论文	C	C	C	C	zsh	2019-12-21 00:2...	信息检索	详情 删除
6	论文	第一次论文...	王德等	王德等	这是我的第一次论文提交测试	zsh	2019-12-21 00:2...	文本摘要	详情 删除
9	代码	政中文	王德等	王德等	政中文后第一次测试	zsh	2019-12-21 00:3...	文本摘要	详情 删除
10	数据集	第三次	王德等	王德等	第三次测试	zsh	2019-12-21 00:4...	信息检索	详情 删除
11	数据集	myc	myc	123	123	zsh	2019-12-21 15:4...	信息检索	详情 删除
12	论文	测试上传啊	王德等啊	王德等啊	王德等测试上传文件	zsh	2019-12-23 17:5...	推荐系统	详情 删除
13	代码	上传测试	王德等	王德等	王德等	zsh	2019-12-23 18:1...	文本摘要	详情 删除
14	数据集	222	22	22	111	zsh	2020-01-04 00:4...	文本摘要	详情 删除
15	论文	1	1	1	1	zsh	2020-01-07 00:1...	信息检索	详情 删除

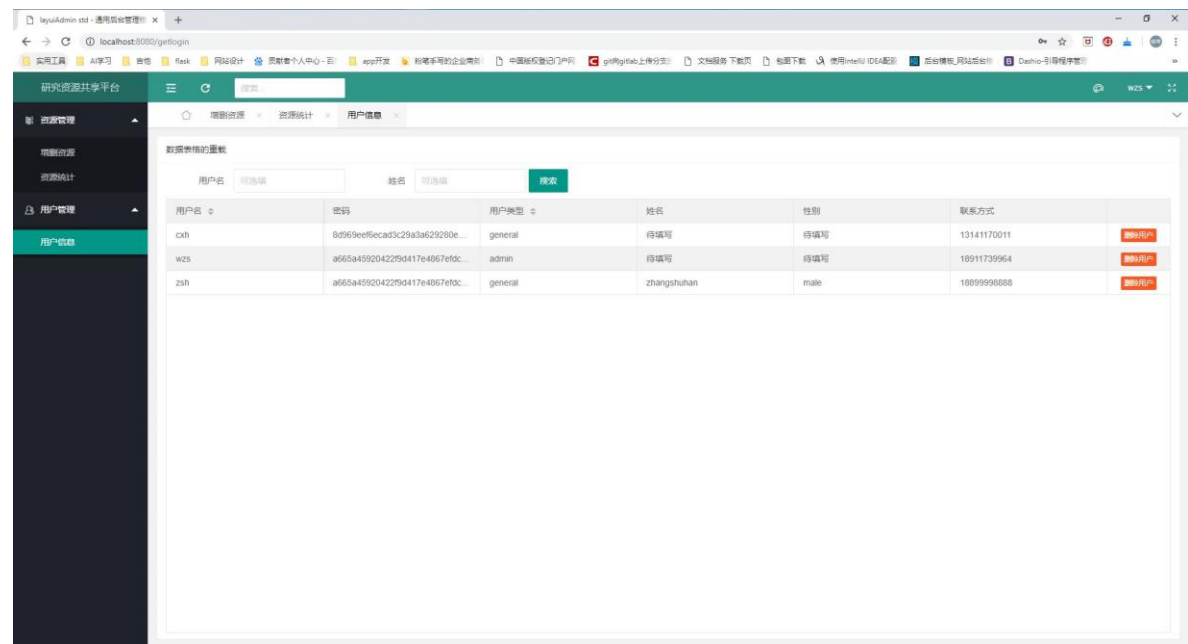
系统资源统计



用户管理页面

以 LayUI 动态表格形式展示所有用户基本信息，包括(用户名，用户密码（经过 SHA256 加密），用户类型，姓名，联系方式)，右侧设置删除用户按钮。

点击删除用户按钮，弹出确认删除对话框，需要再次确认之后，再进行用户的删除。



部署情况

```

:: Spring Boot :: (v2.2.2.RELEASE)

2020-01-07 00:05:32.876 INFO 23320 --- [ restartedMain] b.w.r.ResourceSharingPlatformApplication : Starting ResourceSharingPlatformApplication on DESKTOP-BDUB2T7 with PID 23320 (
2020-01-07 00:05:32.882 INFO 23320 --- [ restartedMain] b.w.r.ResourceSharingPlatformApplication : No active profile set, falling back to default profiles: default
2020-01-07 00:05:33.013 INFO 23320 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : DevTools property defaults active! Set 'spring.devtools.add-properties' to 'fal
2020-01-07 00:05:33.014 INFO 23320 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' prop
2020-01-07 00:05:35.943 INFO 23320 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2020-01-07 00:05:35.972 INFO 23320 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2020-01-07 00:05:35.973 INFO 23320 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.29]
2020-01-07 00:05:35.983 INFO 23320 --- [ restartedMain] o.a.catalina.core.AprLifecycleListener : Loaded APR based Apache Tomcat Native library [1.2.23] using APR version [1.7.0
2020-01-07 00:05:35.984 INFO 23320 --- [ restartedMain] o.a.catalina.core.AprLifecycleListener : APR capabilities: IPv6 [true], sendfile [true], accept filters [false], random [
2020-01-07 00:05:35.984 INFO 23320 --- [ restartedMain] o.a.catalina.core.AprLifecycleListener : APR/OpenSSL configuration: useAprConnector [false], useOpenSSL [true]
2020-01-07 00:05:36.020 INFO 23320 --- [ restartedMain] o.a.catalina.core.AprLifecycleListener : OpenSSL successfully initialized [OpenSSL 1.1.1c 28 May 2019]
2020-01-07 00:05:36.275 INFO 23320 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2020-01-07 00:05:36.275 INFO 23320 --- [ restartedMain] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 3261 ms
2020-01-07 00:05:37.749 INFO 23320 --- [ restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-01-07 00:05:38.027 INFO 23320 --- [ restartedMain] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page template: index
2020-01-07 00:05:39.686 INFO 23320 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2020-01-07 00:05:39.761 INFO 23320 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2020-01-07 00:05:39.765 INFO 23320 --- [ restartedMain] b.w.r.ResourceSharingPlatformApplication : Started ResourceSharingPlatformApplication in 7.819 seconds (JVM running for 11.
2020-01-07 00:05:46.903 INFO 23320 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2020-01-07 00:05:46.904 INFO 23320 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2020-01-07 00:05:46.915 INFO 23320 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 11 ms
2020-01-07 00:05:50.738 INFO 23320 --- [nio-8080-exec-2] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2020-01-07 00:05:51.053 INFO 23320 --- [nio-8080-exec-2] com.zaxxer.hikari.pool.PoolBase : HikariPool-1 - Driver does not support get/set network timeout for connections.
```


（四）结论分析

1. 问题与解决方法

1) 前端 JavaScript 知识欠缺，导致一些复杂的用户交互无法实现。

解决方案：采取 LayUI 框架模板，自学 JavaScript 基础知识。

2) SpringMVC 框架搭建复杂且依赖的包较多。

解决方案：采用 SpringBoot 创建项目

3) 系统安全性控制较难

解决方案：

- a) 设置非法访问拦截器
- b) PreparedStatement 防止 SQL 注入
- c) 用户密码进行 SHA-256 加密
- d) 在程序业务逻辑中进行身份验证等安全性处理

2. 收获和体会

通过此次课设项目的设计、实施与部署，体会到了进行全栈开发的过程。

我在这次课设中最大的收获，不仅仅是完成了一个网站项目，而是更新了自己的学习方式，改变了自己的学习观念。现在我对编程充满热情，懂得什么时候应该执拗的进行练习，什么时候应该停下来思考总结。

在这次课设中有以下三个方面令我感触最深

1, 编程的 self-teach

学习其实是自己教授自己的过程，在学习编程的道路上，按顺序做到这三点就可以进步很快。重复训练，形成肌肉记忆，获得反馈。思考，总结，归纳，演绎。面对 bug 的时候情绪要稳定，要能够耐心去解决这些 bug。

2, 有效的项目管理

项目管理是否有效，主要是看项目是否处于正常推进的状态。不要因为功能的实现细节止步不前，可以先接受不那么完美的点，推进了项目以后再去细节的优化。

3, 高效的 springBoot

SpringBoot 使得开发者不用重复造轮子,大大加快了开发速度。约定大于配置的原则,也为项目开发减少了很多繁琐的配置,使得开发者可以将精力集中在处理业务逻辑上,而不是繁多的配置问题上。

3. 尚存在的问题

1) 系统功能有待完善,后期可以添加更多功能

解决方案:采取面向切面编程,预留切面接口,方便后期增加功能。

2) 网站虽然是响应式前端,但是数据表格的移动端适配存在部分问题

3)