



计算机与软件工程学院

课程设计说明书

课 程 名 称: XXXXXXXXXXXXXXXXXX综合设计

课 程 代 码: XXXXXXXXXXX

题 目:

年级 / 专业: 201 /计算机科学与技术

学 生 姓 名:

学　　 号:

开 始 时 间:20 年月日

完 成 时 间:20 年月 日

设计成绩：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学习能力  （20） | 解决方案  （20） | UI设计  （20） | 技术水平  （20） | 协作与表达  （20） | 总 分（100） |
|  |  |  |  |  |  |

指导教师(签名) 20 年 月 日

目 录

摘要

[1 前言 1](#_Toc43456411)

[1.1 问题的提出 1](#_Toc43456412)

[1.2 任务与分析 1](#_Toc43456413)

[2.系统总体设计 2](#_Toc43456414)

[2.1 开发工具 2](#_Toc43456415)

[2.2 系统框图 2](#_Toc43456416)

[2.3 模块功能 4](#_Toc43456417)

[2.3.1 登录功能 4](#_Toc43456418)

[2.3.2 新增学生信息 4](#_Toc43456419)

[2.3.3 删除学生信息 5](#_Toc43456420)

[2.3.4修改学生信息 5](#_Toc43456421)

[2.3.5 查询学生信息 6](#_Toc43456422)

[3 交互界面设计 6](#_Toc43456423)

[3.1 控件介绍 6](#_Toc43456424)

[3.2 Form表单 8](#_Toc43456425)

[3.3 Button按钮 9](#_Toc43456426)

[3.4 Table表格 9](#_Toc43456427)

[3.5 Radio单选框 10](#_Toc43456428)

[3.6 导航选项 11](#_Toc43456429)

[4 功能详细设计 12](#_Toc43456430)

[4.1 数据库 13](#_Toc43456431)

[4.1.1 数据库名称及描述 13](#_Toc43456432)

[4.1.2 数据库ER图 14](#_Toc43456433)

[4.1.3 数据库表 14](#_Toc43456434)

[4.2 数据库连接 16](#_Toc43456435)

[4.3 用户的登录 19](#_Toc43456436)

[4.4 学生信息的增加 23](#_Toc43456437)

[4.5 学生信息的修改 24](#_Toc43456438)

[4.6 学生信息的删除 26](#_Toc43456439)

[4.7 学生信息的查询 28](#_Toc43456440)

[4.8 学生信息的导出 32](#_Toc43456441)

[4.9 软件使用说明 34](#_Toc43456442)

[5 系统功能测试 42](#_Toc43456443)

[结 论 44](#_Toc43456444)

[致 谢 45](#_Toc43456445)

[参考文献 46](#_Toc43456446)

**摘 要**

随着计算机的普及，在计算机发展突飞猛进的今天，国内各高校中各类学术管理系统已经不是一个新鲜事物了。对于学校而言，实现办公自动化将大大提高学校管理的工作效率。

当今时代是信息时代飞速发展。各个行业都需要进行信息处理，学生信息管理系统是一个学校不可缺少的重要部分，它为用户提供充足的信息和快捷的查询手段。学生信息管理越趋繁琐、复杂，工作业务繁杂，工作量大，传统的学生管理模式已经暴露出种种弊端：难以统一调配和处理，效率极低，缺乏科学性以及合理性。

信息的复杂化，计算机的智能化都标志着计算机在信息处理方面将有着无穷的潜力。将庞大的信息进行合理、智能、快捷地联合处理，极大提高用户工作质量和效率。

**关键词：**信息管理；计算机；对策

# 1 前言

## 1.1 问题的提出

在现代化的信息管理中，学生管理系统有着十分重要的作用。为了提高学校对学生信息管理的工作效率和工作质量。

学生信息管理系统综合学校教师和学生地实际情况，开发一个满足学校学生信息管理工作需求的基于WEB的学生信息管理系统

## 1.2 任务与分析

本课题主要的任务是为数计学院开发一个web管理系统，实现学生信息管理基本功能。

(1) 数据库设计: 参考学院学生管理过程,设计出数据库和表结构(如辅导员表,专业表，学生基本信息表,家庭信息表，获奖情况表，缺旷考核表等),录入部分初始数据,并测试字段有效性、表间约束等，保证数据库设计正确。

(2) Web界面风格设计：确定本站点风格，尽量格调一致。

(3)用户管理：本系统对管理员、辅导员，学生,不同级别用户在登陆后的操作权限不同。

(4)数据录入功能：实现各种数据的录入入库

(5) 学生情况管理:管理员以上权限的用户能实现对学生信息的录入、查询、修改、删除等维护功能。

（6）学生操作功能：学生仅对自己的情况查询，可对自己的非关键数据进行调整修改。完善自己的家庭数据等.

（7）其它管理功能：多元化的查询功能；获奖情况管理；缺课和旷课等管理。

（8）输出功能：能按多种方式排序查询结果，并以excel报表方式输出结果

分析：

(1)数据库的设计涉及多个表，标语表之间存在着关联，需要设计好表的主键与外键，确保数据库设计正确正常运行。

(2)系统需要不同权限，管理员、教师和学生的操作权限不同，需要在设计数据库时，添加一个权限字段，以保证权限的不同，使之能够进去的页面不通。

(3)包含多个页面、多种功能，需要对页面布局进行设计。不显臃肿和繁杂，让用户体验页面设计之美，方便操作。

# 2.系统总体设计

## 2.1 开发工具

本学生信息管理系统选用IntelliJ IDEA旗舰版，是Java变成语言开发的集成环境, IDEA所提供的智能编码，可以减少我的工作。IntelliJ在业界被公认为最好的java开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。

本次任务设计采用B/S构架，IDEA对JSP的完全支持，不需要任何插件是我选用为开发工具的主要原因。非常适合本次任务设计。对XML的完美支持，使得使用SSM框架完成的本次任务设计完美运行。

前台语言为JavaScript，前台中采用了JQuery、BootStrap等框架，使得开发过程变得简单高效美观。

开发环境：操作系统Windows 7及以上版本；IntelliJ IDEA旗舰版；Apache Tomcat 8.0.27；MySql 5.7.30；Java 1.8；Maven 3.3.6

运行环境：操作系统 Windows 7及以上版本；Apache Tomcat 8.0.27；MySql 5.7.30；Java 1.8；IntelliJ IDEA旗舰版；Maven 3.3.6等

## 2.2 系统框图

（1）系统组成框图：图2-1

主控模块

学生基本信息增删改查

学生出勤信息增删改查

学生获奖信息增删改查

学生基本信息增删改查

学生家庭信息增删改查

图2-1 系统组成框图

（2）系统流程图：图2-2

学生信息管理系统

登录

教师

管理员

个人基础信息

个人家庭信息

个人获奖信息

个人出勤信息

学生基础信息

学生家庭信息

学生获奖信息

学生出勤信息

查

增 删 改 查

学生家庭信息

学生获奖信息

学生基础信息

学生

改 查

学生出勤信息

图2-2 学生信息管理系统流程图

## 2.3 模块功能

### 2.3.1 登录功能

（1）管理员登录

进入到登录页面，选择管理员登录选项，输入管理员的管理员用户名和密码即可登录本学生信息登陆系统。

所需数据为管理员用户名和密码。

（2）教师登录

进入到登录页面，选择教师登录选项，输入教师的教师用户名和密码即可登录本学生信息登陆系统。

所需数据为教师用户名和密码。

（3）学生登录

进入到登录页面，选择学生登录选项，输入学生的学生用户名和密码即可登录本学生信息登陆系统。

所需数据为学生用户名和密码（用户名为学生姓名，密码为学生学号）。

### 2.3.2 新增学生信息

（1）新增学生基础信息

管理员权限可新增学生基础信息，包括学生学号、学生姓名、学生性别、学生年龄以及学生专业。

在学生基础信息页面点击添加信息按钮，输入对应数据即可添加学生基础信息到数据库。

教师和学生无权新增学生基础信息。

（2）新增学生家庭信息

管理员权限可新增学生家庭信息，包括学生学号、学生姓名、学生家庭地址、学生家庭人口数量以及学生家庭电话。

需要注意的是，添加的学生学号以及姓名需要在拥有学生基础信息的基础上，否则添加无效失败。

在学生家庭信息页面点击添加信息按钮，输入对应数据即可添加学生基础信息到数据库。

教师和学生无权新增学生家庭信息

（3）新增学生获奖信息

管理员权限可新增学生家庭信息，包括学生学号、学生姓名、学生获奖名称、学生获奖时间以及学生获奖备注。

需要注意的是，添加的学生学号以及姓名需要在拥有学生基础信息的基础上，否则添加无效失败。

在学生获奖情况页面点击添加信息按钮，输入对应数据即可添加学生基础信息到数据库。

教师和学生无权新增学生获奖信息。

（4）新增学生考勤信息

管理员权限可新增学生家庭信息，包括学生学号、学生姓名、学生缺勤次数、学生缺勤处分以及学生缺勤备注。

需要注意的是，添加的学生学号以及姓名需要在拥有学生基础信息的基础上，否则添加无效失败。

在学生考勤情况页面点击添加信息按钮，输入对应数据即可添加学生基础信息到数据库。

教师和学生无权新增学生考勤信息。

### 2.3.3 删除学生信息

（1）删除学生基础信息

管理员权限可以删除学生基础信息。

需要注意，删除学生基础信息要确保要删除的学生在其他信息（包括学生家庭信息、学生获奖信息和学生考勤信息）任一信息系统里没有该学生的对应信息，方可删除，否则删除失败。

教师和学生无权删除学生基础信息。

（2）删除学生家庭信息

管理员权限可以删除学生家庭信息。

点击家庭信息页面“操作”中的删除按钮即可。

教师和学生无权删除学生家庭信息。

（3）删除学生获奖信息

管理员权限可以删除学生获奖信息。

点击获奖信息页面“操作”中的删除按钮即可。

教师和学生无权删除学生获奖信息。

（4）删除学生出勤信息

管理员权限可以删除学生出勤信息。

点击出勤信息页面“操作”中的删除按钮即可。

教师和学生无权删除学生出勤信息。

### 2.3.4修改学生信息

（1）修改学生基础信息

管理员权限可以修改所有学生基础信息。

学生权限可以修改学生个人的非关键基础信息（包括学生个人的年龄、性别、专业）。

教师权限无权修改学生基础信息。

（2）修改学生家庭信息

管理员权限可以修改所有学生家庭信息。

学生权限可以修改学生个人的非关键家庭信息（包括学生个人的家庭地址、人口数量、家庭电话）。

教师权限无权修改学生家庭信息。

（3）修改学生获奖信息

管理员权限可以修改所有学生获奖信息。

学生权限可以修改学生个人的非关键获奖信息（包括学生个人的获奖名称、获奖时间、获奖备注）。

教师权限无权修改学生获奖信息。

（4）修改学生出勤信息

管理员权限可以修改所有学生出勤信息。

教师和学生无权修改学生出勤信息。

### 2.3.5 查询学生信息

（1）查询学生基础信息

管理员和教师权限可以查询学生所有基础信息。

学生只可看到个人的基础信息。

（2）查询学生家庭信息

管理员和教师权限可以查询学生所有家庭信息。

学生只可看到个人的家庭信息。

（3）查询学生获奖信息

管理员和教师权限可以查询学生所有获奖信息。

学生只可看到个人的获奖信息。

（4）查询学生出勤信息

管理员和教师权限可以查询学生所有出勤信息。

学生只可看到个人的出勤信息。

# 3 交互界面设计

## 3.1 控件介绍

控件是用户与之交互以输入或操作数据的对象，是具有用户界面的组件。

本学生信息管理系统主要使用了一下控件：input文本输入，form表单，button按钮，radio单选按钮，table表格等。

除了以上一些基础常见的html控件等，本系统采用了BootStrap(web框架)前端开发框架。它是基于HTML、CSS、JavaScript开发的简洁、直观、强悍的前端开发框架，使得Web开发更快捷。BootStrap提供了优雅的HTML和CSS规范。包括了丰富的Web组件。 另外，使用了BootStrap支持的Glyphicons字体图标，使得页面更加简洁美观。

本系统主要有两个界面。一是系统登录页面，如图3-1。



图3-1 系统登陆界面

二是信息管理系统主界面，如图3-2。



图3-2 系统主界面

## 3.2 Form表单

为了用户输入数据并向服务器端传送，本系统使用了大量的表单控件。如图3-3、图3-4。



图3-3 登录界面From表单



图3-4 添加学生信息界面Form表单

本系统一共使用了17个form表单，方便用户输入各个页面各个功能需要的各项数据，方便使之向服务器端传输数据。

Form表单中可以使用文本框、复选框、按钮以及单选框等组件，满足不同功能的需求，非常强大且简便。

使用BootStrap提供的表单组件可以使表单界面更加美观，为每个input标签嵌套上class属性form-group，让各个组件之间井然有序。

## 3.3 Button按钮

在一个网页中，button按钮使必不可少的。

它为用户提供按钮，以便完成各种功能：跳转页面、确定信息等等。

本系统使用了大量的按钮，来使用户添加信息、确定信息、查询信息等。

如图3-5



图3-5 各界面不同功能按钮

使用BootStrap提供的CSS美化，添加属性class="btn btn-primary"，使按钮样式为主色按钮（蓝色），与整个网页页面格调相匹配，达到色调样式格调一致的效果，符合大部分人的主观审美。

## 3.4 Table表格

本系统包含多中数据：学生信息、家庭信息、获奖信息、出勤信息。为

了使数据更直观，使用table表格将数据可视化。并且使用户方便对数据信息进行各种操作。

BootStrap相应提供了table表格组件，并拥有丰富的属性支持以供选择。在此选择最基础样式，简约直观，与整体格调相协调。

Table表格能够将数据完美呈现在页面之中，使数据可视化。

Table表格可以很容易建立起结构化的界面，通过table自身的参数定义，能把页面布局很快定义成所需要的效果。

本系统所有数据都使用table表格呈现。

如图3-6所示。



图3-6 Table表格中学生信息数据

## 3.5 Radio单选框

Radio单选框为用户提供了可选择的功能，本系统一共有两种功能类似的radio单选框组件。

一是用户在登陆界面可以选择管理员、教师和学生其一选项进行登录，以便获得不同权限。如图3-7



图3-7 登陆界面radio单选框

二是在查询时，用户可以选择使用学生学号或姓名进行查询。如图3-8



图3-8 查询信息radio单选框

Radio单选框使用简洁，小组件实现大功能。

## 3.6 导航选项

本系统为学生信息管理系统，期中学生信息有学生基本信息、学生家庭信息、学生获奖信息以及学生出勤信息。为了使用户方便跳转查看各类信息，本系统在页面左侧设置了一个导航栏，用户点击相应信息即可查看对应信息并对其进行操作。

设置导航栏可以让用户操作方便。本系统导航菜单栏使用a标签href属性进行跳转各自页面，简洁方便快捷。并使用Bootstrap中的class="nav nav-pills nav-stacked"来进行样式修改，使导航栏融入整个页面，与之格调相同。

使用如图3-9。



图3-9 左侧系统菜单导航栏

# 4 功能详细设计

学生信息管理系统基于SSM框架开发，是由Spring和MyBatis两个开源框架整合而成，完成此信息管理系统web项目的开发非常适合。本系统大致结构如下：

①Pojo层：对应的数据库表的实体类，如图4-1

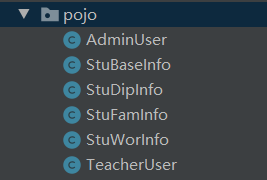
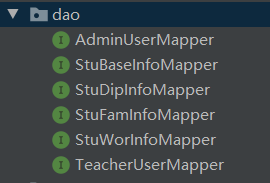


图4-1 pojo层实体类

②Dao层：数据访问层，负责与数据库进行访问以及对表、对实体类实现增删改查操作，如图4-2



4-2 dao层接口类

③service层：主要负责业务模块的逻辑设计，为service接口和serviceImpl实现类。调用Dao层实现操作。如图4-3

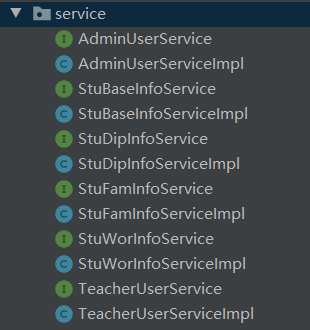
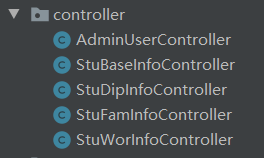


图4-3 service层接口类和实现类

④controller层：负责具体的业务模块流程控制，如图4-4



4-4 controller层实现类

本系统对象的调用流程：

数据库

Dao

Service

前端JSP

Controller

系统所有框架构成如下：如图4-5

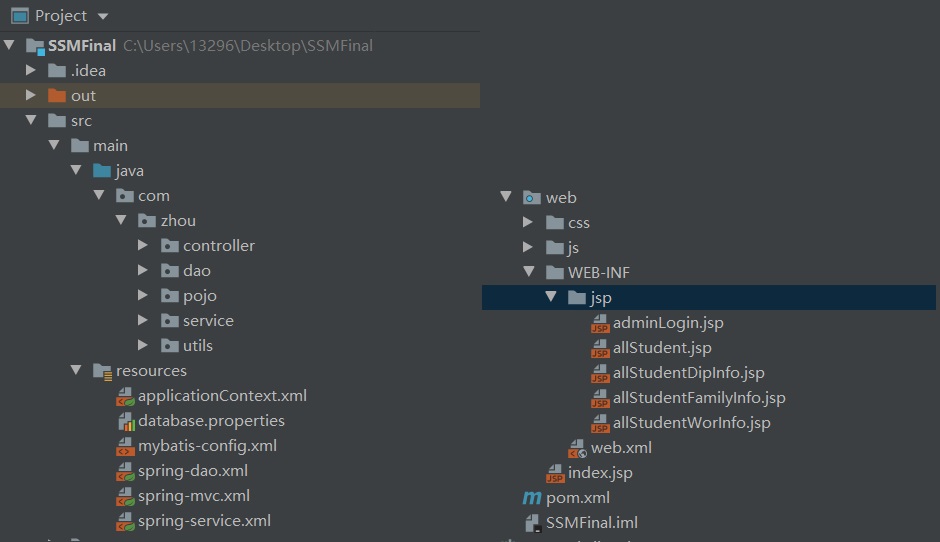


图4-5 本系统框架结构

## 4.1 数据库

4.1.1 数据库名称及描述

本系统创建六个数据库，以下是对其描述：

（1）管理员表：用户名，密码

（2）教师表：用户名,密码

（3）学生基础信息表：学号，姓名，性别，年龄，专业

（4）学生家庭信息表：学号，姓名，地址，人口，电话

（5）学生获奖信息表：学号，姓名，获奖名称，时间，备注

（6）学生出勤信息表：学号，姓名，缺勤次数，处分，备注

4.1.2 数据库ER图

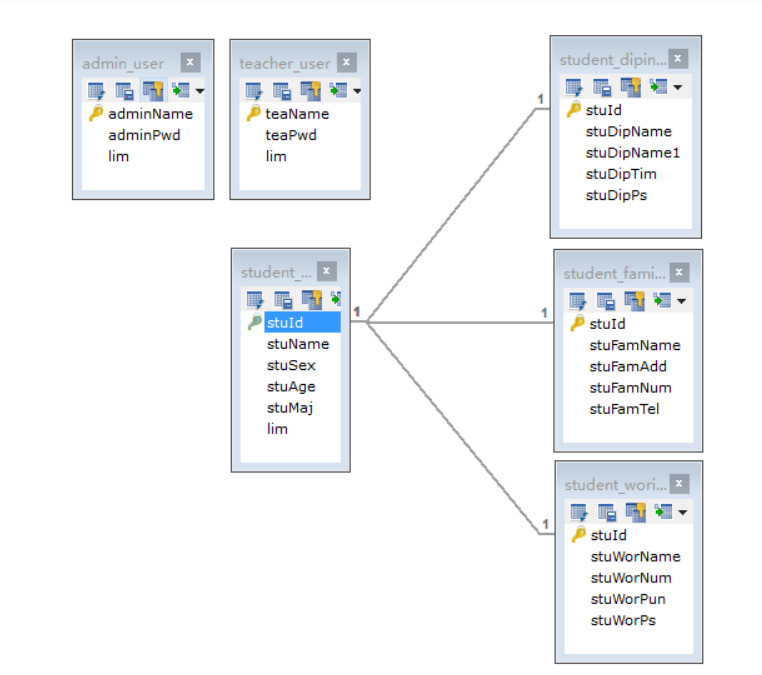


图4-6左侧系统菜单导航栏

4.1.3 数据库表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| 用户名 | varchar | 50 | 否 | 无 | 主键，管理员用户名 |
| 密码 | varchar | 50 | 否 | 无 | 管理员密码 |
| 权限 | int |  | 否 | 1 | 管理员权限 |

表4-7 管理员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| 用户名 | varchar | 50 | 否 | 无 | 主键，教师用户名 |
| 密码 | varchar | 50 | 否 | 无 | 教师密码 |
| 权限 | int |  | 否 | 0 | 教师权限 |

表4-8教师表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| 学号 | int |  | 否 | 无 | 主键，学生学号 |
| 姓名 | varchar | 50 | 否 | 无 | 学生姓名 |
| 性别 | varchar | 50 | 否 | 无 | 学生性别 |
| 年龄 | varchar | 50 | 否 | 无 | 学生年龄 |
| 专业 | varchar | 50 | 否 | 无 | 学生专业 |
| 权限 | int |  | 否 | -1 | 学生权限 |

表4-6 学生基础信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| 学号 | int |  | 否 | 无 | 外键，学生学号 |
| 姓名 | varchar | 50 | 否 | 无 | 学生姓名 |
| 家庭地址 | varchar | 50 | 否 | 无 | 学生家庭地址 |
| 家庭人口数量 | int |  | 否 | 无 | 学生家庭人口数量 |
| 家庭电话 | varchar | 50 | 否 | 无 | 11位 |

表4-10 学生家庭信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| 学号 | int |  | 否 | 无 | 外键，学生学号 |
| 姓名 | varchar | 50 | 否 | 无 | 学生姓名 |
| 获奖名称 | varchar | 50 | 否 | 无 | 获奖名称 |
| 获奖时间 | date |  | 否 | 无 | 注意date类型 |
| 获奖备注 | varchar | 50 | 否 | 无 | 获奖备注 |

表4-11 学生获奖信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| 学号 | int |  | 否 | 无 | 外键，学生学号 |
| 姓名 | varchar | 50 | 否 | 无 | 学生姓名 |
| 缺勤次数 | int |  | 否 | 无 | 缺勤次数 |
| 缺勤处分 | varchar |  | 否 | 无 | 缺勤处分 |
| 缺勤备注 | Varchar | 50 | 否 | 无 | 缺勤备注 |

图4-12 学生考勤信息

## 4.2 数据库连接

本系统数据库使用Mybatis，是一个基于Java的持久层框架。它避免了几乎所有的JDBC代码和手动设置参数以及获取结果集，使用简单的XML或注解用于配置和原始反射。

它非常灵活，解除SQL与程序代码的耦合，通过提供DAO层，将业务逻辑和数据访问逻辑分离，使系统的设计更清晰，更易维护，更易单元测试，提高了可维护性。

Mybatis通过配置文件创建sqlsessionFactory，sqlsessionFactory根据配置文件。

配置文件来源于两个方面:一个是xml，一个是Java中的注解，获取sqlSession。SQLSession包含了执行sql语句的所有方法，可以通过SQLSession直接运行映射的sql语句，完成对数据的增删改查和事物的提交工作，用完之后关闭SQLSession。

数据库连接文件database.properties主要代码：

jdbc.driver=com.mysql.jdbc.Driver

//在使用MySql8.0及以上版本时，需要设置时区，即增加一个时区的配置：

//&serverTimezone=Asia/Shanghai;

jdbc.url=jdbc:mysql://localhost:3306/student\_info?useSSL=false&useUnicode=true&characterEncoding=utf8&allowMultiQueries=true

//实现支持多条SQL语句同时执行

jdbc.username=root

jdbc.password=981206

在mybatis-config.xml配置文件中，其主要配置代码为：

<configuration>

<settings>

<setting name="logImpl" value="STDOUT\_LOGGING"/>

</settings>

<!--配置数据源。交给Spring去做-->

<!--取别名-->

<typeAliases>

<package name=""/>

</typeAliases>

<!-- 配置分页插件 -->

<plugins>

<plugin interceptor="com.github.pagehelper.PageInterceptor">

<!-- 支持Oracle，MySQL，SQLite，MariaDB，Hsqldb，PostgreSQL -->

<property name="dialect" value="com.github.pagehelper.PageHelper"/>

</plugin>

</plugins>

</configuration>

在次配置文件中，需要配置一个分页插件PageHelper，为以后的分页需求做准备。

为dao层创建配置文件spring-dao.xml，配置整合MyBatis。其步骤主要为：

1. 关联数据库文件
2. 创建数据库连接池：

这一步使用c3p0连接池，c3p0实现自动化操作，自动加载配置文件，并且设置到对象中，对比dbcp以及其他连接池，c3p0非常方便。

1. 配置连接池属性以及配置c3p0连接池的私有属性。
2. 配置SqlSessionFactory对象：

注入数据库连接池、绑定MyBatis全局配置文件mybatis-config.xml。

1. 配置扫描Dao接口包，动态实现Dao接口注入到spring容器中：

注入sqlsessionfactory以及给出需要扫描的Dao接口包。

其主要代码为：

<!-- 配置整合mybatis -->

<!-- 1.关联数据库文件 -->

<context:property-placeholder location="classpath:database.properties"/>

<!-- 2.数据库连接池 -->

<!--数据库连接池

dbcp 半自动化操作 不能自动连接

c3p0 自动化操作（自动的加载配置文件 并且设置到对象里面）

-->

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">

<!-- 配置连接池属性 -->

<property name="driverClass" value="${jdbc.driver}"/>

<property name="jdbcUrl" value="${jdbc.url}"/>

<property name="user" value="${jdbc.username}"/>

<property name="password" value="${jdbc.password}"/>

<!-- c3p0连接池的私有属性 -->

<property name="maxPoolSize" value="30"/>

<property name="minPoolSize" value="10"/>

<!-- 关闭连接后不自动commit -->

<property name="autoCommitOnClose" value="false"/>

<!-- 获取连接超时时间 -->

<property name="checkoutTimeout" value="10000"/>

<!-- 当获取连接失败重试次数 -->

<property name="acquireRetryAttempts" value="2"/>

</bean>

<!-- 3.配置SqlSessionFactory对象 -->

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

<!-- 注入数据库连接池 -->

<property name="dataSource" ref="dataSource"/>

<!-- 绑定MyBaties全局配置文件:mybatis-config.xml -->

<property name="configLocation" value="classpath:mybatis-config.xml"/>

</bean>

<!-- 4.配置扫描Dao接口包，动态实现Dao接口注入到spring容器中 -->

<!--解释 ：https://www.cnblogs.com/jpfss/p/7799806.html-->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<!-- 注入sqlSessionFactory -->

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>

<!-- 给出需要扫描Dao接口包 -->

<property name="basePackage" value="com.zhou.dao"/>

</bean>

</beans>

在这一操作上，我主要对比了c3p0、dbcp、druid以及HikariCp这四个主流连接池。

HikariCP连接池性能强悍、字节码精简；

dbcp集成性能好；

druid拥有强大的监控性能、功能全面。

然后c3p0的功能简单易用，稳定性好是我首选的主要原因，虽然有性能上的稍微劣势，不过此学生信息查询系统对比性能上不需要太强。

Mybatis使用的具体流程：如图4-13

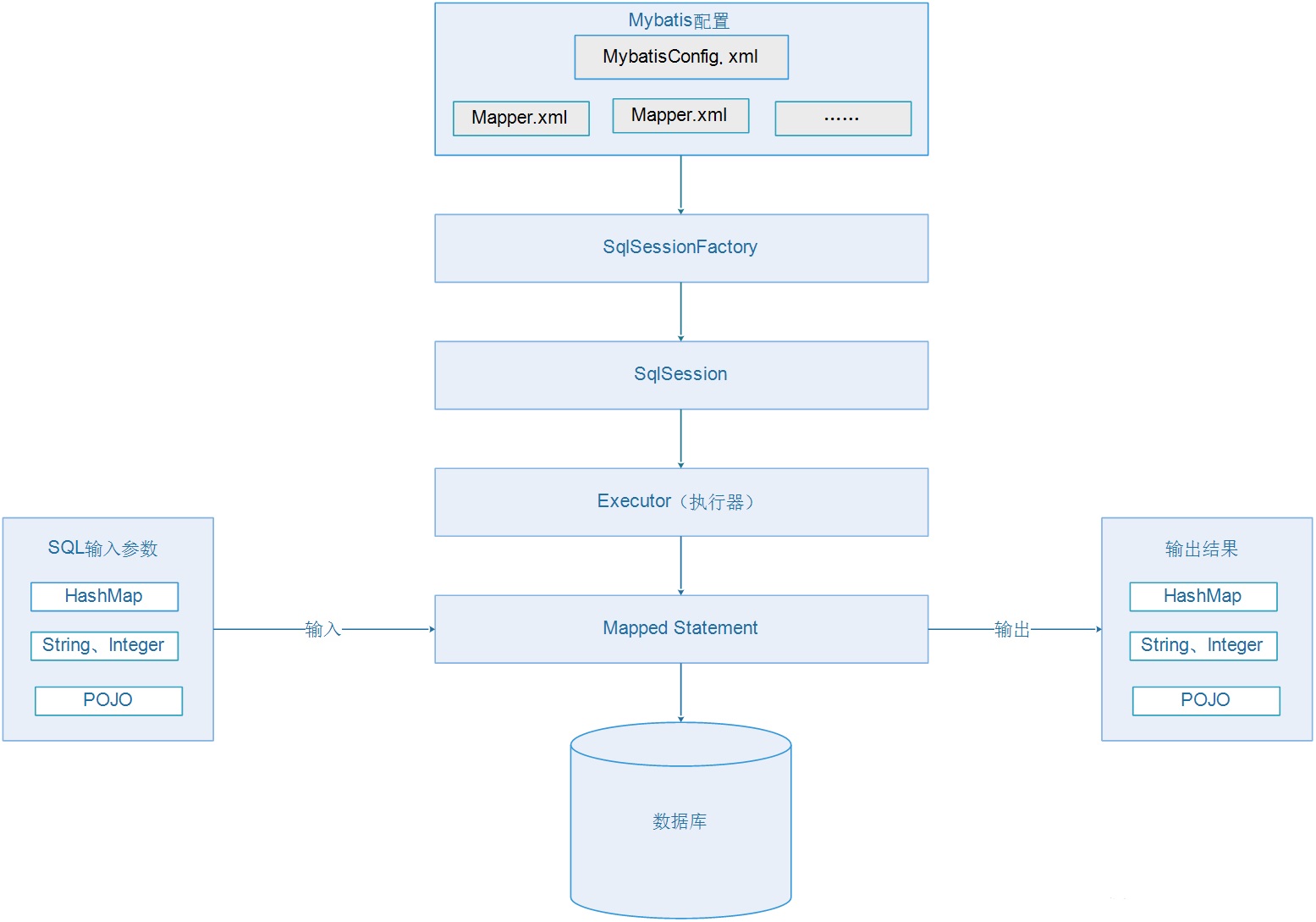


图4-13 MyBatis工作流程图

## 4.3 用户的登录

在用户登录界面，提供了用户三个登录选项，分别是管理员、教师和学生，再输入正确的用户名和密码以及选择登录用户选择后，即可进入学生信息管理系统并拥有不同权限。

管理员拥有操作所有功能的权限，教师只限于查看所有的学生信息，而学生本人只限查看个人的学生信息。

在数据库中，管理员表、教师表以及学生表中都设置了lim字段，以int类型数字1、0、-1代表不同权限。

在登陆时，选择单选框中的管理员、教师、学生选项后，并且在正确输入用户名和密码后，登陆界面会提交表单到controller层中进行判断用户以及对数据库的查询。

主要代码为：1代表管理员，0代表教师，-1代表学生

@RequestMapping("/adminLogin")

public String login(@RequestParam(value = "adminName", required = false) String adminName, @RequestParam(value = "adminPwd") String adminPwd, Model model, HttpServletRequest request, HttpServletResponse response) {

String optRadio = request.getParameter("optRadio");

if (optRadio.equals("1")) {

try {

AdminUser adminUser1 = adminUserService.queryUser(adminName);

if (adminUser1.getAdminName().equals(adminName) && adminUser1.getAdminPwd().equals(adminPwd)) {

model.addAttribute("name", adminUser1.getAdminName());

model.addAttribute("lim", adminUser1.getLim());

return "redirect:/student/allStudent";//重定向allBook请求

} else {

model.addAttribute("error\_login", "用户名或密码错误");

return "adminLogin";

}

} catch (Exception e) {

model.addAttribute("error\_login", "用户名或密码错误");

return "adminLogin";

}

} else if (optRadio.equals("0")) {

try {

TeacherUser teacherUser = teacherUserService.queryUser(adminName);

if (teacherUser.getTeaName().equals(adminName) && teacherUser.getTeaPwd().equals(adminPwd)) {

model.addAttribute("name", teacherUser.getTeaName());

model.addAttribute("lim", teacherUser.getLim());

return "redirect:/student/allStudent";//重定向allBook请求

} else {

model.addAttribute("error\_login", "用户名或密码错误");

return "adminLogin";

}

} catch (Exception e) {

model.addAttribute("error\_login", "用户名或密码错误");

return "adminLogin";

}

} else if (optRadio.equals("-1")) {

try {

List<StuBaseInfo> stuBaseInfos = stuBaseInfoService.queryStudentById(Integer.parseInt(adminPwd));

StuBaseInfo stuBaseInfo = stuBaseInfos.get(0);

if (String.valueOf(stuBaseInfo.getStuId()).equals(adminPwd) && stuBaseInfo.getStuName().equals(adminName)) {

model.addAttribute("id", stuBaseInfo.getStuId());

model.addAttribute("name", stuBaseInfo.getStuName());

model.addAttribute("lim", stuBaseInfo.getLim());

return "redirect:/student/allStudent";//重定向allBook请求

} else {

model.addAttribute("error\_login", "用户名或密码错误");

return "adminLogin";

}

} catch (Exception e) {

model.addAttribute("error\_login", "用户名或密码错误");

return "adminLogin";

}

}

return "adminLogin";

}

在对数据库的操作mapper层，查询代码为：

@Mapper

public interface AdminUserMapper {

@Select("select \* from student\_info.admin\_user where `adminName`=#{adminName}")

AdminUser queryUser(@Param("adminName") String adminName);

}

这里以对管理员表的查询为例，mapper注解的使用将摒弃对xml文件的配置，其他查询语句与之类似。

在查询后，controller层将返回查询的结果以及将lim字段的值返回给登录界面进行判断，使用jQuery来接收url地址中的用户名和以数字为表示的权限值。

主要代码为：

{

var query = window.location.search.substring(1);

var vars = query.split("&");

for (var i = 0; i < vars.length; i++) {

var pair = vars[i].split("=");

if (pair[0] == variable) {

return pair[1];

}

}

return (false);

}

const limVal = getQueryVariable("lim");

if (limVal === "0") {

$(function () {

$("#addBtn").addClass("hidden");//增加隐藏

//$("#selBtn").addClass("hidden");

$('table tr').find('td:eq(5)').hide();

$("#opr").hide();

});

} else if (limVal === "-1") {

$(function () {

$("#addBtn").addClass("hidden");//增加隐藏

$("#selBtn").addClass("hidden");

$("#exBtn").addClass("hidden");

$("#DeleteBtn").addClass("hidden");

$("#upStuName").attr("readonly", "true");

$("#noName").removeClass("hidden");

// $('table tr').find('td:eq(5)').hide();

// $("#opr").hide();

});

}

通过判断url中的lim字段来进行页面按钮的隐藏，比如如果lim为-1，将会在页面中隐藏添加信息按钮、查询信息按钮、删除按钮等，实现对权限的控制操作。

## 4.4 学生信息的增加

在每个学生信息的页面中，都设置有添加信息按钮，不通页面添加对应不同信息。再点击后，将会弹出Bootstrap提供的模态框进行数据的输入以及添加。

模态框将不会跳转页面，而是弹出覆盖再父窗体上的子窗体，可以在不离开父窗体的情况下进行互动。以此作为添加界面非常合适，如图4-14

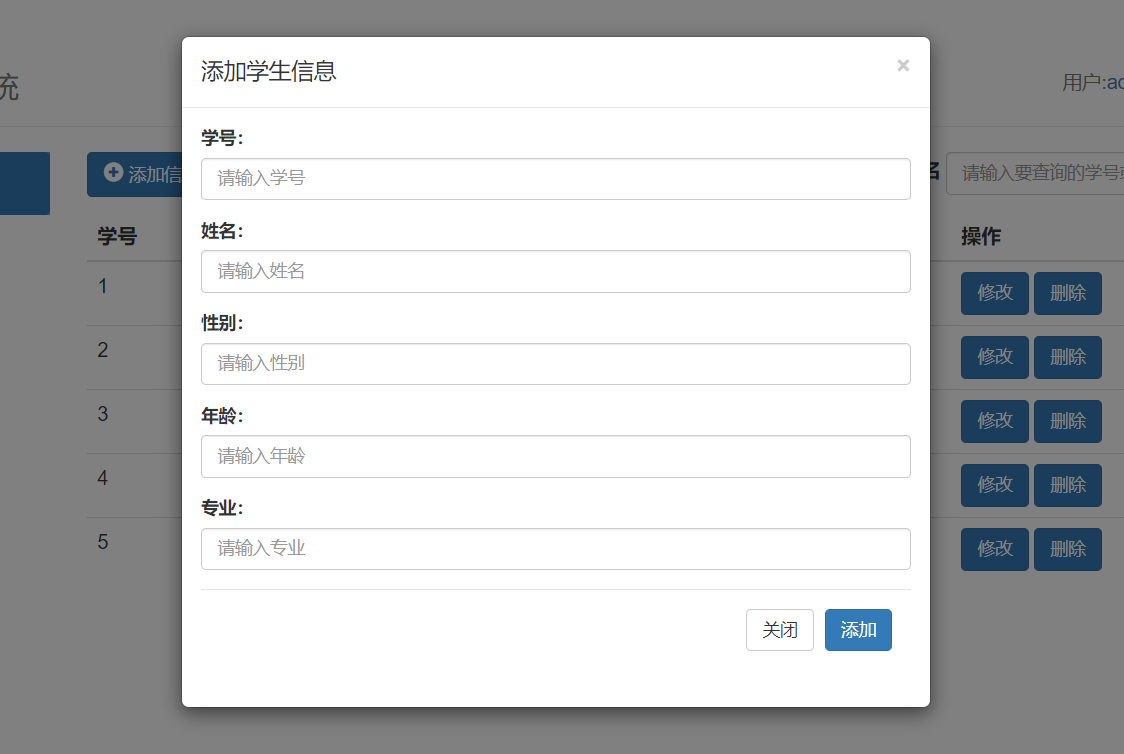


图4-14 添加信息模态框

以添加学生基础信息为例，在输入正确的信息后，点击添加将提交表单到controller层，由controller层对请求进行处理：

主要代码为：

//增加一个学生

@RequestMapping("/addStudent")

public String addStudent(StuBaseInfo stuBaseInfo, Model model, HttpServletRequest request, HttpServletResponse response) {

String name = request.getParameter("name");

String lim = request.getParameter("lim");

try {

stuBaseInfoService.addStudent(stuBaseInfo);

model.addAttribute("error\_Student", "添加成功");

model.addAttribute("name", name);

model.addAttribute("lim", lim);

} catch (Exception e) {

model.addAttribute("error\_Student", "添加失败");

model.addAttribute("name", name);

model.addAttribute("lim", lim);

}

return "redirect:/student/allStudent";

}

添加成功后将会返回主界面，有主界面查询并显示数据。添加失败将会把添加失败字符串传递给主界面，将会在主界面接收并显示，以提示用户输入有误，添加失败。

在mapper层，主要代码为：

//增加一个学生  
@Insert(" insert into student\_info.student\_baseinfo " +  
 "(`stuId`,`stuName`,`stuSex`,`stuAge`,`stuMaj`) values " +  
 "(#{stuId},#{stuName},#{stuSex},#{stuAge},#{stuMaj})")  
int addStudent(StuBaseInfo stuBaseInfo);

在每个页面增加完后，都会自动返回主界面并且自动刷新数据和显示，符合基本信息添加以及用户的使用习惯，提高用户使用体验。

## 4.5 学生信息的修改

对于信息的修改，将修改按钮设置在表格中，点击对应的学生信息即可修改。修改的操作难点在于点击修改后，需要把选中的数据填充到填出的修改模态框中，方便用户进行修改。

修改模态框和添加模态框如出一辙。

在思考许多方案后，我选择了使用jQuery来进行数据的选择和填充，这样减少了后台代码量，不经过后台而仅仅依靠纯html标签来进行操作，可以加快浏览器相应速度。

主要代码为：

function update(obj) {//点击修改按钮，给模态框加载信息并且弹出模态框

var tds = $(obj).parent().parent().find('td');

$("#upStuId").val(tds.eq(0).text());

$("#upStuName").val(tds.eq(1).text());

$("#upStuSex").val(tds.eq(2).text());

$("#upStuAge").val(tds.eq(3).text());

$("#upStuMaj").val(tds.eq(4).text());

$('#myModalUpdate').modal('show');

}

通过寻找table表格中的td标签来依次对修改模态框中的输入框进行数据填充。点击后填充效果如图4-15（点击修改学号为1的学生）

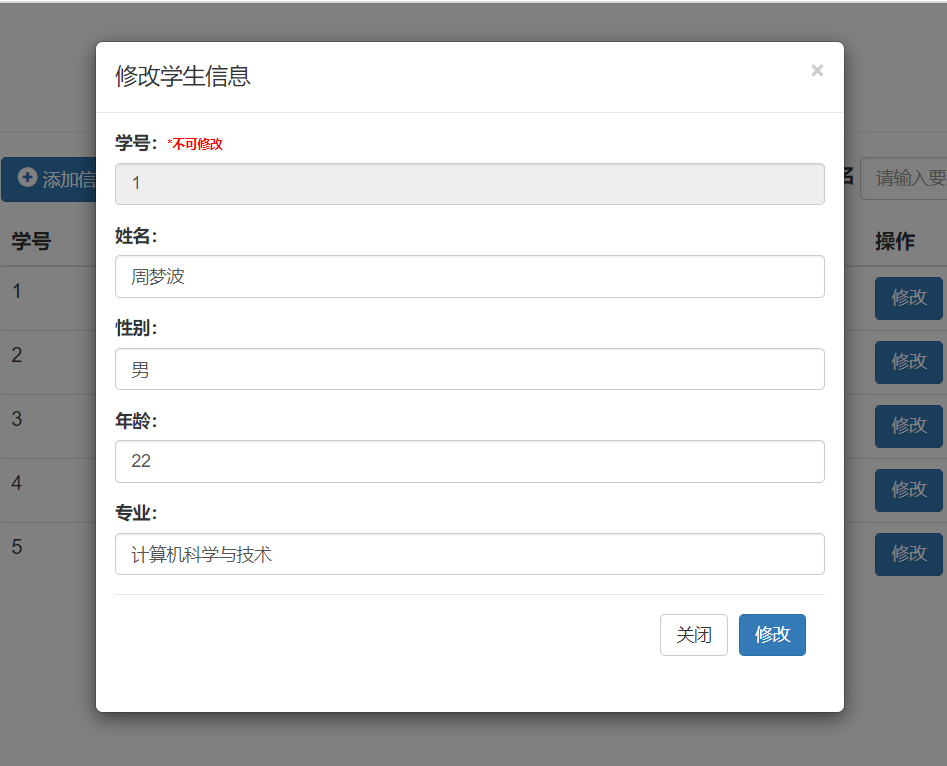


图4-15 修改学号为1的学生信息

在填写修改后的数据后（学生学号不可修改）后，数据将会提交给后台进行处理修改。

主要代码为：

//修改学生信息

@RequestMapping("/updateStudent")

public String updateStudent(StuBaseInfo stuBaseInfo, Model model, HttpServletRequest request, HttpServletResponse response) {

String name = request.getParameter("name");

String lim = request.getParameter("lim");

try {

int i = stuBaseInfoService.updateStudent(stuBaseInfo);

if (i > 0) {

model.addAttribute("error\_Student", "修改成功");

model.addAttribute("name", name);

model.addAttribute("lim", lim);

} else {

model.addAttribute("error\_Student", "修改失败");

model.addAttribute("name", name);

model.addAttribute("lim", lim);

}

} catch (Exception e) {

model.addAttribute("error\_Student", "修改失败");

model.addAttribute("name", name);

model.addAttribute("lim", lim);

}

return "redirect:/student/allStudent";

}

## 4.6 学生信息的删除

学生信息的删除和修改一样，都需要进行数据的填充。点击删除后，会弹出提示框，让用户二次确定，以保证不是误触。

删除依旧是使用jQuery进行学生学号填充，用户将会看到带有学生学号的提示框来确定删除该学生信息。

如图4-16所示



图4-16 删除确认框

需要注意的是，由于数据库的设计，学生家庭信息、获奖信息和出勤信息的学号设置为外键，是学生基本信息主键学号构成的外键，所以在删除学生基础信息时，需要确保该学号学生在其他表中没有信息。

删除由controller层执行，主要代码为：

@RequestMapping("/deleteStudent")

public String deleteStudent(@RequestParam(value = "delStuId", required = false) String stuId, Model model, HttpServletRequest request, HttpServletResponse response) {

String name = request.getParameter("name");

String lim = request.getParameter("lim");

try {

int id = Integer.parseInt(stuId);

stuBaseInfoService.deleteStudentById(id);

model.addAttribute("error\_Student", "删除成功");

model.addAttribute("name", name);

model.addAttribute("lim", lim);

} catch (Exception e) {

model.addAttribute("error\_Student", "删除失败");

model.addAttribute("name", name);

model.addAttribute("lim", lim);

}

return "redirect:/student/allStudent";

}

## 4.7 学生信息的查询

在主界面加载后，将会请求查询数据库所有学生信息并返回在table表格中显示。只要进行任意操作：学生信息的增肌、删除、修改后都会返回主界面并查询所有信息后显示。

本系统在管理员和教师界面，提供了学生信息的查询功能。并提供了学生学号和学生姓名两种不同的查询方式，提供多元化查询。如图4-17。



图4-17 学生信息的查询

查询以表单提交，主要代码为：

<form id="selBtn" class="form-inline"

action="${pageContext.request.contextPath}/student/queryStudent?name=${param.name}&lim=${param.lim}"method="post" style="float: right">

<span style="color: red;font-weight: bold">

${error\_queryStudent}

</span>

<div class="form-group" style="font-size: 16px">

<div class="radio-inline">

<label><input type="radio" name="optRadio" value="1" checked>

学号

</label>

</div>

<div class="radio-inline">

<label><input type="radio" name="optRadio" value="0">姓名</label>

</div>

</div>

<input type="text" name="queryStuById" class="form-control" placeholder="请输入要查询的学号或姓名"

Required oninvalid="setCustomValidity('请输入学号或姓名')" oninput="setCustomValidity('')">

<input type="submit" value="查询" class="btn btn-primary">

</form>

提交的radio单选用“1”和“0”表示使用的查询方式，后台接收并判断调用不同的查询语句。

主要代码为：

//查一个学生（ById或name

@RequestMapping("/queryStudent")

public String toUpdateStudent(@RequestParam(value = "queryStuById", required = false) String queryStuById, Model model, HttpServletRequest request) {

String optRadio = request.getParameter("optRadio");

if (optRadio.equals("1")) {

try {

int id = Integer.parseInt(queryStuById);

List<StuBaseInfo> list = stuBaseInfoService.queryStudentById(id);

if (list.size() == 0) {

model.addAttribute("error\_queryStudent", "查询无果");

} else {

model.addAttribute("error\_queryStudent", "查询成功");

model.addAttribute("list", list);

}

} catch (Exception e) {

model.addAttribute("error\_queryStudent", "查找无果");

}

} else if (optRadio.equals("0")) {

try {

List<StuBaseInfo> list = stuBaseInfoService.queryStudentByName(queryStuById);

if (list.size() == 0) {

model.addAttribute("error\_queryStudent", "查询无果");

} else {

model.addAttribute("error\_queryStudent", "查询成功");

model.addAttribute("list", list);

}

} catch (Exception e) {

model.addAttribute("error\_queryStudent", "查找无果");

}

}

return "allStudent";

}

查询后返回StuBaseInfo类型（不通页面类型不同）的List集合，在界面显示。

在查询所有信息并在table表格显示中，本系统支持分页设计，将以每页5组学生数据进行自动分页。

分页使用MyBatis支持的PageHelper插件完成。

后台主要代码为：

PageHelper.startPage(pg, 5);

PageInfo<StuBaseInfo> pageInfo = new PageInfo<StuBaseInfo>(list);

model.addAttribute("page", pageInfo);

前端jsp页面进行接收和处理，主要代码为：

<div style="text-align: center">

当前第<a href="">${page.pageNum}</a>页,共<a href="">${page.pages }</a>页,总<a href="">${page.total}</a>条记录

<nav aria-label="Page navigation">

<ul class="pagination">

<li>

<a href="${pageContext.request.contextPath}/student/allStudent?name=${param.name}&lim=${param.lim}&pg=1">首页</a>

</li>

<c:if test="${page.hasPreviousPage}">

<li>

<a href="${pageContext.request.contextPath}/student/allStudent?name=${param.name}&lim=${param.lim}&pg=${page.pageNum-1}">上一页</a>

</li>

</c:if>

<li>

<c:forEach items="${page.navigatepageNums }" var="page\_Nums">

<c:if test="${page\_Nums==page.pageNum }">

<li class="active">

<a href="#">${page\_Nums}</a>

</li>

</c:if>

<c:if test="${page\_Nums!=page.pageNum }">

<li>

<a href="${pageContext.request.contextPath}/student/allStudent?name=${param.name}&lim=${param.lim}&pg=${page\_Nums}">${page\_Nums}</a>

</li>

</c:if>

</c:forEach>

</li>

<c:if test="${page.hasNextPage}">

<li>

<a href="${pageContext.request.contextPath}/student/allStudent?name=${param.name}&lim=${param.lim}&pg=${page.pageNum+1}"

aria-label="Next">下一页</a>

</li>

</c:if>

<li>

<a href="${pageContext.request.contextPath}/student/allStudent?name=${param.name}&lim=${param.lim}&pg=${page.pages}">末页</a>

</li>

</ul>

</nav>

</div>

分页效果如图4-18



图4-18 分页结果显示

## 4.8 学生信息的导出

本系统支持学生信息的导出，用户可以选择以excel格式导出。在每个信息界面，点击导出按钮即可对当前页面信息进行导出下载。

使用jQuery来点击导出按钮并跳转到controller层对应的请求。请求调用utils工具类中的StuInfoToExcel类，主要代码为：

public HSSFWorkbook exStuBaseInfoToExcel(List<StuBaseInfo> stuBaseInfo)

{

HSSFWorkbook wb = new HSSFWorkbook();

HSSFSheet sheet =wb.createSheet("学生基本信息");

HSSFRow row = sheet.createRow(0);

row.createCell(0).setCellValue("学号");

row.createCell(1).setCellValue("姓名");

row.createCell(2).setCellValue("性别");

row.createCell(3).setCellValue("年龄");

row.createCell(4).setCellValue("专业");

//导入数据

for(int i=0;i<stuBaseInfo.size();i++){

row=sheet.createRow(i+1);

StuBaseInfo stuBaseInfo1 = stuBaseInfo.get(i);

row.createCell(0).setCellValue(stuBaseInfo1.getStuId());

row.createCell(1).setCellValue(stuBaseInfo1.getStuName());

row.createCell(2).setCellValue(stuBaseInfo1.getStuSex());

row.createCell(3).setCellValue(stuBaseInfo1.getStuAge());

row.createCell(4).setCellValue(stuBaseInfo1.getStuMaj());

}

return wb;

}

在controller层处理请求。

主要代码为：

@RequestMapping("/exToExcel")

public void exToExcel(HttpServletRequest request,HttpServletResponse response) throws IOException

{

List<StuBaseInfo> list = stuBaseInfoService.queryAllStudent();

stuInfoToExcel = new StuInfoToExcel();

HSSFWorkbook wb = stuInfoToExcel.exStuBaseInfoToExcel(list);

String fileName = "学生基本信息.xls";

fileName = new String(fileName.getBytes("gb2312"),"ISO-8859-1");

response.setContentType("application/vnd.ms-excel");

response.setHeader("Content-disposition", "attachment;filename="+fileName);

OutputStream outputStream = response.getOutputStream();

wb.write(outputStream);

outputStream.flush();

outputStream.close();

}

导出信息下载如图4-19

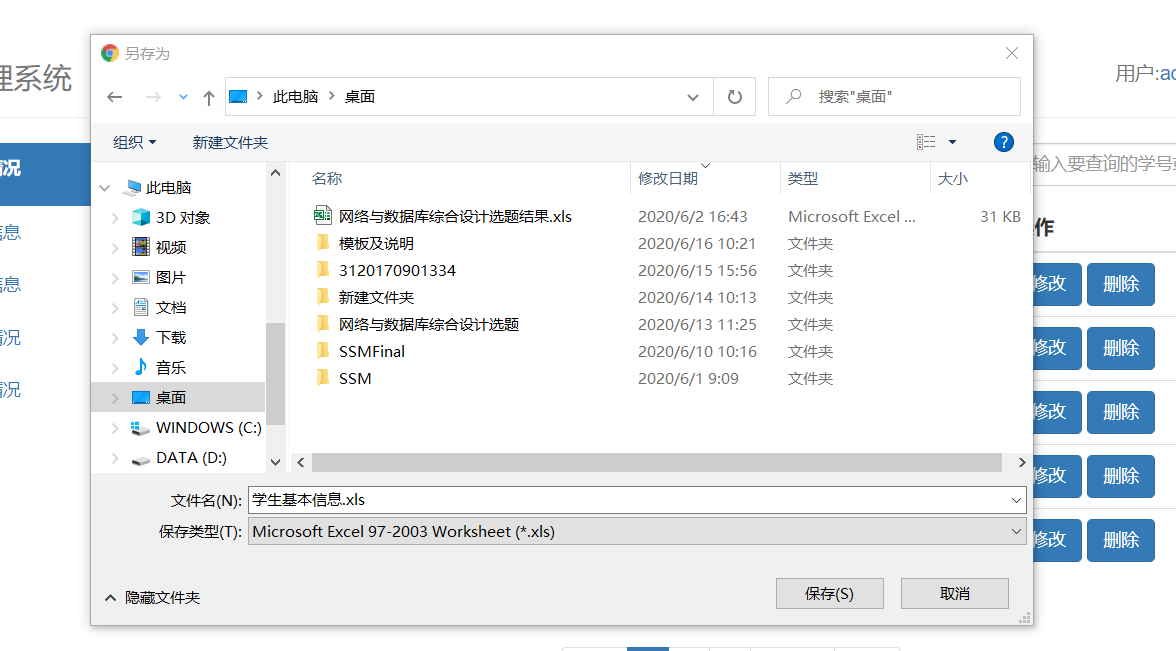


图4-19 导出信息下载

## 4.9 软件使用说明

1、用户登录

输入正确用户名和密码点击登录，图4-20



图4-20 用户登录

若登陆失败，会出现红色提示，用户名或密码错误，如图4-21



图4-21 登陆失败

2、登陆成功，进入学生信息管理系统主界面，图4-22



图4-22 系统主界面

3、管理员添加学生信息，图4-23

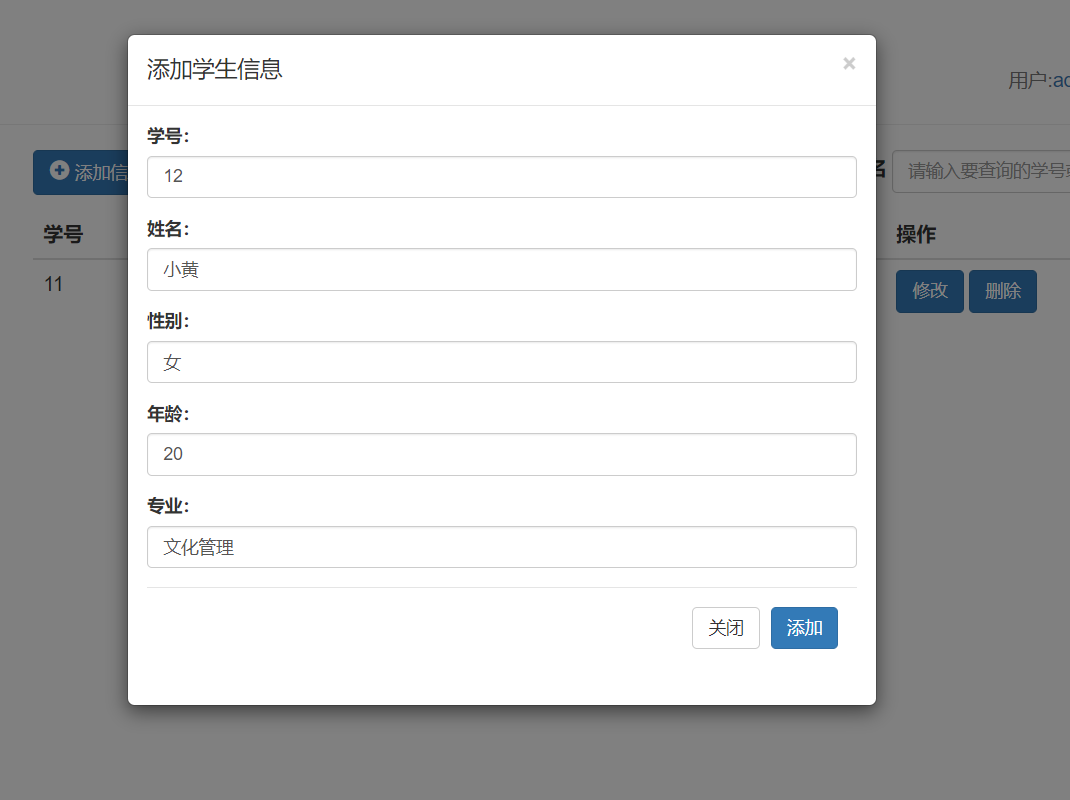


图4-23 添加学生信息

添加成功：出现成功提示，图4-24



图4-24 添加成功

添加失败：出现失败提示，图4-25



图4-25 添加失败

4、管理员修改学生信息，如图4-26修改学号为1的信息，年龄22改为23

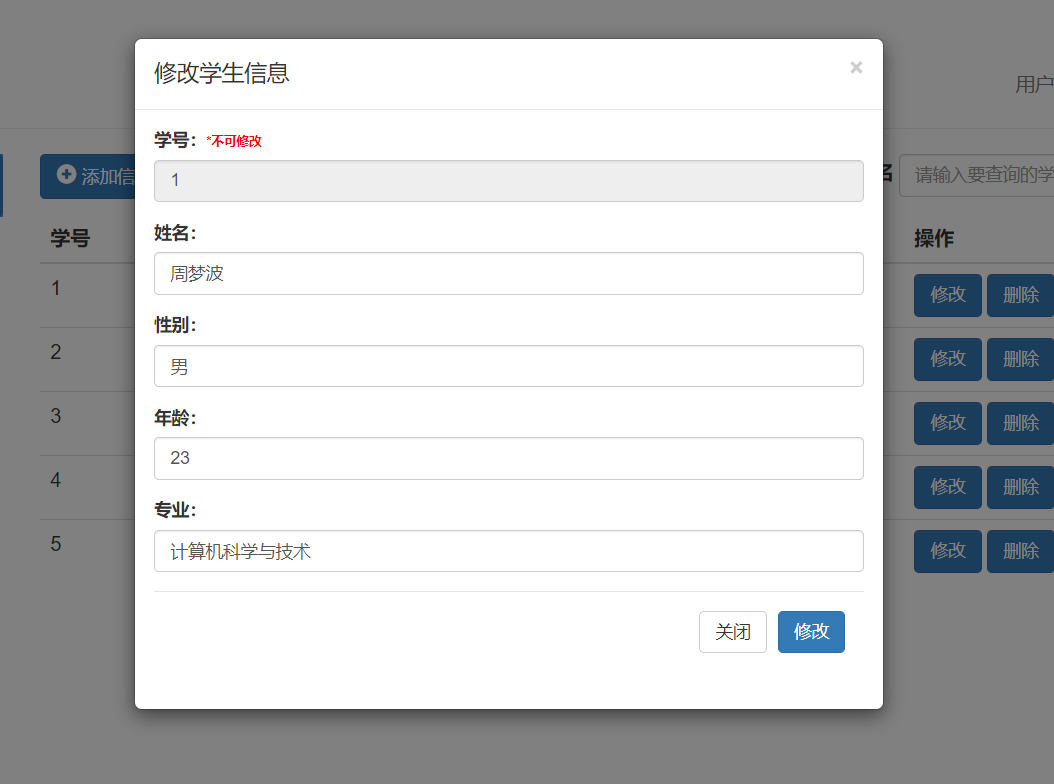


图4-26 修改学生信息

修改成功，出现成功提示，图4-27



图4-27 修改成功

修改失败，出现失败提示，图4-28



图4-28 修改失败

5、管理员删除学生信息，图4-29，删除学生学号为12的学生信息



图4-29 删除学生信息

删除成功，图4-30



图4-30 删除成功

删除失败，图4-31



图4-31 删除失败

6、管理员查询学生信息，图4-32



图4-32 查询学号为1的学生信息

查询成功，图4-33



图4-33 查询成功

查询学号为33的学生信息，失败，图4-34



图4-34 查询失败

7、导出信息，图4-35

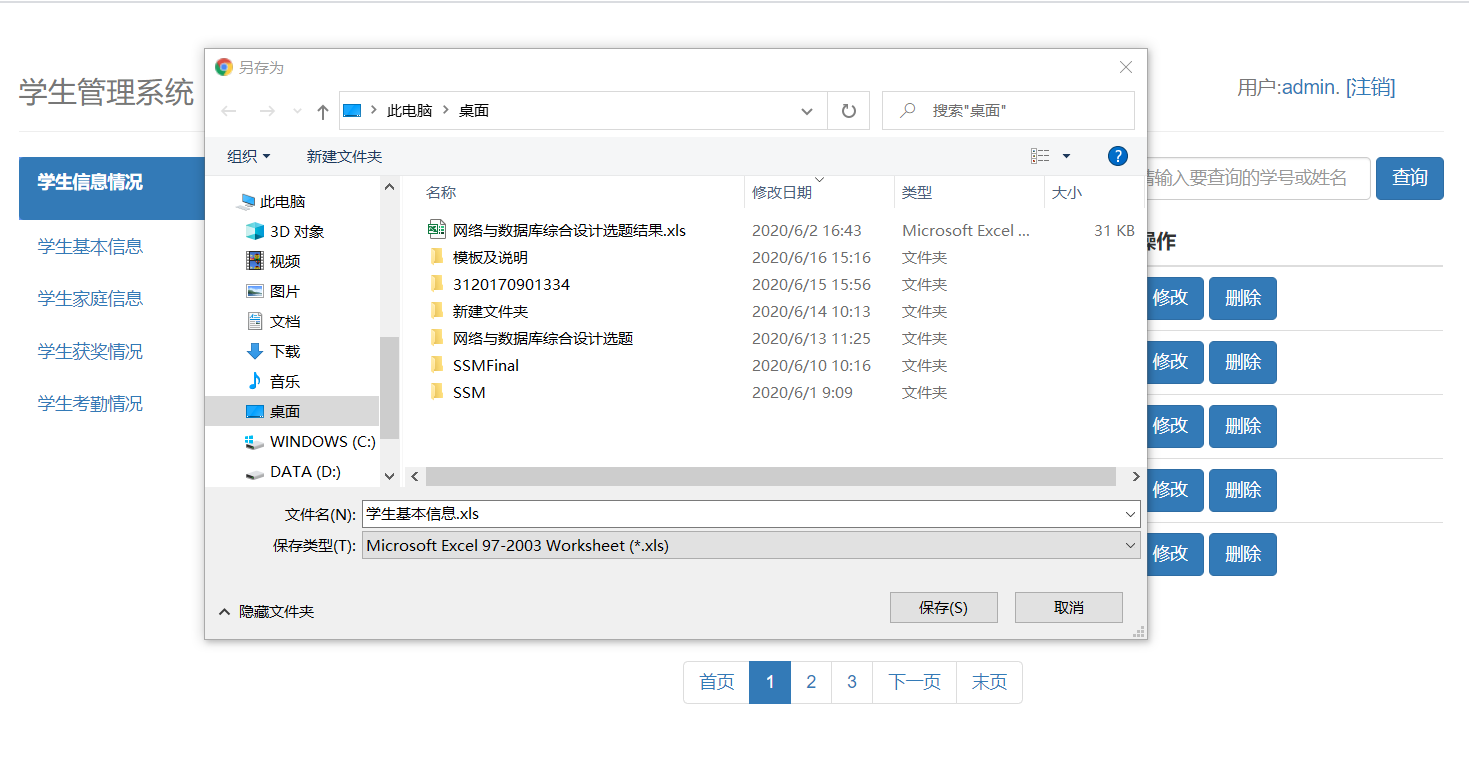


图4-35 到处信息

信息下载及内容。图4-36

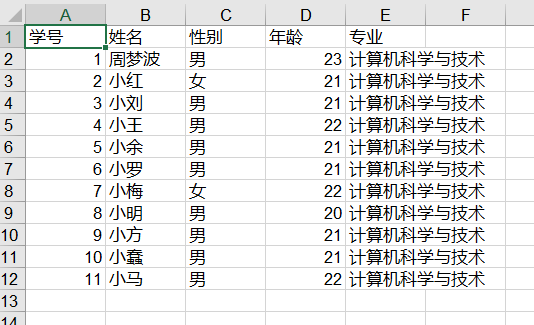
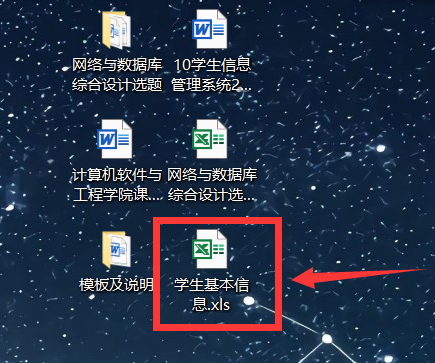


图4-36 到处信息下载及内容

# 5 系统功能测试

经过4.9软件使用说明，以及测试过管理员用户全部功能运行正常。以下是教师登录以及系统功能测试



图5-1 教师登录



图5-2 教师页面

教师拥有查看查询所有学生信息的权限，所以在主页面看不到添加信息、删除和修改按钮，运行成功。



图5-3 教师查询功能

教师查询功能运行成功。

以下是学生登录。



5-4学生登录

学生登录只可查看个人信息。



图5-5 学生个人信息

学生权限查看个人信息成功。

综上所述，本系统三个用户权限测试完成，系统各功能运行正常。

# 结 论

本次学生信息管理系统任务圆满完成，系统包括三个用户的登录，拥有不同权限，可以对学生数据进行不通的操作。

学生信息管理系统界面整体简洁美观，风格整齐划一。经过朋友同学的测试，操作方便方便，运行速度快。且在多人同时操作中，表现良好，没有出现错误。

在完成本次实训中，遇到了许多大大小小的问题，通过百度查询资料、书籍查询资料以及询问同学和老师们，我成功解决了所有的问题。在解决问题的过程中，我是收获最大的，错误让我清楚运行流程。

本次课程设计，我对SSM框架的理解更加深刻，运用更加熟练，并且了解了一个全新的css\html框架BootStrap，拓展了我的知识面。

在完成过程中，每一个功能的实现会发现拥有很多种方法，各有优点缺点，每一种全新的实现方式都使我受益匪浅。

# 致 谢

我们认为，在这学期的实验中，在收获知识的同时，还收获了阅历，收获了成熟，在此过程中，我们通过查找大量资料，请教老师，以及不懈的努力，不仅培养了独立思考、动手操作的能力，在各种其它能力上也都有了提高。更重要的是，在实验课上，我们学会了很多学习的方法。而这是日后最实用的，真的是受益匪浅。要面对社会的挑战，只有不断的学习、实践，再学习、再实践。  
不管怎样，这些都是一种锻炼，一种知识的积累，能力的提高，只有掌握了这些，才可以更进一步,取得更好的成绩。  
而且，这对于我们的将来也有很大的帮助。以后，不管有多苦，我想我们都能变苦为乐，找寻有趣的事情，发现其中珍贵的事情。

感谢大学这学期以来，老师对我学习上的帮助和生活上的关怀，老师严谨求实的治学态度，踏实坚韧的工作精神，将使我终生受益。  
感谢我的室友，在我课题研究过程中给予我技术上的极大支持和心理上的鼓励，与室友的互相帮助是一件快乐的事情。  
此次设计也让我明白了思路即出路，有什么不懂不明白的地方要及时请教或.上网查询，只要认真钻研,动脑思考，动手实践，就没有弄不懂的知识，收获颇丰。

# 参考文献

[1] 何玉洁.数据库原理与应用.机械工业出版社,2017

[2] 郭兴峰.ASP.NET动态网站开发基础教程 (c#篇). 清华大学出版社, 2014

[3] 李志云.Visual C#2012数据库开发案例教程.机械工业出版社, 2013

[4] 李国娟．基于JavaEE架构的网上购物系统的设计与实现[D]．河北工业大学，2011

[5] 霍斯特曼科内尔．Java核心技术[M]．北京：机械工业出版社，2014

[6] 张永强. 计算机软件Java编程特点及其技术分析[J]. 计算机产品与流通，2019

[7] 黄文娟. 基于Java和MySQL的图书馆信息化管理系统设计[J]. 电子设计工程，2019