

# 中国石油大学胜利学院

## 本科生毕业设计（论文）

题    目： 基于 B/S 架构的学院信息  
交流平台设计与实现

学生姓名： \_\_\_\_\_

学    号： \_\_\_\_\_

院    系： \_\_\_\_\_

专业年级： \_\_\_\_\_

指导教师： 王雪冰

2017 年 6 月 9 日

## 摘 要

随着网络技术的蓬勃发展和信息化建设的推进，基于 B/S 架构的信息平台在各个领域都体现出了很大的需求空间，信息交流平台已然成为推进学校教学发展的重要手段。为提升学院信息交流的网络环境，针对学院现有平台的不足，分析设计了一整套功能较完善的信息交流系统方案。本系统是一个基于 SSH 框架开发,结合前端使用 Bootstrap 框架、Easyui 框架，并采用 Mysql 数据库的 B/S 三层体系结构的应用程序。是一个面向在校大学生，为其提供活动、社团和学生会信息管理和共享的综合的管理系统。实现管理员通过本系统可完成新闻通知发布、用户管理、角色管理、权限管理等多方面的工作以及网站维护等；学生则可通过本系统来参与活动、组织活动、参加社团以及参加学生会等功能，以此建立一个以学生为中心，广泛、快速、自由的信息交流平台。

**关键词：**信息交流平台；SSH 框架；应用程序

## ABSTRACT

With the vigorous development of network technology and the advancement of information construction, the information platform based on B/S architecture has a great demand space in various fields. The information exchange platform has become an important technical means to promote the development of school teaching. In order to improve the network environment of college information exchange, according to the shortcomings of the existing platform, the design and design of a complete set of functions of the better information exchange system program. The system is based on the SSH framework development, combined with front-end use Bootstrap framework, Easyui framework, and the use of Mysql database B/S three-tier architecture applications. Is a comprehensive management system for college students who provide activities, associations and student information management and sharing. Implementation of the system through the system to complete the news notification release, user management, role management, rights management and other aspects of the work and site maintenance; students can participate in activities through the system, organize activities, participate in community and participate in student associations and other functions , In order to establish a student-centered, extensive, fast and free information exchange platform.

**Keywords:** Information exchange platform; SSH framework; application program

# 目 录

第一章 绪论 .....	1
1.1 研究背景和意义 .....	1
1.2 国内外研究现状 .....	1
1.3 论文主要研究内容 .....	2
第二章 关键技术介绍 .....	3
2.1 相关技术介绍 .....	3
2.1.1 Tomcat 平台 .....	3
2.1.2 SSH 框架 .....	4
2.1.3 Mysql 功能简介 .....	4
2.1.4 Bootstrap 功能简介 .....	5
2.1.5 EasyUI 功能简介 .....	5
2.1.6 Jsp 功能简介 .....	5
2.2 原理介绍 .....	7
2.2.1 B/S 架构 .....	7
2.2.2 SSH 框架 .....	7
2.3 系统开发环境 .....	10
第三章 需求分析 .....	11
3.1 系统功能描述 .....	11
3.2 系统用例图 .....	11
3.2.1 用户管理 .....	11
3.2.2 角色管理 .....	16
3.2.3 权限管理 .....	21
3.2.4 新闻管理 .....	25
3.2.5 一级导航 .....	26
3.2.6 二级导航 .....	26
第四章 系统概要设计 .....	27
4.1 系统结构的设计 .....	27

4.1.1 设计思想 .....	27
4.1.2 设计原则 .....	27
4.1.3 系统结构设计 .....	27
4.2 数据库设计 .....	28
4.2.1 设计原则 .....	28
4.2.2 逻辑设计 .....	28
4.3 系统功能设计 .....	30
4.3.1 用户登录 .....	31
4.3.2 退出系统 .....	31
4.3.3 用户管理 .....	31
4.3.4 角色管理 .....	34
4.3.5 新闻管理 .....	36
4.3.6 一级导航管理 .....	36
4.3.7 二级导航管理 .....	37
第五章 系统详细设计与实现 .....	38
5.1 后台登录模块 .....	38
5.1.1 用户管理界面 .....	38
5.1.2 关键代码实现 .....	38
5.2 用户管理模块 .....	39
5.2.1 用户管理界面 .....	39
5.2.2 关键代码实现 .....	39
5.3 角色管理模块 .....	41
5.3.1 角色管理界面 .....	41
5.3.2 关键代码实现 .....	41
5.4 权限管理模块 .....	42
5.4.1 权限管理界面 .....	42
5.4.2 关键代码实现 .....	42
5.5 新闻管理模块 .....	43
5.6 一级导航管理模块 .....	43

5.7 二级导航管理模块 .....	44
第六章 系统测试 .....	45
6.1 测试概述 .....	45
6.1.1 测试重要性 .....	45
6.1.2 测试目的 .....	46
6.1.3 测试步骤 .....	46
6.1.4 测试用例 .....	47
6.2 本章小结 .....	47
第七章 结论 .....	48
参考文献 .....	49
致 谢 .....	50

## 第一章 绪论

### 1.1 研究背景和意义

随着高等院校的办学规模日益扩大化，生源也相应的日益增多，学院各种信息的汇总和发布等工作面临着诸多困难，而高校学院日常信息的发布尤其重要，因此学院信息管理的信息化、网络化越来越受到人们的重视。

互联网作为一种影响力日益扩大的媒体形式，对教育行业带来了极大的便利和深远的影响，学院信息管理的自动化已成为必然趋势。为提升学院信息交流的网络环境，迫切需要开发一个信息交流平台，以此来实现校内信息共享。信息交流平台的出现势必能够有效的改善学院信息管理的校绩，提高了办公效率，同时又可以为学生提供了一个自由化的交流平台，对推动交流信息化发展具有重要的意义。通过信息交流平台，学生可以随时了解校园新闻动态，老师也可以随时发布通知公告，同时又给予校外人员一个了解、认识学院的平台<sup>[1]</sup>。

### 1.2 国内外研究现状

学院信息交流平台是学校数字化校园建设管理的辅助系统，而在信息交流平台中重要的一部分就是新闻公告的发布，营造一个资源共享，安全可靠的信息管理环境<sup>[2]</sup>。

在国外，学院网站的发展大致分为两个不同的阶段：一个是以单机以及独立的业务系统为主的学院网站，另一种则是以局域网和数据库为基础的基于网络的学院网站发展阶段。新媒体技术经历了十几年甚至更多年的发展，他们的核心功能基本上已经能够适应社会的飞速发展和人们更高的要求。另外，他们会把客户的实际需求作为最基本的考虑因素，尽量实现易操作高集成化的系统。从新闻的采集工作开始新闻的提交，审核，修改，发布等各个环节都实现无缝对接。网络新闻会涉及到多领域的实时动态，也可以完成和用户的交互，还可以将传统媒体的文字转化为声音、图像、视频，使用户一目了然。美国的 **BuzzFeed** 通过网络从大量新闻博客中搜索新闻，为用户提供最新热门事件。

在国内，很多高校已经开放和使用了学院网站系统，但由于管理模式不同，以及各学校自身的特点和具体情况，所以在管理方法和方式上不尽相同，因此可重用性不高，这个问题有待解决。过去的时候，大部分网站采用静态 **HTML** 页面，管理人员必须手动制作新的网页，上传新的网页之后还需要修改链接；对于用户来说，管理人员发布新的

网页时，用户无法访问网站。随着中国经济实力和科技实力的增强，新闻发布管理系统的技术在不断进步，现在的技术已经可以解决这种问题了。

### 1.3 论文主要研究内容

本课题主要以高校学院信息交流平台开发的全过程为主脉，结合 UML 中的各种模型图及文字描述，详细介绍了本学院信息交流平台的需求、设计、实现。

本论文的内容安排如下：

第一章引言，其中介绍了研究的背景和意义，一级国内外研究现状。

第二章系统开发环境及开发工具介绍，其中包括对 B/S 结构、SSH 框架、Bootstrap 框架以及 EasyUI 框架的简单介绍。

第三章需求分析，其中包括系统功能描述，以及系统各个模块的用例图。

第四章系统设计，其中包括系统结构设计、数据库设计、系统功能设计等。

第五章系统实现，主要展示了系统各个模块的界面设计以及关键实现代码。

第六章系统测试，其中包括测试的重要性、测试的目的以及测试的步骤等。

第七章总结，对在写整篇论文过程中的感受进行总结。



## 第二章 关键技术介绍

### 2.1 相关技术介绍

本系统是在 SSH 框架，同时结合 Bootstrap、Easyui 前端框架，并使用 Mysql 数据库进行开发的。

#### 2.1.1 Tomcat 平台

Tomcat 服务器是一个免费的开放源代码的 WEB 应用服务器，是 Apache 软件基金会的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些学校及个人共同开发而成。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现，Tomcat7 支持最新的 Servlet 2.4 和 JSP 2.0 规范。因为 Tomcat 技术先进、性能稳定，而且免费，因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的 WEB 应用服务器<sup>[10]</sup>。

Tomcat 很受广大程序员的喜欢，因为它运行时占用的系统资源小，扩展性好，支持负载均衡与邮件服务等开发应用系统常用的功能；而且它还在不断的改进和完善中，任何一个感兴趣的程序员都可以更改它或在其中加入新的功能。

Tomcat 是一个小型的轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好 Apache 服务器，可利用它响应对 HTML 页面的访问请求。实际上 Tomcat 部分是 Apache 服务器的扩展，但它是独立运行的，所以当你运行 Tomcat 时，它实际上作为一个与 Apache 独立的进程单独运行的。

自从 JSP 发布之后，推出了各式各样的 JSP 引擎。Apache Group 在完成 GNUJSP1.0 的开发以后，开始考虑在 Sun 的 JSDK 基础上开发一个可以直接提供 WEB 服务的 JSP 服务器，当然同时也支持 Servlet，这样 Tomcat 就诞生了。Tomcat 是 Jakarta 项目中的一个重要的子项目，其被 JavaWorld 杂志的编辑选为 2001 年度最具创新的 Java 产品，同时它又是 Sun 公司官方推荐的 Servlet 和 JSP 容器，因此其越来越受到软件公司和开发人员的喜爱。Servlet 和 JSP 的最新规范都可以在 Tomcat 的新版本中得到实现。其次，Tomcat 也是完全免费的软件，任何人都可以从互联网上自由地下载。

### 2.1.2 SSH 框架

集成 SSH 框架的系统从职责上分为四层：表示层、业务逻辑层、数据持久层和域模块层，以帮助开发人员在短期内搭建结构清晰、可复用性好、维护方便的 Web 应用程序。其中使用 Struts 作为系统的整体基础架构，负责 MVC 的分离，在 Struts 框架的模型部分，控制业务跳转，利用 Hibernate 框架对持久层提供支持，Spring 做管理，管理 struts 和 hibernate。具体做法是：用面向对象的分析方法根据需求提出一些模型，将这些模型实现为基本的 Java 对象，然后编写基本的 DAO(Data Access Objects)接口，并给出 Hibernate 的 DAO 实现，采用 Hibernate 架构实现的 DAO 类来实现 Java 类与数据库之间的转换和访问，最后由 Spring 做管理，管理 Struts 和 Hibernate<sup>[3]</sup>。

### 2.1.3 Mysql 功能简介

Mysql 是一个小型关系型数据库管理系统，开发者为瑞典 MysqlAB 公司，在 2008 年 1 月 16 号被 Sun 公司收购。Mysql 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择 Mysql 作为网站数据库。如雅虎、阿尔卡特—朗讯、Google 公司、诺基亚公司、百度等公司就采用了 Mysql 数据库。Mysql 数据库可以称得上是目前运行速度最快的 SQL 语言数据库。除了具有许多其它数据库所不具备的功能和选择之外 Mysql 数据库是一种完全免费的产品，用户可以直接从网上下载。

Mysql 数据库主要有以下特点：

（1）可移植性：

使用 C 和 C++编写，并使用了多种编译器进行测试，保证源代码的可移植性。

（2）可扩展性和灵活性：

Mysql 可以支持 UNIX、Linux 和 Windows 等多种操作系统平台。在一个操作系统中实现的应用可以很方便地移植到其他操作系统。Mysql 作为开源性质的数据库服务器，可以为那些想要增加独特需求的用户提供完全定制的功能。

（3）强大的数据保护功能：

Mysql 有一个非常灵活且安全的权限和密码系统，为确保只有获授权用户才能进入该数据库服务器，所有的密码传输均采用加密形式，同时也提供 SSH 和 SSI 支持，以实现安全和可靠的连接。Mysql 强大的数据加密和解密功能，可以保证敏感数据不受未经授权的访问。

（4）支持大型的数据库：

虽然对于编写的网页来说，只要能够存放数百条以上的记录数据就是够了，但 Mysql 可以方便地支持上千万条记录的数据库。作为一个开放源代码的数据库，Mysql 可以针对不同的应用进行相应的修改。

（5）超强的稳定性：

Mysql 拥有快速、稳定且基于线程的内存分配系统，可以持续使用而不必担心其稳定性。线程是轻量级的进程，它可以灵活地为用户提供服务，而不占用过多的系统资源。用多线程和 C 语言实现的 Mysql 能很容易地充分利用 CPU。

（6）强大的查询功能：

Mysql 支持查询的 select 和 where 语句的全部运算符和函数，并且可以在同一查询中混用来自不同数据库的表，从而使得查询变得快捷、方便。

#### 2.1.4 Bootstrap 功能简介

Bootstrap，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。它由 Twitter 的设计师 Mark Otto 和 Jacob Thornton 合作开发，是一个 CSS/HTML 框架。Bootstrap 提供了优雅的 HTML 和 CSS 规范，它即是由动态 CSS 语言 Less 写成。Bootstrap 一经推出后颇受欢迎，一直是 GitHub 上的热门开源项目，包括 NASA 的 MSNBC（微软全国广播公司）的 Breaking News 都使用了该项目。国内一些移动开发者较为熟悉的框架，如 WeX5 前端开源框架等，也是基于 Bootstrap 源码进行性能优化而来。

#### 2.1.5 EasyUI 功能简介

jQuery EasyUI 是一组基于 jQuery 的 UI 插件集合体，而 jQuery EasyUI 的目标就是帮助 WEB 开发者更轻松的打造出功能丰富并且美观的 UI 界面。开发者不需要编写复杂的 JavaScript，也不需要了解 CSS 样式有深入的了解，开发者需要了解的只有一些简单的 HTML 标签。

#### 2.1.6 Jsp 功能简介

JSP(Java Server Pages) 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准。JSP 技术有点类似 ASP 技术，它是在传统的网页 HTML 文件中插入 Java 程序段和 JSP 标记，从而形成 JSP 文件。用 JSP 开发的 WEB 应用是跨平台的，即能在 Linux 下运行，也能在其他操作系统上运行<sup>[9]</sup>。

JSP 技术使用 Java 编程语言编写类 XML 的 tags 和 scriptlets, 来封装产生动态网页的处理逻辑。网页还能通过 tags 和 scriptlets 访问存在于服务端的资源的应用逻辑。JSP 将网页逻辑与网页设计和显示分离, 支持可重用的基于组件的设计, 使基于 WEB 的应用程序的开发变得迅速和容易。WEB 服务器在遇到访问 JSP 网页的请求时, 首先执行其中的程序段, 然后将执行结果连同 JSP 文件中的 HTML 代码一起返回给客户。插入的 Java 程序段可以操作数据库、重新定向网页等, 以实现建立动态网页所需要的功能。JSP 与 Java Servlet 一样, 是在服务器端执行的, 通常返回该客户端的就是一个 HTML 文本, 因此客户端只要有浏览器就能浏览。

JSP 页面由 HTML 代码和嵌入其中的 Java 代码所组成。Java Servlet 是 JSP 的技术基础, 而且大型的 WEB 应用程序的开发需要 Java Servlet 和 JSP 配合才能完成。JSP 具备了 Java 技术的简单易用, 完全面向对象, 具有平台无关性且安全可靠, 主要面向因特网的所有特点。

#### JSP 技术的强势:

(1) 一次编写, 到处运行。在这一点上 Java 比 PHP 更出色, 除了系统之外, 代码不用做任何更改。

(2) 系统的多平台支持。基本上可以在所有平台上的任意环境中开发、系统部署、扩展。相比 ASP/PHP 的局限性是显而易见的。

(3) 强大的可伸缩性。从只有一个小的 Jar 文件就可以运行 Servlet/JSP, 到由多台服务器进行集群和负载均衡, 到多台 Application 进行事务处理、消息处理, 一台服务器到无数台服务器, Java 显示了一个巨大的生命力。

(4) 多样化和功能强大的开发工具支持。这一点与 ASP 很像, Java 已经有了许多非常优秀的开发工具, 而且许多可以免费得到, 并且其中许多已经可以顺利的运行于多种平台之下。

#### JSP 技术的弱势:

(1) 与 ASP 一样, Java 的一些优势正是它致命的问题所在。正是由于为了跨平台的功能, 为了极度的伸缩能力, 所以极大的增加了产品的复杂性。

(2) Java 的运行速度是用 class 常驻内存来完成的, 所以它在一些情况下所使用的内存比起用户数量来说确实是“最低性能价格比”了。从另一方面, 它还需要硬盘空间来储存一系列的.java 文件和.class 文件, 以及对应的版本文件。

## 2.2 原理介绍

### 2.2.1 B/S 架构

目前，网络应用软件运行的模式主要有两类：客户机/服务器(C/S)模式和浏览器/服务器(B/S)模式<sup>[13]</sup>。C/S 模式是将数据库(如试题库)内容放在远程的服务器上，在客户机上安装相应软件。C/S 结构在技术上虽然很成熟，但该结构存在灵活性差、升级困难、维护工作量大和系统安全性差等缺陷，已较难适应当前信息技术与网络技术发展的需要。学院信息交流平台采用 B/S 结构，由 WEB 浏览器、WEB 服务器、数据库服务器三层分布结构组成。由于采用了互联网的相关技术，B/S 结构具有系统维护升级方便、跨平台操作、具有良好的开放性和可扩充性以及系统安全性等优点<sup>[7]</sup>。

用户可通过 Internet 利用浏览器登录到 WEB 服务器上，请求调用 JSP 文件，打开该 JSP 页面，执行相应的脚本，执行结果产生动态生成的 HTML 页面并返回浏览器，进行应答<sup>[14]</sup>。用户通过浏览器对 HTML 页面进行相应的操作请求，则 JSP 通过 JDBC 访问数据库服务器，对试题库进行访问查询，然后动态生成一个 HTML 页面将查询结果返回浏览器，供用户使用<sup>[15]</sup>。其总体方案见图 2-1 所示。

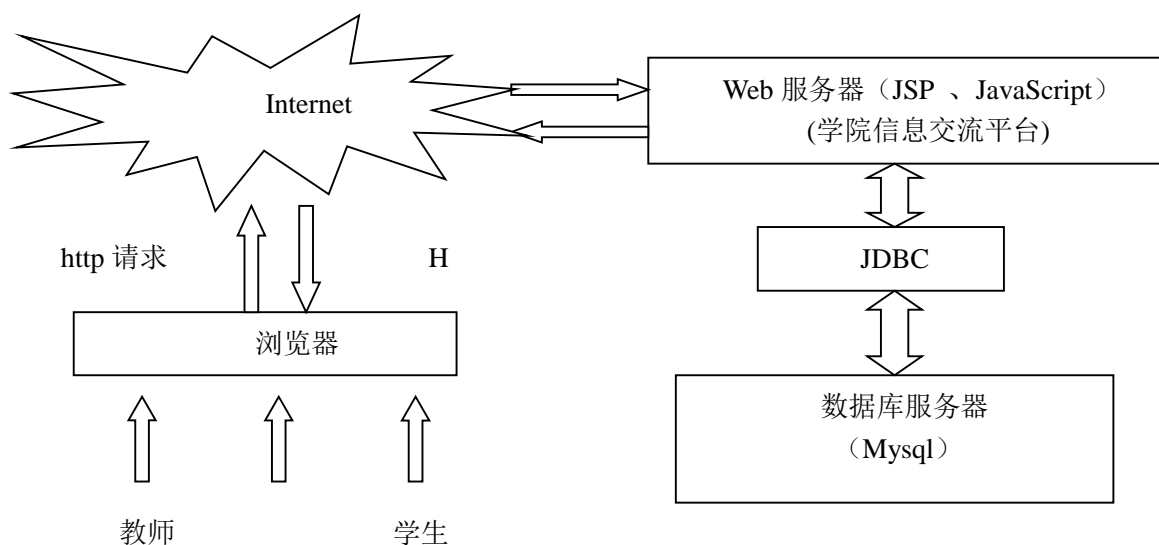


图 2-1 基于 B/S 模式的信息交流平台系统总体方案

### 2.2.2 SSH 框架

#### 1. Spring 框架

内部最核心的就是 IOC 了，动态注入，让一个对象的创建不用 New 了，可以自动的生产，这其实就是利用 Java 里的反射，反射其实就是在运行时动态的去创建、调用对象，Spring 就是在运行时，跟 Xml Spring 的配置文件来动态的创建对象，和调用对

象里的方法的<sup>[4]</sup>。

Spring 还有一个核心就是 AOP 这个就是面向切面编程，可以为某一类对象进行监督和控制（也就是在调用这类对象的具体方法的前后去调用你指定的模块）从而达到对一个模块扩充的功能。这些都是通过配置类达到的。

Spring 框架总体架构图如下图 2-2 所示。

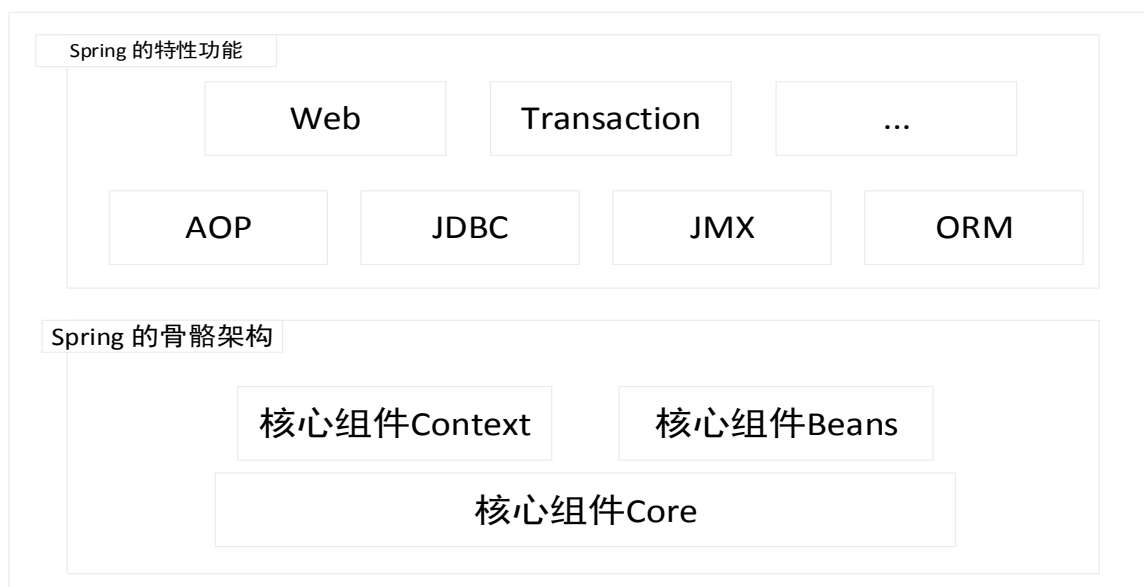


图 2-2 Spring 框架总体架构图

## 2. Hibernate 框架

Spring 框架总体架构图如下图 2-3 所示。

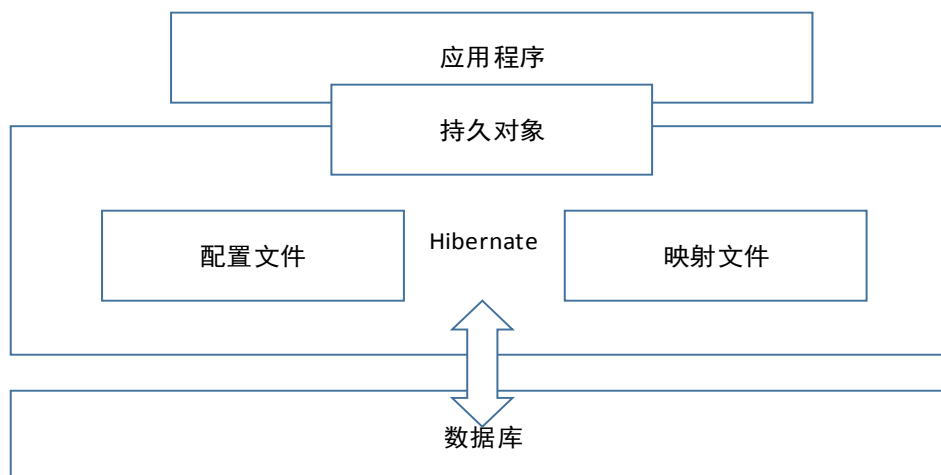


图 2-3 Hibernate 框架体系架构图

在基于 MVC 设计模式的 JAVA WEB 应用中，Hibernate 可以作为模型层/数据访问层。它通过配置文件(hibernate.properties 或 hibernate.cfg.xml)和映射文件(\*.hbm.xml)把

Java 对象或 PO(Persistent Object,持久化对象)映射到数据库中的数据库，然后通过操作 PO，对数据表中的数据进行增，删，改，查等操作<sup>[6]</sup>。

除配置文件，映射文件和持久化类外，Hibernate 的核心组件包括以下几部分：

- （1）Configuration 类：用来读取 Hibernate 配置文件，并生成 SessionFactory 对象。
- （2）SessionFactory 接口：SessionFactory 负责 Session 实例的创建。可以通过 Configuration 实例创建。
- （3）Session 接口：用来操作 PO。它有 get()、load()、save()、update()和 delete()等方法用来对 PO 进行加载、保存、更新及删除等操作。它是 Hibernate 的核心接口。
- （4）Query 接口：用来对 PO 进行查询操。它可以从 Session 的 createQuery()方法生成。
- （5）Transaction 接口：用来管理 Hibernate 事务，它主要方法有 commit()和 rollback()，可以从 Session 的 beginTrancation()方法生成。

### 3. Struts2 框架

在 Struts2 框架中的处理大概分为以下几个步骤<sup>[5]</sup>：

- （1）客户端初始化一个指向 Servlet 容器（例如 Tomcat）的请求。
- （2）这个请求经过一系列的过滤器（Filter）（这些过滤器中有一个叫做 ActionContextCleanUp 的可选过滤器，这个过滤器对于 Struts2 和其他框架的集成很有帮助，例如：SiteMesh Plugin）。
- （3）接着 FilterDispatcher 被调用，FilterDispatcher 询问 ActionMapper 来决定这个请是否需要调用某个 Action。
- （4）如果 ActionMapper 决定需要调用某个 Action，FilterDispatcher 把请求的处理交给 ActionProxy。
- （5）ActionProxy 通过 Configuration Manager 询问框架的配置文件，找到需要调用的 Action 类。
- （6）ActionProxy 创建一个 ActionInvocation 的实例。
- （7）ActionInvocation 实例使用命名模式来调用，在调用 Action 的过程前后，涉及到相关拦截器（Interceptor）的调用。
- （8）一旦 Action 执行完毕，ActionInvocation 负责根据 struts.xml 中的配置找到对应的返回结果。返回结果通常是（但不总是，也可能是另外的一个 Action 链）一个需

要被表示的 JSP 或者 FreeMarker 的模版。在表示的过程中可以使用 Struts2 框架中继承的标签。在这个过程中需要涉及到 ActionMapper。

Struts2 框架体系架构图如下图 2-4 所示。

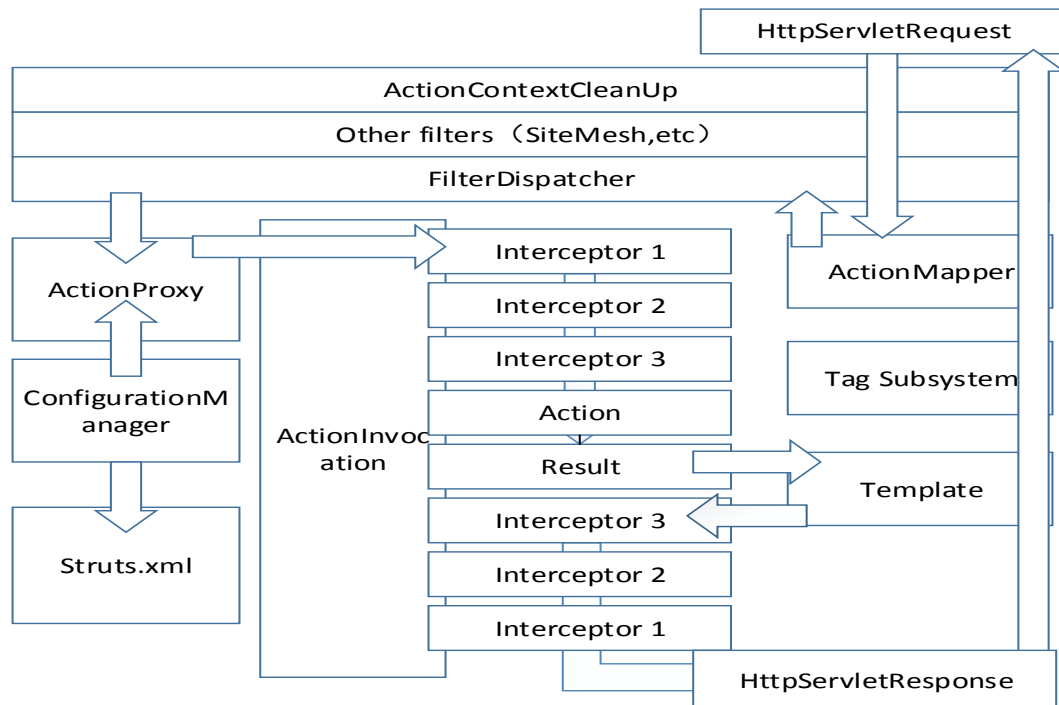


图 2-4 Struts2 框架体系架构图

## 2.3 系统开发环境

服务器操作系统：Windows 7/8

应用服务器：Tomcat 6/7

数据库：Mysql

开发工具：Myeclipse



## 第三章 需求分析

### 3.1 系统功能描述

本系统主要实现了权限管理、内容管理、导航管理、系统管理三大子模块，其中权限管理模块主要实现了用户管理、角色管理、菜单管理功能；内容管理模块主要实现了新闻管理功能；导航管理主要实现了一级导航管理及二级导航管理功能；系统管理主要实现了修改密码、安全退出功能。

### 3.2 系统用例图

用例图表述了各种外部执行者与系统所提供的用例之间的连接。该信息交流平台的总用例图如下 3-1 所示。

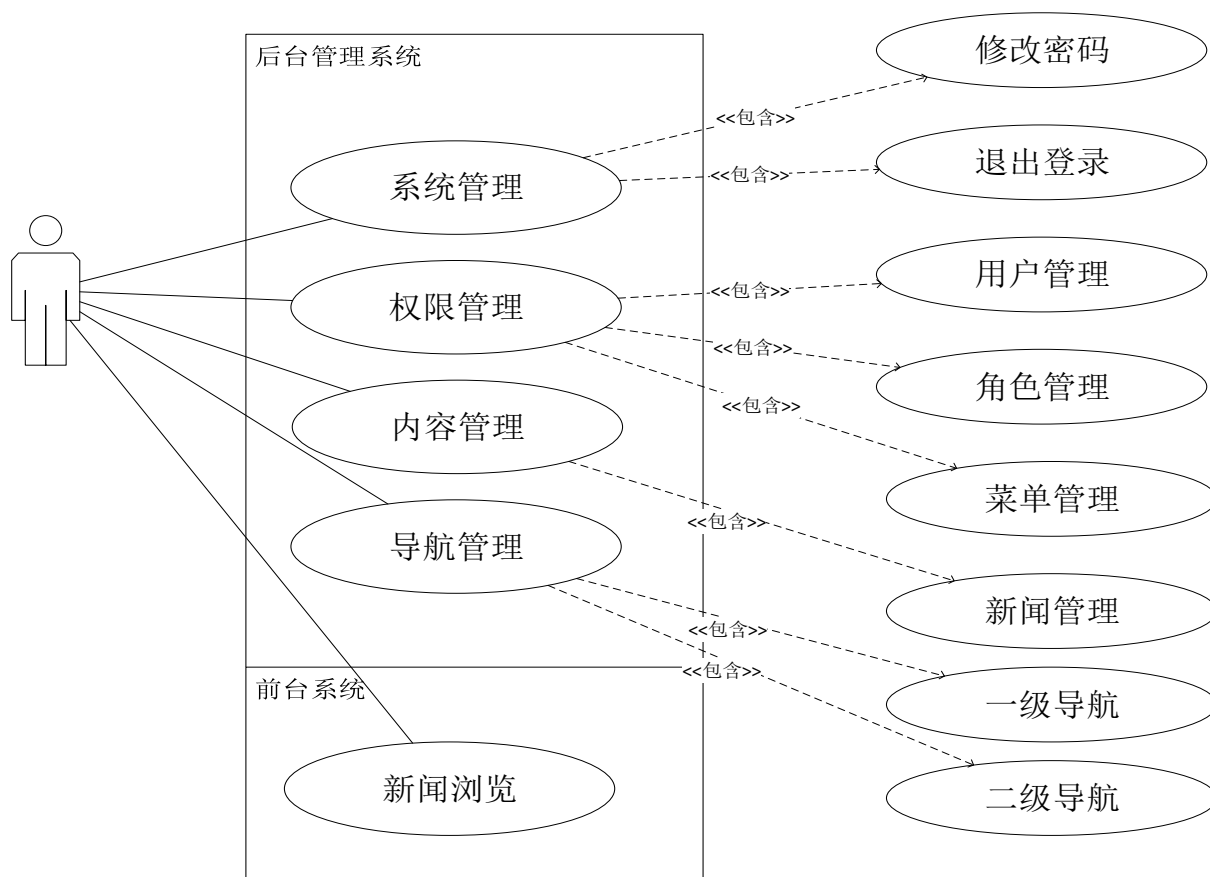


图 3-1 学院信息交流平台用例图

#### 3.2.1 用户管理

用户管理部分，包括对用户的增删该查。

用户管理用例图如下图 3-2 所示。

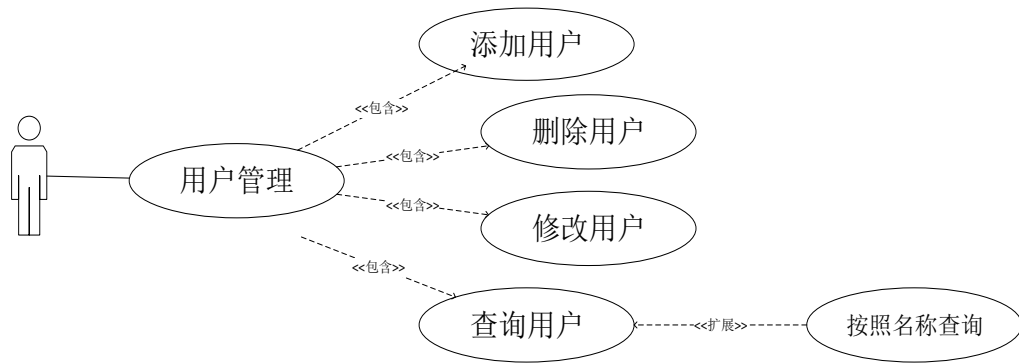


图 3-2 用户管理用例图

用户管理模块类图，如下图 3-3 所示。

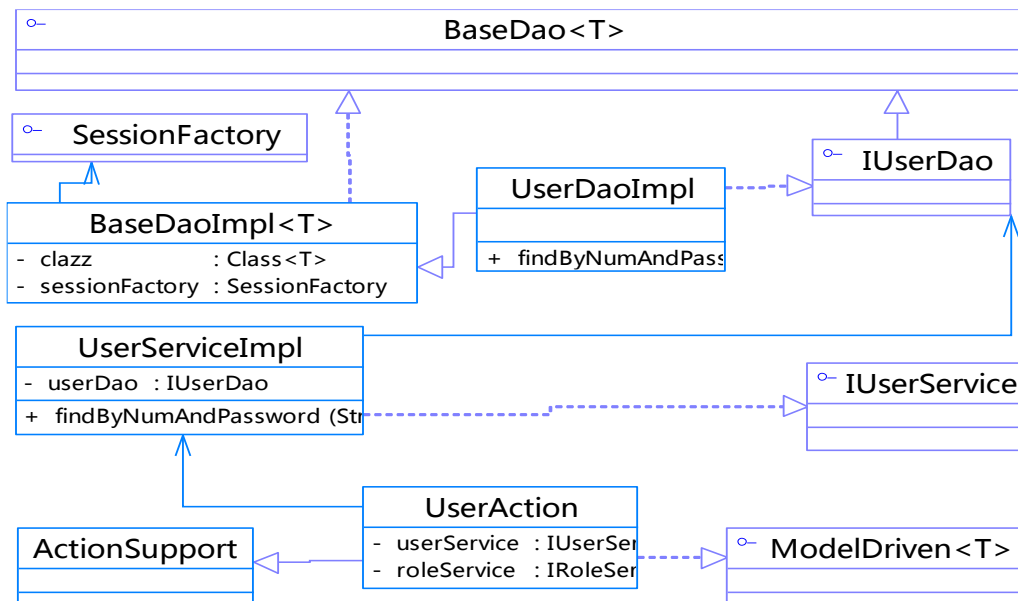


图 3-3 用户管理类图

以下将具体介绍用户管理的四大模块，分别为：添加用户模块、修改用户模块、删除用户模块、查询用户列表模块。

**时序图：**是一种 UML 交互图。它通过描述对象之间发送消息的时间顺序显示多喝对象之间的动作协作。它可以表示用例的行为顺序，当执行一个用例行为时，其中的每一条消息对应一个类操作或状态机中引起转换的触发事件。

**协作图：**是一种交互图，强调的是发送和接收消息的对象之间的组织结构。一个协作图显示了一系列的对象和在这些对象之间的联系以及对象间发送和接收的消息。

#### （1）添加用户模块

添加用户，并同时添加用户所拥有的角色。

添加用户时序图如下图 3-3 所示。

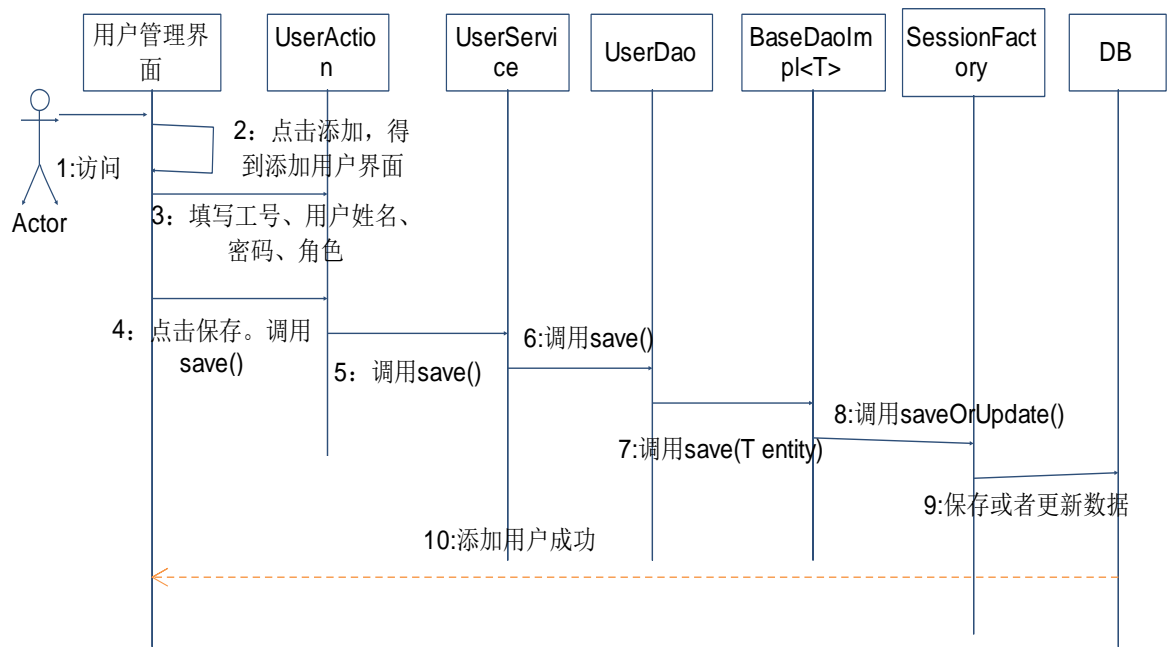


图 3-3 添加用户时序图

添加用户协作图如下图 3-4 所示。

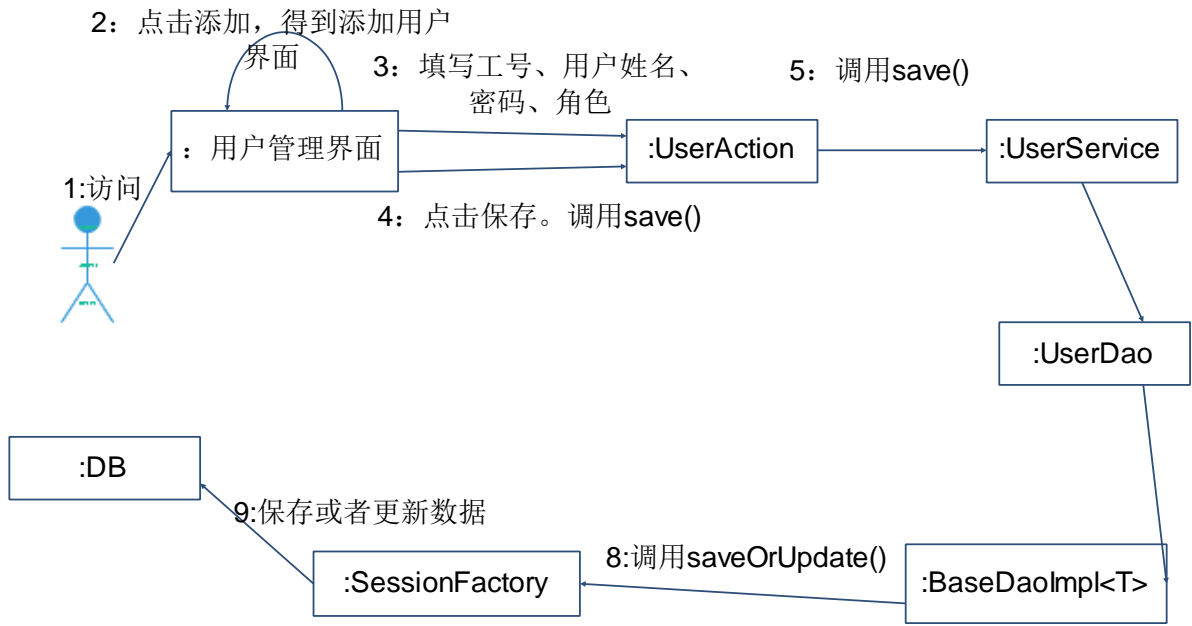


图 3-4 添加用户协作图

(2) 修改用户模块

修改用户时序图如下图 3-5 所示。

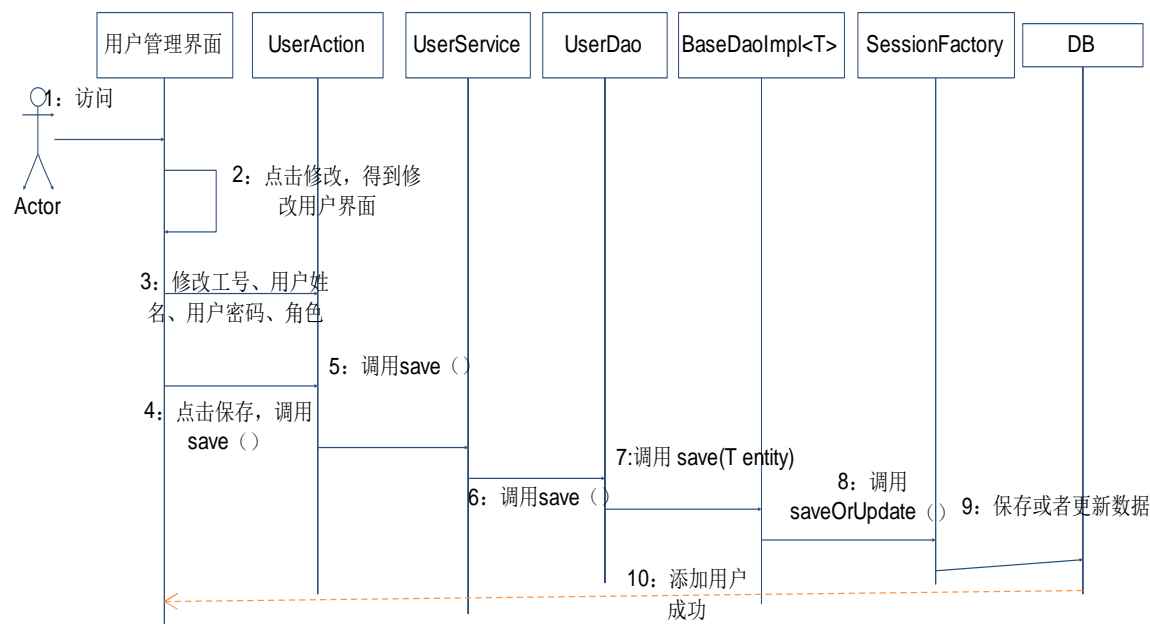


图 3-5 修改用户时序图

修改用户协作图如下图 3-6 所示。

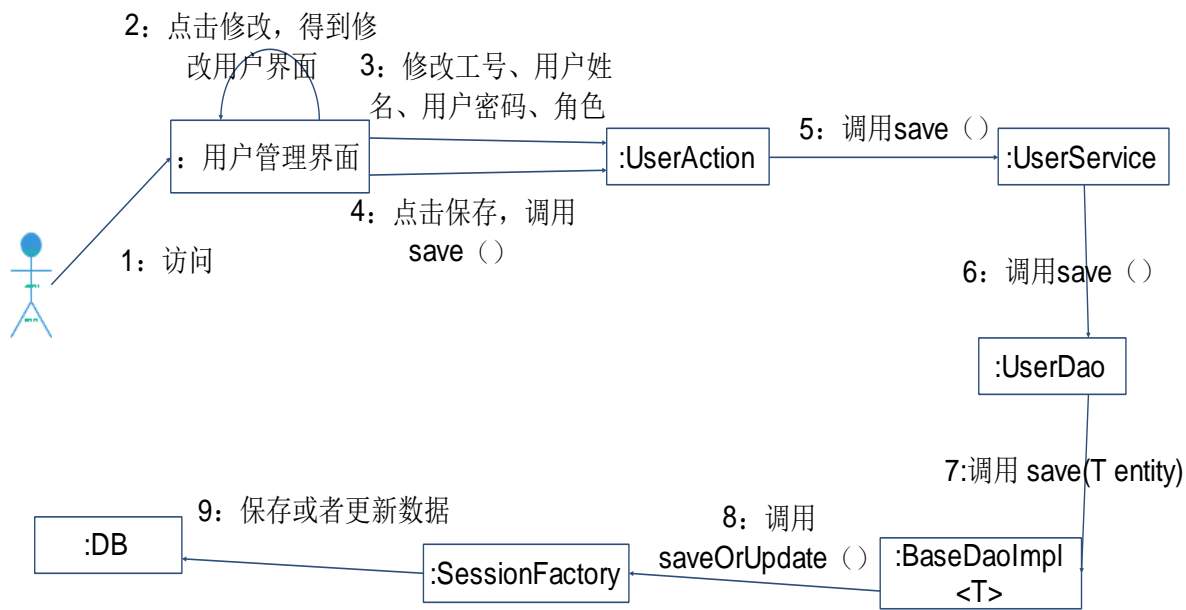


图 3-6 修改用户协作图

(3) 删除用户模块

删除用户时序图如下图 3-7 所示。

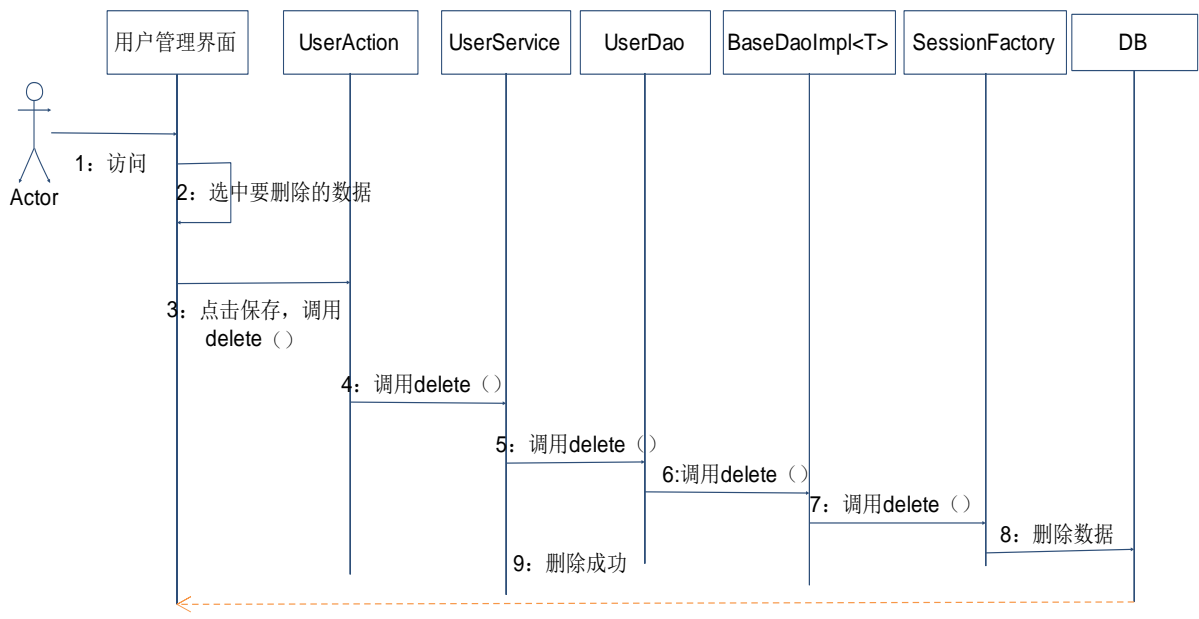


图 3-7 删除用户时序图

删除用户协作图如下图 3-8 所示。

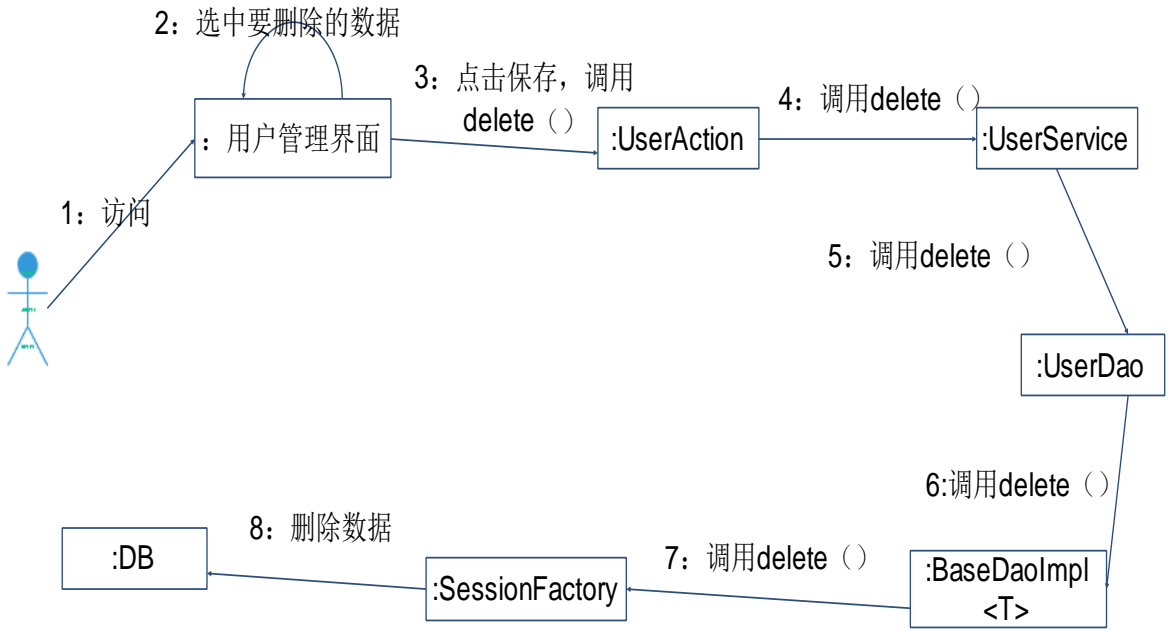


图 3-8 删除用户协作图

(4) 查看用户列表模块

查看用户列表时序图如下图 3-9 所示。

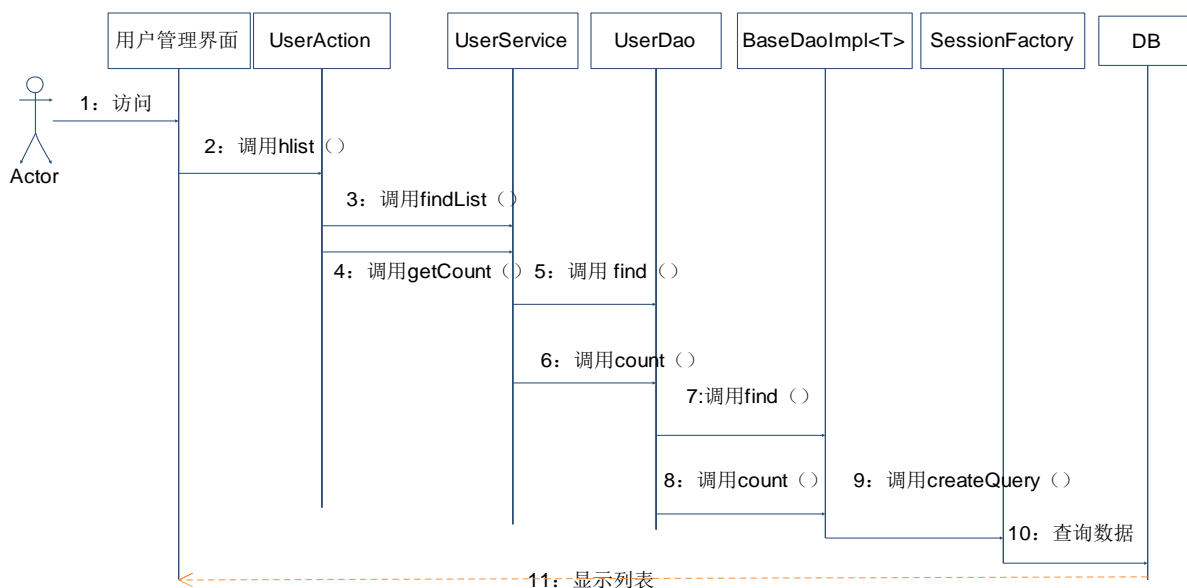


图 3-9 查看用户列表时序图

查看用户列表协作图如下图 3-10 所示。

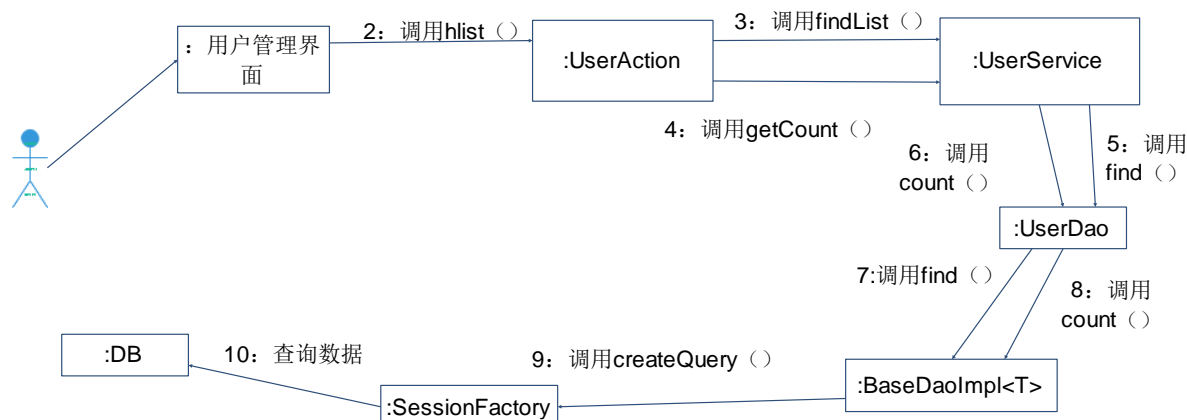


图 3-10 查看用户列表时序图

### 3.2.2 角色管理

角色管理部分，包括对角色的增删该查，其用例图如下图 3-11 所示。

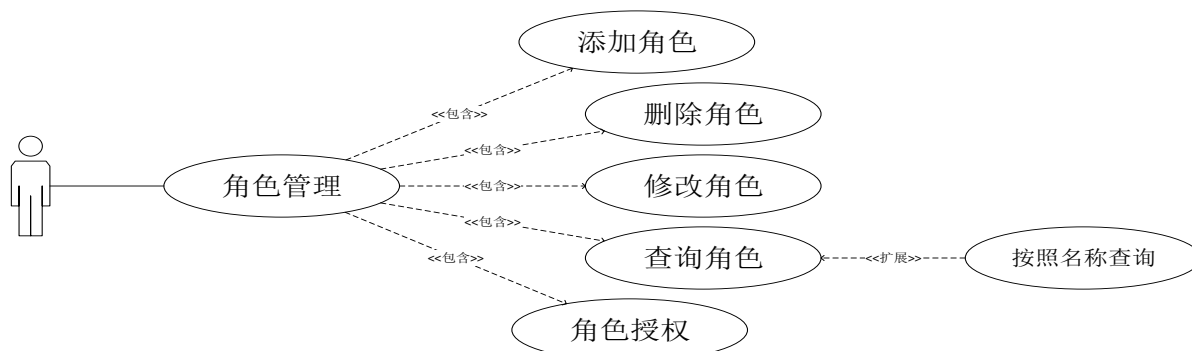


图 3-11 角色管理用例图

角色管理模块类图，如下图 3-12 所示。

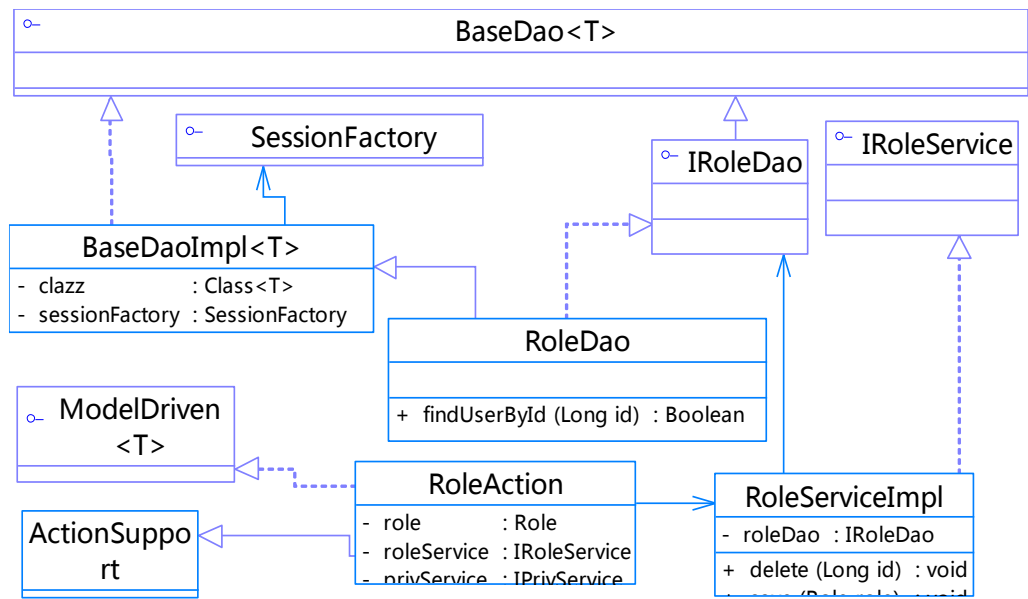


图 3-12 角色管理类图

以下将具体介绍角色管理的四大模块，分别为：添加角色模块、修改角色模块、删除角色模块，查询角色列表模块。

(1) 添加角色模块

添加角色时序图如下图 3-13 所示。

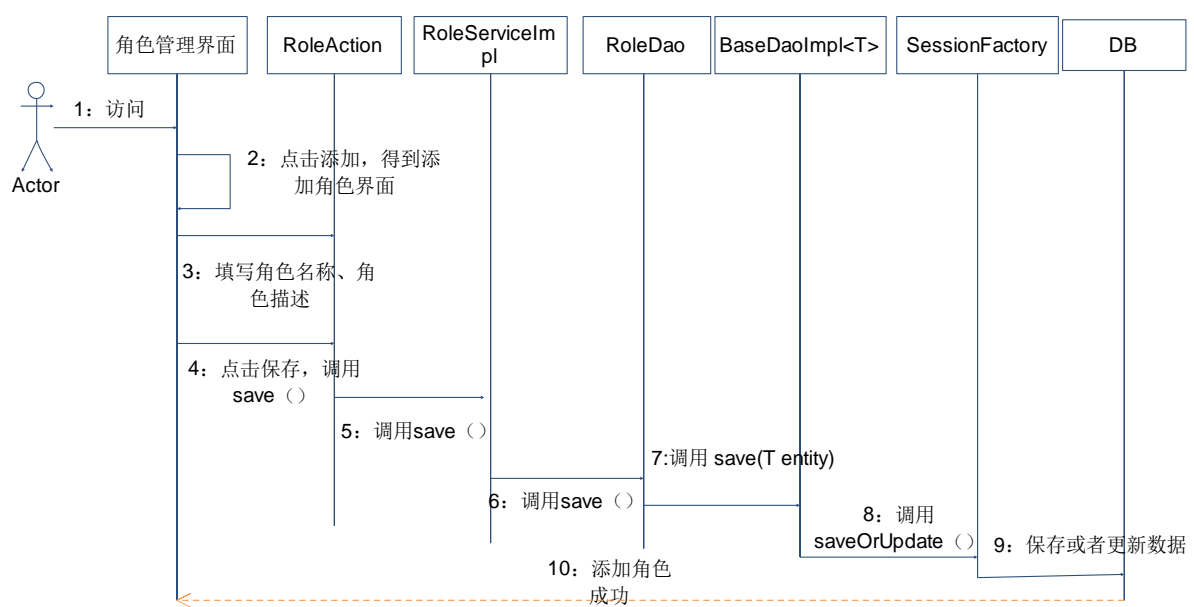


图 3-13 添加角色时序图

添加角色协作图如下图 3-14 所示。

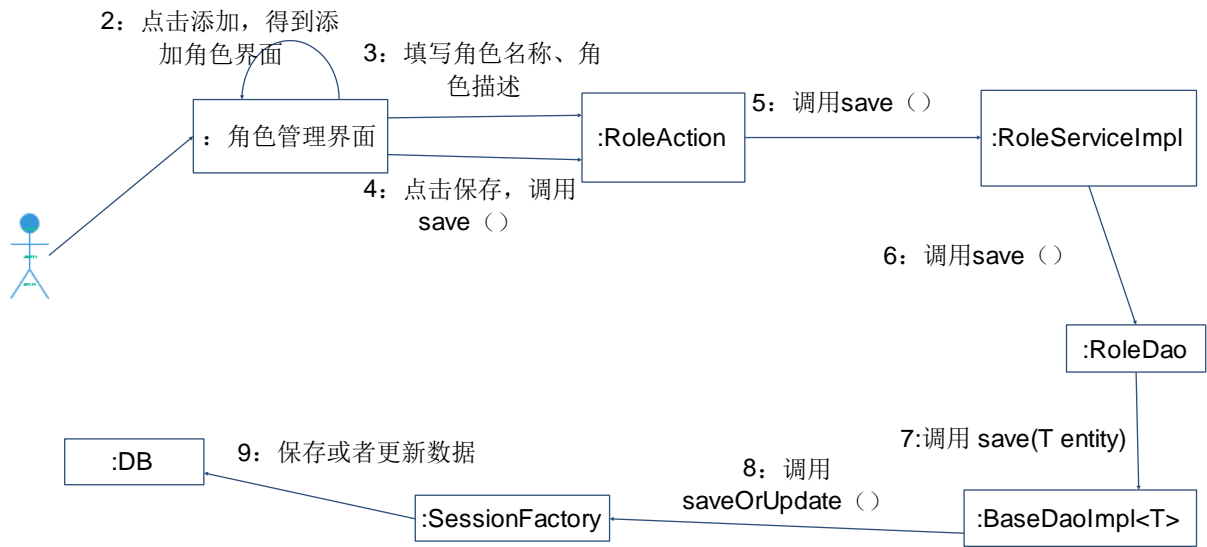


图 3-14 添加角色协作图

(2) 修改角色模块

修改角色时序图如下图 3-15 所示。

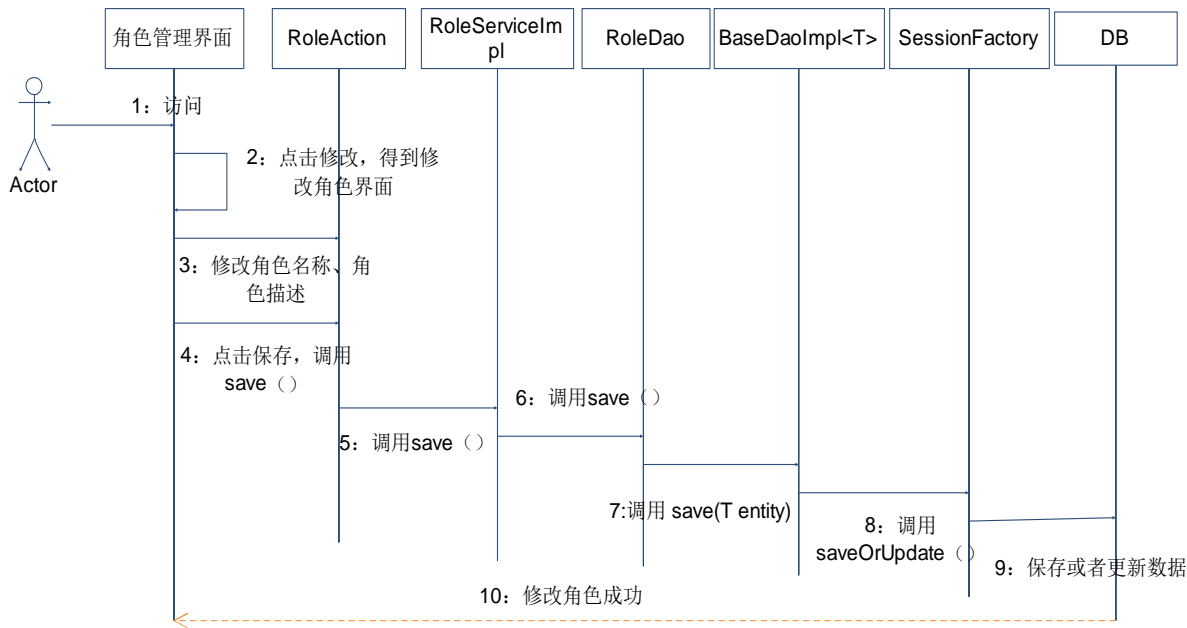


图 3-15 修改角色时序图

修改角色协作图如下图 3-16 所示。



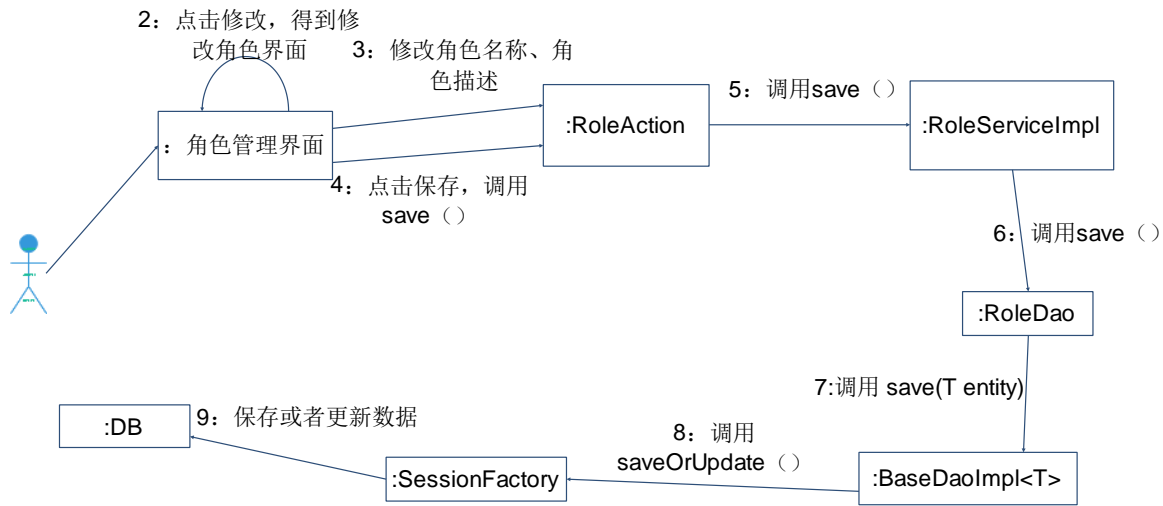


图 3-16 修改角色协作图

(3) 删除角色模块

删除角色时序图如下图 3-17 所示。

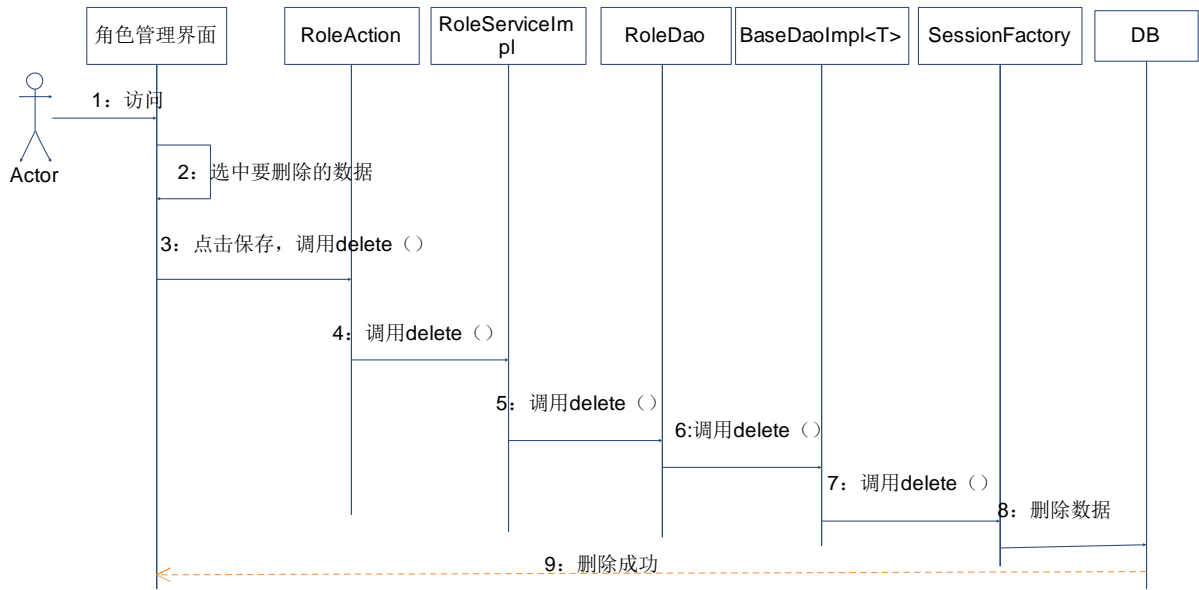


图 3-17 删除角色时序图

删除角色协作图如下图 3-18 所示。

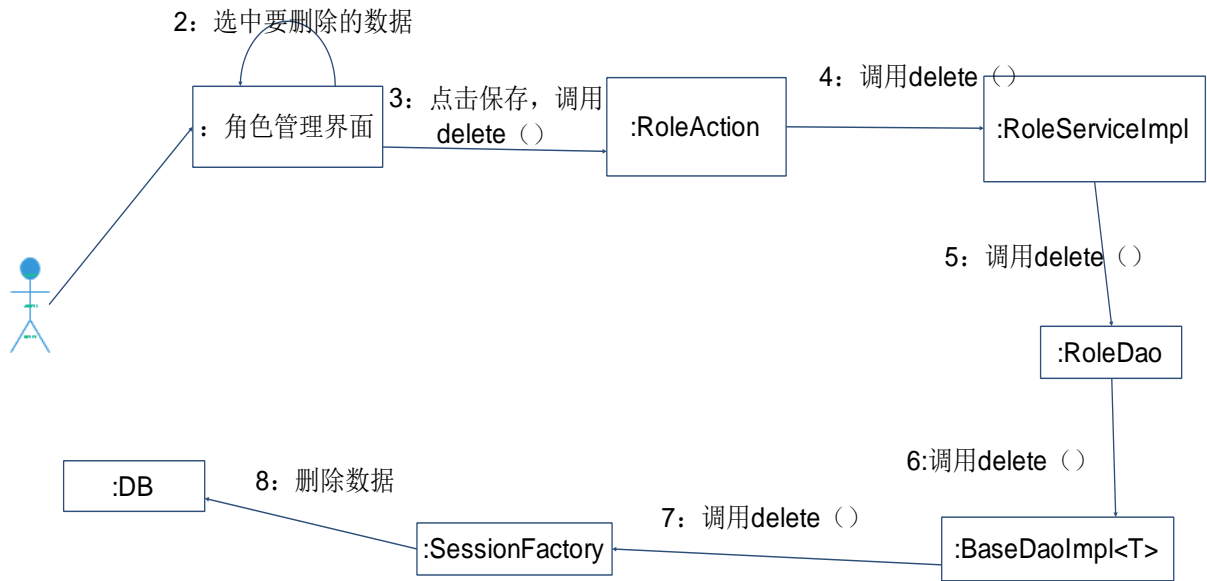


图 3-18 删除角色协作图

(4) 查看角色列表模块

查看角色列表时序图如下图 3-19 所示。

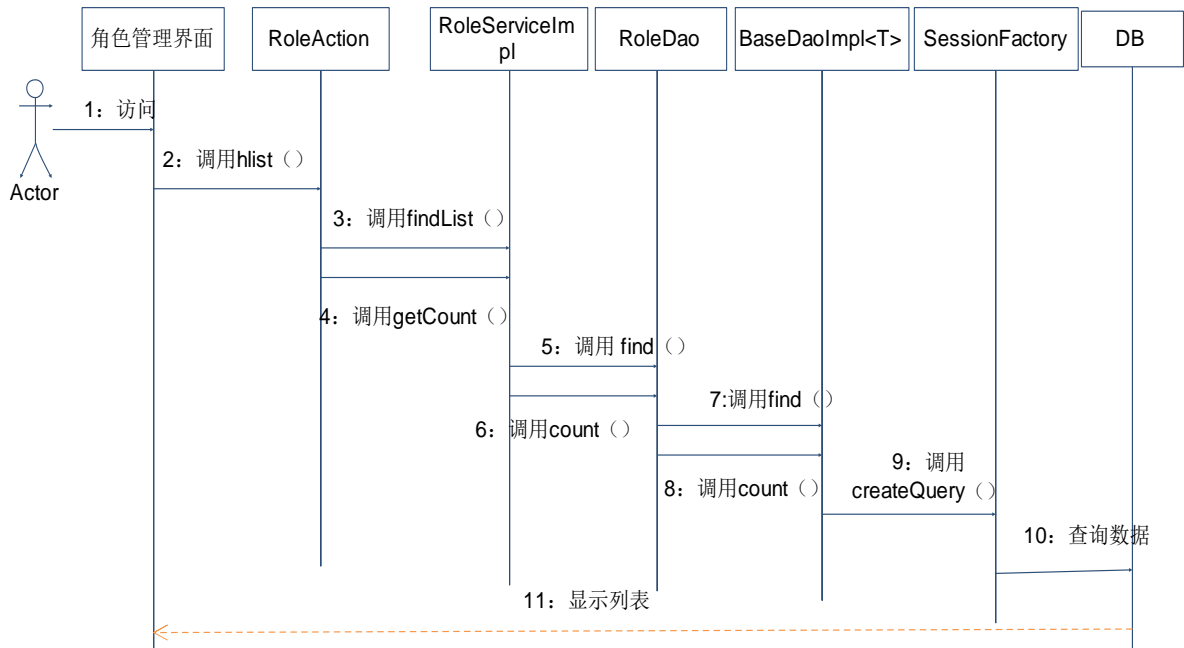


图 3-19 查看角色列表时序图

查看角色列表协作图如下图 3-20 所示。

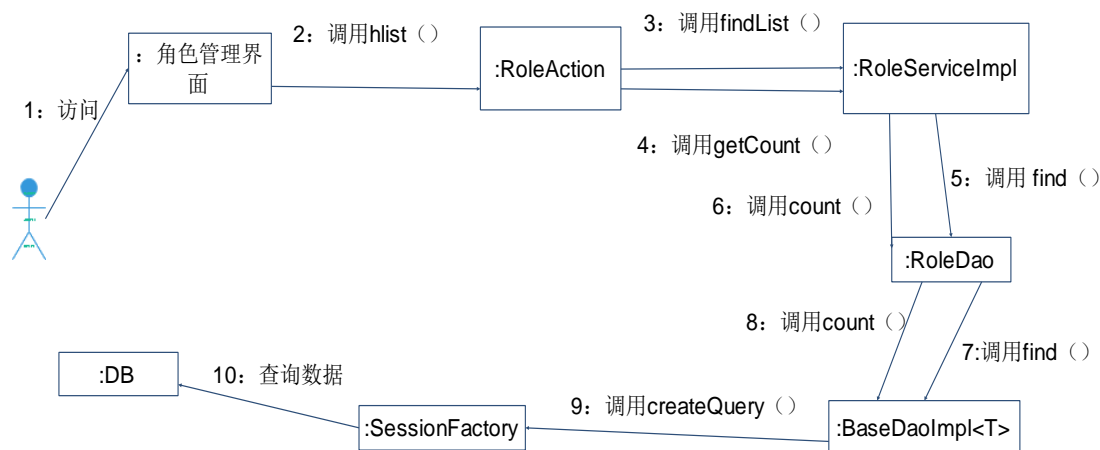


图 3-20 查看角色列表协作图

### 3.2.3 权限管理

权限管理部分，包括对权限增删该查，其用例图如下图 3-21 所示。

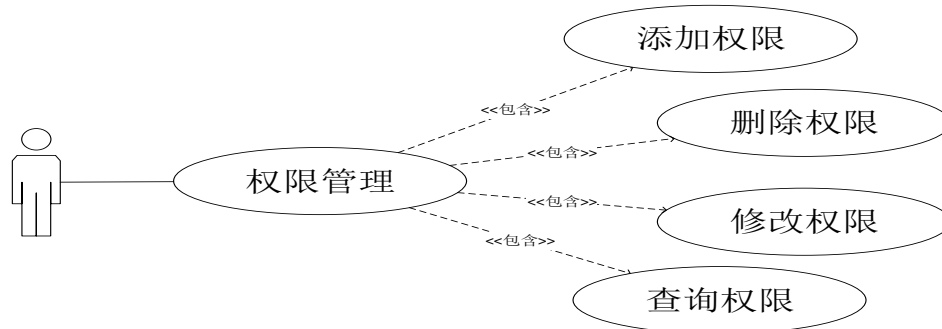


图 3-21 权限管理用例图

权限管理模块类图，如下图 3-22 所示。

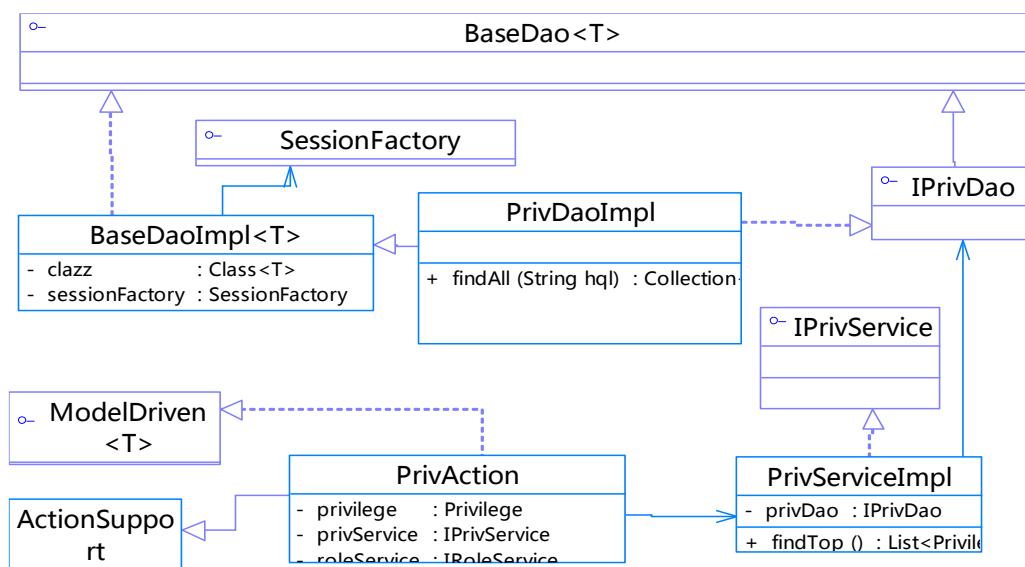


图 3-22 权限管理类图

以下将具体介绍角色管理的四大模块，分别为：添加角色模块、修改角色模块、删

除角色模块，查询角色列表模块。

（1）添加权限模块

添加权限管理模块时序图，如下图 3-23 所示。

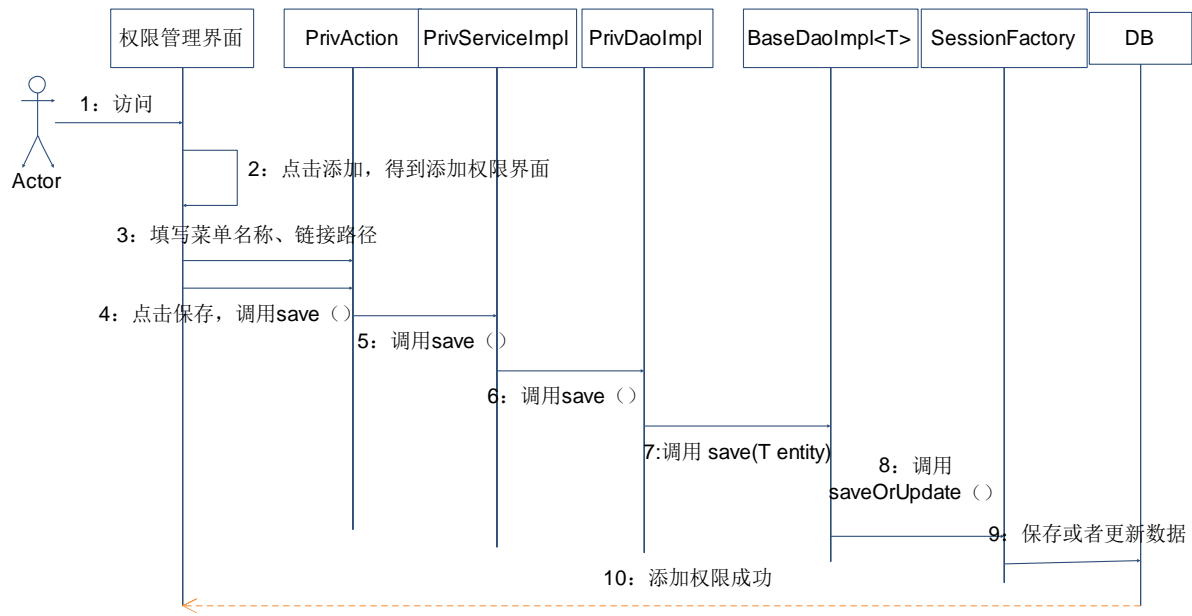


图 3-23 添加权限时序图

添加权限管理模块协作图，如下图 3-24 所示。

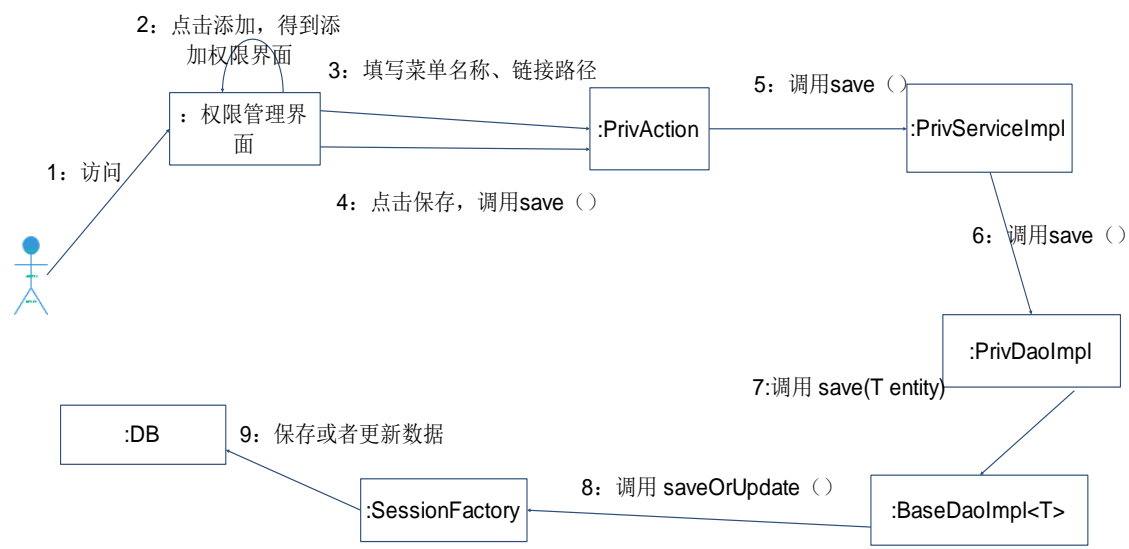


图 3-24 添加权限协作图

（2）修改权限模块

修改权限管理模块时序图，如下图 3-25 所示。

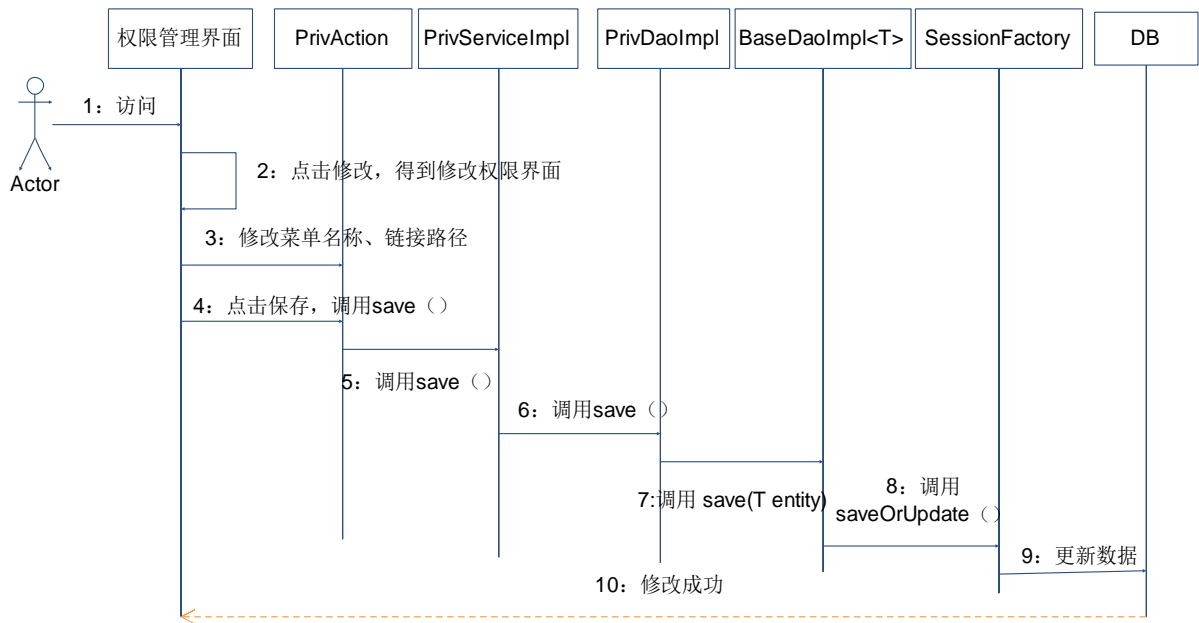


图 3-25 修改权限时序图

修改权限管理模块协作图，如下图 3-26 所示。

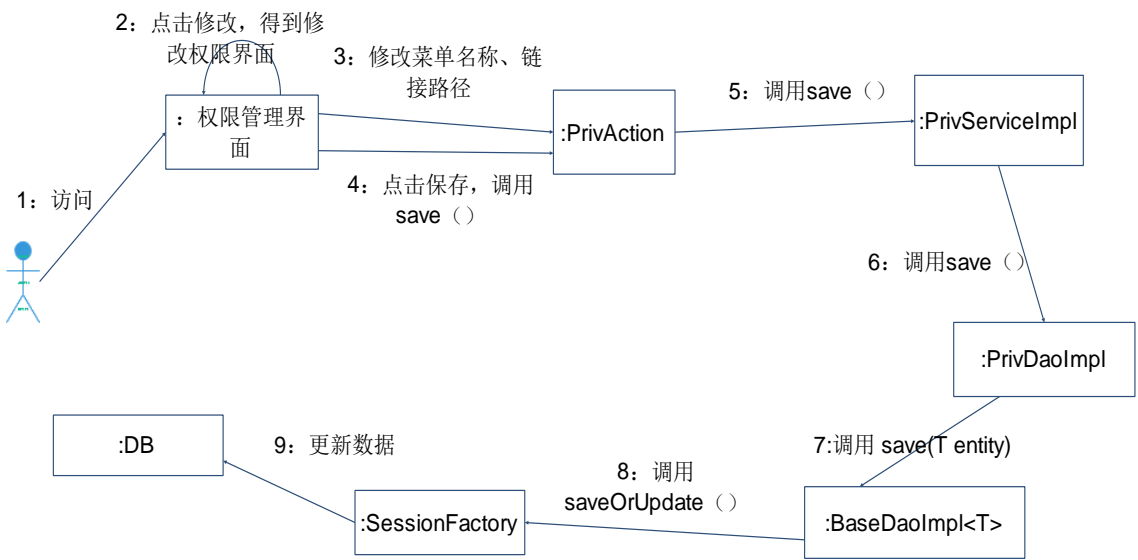


图 3-26 修改权限协作图

(3) 删除权限模块

删除权限管理模块时序图，如下图 3-27 所示。

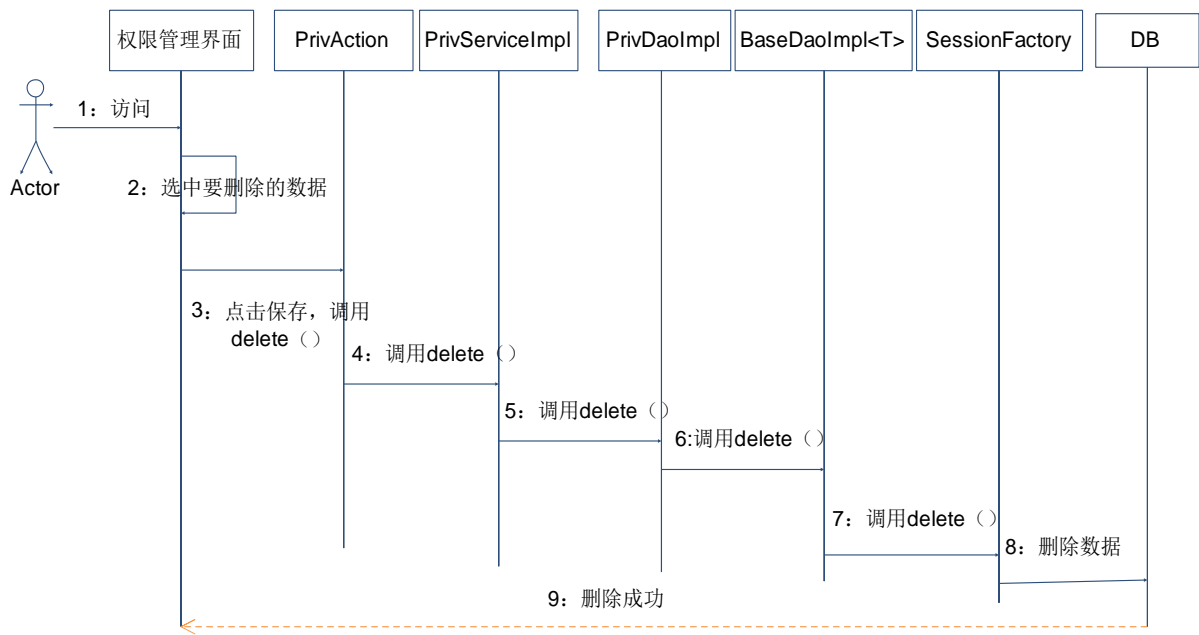


图 3-27 删除权限时序图

删除权限管理模块协作图，如下图 3-28 所示。

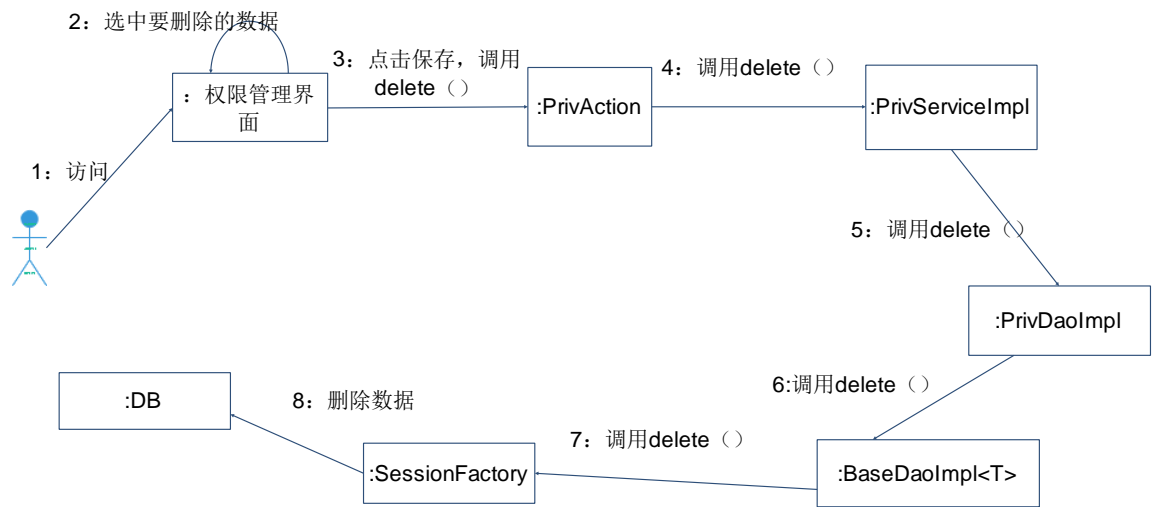


图 3-28 删除权限协作图

（4）查看权限列表模块

查看权限列表模块时序图，如下图 3-29 所示。

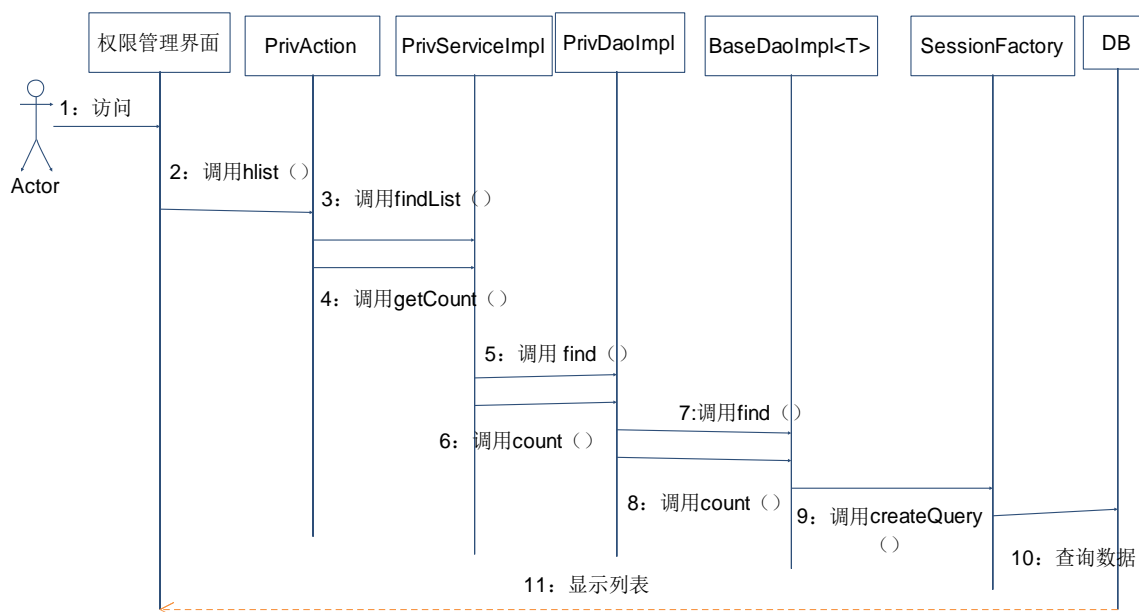


图 3-29 查看权限列表时序图

查看权限列表模块协作图，如下图 3-30 所示。

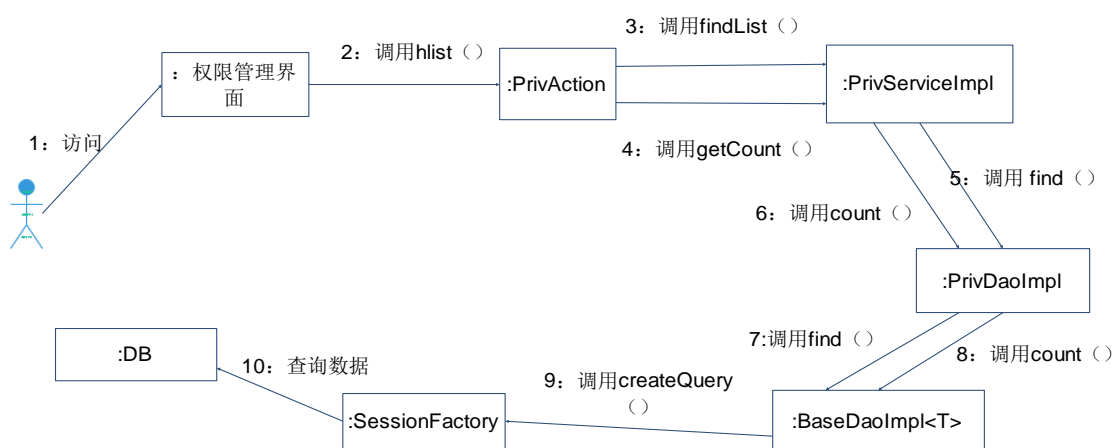


图 3-30 查看权限列表协作图

### 3.2.4 新闻管理

新闻管理部分，包括对新闻的增删该查，其用例图如下图 3-31 所示。

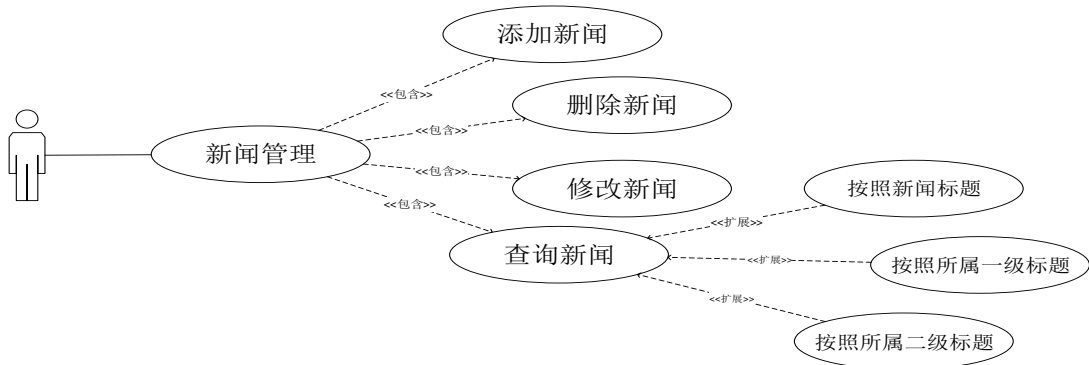


图 3-31 新闻管理用例图

### 3.2.5 一级导航

一级导航管理部分，包括对一级导航的增删该查，其用例图如下图 3-32 所示。

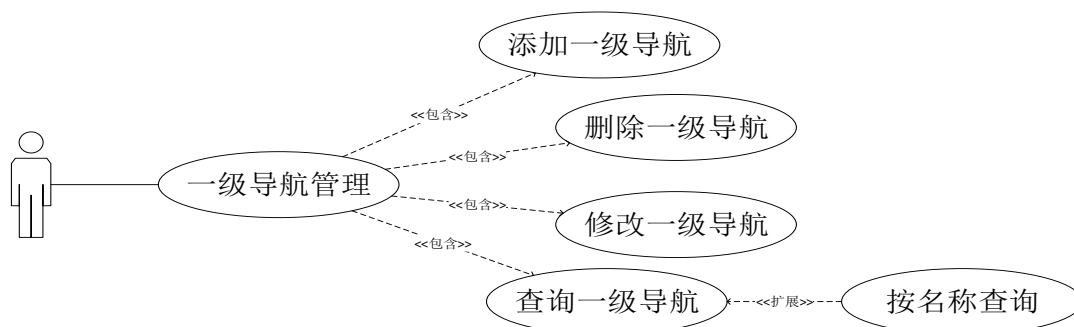


图 3-32 一级导航管理用例图

### 3.2.6 二级导航

二级导航管理部分，包括对二级导航的增删该查，其用例图如下图 3-33 所示。

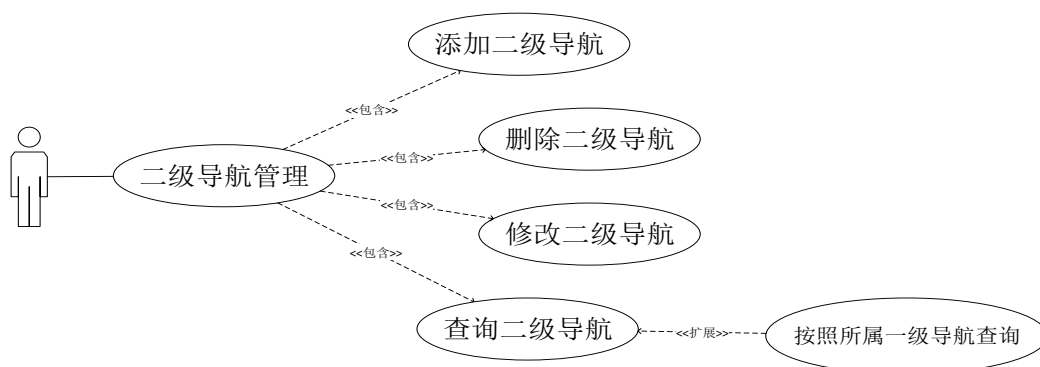


图 3-33 二级导航管理用例图



## 第四章 系统概要设计

### 4.1 系统结构的设计

#### 4.1.1 设计思想

系统分成几个相对独立的模块，但这些模块都进行集中式管理。

分层的模块化程序设计思想，整个系统采用模块化结构设计。作为应用程序有较强的可操作性和扩展性<sup>[8]</sup>。

合理的数据流设计，在应用系统设计中，相对独立的模块间以数据流相互连接，使各模块间的耦合性较低，方便系统运行，提高系统安全性。

#### 4.1.2 设计原则

为了使本系统功能齐全完备，操作简便，最大限度的提高软件的质量，从而满足用户的实际需要，在设计开发过程中遵循各项原则：合法性原则，实用性原则，准确性原则，易操作原则，源程序可读性原则，优化原则，安全性原则。

#### 4.1.3 系统结构设计

整个系统分为前台和后台管理系统两大部分，前台主要是新闻信息、通知公告的展示。后台管理系统包括用户管理、角色管理、权限管理、新闻管理、一级/二级导航管理等。

系统总体结构如图 4-1 所示。

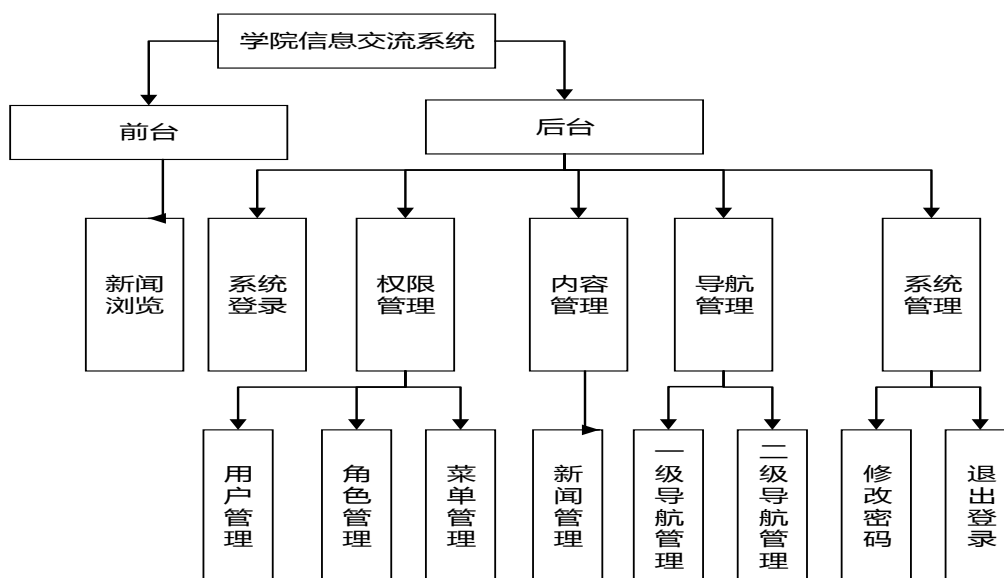


图 4-1 学院信息交流平台系统模块图

4.2 数据库设计

4.2.1 设计原则

E-R 图，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型 [11]。

学院信息交流平台数据库 E-R 模型图如图 4-2 所示。

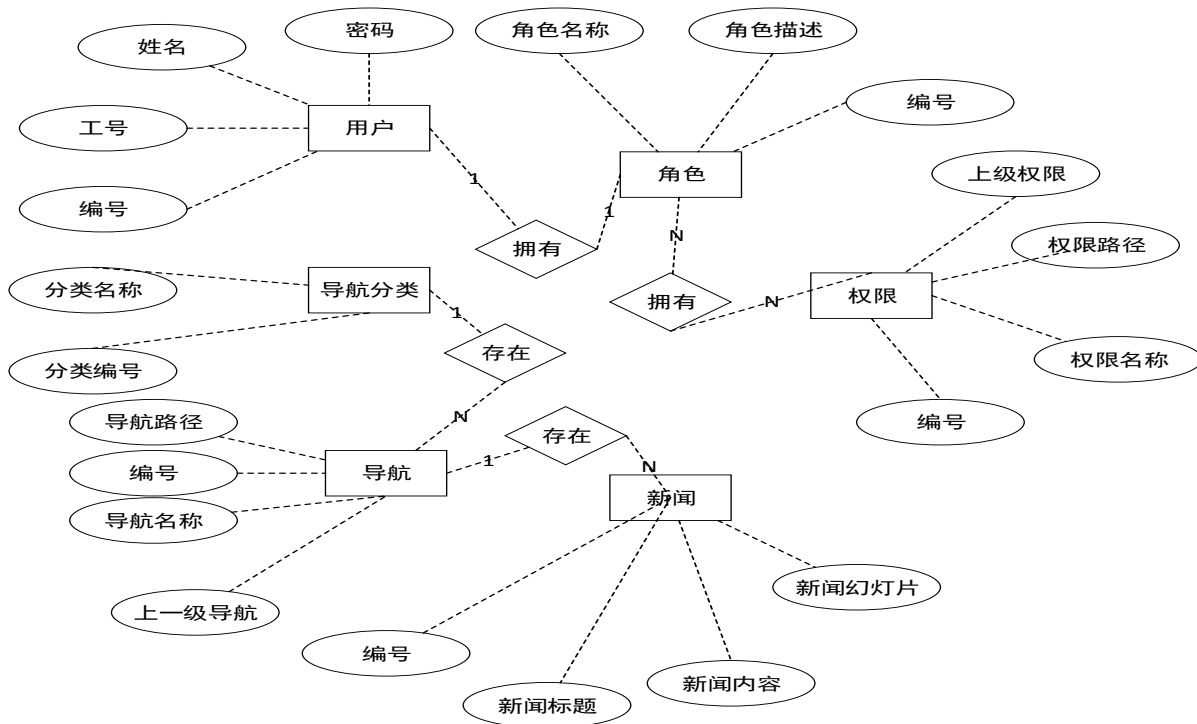


图 4-2 学院信息交流平台 E-R 图

4.2.2 逻辑设计

用户表，用于存储用户信息，编号为主键是唯一标识。表中各字段功能定义如表 4-1 所示。

表 4-1 用户表（users 表）		
字段名	字段类型	对应中文名
id	bigInt	用户编号
name	varchar	姓名
password	varchar	密码
num	varchar	工号

角色表，用于存储角色信息，编号为主键是唯一标识。表中各字段功能定义如表 4-2 所示。

表 4-2 角色表（role 表）

字段名	字段类型	对应中文名
id	bigInt	编号
name	varchar	角色名称
description	varchar	角色描述
createTime	varchar	创建时间

用户-角色表，用于存储用户对应的角色信息。表中各字段功能定义如表 4-3 所示。

表 4-3 用户-角色表（user\_role 表）

字段名	字段类型	对应中文名
userId	bigInt	用户编号
roleId	bigInt	角色编号

权限表，用于存储权限信息，编号为主键是唯一标识。表中各字段功能定义如表 4-4 所示。

表 4-4 权限表（privilege 表）

字段名	字段类型	对应中文名
id	bigInt	编号
text	varchar	权限名称
url	varchar	链接路径
pid	varchar	上一级权限编号

角色-权限表，用于存储角色对应权限信息，表中各字段功能定义如表 4-5 所示。

表 4-5 用户表（users 表）

字段名	字段类型	对应中文名
roleId	bigInt	角色编号
privId	bigInt	权限编号

导航分类表，用于存储导航分类信息，编号为主键是唯一标识。表中各字段功能定义如表 4-6 所示。

表 4-6 导航分类表（navcat 表）

字段名	字段类型	对应中文名
id	bigInt	编号
name	varchar	姓名
remark	varchar	备注

导航表，用于存储导航信息，编号为主键是唯一标识。表中各字段功能定义如表 4-7 所示。

表 4-7 导航表（nav 表）

字段名	字段类型	对应中文名
id	bigInt	编号
name	varchar	导航名称
pid	varchar	上级导航 id
path	varchar	链接路径
cid	bigInt	所属分类 id

新闻表，用于存储新闻信息，编号为主键是唯一标识。表中各字段功能定义如表 4-8 所示。

表 4-8 新闻表（news 表）

字段名	字段类型	对应中文名
id	bigInt	编号
title	varchar	新闻标题
content	varchar	新闻内容
picture	varchar	图片路径
nid	bigInt	导航标号
userId	bigInt	用户编号

### 4.3 系统功能设计

整体结构分为前台和后台两大部分。前台包括浏览新闻功能。后台管理包括对用

户、角色、权限、导航、新闻模块进行增删改查等不同操作，以及管理员的登录、修改密码、退出。下面主要对后台部分介绍。

### 4.3.1 用户登录

功能描述：已注册的用户可以根据注册账号和密码登入系统，如果信息有误，系统提示用户重新输入。

功能实现：用户点击“登陆”按钮，填写登入信息。信息无误，完成登录。信息错误系统提示登录失败。

用户登录的具体流程，如下图 4-3 所示。

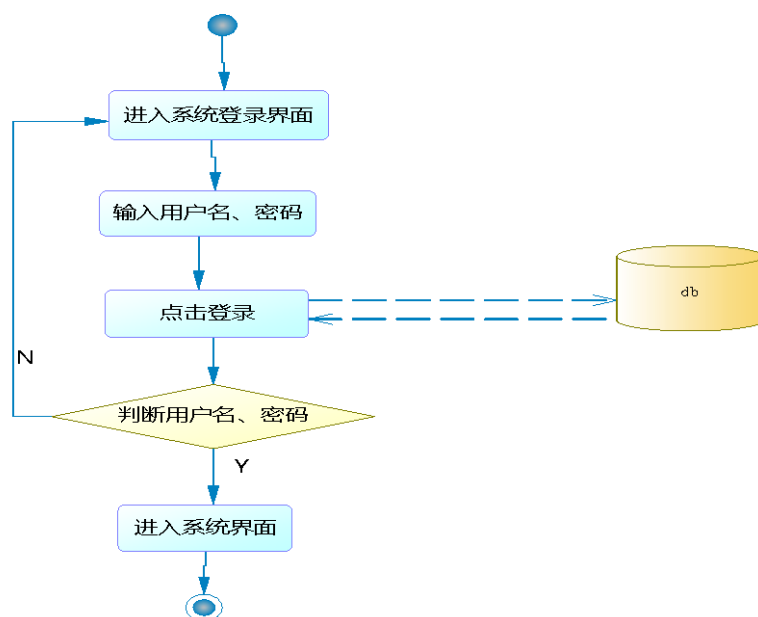


图 4-3 学院信息交流平台登录流程图

### 4.3.2 退出系统

功能描述：用户可以根据注册账号和密码登入系统，如果不进行相关操作可退出系统。

功能实现：用户点击“退出”按钮，退出系统，回到登录页面。

### 4.3.3 用户管理

#### (1) 添加用户

功能描述：用户管理界面，添加新用户。

功能实现：管理员进入用户管理界面，点击“添加”按钮，输入相应的课程信息，点击“添加”即可完成。

添加用户的具体流程，如下图 4-4 所示。

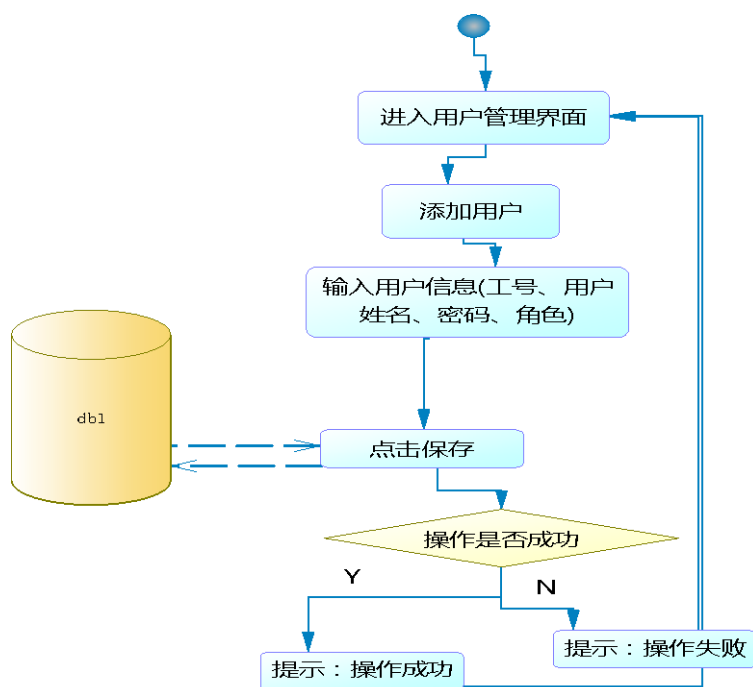


图 4-4 添加用户流程图

## (2) 修改用户

功能描述：用户管理界面，修改用户信息。

功能实现：管理员进入用户管理界面，点击“修改”按钮，修改对应的用户信息，点击“修改”即可完成。

修改用户的具体流程，如下图 4-5 所示。

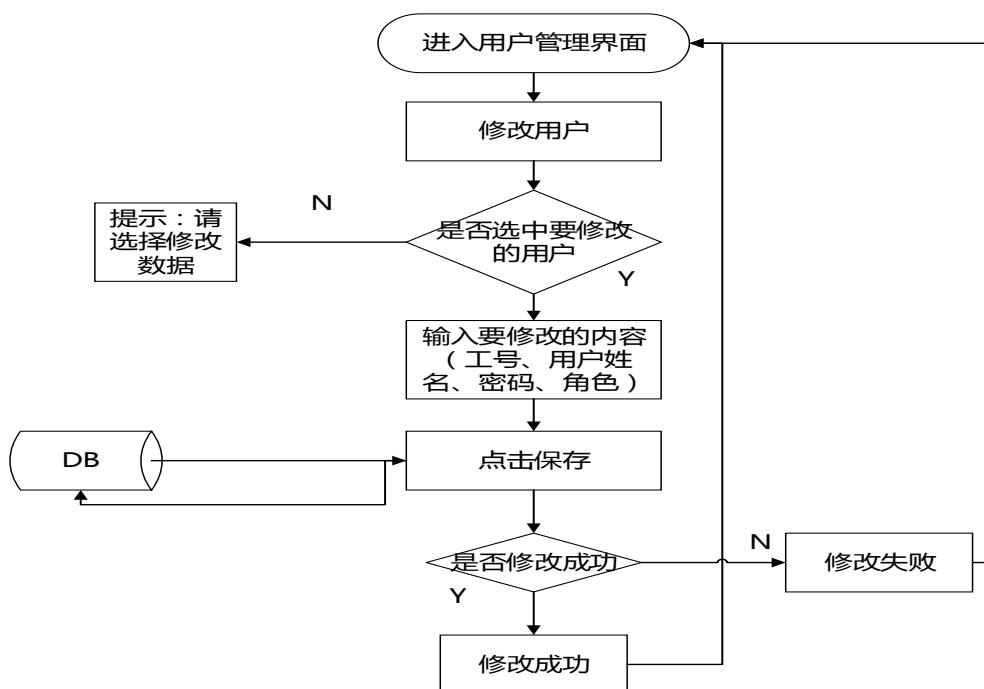


图 4-5 修改用户流程图

### （3）删除用户

功能描述：用户管理界面，删除用户信息。

功能实现：管理员进入用户管理界面，点击“删除”按钮，即可删除废弃的用户信息。

删除用户的具体流程，如下图 4-6 所示。

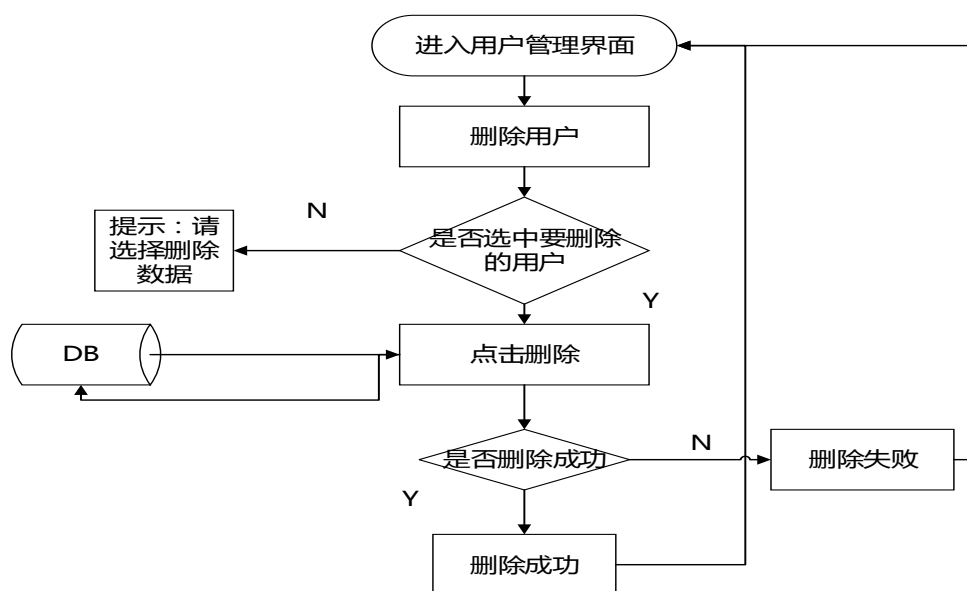


图 4-6 删除用户流程图

### （4）查询用户

功能描述：用户管理界面，查看用户列表。

功能实现：管理员进入用户管理界面，即可查看用户列表。

查询用户的具体流程，如下图 4-7 所示。

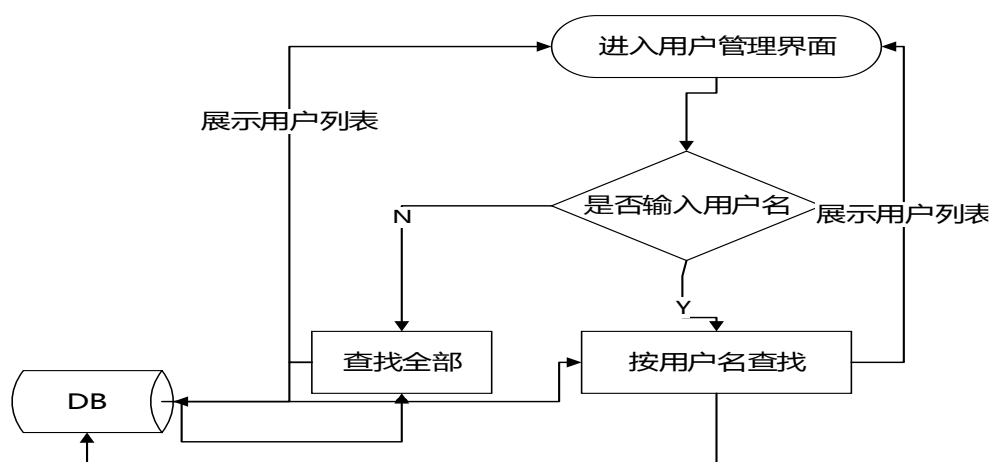


图 4-7 查看用户列表流程图

#### 4.3.4 角色管理

##### （1）添加角色

功能描述：角色管理界面，添加新角色。

功能实现：管理员进入角色管理界面，点击“添加”按钮，输入相应的角色信息，点击“添加”即可完成。

添加角色的具体流程，如下图 4-8 所示。

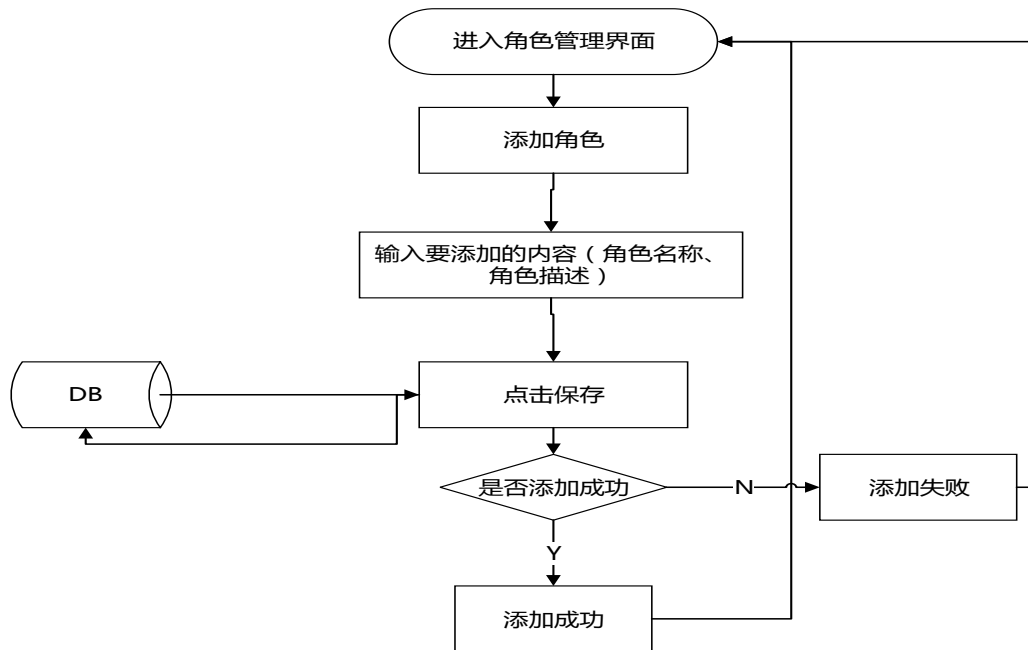


图 4-8 添加角色流程图

##### （2）修改角色

功能描述：角色管理界面，修改角色信息。

功能实现：管理员进入角色管理界面，点击“修改”按钮，修改对应的角色信息，点击“修改”即可完成。

修改角色的具体流程，如下图 4-9 所示。



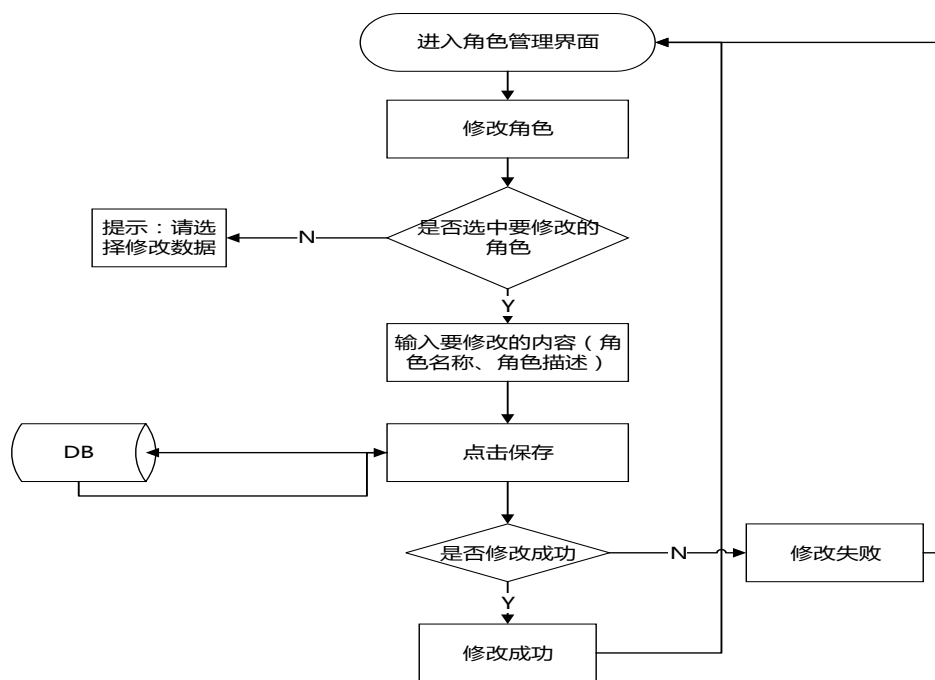


图 4-9 修改角色流程图

### （3）删除角色

功能描述：角色管理界面，删除角色信息。

功能实现：管理员进入角色管理界面，点击“删除”按钮，即可删除废弃的角色信息。删除角色的具体流程，如下图 4-10 所示。

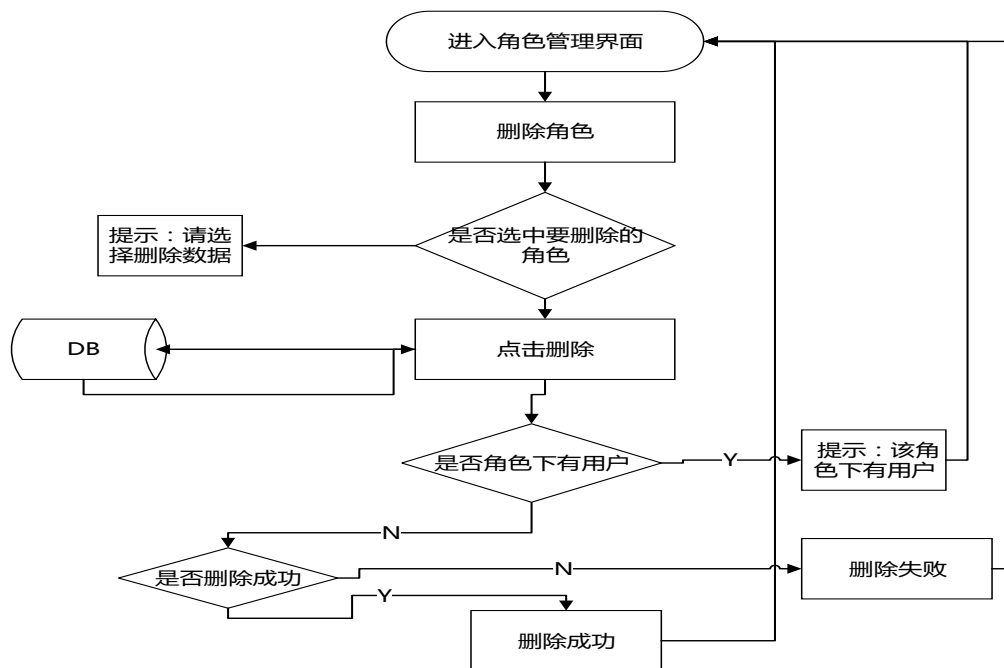


图 4-10 删除角色流程图

### （4）查询角色

功能描述：角色管理界面，查看角色列表。

功能实现：管理员进入角色管理界面，即可查看角色列表。

查询角色的具体流程，如下图 4-11 所示。

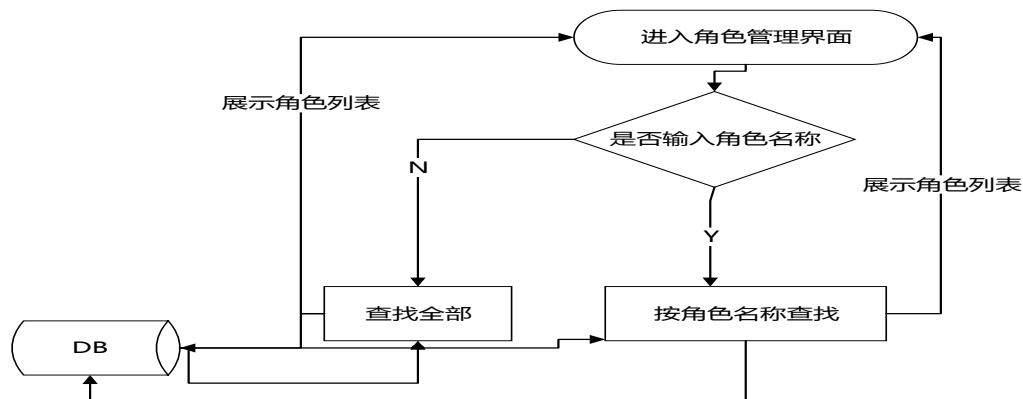


图 4-11 查看角色列表流程图

### 4.3.5 新闻管理

#### （1）添加新闻

功能描述：新闻管理界面，添加新闻。

功能实现：管理员进入新闻管理界面，点击“添加”按钮，输入相应的新闻信息，点击“添加”即可完成。

#### （2）修改新闻

功能描述：新闻管理界面，修改新闻信息。

功能实现：管理员进入新闻管理界面，点击“修改”按钮，修改对应的新闻信息，点击“修改”即可完成。

#### （3）删除新闻

功能描述：新闻管理界面，删除新闻信息。

功能实现：管理员进入新闻管理界面，点击“删除”按钮，即可删除废弃的新闻信息。

#### （4）查询新闻

功能描述：新闻管理界面，查看新闻列表。

功能实现：管理员进入新闻管理界面，即可查看新闻列表。

### 4.3.6 一级导航管理

#### （1）添加一级导航

功能描述：一级导航管理界面，添加一级导航：

功能实现：管理员进入一级导航管理界面，点击“添加”按钮，输入相应的一级导航信息，点击“添加”即可完成。

#### （2）修改一级导航

功能描述：一级导航管理界面，修改一级导航信息。

功能实现：管理员进入一级导航管理界面，点击“修改”按钮，修改对应的一级导航信息，点击“修改”即可完成。

#### （3）删除一级导航

功能描述：一级导航管理界面，删除一级导航信息。

功能实现：管理员进入一级导航管理界面，点击“删除”按钮，即可删除废弃的一级导航信息。

#### （4）查询一级导航

功能描述：一级导航管理界面，查看一级导航列表。

功能实现：管理员进入一级导航管理界面，即可查看一级导航列表。

### 4.3.7 二级导航管理

#### （1）添加二级导航

功能描述：二级导航管理界面，添加二级导航。

功能实现：管理员进入二级导航管理界面，点击“添加”按钮，输入相应的二级导航信息，点击“添加”即可完成。

#### （2）修改二级导航

功能描述：二级导航管理界面，修改二级导航信息。

功能实现：管理员进入二级导航管理界面，点击“修改”按钮，修改对应的二级导航信息，点击“修改”即可完成。

#### （3）删除二级导航

功能描述：一级导航管理界面，删除二级导航信息。

功能实现：管理员进入二级导航管理界面，点击“删除”按钮，即可删除废弃的二级导航信息。

#### （4）查询二级导航

功能描述：一级导航管理界面，查看一级导航列表。

功能实现：管理员进入二级导航管理界面，即可查看二级导航列表。

## 第五章 系统详细设计与实现

本章节，对系统实现效果做展示，并着重讲解了系统代码具体实现。下面是每个模块具体的界面设计以及实现的关键代码。

### 5.1 后台登录模块

#### 5.1.1 用户管理界面

用户登录界面如下图 5-1 所示。



图 5-1 用户登录界面

#### 5.1.2 关键代码实现

```
public String login() {  
    if (user.getNum() != null) {  
        User curUser =  
userService.findByNumAndPassword(user.getNum(),  
  
DigestUtils.md5Hex(this.user.getPassword()));  
        if (curUser != null) {  
  
            ActionContext.getContext().getSession().put("curUser",  
curUser);  
            return "admin_index";  
        } else {  
            addFieldError("login", "用户名或密码不正确！");  
            return "login_fail";  
        }  
    } else {  
        return "login_view";  
    }  
}
```

```
}  
}
```

5.2 用户管理模块

5.2.1 用户管理界面

用户管理界面设计如图 5-2 所示，当前展示的是用户列表，可以通过点击“删除”，“修改”，“删除”按钮对用户进行操作。



图 5-2 用户管理界面

5.2.2 关键代码实现

此代码用户后台返回用户列表信息：

```
public String hlist() throws Exception {  
    List<User> uList = userService.findList(userName, new PageBean(page, rows));  
    Long total = userService.getCount(userName);  
    UserPropertyFilter filter = new UserPropertyFilter();  
    filter.setExcludes(new HashMap<Class<?>, String[]>() {  
        {  
            put(User.class, new String[] { "news", "createTime", "updateTime" });  
            put(Role.class, new String[] { "parent", "users", "privileges", "updateTime", "createTime" });  
        }  
    });  
    String jsonStr = JSON.toJSONString(uList, filter);  
    JSONObject result = new JSONObject();  
    result.put("rows", jsonStr);  
    result.put("total", total);  
    ResponseUtil.write(ServletActionContext.getResponse(), result);  
    return null;  
}
```

前台展示用户列表：

```

$("#datagrid").datagrid({
    title:'用户列表',
    width:'auto',
    height:'auto',
    url:'user_hlist.action',
    fitColumns:true,
    rownumbers:true,
    pagination:true,
    columns:[[
        {
            field:'ck',
            width:50,
            checkbox:true
        },{
            field:'id',
            title:'编号',
            width:20,
            align:'center',
            hidden:false
        },{
            field:'num',
            title:'工号',
            width:50,
            align:'center'
        },{
            field:'name',
            title:'姓名',
            width:50,
            align:'center'
        },{
            field:'password',
            title:'密码 ',
            width:100,
            align:'center'
        },
        {
            field:'roles',title:'角色名称', width:80,
            formatter: function(value,row,index){
                for(var key in value[0]){
                    return value[0]["name"];
                }
            }
        }
    ]
})
    
```

```
]],
toolbar:'#tb'

});
```

5.3 角色管理模块

5.3.1 角色管理界面

角色管理界面设计如图 5-3 所示，当前展示的是角色列表信息，可以通过点击“删除”，“修改”，“删除”按钮对角色进行操作。

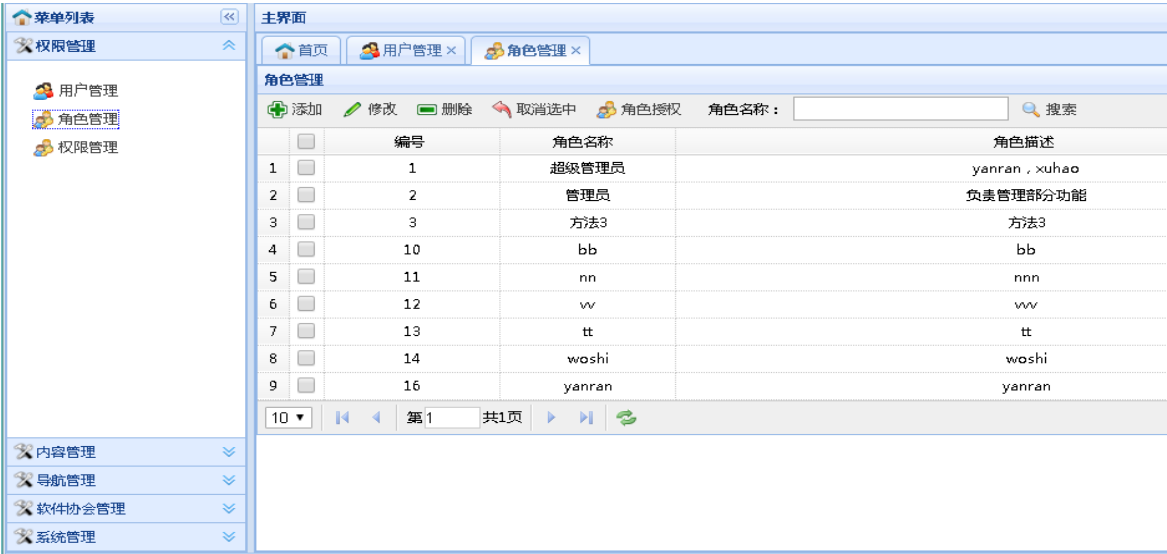


图 5-3 角色管理界面

5.3.2 关键代码实现

前台添加角色：

```
function saveRole(){
    $("#myform").form('submit',{
        url:'role_save.action',
        onSubmit:function(){
            return $(this).form("validate");
        },
        success:function(result){
            if(result == 1){
                $("#mydialog").dialog('close');
                $("#datagrid").datagrid('reload');
                $("#datagrid").datagrid('unselectAll');
                $.messager.show({
                    title:'提示信息',
                    msg:'保存成功！'
                });
            }else{
                $("#mydialog").dialog('close');
            }
        }
    });
}
```

```
$.messenger.alert('系统提示','操作失败！ ');
}
}
});
}
```

后台：

```
public String save() throws Exception {
    roleService.save(role);
    int result = 1;
    ResponseUtil.write(ServletActionContext.getResponse(), Integer
        .valueOf(result));
    return null;
}
```

5.4 权限管理模块

5.4.1 权限管理界面

权限管理界面，如下图 5-4 所示：

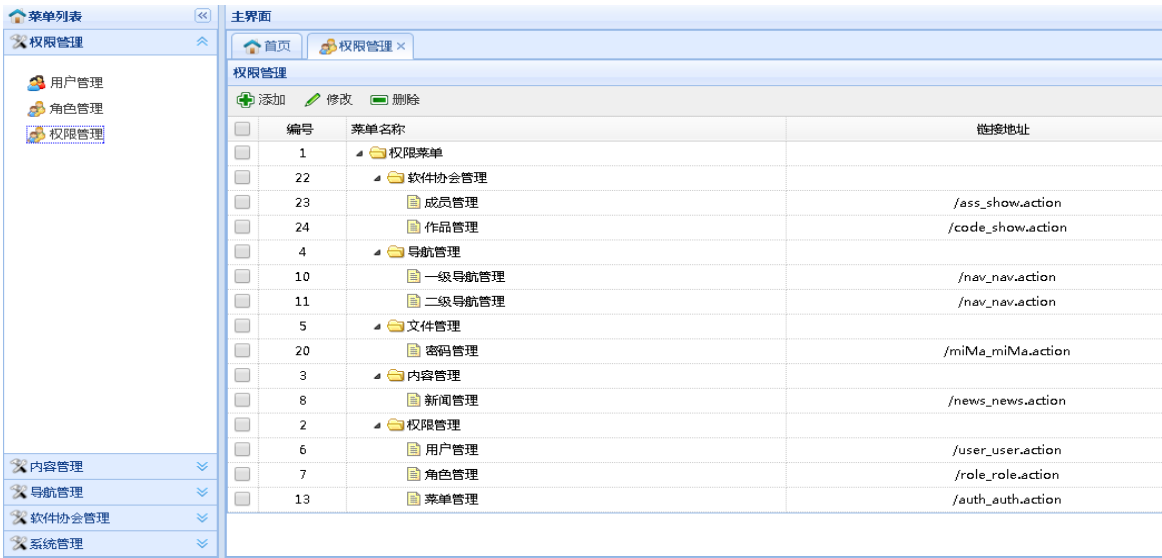


图 5-4 权限管理界面

5.4.2 关键代码实现

前台修改界面：

```
function editAuth(){
    var node = $("#treegrid").treegrid('getSelected');
    if(node == null){
        $.messenger.alert("系统提示！","请选择要修改的节点！");
        return ;
    }else{
        $("#mydialog").dialog('open');
        $("#mydialog").dialog('setTitle','修改节点信息！');
    }
}
```



```
        $("#id").val(node.id);
        $("#text").val(node.text);
        $("#url").val(node.url);
        url = "priv_save.action?";
    }
}

后台：

public String save() throws Exception {
    Privilege findById = null;
    if (parentId != null) {
        findById = privService.findById(parentId);
        privilege.setParent(findById);
    } else {
        findById = privService.findById(privilege.getId());
        findById.setText(privilege.getText());
        findById.setUrl(privilege.getUrl());
        privilege = findById;
    }
    privService.save(privilege);
    int result = 1;
    ResponseUtil.write(ServletActionContext.getResponse(), Integer.valueOf(result));
    return null;
}
```

5.5 新闻管理模块

新闻管理界面如图 5-5 所示。



图 5-5 权限管理界面

5.6 一级导航管理模块

一级导航管理界面如下图 5-6 所示。

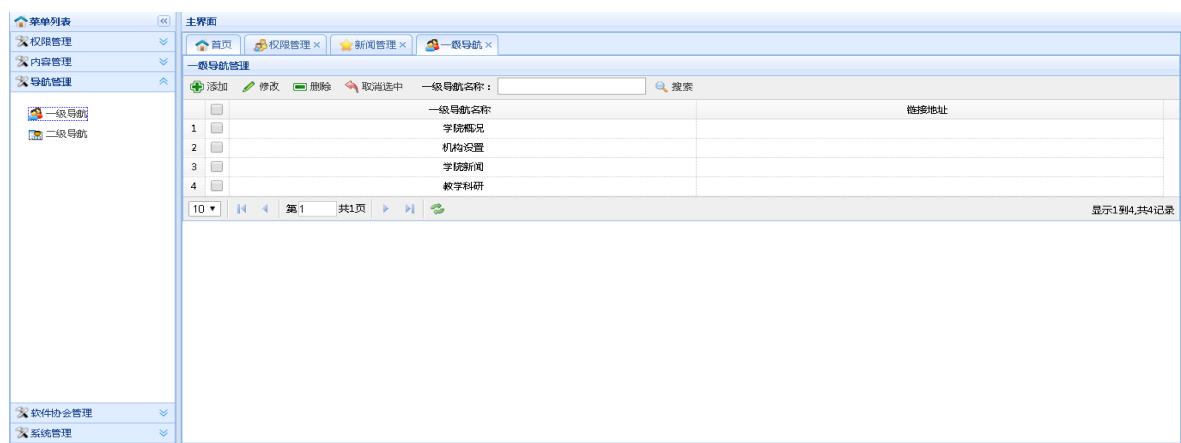


图 5-6 一级导航管理界面

## 5.7 二级导航管理模块

二级导航管理界面如下图 5-7 所示。

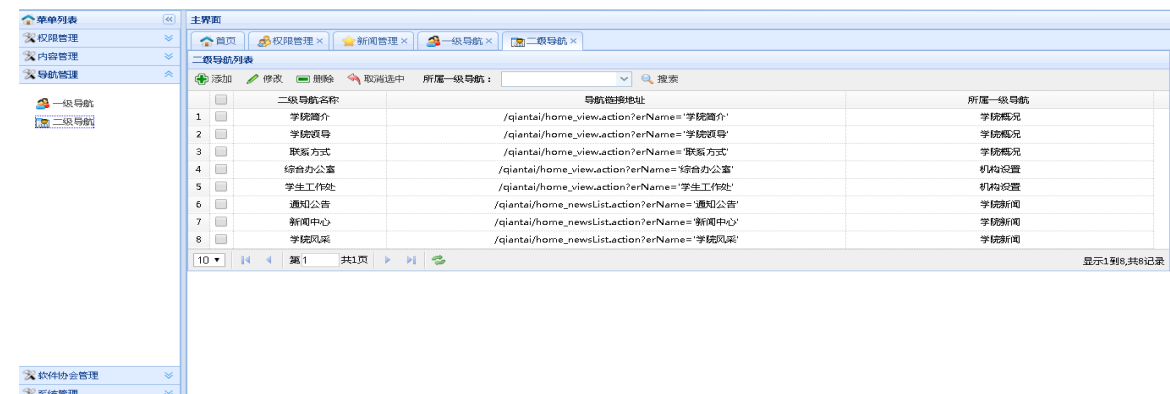


图 5-7 二级导航管理界面

## 第六章 系统测试

### 6.1 测试概述

在设计系统的过程中，存在一些错误是必然的。对于语句的语法错误，在程序运行时自动提示，并请求立即纠正，因此，这类错误比较容易发现和纠正。但另一类错误是在程序执行时由于不正确的操作或逻辑错误导致的错误结果。这类错误隐蔽性强，有时会出现，有时又不出现，因此，对这一类动态发生的错误的排查是耗时费力的。

#### 6.1.1 测试重要性

软件测试在软件生命周期中占据重要的地位，在传统的瀑布模型中，软件测试学仅处于运行维护阶段之前，是软件产品交付用户使用之前保证软件质量的重要手段。近来，软件工程界趋向于一种新的观点，即认为软件生命周期每一阶段中都应包含测试，从而检验本阶段的成果是否接近预期的目标，尽可能早的发现错误并加以修正，如果不在早期阶段进行测试，错误的延时扩散常常会导致最后成品测试的巨大困难<sup>[12]</sup>。

事实上，对于软件来讲，不论采用什么技术和什么方法，软件中仍然会有错。采用新的语言、先进的开发方式、完善的开发过程，可以减少错误的引入，但是不可能完全杜绝软件中的错误，这些引入的错误需要测试来找出，软件中的错误密度也需要测试来进行估计。测试是所有工程学科的基本组成单元，是软件开发的重要部分。自有程序设计的那天起测试就一直伴随着。统计表明，在典型的软件开发项目中，软件测试工作量往往占软件开发总工作量的 40% 以上。而在软件开发的总成本中，用在测试上的开销要占 30% 到 50%。如果把维护阶段也考虑在内，讨论整个软件生存期时，测试的成本比例也许会有所降低，但实际上维护工作相当于二次开发，乃至多次开发，其中必定还包含有许多测试工作。

在实践中，软件测试的困难常常使人望而却步或敷衍了事，这是由于对测试仍然存在一些不正确的看法和错误的态度，这包括：

- （1）认为测试工作不如设计和编码那样容易取得进展难以给测试人员某种成就感。
- （2）以发现软件错误为目标的测试是非建设性的，甚至是破坏性的，测试中发现错位是对责任者工作的一种否定。
- （3）测试工作枯燥无味，不能引起人们的兴趣。

（4）测试工作是艰苦而细致的工作。

（5）对自己编写的程序盲目自信，在发现错误后，顾虑别人对自己的开发能力的看法。

这些观点对软件测试工作是极为不利的，必须澄清认识、端正态度，才可能提高软件产品的质量。

### 6.1.2 测试目的

如果测试的目的是为了尽可能多地找出错误，那么测试就应该直接针对软件比较复杂的部分或是以前出错较多的位置：

- （1）软件测试是为了发现错误而执行程序的过程。
- （2）测试是为了证明程序有错，而不是证明程序无错误。
- （3）一个好的测试用例是在于它能发现至今未发现的错误。
- （4）一个成功的测试是发现了至今未发现的错误的测试。

这种观点可以提醒人们测试要以查找错误为中心，而不是为了演示软件的正确功能。但是仅凭字面意思理解这一观点可能会产生误导，认为发现错误是软件测试的唯一目，查找不出错误的测试就是没有价值的，事实并非如此。

首先，测试并不仅仅是为了要找出错误。通过分析错误产生的原因和错误的分布特征，可以帮助项目管理者发现当前所采用的软件过程的缺陷，以便改进。同时，这种分析也能帮助我们设计出有针对性地检测方法，改善测试的有效性。其次，没有发现错误的测试也是有价值的，完整的测试是评定测试质量的一种方法。

### 6.1.3 测试步骤

与开发过程类似，测试过程也必须分步骤进行，每个步骤在逻辑上是前一个步骤的继续。大型软件系统通常由若干个子系统组成，每个子系统又由若干个模块组成。因此，大型软件系统的测试基本上由下述几个步骤组成：

#### （1）模块测试

在这个测试步骤中所发现的往往是编码和详细设计的错误。

#### （2）系统测试

在这个测试步骤中发现的往往是软件设计中的错误，也可能发现需求说明中的错误。

#### （3）验收测试

在这个测试步骤中发现的往往是系统需求说明书中的错误。

6.1.4 测试用例

用户登录测试用例如下图 6-1：

功能特性		用户登录验证			
测试目标		验证输入合理性			
测试数据		用户名称：123 、 密码：123			
测试内容	操作描述	数据	期望结果	实际结果	测试状态
1	输入用户名、密码	用户名为空	提示：用户名或密码不正确	提示：用户名或密码不正确	与期望结果相同
2	输入用户名、密码	密码为空	提示：用户名或密码不正确	提示：用户名或密码不正确	与期望结果相同
3	输入用户名、密码	用户名称：123 密码：123	登录正常	登录正常	与期望结果相同

图 6-1 用户登录测试用例

6.2 本章小结

通过对程序的调试，查找资料我学到了一些程序调试的基本方法，也学到了许多新的知识，积累了更多的经验。

## 第七章 结论

经过一个月的努力，最终我们圆满的实现了整个系统，从一开始的看视频学习 SSH 框架，到后来抛弃视频，从零做起。整个过程中遇到很多问题，例如：SSH 框架整合不成功，获取不到数据，空指针，数组越界，编译器的抉择等等...但最后凭借我们的努力和坚持，通过各方面的搜集资料，成功解决了问题，这告诉我们“困难并不可怕，可怕的是失去一定能解决它的信心”，我相信它会成为我一次人生阅历，告诫着我无论如何都不能轻言放弃。与此同时还让我体会到团队合作的重要性，原本看似遥不可及的程序，却因为我们精诚合作，由理想变成了现实。正所谓“众人拾柴火焰高”，说的也是这个道理吧。

当然，系统中也存在着各种各样的问题，比如说：引入的编译器是标准版的，无法实现图片、文本居中效果；是否支持高并发；界面的设计方面还有待改善...以后，我会逐步改善系统，解决这些问题。

我希望在未来的工作中，依然能如此这般，脚踏实地。用心去做好每件事。

## 参考文献

- [1] 邱光谊.管理信息系统, 电子工业出版社 2002.
- [2] 常晋义等.管理信息系统, 中国电力出版社 2002.
- [3] 李刚.轻量级 J2EE 企业应用实战— Struts+Spring+Hibernate 整合开发[M].北京: 电子工业出版社, 2007: 158-159.
- [4] 沃尔斯.Spring in Action[M].北京: 人民邮电出版社, 2006: 56-59.
- [5] 李刚.Struts 权威指南[M].电子工业出版社, 2007: 156-168.
- [6] 曹晓钢, 唐勇, 夏昕.深入浅出 Hibernate[M].北京: 电子工业出版社, 2005.6.
- [7] 基于 B/S 结构的软件开发技术,西安电子科技大学出版社, 2006.
- [8] 郝晓玲.基于 JSP 的信息系统设计与实现[J].微计算机应用, 2004.
- [9] 胡晓旭.基于 B/S 模式和 JSP 技术的网络教学系统的研究与实现[D], 2010.
- [10] 白瑞俊 田全红 刘文杰 张其斌.中间件技术与 J2EE 平台综述[J].计算机光盘软件与应用, 2014.
- [11] 萨师煊, 王珊.数据库系统概论 第 3 版.北京: 高等教育出版社, 2000.
- [12] 张海藩.软件工程导论 第 3 版.北京: 清华大学出版社, 1998.
- [13] 范绕, 胡博.基于 B/S 架构的高校教学视频资源库平台的设计[J].科教平台 2016.
- [14] 汤谷云, 崔晨荣.基于 B/S 的教学资源系统的设计[J].简报, 2003.
- [15] 钟峰.基于 B/S 架构工程教育实践过程管理系统[D], 2012.

## 致 谢

大学四年的学习生活即将结束，在此，我要感谢所有曾经教导过我的老师和关心过我的同学，他们在我成长过程中给予了我很大的帮助。本文能够成功的完成，要特别感谢我的导师王雪冰老师的关怀和教导。

通过此次的论文我学到了很多知识，跨越了传统方式下的教与学的体制束缚。在论文的写作过程中，我通过查资料和搜集有关的文献，培养了自学能力和动手能力，并且由原来被动接受知识转换为主动的寻求知识，这是我学习方法上的一个重大突破。通过毕业论文，我学会了如何将学到的知识转化为自己的东西，学会了怎么更好地处理知识和实践相结合的问题。在论文的写作过程中我也学到了做任何事情所要有的态度和心态，首先是做学问要一丝不苟，对于发展过程中出现的任何问题和偏差都不要轻视，要通过正确的途径去解决。在做事情的过程中要有耐心和毅力，不要一遇到困难就打退堂鼓。只要坚持下去就会找到很好的解决办法。

总之，此次论文的写作过程让我收获很多，它为我大学三年划上了一个完美的句号，也为我将来的人生之路做好了一个很好的铺垫。

再次感谢我的大学 and 所有帮助过我并给我鼓励的老师、同学和朋友们，谢谢你们。