



北華大學

本科毕业论文（设计）

学生成绩学分制管理系统

学 号：201517010116

学生姓名：吕钢

指导教师：

所在学院：计算机科学技术学院

所学专业：计算机科学与技术

2019 年 6 月

摘 要

本文讨论了一个基于学生成绩学分制管理系统设计的开发过程。系统基于 SpringBoot2.x 技术和 MySQL 数据库技术以及 Redis 数据库进行开发,然后运用 CSS 和 JavaScript 对 HTML5 静态页面进行美化。本次设计将系统分成前端系统和后端系统。前端页面采用 HTML5+CSS 构建前端样式, JavaScript+JQuery 完成前端页面逻辑编写, Go-Server 完成前端的服务发布; 后端采用 SpringBoot2.x 作为微服务框架的核心支持,使用 MySQL8 存储数据,Redis 作为缓存存储和队列效果的实现,Docker 容器化管理服务。微服务架构其优势为即使其中一个服务出现问题,也不会影响其它服务的正常运行,前后端分离,使得服务的界限更加明确,便于维护和操作。整个系统采用行业较为新兴的技术,功能更流畅,性能更强大,维护监控更为方便。从人员操作上,界面友好,灵活实用,满足学校管理的相关要求。

关键词: 管理; 学生成绩; 学分; Go-Server;

Abstract

This paper discusses a development process based on the design of the student credit system management system. The system is based on SpringBoot 2.x technology and MySQL database technology and Redis database development, and then use CSS and JavaScript to beautify HTML5 static pages. This design divides the system into front-end systems and back-end systems. The front-end page uses HTML5+CSS to build the front-end style, JavaScript+JQuery completes the front-end page logic, Go-Server completes the front-end service release; the back-end uses SpringBoot2.x as the core support of the micro-service framework, uses MySQL8 to store data, and Redis as the cache. Implementation of storage and queue effects, Docker containerized management services. The advantage of the micro-service architecture is that even if one of the services has problems, it will not affect the normal operation of other services, and the front-end and the end will be separated, making the boundaries of the service clearer and easier to maintain and operate. The whole system adopts the industry's relatively new technology, the function is more smooth, the performance is more powerful, and the maintenance monitoring is more convenient. From the personnel operation, the interface is friendly, flexible and practical, to meet the relevant requirements of school management.

Key words: Management; Student Achievement; Credits;

目 录

摘 要.....	I
Abstract.....	II
第 1 章 概述.....	1
1.1 项目开发背景.....	1
1.2 选择此项目的依据和目的.....	1
1.3 国内外动态.....	1
1.4 项目开发思想.....	1
第 2 章 方案论证.....	3
2.1 方案分析.....	3
2.1.1 方案一.....	3
2.1.2 方案二.....	3
2.1.3 方案三.....	4
2.2 最佳方案.....	4
2.3 系统可行性分析.....	4
2.3.1 技术可行性分析.....	4
2.3.2 经济可行性分析.....	5
2.3.3 操作可行性分析.....	5
2.3.4 法律可行性分析.....	5
第 3 章 需求分析.....	6
3.1 问题分析.....	6
3.2 需求描述.....	6
3.3 系统数据流图.....	7
3.4 数据字典.....	8
3.4.1 数据字典 user 表.....	8
3.4.2 数据字典 course 表.....	9
3.4.3 数据字典 lesson 表.....	9
3.4.4 数据字典 teach 表.....	10
3.5 系统 E-R 图.....	11
3.6 数据库概念结构设计.....	13
第 4 章 过程论证.....	14
4.1 概要设计.....	14
4.2 数据库的连接.....	15

4.2.1 创建并配置连接数据库.....	15
4.2.2 MySQL 的连接.....	16
4.3 数据库的详细设计和实现.....	16
4.4 系统详细设计的主要思路.....	23
4.5 系统模块的设计.....	23
4.6 前台功能模块的实现.....	24
4.6.1 用户登录时的设计.....	24
4.6.2 用户信息模块设计.....	27
4.6.3 课程学分管理设计.....	30
4.7 后端功能模块的实现.....	40
4.7.1 用户登陆模块设计.....	40
4.7.2 用户信息模块设计.....	42
4.7.3 课程信息模块设计.....	42
4.7.4 课程管理模块设计.....	43
第 5 章 结果分析.....	45
5.1 软件测试.....	45
5.1.1 黑盒测试前端页面.....	45
5.1.2 白盒测试后端代码.....	47
5.2 结果分析.....	50
结 论.....	51
致 谢.....	52
参考文献.....	53
附 录.....	54

第 1 章 概述

1.1 项目开发背景

21 世纪以来，中国现代大学教育以世界领先的的发展速度，实现着科学技术教育和人才培养的光荣使命，育人为先的教师和刻苦求学的学子数量持续上升，学生的成绩与学分挂钩，检验学生在校内的学习水平是否达到学校的目标^[1]。

1.2 选择此项目的依据和目的

该项管理非常重要并且记录、计算、操作繁琐，在 IT 行业高速发展的今天，新兴技术的出现，使用计算机完成这一切的管理变得现实、简单高效。Web 开发的技术高速发展，使得不需要使用开发软件浪费大量的人力和物力，以及大量的时间。

项目的目的旨在使得学院管理、老师、学生可以方便快捷的使用，完成所需要的课程、成绩、学分的管理^[2]。

1.3 国内外动态

国内在 web 项目开发这一方面，约 40% 的项目在使用 SSM 框架开发，20% 的项目在使用 ssh 框架开发，超过 30% 在使用基于 springboot 的微服务思想及 spring jpa 的存储辅助^[3]。

国外在 web 开发更多的习惯于使用 springboot 的微服务思想及 hibernate 代替国内使用更多的 mybatis 来用于存储辅助^[4]。

1.4 项目开发思想

项目的管理角色分为学院管理员、教师、学生三个角色参与。学院管理员提供课程列表，学分要求，教师和学生与课程相互绑定，为多对多的关系，教师最后给学生提供最终的成绩^[5]。

本次设计将系统分成前端系统和后端系统。前端页面采用 HTML5+CSS 构建前端样式，JavaScript+jQuery 完成前端页面逻辑编写，Go-Server 完成前端的服务发布；后端采用 SpringBoot2.x 作为微服务框架的核心支持^[6]，使用 MySQL8 存储数据，Redis 作为缓存存储和队列效果的实现，Docker 容器化管理服务。

表 1-1 技术使用比较

语言技术名称	Java	Python	PHP
微服务架构	提供了 SpringBoot 微服务架构便于操作实现	运行速度较快，提供微服务支持，但对大量数据处理支持不足	提供微服务支持，但实现较为困难，对于高并发处理支持不足
数据库	提供了 hibernate、mybatis 及 spring jpa	提供了 MySQLdb 的 api 接口及安装包	提供了 ODBC 数据库
国内外技术名称	国内		国外
Web 开发	ssh、SSM、SpringBoot+mybatis 基于需求使用		ssh、SSM、SpringBoot+spring jpa(hibernate)基于需求使用
使用平台	Windows 系统		Ubuntu 系统
性能	性能较为一般，运行服务较多		系统安全稳定
操作	操作方式较为友好，图形化操作界面操作方便，支持大量驱动		指令运行，权限管理非常严格，开源免费正版

第 2 章 方案论证

2.1 方案分析

2.1.1 方案一

系统软件开发。使用软件开发可以满足项目开发的业务需求，技术使用为 Java 语言、RMI 通信机制、JDBC 数据库访问通用接口、MySQL 数据库、Java Servlet web 服务网络的网络组件、GUI 图形界面等，但拥有诸多缺陷：

- （1）软件开发周期时间长，前端与后端代码交织在一起开发，无法同步开发，只能同时开发后端 Java 服务端代码和前端 GUI 图形代码；
- （2）界面简陋，只能使用 Java 自带的图形图画，界面死板；
- （3）可移植性差，只能以软件安装包或软件压缩包的方式复制粘贴工具，而且需要在每台环境都配置支持软件运行的环境，如 JDK 何 MySQL 服务；
- （4）监控不方便，系统出现问题时无法快速准确的定位到出错的所在，进行修复。

2.1.2 方案二

Web 技术 SSM 框架开发。可以满足项目开发的业务需求，技术使用为 Java 语言、JSP 页面、MyBatis 数据库管理应用、MySQL 数据库、Java Servlet web 服务网络的网络组件，解决了方案 2.1 中如下问题：

- （1）页面为 JSP 页面，支持 HTML、CSS 和 JavaScript 组合的编写方式，在页面的美观上大大提升，增加用户的体验感；
- （2）可移植性较强，多为页面访问，移植体积大大降低，页面操作使得系统的使用更为方便，提高用户的使用效率。

可同时也存在以下缺陷：

- （1）软件开发周期时间长，前端 JSP 页面和 java 后端代码绑定在一起无法同步开发无法同步开发，只能同时开发后端 Java 服务端代码和前端 JSP 页面的 HTML、CSS 代码；
- （2）监控不方便，JSP 页面需要与 Java 后端服务绑定在一起，共用一个 tomcat 服务器加载，出现问题时无法做出区分，当一部分服务出现异常时，其它服务将无法正常使用。

2.1.3 方案三

Web 技术 SpringBoot 框架开发。可以满足项目开发的业务需求,技术使用为 Java 语言、HTML+CSS+JavaScript 页面、Mybatis 数据库应用、MySQL 数据库、SpringBoot 微服务架构、Redis 数据缓存, 解决方案 2.2 中的所有问题:

(1) 软件开发周期时间短, 前端页面开发与后端服务开发完全区分开, 双方可以同时开发, 大大加快了开发进度, 缩短周期时间;

(2) 前端页面支持 HTML+CSS+JavaScript 编写, 美观程度大大提升;

(3) 可移植性强, 只需要在一台服务器上配置好服务端代码, 其它访问的客户端电脑都可以直接通过页面访问, 进行访问操作;

(4) 监控简单容易, 由于前端采用简单的 Go-Server 服务启动 HTML 页面, 对于前端的错误我们只需监控页面错误和 Go-Server 的服务是否有误, 对于后端代码我们可以采用 monitor 监控后端的日志打印来进行监控

2.2 最佳方案

语言使用 java, 技术成熟完善, 社区人员众多, 解决方案较多, 技术开源免费, 而且开发成本低, 熟练度较高。

数据库使用 MySQL 数据库, 开源免费, 而且性能较高, 占用硬盘和内存较小, 而且 JDBC 支持友好, 便于 Java 客户端的操作。

架构技术采用方案 2.3 中的 web 技术 SpringBoot 开发, 优点如下:

(1) 开发周期时间短, 前后端可同时开发;

(2) 开发成本低, 只需要一台服务器;

(3) 页面操作友好, 界面美观程度高;

(4) 系统在不同环境上可移植部署, 移植性强;

(5) 对微服务服务后台控制监控简单;

(6) 系统可拆分, 进行维护时方便。

2.3 系统可行性分析

2.3.1 技术可行性分析

SpringBoot 当前已经发展并稳定的应用到企业级开发中, 有着成熟的应用经验, Spring Data JPA 可以支持来自不同的数据存储技术, 满足我们对 MySQL 数据库和 Redis 数据的操作, 我们可以使用 pom.xml 添加我们需要的依赖。

2.3.2 经济可行性分析

SpringBoot 为开源技术,可以免费使用,MySQL 同样对于我们而言也是免费的,在技术使用上我们无需支付任何费用,在服务器购买上,当服务性能要求不高时,一台服务器即可满足需要,经济话费大大降低

2.3.3 操作可行性分析

系统的技术操作多为页面点击和信息录入,而且见面提供了大量的提示信息,操作简单方便,界面美观,用户操作舒适度高。

2.3.4 法律可行性分析

根据中华人民共和国法律规定,此系统未不经同意使用非开源技术,系统的设计和数据内容不涉及私密安全内容,合乎法律的要求。

第3章 需求分析

3.1 问题分析

21 世纪以来，中国现代大学教育以世界领先的的发展速度，实现着科学技术教育和人才培养的光荣使命，育人为先的教师和刻苦求学的学子数量持续上升，学生的成绩与学分挂钩，检验学生在校内的学习水平是否达到学校的目标。

该项管理非常重要并且记录、计算、操作繁琐，在 IT 行业高速发展的今天，使用计算机完成这一切的管理变得现实、简单高效。Web 开发的技术高速发展，使得不需要使用开发软件浪费大量的时间。

我们需要开发一套学生学分成绩管理系统，使得学校、教师和学生可以方便、高效和准确的完成相应的操作，学校可以根据教育计划合理安排课程，教师根据学生平时分、测试分和考试分给予最终的成绩和学分，学生可以抢取课程和查询自己课程的相应成绩。

3.2 需求描述

学生学分成绩管理系统的目的是旨在学院管理员、教师和学生三种用户角色之间的交互，主要的关系对象为课程、成绩、学分三部分[7]。

(1) 需要构建以用户表为核心，角色表和权限表来划分用户的操作功能，用来区分在系统当中可获得的权限。

(2) 登陆系统，不同角色登陆进入不同访问页面。

(3) 高权限的角色允许对低权限用户进行增删改查操作，如学院管理员可以添加教师、学生到系统的用户中，教师可以查看学生列表。

(4) 学院管理员可以添加、修改、删除、查询课程信息，并对的学分设置和分配比例，以及课程的属性为选修或必修。

(5) 教师在自己课程中可以对学进行移除，查询学生详细信息，对学生在平时分、测试（实验）分和考试分进行填写。

(6) 学生可以查询自己的课程及相关的成绩，对于选修课，学生可以自己自由选择，抢占名额，同时也会有成绩。

(7) 对于课程，我们分为选修课程和必修课程，必修课由学院管理员直接指定，选修课由学生选定课程和老师。

(8) 必修课程有学院管理制定，指定教师和学生，为其制定平时分、实验分和考试分占的比重，以及这门课程所代表的学分。

(9) 教师根据学生平时表现和上课情况给予学生平时分，根据课堂测试或课程设计给予测试分，根据期末测试给予考试分。

(10) 学生可以查询选修课列表，进行抢占选修课，同时可以查询自己的考试成绩和最终获得的学分。

3.3 系统数据流图

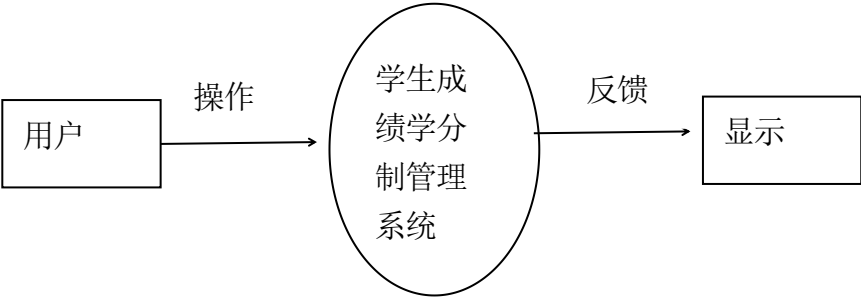


图 3.1 顶级流图

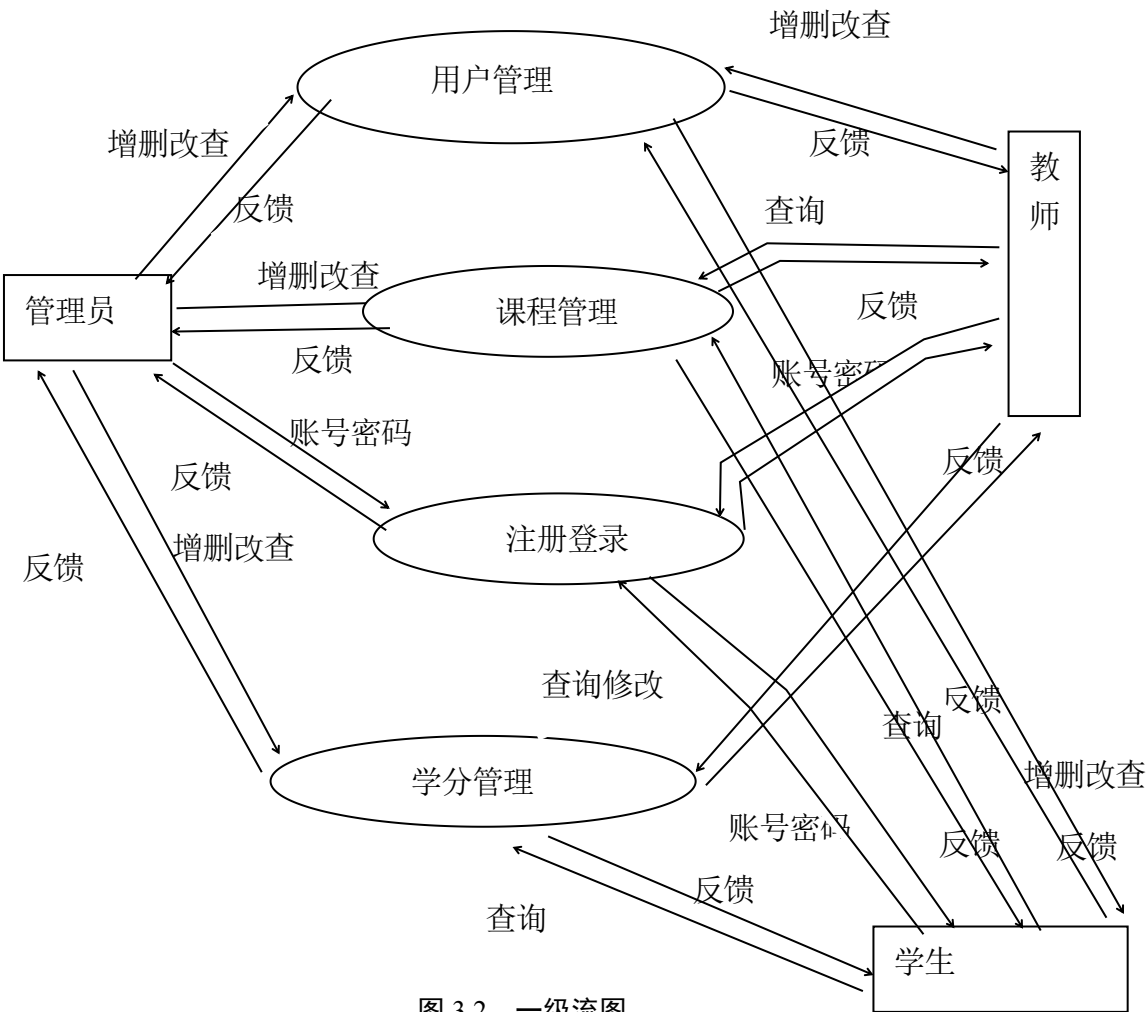


图 3.2 一级流图

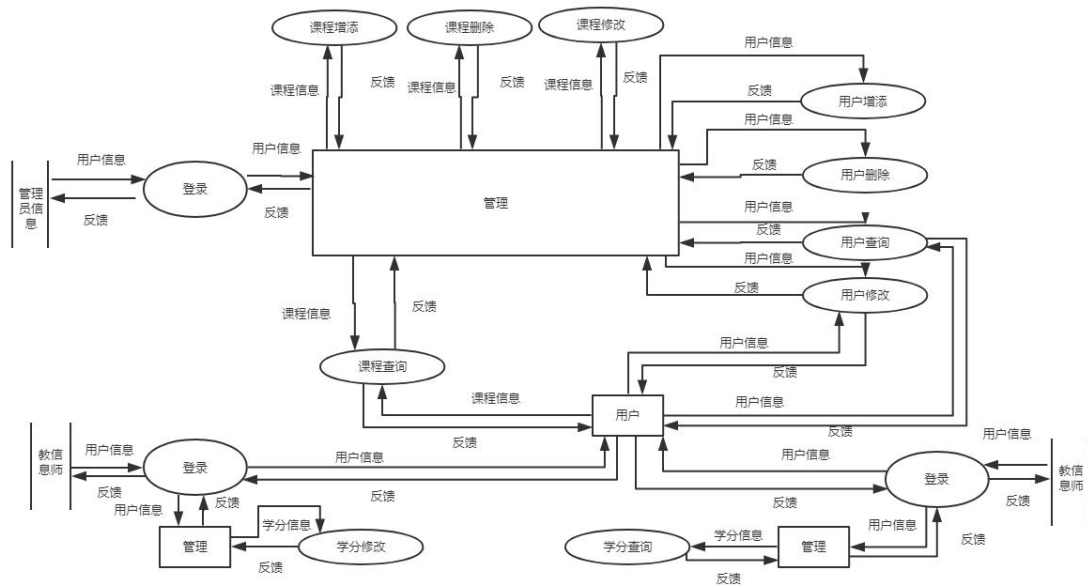


图 3.3 二级流程图

3.4 数据字典

3.4.1 数据字典 user 表

数据文件：用户列表

文件组成：用户序号 user_id, 唯一识别 code 码, 账号名 username, 性别 gender, 邮箱 email, 联系电话 phone, 登陆密码 password, 用户类型 type, 头像链接 avatar, 用户描述 description, 是否需要验证码 captcha_status, 是否重置密码 reset_status, 组织机构 id org_id, 数据状态位 status, 创建时间 create_time, 更新时间 update_time, 创建者 create_id

数据项：用户序号 user_id; 数据类型：bigint; 数据长度：20;

数据项：唯一识别 code 码; 数据类型：varchar; 数据长度：64;

数据项：账号名 username; 数据类型：varchar; 数据长度：64;

数据项：邮箱 email; 数据类型：varchar; 数据长度：64;

数据项：联系电话 phone; 数据类型：varchar; 数据长度：64;

数据项：性别 gender; 数据类型：varchar; 数据长度：16;

数据项：登陆密码 password; 数据类型：varchar; 数据长度：64;

数据项：用户类型 type; 数据类型：varchar; 数据长度：64;

数据项：头像链接 avatar; 数据类型：varchar; 数据长度：1024;

数据项：用户描述 description; 数据类型：varchar; 数据长度：1024;

数据项：是否需要验证码 captcha_status; 数据类型：tinyint; 数据长度：4;

数据项：是否重置密码 reset_status; 数据类型：tinyint; 数据长度：4;

数据项: 组织机构 id org_id; 数据类型: bigint; 数据长度: 20;
数据项: 数据状态位 status; 数据类型: varchar; 数据长度: 16;
数据项: 创建时间 create_time; 数据类型: timestamp; 数据长度: 0;
数据项: 更新时间 update_time; 数据类型: timestamp; 数据长度: 0;
数据项: 创建者 create_id; 数据类型: bigint; 数据长度: 20;

3.4.2 数据字典 course 表

数据文件: 课程列表

文件组成: 课程序号 course_id, 唯一识别 code 码, 课程名 name, 课程类型 type, 课程描述 description, 大学 id university_id, 大学名 university_name, 学院 id faculty_id, 学院名 faculty_name, 系 id subject_id, 系名 subject_name, 表现分百分比 behavior, 实验分占百分比 test, 考试分占百分比 exam, 满学分 credit, 授课时长 teach_time, 数据状态位 status, 创建时间 create_time, 更新时间 update_time, 创建者 create_id

数据项: 课程序号 user_id; 数据类型: bigint; 数据长度: 20;
数据项: 唯一识别 code 码; 数据类型: varchar; 数据长度: 64;
数据项: 课程名 name; 数据类型: varchar; 数据长度: 64;
数据项: 课程类型 type; 数据类型: varchar; 数据长度: 64;
数据项: 课程描述 description; 数据类型: varchar; 数据长度: 64;
数据项: 大学 id university_id; 数据类型: varchar; 数据长度: 16;
数据项: 大学名 university_name; 数据类型: varchar; 数据长度: 64;
数据项: 学院 id faculty_id; 数据类型: varchar; 数据长度: 64;
数据项: 学院名 faculty_name; 数据类型: varchar; 数据长度: 64;
数据项: 系 id subject_id; 数据类型: varchar; 数据长度: 1024;
数据项: 系名 subject_name; 数据类型: varchar; 数据长度: 1024;
数据项: 表现分百分比 behavior; 数据类型: tinyint; 数据长度: 4;
数据项: 实验分占百分比 test; 数据类型: tinyint; 数据长度: 4;
数据项: 考试分占百分比 exam; 数据类型: bigint; 数据长度: 20;
数据项: 满学分 credit; 数据类型: bigint; 数据长度: 20;
数据项: 授课时长 teach_time; 数据类型: bigint; 数据长度: 20;
数据项: 数据状态位 status; 数据类型: varchar; 数据长度: 16;
数据项: 创建时间 create_time; 数据类型: timestamp; 数据长度: 0;
数据项: 更新时间 update_time; 数据类型: timestamp; 数据长度: 0;
数据项: 创建者 create_id; 数据类型: bigint; 数据长度: 20;

3.4.3 数据字典 lesson 表

数据文件: 教学列表

文件组成: 课程序号 `lesson_id`, 唯一识别 `code` 码, 课程名 `name`, 课程类型 `type`, 课程描述 `description`, 大学 `id university_id`, 大学名 `university_name`, 学院 `id faculty_id`, 学院名 `faculty_name`, 系 `id subject_id`, 系名 `subject_name`, 数据状态位 `status`, 创建时间 `create_time`, 更新时间 `update_time`, 创建者 `create_id`

数据项: 课程序号 `lesson_id`; 数据类型: `bigint`; 数据长度: 20;

数据项: 唯一识别 `code` 码; 数据类型: `varchar`; 数据长度: 64;

数据项: 课程名 `name`; 数据类型: `varchar`; 数据长度: 64;

数据项: 课程类型 `type`; 数据类型: `varchar`; 数据长度: 64;

数据项: 课程描述 `description`; 数据类型: `varchar`; 数据长度: 64;

数据项: 大学 `id university_id`; 数据类型: `varchar`; 数据长度: 16;

数据项: 大学名 `university_name`; 数据类型: `varchar`; 数据长度: 64;

数据项: 学院 `id faculty_id`; 数据类型: `varchar`; 数据长度: 64;

数据项: 学院名 `faculty_name`; 数据类型: `varchar`; 数据长度: 64;

数据项: 系 `id subject_id`; 数据类型: `varchar`; 数据长度: 1024;

数据项: 系名 `subject_name`; 数据类型: `varchar`; 数据长度: 1024;

数据项: 数据状态位 `status`; 数据类型: `varchar`; 数据长度: 16;

数据项: 创建时间 `create_time`; 数据类型: `timestamp`; 数据长度: 0;

数据项: 更新时间 `update_time`; 数据类型: `timestamp`; 数据长度: 0;

数据项: 创建者 `create_id`; 数据类型: `bigint`; 数据长度: 20;

3.4.4 数据字典 `teach` 表

数据文件: 授课列表

文件组成: 课程序号 `teach_id`, 唯一识别 `code` 码, 教学 `id lesson_id`, 授课名 `name`, 课程描述 `description`, 系 `id subject_id`, 系名 `subject_name`, 表现分 百 `behavior`, 实验分 `test`, 考试分 `exam`, 学分 `credit`, 最终成绩 `score`, 数据状态位 `status`, 创建时间 `create_time`, 更新时间 `update_time`, 创建者 `create_id`

数据项: 课程序号 `teach_id`; 数据类型: `bigint`; 数据长度: 20;

数据项: 唯一识别 `code` 码; 数据类型: `varchar`; 数据长度: 64;

数据项: 课程名 `name`; 数据类型: `varchar`; 数据长度: 64;

数据项: 课程描述 `description`; 数据类型: `varchar`; 数据长度: 64;

数据项: 表现分百分比 `behavior`; 数据类型: `tinyint`; 数据长度: 4;

数据项: 实验分占百分比 `test`; 数据类型: `tinyint`; 数据长度: 4;

数据项: 考试分占百分比 `exam`; 数据类型: `bigint`; 数据长度: 20;

数据项: 满学分 `credit`; 数据类型: `bigint`; 数据长度: 20;

数据项: 最终成绩 `score`; 数据类型: `bigint`; 数据长度: 20;

数据项: 系 `id subject_id`; 数据类型: `varchar`; 数据长度: 1024;

数据项: 系名 `subject_name`; 数据类型: `varchar`; 数据长度: 1024;

数据项: 数据状态位 `status`; 数据类型: `varchar`; 数据长度: 16;

数据项：创建时间 create_time；数据类型：timestamp；数据长度：0；；

3.5 系统 E-R 图

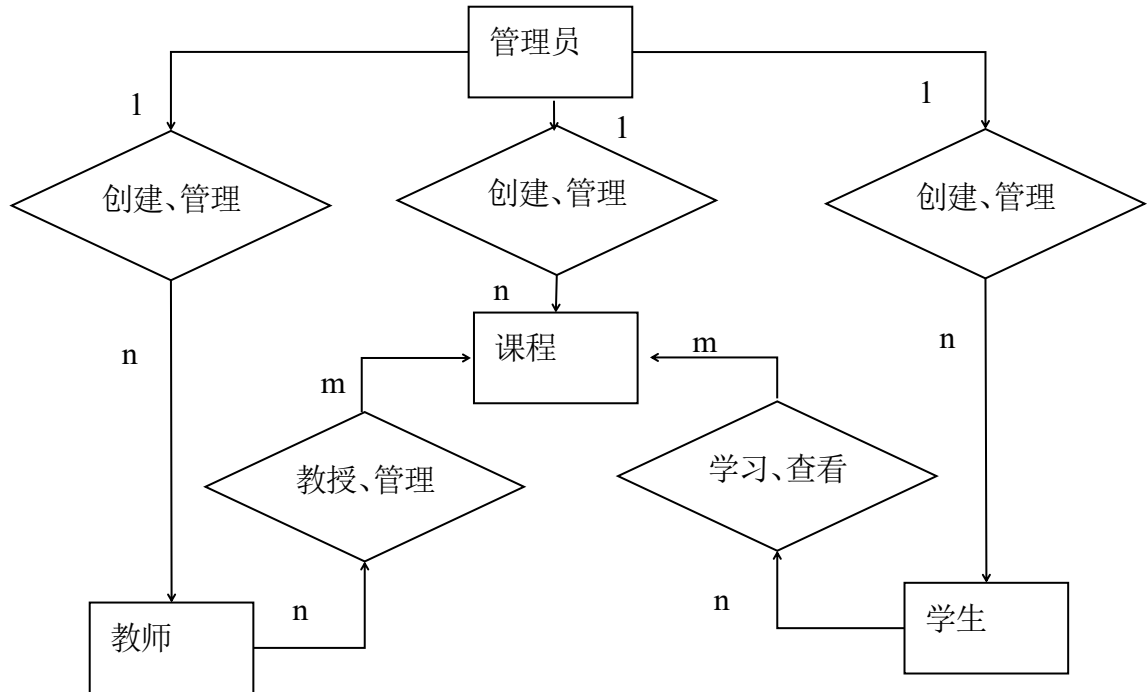


图 3-4 E-R 图

管理员详细结构

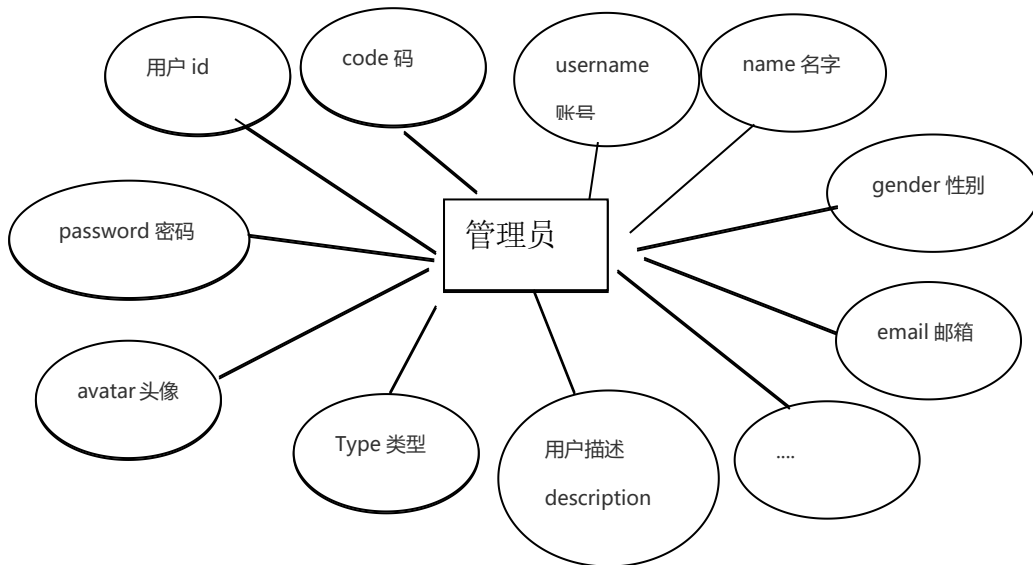


图 3-5 管理员实体属性图

教师员详细结构

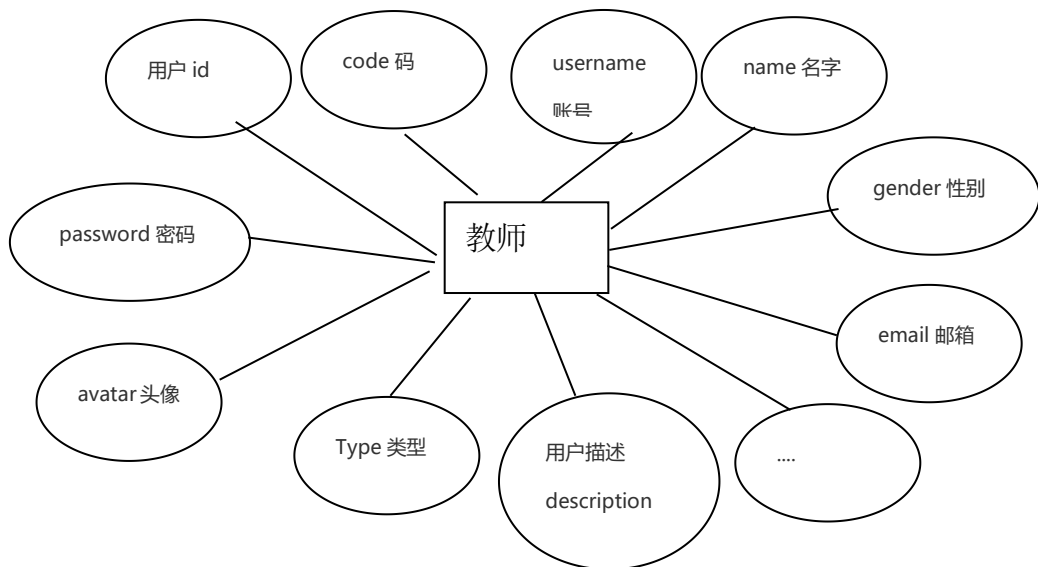


图 3-6 教师结构图

学生详细结构

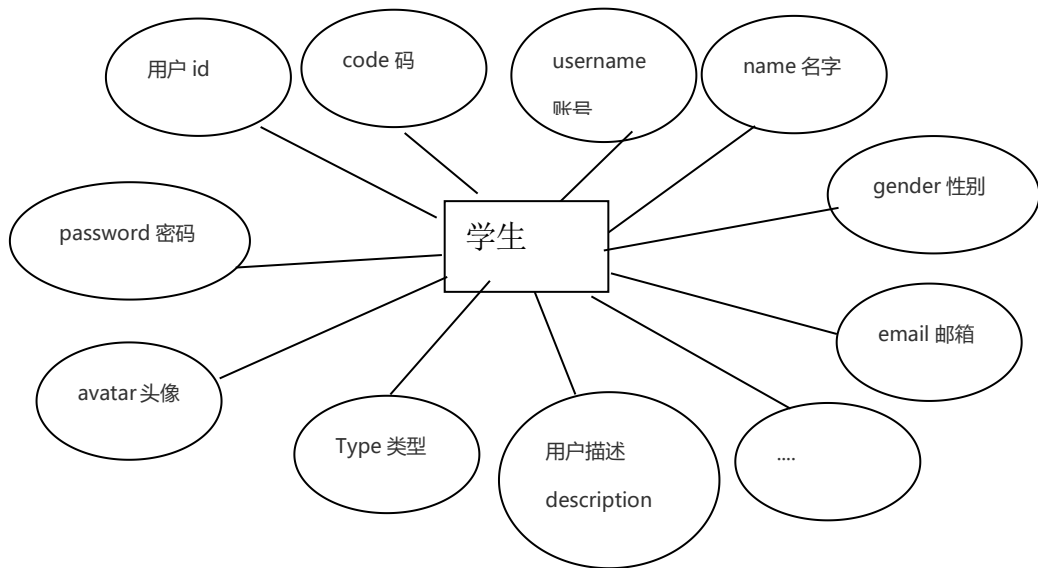


图 3-7 学生结构图

课程详细结构

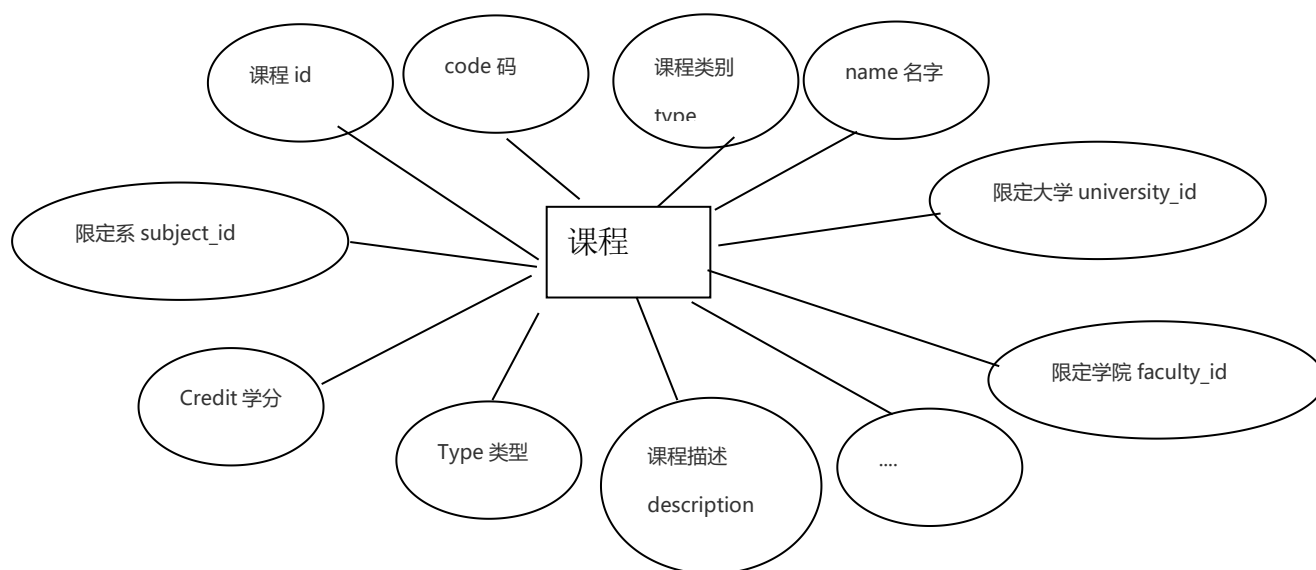


图 3-8 课程结构图

3.6 数据库概念结构设计

根据微服务架构，我们拆分三个以上的微服务，根据微服务查分数据库，使得数据库的数据得以分散，减轻数据库的压力。

系统的主要内容三个不同角色关于学分成绩和课程的关系和对应的操作，我们将数据库进行拆分，分别为用户库（account）和课程库（credit）。

用户库为 account 库，用于存储用户信息、角色信息和权限信息。在 account 库中我们分为几个独立的表结构：user 用户表，role 角色表和 fun 权限表，用于存储不同用户进入的不同操作，当然我们构建了它们三个的关联表用户和角色的 user-role 关联表和角色和权限的管理表 role-fun 关联表。

课程库为 credit 库，用于存储课程信息、教学信息和学生成绩信息。三个表分别为 course 表，存储课程的基本信息，如课程的名称，类别，分数占比和最终的学分，lesson 表用于分配教师，指定教师可以教授这门课程并可以给予学分，teach 表用于存储这么课教授的学生信息以及学生获得学分^[8]。

第 4 章 过程论证

4.1 概要设计

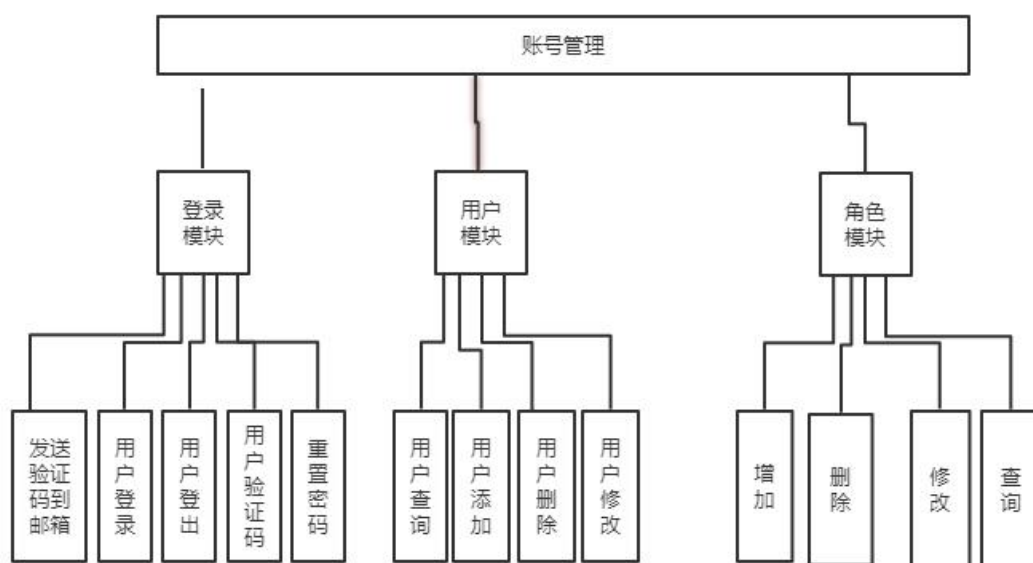


图 4-1 账号管理模块设计图

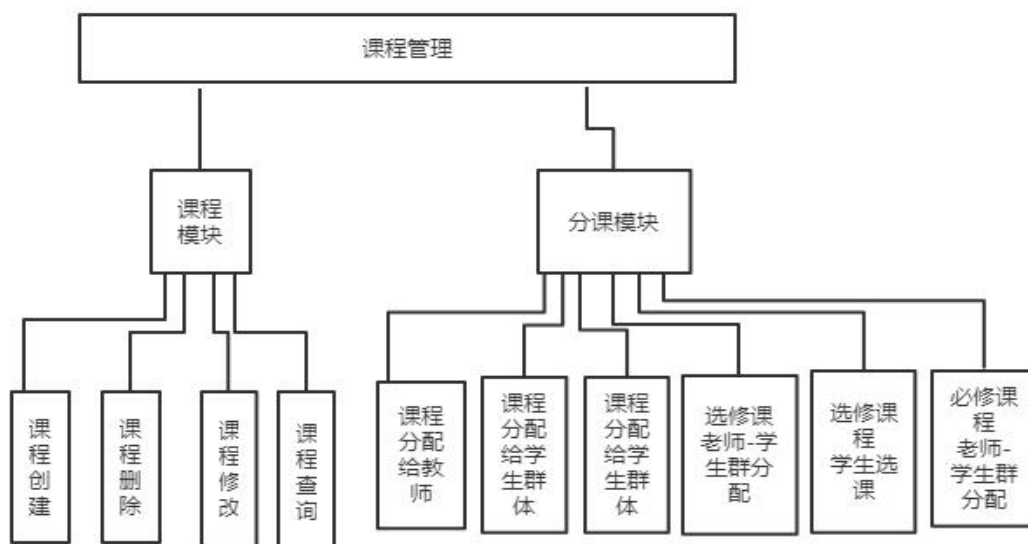


图 4-2 课程管理模块设计图

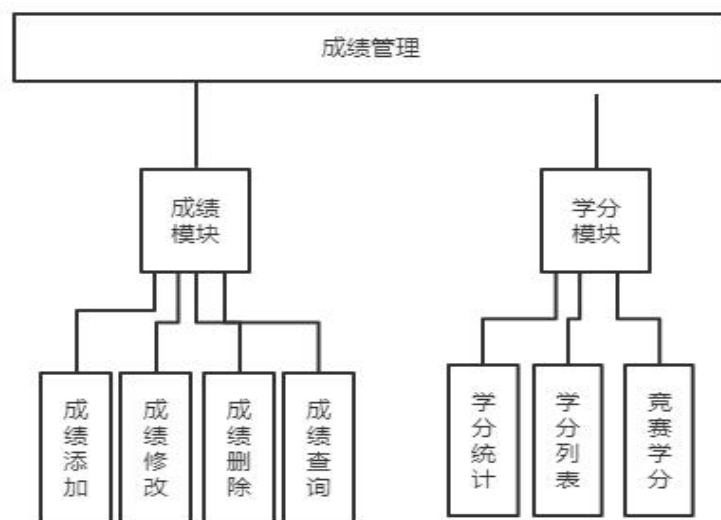


图 4-3 成绩管理模块设计图

4.2 数据库的连接

4.2.1 创建并配置连接数据库

JDBC (Java Data Base Connectivity) 是用于 Java 的数据库连接。它是 Java 语言中的应用程序接口，用于管理客户端程序如何访问数据库。它提供了查询 (Query) 和更新 (Insert、Update、Delete) 数据库中数据的方法。JDBC 只能用于连接关系型数据库 (如 MySQL、Oracle、SQLServer 等)，无法连接非关系型数据库 (Redis、MongoDB、ElasticSearch 等)。

我们通过 JDBC 提供的 JDBC API 接口向我们选定的关系型数据库发送 SQL 语句，无须我们为访问数据库编辑原生的连接程序。Java 数据库连接体系结构是 Java 应用程序连接到数据库的标准方法。JDBC 是 Java 程序员的 API，也是服务提供者连接数据库的接口模型。Java 数据库连接体系结构是 Java 应用程序连接到数据库的标准方法，是连接数据库的服务提供者的接口模型。

客户端通过 Java 代码执行程序，访问 JDBC API 发送请求，在数据库服务端会执行 SQL 语句，并将执行结果返回到响应结果中。

JDBC 的连接过程为：

- (1) 与数据库建立连接；
- (2) 发送 SQL 语句，返回执行结果；
- (3) 关闭数据库的连接，避免资源浪费。

4.2.2 MySQL 的连接

我们的存储数据库使用 MySQL, MySQL 数据库在开发中较于其它关系型数据库有着大量的优势:

(1) 成本较低, 价格对于大部分使用用户来说基本是免费使用的, 相比于 Oracle 和 SQLserver 的付费使用, 成本非常的低;

(2) 易于使用, 复杂程度低, 高性能, 相对简单, 服务稳定, 持久化层优秀, 避免数据丢失;

(3) 服务占用硬盘大小和内存大小较小, mysql-8.0.13-winx64 的大小仅仅为 192M;

(4) 支持 SQL (Structured Query Language) 语言对接数据库, 对数据进行查询和修改操作;

(5) 连接安全, MySQL 可通过 url 访问连接, 可由 ip 地址+端口组合访问, 可以控制在局域网内特定人员访问, 也可以共享到公网上, 远程访问数据库;

(6) 可移植性, MySQL 拥有 Windows 版本和 Unix 版, 支持在不同类型服务器上运行, 而且数据库的结构和数据内容由 SQL 决定, 需要移植数据库时, 我们只需要导出 SQL 脚本语句, 在目标环境导入即可;

(7) 性能高效, MySQL 服务器理论上没有用户数量和访问次数限制, 允许集群布置, 允许多个客户端访问同一个数据库、同一个表乃至同一个数据项, 在访问请求速度和 SQL 脚本处理速度上, 和其它关系型数据库(费付费星)相比是最快的;

(8) 开源项目代码, 使得安全性大大提升, 我们可以学习或根据我们的需求修改源代码。

在 SpringBoot 中, 我们可以使用 spring 自带的 MySQL 依赖, 在 application.yml 中配置连接参数。

4.3 数据库的详细设计和实现

功能: 用户登录成功时的基本信息

表 4-1 user 表

Field	Type	Null	Key	Default	Comment
user_id	bigint(20)	NO	PRI	(NULL)	ID
code	varchar(64)	NO	UNI	(NULL)	编码
username	varchar(64)	NO	UNI	(NULL)	用户名
name	varchar(64)	NO		(NULL)	名字
gender	varchar(16)	NO		(NULL)	性别
email	varchar(64)	NO	UNI	(NULL)	邮箱
phone	varchar(16)	NO	UNI	(NULL)	手机号
password	varchar(64)	NO		(NULL)	密码
type	varchar(16)	YES		(NULL)	类型
avatar	varchar(1024)	YES		(NULL)	头像
description	varchar(1024)	YES		(NULL)	描述
captcha_status	tinyint(4)	NO		0	是否需要验证码
reset_status	tinyint(4)	NO		1	是否已重置
org_id	bigint(20)	NO	MUL	(NULL)	组织机构 ID
creator_id	bigint(20)	NO		(NULL)	创建者
status	varchar(16)	NO		enabled	状态
create_time	timestamp	NO		CURRENT_TIMESTAMP	创建时间
update_time	timestamp	NO		CURRENT_TIMESTAMP	更新时间

功能：用户的角色表

表 4-2 role 表

Field	Type	Null	Key	Default	Comment
role_id	bigint(20)	NO	PRI	(NULL)	ID
code	varchar(64)	NO	UNI	(NULL)	编号
name	varchar(64)	NO		(NULL)	名称
description	varchar(1024)	YES		(NULL)	描述
type	varchar(16)	NO		(NULL)	类型
creator_id	bigint(20)	NO	MUL	(NULL)	创建者
status	varchar(16)	NO		enabled	状态
create_time	timestamp	NO		CURRENT_TIMESTAMP	创建时间
update_time	timestamp	NO		CURRENT_TIMESTAMP	更新时间

功能：用户的权限表

表 4-3 fun 表

Field	Type	Null	Key	Default	Comment
func_id	bigint(20)	NO	PRI	(NULL)	ID
code	varchar(64)	NO	UNI	(NULL)	编号
name	varchar(64)	NO	UNI	(NULL)	名称
description	varchar(1024)	YES		(NULL)	描述
url	varchar(512)	NO		(NULL)	路径
icon	varchar(16)	YES		(NULL)	图标
type	varchar(16)	NO		(NULL)	类型
number	bigint(20)	NO		0	序号
parent_id	bigint(20)	NO		0	父 ID
root_id	bigint(20)	NO		0	根节点
status	varchar(16)	NO		enabled	状态
create_time	timestamp	NO		CURRENT_TIMESTAMP	创建时间
update_time	timestamp	NO		CURRENT_TIMESTAMP	更新时间

功能：用户的角色配置表

表 4-4 user-role 表

Field	Type	Null	Key	Default	Comment
user_role_id	bigint(20)	NO	PRI	(NULL)	ID
user_id	bigint(20)	NO	MUL	(NULL)	用户 ID
role_id	bigint(20)	NO	MUL	(NULL)	角色 ID
status	varchar(16)	NO		enabled	状态
create_time	timestamp	YES		CURRENT_TIMESTAMP	创建时间
update_time	timestamp	YES		CURRENT_TIMESTAMP	更新时间

功能：用户的角色对于权限配置表

表 4-5 role-fun 表

Field	Type	Null	Key	Default	Comment
func_role_id	bigint(20)	NO	PRI	(NULL)	ID
func_id	bigint(20)	NO	MUL	(NULL)	功能 ID
role_id	bigint(20)	NO	MUL	(NULL)	角色 ID
status	varchar(16)	NO		enabled	状态
create_time	timestamp	NO		CURRENT_TIMESTAMP	创建时间
update_time	timestamp	NO		CURRENT_TIMESTAMP	更新时间

功能：组织机构表

表 4-6 org 表

Field	Type	Null	Key	Default	Comment
org_id	bigint(20)	NO	PRI	(NULL)	ID
code	varchar(64)	NO	UNI	(NULL)	编号
name	varchar(64)	NO		(NULL)	名称
type	varchar(64)	NO		(NULL)	类型
number	bigint(20)	NO		0	序号
description	varchar(1024)	YES		(NULL)	描述
parent_id	bigint(20)	NO		0	父 ID
root_id	bigint(20)	NO		0	
creator_id	bigint(20)	NO	MUL	(NULL)	创建者
status	varchar(16)	NO		enabled	状态

功能：课程的概要信息表

表 4-7 course 表

Field	Type	Null	Key	Default	Comment
course_id	bigint(20)	NO	PRI	(NULL)	
code	varchar(64)	NO		(NULL)	
name	varchar(64)	NO		(NULL)	
description	varchar(1024)	YES		(NULL)	任务描述
type	varchar(16)	YES		(NULL)	课程类型(required 必修,elective 选修)
university_id	bigint(20)	YES		(NULL)	限定大学 id
university_name	varchar(64)	YES		(NULL)	
faculty_id	bigint(20)	YES		(NULL)	限定的学院 id
faculty_name	varchar(64)	YES		(NULL)	
subject_id	bigint(20)	YES		(NULL)	限定系 id
subject_name	varchar(64)	YES		(NULL)	
behavior	int(11)	YES		(NULL)	表现分百分比, 结果/100
test	int(11)	YES		(NULL)	实验分占百分比, 结果/100
exam	int(11)	YES		(NULL)	考试分占百分比, 结果/100
credit	int(11)	NO		0	满学分
teach_time	int(64)	NO		0	授课时长
maxnum	int(11)	NO		(NULL)	上限人数
status	varchar(16)	NO		enable	是否禁用
create_id	bigint(20)	YES		(NULL)	创建者 id
create_time	timestamp	YES		(NULL)	
update_time	timestamp	YES		(NULL)	

功能：课程的教学信息表

表 4-8 lesson 表

Field	Type	Null	Key	Default	Comment
lesson_id	bigint(20)	NO	PRI	(NULL)	
code	varchar(64)	NO		(NULL)	
course_id	bigint(20)	YES		(NULL)	课程 id
name	varchar(64)	NO		(NULL)	
description	varchar(1024)	YES		(NULL)	任务描述
num	int(11)	YES		(NULL)	已选人数
type	varchar(16)	YES		nofull	课程类型(full 已满,surplus 有余,stop 终止,)
teacher_id	bigint(20)	NO		(NULL)	授课老师 id
teacher_name	varchar(64)	NO		(NULL)	授课老师姓名
status	varchar(16)	NO		enable	是否可用(enable,disable)
create_id	bigint(20)	YES		(NULL)	创建者 id
create_time	timestamp	YES		(NULL)	
update_time	timestamp	YES		(NULL)	

功能：学生的课程成绩表

表 4-9 teach 表

Field	Type	Null	Key	Default	Comment
teach_id	bigint(20)	NO	PRI	(NULL)	
code	varchar(64)	NO		(NULL)	
lesson_id	bigint(20)	NO		(NULL)	课程 id
name	varchar(64)	YES		(NULL)	
description	varchar(1024)	YES		(NULL)	学生描述
student_id	bigint(20)	NO		(NULL)	学生 id
student_name	varchar(64)	YES		(NULL)	学生名
type	varchar(16)	YES		teaching	学生授课状态(teaching 授课中, taught 授课结束, lose 未参加)
behavior	int(11)	YES		(NULL)	表现分, 结果/10
test	int(11)	YES		(NULL)	实验分, 结果/10
exam	int(11)	YES		(NULL)	考试分, 结果/10
score	int(11)	YES		(NULL)	最终成绩
credit	int(11)	YES		(NULL)	学分, 结果/10
status	varchar(16)	NO		enable	是否可用(enable,disable)
create_id	bigint(20)	YES		(NULL)	创建者 id
create_time	timestamp	YES		(NULL)	
update_time	timestamp	YES		(NULL)	
status	varchar(16)	NO		enable	是否禁用
create_id	bigint(20)	YES		(NULL)	创建者 id
create_time	timestamp	YES		(NULL)	
update_time	timestamp	YES		(NULL)	

4.4 系统详细设计的主要思路

将整个学生学分成绩管理系统拆分为几个微服务架构，分散服务压力，根据业务处理，即使其中一个模块出现问题，也不会影响其他的模块的运行和使用。

根据业务逻辑，我们将其分为用户管理、课程管理和学分管理^[9]。

用户管理模块我们可以创建、删除、修改、查询用户，给每个用户分配角色，给每个角色分配相对应的权限。在三个角色中，每个角色都可以查询和修改自己的用户信息，但只有管理员可以创建、修改、删除和查询其它用户，因为用户模块是系统最为核心的部分之一，关系到整个系统的完整运行。登录需要提供账号、密码和验证码，只有验证操作用户是系统用户后，我们根据用户的角色显示不同的界面，以提供不同的权限操作。我们还提供了如用户忘记密码可以提供邮箱发送验证码修改密码。

课程模块我们可以创建、删除、修改和查询课程的信息。管理员可以创建、修改、删除和查询课程信息，如课程的名称、课程类别和最核心的平时分、测试分、考试分的占比和该项课程总学分的设定。课程分为必修课和选修课，必修课由管理员指定课程-教师-学生的关系，而选修课由管理员指定课程-教师的关系，而学生选择课程和教师选择自己想要选修的课程。教师根据学生的平时表现给予平时分，根据测试成绩给予测试分，根据考试成绩给予考试分。

成绩模块我们可以创建、删除、修改和查询成绩的信息。教师对于选修课课程根据学生的表现和测试表现给予平时分和测试分，教师对于必修课课程根据学生的表现、测试表现和考试分数分别给予平时分、测试分和考试分，最后由比例占重得出最后的成绩。而学院管理员则在学生获得学校比赛奖项或参加学校活动，给予竞赛学分或创新与素质拓展学分^[10]。

4.5 系统模块的设计

（1）用户模块：用户输入登录、用户重置密码、用户修改个人信息、用户查询个人信息、管理员添加用户、管理员对其它人员的信息查询和修改；

（2）课程模块：管理员创建课程、管理员删除课程、管理员修改课程、管理员查询课程、管理员为课程分配教师、管理员为课程分配成绩比例和学分值、管理员为必修课课程分配学生、教师查询个人课程信息、学生查看个人课程信息；

（3）成绩模块：管理员查询学生成绩信息、管理员给学生添加非课程学分信息、教师查询学生成绩信息、教师修改学生成绩信息、学生查询课程成绩信息、学生查询非课程信息

4.6 前台功能模块的实现

4.6.1 用户登录时的设计

我们需要提供账号和密码的输入框，以及提交按钮，用来验证用户是否合法。同时我们添加图形验证码，防止脚本恶意登录。同时，我们需要提供用户忘记密码时的操作接口。



图 4-4 登陆页面图

在登录逻辑上我们需要对账号、密码、验证码进行初步验证，不合乎规定的请求信息不予以请求，直接在前端进行拦截，减少请求次数，降低服务器压力，对于账号和密码我们限制长度为 5~15 位，而且字符仅仅为字母和数字组合，对于验证码我们限制为 4 位，同样字符仅仅为字母和数字组合。

相关代码

```
var usernameReg = /^[0-9]{5,15}/;  
var passwordReg = /^[0-9a-zA-Z]{5,15}/;  
var captchaReg = /^[0-9a-zA-Z]{4}$/;  
//登陆  
function login(){  
  var username = $("#username").val();  
  var password = $("#password").val();  
  var captcha = $("#captcha").val();  
  console.log(username+"--"+password+"--"+captcha);  
  var boolUsername = usernameReg.test(username);  
  var boolPassword = passwordReg.test(password);  
  var boolCaptha = captchaReg.test(captcha);
```

重置密码



图 4-5 重置密码页面图

相关代码

```
<form class="pt-3"><div class="form-group">
<input type="email" class="form-control form-control-lg" id="email" placeholder="
    邮箱"></div><div class="form-group">
<input type="password" class="form-control form-control-lg" id="newPassword"
    placeholder="新密码"></div>
<div class="form-group"><input type="password" class="form-control
    form-control-lg" id="confirmPassword" placeholder="再输入一次密码
"> </div>
<div class="form-group"> <input type="text" class="form-control form-control-lg"
    style="width:50%;float: left;" id="captcha" placeholder="Captcha">
<button type="button" class="btn btn-gradient-primary mr-2" id="captchaBtn"
    onclick="getCaptha()">获取验证码</button>
</div><div class="mt-3"> <a class="btn btn-block btn-gradient-primary btn-lg
    font-weight-medium auth-form-btn" href="#"
    onclick="changePassword()">重置密码</a> </div></form>
```

登陆我们发生 AJAX 请求到后端，将输入的基本数据传输到后端中，让后端处理相关的数据，规定请求的 API 链接及数据类型和 JSON 数据。根据后端不同的返回结果我们做出不同的判断，成功我们进入用户界面，失败我们则显示弹出相关的错误信息，在错误之后，我们将弹出验证码输入框，需要对用户登录进行安全校验，验证用户是否为非法登录，如脚本或机器人。

```
success:function(data){
console.log("data",data);
if(data.code==200){
var str = JSON.stringify(data.data);
```

```

console.log("data.data",data.data);
// setCookie("user",data.data);
// var inf = getCookie("user");
// console.log("inf",inf);
window.location="http://localhost:8000/pages/index.HTML";
}else{
alert("code:"+data.code+"\n msg"+data.msg);
//登陆错误显示验证码
hideCaptcha();
showCaptcha();
}
},error:function(error){
console.log("error",error);
// window.location="/pages/error/error.HTML";
}
});

```



图 4-6 登陆页面提供验证码图

验证码只是用来判断用户是不是机器或脚本操作，在用户第一次登陆时，我们无需显示验证码，用户只需要提供账号和密码就可以，而当该用户名第一次登陆失败时，则需要提供验证码验证身份，登陆成功以后则不需要再提供验证码。

相关代码

//获取验证码

```

function showCaptcha(){
/*
第一次登陆不显示验证码，更改账号不显示验证码
*/

```

```

console.log("显示验证码");
$("#captchaDiv").HTML(
'<input type="text" class="form-control form-control-lg" style="width: 50%;float:
    left;" id="captcha" placeholder="captcha">'+
'</image>'
)
}

//更换账号隐藏验证码
function hideCaptcha(){
console.log("显示验证码");
$("#captchaDiv").HTML("");
}

```

4.6.2 用户信息模块设计

我们需要提供用户添加、删除、修改和查询界面，添加用户是从无到有，所以我们单独当成一个页面，不需要提供其他的数据支持，而删除和修改操作必须明确是哪一条数据项，我们才能予以修改和删除数据信息，所以我们可以将删除和修改操作放在查询页面上。

用户添加设计：我们可以添加账号名，密码，姓名和邮箱以及角色等关键信息的输入添加，还有性别、联系电话、组织机构、图片和个人描述等补充添加，使得用户的信息更为全面详细可靠。

账号	superadmin	密码	*****
姓名	lvgang1	邮箱	2637977081@qq.com
电话号码	13244231900	性别	男
角色	超级管理员	组织结构	北京大学
			jianchi.jpg
<div>  <div> 个人描述sda </div> </div>			

图 4-7 用户创建页面图

相关代码


```

<form          enctype="multipart/form-data;charset=utf-8"          method="post"
              name="userForm">
    <div class="form-group row"><label for="username" class="col-sm-1
col-form-label" style="width:15%">账号</label>
    <input type="text" class="form-control" id="form_username" placeholder="
    账号" style="width:30%">
    <label for="password" class="col-sm-1 col-form-label">密码</label>
        <input type="password" class="form-control" id="form_password"
placeholder="密码" style="width:30%"></div>
        <div class="form-group row"><label for="name" class="col-sm-1
col-form-label">姓名</label>
        <input type="email" class="form-control" id="form_name" placeholder="姓
        名" style="width:30%">
    <label for="email" class="col-sm-1 col-form-label">邮箱</label>
    <input type="text" class="form-control" id="form_email" placeholder="邮箱
        地址" style="width:30%"></div>
    <div class="form-group row"><div class="col-sm-6 row">
        <label for="phone" class="col-sm-2 col-form-label">电话号码</label>
        <input type="text" class="form-control" id="form_phone" placeholder="电
        话号码" style="width:30%;">
    </div>
    <label for="gender" class="col-sm-1 col-form-label">性别</label>
    <select class="form-control" id="form_gender" style="width:20%;"><option
        value="male">男</option><option value="female">女</option>
    </select></div>
        <div class="form-group row"><label for="role" class="col-sm-1
col-form-label" >角色</label>
        <select class="form-control" id="form_role" style="width:20%;"
onclick="getOrgList()"></select>
    <label for="gender" class="col-sm-1 col-form-label" style="width:50px">组
        织结构</label>
    <select class="form-control" id="form_org" style="width:20%;"></select>
    <div class="col-sm-4 row">
        <input type="file" name="img[]" class="file-upload-default"
id="form_avatar" onchange="uploadAvatar()">
    <input type="text" class="form-control file-upload-info" style="width:50%;"
        disabled="" placeholder="上传头像" >
    <span class="input-group-append">

```

```

        <button class="file-upload-browse btn btn-gradient-primary"
            type="button">上传</button></span>div></div>
<div class="form-group row">
    <div style="width:140px;height:160px;">
        
    </div>
    <textarea class="form-control" id="form_description" rows="4"
        style="width:70%">个人描述</textarea>
</div>
</form>

```

用户查询、删除和修改操作：这里我们可以使用账号、名字和邮箱进行模糊查询，性别、角色、组织机构和状态进行准确查询，对于删除操作，我们只是对其进行逻辑删除，将其状态为置位为禁用状态，使得该用户变得不可用。

图 4-8 用户查询、删除和修改操作页面图

图 4-9 用户详情页面图

相关代码

```

<table class="table table-striped">
<thead><tr><th>头像</th><th>账号</th><th>名字</th><th>性别</th><th>邮箱
</th><th>电话</th><th>角色</th><th>所属</th><th colspan="3"
style="">操作</th>
</tr></thead>
<tbody id="userList"><tr id='1'><td class="py-1"></td>
<td>lvgang</td><td>吕钢</td><td>男</td><td>2637977000@qq.com</td><td>18844210000</td><td>超级
管理员</td><td>北华大学</td>
<td style="color:#07cd22" onclick='able(1,"enable")>启用</td><td
style="color:#fed713" onclick='detail()'>详情</td>
<td style="color:#b66dff" onclick='change()'>修改</td></tr>
<tr id='2'><td class="py-1"></td>
<td>lvgang1</td><td>吕钢1</td><td>男</td><td>2637977001@qq.com</td><td>18844210001</td><td>超级
管理员</td><td>北华大学</td>
<td onclick='able(2,"enable")>启用</td><td onclick='detail()'>详情</td><td
onclick='change()'>修改</td>
</tr></tbody></table>

```

4.6.3 课程学分管理设计

我们需要先添加一个基础课程，设置课程的一些基本信息，如课程名，课程类型为选修还是必修，所属于哪个学校、学院或班级，课程名不可以重复。

成功添加课程后，我们基于课程添加，每一个课程都可以添加多个教师，一位教师也可以参与多门课程，同时我们给这门课程设定平时分、测试分和考试的占比，以及最终的课程学分，这样就形成了教学列表。

根据选修课还是必修课我们进行不同操作，选修课我们由学生进行抢占操作，而必修课我们由管理员指定学生选择。

添加课程：我们可以对课程名、分类、学分、学时、预计人数、平时分百分比、实验分百分比和考试分等百分的关键信息进行填充，对于必修课，预计人数为提示信息，对于选修课预计人数为最多人数。

课程管理

添加课程

课程名	<input type="text" value="高等数学2"/>	大学	<input type="text" value="北华大学"/>		
学院	<input type="text" value="计算机科学技术学院"/>	系/班	<input type="text" value="计算机科学与技术"/>		
分类	<input type="text" value="必修"/>	学分	<input type="text" value="5"/>	学时	<input type="text" value="50"/>
		预计人数	<input type="text" value="40"/>		
平时分百分比	<input type="text" value="0"/>	实验分百分比	<input type="text" value="0"/>	考试分百分比	<input type="text" value="100"/>

提交

图 4-10 课程添加页面图

相关代码

```
<form      enctype="multipart/form-data;charset=utf-8"      method="post"
      name="userForm">
  <div class="form-group row">
    <label for="name" class="col-sm-1 col-form-label">课程名</label>
    <div class="col-sm-3"><input type="text" class="form-control" id="name"
      placeholder="课程名"></div></div>
  <div class="form-group row">
    <label for="type" class="col-sm-1 col-form-label">分类</label>
    <div class="col-sm-3">
      <select class="form-control" id="type" ><option value="required"> 必修
        </option><option      value="elective">      选      修
        </option></select></div></div>
  <div class="form-group row">
    <label for="university" class="col-sm-1 col-form-label">大学</label>
    <div class="col-sm-3">
      <select  class="form-control"  id="university"
        onchange="changeOption('university','faculty')"><option> 全 部
      </option></select></div></div>
    <div class="form-group row"><label for="faculty" class="col-sm-1
      col-form-label">学院</label>
      <div class="col-sm-3"><select class="form-control" id="faculty"
        onchange="changeOption('faculty','subject')"><option> 全 部
      </option></select></div></div>
    <div class="form-group row"><label for="subject" class="col-sm-1
```

```

col-form-label">系/班</label>
<div class="col-sm-3"><select class="form-control" id="subject" ><option>
    全部</option></select></div></div>
</form>
课程列表

```

账号管理
用户列表

课程名 分类

大学 学院 系/班

课程名	课程类型	所属大学	所属学院	所属系/班	操作
高等数学3	选修	北华大学	计算机科学技术学院	全部	删除 详情 修改 添加教师
高等数学4	选修	北华大学	全部	全部	删除 详情 修改 添加教师
高等数学5	选修	北华大学	全部	全部	删除 详情 修改 添加教师
大学英语1	选修	北华大学	全部	全部	删除 详情 修改 添加教师
课程1	必修	全部	全部	全部	删除 详情 修改 添加教师

图 4-11 课程列表页面图

页面分为两部分，一部分为筛选条件，我们通过不同的筛选的条件，缩减查询的数据时间和数据数量，筛选条件为

相关代码

```

<form      enctype="multipart/form-data; charset=utf-8"      method="post"
      name="userForm">
<div class="form-group row">
<label for="name" class="col-sm-1 col-form-label">课程名</label>
  <div class="col-sm-3"><input type="text" class="form-control" id="name"
    placeholder="课程名">
  </div>
<label for="type" class="col-sm-1 col-form-label">分类</label>
  <div class="col-sm-3"><select class="form-control" id="type"><option
    value="all">全部</option>
  <option value="required">必修</option><option value="elective">选修</option>
  </select></div>
  <button type="button" onclick="getCourseList()" style="margin-left:5%" class="btn
    btn-gradient-primary mr-2">查询</button>
</div><div class="form-group row"><label for="university" class="col-sm-1
  col-form-label">大学</label>

```

```

<div class="col-sm-3"><select class="form-control" id="university"
    onchange="changeOption('university','faculty')">
<option value="all">全部</option></select></div>
<label for="faculty" class="col-sm-1 col-form-label">学院</label>
<div class="col-sm-3"><select class="form-control" id="faculty"
    onchange="changeOption('faculty','subject')">
<option value="all">全部</option></select></div>
<label for="subject" class="col-sm-1 col-form-label">系/班</label>
<div class="col-sm-3">
<select class="form-control" id="subject"><option value="all"> 全 部
    </option></select>
</div></div></form>

```

教学列表：我们已经将老师和课程进行匹配了，现在我们要给这些老师分配学生进行教学，同时我们可以修改教学信息，对其进行删除或者查看详情。

账号管理
用户列表

教学名 课程类型

所属课程 授课老师 状态

教学名	当前人数	课程类型	教师名	操作
班级1	2	未滿	teacher2	删除 详情 修改 添加学生
班级4	0	未滿	teacher0	删除 详情 修改 添加学生
班级9	0	未滿	teacher0	删除 详情 修改 添加学生

图 4-12 教学列表页面图

相关代码

```

<form enctype="multipart/form-data; charset=utf-8" method="post"
    name="userForm">
<div class="form-group row"><label for="name" class="col-sm-1 col-form-label">
    教学名</label>
<div class="col-sm-3"><input type="text" class="form-control" id="name"
    placeholder="教学名">
</div><label for="type" class="col-sm-1 col-form-label">课程类型</label>
<div class="col-sm-3"><select class="form-control" id="type">
<option value="all">全部</option><option value="full">满员</option>
    <option value="surplus">待选</option><option value="stop">停止</option>
</select></div>
    <button type="button" onclick="getTeachList()" style="margin-left:5%" class="btn
        btn-gradient-primary mr-2">查询</button>

```

```

</div> <div class="form-group row"><label for="university" class="col-sm-1
col-form-label">所属课程</label>
<div class="col-sm-3"><select class="form-control" id="course" onchange="">
<option value="all">全部</option></select></div>
<label for="teacher" class="col-sm-1 col-form-label">授课老师</label>
<div class="col-sm-3"><select class="form-control" id="teach" onchange="">
<option value="all">全部</option></select></div>
<label for="status" class="col-sm-1 col-form-label">状态</label>
<div class="col-sm-3"><select class="form-control" id="status">
<option value="enable"> 启 用 </option><option value="disable"> 禁 用
</option></select>
</div></div></form></div></div>

```

班级列表：我们已经添加了学生到课程和老师名下，同时我们可以移除他们，查看学生的分数和当前的成绩详情。

账号管理
班级列表

教学名 课程类型

状态

学生名	班级名	学生授课状态	表现分	实验分	考试分	最终成绩	学分	操作
student1	班级1	授课中	0	0	0	0	0	删除
student2	班级1	授课中	0	0	0	0	0	删除

图 4-13 班级列表页面图

相关代码

```

<form          enctype="multipart/form-data;charset=utf-8"          method="post"
name="userForm">
<div class="form-group row"><label for="name" class="col-sm-1 col-form-label">
教学名</label>
<div class="col-sm-3"><input type="text" class="form-control" id="name"
placeholder="教学名"></div>
<label for="type" class="col-sm-1 col-form-label">课程类型</label>
<div class="col-sm-3"><select class="form-control" id="type"><option value="all">
全部</option><option value="full">满员</option>
<option value="surplus"> 待 选 </option><option value="stop"> 停 止
</option></select></div>
<button type="button" onclick="getTeachList()" style="margin-left:5%" class="btn
btn-gradient-primary mr-2">查询</button>

```

```

</div><div class="form-group row"><label for="university" class="col-sm-1
col-form-label">所属课程</label>
<div class="col-sm-3"> <select class="form-control" id="course" onchange="">
<option value="all">全部</option></select></div>
<label for="teacher" class="col-sm-1 col-form-label">授课老师</label>
<div class="col-sm-3"><select class="form-control" id="teach" onchange="">
<option value="all"> 全 部 </option></select></div><label for="status"
class="col-sm-1 col-form-label">状态</label>
<div class="col-sm-3"><select class="form-control" id="status">
<option value="enable">启用</option>
<option value="disable">禁用</option></select></div></div></form>

```

学生选课列表：对于选修课程，我们给学生提供选择的余地，可以查看该课程的当前人数、学分、最大人数、教学时长、教师，点击选取操作后，就会在学生的课程中添加了这一课程

教学名	当前人数	学分	最大人数	教学时长	课程类型	教师名	操作
选修课教学1	0	5	40	15	未滿	teacher0	选取

图 4-14 学生选课列表页面图

相关代码

```

<form          enctype="multipart/form-data;charset=utf-8"          method="post"
name="userForm">
<div class="form-group row"><label for="name" class="col-sm-1 col-form-label">
教学名</label>
<div class="col-sm-3"><input type="text" class="form-control" id="name"
placeholder="教学名"></div>
<label for="type" class="col-sm-1 col-form-label"> 课 程 类 型 </label><div
class="col-sm-3">
<select class="form-control" id="type"><option value="all"> 全 部 </option><option
value="full">满员</option>
<option value="surplus"> 待 选 </option><option value="stop"> 停 止
</option></select></div>
<button type="button" onclick="getTeachList()" style="margin-left:5%" class="btn
btn-gradient-primary mr-2">查询</button>
</div><div class="form-group row"><label for="university" class="col-sm-1
col-form-label">所属课程</label>

```



```

<div class="col-sm-3"><select class="form-control" id="course" onchange="">
  <option value="all"> 全 部 </option></select></div><label for="teacher"
    class="col-sm-1 col-form-label">授课老师</label>
<div class="col-sm-3"><select class="form-control" id="teach"
  onchange=""><option value="all">全部</option>
</select></div><label for="status" class="col-sm-1 col-form-label"> 状 态
  </label><div class="col-sm-3">
<select class="form-control" id="status"><option value="enable">启用</option>
<option value="disable">禁用</option></select></div></div></form>

```

学生成绩列表：学生可以根据教学名、课程类型以及状态查询自己选修课程或必修课程的状态、班级名、学生授课状态、表现分、实验分、考试分、最终成绩和最后获得学分

学生名	班级名	学生授课状态	表现分	实验分	考试分	最终成绩	学分
student1	班级19	授课中	0	0	0	0	0

图 4-15 学生成绩列表页面图

相关代码

```

<form enctype="multipart/form-data; charset=utf-8" method="post"
name="userForm"><div class="form-group row">
  <label for="name" class="col-sm-1 col-form-label">教学名</label>
  <div class="col-sm-3"><input type="text" class="form-control" id="name"
placeholder="教学名">
  </div><label for="type" class="col-sm-1 col-form-label"> 课 程 类 型
</label><div class="col-sm-3">
    <select class="form-control" id="type"><option value="all">全部</option>
    <option value="full"> 满 员 </option><option value="surplus"> 待 选
</option><option value="stop">停止</option>
  </select></div><button type="button" onclick="getTeachList()"
style="margin-left:5%" class="btn btn-gradient-primary mr-2">查询</button>
  </div><div class="form-group row"><label for="status" class="col-sm-1
col-form-label">状态</label>
  <div class="col-sm-3"><select class="form-control" id="status"><option
value="enable">启用</option>
  <option value="disable">禁用</option></select></div></div></form>

```

学生学分列表：此处可以查看学生获得的费课程学分，通过学分类型进行模糊查询可以获得学生获得哪些类型的学分以及学分的数值是多少。

学分类型

创新学分

查询

学号	名字	学分	类型
1131	student1	4	创新学分
1131	student1	3	创新学分

图 4-16 学生成绩列表页面图

相关代码

```
<form      enctype="multipart/form-data;charset=utf-8"      method="post"
name="userForm"><div class="form-group row">
  <label for="name" class="col-sm-1 col-form-label">教学名</label>
  <div class="col-sm-3"><input type="text" class="form-control" id="name"
placeholder="教学名">
</div><label for="type" class="col-sm-1 col-form-label"> 课 程 类 型
</label><div class="col-sm-3">
  <select class="form-control" id="type"><option value="all">全部</option>
  <option value="full"> 满 员 </option><option value="surplus"> 待 选
</option><option value="stop">停止</option>
</select></div><button      type="button"      onclick="getTeachList()"
style="margin-left:5%" class="btn btn-gradient-primary mr-2">查询</button>
</div><div class="form-group row"><label for="status" class="col-sm-1
col-form-label">状态</label>
<div class="col-sm-3"><select class="form-control" id="status"><option
value="enable">启用</option>
<option value="disable">禁用</option></select></div></div></form>
```

教学管理列表：教师可以根据教学名模糊查询，根据课程类型和所属课程进行准确查询，获得课程的相关信息如课程类型和当前人数等信息，教师可以对指定的课程进入开始给予分数

账号管理				
用户列表				
教学名	<input type="text" value="教学名"/>	课程类型	<input type="text" value="全部"/>	查询
所属课程	<input type="text" value="全部"/>			
教学名	当前人数	课程类型	教师名	操作
选修课教学1	0	未满	teacher1	给予分数

图 4-17 教学管理列表页面图

相关代码

```
<form      enctype="multipart/form-data;charset=utf-8"      method="post"
name="userForm">
    <div      class="form-group      row"><label      for="name"      class="col-sm-1
col-form-label">教学名</label>
        <div      class="col-sm-3"><input      type="text"      class="form-control"      id="name"
placeholder="教学名">
        </div><label      for="type"      class="col-sm-1      col-form-label"> 课 程 类 型
</label><div      class="col-sm-3">
            <select      class="form-control"      id="type"><option      value="all">全部</option>
            <option      value="full">满员</option><option      value="surplus">待选</option>
            <option      value="stop">停止</option></select></div>
            <button      type="button"      onclick="getTeachList()"      style="margin-left:5%"
class="btn btn-gradient-primary mr-2">查询</button>
        </div><div      class="form-group      row"><label      for="university"      class="col-sm-1
col-form-label">所属课程</label>
            <div      class="col-sm-3"><select      class="form-control"      id="course"      onchange="">
            <option      value="all">全部</option></select>
            </div></div></form>
```

成绩管理列表：教师可以根据教学名模糊查询，根据课程类型和所属教学进行准确查询，获得教学的相关信息如学生的姓名、班级名、授课状态、表现分、实验分、考试分、最终成绩以及想关分数，教师可以对指定的课程进入开始给予分数，对表现分、实验分、考试分进行修改添加。

教学名	<input type="text" value="教学名"/>	课程类型	<input type="text" value="全部"/>	<input type="button" value="查询"/>
状态	<input type="text" value="启用"/>			

学生名	班级名	学生授课状态	表现分	实验分	考试分	最终成绩	学分	操作
student1	班级19	授课中	0	0	0	0	0	修改分数
student2	班级19	授课中	0	0	0	0	0	修改分数

图 4-18 成绩管理列表页面图

相关代码

```
<form      enctype="multipart/form-data;charset=utf-8"      method="post"
name="userForm">
    <div      class="form-group      row"><label      for="name"      class="col-sm-1
col-form-label">教学名</label>
```

```

<div class="col-sm-3"><input type="text" class="form-control" id="name"
placeholder="教学名">
</div><label for="type" class="col-sm-1 col-form-label">课程类型</label>
<div class="col-sm-3"><select class="form-control" id="type">
<option value="all">全部</option><option value="full">满员</option>
<option value="surplus">待选</option><option value="stop">停止</option>
</select></div><button type="button" onclick="getTeachList()"
style="margin-left:5%" class="btn btn-gradient-primary mr-2">查询</button>
</div><div class="form-group row"><label for="status" class="col-sm-1
col-form-label">状态</label>
<div class="col-sm-3"><select class="form-control" id="status"><option
value="enable">启用</option>
<option value="disable">禁用</option></select></div></div></form>

```

学分修改：教师可以对指定的学生和课程，进入开始给予分数，对表现分、实验分、考试分进行修改添加，同时可以修改授课的状态。

图 4-19 学分修改页面图

相关代码

```

<form enctype="multipart/form-data; charset=utf-8" method="post"
name="userForm">
<div class="form-group row"><label for="username" class="col-sm-1
col-form-label" style="width:15%">学生名
</label><input type="text" class="form-control" id="studentName" readonly
style="width:30%"><input type="text" class="form-control" id="studentId"
readonly style="width:30%;display: none">
<label for="gender" class="col-sm-1 col-form-label">授课状态</label>
<select class="form-control" id="type" style="width:20%;">
<option value="teaching">授课中</option><option value="teached">授课完
</option>
<option value="lose">缺课</option></select>
</div><div class="form-group row">

```

```

        <label for="name" class="col-sm-1 col-form-label">平时分</label>
        <input type="email" class="form-control" id="behavior" value="0"
style="width:15%">
        <label for="email" class="col-sm-1 col-form-label">测试分</label>
        <input type="text" class="form-control" id="test" value="0"
style="width:15%">
        <label for="phone" class="col-sm-2 col-form-label">考试分</label>
        <input type="text" class="form-control" id="exam" value="0"
style="width:15%;">
    </div></form>

```

4.7 后端功能模块的实现

4.7.1 用户登陆模块设计

用户登陆，我们需要验证该用户是否存在，是否合法，是否需要验证码，当用户成功登陆后，我们需要查询出用户的角色和拥有哪些权限，同时将用户信息缓存到 redis 中，可以做到用户不用保持持久存在^[11]。我们设置了唯一 code 码用来确认用户在 redis 中的缓存信息^[12]，token 则用来存储一些基本信息和用户登录的时效性，超过我们规定的时间（24 小时），则 token 失效^[13]，我们不允许其再继续访问。

相关代码

```

@PostMapping("login")
public ResultVo login(@RequestBody @Validated({Login.class}) UserVo temp)
    throws Exception {
    VoUtils.copyProperties(temp, userVo, "commonName", "password", "random",
        "captcha");
    UserVo persist = userService.findUser(userVo.getCommonName());
    if (persist != null) {
        if (persist.getCaptchaStatus()) {
            if (!StringUtils.isEmpty(userVo.getCaptcha())) {return
resultVo.getInstance(MsgConstant.CAPTCHA_NOT_EXISTED);
            }if (StringUtils.isEmpty(userVo.getRandom()) &&
StringUtils.isEmpty(userVo.getCaptcha())) {
                String captcha = CaptchaUtil.getCaptcha(template,
userVo.getRandom());
                if (userVo.getCaptcha().equalsIgnoreCase(captcha))
                    {loginUser(persist);
                    }
                else
                    {resultVo.getInstance(MsgConstant.CAPTCHA_ERROR);}
            }
        }
    }
}

```

```

        } else {resultVo.getInstance(MsgConstant.CAPTCHA_ERROR);
        }} else {loginUser(persist);}} else {
resultVo.getInstance(MsgConstant.LOGIN_ERROR);
}
return resultVo;
}

```

用户输入账号密码时，我们需要将密码转换成我们的加密方式^[14]，再与数据库中的数据进行对比，登陆成功后，我们将用户信息缓存到 redis 中，并给以用户权限和有效登陆时效。

相关代码

```

/**
 * 验证用户并记录用户状态
 */
private void loginUser(UserVo persist) throws JsonProcessingException {
    if (BCrypt.checkpw(userVo.getPassword(), persist.getPassword())) {
        if (StatusConstant.ENABLED.equals(persist.getStatus())) {
            String code = persist.getCode();
            String          jwtToken          =          jwtUtil.createJWT(code,
            objectMapper.writeValueAsString(new          UserBo(persist.getUserId(),
            persist.getCode(),          persist.getUsername(),          persist.getName(),
            persist.getPhone(),
            persist.getEmail(),persist.getGender(),persist.getAvatar())));
            persist.setJwtToken(new          JwtToken(code,
            TypeConstant.BEARER_TOKEN, jwtToken, jwtUtil.getTtlMillis()));
            // 缓存用户信息 template.opsForValue().set(code,
            objectMapper.writeValueAsString(persist));
            // 不需要验证码 persist.setCaptchaStatus(false);
            userService.updateCaptchaStatus(persist);
            resultVo.getInstance(HttpStatus.OK.toString(), persist);
        } else {resultVo.getInstance(MsgConstant.LOGIN_USER_ERROR);
        }
    } else {// 需要验证码 persist.setCaptchaStatus(true);
            userService.updateCaptchaStatus(persist);
            resultVo.getInstance(MsgConstant.LOGIN_ERROR);
        }
    }
}

```

4.7.2 用户信息模块设计

我们可以对用户信息进行最基础的添加、删除、修改和查询用户的信息，对于密码，我们不建议直接存储明文密码，一旦数据库信息被泄露，账号信息就不会安全，所以我们需要对密码进行我们自定义的规则进行加密操作后再存储到数据中。

相关代码

```
/**
 * 创建
 */
@PostMapping("add")
public ResultVo add(@RequestBody @Validated( {UserAdd.class }) UserVo temp )
    throws Exception {
    VoUtils.copyProperties(temp, userVo,"username", "name", "email", "phone",
        "password","gender", "avatar", "description", "orgId", "roleVoList",
        "status");
    if(userService.isExistedUser(userVo))
        {resultVo.getInstance(MsgConstant.USER_EXISTED, userVo);
        } else {if (userVo.getStatus() != null &&
            StatusEnum.getStatus(userVo.getStatus())== null) {
            resultVo.getInstance(MsgConstant.INVALID_DATA);
        }else {userVo.setCode(UUIDUtil.getUUID());
            userVo.setCreatorId(loginUser.getUserId());
            userVo.setPassword(BCrypt.hashpw(userVo.getPassword(),
                BCrypt.gensalt(bcryptSalt)));
            userVo = userService.saveUser(userVo);
            if (userVo != null) {resultVo.getInstance(HttpStatus.OK.toString(),
                userVo);
            } else {resultVo.getInstance(MsgConstant.CREATE_ERROR, userVo);}}}
    return resultVo;
}
```

4.7.3 课程信息模块设计

我们可以课程信息进行最基础的添加、删除、修改和查询课程的基本信息，我们需要对课程表、教学表和班级表进行列表查询和详情查询，获得更多的了解。

对于课程我们添加相关的筛选信息，在数据库中查询列表信息，获取相对应的数据信息。

相关代码

```
VoUtils.copyProperties(temp, courseVo,
```

```

        "name","type","universityId","universityName","facultyId","facultyName",
        "subjectId","subjectName", "page", "rows");
List<Course> list = courseService.findByPage(courseVo);
PageInfo<Course> pageInfo = new PageInfo<Course>(list);
resultVo.getInstance(HttpStatus.OK.toString(), pageInfo);
return resultVo;

```

对于课程我们添加相关的基本信息，将数据添加到数据库成功存储，并得到存储的结果

相关代码

```

VoUtils.copyProperties(temp,
        courseVo,"name","type","universityId","universityName","facultyId","facultyName",
        "subjectId","subjectName");
if(courseService.isExistedCourse(courseVo)){
    resultVo.getInstance(MsgConstant.COURSE_EXISTED);
}else {
    courseVo.setCode(UUIDUtil.getUUID());
    courseVo.setCreateId(loginUser.getUserId());
    courseVo.setCreateTime(new Date());
    courseVo = courseService.saveCourse(courseVo);
    if(courseVo!=null){
        resultVo.getInstance(HttpStatus.OK.toString(),courseVo);
    }else {
        resultVo.getInstance(MsgConstant.CREATE_ERROR);
    }
}
return resultVo;
}

```

4. 7. 4 课程管理模块设计

课程的添加

```

courseVo,"name","type","universityId","universityName","facultyId","facultyName",
        "subjectId","subjectName");
if(courseService.isExistedCourse(courseVo)){
    resultVo.getInstance(MsgConstant.COURSE_EXISTED);
}else {
    courseVo.setCode(UUIDUtil.getUUID());
    courseVo.setCreateId(loginUser.getUserId());
    courseVo.setCreateTime(new Date());

```



```

        courseVo = courseService.saveCourse(courseVo);
        if(courseVo!=null){
            resultVo.getInstance(HttpStatus.OK.toString(),courseVo);
        }else {
            resultVo.getInstance(MsgConstant.CREATE_ERROR);
        }
    }
}

```

课程的修改

```

VoUtils.copyProperties(temp,
    courseVo,"courseId","name","type","universityId","universityName","facultyId","facultyName","subjectId","subjectName");
courseVo.setUpdateTime(new Date());
Boolean bool = courseService.updateCourse(courseVo);
if(bool){
    resultVo.getInstance(HttpStatus.OK.toString(),bool);
}else {
    resultVo.getInstance(MsgConstant.INVALID_DATA);
}
}

```

课程的详细查询

```

CourseVo courseVo = new CourseVo();
courseVo.setCourseId(id);
courseVo.setUpdateTime(new Date());
courseVo.setStatus(StatusEnum.DISABLED.toString());
Boolean bool = courseService.updateCourse(courseVo);
if(bool){
    resultVo.getInstance(HttpStatus.OK.toString(),bool);
}else {
    resultVo.getInstance(MsgConstant.INVALID_DATA);
}
}

```

课程的列表查询

```

VoUtils.copyProperties(temp, courseVo,
    "name","type","universityId","universityName","facultyId","facultyName","subjectId","subjectName", "page", "rows");
List<Course> list = courseService.findByPage(courseVo);
PageInfo<Course> pageInfo = new PageInfo<Course>(list);
resultVo.getInstance(HttpStatus.OK.toString(), pageInfo);
return resultVo;

```

第 5 章 结果分析

5.1 软件测试

5.1.1 黑盒测试前端页面

登陆页面测试

输入错误的用户名和密码，点击登陆按钮，页面弹出警告信息，提示信息为：用户名不存在或密码错误，无法进入正常页面[6] [11]。



图 5-1 登陆错误图

登陆错误一次后，页面会多出输入验证的输入框，验证码为必须的输入，如果没有输入则弹出错误，错误信息为验证码长度必须达到 4~6 位之间



图 5-2 登陆验证码错误图

验证码超过 6 位也会弹出错误信息：验证码长度必须达到 4~6 位之间



图 5-3 登陆验证码错误图

输入正确的用户名密码后，则可以正确的进入

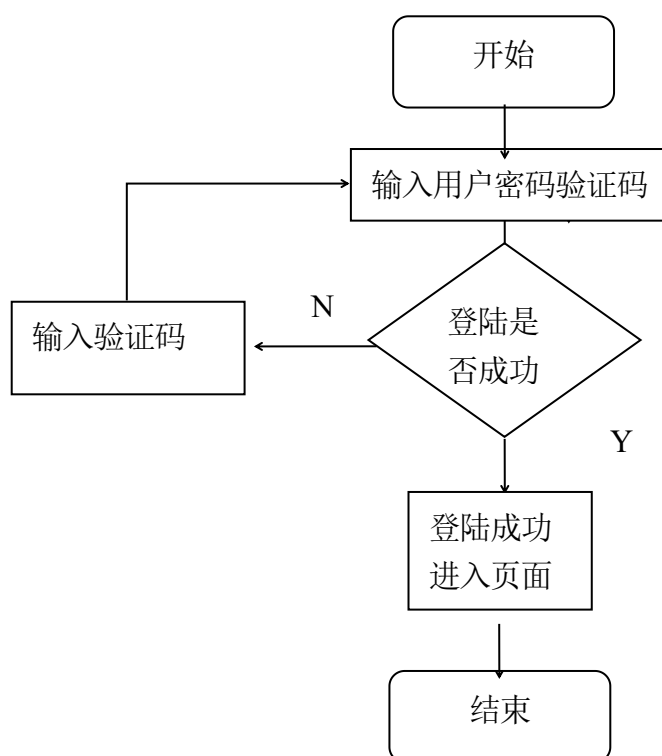


图 5-4 登陆验证码错误流程图

操作步骤：1.输入用户名 2.输入密码 3.输入验证码 4.点击登陆 5.登陆成功 6.弹出框显示错误的信息

第一次登陆成功：1->2->4->5

第一次登陆失败，第二次登陆成功：1->2->4->6->1->2->3->4->5

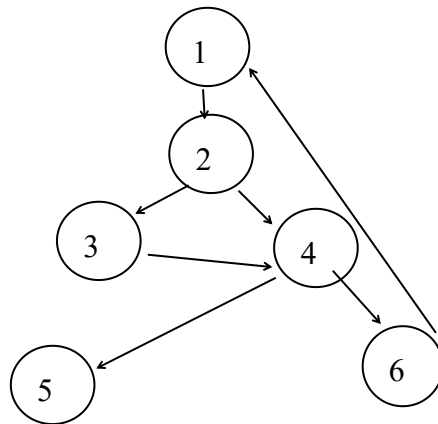


图 5-5 登陆验证码错误过程图

5.1.2 白盒测试后端代码

针对后端代码用户登陆模块代码

```
@PostMapping("login")
public ResultVo login(@RequestBody @Validated({Login.class}) UserVo temp)
    throws Exception {
    VoUtils.copyProperties(temp, userVo, "commonName", "password",
        "random", "captcha");
    UserVo persist = userService.findUser(userVo.getCommonName());
    if (persist != null) {
        if (persist.getCaptchaStatus()) {
            if (StringUtils.isEmpty(userVo.getRandom()) &&
                StringUtils.isEmpty(userVo.getCaptcha())) {
                String captcha = CaptchaUtil.getCaptcha(template,
                    userVo.getRandom());
                if (userVo.getCaptcha().equalsIgnoreCase(captcha)) {
                    loginUser(persist);
                } else {

resultVo.getInstance(MsgConstant.CAPTCHA_ERROR);
                }
            } else {
                resultVo.getInstance(MsgConstant.CAPTCHA_ERROR);
            }
        }
    }
}
```

```

        } else {
            loginUser(persist);
        }
    } else {
        resultVo.getInstance(MsgConstant.LOGIN_ERROR);
    }
    return resultVo;
}

```

我们与之对应的黑盒测试代码为

@test

```

public void testLogin(){
    AccountController account = new AccountController();
    User user =new User();
    //账号错误
    user.setUsername("aaa");
    user.setPassword("111");
    ResultVo resultVo = account.login(user);
    assertEquals(resultVo.getCode(),"1110002");

    //密码输入错误
    user.setUsername("superadmin");
    user.setPassword("111");
    ResultVo resultVo = account.login(user);
    assertEquals(resultVo.getCode(),"1110002");

    //验证码输入为空
    user.setUsername("superadmin");
    user.setPassword("12345678");
    user.setPassword(null);
    ResultVo resultVo = account.login(user);
    assertEquals(resultVo.getCode(),"1110022");

    //验证码错误
    user.setUsername("superadmin");
    user.setPassword("12345678");
    user.setPassword("1111");
}

```

```

ResultVo resultVo = account.login(user);
assertEquals(resultVo.getCode(),"1110021");

//登陆成功
user.setUsername("superadmin");
user.setPassword("12345678");
user.setPassword("abcd");
ResultVo resultVo = account.login(user);
assertEquals(resultVo.getCode(),"200");
}

```

表 5.1 用例测试表

用例编号	用户名	密码	验证码	预期输出	结果输出
1	superadmin	12345678	无	200	200
2	admin	12345678	无	1110002	1110002
3	superadmin	12345678	无	1110022	1110022
4	superadmin	12345678	all	1110021	1110021
5	superadmin	123456789	abcd	1110002	1110002
6	superadmin	12345678	abcd	200	200

Windows 环境上，直接使用 IntelliJ IDEA 运行项目来启动服务工程，通过 Go-Server 访问页面可以正常的运行。

Ubuntu 环境上，SpringBoot 项目支持打包成 jar 包可以移植到 Ubuntu 环境上运行，通过 Go-Server 访问页面可以正常的运行。

多个用户在不同机器上登陆，访问服务，同时进行点击操作，增加、删除、修改和查询操作。同时可以使用 Webbench 工具来对系统进行压力测试，Webbench 可以测试同一硬件上不同服务的性能，以及不同硬件上的相同服务。webbench 的标准测试可以向我们展示有关服务器的两件事：每秒请求数和每秒传输的数据量。Webbench 不仅能够测试静态页面，还能够测试动态页面（ASP, PHP, JAVA, CGI）。

还支持对包括 SSL 的安全网站（例如电子商务站点）进行静态或动态性能测试。我们新建 100 个线程，每个线程同时发送 1000 次请求，计算服务压力[9]。

5.2 结果分析

系统在硬件条件支持足够的情况下允许百万级的并发量，而不会出现问题，可以稳定运行，保证数据不会出现紊乱，如当系统的规模够大时，我们可以采用分布式的服务架构来支持，采用 MongoDB 或者 ElasticSearch 来代替 MySQL 进行大量的数据存储、删除、修改和查询。

结 论

系统设计完成的结果说明了一下几点：

(1) 在框架结构上，无论是大型项目还是小型项目，SpringBoot 都可以完美的支持。

(2) 在开发难度上。SpringBoot 相较于软件开发、ssh 框架和 SSM 框架，配置相当方便简单，而且层次清晰，我们可以在一台服务器上使用不同的端口来同时支持多个服务同时运行，配置和一老配置简单有效。

(3) 本系统的开发意义上。我们使得原来的数据录入、存储、修改和查询变得更加的安全、有效和快速以及稳定。

(4) 在系统的维护上。我们提供了 Monitor 监控机制，每当后台服务出现问题时，我们可以查询 Monitor 提供的系统日志来进行查询、定位并予以修改。

(5) 技术使用上。Redis 数据库存储作为缓存机制明显优于 Session 机制，当服务中断时，Session 中缓存的数据会消失，而 Redis 会继续保留。同时 Redis 的存储机制使用与队列，可以用来恒定唯一数量，使得 40 份数据在高并发情况下^[15]，只会存在 40 份，完美契合抢课机制。

(6) 学习新技术。最重要的是练习和实践，这样才能尽快了解它，掌握它。

(7) 一个系统的发展，经验是最重要的，缺乏经验，难免会有很多考虑因素会被遗漏丢失，造成系统在后续使用中缺少或部分功能出现问题。

(8) 系统界面必须美观。优美的页面可以吸引用户，友好和齐全的功能页面能使得用户的体验更为美好。

(9) 系统的还有很大的发展空间。如当系统的规模够大时，我们可以采用分布式的服务架构来支持，采用 MongoDB 或者 ElasticSearch 来代替 MySQL 进行大量的数据存储、删除、修改和查询。

基于学生成绩学分制管理系统设计采用面向软件工程结构编程、面向接口开发的设计思想，首先分析需求，确定数据流图和数据字典。系统分为几个部分：用户模块，课程模块和学分模块，根据数据流方向分析，总结设计和详细设计，大纲设计确定界面，设计数据库，并制定大纲设计规范。详细设计确定系统模块并使用流程图来构建程序并在编码阶段对其进行代码开发。学生成绩学分制管理系统可以实现学生的基本信息，教师的基本信息，学生查询，课程查询，学分统计等功能。但是，由于缺乏软件工程知识和编码程序设计，在项目的概要设计和详细设计中有很多不成熟、不合理的方法，系统中有许多不合理或不是期待的地方，所以将来我希望改进这个系统和可维护性。

致 谢

大学四年的生活悄悄然流逝在我们的指间，我们奔往梦想，走向社会，步入职场。在大四的这年，我们迈向了社会，有了第一份工作，领了第一份薪酬，开启了全新的生活体验。写完这篇论文后，一股子自豪、如释重负但却带着点唏嘘得到感觉涌上心头。

首先非常诚挚的、充满敬意的感谢我的指导老师。在大三那年开始授课于我，再一次实验课上，老师询问我想不想在毕业时跟她完成毕业设计，就这样有幸得到老师的指导论文的编写，修改和审查。她在忙碌的教学工作中，不辞辛苦挤出自己的休闲时间，甚至是在周末，为我做出最适合的指导，心平气和的指出我们问题的错误和不适之处。同时，还有其他教导过我的老师们，谢谢你们严谨但不缺风趣、细致但放开空间的教学，一丝不苟是你们教学工作的最高评价。祝你们，阖家幸福，工作顺利，身体健康。

感谢父母的养育之恩，感谢大学四年同学室友带来的欢乐和共同进步，感谢职场中领导的赏识和同事的认可。

谨以此最后致谢百忙中抽出时间对我的论文进行审查的各位老师，恳请各位老师对我的论文多加匹配指正，使我及时完善本文的不足之处。

参考文献

- [1] 徐洪霞. 网站设计必学的 66 个实例[M]. 上海: 上海科学技术出版社, 2008.
- [2] 苏建波. 计算机专业毕业设计题目大全[Z]. 博客, 2014
- [3] Kenneth, C, Laudon. MANAGEMENT INFORMATION SYSTEMS (Sixth Edition) [M]. 北京: 高等教育出版社, 2001.
- [4]. Mcleod, Management Information System [M], 2010.
- [5] 杨晓钟, 孙振萍. 网站设计与开发[M]. 机械工业出版社, 2005.
- [6] 谢鹏, 韩正之. 基于 B/S 结构的系统设计[M]. 南京: 计算机应用研究, 1999.
- [7]. Lindsay Turner. Tourism demand modelling and forecasting [M]. Lindsay Turner, 2001.
- [8] 王珊, 萨师炫. 数据库系统概论 (第五版) [M]. 北京: 高等教育出版社, 2009.
- [9] 孙卫琴. java 面向对象编程[M]. 北京: 电子工业出版社, 2012.
- [10] Chef Arnold; Arnold; Arnold, Glen. Value-based Management [M]. 北京: 电子工业出版社, 2000.
- [11] H, M, Deitel. Small Java How to Program Sixth Edition[M]. 北京: 电子工业出版社, 2005.
- [12] 黄健宏. Redis 设计与实现[M]. 机械工业出版社, 2014-6
- [13] 阮一峰. JSON Web Token 入门教程[Z]. 博客, 2018
- [14] bcombettter. Java 通过 BCrypt 加密[Z]. 博客, 2018
- [15] 葛一鸣, 郭超. Java 高并发程序设计[M]. 北京: 电子工业出版社, 2008

附 录

附录 1 系统使用说明书

(1) 系统安装, 我们需要启动 MySQL8 服务、Redis 服务以及 Nginx 服务用于助系统相关功能运行, 我们在开发环境中将各个服务打包成 jar 包, 在 Windows 环境中使用 CMD 指令, 在 Linux 中使用命令行运行即可, 我们同时需要配置访问 Go-Server 服务, 用于前端页面的服务启动。

(2) 系统使用, 我们根据不同的用户登录获取的不同权限, 进入不同的用户页面, 在遗忘密码时, 我们可以使用邮件系统找回自己的密码。管理员可以对用户更删改查, 对课程信息进行更删改查, 还可以对学生额外的学分进行增删改查; 老师可以对学生的成绩进行更删改查; 学生对选修课程的选定可以进行添加, 对自己的学分和成绩可以进行查看。

附录 2 需求文档

学生学分成绩管理系统的目的是旨在学院管理员、教师和学生三种用户角色之间的交互, 主要的关系对象为课程、成绩、学分三部分。

(1) 需要构建以用户表为核心, 角色表和权限表来划分用户的操作功能, 用来区分在系统当中可获得的权限。

(2) 登陆系统, 不同角色登陆进入不同访问页面。

(3) 高权限的角色允许对低权限用户进行增删改查操作, 如学院管理员可以添加教师、学生到系统的用户中, 教师可以查看学生列表。

(4) 学院管理员可以添加、修改、删除、查询课程信息, 并对的学分设置和分配比例, 以及课程的属性为选修或必修。

(5) 教师在自己课程中可以对学进行移除, 查询学生详细信息, 对学生进行平时分、测试(实验)分和考试分进行填写。

(6) 学生可以查询自己的课程及相关的成绩, 对于选修课, 学生可以自己自由选择, 抢占名额, 同时也会有成绩。

(7) 对于课程, 我们分为选修课程和必修课程, 必修课由学院管理员直接指定, 选修课由学生选定课程和老师。

(8) 必修课程有学院管理制定, 指定教师和学生, 为其制定平时分、实验分和考试分占的比重, 以及这门课程所代表的学分。

(9) 教师根据学生平时表现和上课情况给予学生平时分, 根据课堂测试或课程设计给予测试分, 根据期末测试给予考试分。

学生可以查询选修课列表, 进行抢占选修课, 同时可以查询自己的考试成绩和最终获得的学分

附录3 用户登录代码

```
/**
 * 用户登录
 */
@PostMapping("login")
public ResultVo login(@RequestBody @Validated({Login.class}) UserVo temp)
    throws Exception {
    VoUtils.copyProperties(temp, userVo, "commonName", "password",
        "random", "captcha");
    UserVo persist = userService.findUser(userVo.getCommonName());
    if (persist != null) {
        if (persist.getCaptchaStatus()) {
            if (!StringUtils.isEmpty(userVo.getCaptcha())) {
                return
                    resultVo.getInstance(MsgConstant.CAPTCHA_NOT_EXISTED);
            } if (StringUtils.isEmpty(userVo.getRandom()) &&
                StringUtils.isEmpty(userVo.getCaptcha())) {
                String captcha = CaptchaUtil.getCaptcha(template,
                    userVo.getRandom());
                if (userVo.getCaptcha().equalsIgnoreCase(captcha)) {
                    loginUser(persist);} else
                {resultVo.getInstance(MsgConstant.CAPTCHA_ERROR);
                }} else
                {resultVo.getInstance(MsgConstant.CAPTCHA_ERROR);
                }} else {loginUser(persist);} else {
                resultVo.getInstance(MsgConstant.LOGIN_ERROR);}return
                resultVo;}

/**
 * 验证用户并记录用户状态
 */
private void loginUser(UserVo persist) throws JsonProcessingException {
    if (BCrypt.checkpw(userVo.getPassword(), persist.getPassword())) {
        if (StatusConstant.ENABLED.equals(persist.getStatus())) {
            String code = persist.getCode();
            String jwtToken = jwtUtil.createJWT(code,
                objectMapper.writeValueAsString(new UserBo(persist.getUserId(),
                    persist.getCode(), persist.getUsername(), persist.getName(),
```

```

        persist.getPhone(),
        persist.getEmail(),persist.getGender(),persist.getAvatar()));
        persist.setJwtToken(new                                JwtToken(code,
        TypeConstant.BEARER_TOKEN, jwtToken, jwtUtil.getTtlMillis()));
        // 缓存用户信息
        template.opsForValue().set(code,
        objectMapper.writeValueAsString(persist));
        // 不需要验证码
        persist.setCaptchaStatus(false);
        userService.updateCaptchaStatus(persist);
        //persist.set
        resultVo.getInstance(HttpStatus.OK.toString(), persist);
    } else {
        resultVo.getInstance(MsgConstant.LOGIN_USER_ERROR);
    }
} else {
    // 需要验证码
    persist.setCaptchaStatus(true);
    userService.updateCaptchaStatus(persist);
    resultVo.getInstance(MsgConstant.LOGIN_ERROR);
}
}
}

```

附录 4 获取图形验证码

```

/**
 * 获取验证码
 */
@GetMapping("captcha")
public ResponseEntity<byte[]> captcha(@RequestParam("random") String
    random, HttpEntity<byte[]> requestEntity) {
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.setCacheControl("no-cache,no-store,must-revalidate");
    responseHeaders.setPragma("no-cache");
    responseHeaders.setExpires(0);
    responseHeaders.setContentType(MediaType.IMAGE_JPEG);
    String captcha = RandomStringUtils.random(4, CaptchaUtil.CHARS);
    template.opsForValue().set(random, captcha.toLowerCase(),
        captchaTimeout, TimeUnit.MILLISECONDS);
}

```

```

        byte[] imageByte = CaptchaUtil.createImage(captcha);
        return new ResponseEntity<byte[]>(imageByte, responseHeaders,
            HttpStatus.OK);
    }

```

附录 5 邮箱操作

```

/**
 * 验证码发送到邮箱
 */
@RequestMapping("email/captcha")
public ResultVo emailCaptcha(@RequestBody
    @Validated({EmailCaptcha.class}) UserVo temp) throws
    MessagingException, IOException {
    System.out.println(temp);
    UserVo persist = userService.findUser(temp.getEmail());
    if (persist == null) {
        return resultVo.getInstance(MsgConstant.INVALID_USER);
    }
    String captcha = RandomStringUtils.random(6, CaptchaUtil.NUMBERS);
    template.opsForValue().set(temp.getRandom(), captcha.toLowerCase(),
        captchaTimeout, TimeUnit.MILLISECONDS);
    String subject = messageSource.getMessage("forgot.password.subject", null,
        LocaleContextHolder.getLocale());
    String content = messageSource.getMessage("forgot.password.content",
        new Object[]{captcha, temp.getEmail(), DateFormatUtils.format(new
            Date(), "yyyy-MM-dd")}, LocaleContextHolder.getLocale());
    try {
        ManagedChannel channel =
            SingletonMessageChannel.getManagedChannel();
        MessageServiceGrpc.MessageServiceBlockingStub stub =
            MessageServiceGrpc.newBlockingStub(channel);

        //      HashMap<Object, Object> jsonDataMap = new HashMap<>(5);
        //      jsonDataMap.put("subject", subject);
        //      jsonDataMap.put("content", content);
        //      jsonDataMap.put("recipientEmail", temp.getEmail());
        EmailBo emailBo = new EmailBo();
        emailBo.setType("captcha");
    }
}

```

```

        emailBo.setContent(content);
        emailBo.setSubject(subject);
        emailBo.setToEmail(temp.getEmail());
        //把包装的数据发送给 grpc 服务端
        String jsonData = objectMapper.writeValueAsString(emailBo);
        MessageRequest request =
        MessageRequest.newBuilder().setJsonData(jsonData).build();
        //grpc 服务端响应
        MessageResponse response = stub.captcha(request);
        String responseJsonData = response.getJsonData();
        Boolean resultBool = objectMapper.readValue(responseJsonData,
        Boolean.class);
        if (resultBool) {
            resultVo.getInstance(HttpStatus.OK.toString());
        } else {
            resultVo.getInstance(MsgConstant.SEND_EMAIL_ERROR);
        }
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    return resultVo;
}

/**
 * 验证邮箱验证码
 */
@PostMapping("email/captcha/valid")
public ResultVo emailCaptchaValid(@RequestBody
    @Validated({EmailCaptchaValid.class}) UserVo temp) throws
    JsonProcessingException {
    String captcha = CaptchaUtil.getCaptcha(template, temp.getRandom());
    if (captcha != null && captcha.equals(temp.getCaptcha())) {
        UserVo persist = userService.findUser(temp.getEmail());
        if (persist == null) {
            return resultVo.getInstance(MsgConstant.INVALID_USER);
        }
        String code = persist.getCode();
        String jwtToken = jwtUtil.createJWT(code,

```

```

        objectMapper.writeValueAsString(new UserBo(persist.getUserId(),
        persist.getCode(), persist.getUsername(), persist.getName(),
        persist.getPhone(),
        persist.getEmail(),persist.getGender(),persist.getAvatar()));
        persist.setJwtToken(new JwtToken(code,
        TypeConstant.BEARER_TOKEN, jwtToken, jwtUtil.getTtlMillis()));
        // 缓存用户信息
        template.opsForValue().set(code,
        objectMapper.writeValueAsString(persist));
        resultVo.getInstance(HttpStatus.OK.toString(), persist);
    } else {
        resultVo.getInstance(MsgConstant.CAPTCHA_ERROR);
    }
    return resultVo;
}

```