

# 武汉轻工大学

## 硕士专业学位论文

### 基于 Spring Boot 的学生信息管理系统 的设计与实现

论文作者：杨东

指导教师：陈西曲

专业学位类别：工程硕士

专业学位领域：计算机技术

所在学院：电气与电子工程学院

2021 年 6 月

分类号 \_\_\_\_\_

密 级 \_\_\_\_\_

U D C \_\_\_\_\_

学校代码 10496

武汉轻工大学

# 硕士学位论文

基于 Spring Boot 的学生信息管理系统的设计  
与实现

Design and Implementation of Student  
Information Management System Based on  
Spring Boot

论文作者 \_\_\_\_\_ 杨东 \_\_\_\_\_

指导教师 \_\_\_\_\_ 陈西曲 \_\_\_\_\_

专业学位类别 \_\_\_\_\_ 工程硕士 \_\_\_\_\_

专业学位领域 \_\_\_\_\_ 计算机技术 \_\_\_\_\_

论文提交日期 2021.06.06 论文答辩日期 2021.05.29

答辩委员会主席 \_\_\_\_\_ 艾青松 \_\_\_\_\_ 评阅人 \_\_\_\_\_ 盲审 \_\_\_\_\_

2021 年 6 月

## 论文独创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果，除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的研究成果，也不包含为获得 **武汉轻工大学** 或其它教育机构的学位或证书而使用过的材料。对本文的研究做出重要贡献的个人或集体均已在文中作了明确的说明并表示谢意。本人完全意识到本声明的法律责任和法律后果由本人承担。

论文作者签名：杨东

签字日期：2021 年 6 月 10 日

## 论文授权使用授权书

本学位论文作者完全了解 **武汉轻工大学** 有关保留、使用学位论文的规定，特授权 **武汉轻工大学** 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或其他复制手段保存、汇编，以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复本和电子文件。

（保密的学位论文在解密后适用本授权说明）

论文作者签名：杨东

签字日期：2021 年 6 月 10 日

导师签名：陈西中

签字日期：2021 年 6 月 10 日

## 摘 要

学生信息管理工作是我国高等院校管理的一个重要组成部分,在日常学生信息管理工作过程中,各种学生管理信息不断地增加,采用传统的手工管理方式很难去准确进行学生信息管理,显得十分费时费力,工作中的质量和管理效率都不能及时得到有效保障。因此我国各院校纷纷开发并推出一系列高校学生个人信息安全管理软件系统,以适用于日常的院校学生信息管理工作。随着时间的推移,技术的进步,大部分以J2EE 框架为核心的管理系统出现了容易卡顿、界面老化等问题,已经满足不了高校日益繁重的学生信息管理工作的需要。基于此目的,本文设计并实现了基于 Spring Boot 的学生信息管理系统。而且该系统有着多样表达、高频利用等特征。因此,选取的开发框架为 Spring Boot,其特征为维护简单、开发便捷。

结合软件系统的生命周期进行设计,首先分析了软件的功能需求,如核心功能、安全要求、性能要求,然后设计了系统的整体功能框架,信息管理系统主要包括了学院信息管理服务等五个模块,并详细分析管理模块的具体需求。结合分析结果,利用相关的开发工具进行系统的整体设计及开发,主要包括学生用户信息管理和学生专业信息管理等核心业务功能,为了验证系统运行效果,使用多种测试法对其安全性、核心功能及各方面性能展开全面测试。

总结得出,此次开发期间,各项开发任务基本完成,实现了系统各项功能,在安全性与性能方面也得到足够保障。此次选取的开发框架为 MyBatis、Spring Boot 等框架,代码的耦合度较低,开发及维护都比较便捷。同时,选取的是前端模板开发和插件发布方法,这也让管理系统能够运行于各类网络环境,且具有简洁的操作界面,对操作要求较低。本文最后详细测试了系统的基本功能,确保系统可以稳定运行,具有较高的响应速度,满足管理工作的各类需求。

**关键词:** 管理系统; Spring Boot 框架; 软件测试; 模块设计

## ABSTRACT

Student information management has been an important part of the management in Colleges and Universities in China. In the daily process of student information management, with the continuous increase of various student management information, it is difficult to accurately manage students' information by using this traditional non manual management method, it seems to be very time-consuming and laborious, and the quality and management efficiency in the work can't be effectively guaranteed in time. Therefore, a series of personal information security management software systems have been developed and launched in various colleges and universities in China, which can be applied to the daily information management of college students. With the development of technology, most of the management systems with J2EE framework have been prone to Cayton and interface aging, which can't meet the needs of the increasingly heavy student information management in Colleges and Universities. Based on this purpose, this paper designs and implements the student information management system based on spring boot. The system has many features such as expression and high frequency utilization. Therefore, the selected development framework is Spring Boot, which is characterized by simple maintenance and convenient development.

Combined with the life cycle of software system, the article analyzes the function requirements of software, such as core function, security requirements and performance requirements, then designs the overall functional framework of the system. The information management system mainly includes five modules, such as college information management service module, and analyzes the specific requirements of management module in detail. Combined with the analysis results, the system is designed and developed by using relevant development tools, including the core business functions such as student user information management and student professional information management. In order to verify the operation effect of the system, the safety, core competence and performance of all aspects are tested comprehensively by various test methods.

The conclusion is that during the development, all development tasks are basically completed, and all functions of the system are realized, and the security and performance are also guaranteed. The selected development framework is mybatis, Spring Boot and other frameworks. The code has high coupling degree and is convenient to develop and

maintain. At the same time, the front-end development mode and plug-in release method are selected, which also enables the management system to run in various network environments, and has a simple operation interface, which has low requirements for operation. Finally, the basic functions of the system are tested in detail, which ensures the system can run stably, has high reaction rate and meets the various requirements of management.

**Key Words:** management system; spring boot; framework software test; module design

# 目 录

摘 要.....	I
ABSTRACT.....	II
1 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究概况.....	2
1.2.1 国内系统的发展现状.....	2
1.2.2 国外系统的发展现状.....	3
1.3 主要研究内容.....	3
1.4 本文的组织结构.....	4
1.5 本章小结.....	5
2 关键技术分析.....	6
2.1 Web MVC 模式.....	6
2.2 Spring Boot 框架.....	8
2.2.1 Spring 框架.....	8
2.2.2 Spring Boot 框架.....	9
2.3 MyBatis 框架.....	10
2.4 Free Marker 技术.....	11
2.5 分布式架构技术.....	12
2.6 本章小结.....	13
3 系统分析与设计.....	15
3.1 系统需求分析.....	15
3.1.1 系统需求概述.....	15
3.1.2 系统功能需求分析.....	16
3.1.3 系统用户及用例分析.....	17
3.1.4 系统性能需求分析.....	19
3.1.5 系统安全需求分析.....	20

3.2	系统总体设计.....	20
3.2.1	系统体系结构设计.....	20
3.2.2	系统功能结构设计.....	22
3.3	系统功能模块设计.....	22
3.3.1	用户信息管理模块.....	22
3.3.2	学院管理模块.....	24
3.3.3	专业管理模块.....	26
3.3.4	学生管理模块.....	28
3.3.5	系统管理模块.....	29
3.4	系统数据库设计.....	31
3.4.1	数据库建模说明.....	32
3.4.2	数据库表结构说明.....	32
3.5	本章小结.....	36
4	系统实现与测试.....	37
4.1	系统开发环境.....	37
4.2	系统功能模块实现.....	37
4.2.1	系统公共部分说明.....	37
4.2.2	用户信息管理模块实现.....	38
4.2.3	学生管理模块实现.....	40
4.2.4	学院管理模块实现.....	43
4.2.5	专业管理模块实现.....	44
4.2.6	系统管理模块实现.....	46
4.3	分布式架构技术实现.....	48
4.4	Zookeeper 消息中间件部署.....	49
4.5	系统测试.....	49
4.5.1	测试方法.....	49
4.5.2	测试用例.....	50
4.5.3	测试结论.....	54
4.6	本章小结.....	55
5	总结与展望.....	56
5.1	总结.....	56



5.2 展望.....	56
参考文献.....	58
致 谢.....	61

# 1 绪论

## 1.1 研究背景与意义

作为社会人才培养与学校教学科研的重要活动场所,高等院校管理中的学生信息管理工作不仅是一项重要的教学管理任务,同时也应该是直接反应我国高等院校学生管理水平与学校信息化体系建设的重要依据。随着我国高等教育事业规模的不断发展扩大,毕业学生人数的不断扩大增多,学生个人信息的收集管理工作也日益艰巨。特别是由于近几年来,国家教育政策的宏观调整,我国许多高等院校进行大规模的本科扩招,给我国高等院校的本科教学管理、学生生活管理、后勤保障管理等各个方面都带来很大的管理压力<sup>[1]</sup>。

近些年,我国大部分高校已经采用了基于 Java Web 技术或 ASP.NET 技术的信息管理系统。也确实显著提高了学生信息管理工作的效率。但是随着技术的进步,现有的学生信息管理系统已经满足不了学生工作的发展需要。出现了一些新的问题。例如,界面老化不美观、操作流程复杂、设计不够人性化。可靠性稳定性不够高,解决高并发场景的能力不强。面对以上一些问题,重新设计一款基于新技术的高校学生信息管理系统就显得尤为必要了。

高校学生信息管理系统意义主要体现在以下几个方面:

### (1) 提高教职员工工作效率,降低大部分高校办公费用

有了学生信息文档管理办公系统后,可以大大简化教师日常的办事工作流程,同时在日常办公活动过程中,教师能够通过学生信息管理软件系统进行实时管理存档以及及时发布办公信息,不用再打印转换成任何纸质办公文档,既大大提高了教师办事效率,又同时可以轻松实现准确的无纸化文档办公,降低教师办公成本费用。

### (2) 规范工作流程

学生信息管理系统可以对师生的日常工作过程进行记录,能够做到随时随地及时地查询,便于管理部门对学生工作进行监督,也能够给学生管理工作的总结和交流带来一定的便利。

### (3) 实现分布式办公

由于学生信息管理软件系统内部采用了现代互联网无线接入管理技术,使高校学生信息管理工作不仅仅只是局限于学生办公室,可以大幅扩大学生办公的网络地理应用范围,真正切实做到了学生工作移动在线办公和学生工作异地在线办公,办公管理形式更加灵活和方便<sup>[2]</sup>。

学生个人信息管理软件系统针对我国高校学生个人信息的使用特点以及高校管理中实际工作需要而进行设计,能够直接有效地实现高校学生个人信息管理工作的流程化和信息化,减轻教职员工的日常工作压力,方便高效率、规范化地收集、管理大量的高校学生个人信息,并有效避免人为误操作的管理错误和不规范管理行为。而且,学生信息管理系统有效增加了数据的安全性和可靠性。

## 1.2 国内外研究概况

信息管理技术的不断发展和信息管理的数字化,使绝大多数高校的学生个人信息管理工作取得长足进步。数字化信息校园已经成为许多地方高校的战略目标和发展趋势,许多高校根据自身所属地区以及高校校园规模的不同特点,开发出了各式各样的高校学生信息管理系统,将一些原有的高校传统管理模式转变成更为高效的信息系统管理模式,不仅有效提高了大量学生信息的收集管理效率,而且大大提高了高校信息管理系统的数据准确性和信息交流性<sup>[3]</sup>。

### 1.2.1 国内系统的发展现状

在国内,学生信息管理系统大致经历了三个显著时期。第一时期为数据单机处理时期,该时期也是学生信息管理系统发展的初始阶段。该阶段中所有学生的信息采用集中的方式管理与存储,能够在单机上实现较为简单的管理。该阶段的系统效率受限于计算机本身的性能以及集中式的结构<sup>[4]</sup>。第二时期为数据综合处理时期,随着计算机技术与信息技术的不断发展,计算机以及网络等硬件设备性能的提高,互联网的普及为学生信息管理系统的发展带来了新的机遇与挑战,人机交互界面更加清晰友好。但系统管理效率受限于较为昂贵的软硬件设备<sup>[5]</sup>。第三时期为数据系统处理时期。随着分布式技术和数据库技术等技术的不断发展,学生信息管理系统发展为客户机/服务器和浏览器/服务器两种模式的管理系统,这些技术的发展为学生信息管理系统的发展提供了充分的保障<sup>[6]</sup>。

整体来看,国内高校的信息化管理平台发展可以归结为几个阶段:单机模式、客户端-服务器模式和浏览器-服务器模式<sup>[7]</sup>。其中,单机模式是计算机刚引入高校时采用的管理模式,所有的信息集中存储在单台计算机上。单机模式的数据管理方法在信息管理过程中以串行方式处理进行,单点瓶颈问题较为明显,效率较低;同时数据修改更新容易出错;更重要的是所有数据几种存储的方式不能保证数据的可靠性,在单机损坏情况下极易丢失数据<sup>[8]</sup>。随着校园网与互联网的不断发展,单机模式已经不能满足高校发展的需求,客户端-服务器模式应运而出。相比于单机模式,客户端-服务器模式能够支持并行的数据管理,极大地提高了数据管理的效率<sup>[9]</sup>。同时客户端-服

器模式在使用过程中每台客户端都需要单独安装，给用户的使用造成了极大的不便。同时客户端-服务器模式需要同时开发客户端与服务器系统，开发成本较高。随着计算机性能不断提升，轻客户端模式：浏览器服务器模式应运而生。相比于客户端-服务器模式，浏览器-服务器模式允许所有的客户端采用计算机内置的浏览器即可完成数据的管理，极大地降低了使用成本与开发成本。因此，浏览器-服务器模式逐渐成为了当前高校信息化管理平台的主流模式。典型的信息化管理系统如清华大学开发的“UEAS 高校教务管理系统”、杭州正方软件股份有限公司的“正方高校现代教学管理信息系统”以及湖南强智科技有限公司的“强智综合教务管理系统”等<sup>[10]</sup>。

### 1.2.2 国外系统的发展现状

相比于国内的信息系统开发情况，国外的大中型软件开发公司经验更加丰富，技术更加成熟，具有较好的兼容性与可扩展性，系统能够随着学生规模的增大而自动扩展，结构方便调整，易于升级。因此，较好可扩展性的系统既能够满足当前的需求，同时也能够为后续的发展留有较大的空间。而在信息标准化和规范化的基础上，对信息进行合理的布局，使得相关人员可以按照各自的权限进行信息查询和维护。典型的信息管理系统包括日本广播协会信息管理系统，美国 IBM 公司推出的生产信息控制系统等<sup>[11]</sup>。其中广播协会信息管理系统在全球范围内得到了广泛的应用。总之，因其开发过程遵循的可扩展性、规范性等因素，使得系统能够获得有效的扩展性、安全性，系统的规模结构能够随着需求的不同自适应地改变，国外的信息管理技术相比国内的管理技术更为先进、成熟<sup>[12]</sup>。特别是一些教育发达的欧美国家，不仅开发出新的先进的管理系统，还设计研究出更为先进的学生管理办法与模式。世界上的各个发达国家对教育信息化高度重视，同时培养一批专业的技术团队以提供相应的技术支持。另外，发达国家的高校都拥有自己的高带宽校园网络，为信息化的管理平台提供了坚实的基础。实际上，高校信息化管理是国外很多高校已经进行多年探索的教育目标<sup>[13]</sup>。具体来讲，发达国家的高校在六七十年代已经形成了一套规范、成熟的数字化校园平台。目前美国已经有 85% 的高校能为学生提供在线课程目录服务、72% 的高校能够为学生提供在线课程注册系统。信息化管理平台在欧洲的各个大学也相继建立起来<sup>[14]</sup>。

## 1.3 主要研究内容

此次设计中，开发者以面向对象为设计理念，运用了信息工程学相关的理论，并详细研究了基于 Spring Boot 框架的高校学生信息管理软件系统。在具体编写代码时，开发者运用了 Java 编程语言及 IDEA 编译器，同时还使用了当前比较流行的开发框架，如 Mybatis、Spring Boot 等，研究内容分为以下几部分：

(1)对学生信息管理软件系统中要运用的各种核心技术展开详细研究,并结合当前主流的开发方法、开发理念,对此次研究课题进行全面分析。**Spring boot + Mybatis** 框架对整个系统的综合搭建,本质上来说就是一个基于 **MVC** 的系统框架,整个系统框架的各种代码彼此交互形成了松散且彼此耦合的框架结构,包括三层,分别是 **controller** 层、**model** 层以及 **view** 层。接着,构建了 **controller** 控制器与 **view** 视图、**model** 层与 **controller** 控制器交互的系统框架。随后,设计了系统的框架项目结构,按照系统应用环境方面的需求对系统框架项目、代码结构进行组件层的添加以及相关交互系统结构的综合搭建管理操作,项目还将初步采用一个基于 **maven** 的框架项目,管理整个系统框架。这是项目的一个框架生命周期。

(2)功能辅助分析。在设计时,对系统开发需要用到的各种核心技术开展针对性分析,并梳理系统的性能与功能,辅助分析系统对安全提出的相应要求,结合制图工具 **Visio** 手动绘制系统相关功能图表并进行相关辅助分析说明。

(3)结构总体规划。总体设计了系统功能模块结构以及整体架构等,并且从系统功能模块出发展开详细结构设计;通过系统后台展开实体数据库结构设计:对系统的物理结构进行综合设计,然后明确系统实体与物理实体的关联,利用 **Power Designer** 设计工具绘制设计出一个对应的系统逻辑模型和相应的物理模型;围绕系统安全实施过程验证,通过构建多重验证机制,使系统和数据有更高的安全性和可靠性。

(4)模块系统设计实现。通过使用开发方法及其工具,结合功能设计工具来准确实现系统的关键核心业务功能,并对管理模块其核心业务功能的实现进行具体描述。以及同时借助各种绘图处理工具,绘制与其相关联的图表资料来详细说明整个管理系统关键功能的过程。

(5)测试学生信息管理系统的系统代码,对系统的代码结构与功能进行测试,引入各种测试用例,阐述测试结果,按照测试结果,对系统的功能、性能及结构加以改进。

## 1.4 本文的组织结构

为了更加直观地阐述本研究论文的主要结构内容,现在将本研究论文的主要组织以及结构内容做简单性的介绍,具体内容如下:

第一章 论文绪论:主要内容是学生信息管理系统的研究背景、开发意义,并总结了当前研究情况,本文的研究组织理论结构以及本论文的重点研究课题内容等。

第二章 系统关键技术分析:具体分析该系统在设计及实现期间用到的核心技术及相关理论。包括了 **Web MVC** 设计模式, **Spring** 设计框架和 **Spring boot** 设计框架,以及持久层设计框架 **MyBatis** 等,还介绍了系统所涉及到的其他开发技术,包括视图表现层所使用的技术 **Free Marker** 等。

第三章 系统需求分析与设计：阐述了系统的总体开发需求，并提出如何设计及实现系统各项需求。包括系统功能体系结构设计等。而且，还具体设计了数据库。

第四章 系统设计实现与功能测试：主要对不同系统功能模块的详细结构设计等内容进行了介绍，并用结果图片的形式进行展示。同时，还重点介绍了本系统的测试工作及其测试结果。

第五章 展望总结与未开发展方向：主要内容是对本硕士论文的主要研究内容和目前取得的主要研究成果情况进行全面的展望总结，同时也阐述了学生信息管理系统存在的一些不足以及未来的改进方向和继续研究的展望。

## 1.5 本章小结

这一章重点阐述了系统开发的主要背景及意义，国内外的研究概况，分为国内和国外两个部分进行阐述，然后介绍了研究的重要内容，总结了研究结构和章节分布情况。

## 2 关键技术分析

在这一章，主要分析了学生信息管理系统的设计及实现期间需要用到的关键技术，如 web 开发模式、开发框架 Mybatis、Spring Boot 等相关技术及工具，还有表现层所使用的 Free Marker 技术，本章最后还介绍了为解决学生信息管理系统特有的高并发问题而采用的分布式架构技术。

### 2.1 Web MVC 模式

在软件开发行业，MVC 的应用极为常见，它是一种图形化设计模式，使开发工作的进度更加直观。MVC 把应用程序的架构划分成不相关的三个部分，这三个部分通常用于开发图形用户界面，它将信息的内部表示、信息面向用户的接收与表示相分离，如今，该模式在 PC、web 等领域的应用越来越常见。

MVC 由模型、视图和控制器三部分构成，同时模型层是关键部分，模型层对系统的业务逻辑与数据进行管理。而视图层则主要是提供展示数据、图表等的一种方式；具体表现形式有图表与视图。对比来说，控制器的作用是对视图向数据模型的转化进行控制，对输入的数据起到接收作用，然后转换成后台数据，生成数据模型或对象视图，三者的交互方式主要是数据模型从控制器获取到请求，然后收到用户的输入，当控制器在收到请求后，就将输入数据与对象进行交互，接着在视图界面输出数据模型，按照用户规定的格式，将数据展示给用户。Web MVC 的基本工作流程如图 2.1 所示：

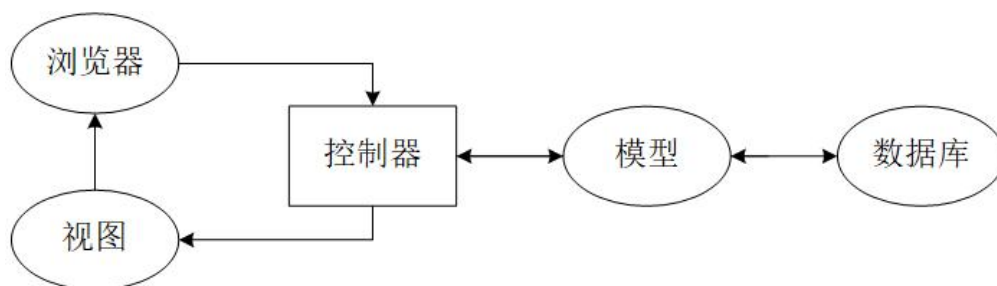


图 2.1 Web MVC 结构图

Web MVC 是 MVC 在传统 web 企业环境开发中的一个应用，当它被加载到新的软件并被移植升级到 Java 时，Java 开发人员已经开始大规模地使用并与 MVC 开发模式相结合，以至于后来的其他 Java 开发框架，如 Spring 中都延续了这种合作关系，在 Django 和 Rails 两个框架中，两者都特别强调了快速开发部署，使得 MVC 在除了传统的大型企业开发环境之外，也得到了越来越广泛的扩展应用。

Web MVC 基本工作流程：首先，控制器截获用户发出的请求；第二，控制器对模型实施调用，然后进行文件读写；第三，模型访问数据库；第四，数据库将数据提供给模型；第五，利用模型把数据传输到控制器上；第六，通过控制器将数据传输到视图；第七，完成视图的渲染，并将结果展示给用户。

MVC 模式一般会构建三层结构，而在层次结构中三级结构最为常见，详见下图。在该结构中，面向用户端的访问逻辑层和基于数据层的访问逻辑层之间分别插入了一个新的中间层，在中间层之后是数据表示层，最后就形成数据表示层、业务逻辑层及数据访问层。

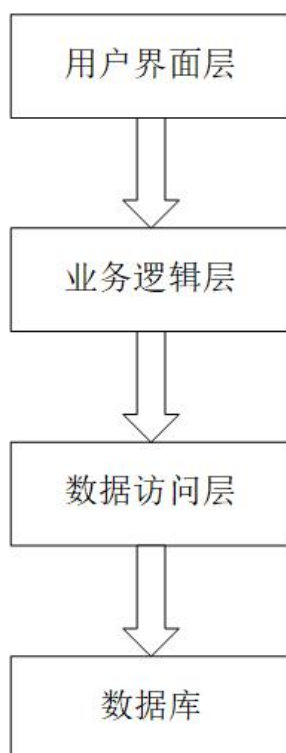


图 2.2 三层架构图

表示层(UI)，向所有用户提供显示的操作界面；其中，业务逻辑层主要是完成各项业务的处理；数据访问层是对数据库的所有信息进行封装处理，方便访问层与表示层进行访问。

具体而言，MVC 三级结构并未和三级结构系统直接联系在一起，而是因为其分层的目的是为了对系统的结构进行解耦。因此，通常将 MVC 和三层架构建立起联系。三级结构和 MVC 在各层次方面存在映射关系，详见下图 2.3 所示。MVC 的视图和控制层对应三级体系结构中的表示层。管理视图界面的显示，控制器控制从客户端到服务器端的数据交互，对应于表示层的服务连接。MVC 模型对应于三级架构中的业务逻辑层和数据访问层，数据可分成两种类型：一类是为业务逻辑提供数据服务；还有一类是数据库在访问时提供的相关服务。



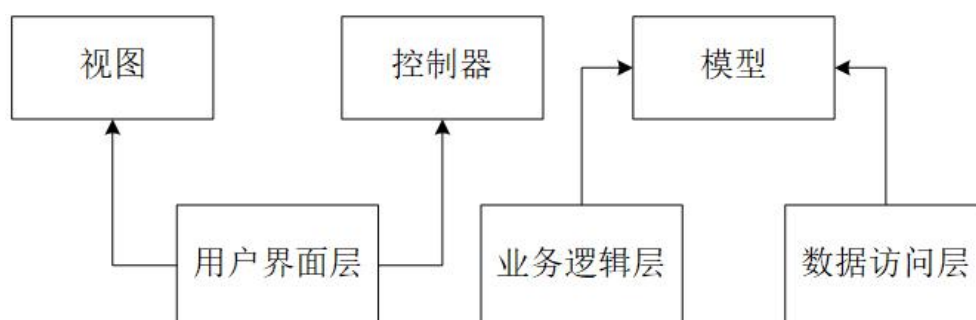


图 2.3 三层架构与 MVC 的层次映射关系图

MVC 模式广泛地被应用在各种 Java Web 框架体系中，按照开发功能，把所有代码分成不同部分，然后利用开发工具进行针对性开发，如此就显著提升了开发效率，使开发质量得到提升<sup>[15]</sup>。

## 2.2 Spring Boot 框架

### 2.2.1 Spring 框架

基于 Linux 系统，使用 Java 语言开发得到了 Spring 框架，针对文件管理，该框架提供了控制与反转的容器<sup>[16]</sup>。尽管该框架并未限制开发语言，但实际上在 Java 等程序中的应用极为常见，这也促使其经常流行于应用软件开发中。因此，它后来被用作 EJB 模型的替代品，或是相应补充。

在软件开发中，Spring 框架能够解决不少开发难题，如在面向切面开发过程中解决各种事务管理问题，集成多种安全可用的持续开发技术平台来有效管理其对数据库的访问等<sup>[17,18]</sup>。

该框架还提供基于软件反转(IOC)的控制功能，该控制功能可以更好地测试及维护各个项目，构建反射管理结构，为所有用户提供一致的项目管理办法与软件配置功能。例如在该框架当中，利用容器管理就能够自动扫描指定对象的特定类或 xml 文件等，并自动配置相应对象。因此，开发者就能够建立依赖文件的搜索机制，或是把依赖文件引入到搜索机制中，更快得到搜索结果<sup>[19]</sup>。

Spring 框架结构让开发者有一个可以面向不同切面的程序编程语言结构框架(AOP)，它让开发者可以基于代理配置以及运行程序时自动完成代理的配置，在具体开发中，面向切面开发时将模块进行调节，使其完成模块化操作。该框架仅仅给出面向切面开发的常见功能。虽然该框架比不上 AspectJ 框架，但是借助 Spring AOP 技术，它可以实现 Spring 框架中的事务管理和远程访问功能，能够满足编程基本需求。

Spring 的本地事务管理系统框架以 Java 平台为基础，能够提供事务的抽象处理功能，可以让全局事务、本地事务及其他处理事务共用保存点，而且此框架能够运行

于 Java 平台的各种开发环境。

Spring 系统集成了诸多事务管理模板，该系统能够利用 xml、管理模板对事务完成配置，同时还能够将数据保存、信息传递等功能集成在事务处理框架上。

Spring 的动态数据资源访问管理框架具有非常重要的作用，能够帮助开发者高效利用各种动态数据资源，而且还能够满足 Apache OJB、Hibernate 等各类事务管理框架的运行要求，同时还能够和 Spring 的数据事务管理共同使用，从而为各类数据访问事务提供相应数据。

起初，Spring 框架并没有计划建立 Web MVC 框架，但框架的开发人员认为表示层与需求层之间以及需求层与模型层的分离并不能解决所有的开发问题。对此，就开发了 Spring MVC<sup>[20]</sup>。

## 2.2.2 Spring Boot 框架

Spring Boot 的开发团队来自美国的 Pivotal 公司，是 2013 年研发的免费、轻量级、开源的系统框架，并在次年发布了首个版本，是基于 spring4.0 而开发的新框架，不仅具有原有的核心功能，同时还添加了更多新功能，对软件的配置进行了诸多简化等，对 Spring 框架进行一定程度的优化，促使软件系统的开发更为便捷，开发效率更高<sup>[21]</sup>。

而且，Spring Boot 搭载了很多其他框架版本，使各个版本存在的彼此冲突得到妥善解决，并有效处理了文件引用困难等各种问题。

Spring Boot 的使用特征主要包括：第一，可创建出独立的 Spring 应用程序，并且以其 Gradel 或 Maven 插件为基础，创建能够执行的 Wars 和 Jars；第二，内嵌 Jetty 或 Tomcat 等 Servlet 容器；第三，项目提供自动配置对象的“starter”。该项目使用对象对模型进行配置，使得配置得到了很大程度的简化，包括了 Maven 等；第四，尽可能对 Spring 容器进行自动配置；第五，软件特性主要包括了健壮性检查和指标配置等；第六，绝对没有生成冗余代码，无需 xml 配置<sup>[22]</sup>。

在框架中 Spring Boot 约定了重要的两个使用策略：约定价值大于配置和开箱之后即可使用。开箱之后即可使用的解决方法首先是通过向 Maven 依赖的 pom 文件中直接添加项目相关的依赖包文件。对项目管理的生命周期加以优化，接着利用对应的注释文件来代替复杂的 pom 项目配置文件。通过加入这一项功能，能够让开发者更加便捷地管理系统数据，不再需要进行各种功能配置，使开发者能够专注实现逻辑层。约定大于配置，作为一种新的设计范式，是根据 Spring Boot 约定加以配置的，要求开发者将约定信息加入到指定结构。尽管此设计会使框架灵活性减低，但也极大提升了对错误信息的检出率，使 xml 配置减少，同时可以自动编译源代码，自动完成代码测试等相关工作<sup>[23]</sup>。

在 Spring Boot 开发平台中，列出了一些前端开发模板的经典框架，以及各种网

站模板,阐述了模板开发技术,如 Thymeleaf 等,都是使用 Java 等编程语言完成开发的,对前端模板实施渲染处理,同时输出对应模板文本,把模板界面设计和应用逻辑相分离,此过程中前端开发运用 jQuery、Bootstrap 等,对于网页前端数据的传输文件,选取的格式为 Json,并给出 RESTful API。MVC 等框架可以在数据库进入服务器端后,对并发请求加以处理。前端的数据库对持久层的访问,也主要利用 JPA、MyBatis 等框架,对于后端的数据库,一般选取 SQL sever 与 MySQL,开发工具则使用的是 IDEA 工具<sup>[24]</sup>。

## 2.3 MyBatis 框架

在 Java 开发过程中,Mybatis 作为持久层的框架,可以利用 XML 文本或注解把 SQL 语句与对象实现耦合<sup>[25]</sup>。在 ibatis3.0 中,Mybatis 版本的开发及维护是一项关键内容,主要是含有 iBatis 原始代码及创建者的一个开发工作团队共同负责开发维护。

MyBatis 所具备的特点有:

(1) 使用传统数据对象的关系接口,映射结构框架是把 Java 对象及其结构加以映射,以此当作是数据库表,接着调用封装处理过的对象接口结构,对数据库表进行调用。Mybatis 语句把 Java 对象的接口结构及其映射当作 SQL 语句,对数据库操作起到调用作用,进而对 SQL 语句加以执行;

(2) MyBatis 能够为用户提供复杂数据的处理功能,如复杂数据查询等;在历史遗留系统内,MyBatis 能够使用各种规范化数据库,还可以使用其他基于 SQL 语句的系统,在这些系统中能够起到有效作用;

(3) MyBatis 可以做到编码比 JDBC 更加简便,只需一行声明编码就可以执行 SQL 语句,MyBatis 借助动态映射与声明代码,就能够将 SQL 执行语句与 Java 执行对象完成动态映射;而且 MyBatis 能够利用动态映射建立 SQL 语句;

(4) MyBatis 与 Spring 框架是 Google Guice 的功能集成,该框架能够建立新的业务系统,完成代码编写,不再对业务关系产生依赖;

(5) MyBatis 要求数据库具备缓存功能。利用特定语句对语义加以标记,便于数据库对数据实施检索,能够检索到高速缓存内的数据,不再需要检索数据库内的数据;

(6) MyBatis 还提供了以 Java 映射为基础的缓存,并配置了程序接口,为其他缓存提供连接功能,它还能够连接 OS Cache 等连接器。

在具体组成上,Mybatis 包括接口处理层、功能层、处理层等几部分,具体见图 2.4:

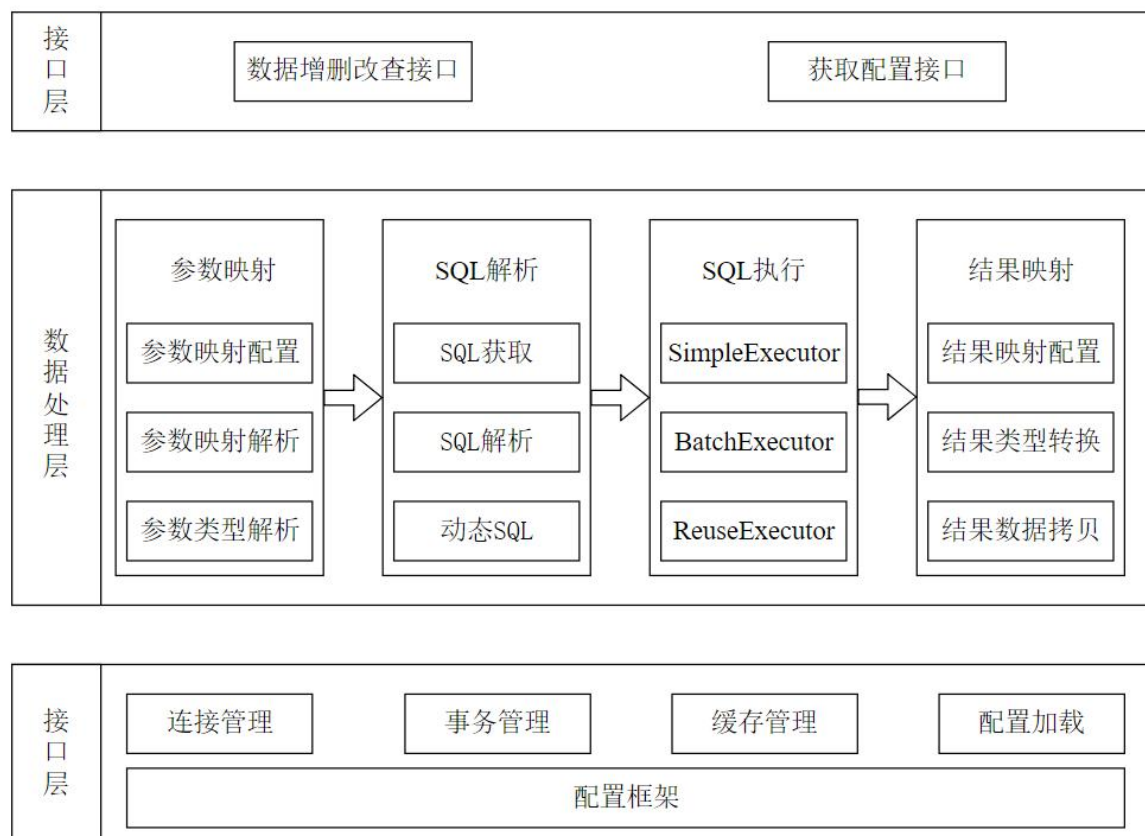


图 2.4 MyBatis 功能架构图

Mybatis 的基础功能为应用层提供各方面支持，利用处理层处理各项业务，如数据配置文件、数据加载等功能。处理层还提供自动检索、执行 SQL 语句，并根据 SQL 的执行结果将其映射至目标对象，在接口应用层，当进行外部开发时，能够提供本地应用程序中的接口，开发人员因此可以更加方便快捷地直接使用本地应用程序中的接口库，并运行应用数据库<sup>[26,27]</sup>。

Mybatis 在设计流程中主要进行如下操作：一是加载 xml 接口的配置文件，对接口实施初始化处理，进而生成接口的 SQL 执行对象；二是如果用户调用 Mybatis 给出的映射接口，则会按照 SQL 及其 CID 参数去实现对应接口和语句执行的对象，然后把等待处理的参数反馈到处理层；三是再次执行处理请求，构建连接，映射参数，解析 SQL 语句，执行 SQL，映射结果，连接释放，进而将处理结果提交<sup>[28]</sup>。

## 2.4 Free Marker 技术

Free Marker 是一个中文模板编码引擎，是一个基于中文模板引擎生成并以文本格式输出的中文通用编码工具，完全使用 Java 代码进行编写。Free Marker 的使用目的是生成 HTML Web 网站，特别是基于 MVC 的应用程序。Free Marker 具有一定的可编程性，但通常使用 Java 程序显示数据。Free Marker 不是一个 web 应用框架，但它更适合作为 web 应用框架的一部分，Free Marker 模板引擎也可以用于非 web 应用

环境, Free Marker 更适合于 model2 组件的显示, JSP 标签库也可以用于模板中, Free Marker 可以使用不同的文本、HTML、XML、RTF、Java 源代码等。Free Marker 易于手动安装, 易于自动开发, 无需安装服务器或其他开发环境; 本地插件也可以使用模板自动加载应用程序, 自动加载任何本地源代码, 如本地的源文件、数据库等。其开发过程是首先生成一个文本, 将其直接保存到本地电子文件, 并将其作为电子邮件形式发送, 然后将其从 web 文件应用程序包中发送并返回 web 应用浏览器。常用的指令格式有: include、if/elseif/elseif、循环结构等, Free Marker 自由标记不直接反映在 Java 对象中, Java 对象以插件方式封装, 在模板中显示变量, 以抽象的方式显示对象, 不受技术细节的干扰<sup>[29]</sup>。

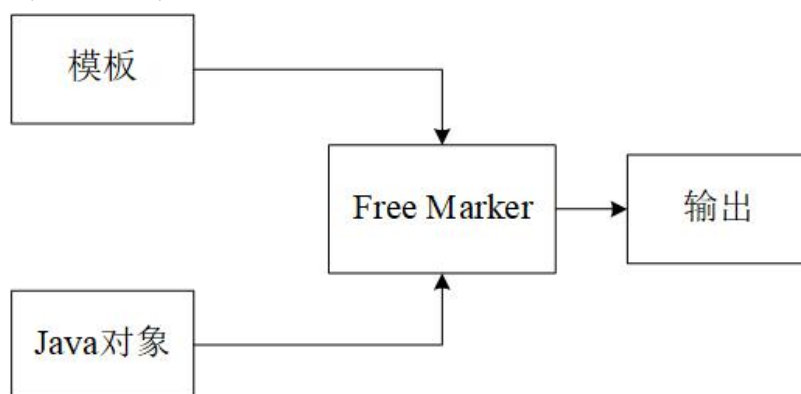


图 2.5 Free Marker 模板引擎示意图

这种工作模式通常被我们称为 MVC 模式, 在一个动态搜索网站中非常广泛被应用, 它有助于将网页内容设计工作人员和网站开发人员完全分离开来, 设计开发人员不必面对模板搜索引擎中复杂的代码逻辑, 如果没有其他程序员来重复编译或重新修改这些代码时, 其也完全可以自己改变一个页面的基本样式, 免费标记的最初设计是在 Web MVC 开发框架中生成 HTML 页面。这些页面即使连接不到服务连接器或 HTML 以及任何 web 相关组件, 也可以在非 web 应用程序环境中使用<sup>[30]</sup>。

Free Marker 是完全免费的, 基于 Apache 许可证版本 2.0。

## 2.5 分布式架构技术

高校学生信息管理系统在使用过程中, 常常会遇到高并发问题的困扰。因为在同一时刻大量学生访问系统进行操作, 会造成系统后台服务器和数据库不堪重负, 系统出现卡顿的现象, 严重时甚至会导致系统崩溃的情况。例如, 在学生选课的高峰阶段以及疫情期间采取的线上授课等使用方式, 这些情况都会加剧高并发问题的产生。因而在本次设计中采用分布式架构技术作为高并发问题的解决方案。其结构图如图 2.6 所示。

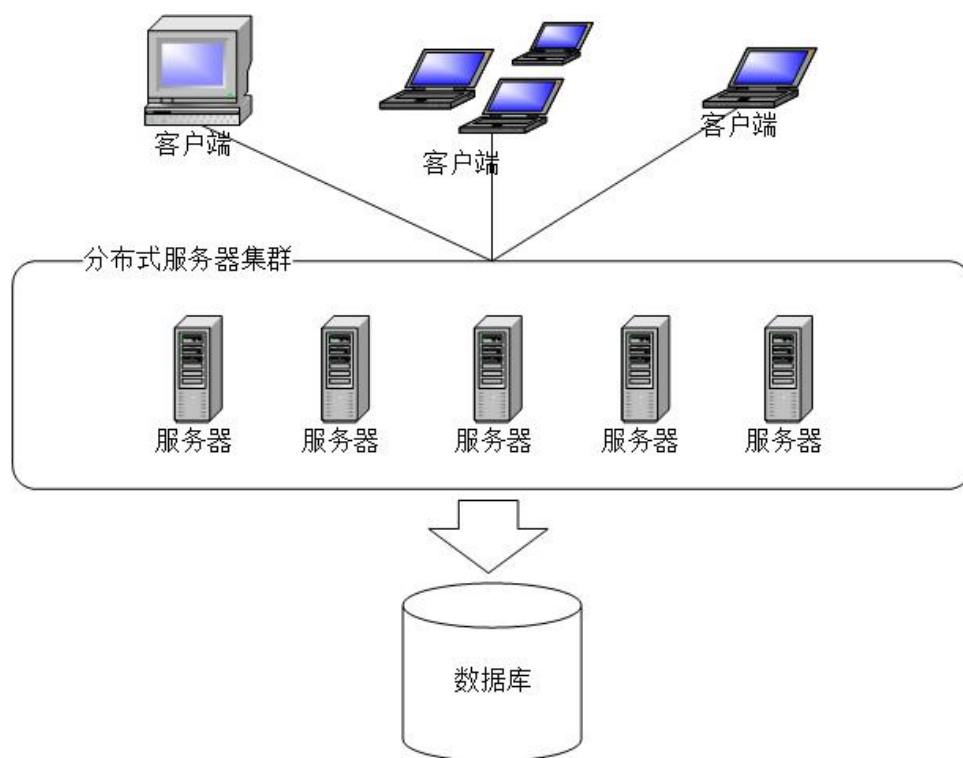


图 2.6 分布式系统结构图

分布式系统是建立在网络之上的软件系统，他有两个主要的特性，内聚性和透明性。内聚性是指每一个数据库分节点高度自治，有本地的数据库管理系统。透明性是指每一个数据库分节点对用户的应用来说都是透明的，看不出是本地调用还是远程访问。分布式系统作为一个整体对用户提供服务，而整个系统的内部则采用分工协作的方式<sup>[31]</sup>。通过 Zookeeper 消息中间件，将多台服务器进行连接，并对集群进行管理和监控以及负责消息在多台服务器中的传递。然后将不同的功能模块放置在不同的服务器，通过这种方式来扩大系统访问容量，提高系统稳定性。

随着网络流量的增加，单一架构已经无法满足高并发的请求，在引入分布式架构来实现系统后，多用户访问系统时会出现数据不一致以及系统不稳定等各种问题。为了解决这些问题，在系统中引入独立的中间件，显得格外需要。本次设计引入的是分布式协调服务 Zookeeper 中间件，Zookeeper 的存储结构采用的是树形存储，Zookeeper 提供了 Java 语言的 API 函数且能保证系统的高可用性和数据的一致性。因此 Zookeeper 往往被称为是协调分布式架构各系统的基础中间件。

## 2.6 本章小结

本章主要介绍了在高校学生信息管理系统的设计开发中需要采用的各种关键技术，包括了在 Java web 系统开发中需要采用的几种经典的 MVC 开发模式、主要框架构建技术 Spring Boot 和持久化框架 Mybatis 以及负责前端界面显示的技术 Free

**Marker**，这些主要技术与数据库 MySQL 共同构成了学生信息管理系统开发所需的关键技术，后续的系统实现将对系统相关技术进行详细的描述。本章最后还介绍了为解决学生信息管理系统特有的高并发问题而采用的分布式架构技术。



### 3 系统分析与设计

本章将深入浅出地阐述高校学生信息管理系统的架构分析。分别针对总体的核心功能和安全需求展开,对系统的核心功能展开架构分析和功能设计,并依据自身功能的迫切需求进行分析,包括系统个性化功能需求分析,以及系统的信息安全设计,包括数据库和系统管理的架构设计等重中之重的理论内容。

#### 3.1 系统需求分析

系统总体设计,项目软件开发中首当其冲的设计阶段就是需求分析,它的含义是指数据设计分析阶段的具体要求,在数据设计分析阶段中,包括系统软件设计、编码文件处理、测试和设备维护等各个阶段<sup>[32]</sup>。只有对系统的用户需求十分清楚了解,才能对其进行如实、详尽地描述,从而给出一些基本模块,并实现完备功能,进而可以对用户案例进行分析,这个设计可以将用户角色和系统有机地契合和关联起来,并对系统的性能和可靠性进行说明描述。

##### 3.1.1 系统需求概述

高校学生日常信息收集管理软件是针对高校学生大量业务信息处理工作而开发的信息管理软件,主要功能是用于加强学校广大学生日常信息的收集管理工作,总体目标也就是实现在校学生日常信息管理工作的系统化、科学化、规范化和自动化,其主要工作任务也就是用计算机对在校学生各种重要信息进行日常管理以及一些常规操作,如信息查询、修改、增加、删除等。

系统对外展示的功能应至少具备用户登陆、用户增加等,同时不管是作为管理员登陆还是普通用户登陆,都应具备基本的信息搜索和浏览功能。普通用户要查看个人信息首先需要登陆系统,拥有修改个人信息(密码、头像等)的权限,系统的管理员拥有更多的功能权限,能对系统的全部功能、信息展开管理,最重要的是具备对学生信息进行增加、修改和删除等功能。

系统后台管理人员,他们对系统的管理具有最高使用权限,对所有用户界面上能够显示出来的功能都具有操作的最高权限,他们具有普通用户不具备的管理功能,如可以直接对普通用户的信息实施修改或注销等操作,同时还能对违反规章制度较严重的在校学生信息进行自动屏蔽、删除的权限等,还可以针对学生信息系统展开实时管理。

高校学生业务信息管理系统具有显著的专业功能,但业务性能、系统安全性同样占有不可或缺的地位,高校学生的信息服务管理系统必须是一个经常被连续性



访问的管理系统，它的可靠性和安全性是必备的；系统操作使用的便捷性，要求系统在操作方面不繁琐、容易操作，以此能起到大大提高学生信息管理工作效率的重要作用；系统的可维护性，当学生信息管理系统的功能一旦有突发问题时，系统维护人员要能够迅速地、精准地定位和发现问题，找出问题的根本所在，及时解决相关问题；系统在学校得以普遍应用，其中包括学生、教师的个人私人信息，其安全性对学校而言也是十分重要的。

### 3.1.2 系统功能需求分析

在上文描述的系统需求功能中，我们可以从中得出高校学生信息管理软件系统的主要功能是具有多个方面的，下面我们从系统的大致使用情况对这六大功能模块加以描述和阐释：

（1）用户信息管理：进入系统主界面，用户进行登录之后，用户可以在个人信息设置中浏览、修改个人信息；用户还能查询其他信息、如学院信息等。

（2）学院信息管理：用户在学院管理模块中可以查看最新更新的学院名称和学院介绍以及学院在系统中的添加时间，用户可通过关键字对系统进行查询；用户可以对学院信息进行修改、增删等，并将所有学院信息进行分页显示。

（3）学院教师个人信息管理：用户进入学校系统的自主管理界面，并进入教师信息管理功能模块，用户可以随时查看系统现有的全部学校教师个人信息，包括所有教师注册编号、教师名称姓名、教师年龄性别、教师职业年龄、教师授课专业以及所有教师的相关个人信息介绍；并且用户可以通过教师名称信息的一个关键字快速查找相关教师个人信息，还能对其自由地修改、添加和删除；所有教师个人信息可以进行自动分页显示。

（4）在校学生相关个人信息管理：用户可以进入系统管理的个人自主信息管理界面，进入在校学生的系统功能模块，用户可以随时随地查看所有在校学生相关个人信息，包括学生学号、姓名、性别、年龄、出生年月、就读学科专业和所在学科班级，以及学生的入学时间等；用户可以通过学生姓名的关键字来查看学生信息是否存在；用户还能对其自由地修改、添加和删除；所有学生信息都进行分页展示。

（5）相关专业信息系统管理：用户进入信息系统的自主管理界面，进入专业信息管理模块，用户就可以实时查看所有相关专业信息，包括相关专业编号、专业名称、所属学院、专业介绍以及添加时间；用户可以通过专业名称的关键字对专业进行搜索查询；用户可以添加、删除和修改专业信息；所有专业信息进行分页显示。

（6）系统管理：用户进入自主管理界面，可以进行实时登录、修改密码和注销账户等操作；用户还能对系统后台信息进行实时数据备份记录，这些都为系统维护提供有力的支撑<sup>[33]</sup>。

### 3.1.3 系统用户及用例分析

从以上系统用户角色的详细分析中,我们能够得出该信息管理服务系统的主要服务群体角色包括两大类:第一大类是一般用户群体角色,也就是使用该系统的普通用户群体,其中主要包括在校教师、在读学生等;第二大类为系统管理员,也就是对信息管理系统进行管理以及维护的工作人员,通常由本校负责对系统进行管理以及维护的专业人员构成,其中教师主要承担审核以及下载该信息管理服务系统内容的工作任务,系统管理员的主要职责是对系统进行管理以及维护。大多数普通用户所执行的基本操作主要是搜索所需要的信息与浏览网页。

当系统确定了用户相应的实例角色,那么也就确定它所对应的具体实例分析内容:

#### (1) 管理员

系统管理员所拥有的操作权限比较全面,他能够实现对系统全部应用功能进行具体管理维护和操作,能够对系统功能的日常管理进行维护以及对系统相关信息进行增删或者调整等操作,他有权删除与网络设计规范不符的相关信息内容,并将其记载下来,以确保系统内部有关信息的安全性、健康性、可靠性和合法性<sup>[34]</sup>。根据上述实例分析过程,系统管理员作为用户活动者的实例框图结构如下图 3.1 所示。

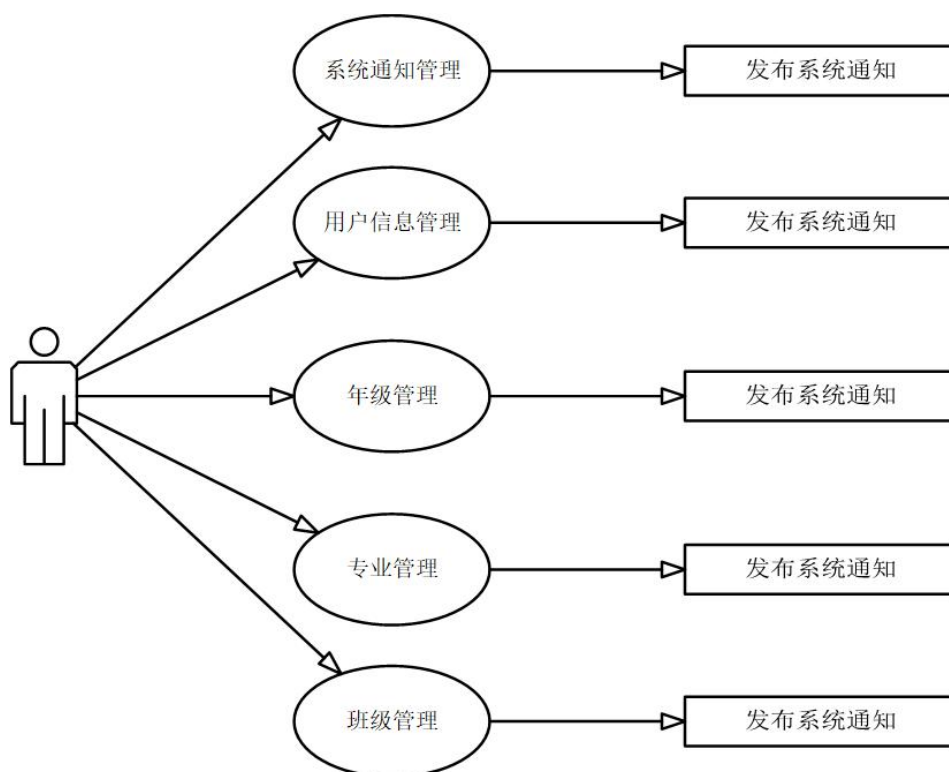


图 3.1 系统管理员的 UML 用例图

高级管理员用户实际用例说明如下:

第一, 系统公告管理: 向用户发布公告通知, 对系统发布公告情况进行管理。

第二, 菜单管理: 具有实时管理系统的所有模块的功能, 包括模块的增删操作和

排序等；对菜单的 URL 和 ICON 以及按钮和显示进行管理。

第三，系统角色管理：对系统的所有角色权限设置进行操作管理，包括所有角色的名称、状态、备注和添加时间；管理员可以赋予角色权限，同时也可以取消角色的权限。

第四，注册用户信息管理：主要对首次注册用户的基本登录信息进行实时管理；作为管理员用户可以随时查看登陆用户的基本身份信息，并且决定用户的使用状态，正常或冻结；用户还可以通过管理员来进行图像信息上传。

第五，日志管理：查看系统的操作者、操作内容以及操作时间；监控和拦截用户的异常操作和访问。

第六，数据备份：对系统的重要数据进行备份，保存存放路径，记录操作时间。

## (2) 普通用户

普通用户通常具备执行下列各项基本功能的权限。其中主要包括：上传图像、查看信息、用户登录、搜索关键字、发布信息等功能。主要活动者为普通用户的用例说明如下所示：

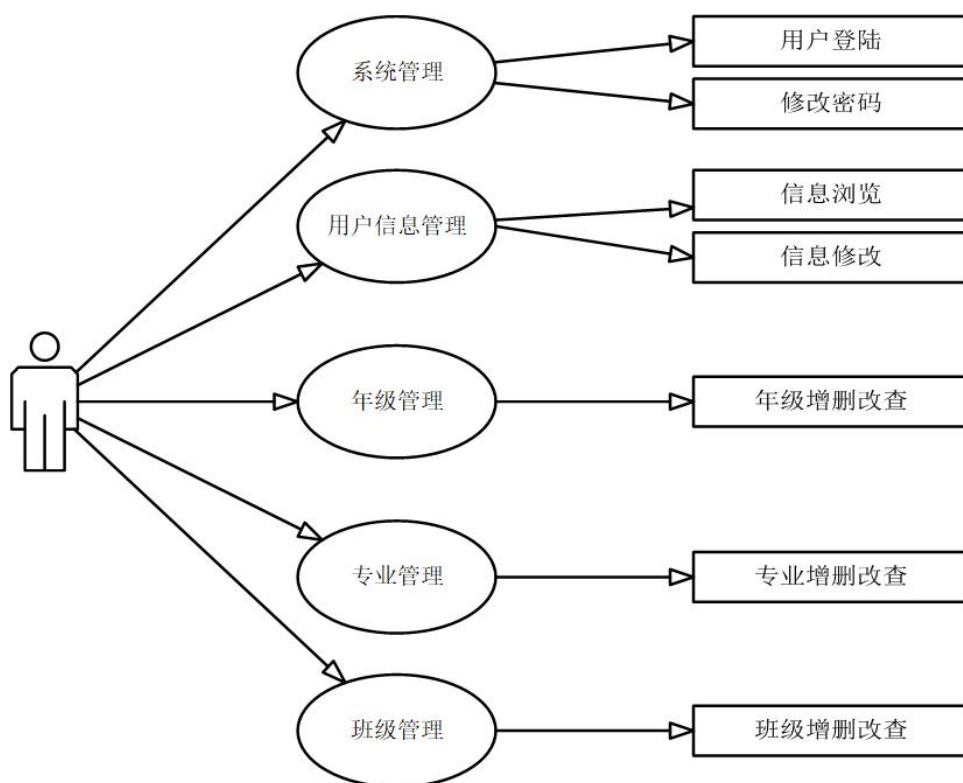


图 3.2 普通用户的 UML 用例图

第一，用户管理功能：普通用户能够实现执行权限内系统有关操作的功能，对于个人账号信息执行有关管理操作，例如注册用户私人账号，对登录密码进行修改，对已登录的私人账号密码进行检验等；

第二，管理用户个人信息功能。管理员能够对个人基础信息进行管理，用户能够实现修改、查询、登录系统等各项管理操作；

第三，年级管理：实现浏览、删除、编辑、修改等各项年级信息的功能；

第四，班级管理：管理班级操作，班级随年级的编辑而创建或直接引用，并且管理员可以查看所有班级信息；

第五，专业管理：对专业信息进行管理操作，可以查看所有的专业名称及任课老师信息，更新并修正所选修的专业信息。

### 3.1.4 系统性能需求分析

系统的软件性能指的是通过最少的系统资源实现最高的系统运行速度。从另外一个方面讲，系统的软件性能指的是各种属性的软件质量。所以在本章节中，分析系统各项性能需求也就是对系统的各项非功能性需求展开相应的分析。

非功能性需求在系统工程中指的是在特定情况下系统的特性需求以及运行情况达到某种特定或者是普遍的需要<sup>[35]</sup>。非功能需求通常包括两大类，一是发展质量，二是执行质量，执行质量指的是在系统运行过程中表现出来的易用性。发展质量指的是在系统软件开发过程中所表现出来的可测试性、可扩展性等<sup>[36,37]</sup>。

学生信息管理系统所面向的对象是全校的教师和学生，站在执行质量的角度，要求系统应该具备良好的可靠性、应用性等性能，站在发展质量的角度，系统应该具备可维护性以及可扩展性等性能。表 3.1 为系统性能需求说明统计表。

表 3.1 系统性能需求说明表

项目	说明
性能	系统在响应时间方面应达到 2/5/10 的常规标准：如果用户获得的响应时间低于 2 秒，那么系统的响应速度被认定为很快；如果用户获得的响应时间在 2-5 秒范围以内，系统的响应速度可被认定为能够接受；如果用户获得的响应时间在 5-10 秒范围以内，系统的响应速度可被认定为很慢，但仍处于能够被接受的区间内；如果用户在 10 秒以上，没有获得响应时，部分用户就会认为该系统性能很差，他们将认为目前的请求未获得响应。
易用性	系统的结构清楚，界面简单直观，所设计的导航科学合理，视觉层次明晰，版面设计合理，能够实现流畅地交互。
可靠性	在系统具体应用过程中，不会产生数据混乱丢失的现象，也不会发生系统崩溃的现象，不会产生界面乱码以及混乱等现象，也不会发生重大的安全性问题。
可扩展性	为了满足系统将来发展的功能需求，系统应能够对二次开发予以支持，系统代码和架构设计过程中应确保松耦合性，如果系统增设新的功能模块，应确保原有的代码及结构不会发生太大的调整。
可维护性	系统应该具备实现对数据进行备份以及恢复的功能，以确保系统出现故障后，

---

能够及时恢复运行。

---

### 3.1.5 系统安全需求分析

高校学生信息管理系统的数据库中涵盖了很多在校学生及教师的相关信息,因而系统的安全可靠性也十分重要,系统设计应达到下列安全要求:

- (1) 系统在操作数据库必须实现有效分隔,不能不通过后端处理就实现对数据库的直接操作,不能够只通过前端界面输入数据就可以实现访问数据库。
- (2) 应该严格检验所录入的数据,不经过检验的数据不能够完成提交。
- (3) 对于录入到系统中的数据,应该进行一定的处理和过滤。
- (4) 针对系统传输的数据,应采取特定的加密措施。
- (5) 对于访问系统的用户应采取相关有效的控制措施<sup>[38]</sup>。

## 3.2 系统总体设计

### 3.2.1 系统体系结构设计

该学生信息管理系统选取 B/S 体系结构作为系统的整体构架。B/S 体系结构主要由两部分构成,一是 Server(服务器端),二是 Browser(浏览器端),它是以 C/S 体系结构为基础,开发的网络模型结构<sup>[39]</sup>。该结构具备统一的客户端,并且体系结构通过浏览器完成设计,而且浏览器端重点显示并传送数据,它实现的业务逻辑很少,大部分业务逻辑通过服务器端完成。该结构下用户模块无需负担额外的客户端程序,进而使系统管理与开发人员的费用及工作量进一步下降,因而总费用也随之下降<sup>[40]</sup>。高校学生信息管理系统的系统架构图如下图 3.3 所示。项目按照系统架构划分主要包括下列内容:一是 MySQL 数据库服务器层,二是 Tomcat Web 服务器层,三是 IE/Chrome 浏览器层。其中 Web 服务器层又可以分为前端页面表示层,包括各种功能管理模块的业务逻辑层,以及包含操作数据库访问的 Mybatis 数据访问层。

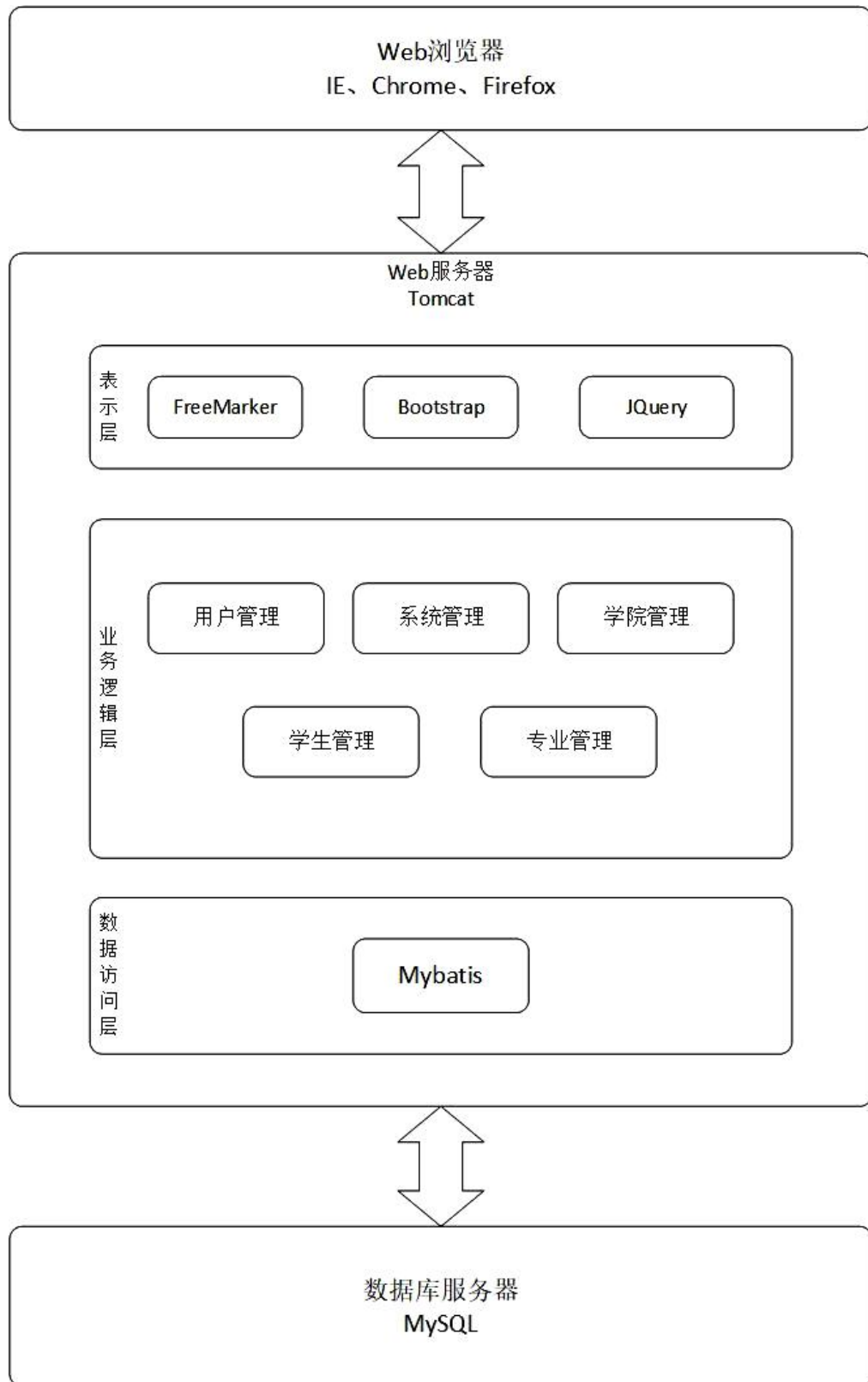


图 3.3 系统架构图

系统架构中与用户直接进行交互的层为 IE/Chrome 浏览器层，用户通过它对系统进行操作以及访问；系统依托 jQuery、Bootstrap 等设计界面与 Free Marker 模板引擎实现互动与显示功能。

在整个系统架构中尤其关键的层是 Tomcat Web 服务器层，浏览器层通过 Tomcat 实现各种功能，Tomcat 被集成后统一设置在 Web 容器中。B/S 体系结构又进一步被细分为三层结构：一是 DAL(数据访问层)，它的作用是对各项功能业务进行部署与设计，二是 BLL(业务逻辑层)，三是 UI(表示层)，整个系统主要实现下列功能：一是系统管理功能，二是实现管理学生信息的功能，三是实现管理专业信息的功能，四是实现管理学院信息的功能，五是实现管理用户信息的功能；

在整个系统架构中，处在底层的为 MySQL 数据库服务器层，该层主要提供 MySQL 数据库的数据查询与存储等有关业务，也是非常关键的一层。

### 3.2.2 系统功能结构设计

通过以上章节我们能够看出该系统的总体功能架构，其主要由下列重要功能模块构成：首先是综合管理专业、学院等个人信息的功能管理模块，其次是互动管理模块，它能够实现系统与用户彼此的互动，最后是系统的功能管理模块，详见图 3.4。各模块功能构架详见其设计部分。

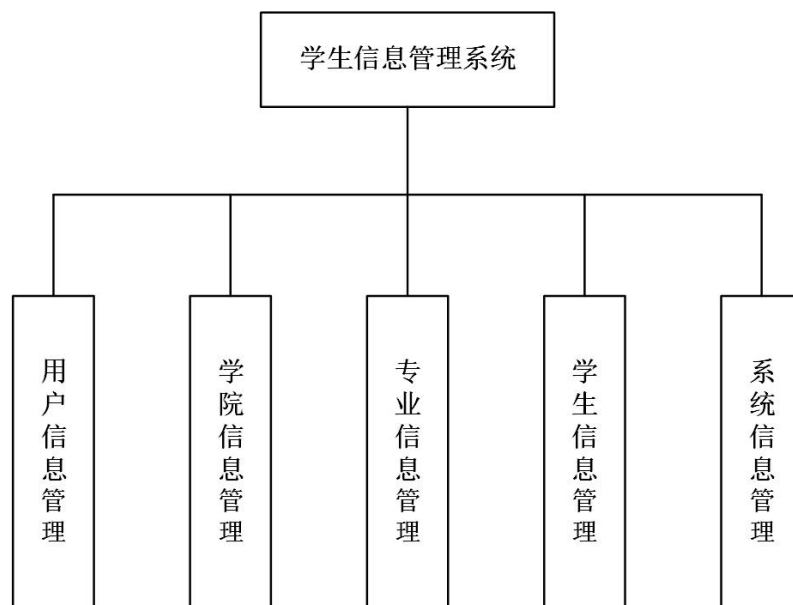


图 3.4 系统总体功能结构图

## 3.3 系统功能模块设计

高校学生信息管理系统由五大管理模块构成：一是学院管理功能模块、二是学生管理功能模块、三是用户信息管理功能模块、四是系统管理功能模块、五是专业管理功能模块。以下分别详细阐述这五大功能管理模块所实现的相应功能。

### 3.3.1 用户信息管理模块

本模块重点用于实现系统的用户信息管理功能：一是查看用户信息；二是修改用户信息；三是搜索用户信息；四是添加用户信息；五是上传和修改用户图像，图 3.5 为本模块结构示意图。

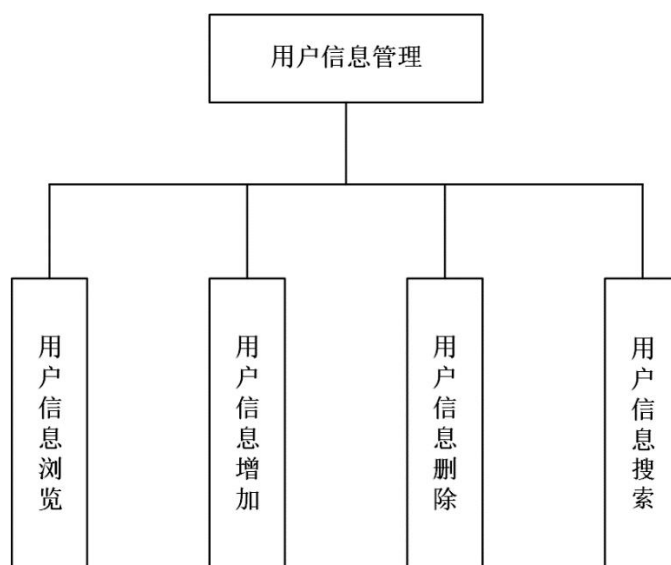


图 3.5 用户信息管理模块的功能结构图

#### （1）用户信息浏览

用户登录系统打开本模块即可对个人信息进行查看和浏览。其中需要从数据库中按照所需要的条件来搜索和查看用户基本信息，且可以根据页面提示完成所需要的操作。

#### （2）用户信息修改

该功能可实现用户修改个人基本信息与主页信息。在实现这一修改功能时需要在数据库中提取前期存储的用户信息，再根据用户的要求来把修改后的信息进行输入，再刷新该修改界面即可。

#### （3）用户信息搜索

管理员登录后台后，在此页面可对完成注册的用户名进行模糊查询，以便于管理员管理用户。其中需要从数据库中按照给定条件来提取所要查找的用户信息，再把相关匹配结果显示在页面上。

#### （4）用户信息添加

用户登陆系统后通过本页面，可以通过点击添加按钮并填写相关的用户注册信息，完成用户信息的添加功能。这里主要涉及到的是将前端页面的输入信息，在数据库中实现信息的添加和更新，然后刷新本界面即可实现添加用户信息。

用户信息的浏览和搜索总是联系在一起，而修改和添加功能只能由管理员实现。这里仅以用户信息修改作为说明，用户基本信息中主要涉及的是用户密码、图像、描述、邮箱和姓名等信息。在设计时，考虑到把密码与用户隐私相关的信息和图像类非字符串等信息进行了界定，按照字符串格式是否属于隐私信息来进行划分，把用户基



本信息区别为三种，第一种是密码；第二种是图像；第三种是姓名、邮箱及描述。以上用户信息采用了统一的修改流程，因此本文把它们概括成为一种。用户信息修改流程图如图 3.6 所示。

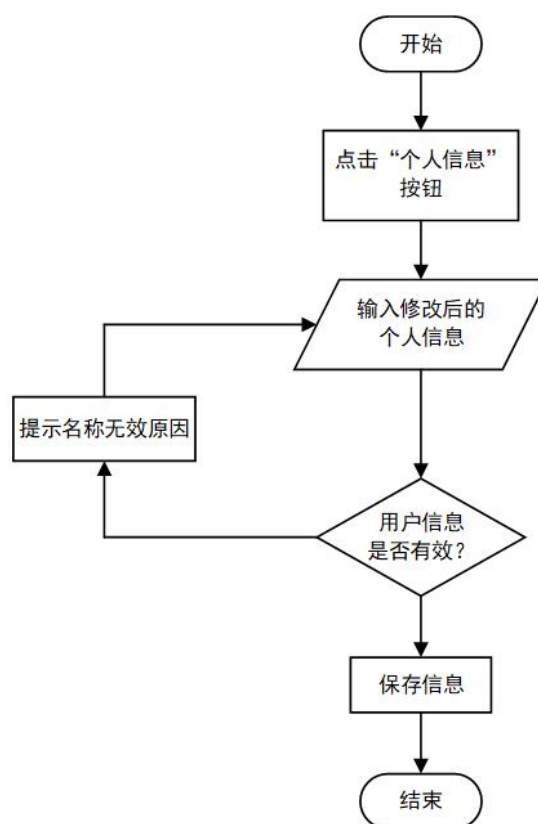


图 3.6 用户信息修改流程设计图

### 3.3.2 学院管理模块

本模块负责实现管理学院信息，其中可以实现以下功能：一是添加学院信息；二是浏览学院信息；三是修改学院信息；四是删除学院信息；五是搜索学院信息。

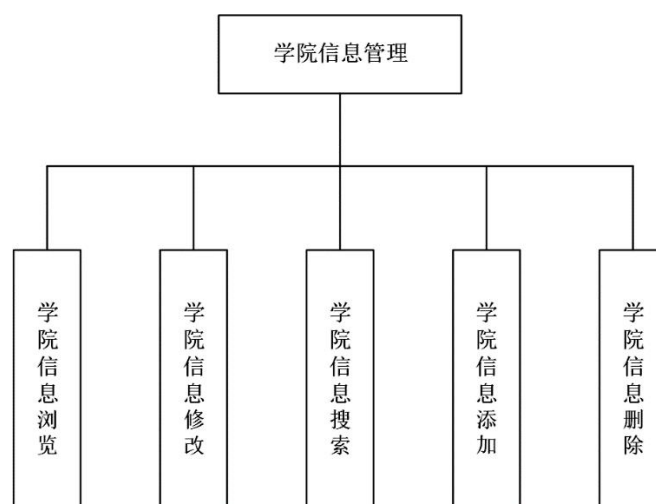


图 3.7 学院管理模块的功能结构设计图

### （1）学院信息增加

用户登录系统即可通过本模块来添加学院信息，通过填写学院名称和学院介绍，系统自动补齐学院编号和添加时间，来完成增加操作。这里主要涉及到的是学院信息的创建，以及将学院信息存储到数据库中，并且刷新用户访问界面等操作。

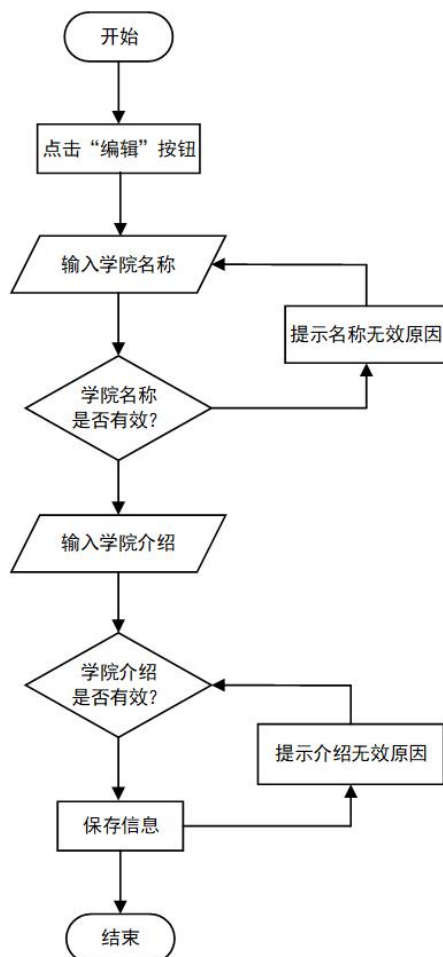


图 3.7 学院信息添加流程设计图

### （2）学院信息浏览

此功能方便用户查看自己的学院信息，包括学院介绍和学院创建时间，并通过分页展示数据，控制页面信息数量。其中需要从数据库存储的数据中按条件查找出学院信息，查找出来的学院基本信息会显示在界面上。

### （3）学院信息修改

用户登录后，通过本功能就可以修改学院信息内容与格式，并可对相关权限进行调整。该功能同样需要通过数据库查找出学院信息，用户再把完成修改的学院信息输入后，数据库就会更新数据，同时把修改后的学院信息和与其关联的最新信息显示在界面上。

### （4）学院信息删除

用户登录后，通过本功能就可以删除学院信息，管理员对用户学院信息进行审核，

如果用户的学院信息不满足规定标准，管理员拥有直接删除此信息的操作权。该功能同样需要通过数据库查找出学院信息和与其关联的其他信息，选中后再点击删除，刷新当前操作界面即可删除学院信息及与其关联的信息。

### （5）学院信息搜索

通过本项功能可以实现用户模糊查询学院名称和相关学院。其中利用给定的条件在数据库中查找到学院信息，匹配结果即会显示在界面上。图 3.7 为学院模块结构示意图。

学院信息修改与学院信息添加仅多设置了一个条件，即需要确定是否存在该学院，其中也包含学院原有信息的回显功能，后面的操作和创建学院基本一致，因为浏览和搜索学院信息两种操作关系密切，所以在删除时仅需要确定有无要删除的学院信息，同时确定该学院信息匹配操作者就可以完成删除操作。

## 3.3.3 专业管理模块

本模块可以实现增删改查专业信息，其重点功能包括：一是添加专业信息；二是查看专业信息；三是修改专业信息；四是删除专业信息；五是搜索专业信息等。管理员拥有以上所有功能；但普通用户只拥有查看和搜索专业信息的权限，在创建专业时必须增加所属学院。专业管理模块的功能结构图如图 3.9 所示。

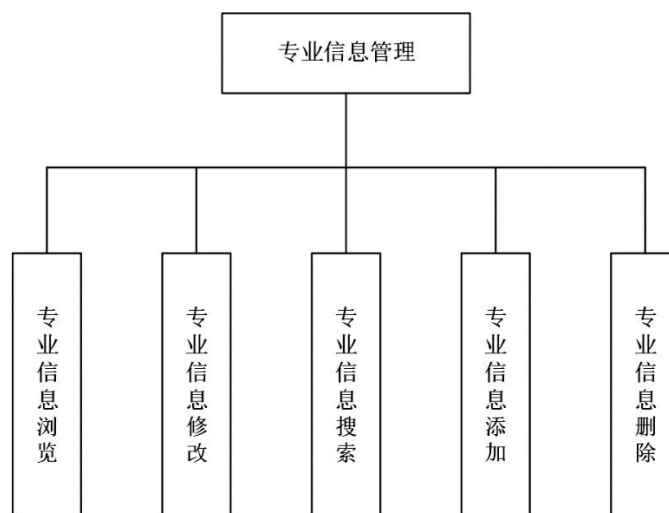


图 3.8 专业管理模块功能结构设计图

### （1）专业信息增加

此功能方便用户在登录之后创建新的专业信息，并通过所属学院与学院管理相联系。其中重点是实现专业信息设计并创建，在数据库中储存新的专业信息，同时进行专业信息界面的刷新即可实现添加专业信息的操作。本文对创建专业的流程进行单独介绍，图 3.10 为专业信息创建流程示意图。

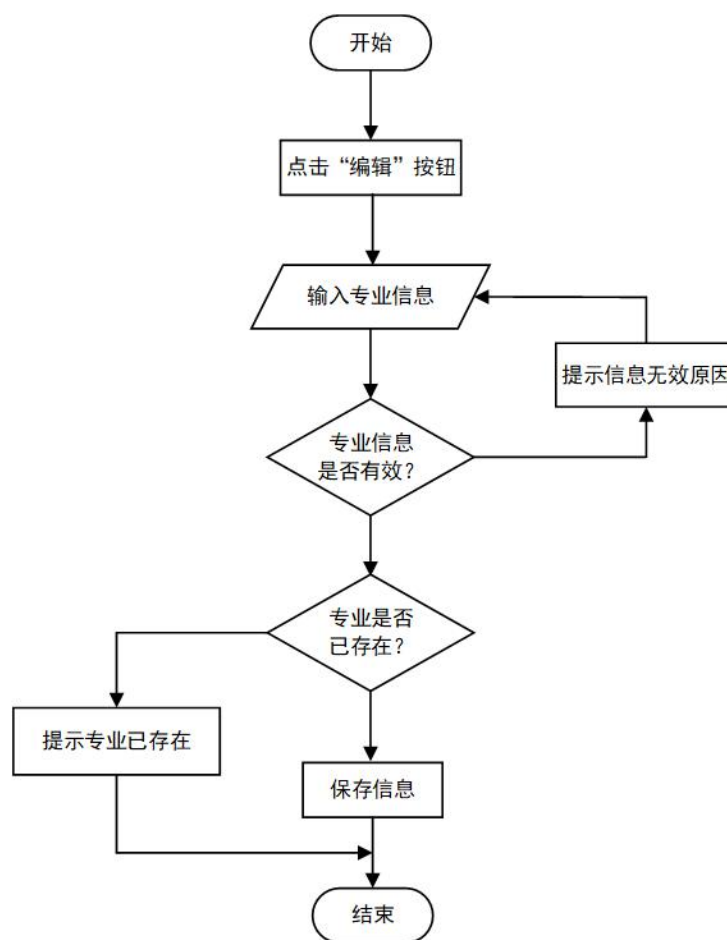


图 3.9 专业信息增加功能的流程图

## (2) 专业信息浏览

此功能可以让用户对本专业信息进行查看，同时可以查看与其关联的学院信息。其中要通过数据库，按照给定条件来查看专业信息，同时把查看结果显示在页面上。

## (3) 专业信息修改

本功能主要是操作者登录后，可实施修改专业信息的相关操作。其中需要通过从数据库中查找出所要修改的专业信息，再按照操作者输入修改的信息更新数据，同时对当前显示页面进行刷新即可实现修改功能。

## (4) 专业信息删除

管理员或用户登录后，本功能可以实现删除已有专业；管理员负责对用户专业进行审核，如果专业信息不匹配学生信息，管理员拥有对此用户专业与其关联信息进行直接删除的管理权。其中需要完成检查相关专业信息的操作，再删除数据库中存储的该专业信息，同时对当前显示页面进行刷新就可执行删除专业信息的操作。

## (5) 专业信息搜索

用户通过本功能可以搜索查看专业名称与其学院信息。其中需要通过数据库按条件提取专业信息与其学院信息，同时当前页面上显示相关查询结果。

普通用户在创建专业时必须依赖于专业归属的学院来完成,因为专业和学院两个管理模块联系紧密;查找和查看专业必须依赖于专业名称对数据库进行搜索与查询;所以修改和删除专业时均必须同步操作修改或删除其所属学院信息。

### 3.3.4 学生管理模块

本模块重点负责管理学生基本信息,这些基本信息主要涉及学生年龄、性别、名字、学号、出生年月、所属专业、所属班级以及入学时间等信息;还包括对学生信息进行添加、删除、编辑和查看详情的操作;还可以通过学生姓名对学生信息进行搜索的功能。

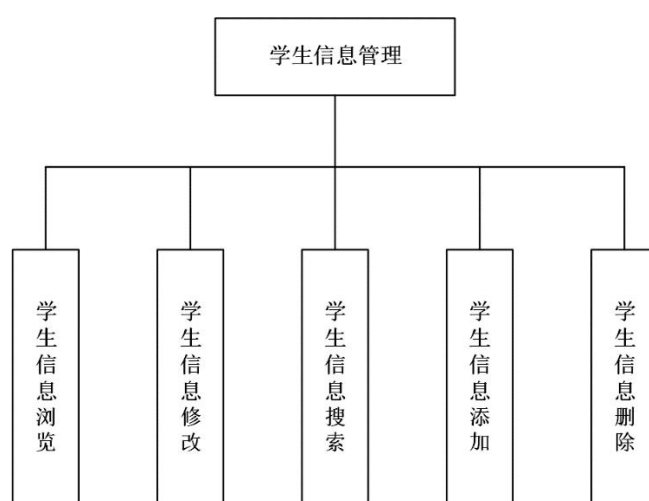


图 3.10 学生管理模块的功能结构图

该模块与专业管理模块相似,其功能结构图如图 3.11 所示。

#### (1) 学生信息增加

管理员或教师可以利用此项功能来实现新生信息的添加,此功能与专业管理模块的增加功能一样,区别在于学生信息不仅与专业信息相关联,还与班级和学院信息相联系。这里主要涉及的是将新生信息储存在数据库里,同时对当前显示界面进行刷新操作即可实现新生信息添加功能。

#### (2) 学生信息浏览

用户通过本功能可对个人信息进行查看,管理员也可利用本功能来管理学生信息,可编辑整理此类信息和各专业与学院班级包含的学生数量信息。其中主要涉及的是从数据库中根据条件获取学生信息,同时实现在当前页面上显示学生信息等操作。

#### (3) 学生信息修改

管理员或用户可利用本功能来修改学生信息。在操作时要按照条件在数据库里提取学生信息,再按照用户所输入的新信息来更新数据,再对当前显示页面进行刷新即可完成学生信息的修改操作。

#### （4）学生信息删除

管理员或教师可通过此功能，在系统中对已经毕业的学生信息进行整理和删除操作；对一些用户错误信息进行更正，管理员拥有对学生用户与其关联信息进行直接删除的管理权。其中要完成检查学生用户与其关联信息的操作，再删除数据库中存储的该学生信息，同时对当前学生浏览页面进行刷新操作即可实现本功能。

#### （5）学生信息搜索

用户利用本功能可以搜索查看学生信息与所属专业及院校，用户利用学生名称在数据库中提取学生信息与其所学专业及所在学院等相关信息，同时当前页面上显示查询搜索结果。

这五种操作的流程和专业管理模块、学院管理模块基本一致，所以本文只选择其中的修改学生信息时的实际操作流程加以具体描述，具体可见图 3.12：

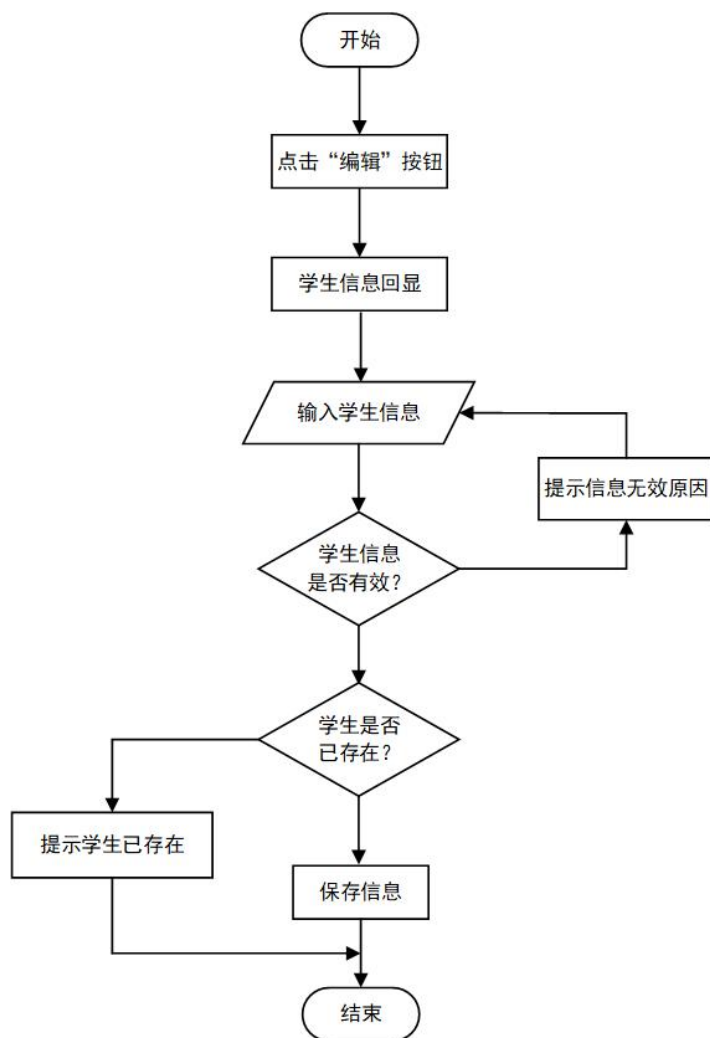


图 3.11 学生信息修改功能的流程图

### 3.3.5 系统管理模块

本模块重点负责实现用户的各种操作，其中主要可以实现以下功能：一是用户注册；二是用户登陆；三是修改密码；四是用户注销等。上述操作都是系统使用中高频发生的，甚至有些操作还涉及到大量用户同时进行引起的高并发问题，这些都需要系统做出相应的处理，以保证系统稳定性。在系统使用中，管理模块也是高频使用的一个功能模块。另外，登陆界面的美观度也十分重要。因为系统登陆界面充当着整个系统的门户。负责给予操作使用者对整个系统的第一印象。关于管理模块基本功能结构情况具体可见图 3.13：

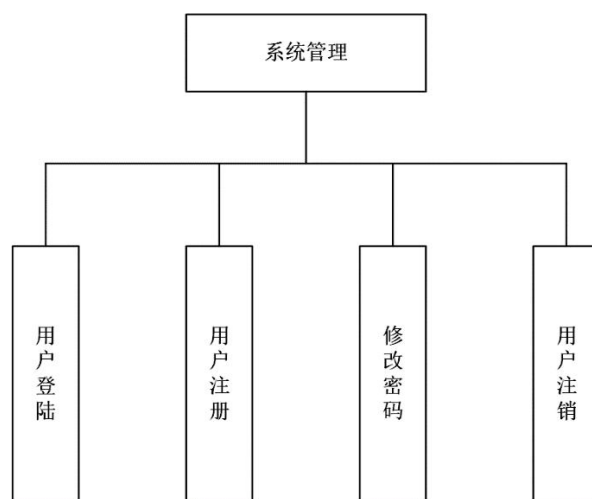


图 3.12 系统管理模块的功能结构设计图

### （1）用户注册

此功能主要实现用户注册，非本校人员不能进行注册。例如新同学和新教职员工的加入，需要增加他们的注册信息，做到及时更新高校管理系统人员名单。其中需要通过按照条件自数据库中搜索出用户对象信息，如果数据库中并未搜索到该用户注册信息时，就可以进行新用户的创建，同时在数据库中更新用户信息，再对当前用户界面进行刷新即可实现新用户的创建操作。

### （2）用户登录

本功能能够满足用户登录需求，用户登录时输入正确的用户名和密码即可登录访问系统。其中需要按照条件自数据库里提取到用户提前设置的登陆信息，并在检验用户提交信息的正确性后，即可完成页面跳转，同时显示登录成功等操作。

### （3）修改密码

用户成功访问系统后即可通过操作实现本功能，用户登录后进入密码修改界面，用密码的正确格式修改原密码，完成保存后再点击退出。其中在操作时要求按照用户信息条件自数据库里提取用户原始密码信息，系统会显示用户修改密码页面，检查用户提交的新密码格式是否满足要求，满足要求后数据库即会存储用户更改后的新密码，再返回登录页面，即完成用户密码修改操作。否则系统会提示用户输入正确密码格式。



#### (4) 用户注销

用户可以通过本功能实现帐户注销和删除用户信息功能。学生在毕业后通过本功能可以把个人账号注销并删除个人基本信息,也可以把个人信息打包后存储在系统中。其中实现本功能需要以下操作,即自数据库里提取用户账户信息,再选择删除账户与其关联信息,删除成功后系统即会返回首页。

用户必须在系统成功注册后,才可以进行登录、密码修改和注销帐号等操作,因此用户使用 ID 功能的必要前提是完成系统注册,以下详细说明用户注册时需要完成的实际操作流程,具体可见图 3.14:

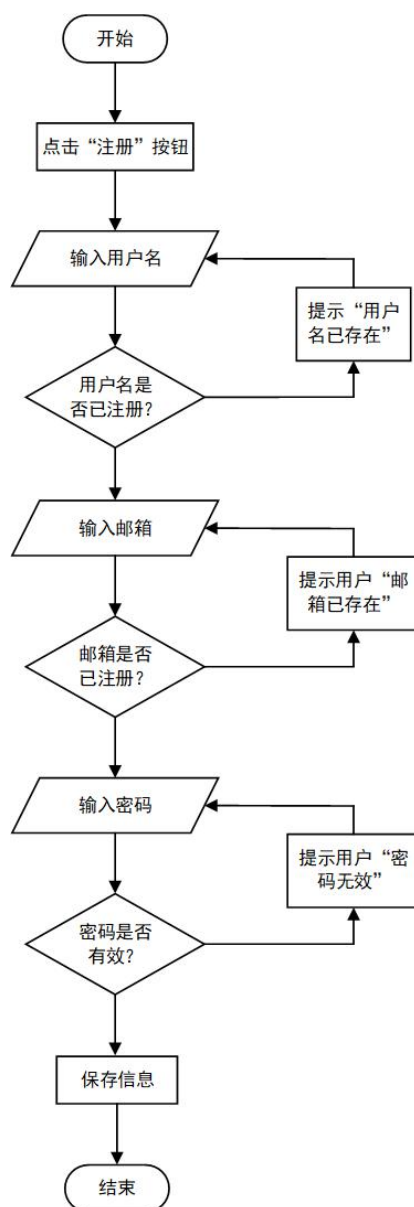


图 3.13 用户注册功能的流程图



### 3.4 系统数据库设计

对于开发一个系统时需要完成相应的数据库设计,该设计主要可划分成四个环节:一是完成需求分析;二是实现概念设计;三是完成逻辑设计;四是实现物理设计。其中本章的 3.1、3.2 小节具体介绍了需求分析,这里不再赘述。而概念设计环节要求按照需求分析结果来构建出单独的概念模型;第三个环节是完成逻辑设计,此时主要的工作是把概念模型向指定数据库的数据模型结构进行转换;然后就可以进行第四个环节的物理设计,此时要按照数据库对应的语言来构建出合理的物理存储结构,并创建出匹配的数据库与表<sup>[4]</sup>。系统在构建数据模型时必须根据数据库的 Power Designer 设计工具来完成;利用需求分析结果作为模块关系构建的依据,以确定各功能模块属性,构建出它们的关系模型;再通过 Power Designer 来建立物理模型并生成逻辑模型;下一步就可以生成 SQL 执行文件,并获得匹配数据库的表结构。

#### 3.4.1 数据库建模说明

按照前 3.1-3.3 节所作的需求分析可知,本系统主要包含三个实体,一是用户;二是学院;三是各专业;同时也包括相关系统实体:一是教师;二是学生;三是年级;四是班级;五是日志;六是菜单。各实体间存在四种关系:一是一对一;二是一对多;三是多对一;四是多对多。

(1) 以下将对三个实体间的关系进行具体介绍:

用户和学院存在一对多关系:即一个学院同时可存在多个用户,但一个用户只能有一个学院;学院与专业的关系为一对多:一个学院可以拥有多个专业,而一个专业只能属于一个学院;专业与用户的关系为一对多:一个专业可以拥有多个用户,而一个用户只能有一个专业(管理员和教师除外)。

(2) 然后说明用户实体与其他系统实体的互动操作关系:

教师:用户与教师属于一一对应关系;学生:用户与学生的关系也是属于一一对应;日志:一个用户能发表日志一篇,但多个用户均可查询该篇日志。

(3) 以下阐述学院和系统两个实体间的互动关系:

年级:一个学院可以有多个年级,但是一个年级只能属于一个学院;班级:一个学院可以有多个班级,一个班级只能属于一个学院。

(4) 最后说明专业实体与系统实体的互动操作关系:

教师:一个专业由一名教师教授,但一个教师可以教授多个专业;学生:一个专业有多名学生,但是一个学生只能属于一个专业。

#### 3.4.2 数据库表结构说明

在系统设计与实现中,开发一个数据库是必不可少的,设计数据库主要是为了方

便系统运行时的数据和信息共享,前端用户和管理员用户之间可以同时访问数据库中的数据,实现信息共享。MySQL 数据库是设计开发中经常被用到的数据库,根据输入和输出数据的要求进行分析,确定表格与结构之间的关系,验证、调整和完善查询和浏览的数据过程。从数据库的特点出发,分析数据库的概念模型,采用合适的转换和构建方法进行处理。按照以上介绍可知,只有用户和学院存在一对多关系,这时会有一个额外的关系表生成,而其他的实体均会仅有一张表生成,不过其中有些表的关系也有一对多的这种关联情况,因此表中会生成对应的外键。系统累计生成 9 张表,系统逻辑模型和所采用的数据库类型参数是 Power Designer 工具生成物理模型的依据,再生成系统的 SQL 执行文件,同时通过基于 SQLyog13.1.5 工具把对应的表生成在 MySQL 数据库中。数据库表之间的关联关系如图 3.15 所示。

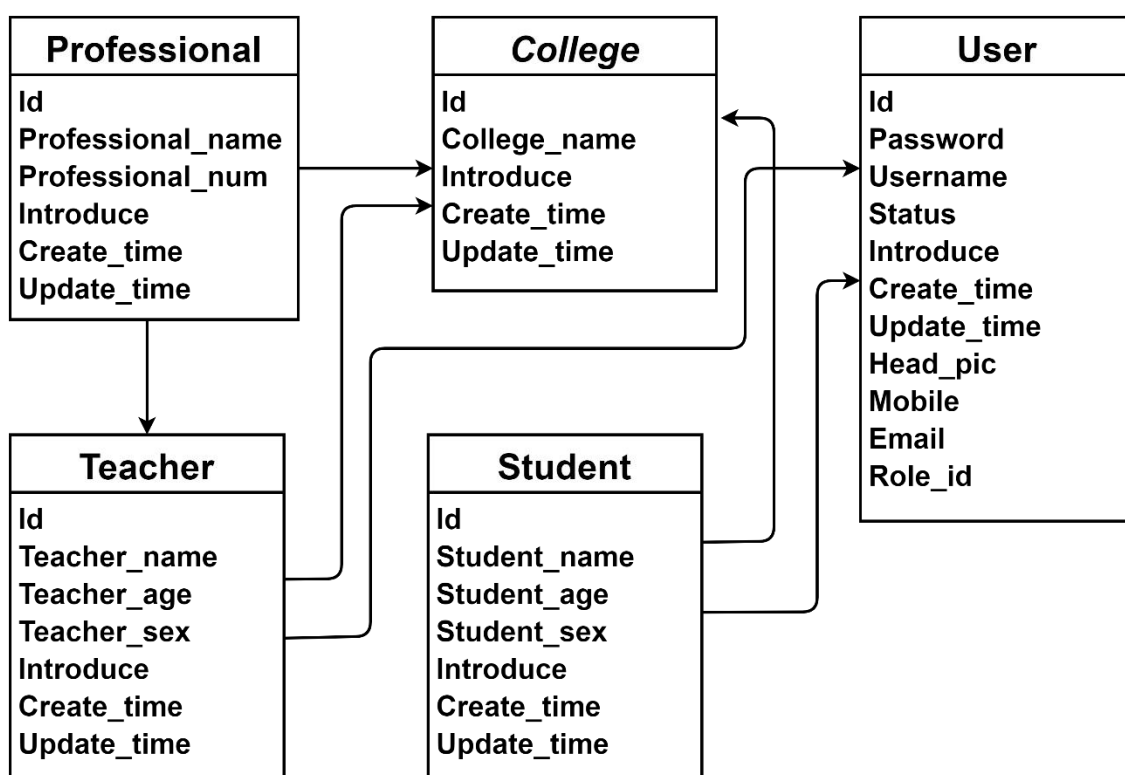


图 3.14 数据库表之间的关联图

### (1) 用户表

其中主要是针对用户相关信息进行存储,具体可见表 3.2:

表 3.2 用户表

字段名称	数据类型及长度	是否主键	备注
YonghuID	Int(20)	是	用户 ID
Chuangjianshijian	Datetime	否	创建时间
Gengxinshijian	Datetime	否	更新时间
Youxiang	Varchar(32)	否	用户邮箱

Touxiang	Varchar(128)	否	用户头像
Lianxidiahua	Varchar(12)	否	联系电话
Denglumima	Varchar(11)	否	登陆密码
Xingbie	Int(11)	否	用户性别
Zhuangtai	Int(11)	否	当前状态
Yonghuming	Varchar(18)	否	用户名

## (2) 学院表

其中主要是针对学院相关信息进行存储，具体可见表 3.3:

表 3.3 学院表

字段名称	数据类型及长度	是否主键	备注
XueyuanID	Int(20)	是	学院 ID
Chuangjianshijian	Datetime	否	创建时间
Gengxinshijian	Datetime	否	更新时间
Mingcheng	Varchar(18)	否	学院名称
Jieshao	Varchar(128)	否	学院介绍

## (3) 专业表

其中主要是针对专业相关信息进行存储，包括专业介绍和专业编号属性，外加一个专业所属学院 ID 作为外键，具体可见表 3.4:

表 3.4 专业表

字段名称	数据类型及长度	是否主键	备注
ZhuanyeID	Int(20)	是	专业 ID
Chuangjianshijian	Datetime	否	创建时间
Gengxinshijian	Datetime	否	更新时间
Jieshao	Varchar(255)	否	专业介绍
Mingcheng	Varchar(6)	否	专业名称
Zhuanyebianhao	Varchar(4)	否	专业编号
SuoshuxueyuanID	Int(20)	否	专业所属学院 ID

## (4) 教师表

教师表包含教师 ID、教师名、教师编号、教师年龄等相关信息，具体可见表 3.5:

表 3.5 教师表

字段名称	数据类型及长度	是否主键	备注
JiaoshiID	Int(20)	是	教师 ID
Chuangjianshijian	Datetime	否	创建时间

Gengxinshijian	Datetime	否	更新时间
Xingming	Varchar(20)	否	教师姓名
Nianling	Int(11)	否	教师年龄
Xingbie	Varchar(10)	否	教师性别
Jianjie	Varchar(80)	否	教师简介
ZhuanyeID	Varchar(10)	否	专业编号
Professional_id	Int(20)	否	所属专业编号

#### (5) 学生表

学生表包含学生 ID、学生姓名、学生学号、学生年龄、出生年月、学生性别、创建时间和更新时间以及学生所属专业 ID 和所属学院 ID 属性，具体可见表 3.6:

表 3.6 学生表

字段名称	数据类型及长度	是否主键	备注
XueshengID	Int(20)	是	学生 ID
Chuangjianshijian	Datetime	否	创建时间
Gengxinshijian	Datetime	否	更新时间
Age	Int(11)	否	学生年龄
Birthday	Datetime	否	出生年月
Sex	Int(11)	否	学生性别
Student_id	Varchar(5)	否	学生学号
Student_name	Varchar(20)	否	学生姓名
C_id	Int(20)	否	所属学院
P_id	Int(20)	否	所属专业

#### (6) 日志表

其中主要是针对日志相关信息进行存储，包括有日志 ID、内容以及更新时间等，具体可见表 3.7:

表 3.7 日志表

字段名称	数据类型及长度	是否主键	备注
RizhiID	Int(20)	是	日志 ID
Chuangjianshijian	Datetime	否	创建时间
Gengxinshijian	Datetime	否	更新时间
Neirong	Varchar(1024)	否	日志内容
Operator	Varchar(20)	否	日志操作者

#### (7) 班级表

班级表包含班级 ID、班级名称以及更新时间属性等，具体可见表 3.8:

表 3.8 班级表

字段名称	数据类型及长度	是否主键	备注
BanjiID	Int(20)	是	班级 ID
Chuangjianshijian	Datetime	否	创建时间
Gengxinshijian	Datetime	否	更新时间
Mingcheng	Varchar(18)	否	班级名称
Grade_id	Int(20)	否	所属年级

(8) 年级表

其中主要是针对年级信息进行存储，具体可见表 3.9:

表 3.9 年级表

字段名称	数据类型及长度	是否主键	备注
NianjiID	Int(20)	是	年级 ID
Chuangjianshijian	Datetime	否	创建时间
Gengxinshijian	Datetime	否	更新时间
Mingcheng	Varchar(20)	否	年级名称
Professional_id	Int(20)	否	所属专业

### 3.5 本章小结

第三章首先重点论述了本系统的需求分析，其中主要分析了功能结构、用户角色与相关用例，同时阐述了系统性能与安全需求；并对系统整体结构进行了讨论，然而按照需求分析结论完成了系统整体功能结构设计和其功能模块设计，逐一介绍了系统可以实现的各项功能，根据各功能模块绘制出功能结构与实现流程示意图等；以 MySQL 为基础设计出系统的 Education 数据库，构建出各种实体与实体关系表；在设计系统结构时验证了各类数据，以确保系统可以稳定运行并实现预期的功能。

## 4 系统实现与测试

本章详细介绍了系统的设计与实现过程，另外还介绍了测试的整个过程。详细介绍了从设计到系统开发，到最终的系统测试等一系列内容。

### 4.1 系统开发环境

系统开发环境主要包括了如下几方面内容：

- （1）硬件环境，CPU 选择的是 Inter(R) Core(TM) i7-8300M、主频为 2.8GHz 以及内存标准为 16G、硬盘为 1T；
- （2）开发语言选择的是 Java；
- （3）开发平台应用的是 IDEA 2019.2，对 JDK 1.8 进行配置，从而确保 Java 更好的运行，Web 服务器选择的是 Tomcat9.0.8，Maven 的项目结构完成项目构建，从而进行管理项目等诸多工作；
- （4）数据存储过程中数据库选择的是 MySQL 8.0.11，其中管理数据库可视化工具选择的是 SQLyog13.1.5，利用操作界面操作各项数据信息，从而促使整个流程具有更高的工作效率；
- （5）通过 Visio 2019 对分析图与流程图进行绘制，如功能流程图和 UML 用例图等；
- （6）通过 Power Designer 16.5 对系统数据库进行建模，从而可以绘制出有关的物理模型与概念模型等；
- （7）通过 Spring Boot 2.0.3 + MyBatis 3.4.4 完成系统框架的开发，将 jar 包导入到 pom 文件；
- （8）操作系统为 Windows 10，浏览器选择的是 Chrome、IE7/8 等。

### 4.2 系统功能模块实现

通过功能模块设计可以发现，系统的模块功能共包括了五部分内容，主要为系统管理模块和学院管理模块等。在本小节当中主要阐述的是公共部分内容，然后详细说明了不同模块所具备的业务逻辑内容。由于篇幅的原因，本文只是对其中的核心功能进行了介绍，阐述了实现的过程。

#### 4.2.1 系统公共部分说明

系统的文件结构主要包括的包共有 5 个,并且每个包也都包括了如下四类:第一,controller。主要存放的是 XXController 类。负责的是前端和后台之间的交互;第二,service。主要存放的是 XXService 接口等,负责的是业务逻辑;第三,dao。主要存放的是 XXDao 接口。负责的是和数据库之间的交互;第四,entity。主要存放的是对象类。如 Student。除此之外,还有其他一些类,比如 aspect 包。主要用来存放切面类等。

在系统中,选取 Free Marker 对模板进行渲染,将其当作是 view 部分,将 Free Marker 当作模板语言,以此接收模板数据。同时,view 与 controller 相连,就要把相关依赖引入到 pom 文件内,将相关配置加入配置文件,把标签库纳入模板,以此对模板的视图加以编辑。

在系统中还应用了很多注解。对于 controller 类,使用的是@Controller 注解。对于 Dao 类,使用的是@Mapper 注解。对于 service 类,使用的是@Service 注解。同时,controller 层也使用了其他注解,如@RequestMapping 等,对参数、请求路径等相关数据进行定义。@ResponseBody 或@RequestBody 还能够反馈或获取固定格式,通过@RequiresPermissions 等对用户的访问权限起到限制作用。对于 service 层利用@Transactional 开展事务管理。对于 Dao 层,选取@Insert 等在相应接口对 SQL 语句加以定义。此次系统开发期间,对于数据库的操作,使用的注解形式为 mapper.xml。

系统利用 MySQL 数据库和 MyBatis 数据持久层框架,通过 SQL 语句对 CRUD 操作进行定义。在配置过程中,需要将有关的依赖引入到 pom 当中,需要进行 MyBatis 的基本配置和数据源配置,通过 Dao 对应的 mapper.xml 实现对 SQL 语句的编写,并且通过 Mapping 等配置来实现 mapper.xml 与 Dao 之间的对应关系。针对缓存和日志等存在“横向”特性的功能,通过 AOP 的注解@Pointcut 等对有关日志操作进行定义。

## 4.2.2 用户信息管理模块实现

用户信息管理模块主要包括了用户查询与信息浏览等诸多功能,通过该模块可以对用户信息进行修改与浏览等操作。其中包括的对象有 UserController 和 User,具体方法有 deleteUser 以及 addUser 等。

在对用户信息进行修改时可以通过模块实现,可以对姓名以及邮箱信息等内容进行修改。因此,对于修改功能而言,可以以具体参数为基础来对属性进行修改,主要应用的方法包括了如下两个:第一,getUser;第二,updateUser。具体说明为:

### (1) 用户信息浏览功能

第一,方法名: getUser;

第二,参数说明: Integer 类型的 ID;

第三,调用函数说明: 对参数进行检查,判断参数是否为空,如果是空的话则进

行相应的记录，提示错误信息，然后直接结束；对有关对象进行查询，参数值不为空的情况下，通过 `getByID` 方法进行调用，从而利用相应的接口来对用户信息进行查询，如果未能查到的话给予相应的反馈信息，比如“此用户不存在”等，然后直接结束；如果查询到的话，就将 `User` 对象数据返回，按照界面需求进行显示。

## （2）用户信息修改功能

第一，方法名：`updateUser`；

第二，参数说明：主要包括了如下三类参数：第一，用户 ID，具体为 `Integer` 类型。用户名称，具体为 `String` 类型。描述，具体为 `String` 类型。邮箱，具体为 `String` 类型；第二，用户 ID，具体为 `Integer` 类型。头像，具体为 `byte` 类型；第三，的用户 ID，具体为 `Integer` 类型。旧密码，具体为 `String` 类型。新密码，具体为 `String` 类型。确认密码，具体为 `String` 类型。

第三，调用方法说明：对参数进行检查，判断参数是否为无效，如果是无效的话则进行相应的记录，提示错误信息，然后直接结束；对有关对象进行查询，在参数值有效的情况下，通过 `getByID` 方法进行调用，从而利用相应的接口来对用户信息进行查询，如果未能查到的话给予相应的反馈信息，比如“此用户不存在”等，然后直接结束；如果查询到的话，就将 `User` 对象数据返回，对 `set` 方法进行调用，放入新对象当中。存储对象方面，对 `update` 方法进行调用，通过 `UserDao` 进行调用对应的接口，从而对用户信息进行更新。具体见下图。



The figure shows a web-based user profile editing interface. At the top, there is a navigation bar with a hamburger menu icon, a chat bubble icon, a user profile icon, and the text '1234567'. Below this is the main content area. On the left, there is a square placeholder for a user avatar, currently showing a cartoon monkey. To its right is a square button with a plus sign for uploading a new avatar. Below these are several form fields: 'Login Name' with the value '1234567', 'Login Password' with masked dots, 'Role' with a dropdown menu showing 'Super Administrator', 'Phone Number' with the value '123456', and 'Email Account' with the value '123456@qq.com'. Below the email field are two rows of radio buttons: 'Gender' with options 'Male', 'Female' (selected), and 'Unknown'; and 'Status' with options 'Normal' (selected) and 'Frozen'. At the bottom left are two buttons: 'Confirm' (highlighted in green) and 'Return'.

图 4.1 系统个人信息设置界面效果图

在信息设置界面有用户头像上传和用户个人信息的输入框和个人信息的单选框。



以及最终信息的确定按钮和取消按钮。界面的右上角还有登陆用户的基本信息。以及修改密码和退出登录选项卡。

### 4.2.3 学生管理模块实现

学生管理模块涉及到的内容主要包括了学生信息浏览、学生信息编辑等内容，包括了对学生信息的修改和增加等操作，涉及对象主要包括 `Student` 以及 `StudentController`，具体方法包括 `update`、`create` 以及 `get`。操作本身并没有复杂的业务逻辑，对应的数据库主要包括了 `insert` 等，通过参数执行获取对应的数据。

#### （1）学生信息编辑功能

该功能主要包括了对信息进行修改与创建等操作，基于此还可以对学生对象进行查询，了解是否存在该数据内容，如果不存在的话，则给予相应的提示，即“学生 XXX 不存在”等，如果存在的话，则把数据信息更新到原有对象当中，对数据库进行更新，接下来将介绍创建方法。

第一，方法名：`createStudent`；

第二，参数说明：`Student` 学生对象，包含必填参数 `String` 类型的姓名（有长度限制）；`String` 类型的学号（必填）；`Char` 类型的性别参数（必填）；包含 `Int` 类型的年龄参数（必填）等内容。

通过图 4.2 可知。主要包括了年龄、姓名以及出生年月日等信息。



图 4.2 学生信息编辑界面实现效果图

第三，调用方法说明：检查参数，对 `checkParams` 方法进行调用，从而对调入参数进行检查，如果属于无效的话，则给出相应的提示信息，比如“无效参数 XXX”等，然后马上结束；如果参数有效的话，那么就继续展开后续操作。对信息进行设置，

通过对 generateID 方法进行调用,从而即可得到学生 ID。对 getUserId 方法进行调用,即可得到学生 ID。对 getByName 进行调用的话,即可对学生具体情况进行检查(存在则调用 getStudent 方法获得学生信息;不存在则先调用 StudentService 的 add 方法创建学生信息),信息设置完毕以后;最后对 set 方法进行调用,把数据信息都设置到 Student 对象当中。存储对象,对 add 方法进行调用,并且对 StudentDao 对应接口进行调用,将学生对象内容插入到数据库中。

## (2) 学生信息浏览功能

学生信息浏览可以查询学生的信息,查询的方式有多种,可以根据 ID 进行查询,也可以根据名字进行模糊查询等。根据名字模糊查询可以在搜索框查询包含这一关键字的所有学生信息,因为搜索框设置的是对学生姓名的查询,所以这一部分主要说明学生姓名的模糊查询。

该功能当中主要对姓名参数展开查询操作,主要包括了根据创建出的时间逆序进行展示,也就是最前面展示的是最新添加的内容。

第一,方法名: getStudents;

第二,参数说明: Limit, 为 Int 类型,对每页的学生数量进行设置; offset, 为 Int 类型,对页数进行设置;

第三,方法调用说明:对参数进行检查,对数据进行查询,是否为空值,如果是无效信息的话,那么直接结束操作;如果是有效的话,那么继续操作。获取数据,对 list 方法进行调用,对 StudentDao 对应接口进行调用,按照时间进行排序,从而可以得到最新的学生集合,然后按照页面需求等内容来展示。

通过下图可知。界面左上角有添加、编辑和删除三个选项,右上角是学生名称的搜索框,界面中间分页显示学生信息数据,左下角有分页选项卡。



图 4.3 学生信息浏览界面效果图

## (3) 学生信息管理功能

此项功能属于后台管理功能的一项重要管理内容，可以对全部学生进行修改、删除以及查询操作，同样属于普通用户，需要按照要求将信息填写完全。并且主要说明的是删除操作，主要包括了如下两种：第一，单个删除 `oneRemove`；第二，批量删除 `allRemove`；从参数类型出发共包括了 `deleteByID` 按照学生 ID 来对信息进行删除操作。`deleteByUserID` 按照用户 ID 来对学生信息进行批量删除操作。`deleteByCollegeID` 按照学院 ID 来对学生进行批量删除操作。此处操作主要是利用多样化的查询方法，核心功能主要为 `deleteByID`，因此下文主要对该方法进行介绍。

第一，方法名：`removeStudent`；

第二，参数说明：`Integer` 类型的学生 ID；

第三，调用方法说明：对参数进行检查，判断参数是否为空，如果是空的话，提示错误信息，然后直接结束；对有关对象进行查询，参数值不为空的情况下，通过 `getByID` 方法进行调用，了解数据库中是否已经有学生的信息，如果未能查到的话给予相应的信息，比如“学生 XXX 不存在”等；如果查询到的话，继续操作。

删除对象，通过对 `deleteStudentRelation` 方法进行调用，以及对 `StudentDao` 的对应接口进行调用，从而将有关学生对象之间的对应关系进行删除，比如专业和学院等数据信息。然后通过对 `deleteByID` 方法进行调用，`StudentService` 调用 `StudentDao` 的对应接口从数据库中删除学生信息。

学生信息管理界面如图 4.4 所示。和学生信息浏览界面相似，学生信息管理界面也有添加、编辑和删除选项卡，学生姓名搜索框，分页信息选项卡等。在页面中展示了学号、姓名、性别、年龄、生日、所属专业、所属班级、添加时间等信息。

+ 添加			- 编辑			× 删除			学生名称· <input type="text"/>		搜索
<input type="checkbox"/>	学号	姓名	性别	年龄	生日	所属专业	所属班级	添加时间			
<input type="checkbox"/>	17102	李同学	男	25	2020-10-27 9:54:00	电气自动化	电气(二)班	2020-10-27 18:40:55			
<input type="checkbox"/>	17105	Grace	女	23	2020-10-9 0:00:00	计算机技术	电气(二)班	2020-10-28 16:52:56			
<input type="checkbox"/>	19216	刘坚强	男	26	1996-6-20 22:00:00	电子信息工程	电子(一)班	2021-1-17 15:34:09			

图 4.4 学生信息管理界面实现效果图

#### 4.2.4 学院管理模块实现

学院管理模块是以学院为对象，执行新增和修改等操作，各项操作和专业之间具有密切关系，并且学院与专业之间的关系属于一种一对多的关系，因此对于各项操作而言，需要注重专业关系表。此处重点介绍创建学院的过程等内容，另外还对查询过程进行了介绍。

##### (1) 学院信息创建功能

在对个人信息进行编辑的时候，在输入框当中将学院信息输入其中，即可完成创建学院的过程；管理员可以直接完成创建操作。此处说明的是管理员对学院进行创建的过程，具体如下所示：

方法名：`createCollege`；

参数说明：`College` 对象，主要包括了 `String` 类型的学院名称；

调用方法说明：

对参数进行检查，判断参数是否有效，如果是无效的话，提示错误信息，然后直接结束；对有关对象进行查询，在参数值有效的情况下，则继续后续添加操作。

存储对象，通过对 `add` 方法进行调用，对 `CollegeDao` 对应 `Insert` 接口进行调用，从而完成学院的创建操作，并且插入到数据库中。通过下图可知。界面中展示了学院介绍和添加时间等信息。

+ 添加    - 编辑    × 删除			学院名称 ▾    工程    搜索
<input type="checkbox"/>	学院名称	学院介绍	添加时间
<input type="checkbox"/>	机械工程 学院	拥有国家数字化设计与制造创新中心、高端数控装备集成攻关大平台。	2020-10-27 18:30:54
<input type="checkbox"/>	食品工程 学院	近十年来，学院承担并完成的国家和企业科研项目千余项。	2020-10-28 15:12:19
<input type="checkbox"/>	电气与电子工程 学院	电气与电子工程专业的特点是电气与电子并重，电力电子与信息电子相融，软件与硬件兼备，装置与系统结合。	2021-1-11 16:43:09

图 4.5 搜索资源展示界面实现效果图

## （2）学院信息查询功能

此项功能包括了多种不同的方式，可以根据 ID 进行查询，也可以根据名字来进行模糊查询等。在对学院信息进行修改与删除的时候，都可以根据 ID 查询来进行相关的操作，在查询学院信息的时候需要根据名字来进行模糊查询，从而展开后续的操作。此处需要明确的是根据分页参数条件以及名字等进行查询。

第一，按名字模糊查询。该功能主要是对 `getListByName` 方法进行调用，并且对于名字参数而言进行“%名字%”包装。在学院表当中执行模糊查询，然后通过集合统一收集查询到的数据结果，并且集合的 `key` 选择 `college`，通过 `json` 格式返至前端，基于此前端展示出相应的结果。

第二，按分页参数条件查询。此功能主要是对 `list` 方法进行调用，对其中的 `CollegeDao` 对应接口进行调用，利用时间排序等方式进行，从而得到最新的学院信息集合，然后从页面需求出发，从而展示出页面参数。

+ 添加

- 编辑

× 删除

学院名称 ▾

传媒

搜索

<input type="checkbox"/>	学院名称	学院介绍	添加时间
<input type="checkbox"/>	艺术与传媒学院	进入新时代，艺术与传媒学院始终秉承“艰苦朴素，求真务实”的校训精神	2021-2-7 19:59:55
<input type="checkbox"/>	人文与传媒学院	学院设中国语言文学、新闻传播学、中国史三个一级学科，已建成从本科到硕士完整的人才培养体系。	2021-2-7 21:06:36

«

1

»

共1页,2条数据

图 4.6 学院信息浏览展示界面图

通过图 4.6 可知。学院信息浏览界面和学院信息管理界面的不同之处在于学院信息浏览界面通过分页展示数据。其他设置和管理界面相同。

### 4.2.5 专业管理模块实现

专业管理模块主要包括了查询、添加和修改等操作。主要涉及对象 `User`，`Professional`，`College`，`Student`，`Teacher`，`Content` 等等。因为专业管理模块多与学院管理模块和学生管理模块进行交互使用，涉及到的方法也很多，任何一个模块都可以自行执行删除与增加等操作，具有基本相同的过程，举例而言。

#### （1）增加专业功能



专业的增加功能主要用于教师和管理员对专业信息进行实时的调整,以方便学生及时了解选课信息,并根据自己的情况做出相应的调整。这里涉及的方法主要是 `addProfessional`。专业信息增加的界面如图 4.7 所示。专业增加界面包括专业编号、专业名称、所属学院、和学院介绍等信息。



图 4.7 专业信息增加界面图

第一, 方法名: `addProfessional`;

第二, 参数说明: `Professional` 对象, 包括必要参数 `Integer` 类型的专业 ID, `String` 类型的专业介绍;

第三, 调用方法说明: 对参数进行检查, 判断参数是否有效, 如果是无效的话, 提示错误信息, 然后直接结束; 有效则调用 `ProfessionalService` 的 `getById` 方法, `ProfessionalService` 调用 `ProfessionalDao` 的对应接口, 查询专业信息是否存在数据库中; 设置添加信息, 调用 `filterProfessional` 方法过滤专业介绍, 判断准确无误后, 调用 `Professional` 对象的 `set` 方法将所有信息设置到新创建的 `Professional` 对象中; 存储对象, 调用 `professionalService` 的 `add` 方法, `professionalService` 调用 `professionalDao` 的对应接口将专业信息存入数据库中。

## (2) 专业查看功能

专业查看通常情况下不仅包括了对当前的专业信息内容进行查看, 另外还可查看其它专业有关的全部信息。并且查看过程中可以只显示其中一条内容, 也可以查看所有专业信息, 这里主要说明查看所有专业信息。

第一, 方法名: `getAllProfessional`;

第二, 参数说明: `Integer` 的专业 ID;

第三, 调用方法说明: 对参数进行检查, 判断调入参数是否为空, 如果为空的话, 提示错误信息, 然后直接结束; 对有关对象进行查询, 参数值不为空的情况下, 则继续后续添加操作。获取对象, 调用 `professionalService` 的 `getById` 方法,

professionalService 调用 professionalDao 的对应接口查询专业信息是否存在数据库中。

专业信息查看界面如图 4.8 所示。专业信息查看界面是将专业信息通过分页展示出来。并提供添加、编辑和删除等操作，以及根据专业名称进行条件搜索。

+ 添加		- 编辑	× 删除	专业名称 ▾		搜索
<input type="checkbox"/>	专业编号	专业名称	所属学院	介绍	添加时间	
<input type="checkbox"/>	1001	通信工程	电气与电子工程学院	由电机工程学系、系统科学与工程学系、应用电子学系和电工电子基础教学中心组成。	2020-10-27 18:31:47	
<input type="checkbox"/>	1002	电子信息工程	电气与电子工程学院	学院英才辈出、桃李满园。	2020-10-28 16:14:45	
<input type="checkbox"/>	1005	计算机技术	数学与计算机学院	学院师资队伍实力雄厚、结构合理、发展迅速。	2020-10-28 16:22:28	

图 4.8 专业信息查看界面图

#### 4.2.6 系统管理模块实现

系统管理模块主要包括了登陆和注册等诸多功能，另外用户还可以执行增加和修改等操作。涉及到的用户对象主要包括了 User 等，另外方法主要包括了 modify 和 login 等。

##### (1) 用户登录功能

用户通过输入用户名和密码进行登录检查。用户退出主要通过对 logout 方法进行调用，此处介绍的主要是 login 方法。

第一，方法名：login；

第二，参数说明：用户名，主要是 String 类型。密码，主要是 String 类型；

第三，方法调用说明：对参数进行检查，通过对 isEmpty 方法进行调用以及对比 null 的方式检查参数，如果为空的话则提示错误信息，然后马上结束；如果不为空的话，就给出下述操作。通过对 encrypt 方法进行调用，数据库中存储的密码为加密的，并且对经过加密处理的密码和用户名进行打包，从而可以得到

UsernamePasswordToken 类，然后对 login 方法进行调用，从而执行登录验证。

系统登陆界面如图 4.9 所示。包括用户名、密码的输入框和验证码的输入框。



图 4.9 系统登陆界面图

登录验证的过程：对 `getByName` 方法进行调用，然后对 `UserDao` 的方法进行调用，获取数据库中有关的数据信息，如果没有相应用户的话，则给予相应的提示，比如“此用户名不存在，请检查用户名”等，然后结束；从用户信息层面出发即可获知存储密码，比对已经经过加密处理的输入密码，如果两种数据不同的话则给予相应的提示，比如“密码错误，请检查密码”等，然后结束；如果相同的话，则成功登录到系统中，跳转到主页面。

## (2) 用户管理功能

系统后台管理中重要的一项功能为用户管理功能，其中管理员拥有最高的权限。通过该功能模块，可以实现对用户进行新增和删除等操作，这些操作的对象都是用户，比如创建用户的方法。管理员对用户进行创建和用户自己注册所要执行的操作步骤基本相同。下图界面中展示了用户的用户名称、头像信息等诸多信息。

+ 添加

✎ 编辑

✖ 删除

用户名 -

搜索

<input type="checkbox"/>	头像	用户名	角色	状态	性别	手机	邮箱	添加时间
<input type="checkbox"/>		1234567	超级管理员	正常	女	123456	123456@qq.com	2020-10-28 19:21:43
<input type="checkbox"/>		王浩同学	超级管理员	冻结	男	12398652112	whpc@qq.com	2021-1-17 13:25:41
<input type="checkbox"/>		赵刚同学	普通用户	正常	男	123456789	@.com	2021-3-9 13:16:47
<input type="checkbox"/>		feixingren	超级管理员	冻结	女	13212324561	feiqq.com	2021-4-5 17:49:53

图 4.10 后台管理之用户管理界面图



然而也具有一定的不足之处，操作对象不仅包括了用户信息，还包括了 short 类型等内容。因此可以对 add 方法等进行调用，并且对 UserDao 进行调用，然后创建出管理员类型的用户等。

### 4.3 分布式架构技术实现

分布式集群将不同的多台计算机通过中间件连接起来，完成以前单节点系统完成的工作量。用户在访问时，不会感受到两种架构的使用区别。本次毕业设计在系统控制器部分采用分布式架构，通过使用 Dubbo 分布式框架，将系统服务的生产者和服务消费者关联起来。服务生产者将自己提供的服务注册到 Zookeeper 注册中心，服务消费者从注册中心登记并获得相应的服务，由 Zookeeper 统一进行管理，保证信息传输的完整性和稳定性。

在虚拟机上面搭建四台 Linux 操作系统，组成一个分布式集群。然后将不同功能模块放置在不同的服务器。例如，fastdfs-image-server\_133 服务器只负责文件的上传和下载，其他功能模块分别放置在另外三台服务器，使用 Zookeeper 消息中间件来进行监控和管理以及消息在服务器中间的传递。通过这样一个分布式集群来提高系统的稳定性，增大系统的容量。在实际开发过程中，通过采购高性能服务器来组建集群，以此解决高并发的的问题，面对同一时刻大量用户访问系统，起到一个削峰的作用。系统除了在数据库访问层中使用 MySQL 数据库，还使用分布式文件服务器 Fast DFS 来完成大量图片数据的存储，以此缓解数据库的压力。本文在 Linux 系统中单独部署一个 Fast DFS 服务器来专门完成大量用户图像的信息存储。Linux 系统操作界面如图 4.11 所示：



图 4.11 分布式集群服务器管理界面图

## 4.4 Zookeeper 消息中间件部署

分布式集群为了获得更好的可靠性服务和更快的响应速度，在系统中采用了部署 Zookeeper 消息中间件的模式。在这里简要说明四台节点服务器的 Zookeeper 部署情况。

(1) 对节点规划如下表 4.1 所示：

表 4.1 Zookeeper 节点规划表

服务	端口号	服务器	容器名
Zk01	8082, 8083, 8084, 8085	192.168.154.110	zk01
Zk02	8082, 8083, 8084, 8085	192.168.154.111	zk02
Zk03	8082, 8083, 8084, 8085	192.168.154.112	zk03
Zk04	8082, 8083, 8084, 8085	192.168.154.113	zk04

(2) 拉取 Zookeeper 镜像，这里选用 Zookeeper3.4 版本。

(3) 创建容器：包括容器的名字、id、服务网络等。

(4) 启动 Zookeeper 容器即可。

## 4.5 系统测试

系统测试是指在一定条件下，为了检测程序错误而执行一些操作的程序运行过程。软件测试和软件系统研发是同时进行的。经过对系统各功能单元的测试，找出系统在开发和设计过程中出现的缺陷和错误，并及时对出现的漏洞和问题进行修复，以及开发方向的调整<sup>[42,43]</sup>。以往的系统开发相关研究结果表明，越早进行系统测试，系统缺陷以及漏洞的修复成本也就会越低，因此在系统开发过程中，最好尽早进行系统测试的相关工作。除此之外，系统测试的另一大作用便是对系统是否能够满足用户的需求进行测试，验证系统的响应时间、对于业务流程的处理时间等因素，最终测试系统的相应功能模块是否符合使用者的使用要求。并且，由于系统的用户满意程度很大一部分取决于系统性能的好坏，需要将对系统性能的测试纳入系统测试过程中。系统在具体开发过程中，系统测试是一项必不可少的工作，在这项工作的基础上评估系统功能模块的实施和运行情况，以便可以准确地实施该系统。在将学生信息管理系统设计完成正式上传到服务器之前，需要采取的步骤之一是对系统进行测试，以确保及时纠正发现的错误<sup>[44]</sup>。

### 4.5.1 测试方法

本系统测试所用方法包括以下两个：

(1) 黑盒测试。又被称为功能测试，它是通过把整个系统当作一个整体，在不了解系统内部数据结构的基础上，经过测试来检查各个功能是否能够实现目标需求，以达到检验系统外部结构的作用。黑盒测试能够检验系统是否能够做到预先的功能需求以及设计需要<sup>[45]</sup>。在黑盒测试过程中，需要对测试要求进行划分，而设计出功能和性能需求分析测试计划，然后将测试计划分解、细化成若干可执行的测试过程并进行测试，首先按照功能模块进行单元测试，然后将单元模块进行组合测试，组合测试通过后将整个系统整合起来进行集成测试。最终对软件系统的性能进行测试和评定。

(2) 白盒测试。也被称为结构测试或者透明盒测试，它是一种依赖于系统内部结构和逻辑处理关系的测试方式，在充分了解系统内部逻辑结构的前提下，检验系统的各项功能和性能是否符合设计目标。在白盒测试过程中，通过覆盖尽量多的各项流程的通道，检查系统的状态，以达到对系统内部结构进行梳理的目的。白盒测试能够检测系统程序中的每条分支和路径，揭示隐藏在代码中的错误，使程序达到设计的最优化。

## 4.5.2 测试用例

测试用例主要包括了性能、功能以及安全等三类测试用例。主要实现安全检查与异常状况等方面的检查与测试。但是因为篇幅的原因，本文通过典型用例来进行详细的描述。

### (1) 系统功能测试用例

此方面的测试内容主要包括了有效与无效两种输入；对应的是正确输入的响应操作以及错误输入的操作情况。

表 4.2 系统正常反馈测试用例结果表

用例编号	项目	前置条件	具体操作	输入信息	输出结果
U001	用户注册	用户已经于浏览器当中将系统服务地址输入，然后点击访问主页面	第一，针对主页面当中“注册”这一按钮进行点击；第二，将注册信息填写完成；第三，点击提交	注册信息输入无误 用户名: yangdong 密码: yangdong666 描述: miaoshu 邮箱: yangdong666@163.com	发出注册成功提示，并跳转至登录主页面
U002	用户登录	用户已经于浏览器当中	第一，针对主页面当中的“登录”这	登录信息输入无误 用户名: yangdong	发出登陆成功提示，并不跳转

	陆	将系统服务 器地址输入，然后点 击访问主页面	一按钮点击；第 二，将登录信息填 写进去；第三，点 击登录	密码： yangdong666	至系统主页面
S001	添加学 生信息	是有效用 户，同时已 经登录成功	第一，针对“添加” 这一按钮进行点 击；第二，将学生 信息填写完成；第 三，点击添加	学生信息输入无误 学号： 0401 姓名： yangdong	发出添加成功 的提示，能够于 “学生列表”里 面对新添加信 息查看
S002	修改学 生信息	是有效用 户，同时已 经登录系 统，点击进 入到“学生 列表”里面	第一，点击所需修 改学生信息；第 二，将修改内容填 写进去；第三，点 击保存保存	新的学生信息输入无 误 学号： 0401 姓名： yangdong	发出修改成功 提示，能够于 “学生列表”里 面对新修改信 息查看
S003	删除 学生 信息	是有效用 户，同时已 经登录系 统，点击进 入到“学生 列表”里面	第一，点击所需删 除学生信息；第 二，点击确认		发出删除成功 提示，在将“学 生列表”刷新之 后，已删除信息 未显示
S004	查询 学生 信息	用户正针对 系统主页面 进行访问	第一，对搜索框进 行点击；第二，输 入查询条件	查询信息输入无误 查询： yangdong	系统进入查询 结果界面，将相 关信息显示
X001	增加学 院信息	是有效用 户，同时已 经登录成功	第一，针对“添加” 按钮进行点击；第 二，将学院信息填 写进入；第三，点 击确定	学院信息输入无误 名称： mingcheng 介绍： jieshao	发出添加成功 的提示，系统跳 转至“学院信息 编辑”页面，能 够在列表里面 看到新信息
X002	删除学 院信息	是有效用 户，同时已	第一，针对所需删 除信息选定；第		发出删除成功 的提示，在将列

经 登 录 系 二，点击删除	表刷新之后，未
统，点击进	显示已删除信
入到“学院	息
列表”里面	

以用户注册成功为例，首先进入用户信息填写界面，输入用户名、登陆密码、所属角色、手机号码和邮箱账号等信息。并在性别和账号状态的单选框里面选择相应的信息。最后点击确定按钮完成注册或选择返回按钮退出用户注册。用户注册成功界面如图 4.12 所示。



图 4.12 用户注册成功测试界面效果图

(2) 系统异常反馈测试

表 4.3 系统异常反馈测试用例结果表

用例 编号	项目	前置条件	具体操作	输入信息	输出结果
U003	用户 异常 注册	用户已经登录系 统且进入注册页 面	将非法字符填写进 去	¥#*!	发出错误提示
			将多于 64 个字符填 写进去	Abcdrfg.....	
			填写已经注册完的 用户名称	Guanliyuan	发出该用户名已 存在的提示
		用户已经登录系	输入字符小于 6 个	000	发出“密码长度

U004	异常登陆	系统且进入注册页面，在密码之外的所有参数都是有效的	输入字数多于 10 个	Yangdongyangdongyangdong	应于 6-10 个字符之间”的提示
		用户已经登录系统且进入注册页面，在描述之外的所有参数都是有效的	输入描述字符超过 255 个	当前学生信息比较多……	发出字符长度应低于 255 个的提示
		用户已经登录系统且进入注册页面，在邮箱之外的所有参数都是有效的	输入不正确的邮箱名称	Yangdong.163.com	发出“邮箱格式错误”的提示
		用户已经登录系统且进入登录界面，在用户名之外，其他参数都是有效的	用户名为空		发出用户名不可为空的提示
			输入错误的用户名	YangDongDong	发出用户名错误的提示

以用户异常注册为例，在注册信息填写完之后，输入密码长度小于规定长度时，系统会给出相应的提示，即“密码长度需要大于 4!”。具体见下图 4.13 所示。



图 4.13 用户注册失败界面图

(2) 系统性能测试用例

以系统性能以及可靠性用例展开说明，具体可见表 4.4:

表 4.4 系统性能测试用例情况表

用例编号	项目	具体操作	测试结果
P001	性能	登陆到系统当中，针对页面响应时间进行记录	响应时间低于 2min
		对学生信息进行编辑，将提交响应时间记录下来	响应时间低于 2min
		进行信息检索，将反馈响应时间记录下来	响应时间低于 2min
		多个用户共同完成登陆操作	响应时间低于 10min
P002	可靠性	单人持续操作 24h，将系统可用时间记录下来	可用时间与总测试时间的比例低于 99%
		多人持续操作 24h，将系统可用时间记录下来	可用时间与总测试时间的比例低于 99%

当用户通过 URL 暴力访问需要访问权限的界面时，系统会跳转到相应的拦截界面。系统通过定义一个权限过滤器，对每一个访问该应用的请求进行过滤，若具有访问权限，则直接跳转到相应界面即可；若不具有访问权限，则跳转到 404 测试界面。其中，由 Java 程序编写的过滤器在这一拦截中扮演了相当重要的角色，其工作原理是当访问请求到达 Web 容器时，过滤器会先对访问请求进行解析，使用解析出的 URL 作为比较对象，从 Map 中查找是否存在相匹配的 key。若存在，则读取其 value，即 Filter 对象的引用，将该引用存入到数组中。然后继续向后查找，直到将该 Map 查找完毕。数组初始化结束后，开始按照数组元素顺序进行执行。所有数组中的 Filter 全部执行完毕后，再跳转到请求的目标资源。

### (3) 安全测试用例

以内容过滤和界面拦截的一些用例展开说明，具体可见表 4.5:

表 4.5 安全测试用例结果表

用例编号	项目	异常情况	具体操作	输出结果
J001	界面拦截	选择借助 url 针对	用户没有登录，而是借助 url 针	转为用户登陆
		所需用户权限地	对信息编辑界面进行强制访问	界面
		址进行“暴力”访问	用户已经登录，借助 url 针对信息编辑界面进行强制访问	转为用户登陆界面

## 4.5.3 测试结论

学生信息管理系统从测试用例出发进行 2 天时间的测试工作，测试用例基本都已经通过了测试，只有极少一部分没有通过。比如新增学生信息时，将信息填写完成并提交之后，系统界面没有变化，依然为提交状态；有少部分内嵌页面的方法路径可利

用 URL 来进行访问等。从测试结果出发进行修正，很多问题得到了有效的解决，促使了系统更好的运行。

## 4.6 本章小结

本章对学生信息管理系统的功能测试与实现过程进行了阐述，首先对系统开发工具、开发环境等内容进行了介绍；利用系统代码结构等内容详细介绍了不同功能模块的实现过程，并针对部分模块功能显示界面进行展示描述；基于 Zookeeper 搭建分布式集群；最后基于系统功能、性能以及安全性等进行测试，并将用例结果列出。



## 5 总结与展望

### 5.1 总结

本论文在介绍系统功能和结构的同时，回答了大致五个问题，即系统是什么，为何设计系统，系统应用的技术有哪些，为何要通过这些技术来解决这些问题，以及具体的实现过程等。这些问题的解决也是系统开发常用的几大步骤。因此，本文分别在绪论章节中学生信息管理系统的研究背景与意义，国内外研究情况，关键技术研究章节和系统需求分析、设计以及系统实现和测试章节来回答并解决以上问题。

(1) 首先分析了系统的几大基本需求，通过对系统功能，用户角色和角色对应的用例分析。并且通过 Visio 等绘图工具绘制流程图，从而得到所需的用例图。从而对系统功能进行详细说明；另外研究分析了系统的非功能性需求，通过系统的性能、易用性、可靠性等几大特性来进行了说明，更加全面地进行了需求分析；

(2) 详细设计了系统的整体结构、数据库，通过展开需求分析：对系统的功能结构以及体系结构进行设计，系统主要包括了学院信息管理模块等模块，并且利用功能结构流程图等诸多表达方式来说明了不同模块所对应的功能设计内容；利用系统功能结构对报表和数据库进行了设计，并且通过 Power Designer 等诸多工具对物理模型和逻辑模型进行了设计，通过工具最终得到了可执行性 SQL 文件，通过数据库生成对应字段和数据库表；在设计系统时，通过分层来对系统结构等内容进行设计。

(3) 选择系统开发环境和架构体系，全面且深入的描述系统可靠性以及主要模块的实现。利用 IDEA 编译器对项目代码进行编写，通过 Spring Boot + MyBatis 框架来实现系统基本架构。本文利用参数方法等内容来对系统功能进行描述；借助框架功能和代码结构来保证系统的高效性能。

(4) 系统结构和功能测试方法主要以黑盒测试和白盒测试为主，白盒测试以开发过程中代码检查为主，并辅以功能运行结果，其中功能测试阶段主要应用的是黑盒测试。测试用例包括性能、功能以及安全测试等三类。黑盒测试主要是通过测试用例展开测试，对测试结果进行记录，从而更好的对错误代码进行修改与完善。

### 5.2 展望

因为理论与系统实现过程的时间有限，另外设计实施的外在条件有限，尽管已经实现了其中的功能，然而还需要进一步改进与完善，还有很多地方需要扩展

与优化，需要继续深入研究思考：

（1）系统的数据库部分还可以考虑采用租用云服务器的形式，将项目代码打包上线，降低成本，减少后期的维护工作；

（2）在安全过滤防护措施方面，系统除了具有针对性的展开防御，有效防御脚本文件展开的攻击，还需要充分重视其他常见脚本攻击，如跨站 DOS/DDOS 脚本攻击等。并且还需要布置各种有效的防御防护措施，确保软件系统更加安全与可靠。

## 参考文献

- [1] 张炜斌. 高校学生信息管理系统的设计与实现[J]. 信息与电脑(理论版), 2020, 32(21): 82-84.
- [2] 万鹏. 基于 JSP 技术的学生信息管理系统设计与实现[D]. 电子科技大学, 2013.
- [3] 彭毅超. 基于 B/S 模式的学生信息管理系统设计与实现[D]. 湖南大学, 2017.
- [4] 朱娜. 基于 JSP 技术的图书管理系统的设计与实现[J]. 黑龙江科学, 2018, 9(16): 11-13.
- [5] 陆丽. MVC 设计模式在 Java Web 开发中的应用研究[J]. 信息通信, 2020(04): 104-106.
- [6] 时月梅. 基于 Spring MVC、MyBatis 实现数据分页显示处理[J]. 信息技术与信息化, 2015(07): 203-206.
- [7] 彭勃. 高校学生信息管理系统设计[J]. 电子制作, 2020(16): 53-54.
- [8] Peng N X, Chen Y Q. Research and Implementation of Interceptor Based General Physical Pagination Component of Mybatis[J]. Applied Mechanics and Materials, 2014, 513-517: 1299-1304.
- [9] Jelena R, Branko M, Dusan S. Modelling and implementation of catalogue cards using FreeMarker[J]. Program: electronic library and information systems, 2009, 43(1): 62-76.
- [10] 韩洪勇, 朱珍珍, 姚连达. 基于 JSP 和 Servlet 的学生信息管理系统[J]. 科技风, 2020(19): 74.
- [11] Reenskaug T, Coplien J. The DCI Architecture: A New Vision of Object-Oriented Programming[J]. Programmer, 2009(6): 99-103.
- [12] Buschmann F, Henney K. Pattern-Oriented software architecture[M]. US: John Wiley & Sons Inc, 2004: 41-42.
- [13] 段升杰. 浅谈软件三层架构开发[J]. 信息与电脑(理论版), 2010(10): 44.
- [14] 孙亚志. 浅谈 Java web 技术 struts2 框架的 MVC 模式[J]. 数字技术与应用, 2013(5): 208-210.
- [15] 张雪敏. 浅议 MVC 设计模式在 Java Web 中的作用[J]. 科技风, 2018(22): 58-59.
- [16] Schaefer C, Ho C, Harrop R. Using Spring Remoting[M]. Pro Spring. 2014: 12-32.
- [17] Deinum M, Rubio D, Long J. Spring MVC: Async Processing[M]. Spring 5 Recipes. 2017: 27-32.
- [18] Pawlak R, Seinturier L. Foundations of AOP for J2EE development[M]. 2005:

- 121-124.
- [19]Johnson R. Expert One-on-One J2EE Design and Development[J]. 2002: 45-46.
- [20]Gutierrez F. Messaging with Redis[M]. Spring Boot Messaging. 2017: 7-16.
- [21]吕宇琛. Spring Boot 框架在 web 应用开发中的探讨[J]. 科技创新导报, 2018(8): 168-173.
- [22]张峰. 应用 Spring Boot 改变 web 应用开发模式[J]. 科技创新与应用, 2017(23): 193-194.
- [23]Reddy K S P. Beginning Spring Boot 2: Applications and Microservices with the Spring Framework[M]. 2017: 65-69.
- [24]荣艳冬. 关于 Mybatis 持久层框架的应用研究[J]. 信息安全与技术, 2015(12): 86-88.
- [25]贺军, 焦荣. 基于 XML 应用的 MyBatis 技术[J]. 火力与指挥控制, 2013(z1): 86-88.
- [26]孟亚辉. 浅谈软件项目开发过程中的需求分析[J]. 科技信息, 2009(11): 435-436.
- [27]李梦槐. 基于 SSH 框架的学生信息管理系统的设计与实现[J]. 江苏科技信息, 2018, 35(16): 52-54.
- [28]陈琳. 网络信息内容审计技术研究[D]. 南京航空航天大学, 2012.
- [29]Stellman A, Greene J. Applied Software Project Management[J]. Oreilly & Assoc Inc, 2006: 113-114.
- [30]Karl E. Software Requirements, Second Edition[M]. US: Microsoft Press, 2003: 23-25.
- [31]Lawrence D. Effective Requirements Practices[J]. Insight, 2001: 120-123.
- [32]韩坤. Web 服务安全会话管理的研究与实现[D]. 北京邮电大学, 2007.
- [33]张良银. 浅论 C/S 和 B/S 体系结构[J]. 工程地质计算机应用, 2006(4): 20-23.
- [34]蔡宏伟, 金连甫, 陈平. 可扩展三层 B/S 体系结构研究和应用[J]. 浙江理工大学学报(自然科学版), 2006, 23(2): 63-66.
- [35]徐兰芳, 吴永英. 数据库设计与实现[M]. 上海交通大学出版社, 2006: 45-48.
- [36]李厚刚, 白玉芹. 加密算法在共享软件加密中的简单应用[J]. 网络安全技术与应用, 2010(09): 44-46.
- [37]王馨慧. 数据挖掘技术在高校学生就业信息管理系统中的应用[J]. 电子世界, 2020(03): 132+135.
- [38]江志华. 学生成绩管理系统的设计与实现[J]. 信息与电脑(理论版), 2020, 32(10): 79-81.
- [39]王保华, 薛锦云, 丁树良. Java 多线程机制及其应用[J]. 计算机与现代化, 2000(1): 1-6.
- [40]宋福英. C 语言基于结构体的学生信息管理系统实现[J]. 计算机时代, 2020(03):

84-86+89.

- [41]Mudunuri, Srinivas. Mybatis in Practice: a step by step approach for learning mybatis framework[M]. CreateSpace Independent Publishing Platform, 2013.
- [42]Marsha C, Paul G. Guest editorial: selected areas in automated software engineering[J]. Automated Software Engineering, 2018, 25(1): 45-46.
- [43]Denisov E. Yu. et al. Automated Software Testing Technologies for Realistic Computer Graphics[J]. Programming and Computer Software, 2021, 47(1): 76-87.
- [44]王超, 张琪立, 田广强. 基于 Spring Boot 框架的学校机房计费管理系统的设计与实现[J]. 电子技术与软件工程, 2020(23): 159-160.
- [45]许悦. 基于 php 技术的学生管理系统设计研究[J]. 科学技术创新, 2020(07): 88-89.

## 致 谢

大学生活一晃而过，回首走过的岁月，心中倍感充实，当我写完这篇毕业论文的时候，有一种如释重负的感觉，感慨良多。在硕士学习即将圆满结束之际，我要对给予我关心和帮助的人以最真挚的感谢！

本论文是在我的导师陈西曲教授的悉心指导下完成的。陈老师渊博的知识、严谨的治学态度、丰富的实践经验无时无刻让我敬仰，在此感谢陈老师在论文开题、初稿以及在论文写作中在专业知识方面对我的培养，给我所提出的专业的有价值的意见和建议，为我提供了极大的帮助。在此向陈老师表示最诚挚的感谢和最崇高的敬意！

感谢武汉轻工大学对我的培养，提供给我这个学习机会，让我在这里不仅学习到了计算机方面的技术和知识，也让我感悟到了很多人生道理，这些技术知识和道理将伴随我一生，让我终生受益！本论文的完成也离不开其他各位老师、同学和朋友的关心与帮助。感谢他们在学习过程中给了我许多鼓励和帮助。

感谢多年来一直默默支持着我的家人，你们就是我拼搏时的坚强后盾，失意时的避风港湾，正是由于你们长久以来的支持和鼓励，让我有了面对一切困难的勇气。在这里我还要特别感谢我的校外导师聂先金伯伯，聂伯伯无论是在专业选择还是职业规划方面都给予我很多的帮助和支持。

毕业在即，在今后的工作和生活中，我会铭记师长们的教诲，继续不懈努力和追求，来报答所有支持和帮助过我的人！

衷心祝愿武汉轻工大学越办越好！祝愿电气与电子工程学院越办越好！