 未标题-1

**2018届毕业生**

**毕业设计说明书**

**题 目: 综合教务管理系统的设计与实现**

**院系名称： 信息学院 专业班级：**

**学生姓名： 学 号：**

**指导教师： 教师职称：**

2018**年 5 月 28 日**

摘 要

如今，教育体制的不断改革和人均教育水平的不断提升，引起高校学生人数不断增多，学生选课与信息管理也增加了高校工作管理人员的日常工作量。

综合教务管理系统主要采用Java语言,前端页面采用JSP和Boostrap，而数据库采用MySQL，然后使用MyBatis进行封装，而页面之间采用MVC模式和JavaScript等技术。而本文主要介绍需求分析、数据库设计、详细设计和系统测试，本系统可供学生、教师和管理员使用，而系统支持学生和教师选课以及学生信息管理和教师信息管理，同时还支持成绩录入分析、退选、课程信息管理、选课管理和专业信息管理。

关键词： JSP 选课 Mybatis 教务管理

|  |
| --- |
| **Title**  Design and Implementation of a Comprehensive  Educational Management System  **Abstract**  Nowadays, the continuous reform of the education system and the continuous improvement of the per capita education level have led to an increase in the number of students in universities, and student selection and information management have also increased the daily workload of university work managers.  The integrated educational management system mainly uses Java language. The front page adopts JSP and Boostrap. The database adopts MySQL, and then uses MyBatis to encapsulate. The MVC mode and JavaScript are used between pages. This article mainly introduces requirements analysis, database design, detailed design, and system testing. The system can be used by students, teachers, and administrators. The system supports student and teacher selection and student information management and teacher information management. It also supports the analysis of scores, withdrawal, course information management, course management and professional information management.  Keywords JSP Elective Mybatis Teaching Management  目 次  [1 引言 5](#_Toc515813322)  [1．1 选题背景 5](#_Toc515813323)  [1．2 研究现状 5](#_Toc515813324)  [1．3 研究内容 6](#_Toc515813325)  [2 技术支持 7](#_Toc515813326)  [2．1 B/S架构 7](#_Toc515813327)  [2．2 框架 7](#_Toc515813328)  [2．3 Boostrap 7](#_Toc515813329)  [2．4 开发环境 7](#_Toc515813330)  [3 需求分析与设计 8](#_Toc515813331)  [3．1 管理员功能 8](#_Toc515813332)  [3．2 教师功能 12](#_Toc515813333)  [3．3 学生功能 13](#_Toc515813334)  [4 概要设计 15](#_Toc515813335)  [4．1 整体功能 15](#_Toc515813336)  [4．2 管理员功能 15](#_Toc515813337)  [4．3 教师功能 16](#_Toc515813338)  [4．4 学生功能 16](#_Toc515813339)  [5 数据库设计 17](#_Toc515813340)  [5．1 数据库需求分析 17](#_Toc515813341)  [5．2 概念结构设计 17](#_Toc515813342)  [5.2.1 实体设计 17](#_Toc515813343)  [5.2.2 E-R图设计 18](#_Toc515813344)  [5．3 逻辑结构设计 19](#_Toc515813345)  [5.3.1 PDM图设计 19](#_Toc515813346)  [5.3.2 数据库表设计 19](#_Toc515813347)  [6 详细设计 25](#_Toc515813348)  [6．1 系统登陆设计 25](#_Toc515813349)  [6．2 管理员模块设计 26](#_Toc515813350)  [6.2.1 学生管理模块 26](#_Toc515813351)  [6.2.2 教师管理模块 32](#_Toc515813352)  [6.2.3 课程管理模块 33](#_Toc515813353)  [6.2.4 选课管理模块 35](#_Toc515813354)  [6.2.5 专业管理模块 37](#_Toc515813355)  [6.2.6 时间设置模块 37](#_Toc515813356)  [6．3 教师模块设计 38](#_Toc515813357)  [6.3.1 选课模块 38](#_Toc515813358)  [6.3.2 当前课表 38](#_Toc515813359)  [6.3.3 成绩管理模块 39](#_Toc515813360)  [6.3.4 成绩分析模块 39](#_Toc515813361)  [6．4 学生模块设计 39](#_Toc515813362)  [6.4.1 选课模块 39](#_Toc515813363)  [6.4.2 成绩模块 40](#_Toc515813364)  [6.4.3 课表模块 40](#_Toc515813365)  [7 系统测试 41](#_Toc515813366)  [7．1 管理员子功能测试 41](#_Toc515813367)  [7．2 教师子功能测试 42](#_Toc515813368)  [7．3 学生子功能测试 44](#_Toc515813369)  [结 论 46](#_Toc515813370)  [致 谢 47](#_Toc515813371)  [参 考 文 献 48](#_Toc515813372) |

# 1 引言

## 1．1 选题背景

随着计算机的飞速发展和我国教育体制的不断进步，社会人均教育水平的不断提升，导致人们对新的生活提出了新的需求方式，同时高校学生的人数也在不断增多。计算机的发展让人们无时无刻都离不开计算机。一方面，传统人工方式操作难度复杂，数据量过大，同时容错率也非常高，给高校工作人员增加了负担，比如管理教师信息，学生信息，课程信息，选课等，也让高校为此付出一定的资金，所以传统人工操作方式已不能满足现如今的需求。故我们需要考虑如何更好的合理、管理这些信息数据，同时高校改革导致学生兴趣多样化，使得选课也成为一个更为艰难的工作，那如何使得选课变得有序不乱呢？另一方面，解决了这些问题为学校、教师和学生三者之间的联系提供了新的发展方向，不仅能大大降低信息管理的错误率，也能为学生和教师选课增加便利，使得学生与老师随时随地操作系统，了解更多有关自己的信息，大大提升了学校管理人员工作的效率。

针对于这些方面的问题，迫切需要研究一个综合教务管理系统，而这些问题正是我们所需要考虑和解决的，也是综合教务管理系统所重点研究的，正是因为如此，综合教务管理系统逐渐成为了高校工作管理的核心。

## 1．2 研究现状

现如今，综合教务管理已经成为高校日常管理不可或缺的部分，该系统相对于国内早期也发展较为成熟，而在市场经济发展中，在国内，由于起步研发综合教务管理系统比较晚的原因，使得我国与国外在这方面存在着比较大的差距，因此研发综合教务管理系统的公司有许许多多，各个公司研发的综合教务管理系统针对于不同的实际情况也各不相同，教育体制的改革，影响着综合教务管理系统往不同的方向发展，也增加了综合教务管理系统中功能的多样化，也正是因为教育体制的改革，使得系统需要有更好的接口进行二次开发；在国外，高校有专门的团队进行专门研究综合教务管理系统，并提供有效的服务和技术支持。针对于该系统的研发，不仅需要研发软件，同时也需要购买一定的硬件设备，需要耗费一定费用。

在国内的研究现状来看，这种系统在一些方面存在着不足，比如安全方面和信息处理方面等，各个现有系统资料单独建立，共享性差，目前我国教育正处于不断改革、创新的阶段，我国需要不断探索适合我国教育的教学形式，不断对系统进行完善和优化。

## 1．3 研究内容

综合教务管理系统的设计与实现的开展，首先需要查阅资料，对该系统有一个初步的了解，给实际功能指导方向，然后开始进行需求分析，只有合理的需求分析才能在短时间开发出更好的系统。故本文围绕综合教务管理流程展开分析，使用B/S结构展开设计，然后通过软件测试来对系统进行优化和纠正。对此，本系统根据不同身份对功能进行划分，不同身份所担任的职责也不相同，职责不同所开发的功能也不相同，而本系统的身份有学生、教师和管理员，管理员持有最高权限。管理员可以管理学生、课程、成绩、选课、教师等的信息。选课是本系统中必不可少的环节。而选课涉及到学生、课程以及教师。而课表信息能让学生了解每一学期的课程，教师也一样。而成绩分析可以让老师更加客观的了解学生的考试情况。对于正式选课是本学期选择下学期的选修课，而重修选课则是选择本学期的必修课，二者是完全不同的。针对于删除操作，由于本系统采用Mybatis框架，根据Mysql语句可以建立表与表之间的关系，使用数据库的多级联动操作来删除相关联的数据。

# 2 技术支持

## 2．1 B/S架构

目前最主要的两种架构是C/S架构和B/S架构，而本系统采用的是B/S架构，B/S是Browser/Server的缩写。Browser指的是浏览器， Server为服务器，而浏览器进行数据的交互需要使用服务器和数据库。B/S架构维护和升级比较简单，成本低，选择性多样化。

## 2．2 框架

Mybatis是目前软件开发中非常流行的数据库框架。MyBatis并不完全是一种ORM框架，它的设计思想和ORM相似，它允许开发人员直接编写SQL语句，使得访问数据库更加灵活[1]。MyBatis 采用配置文件和映射实现java语言和SQL语句之间的连接，将映射的接口和配置文件中的数据库语句一一对应起来。该框架简单易学，没有任何第三方的依赖，通过网上自我学习或书本学习，可以很快掌握Mybatis框架的思想，同时在本系统中运用多级联动操作，关联起表表之间的联系，同时，在进行数据操作时，可以将代码和数据库语言进行分离，每次只需要某个操作时，只需在配置文件中写SQL语句，然后写出相对应的接口，SQL语句与接口形成了映射，最后调用该接口即可。

## 2．3 Boostrap

Bootstrap是目前全球最流行、最火爆的Web前端开发框架之一。它的强大之处在于它将常见的CSS布局小组件和JavaScript插件进行了完整并完善的封装[2]。该前端框架持有多种多样的格式供用户使用，同时针对于排版该框架有栅格系统，栅格系统是将系统分成12列，然后根据不同的划分可以进行排版，Bootstrap含有多种多样的组件，而针对不同组件的使用，可以参考Boostrap官网上的知识进行自学，然后进行重用修改就可以完成，Boostrap还结合了JQuery，可以对其中的组件进行重叠使用。

## 2．4 开发环境

开发语言：java

数据库：MySQL

工具：IntelliJ IDEA , Tomcat, bootstrap, JavaScript, ajax

# 3 需求分析与设计

综合教务管理系统致力于优化高校工作人员的工作方式以及教师和学生的日常学习生活，方便系统管理人员工作，使学校的日常教务管理更加方便高效，而多个功能的交叉增加了系统的复杂程度，故需要开发一套综合教务管理系统来解决这些问题，因而该系统将完成以下任务。

## 3．1 管理员功能

管理员功能矢量图如下图3.1所示。管理员可以对学生的信息进行增加、查询和修改，同时在删除学生信息时需要删除学生的所有信息，包括重修成绩，选课记录，正成绩，并且管理员可以对学生成绩进行模糊查询，还可以浏览学生信息，比如分班级，入学年，学院等。在对学生信息进行添加时，需要严格判断身份证号，包括出生年、月、日，身份证长度，还需要判断手机号长度，同时判断身份证号和手机号是否全为数字，而学生学号根据入学年、学院编号、专业编号和班级里序号进行确定。添加学生信息流程图如下图3.2所示。而对于查询学生信息，其实存在多种查询方式，有学号查询、已有的专业班级进行查询、入学年查询和模糊查询，而管理员可以选择其中的方式，然后输入相应的信息进行查询，查询学生信息流程图如下图3.3所示。

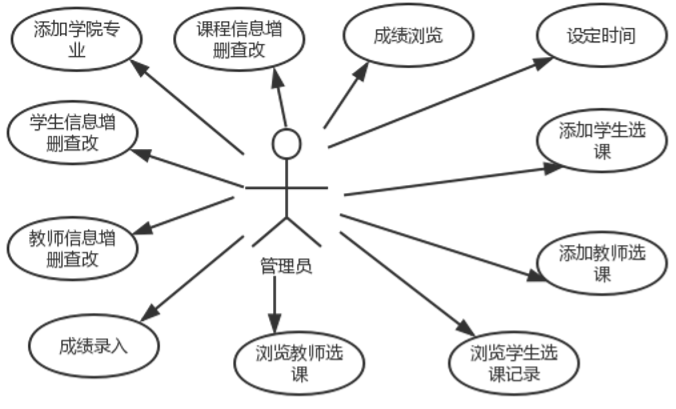


图3.1 管理员功能矢量图

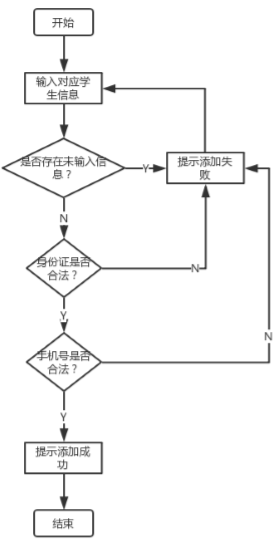
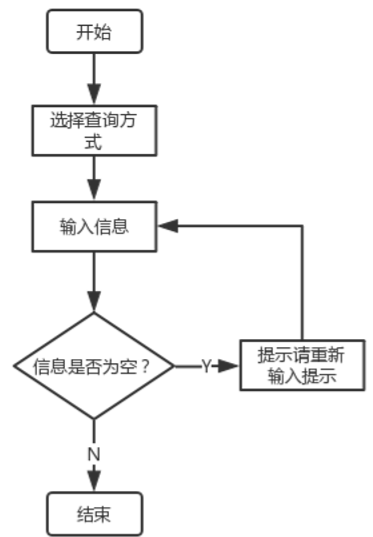
 

图3.2 添加学生流程图 图3.3 查询学生信息流程图

管理员可以对教师的信息进行增加、查询和修改，针对于教师信息的修改，需要对修改教师身份证进行验证，同时也要验证修改后对应信息是否为空，修改教师信息流程图如下图3.4所示。对于在删除教师信息时需要删除教师的全部信息，包括教师选课信息，教师信息，还有选该教师授课的学生的选课信息，包括重修成绩和正成绩，检测教师工号中是否含有非数字字符。

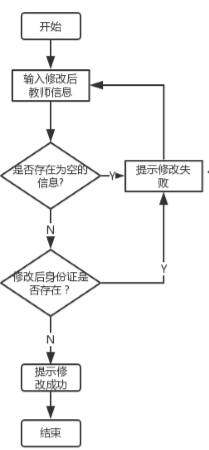


图3.4 修改教师信息流程图

管理员可以对课程的信息进行增加、查询和修改，同时在删除课程信息时采用框架中的多级联动操作，需要删除课程的全部信息，当课程信息不存在时，相对应学生的成绩也就不存在了，以此类推，教师选这课程的信息也需要删除了。

管理员根据入学年、学院和专业进行浏览成绩，并录入成绩，成绩包括平时成绩、考试成绩以及补考成绩和重修成绩。而成绩的范围是在0~100之间，若不在这个范围内，则录入失败，反之则录入成功。录入成绩流程图如下图3.5所示。

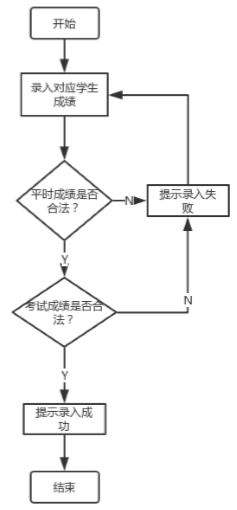


图3.5 录入成绩流程图

管理员根据授课年份和学院进行浏览教师选课信息，并可以添加和修改教师选课信息，在删除教师选课信息时，也要对应删除选该课程的学生成绩(包括重修生的成绩)。而对于学生选课记录，则根据学生入学年和学院进行浏览，可添加学生选课信息，包括重修选课和正修选课两种方式。

管理员可以添加学院信息和专业信息，并且每个学院存在对应的编号，而这每一个学院编号都是唯一的，专业也一样，学生的学号通过这些编号进行形成，专业只能在已有的学院中产生。还可以设置时间进行学生选课时间、初修成绩录入、重修成绩录入、教师选课时间和补考成绩录入时间，这些时间限定了选课的时间。添加学院流程图如下图3.6所示，添加专业流程图如下图3.7所示。

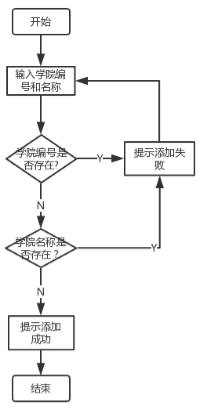
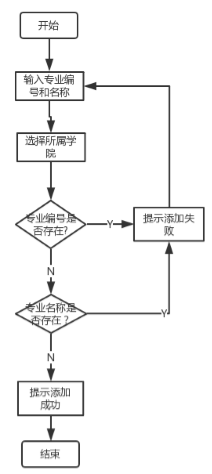
 

图3.6 添加学院流程图 图3.7 添加专业流程图

## 3．2 教师功能

教师功能矢量图如下图3.8所示。教师可以选择所授课程，同时在一定时间内可以退选，同时也可以查看本学期课表。教师可以选择数据库中已经录入的课程，形成教师课程信息，供学生选课，教师也可以根据自身情况退选课程，同时教师可以查看已选择的课程。教师通过浏览授课学生，对相应的学生录入成绩，通过简单分析，生成相应的图形。教师选课流程图如下图3.9所示

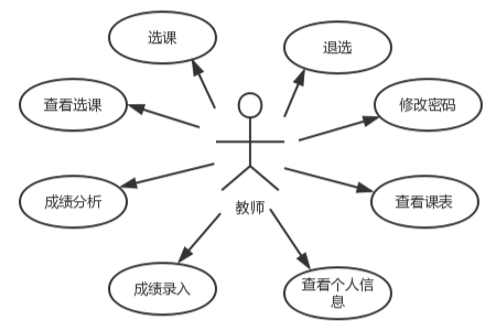


图3.8 教师功能矢量图

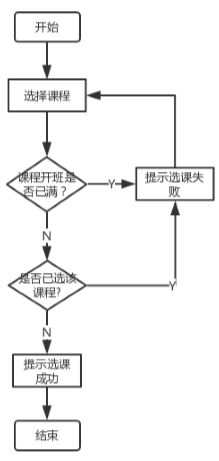


图3.9 教师选课流程图

教师可以查看个人信息，并修改密码；还可以查看授课学生信息，并且可以录入学生成绩(正修成绩、补考成绩和重修成绩)，并进行成绩分析。教师还可以查看历年教授课程的成绩分析情况。

## 3．3 学生功能

学生功能矢量图如下图3.10所示。学生可以查看个人信息以及修改密码，学生在规定时间内进行正修选课和正退选以及重修选课和重修退选。学生可以选择选修课，而必修课则是通过教师选课自动默认选择，而退选，则通过查看已选课程信息进行退选；学生在第一二学期不可以进行选择，从第三学期开始可以选择第一学期的课程重修，而退选，也是需要先浏览已选课程信息进行退选；学生可以查看以往学期的课程以及本学期课程。学生选课流程图如下图3.11所示。

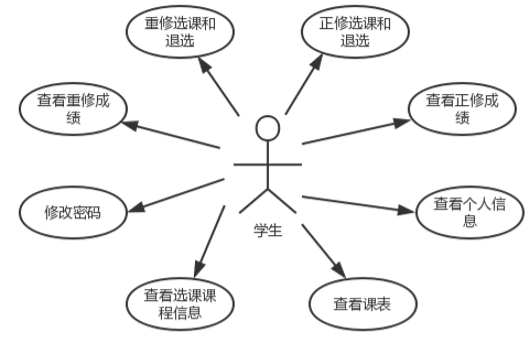


图3.10 学生功能矢量图

对于学生选择选修课，学生可能已经选了该课程，也有可能没有选择课程，若选了则选课失败，反之，本系统设置一个班级为120人，若选择该课程的学生人数已满，则选课失败，反之，则选课成功。

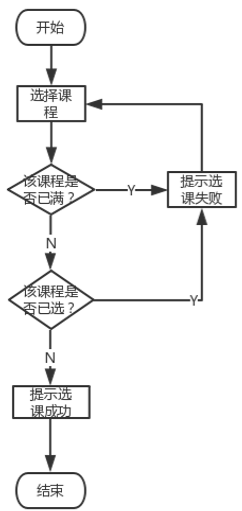


图3.11 学生选课流程图

# 4 概要设计

## 4．1 整体功能

系统的功能图如下图4.1所示。

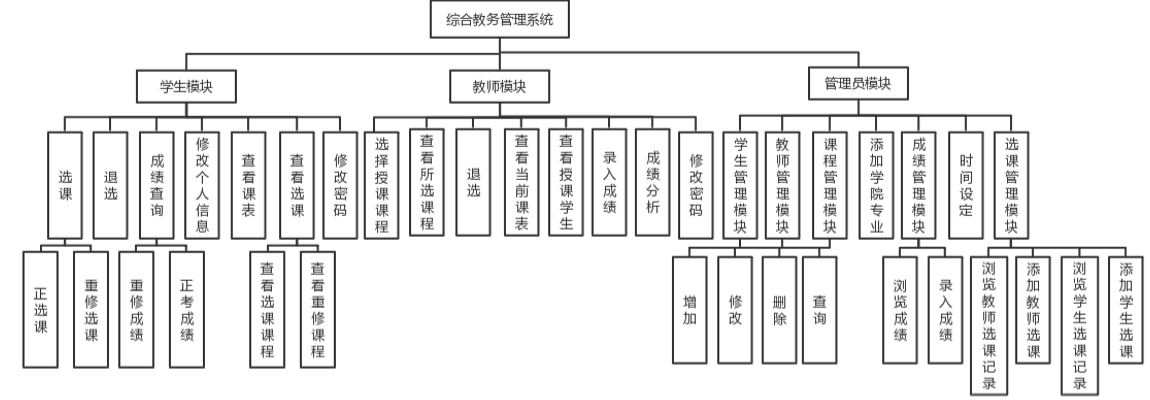


图4.1 系统功能图

## 4．2 管理员功能

管理员的功能如下图4.2所示。

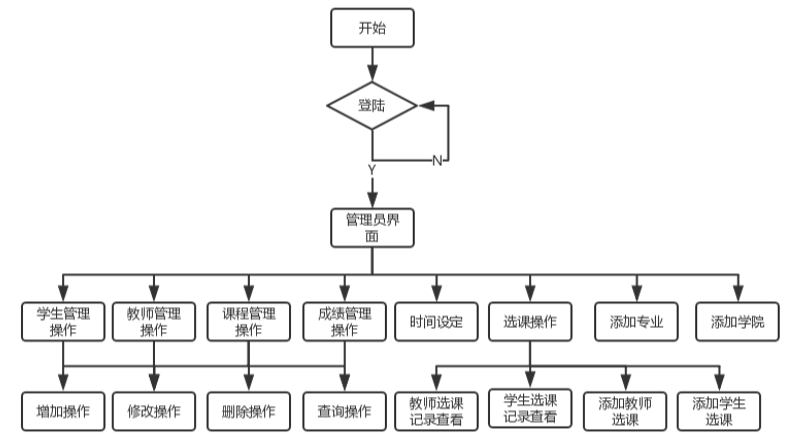


图4.2 管理员功能流程图

## 4．3 教师功能

教师的功能如下图4.3所示。

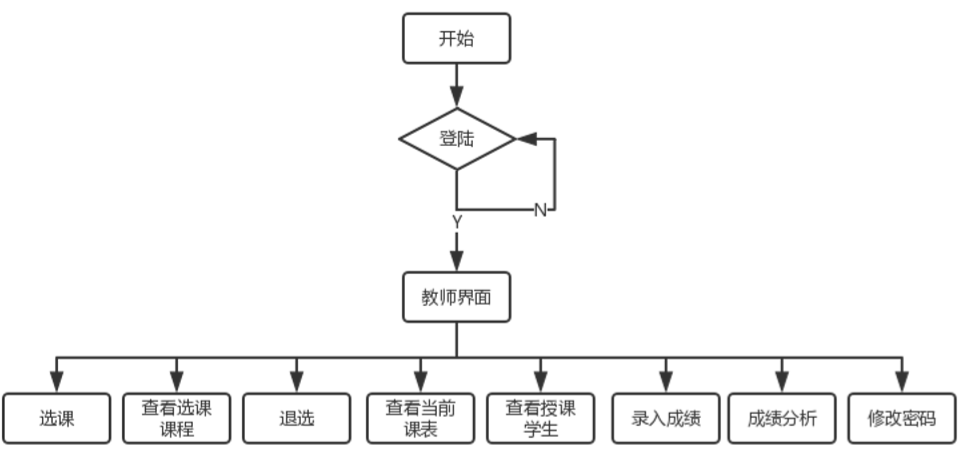


图4.3 教师功能流程图

## 4．4 学生功能

学生的功能如下图4.4所示。

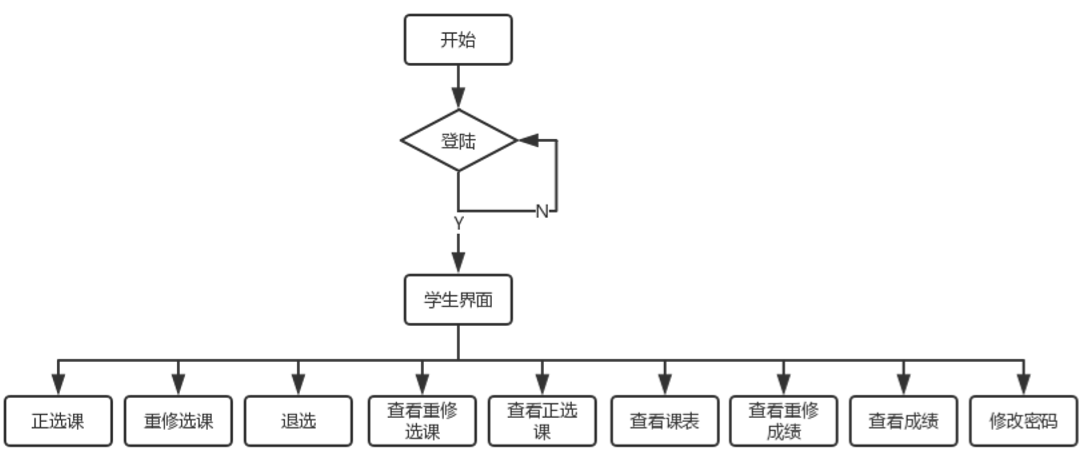


图4.4 学生功能流程图

# 5 数据库设计

## 5．1 数据库需求分析

设计数据库是设计系统不可或缺的重要环节，如果初始数据库设计得不合理，这不仅增加了工作量，而且也会增加一定的费用，还耽误时间，所以数据库的设计尤为重要。而综合教务管理系统的实际应用对象为学生、教师、学校(教务处)。根据实际应用对象，不同的用户所对应的需求也各不相同。对于学生而言，主要查看自己的全部信息以及修改密码，可以选课以及查看所选课程，查看课表和考试成绩；对于教师而言，主要选择所要教授的课程，浏览选课学生的信息以及授课课表，还有成绩录入以及分析；对于管理员而言，主要对教师、学生、课程信息等进行管理，学生成绩的录入以及教师和学生的选课记录，还有增加教师和学生选课等功能。

## 5．2 概念结构设计

### 5.2.1 实体设计

(1) 学生实体如下图5.1所示，学生学号为主键。

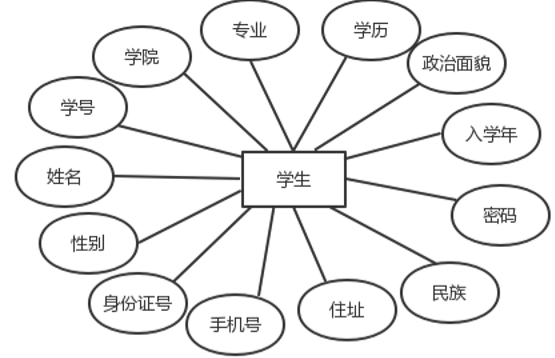
 

图5.1 学生实体 图5.2 教师实体

(2) 教师实体如上图5.2所示，教工号为唯一主键。

(3) 课程实体如下图5.3所示，课程编号为唯一主键。



图5.3 课程实体

(4) 专业实体如下图5.4所示，专业编号为唯一主键。

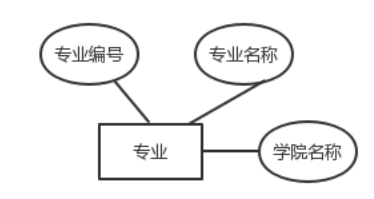
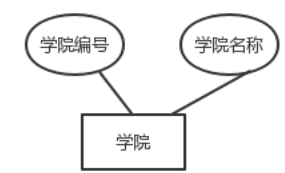
 

图5.4 专业实体 图5.5 学院实体

(5) 学院实体如上图5.5所示，学院编号为唯一主键。

### 5.2.2 E-R图设计

系统全局E-R图如下图5.6所示，一个学院可以存在多个专业，一个专业对应多名学生；一个学院可以存在多名教师，一名教师可以选择多门课程；一名学生可以选择多门课程，产生多条成绩或补考成绩。

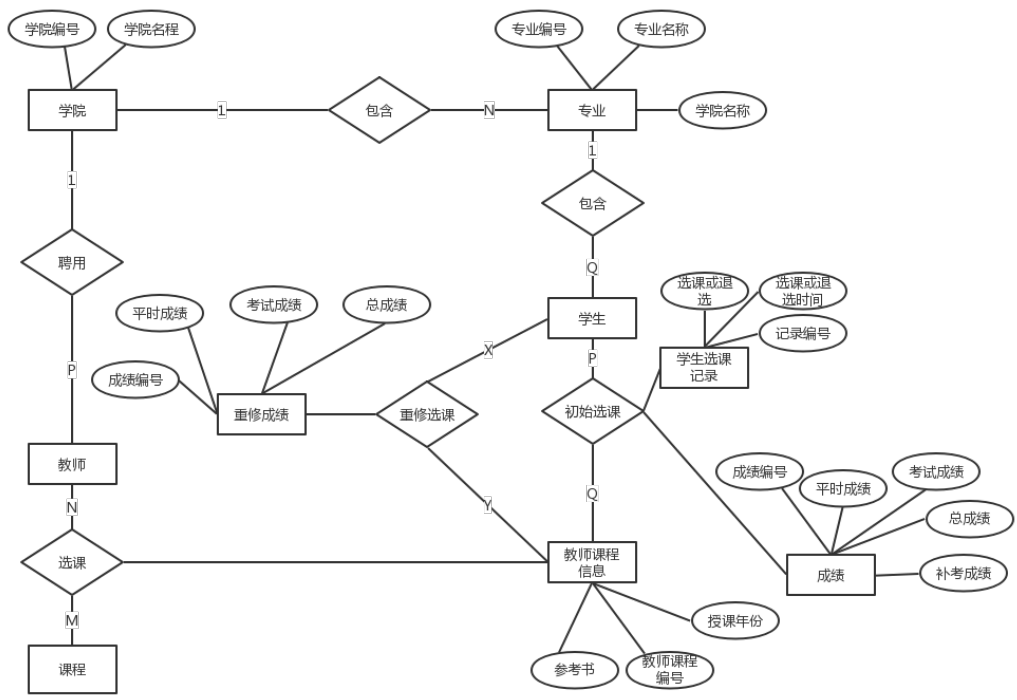


图5.6 系统全局E-R图

## 5．3 逻辑结构设计

### 5.3.1 PDM图设计

根据上面概念结构的设计，综合教务管理系统的数据需要通过数据库进行存储，本系统采用MySQL 数据库进行设计。故本系统将创建manager数据库，主要用来存储本系统中所有的数据信息。而本数据库中将包含student表(学生信息表，如下图5.7)，teacher表(教师信息表，如下图5.8)，course表(课程信息表，如下图5.9),discipline表(专业信息表，如下图5.10)，institute表(学院信息表，如下图5.11)，record表(学生选课记录表，如下图5.12),courseteacher表(教师选课信息表，如下图5.13)，grade表(学生成绩表，如下图5.14),makeup表(学生重修成绩表，如下图5.15),times表(时间设定表，如下图5.16)。

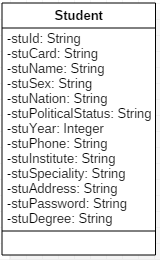
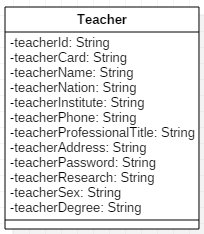
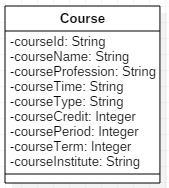
  

图5.7 学生表 图5.8 教师表 图5.9 课程表

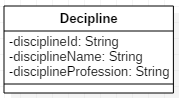
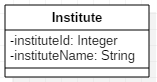
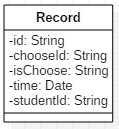
  

图5.10 专业表 图5.11 学院表 图5.12 选课记录表

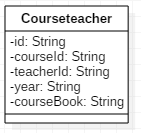
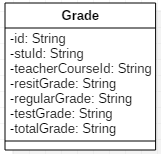
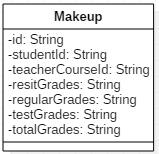
   

图5.13 教师选课表 图5.14 成绩表 图5.15 重修成绩表 图5.16 时间设定表

### 5.3.2 数据库表设计

每一个实体在数据库中都对应一张二维表，该系统中存在着学生信息、教师信息和课程信息，还有学院和专业信息等。对应于选课时间以及录入成绩时间等要求，故需要建立表times；针对于教师选课后需要记录数据，故需要建立表courseteacher；针对于学生选课和记录平时成绩、考试成绩、补考成绩和重修成绩，则需要建立grade表、makeup重修表、学生选课记录表record；同时也需要建立学院表和专业表。

表与表之间没有明显的联系，而在MyBatis中所有的SQL语句与接口都是一一对应的，当需要使用某一条SQL语句时只需要调用其对应的接口，针对于有些相关联的操作，比如删除教师信息操作，这都需要利用多级联动操作，在本系统中，一张表的主键作为下一张表的属性，然后对应进行操作。

(1) 学生信息表

学生信息表用于管理学生个人信息，方便学生选课记录和成绩的产生，同时也用于记录学生的基本信息，包括的信息有学生学号、学生姓名、身份证号等信息。而在该表中，学生学号作为该表的主键。该表中具体信息如下表5-1所示。

表5-1 学生表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 |
| stuId | varchar | 15 | 否 | 是 | 学生学号 |
| stuCard | varchar | 18 | 否 | 否 | 身份证 |
| stuName | varchar | 20 | 否 | 否 | 学生姓名 |
| stuSex | varchar | 4 | 否 | 否 | 学生性别 |
| stuNation | varchar | 10 | 否 | 否 | 民族 |
| stuPoliticalStuatus | varchar | 255 | 否 | 否 | 政治面貌 |
| stuYear | int | 4 | 否 | 否 | 入学年份 |
| stuPhone | varchar | 11 | 否 | 否 | 手机号 |
| stuInstitute | varchar | 30 | 否 | 否 | 所属学院 |
| stuSpeciality | varchar | 40 | 否 | 否 | 所属专业 |
| stuAddress | varchar | 255 | 否 | 否 | 家庭住址 |
| stuDegree | varchar | 10 | 否 | 否 | 学历 |
| stuPassword | varchar | 20 | 否 | 否 | 登陆密码 |

(2) 教师信息表

教师信息表可以用于管理员对教师信息进行管理，方便学生选课和教师授课选课情况，同时主要用于对教师个人信息进行记录。该表包含有教师工号、教师姓名、教师性别、民族、身份证号、所属学院、研究方向、手机号码、学历、政治面貌、职称和登陆密码。其中教师工号作为主键。教师信息和学生信息通过选课进行联系。该表具体信息如下表5-2所示。

表5-2 教师表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | | 主键 | 字段含义 |
| teacherId | varchar | 10 | | 否 | 是 | 教师工号 |
| teacherCard | varchar | 18 | | 否 | 否 | 身份证 |
| teacherName | varchar | 20 | | 否 | 否 | 教师姓名 |
| teacherSex | varchar | 4 | | 否 | 否 | 教师性别 |
| teacherNation | varchar | 10 | | 否 | 否 | 民族 |
| teacherResearch | varchar | 255 | | 否 | 否 | 研究方向 |
| teacherProfessionalTitle | varchar | 30 | | 否 | 否 | 职称 |
| teacherPhone | varchar | 11 | | 否 | 否 | 手机号 |
| teacherInstitute | varchar | 30 | | 否 | 否 | 所属学院 |
| teacherAddress | varchar | 255 | | 否 | 否 | 家庭住址 |
| teacherDegree | varchar | 10 | | 否 | 否 | 学历 |
| teacherPassword | varchar | 20 | | 否 | 否 | 登陆密码 |

(3) 课程信息表

课程信息表可用于教师选课，从而让学生选择所要修的课程；对于成绩，也需要课程信息来进行设置，比如成绩表中必然含有教师选课的信息，而成绩表中也含有教师选课信息。课程信息包括课程编号、课程名称、课时、课程类型、课程开设学期、专业要求、学分、上课时间备注和开设学院，而其中课程编号作为本表的唯一主键。该表具体信息如下表5-3所示。

表5-3 课程表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 |
| courseId | varchar | 10 | 否 | 是 | 课程编号 |
| courseName | varchar | 30 | 否 | 否 | 课程名称 |
| courseProfession | varchar | 255 | 否 | 否 | 专业要求 |
| courseCredit | int | 11 | 否 | 否 | 学分 |
| courseTerm | int | 11 | 否 | 否 | 授课学期 |
| coursePeriod | int | 11 | 否 | 否 | 课时 |
| courseInstitute | varchar | 25 | 否 | 否 | 职称 |
| courseType | varchar | 11 | 否 | 否 | 课程类型 |
| courseTime | varchar | 255 | 否 | 否 | 授课时间 |

(4) 教师课程表

每一位老师都可以选择所要授课的课程，这是学生选课的前提。一名老师可以选择多个课程，但是一个课程也有可能会有多个老师来授课，这要根据该专业的学生人数来决定。教师课程表中包含有编号、课程编号、教师编号、授课年份、参考书，而该表中的编号为唯一主键，该表具体信息如下表5-4所示。

表5-4 教师课程表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 | 备注 |
| id | varchar | 10 | 否 | 是 | 选课编号 |  |
| courseId | varchar | 10 | 否 | 否 | 课程编号 | course表的主键 |
| teacherId | varchar | 10 | 否 | 否 | 教师编号 | teacher表的主键 |
| year | varchar | 255 | 否 | 否 | 授课年份 |  |
| courseBook | varchar | 255 | 否 | 否 | 参考书 |  |

(5) 成绩表和重修成绩表

每一位学生可以选择多位老师所授的不同课程，而最后针对的结果必定是每一科都存在一个分数，需要对学生的选课信息和成绩进行管理，故需要建立一张grade表。教师可以通过该表可以查看授课学生的部分信息。成绩表包含编号、学生学号、教师选课编号、平时成绩、考试成绩、总成绩、补考成绩，而该表中编号作为唯一主键，该表中的各个字段的具体信息如下表5-5所示。而针对于重修表，重修学生不存在补考成绩，故其余的如同于成绩表，重修成绩表的主键为编号，重修成绩表包含有编号、学生学号、教师选课编号、平时成绩、考试成绩、总成绩，该表具体信息如下表5-6所示。

表5-5 成绩表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 | 备注 |
| id | varchar | 10 | 否 | 是 | 成绩编号 |  |
| stuId | varchar | 13 | 否 | 否 | 学生学号 | 表student主键 |
| teachercourseId | varchar | 10 | 否 | 否 | 教师编号 | 表teacher主键 |
| regularGrade | varchar | 11 | 是 | 否 | 平时成绩 |  |
| testGrade | varchar | 11 | 是 | 否 | 考试成绩 |  |
| totalGrade | varchar | 11 | 是 | 否 | 总成绩 |  |
| resitGrade | varchar | 11 | 是 | 否 | 补考成绩 |  |

表5-6 重修成绩表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 | 备注 |
| id | varchar | 10 | 否 | 是 | 成绩编号 |  |
| stuId | varchar | 13 | 否 | 否 | 学生学号 | 表student主键 |
| teachercourseId | varchar | 10 | 否 | 否 | 选课编号 | 表courseteacher主键 |
| regularGrades | varchar | 11 | 是 | 否 | 平时成绩 |  |
| testGrades | varchar | 11 | 是 | 否 | 考试成绩 |  |
| totalGrades | varchar | 11 | 是 | 否 | 总成绩 |  |

(6) 学生选课记录表

每一位学生选课都是存在于记录，管理员能够查看这些记录，每一位学生可以选课也可以退选。学生选课记录表包含记录编号、课程编号、选或退、选课时间以及学生学号。而在本表中记录编号为唯一主键，该表具体信息如下表5-7所示。

表5-7 学生选课记录表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 | 备注 |
| id | varchar | 10 | 否 | 是 | 记录编号 |  |
| chooseId | varchar | 10 | 否 | 否 | 课程编号 | 表course主键 |
| isChoose | varchar | 10 | 否 | 否 | 选或退 |  |
| time | detetime | 0 | 否 | 否 | 选课时间 |  |
| studentId | varchar | 15 | 否 | 否 | 学生学号 | 表student主键 |

(7) 时间设定表

在综合教务管理系统中，学生选课、教师选课、录入成绩以及重修选课并不是时时刻刻都能选的，必须在一定的时间范围内进行操作，而该表中存在五个编号，每一个编号都存在着一定的含义。该表中含有编号、起始时间、终止时间。

该表具体信息如下表5-8所示。

表5-8 时间设定表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 | 备注 |
| id | int | 11 | 否 | 是 | 编号 |  |
| start | datetime | 0 | 否 | 否 | 起始时间 |  |
| end | datetime | 0 | 否 | 否 | 终止时间 |  |

(8) 学院表和专业表

一个学校存在多个学院，同时一个学院也存在多个专业。学院表包含学院编号和学院名称。而专业表包含专业编号、专业名称和学院名称。学院表的具体字段如下表5-9所示，学院编号作为学院表的唯一主键；专业表的具体字段如下表5-10所示，专业编号作为专业表的唯一主键。

表5-9 学院表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 | 备注 |
| instituteId | int | 3 | 否 | 是 | 学院编号 |  |
| instituteName | varchar | 255 | 否 | 否 | 学院名称 |  |

表5-10 专业表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 主键 | 字段含义 | 备注 |
| disciplineId | int | 2 | 否 | 是 | 专业编号 |  |
| disciplineName | varchar | 255 | 否 | 否 | 专业名称 |  |
| disciplineProfession | varchar | 255 | 否 | 否 | 学院名称 |  |

# 6 详细设计

在开始设计综合教务管理系统之前，需要先进行需求分析，然后再进行详细设计；而设计该系统，涉及到三种身份，故需要对身份进行验证。本系统主要是实现用户登陆界面、管理员模块、学生模块和教师模块等的设计、测试。

## 6．1 系统登陆设计

综合教务管理系统涉及到不同用户的功能使用，导致需要实现不同功能的隔离，而系统的登陆界面只是为了验证用户的身份，所以设计简单明了。通过登陆，可以进入到管理员主页面、教师主页面、学生主页面，然后进行不同身份的不同操作。系统登陆界面如下图6.1所示。



图6.1 系统登陆界面

用户可以选择不同身份进行验证，但是需要选择登陆身份，并输入工号和密码，然后进行数据库验证。当身份为管理员时，工号和密码验证通过后进入管理员主界面(如下图6.2所示)；当身份为教师时，验证通过后进入教师主界面(如下图6.3所示)；当身份为学生时，验证通过后进入学生主界面(如下图6.4所示)。



图6.2 管理员登陆后主界面

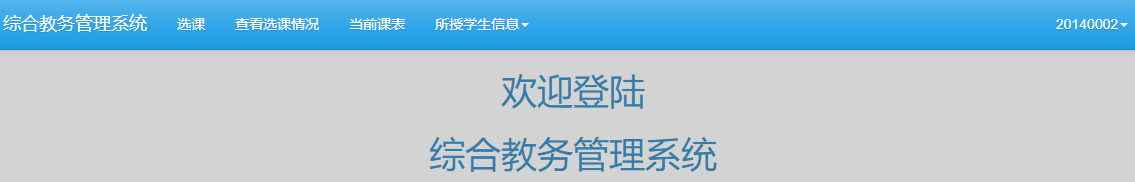


图6.3 教师登陆后主界面



图6.4 学生登陆后主界面

|  |
| --- |
| <select name="choose" class="name" id="choose">  <option value="choose">--请选择身份--</option>  <option value="student">学生</option>  <option value="teacher">教师</option>  <option value="admin">管理员</option> </select>  <input type="text" placeholder="输入工号" class="name\_inp" name="login\_id">  <input type="password" placeholder="输入密码" class="name\_inp" name="login\_password"> |

## 6．2 管理员模块设计

### 6.2.1 学生管理模块

在本系统中，管理员可以对学生信息进行相应管理。对于添加信息而言，管理员对于多个学生可以采用导入的方式进行添加；而对于单个学生，首先是管理员如实对学生信息进行相应输入，若暂无学院信息和专业信息，则需要先添加学院信息和专业信息，然后对于相应的学生信息，需要做相应的处理和限制。添加学生信息如下图6.5所示。上传批量学生信息如下图6.6所示。对于查询信息，管理员选择查询方式，然后进行查询，查询学生信息如下图6.7所示。对于修改和删除，管理员需要对学生信息进行筛选浏览，根据浏览类型和浏览信息以及学生入学年来进行筛选(筛选学生信息如下图6.8)，筛选完后通过相对应的链接进行学生信息操作,修改学生信息如下图6.9所示，删除学生信息如下图6.10所示。



图6.5 添加学生信息

|  |
| --- |
| ServiceMapper serviceMapper = new ServiceMapper(); Institute institute = serviceMapper. getInstituteMapper(). selectByName(add\_studentInstitute);//获取学院 Discipline discipline = serviceMapper.getDisciplineMapper(). selectByName(add\_studentSpeciality); //获取学院  String student\_id = id + string\_class + person\_num; student.setStuSpeciality(add\_studentSpeciality + add\_studentYear.substring(2, 4) + string\_class); serviceMapper.getStudentMapper().insert(student); serviceMapper.close(); |

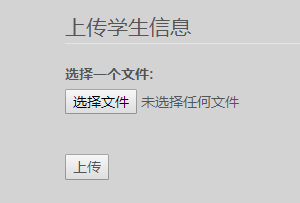


图6.6 上传批量学生信息

|  |
| --- |
| //上传批量学生信息  String fileName = new File(item.getName()).getName(); File storeFile = new File(filePath);  String[] strings = string.split(";"); //按行切割数据参数 for (Institute institute : institutes) {  if (institute.getInstituteName().equalsIgnoreCase(strings[8])) {  institute1 = 1;  institute2 = institute;  break;  } } |

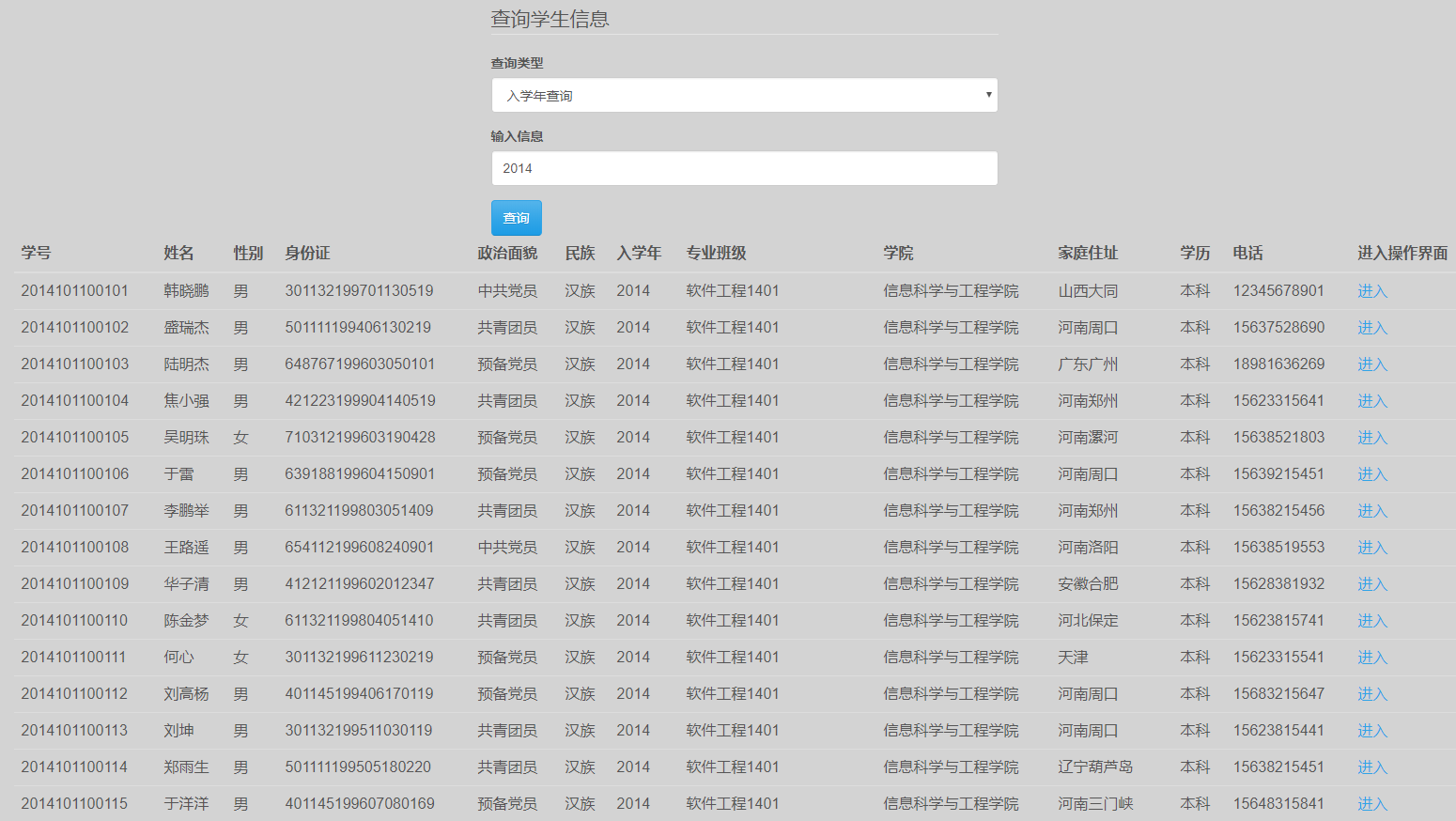


图6.7 查询学生信息

|  |
| --- |
| String type = request.getParameter("type"); String selectStudentInformation = request.getParameter("selectStudentInformation");  list = serviceMapper.getStudentMapper().selectById(selectStudentInformation); //学号查询  list = serviceMapper.getStudentMapper().selectByClass(selectStudentInformation);//专业班级查询  list = serviceMapper.getStudentMapper().selectByYear(Integer.*parseInt*(selectStudentInformation));//入学年查询  list = serviceMapper.getStudentMapper().selectByMatches("%" + selectStudentInformation + "%");//模糊查询 |

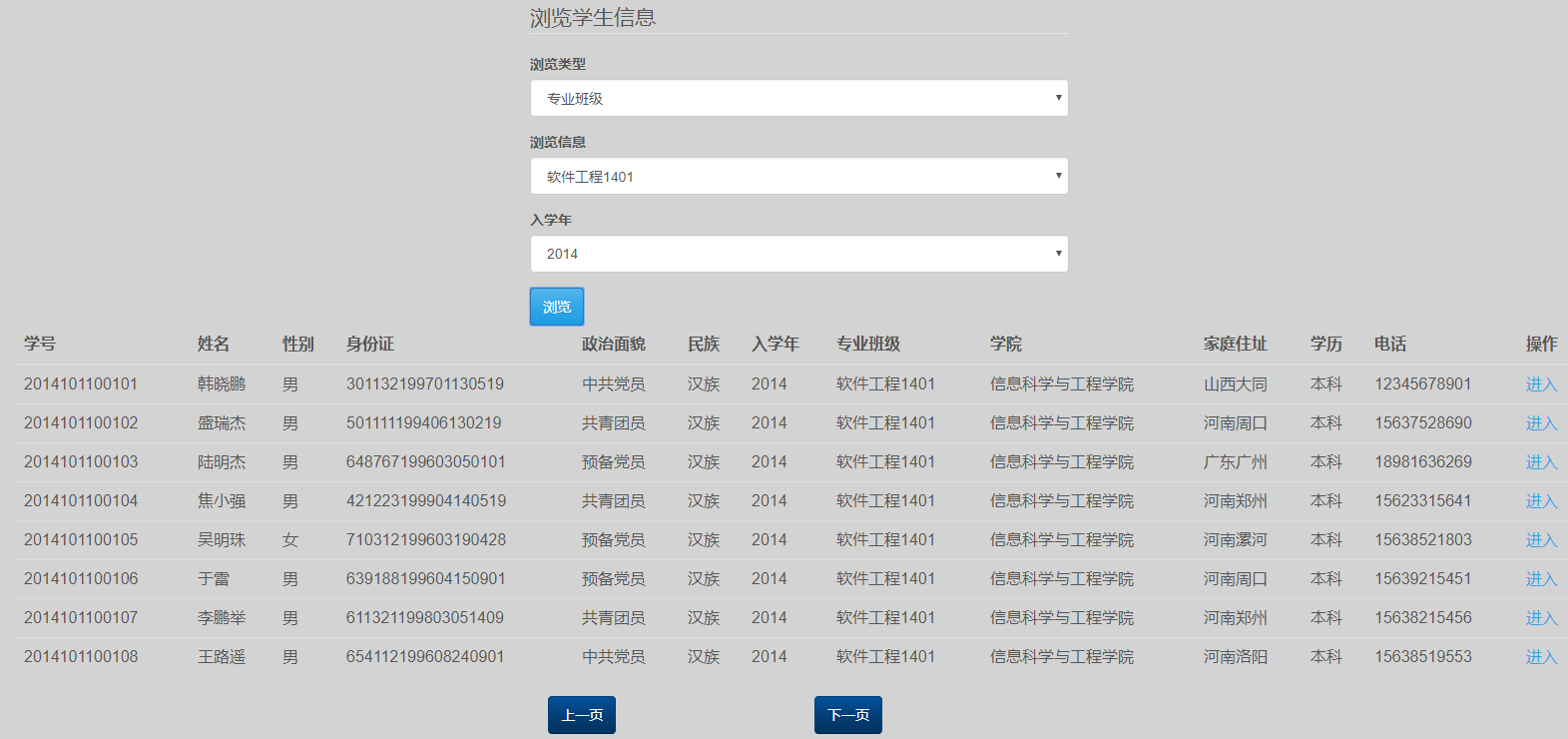


图6.8 筛选学生信息

|  |
| --- |
| studentList = serviceMapper.getStudentMapper().scanBySpecialityAndYear(scanInformation, year); //专业班级浏览  String speciality = scanInformation + "%"; studentList = serviceMapper.getStudentMapper().scanBySpeciality(speciality, year); //专业浏览  studentList = serviceMapper.getStudentMapper().scanByInstitute(scanInformation, year);//学院浏览  Gson gson = new Gson(); String strings = gson.toJson(studentList); |

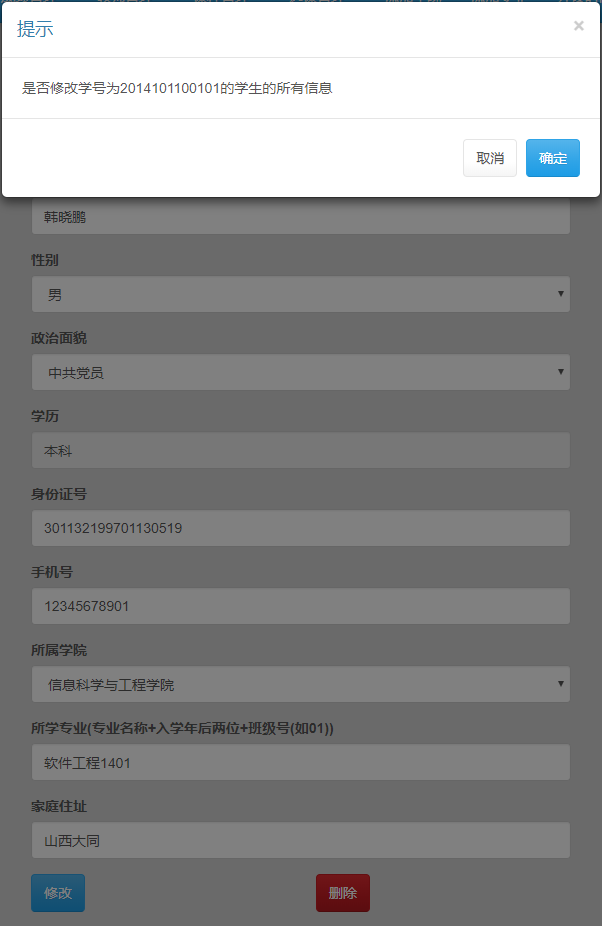


图6.9 修改学生信息

|  |
| --- |
| String classes = update\_studentSpeciality.substring(0, update\_studentSpeciality.length()-4); //获取专业名称 ServiceMapper serviceMapper1 = new ServiceMapper(); Discipline discipline = serviceMapper1.getDisciplineMapper().selectByName(classes); //查看专业信息 List<Student> students = serviceMapper1.getStudentMapper().selectBySpeciality(update\_studentSpeciality); serviceMapper1.close(); |

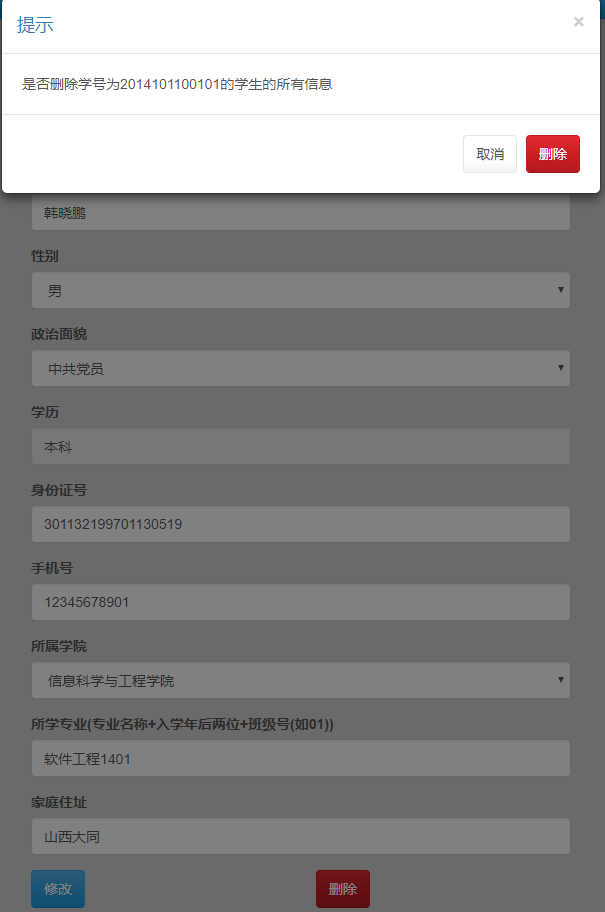


图6.10 删除学生信息

|  |
| --- |
| ServiceMapper serviceMapper = new ServiceMapper(); serviceMapper.getStudentMapper().deleteById(update\_studentId); serviceMapper.getGradeMapper().deleteByStudentId(update\_studentId); serviceMapper.getMakeupMapper().deleteByStuId(update\_studentId); serviceMapper.getRecordMapper().deleteByStudentId(update\_studentId); serviceMapper.close(); |

### 6.2.2 教师管理模块

教师信息在选课模块中占了不可缺少的部分，管理员需要对教师信息进行有效的管理。首先，管理员可以对教师信息进行相应输入，并且需要对身份证号和手机号进行相应的验证处理，还需对教师工号进行处理，判断是否已经存在，若不存在则添加教师信息成功，否则添加失败，管理员可以通过两种方式(工号查询和模糊查询)对教师信息进行相应查询操作，比如筛选教师信息。经过查询之后，可以通过相对应的按钮进行不同的操作。删除教师信息如下图6.11所示。删除教师信息时，该教师所选的课程也将被删除，还有所选课程的学生的成绩信息也将被删除。

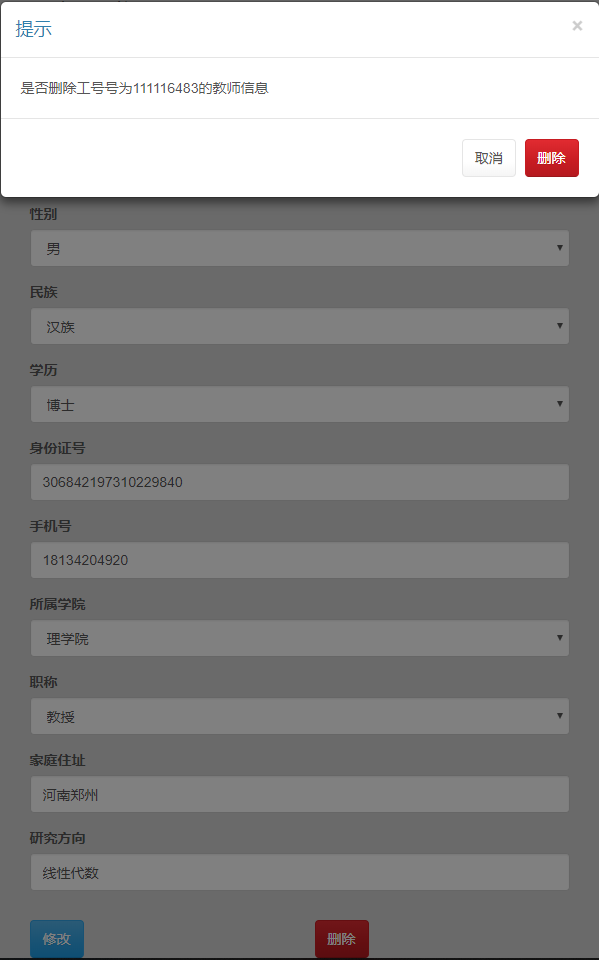


图6.11 删除教师信息

|  |
| --- |
| teacherMapper.deleteById(update\_TeacherId); serviceMapper.getCourseteacherMapper().deleteByTeacherId(update\_TeacherId); List<Courseteacher> courseteacherList = serviceMapper.getCourseteacherMapper().selectByTeacherId(update\_TeacherId); ServiceMapper serviceMapper1 = new ServiceMapper();  serviceMapper1.getMakeupMapper().deleteByTeacherCourse(courseteacher.getId()); serviceMapper1.getGradeMapper().deleteByTeacherCourseId(courseteacher.getId()); serviceMapper1.close(); |

### 6.2.3 课程管理模块

选课模块是本系统中重要的环节，课程信息支撑着选课模块，管理员可以针对于此进行管理。而课程只有教师选择之后，学生才可以看得到课程的，管理员添加课程信息，需要对课程进行相应输入，然后对课程编号进行检测是否存在，若存在则添加失败，否则再验证其它的部分，添加课程信息如下图6.12所示；管理员可以通过两种方式进行查询，比如按编号查询和模糊查询课程，查询之后可以进行相对应的修改和删除操作。



图6.12 添加课程信息

|  |
| --- |
| if (add\_courseProfession.equalsIgnoreCase("所有专业")) {  serviceMapper.getCourseMapper().insert(course);  serviceMapper.close();  response.getWriter().print("<script>alert('课程添加成功')</script>");  response.setHeader("refresh", "0;url=/super\_admin/course/addCourse.jsp"); }  course.setCourseProfession(add\_courseProfession); serviceMapper.getCourseMapper().insert(course); serviceMapper.close(); |

### 6.2.4 选课管理模块

选课是综合教务系统中不可缺少的功能，通过选课会产生一些选课信息，同时也涉及到退选课程，选课也是在一定时间内进行操作，管理员可以查看学生选课记录(如下图6.13所示)和教师选课信息(如下图6.14所示)，而且管理员还可以添加学生选课和教师选课。而当学生选完课程之后，一部分学生是重修成绩，另一部分是正修成绩，管理员可以对成绩进行修改(如下图6.15所示)。同时管理员可以修改教师选课信息中的参考书，同时也可以删除教师选课。



图6.13 查询学生选课记录

|  |
| --- |
| List<Record> recordList = serviceMapper.getRecordMapper().selectByAll(); for (Record record : recordList) {  Student student = serviceMapper.getStudentMapper(). selectByLogin(record.getStudentId());  if (student.getStuYear() == Integer.*parseInt*(year)) {  records.add(record);  } }  List<Courseteacher> courseteacherList1 = operationByInstitute(courseteacherList, institute);  List<Record> recordList = operayionInstitte(records, institute); //通过学院筛选 |



图6.14 查询教师选课信息

|  |
| --- |
| // 初始时显示全部，筛选时根据筛选条件进行筛选  if (year.equalsIgnoreCase("0")) {  courseteacherList = serviceMapper.getCourseteacherMapper().selectByAll(); } else {  courseteacherList = serviceMapper.getCourseteacherMapper().selectByYear(year); } List<Courseteacher> courseteacherList1 = operationByInstitute(courseteacherList, institute); //学院筛选 |



图6.15 修改成绩

### 6.2.5 专业管理模块

学生归属于一个专业，而一个专业也归属于一个学院，管理员可以添加学院信息，并检测学院编号和学院名称是否存在，若存在，则添加失败，否则添加成功；管理员针对于某一个学院添加专业，同时也需要检测专业编号和专业名称是否存在，若存在，则添加失败；若不存在则添加成功。添加学院信息如下图6.16所示，添加专业信息如下图6.17所示。

图6.16 添加学院 图6.17 添加专业

### 6.2.6 时间设置模块

针对于选课、成绩录入等功能的实际实现，必须对这些功能进行时间限制，并不是时时刻刻都可以修改的。故管理员需要进行时间限制，首先选择方式，然后设置起始时间和终止时间，设定时间如下图6.18所示。



图6.18 设定时间

## 6．3 教师模块设计

### 6.3.1 选课模块

教师在选择所要授的课程之后才能让学生进行选课，教师选课分为本专业选课和校级公选课两类。教师选课如下图6.19所示。同时教师可以先查看已选课程信息，并且对应的可以进行退选，而教师在选择必修课时，会相对应专业学生自动默认选课。



图6.19 教师选课

### 6.3.2 当前课表

每一位教师可以查看本学期所要上课程，这个课表根据教师所选的课程来确定。



图6.20 查看课表

### 6.3.3 成绩管理模块

针对于必修课的课程，学生是必修默认选择的，而这个是不需要学生进行选择的。针对于选修课，需要学生自己选择，选择之后教师才可以看到学生选课的情况，同时教师相对应录入学生考试成绩，查看学生选课如下图6.21所示。

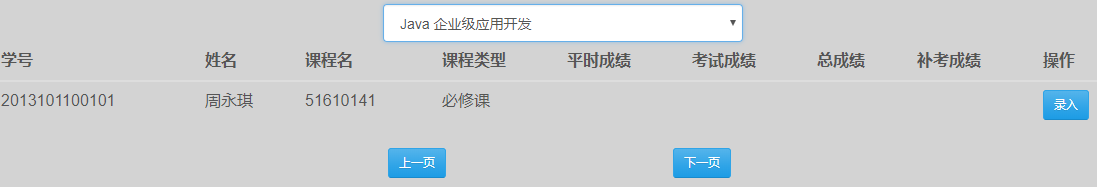


图6.21 查看学生选课

### 6.3.4 成绩分析模块

先对所选本课程的所有学生成绩进行分段，然后进行绘制图形，成绩分析如下图6.22所示。



图6.22 成绩分析

## 6．4 学生模块设计

### 6.4.1 选课模块

学生可以进行本专业选课、校级公选课和重修选课，本专业选课是有补考成绩的，而校级公选课没有补考成绩，更没有重修，在这里重修选课针对于必修课，选修课不存在重修选课。初选选课是选下一个学期的课程，而重修选课时选本学期的重修课，同时，学生可以查看当前所选的初选课程和重修课，并且可以退选。重修选课如下图6.23所示。



图6.23 重修选课

|  |
| --- |
| Makeup makeup1 = serviceMapper.getMakeupMapper().selectByStudentIdAndCourseTeacherId(student\_id, courseteacher.getId()); if (makeup1 != null)  makeups.add(makeup1);  serviceMapper.getMakeupMapper().insertInformation(String.*valueOf*(randomNumber.makeupNumber()), student\_id, courseteacher\_id); serviceMapper.getRecordMapper().insert(new Record(String.*valueOf*(randomNumber.recordNumber()), course\_id, "yes", date, student\_id)); response.getWriter().print(true); |

### 6.4.2 成绩模块

学生可以查到之前到目前所有的考试成绩，包括初选成绩(如下图6.24)和重修成绩。查看初选成绩时，学生可以根据学期进行查询。



图6.24 查询成绩

### 6.4.3 课表模块

课表是每一位学生查看课程信息，而在没有选择学期之前可以查看所学过的所有课程。当学生选择学期时，将会显示该学期所要上课程，包括重修课程，课表如下图6.25所示。



图6.25 查看课表

# 7 系统测试

测试是软件开发过程中非常重要的组成部分，测试主要是为了保证系统能够满足之前的需求。而开发过程中不可能不出现错误，只有在一步一步的调试才能提高系统的正确率，从而达到我们的要求。

## 7．1 管理员子功能测试

管理员在系统中持有最高权限，故管理员所能执行的功能也是最多的。针对管理员所能执行的功能来进行全面测试。为此，针对于每一个功能，都需要进行不同数据的测试，更需要不同的数据来测试以至于解决出现的错误，测试用例如下表7-1至7-6所示。

表7-1 添加学生测试用例

|  |  |
| --- | --- |
| 用例名称 | 增加学生信息 |
| 测试输入：输入姓名、性别、身份证号、手机号、地址等信息 | |
| 预期结果：当点击添加时，如果身份证号的位数不等于18位或手机号不等于11位，再或者含有非数字，则会弹出添加失败；如果该身份证号存在，则也会弹出添加失败。 | |
| 实际结果：无任何异常错误，学生信息添加成功。 | |

表7-2 删除学生信息测试用例

|  |  |
| --- | --- |
| 用例名称 | 删除学生信息 |
| 测试输入：无任何输入 | |
| 预期结果：当管理员点击删除时，应当删除该生的信息、选课记录以及所有选课后的考试成绩等。 | |
| 实际结果：无任何异常错误，删除学生信息成功。 | |

表7-3 添加教师信息测试用例

|  |  |
| --- | --- |
| 用例名称 | 添加教师信息 |
| 测试输入：输入姓名、性别、身份证号、手机号、地址等信息 | |
| 预期结果：当点击添加时，如果身份证号的位数不等于18位或手机号不等于11位，再或者含有非数字，则会弹出添加失败；如果该身份证号存在，则也会弹出添加失败。 | |
| 实际结果：无任何异常错误，学生信息添加成功。 | |

表7-4 删除教师信息测试用例

|  |  |
| --- | --- |
| 用例名称 | 删除教师信息 |
| 测试输入：无任何输入 | |
| 预期结果：当点击删除时，应当将教师信息、教师课程表中的教师所选课程以及教师授课学生的成绩都删除。 | |
| 实际结果：无任何异常错误，教师信息删除成功。 | |

表7-5 增加学生选课测试用例

|  |  |
| --- | --- |
| 用例名称 | 增加学生选课 |
| 测试输入： 无任何输入 | |
| 预期结果： 根据对应的信息进行选择后，则选课成功 | |
| 实际结果：无任何异常错误，增加学生选课成功。 | |

表7-6 录入学生成绩测试用例

|  |  |
| --- | --- |
| 用例名称 | 录入学生成绩 |
| 测试输入： 无任何输入 | |
| 预期结果： 对应的输入平时成绩、考试成绩或补考成绩，若平时成绩和考试成绩含有非数字字符，则录入成绩失败；若平时成绩和考试成绩或补考成绩大于100或小于0，则录入成绩失败，反之则录入成功。 | |
| 实际结果：无任何异常错误，成绩录入成功。 | |

## 7．2 教师子功能测试

教师在选课过程中占有重要的位置，教师可以选择所要教授的课程并且还可以退选，还可以查看当前课表以及录入成绩、成绩分析等。故需要对这些功能进行全面测试。而测试用例如下表7-7所示。教师可以录入选该课程的学生的成绩，而成绩是在0~100之间才可以的，教师录入初始成绩和重修成绩测试用例如下表7-8所示。通过程序进行设计控制，对成绩进行分段得出分析结果，然后采用Echarts进行图形绘制，显示出柱状图和折线图，成绩分析测试用例如下图7-9所示。查看当前课表测试用例如下图7-10所示。

表7-7 选课退选测试用例

|  |  |
| --- | --- |
| 用例名称 | 选课退选 |
| 测试输入： 无任何输入 | |
| 预期结果： 教师可以点击相应课程选课按钮，若该教师已经选了该课程，则提示选课失败；否则选课成功。教师查看选课课程后，管理员可以点击退选，并提示退选成功。 | |
| 实际结果：无任何异常错误，选课或退选成功。 | |

表7-8 录入初始成绩和重修成绩测试用例

|  |  |
| --- | --- |
| 用例名称 | 教师录入初始成绩和重修成绩 |
| 测试输入： 输入平时成绩、考试成绩等 | |
| 预期结果： 教师可以点击相应课程选课按钮，然后显示弹出框，教师输入平时成绩字符、考试成绩字符、补考成绩字符。若该三项中含有非数字字符，则录入失败，最后将三项字符转化为数字，若这三项数字不在0~100之间，则录入失败，否则录入成功。平时成绩\*0.3+考试成绩\*0.7，计算后根据四舍五入得出总成绩。 | |
| 实际结果：无任何异常错误，录入成绩成功。 | |

表7-9 成绩分析测试用例

|  |  |
| --- | --- |
| 用例名称 | 成绩分析 |
| 测试输入： 输入平时成绩、考试成绩等 | |
| 预期结果： 教师可以点击相应课程选课按钮，然后显示弹出框，教师输入平时成绩字符、考试成绩字符、补考成绩字符。若该三项中含有非数字字符，则录入失败，最后将三项字符转化为数字，若这三项数字不在0~100之间，则录入失败，否则录入成功。平时成绩\*0.3+考试成绩\*0.7，计算后根据四舍五入得出总成绩。 | |
| 实际结果：无任何异常错误。 | |

表7-10 教师查看当前课表测试用例

|  |  |
| --- | --- |
| 用例名称 | 查看当前课表 |
| 测试输入： 无任何输入 | |
| 预期结果： 教师查看当前课表的信息，如果教师在本学期没有要上的课程，那么提示本学期暂无课程，否则正确显示课程信息。 | |
| 实际结果：无任何异常错误。 | |

## 7．3 学生子功能测试

每一位学生都可以根据自己的兴趣选择选修课，在本学期选择下一学期的课程，而必修课是在教师选择授课的课程时默认选择的，而重修课是本学期选课的。学生可以选课退选、查看成绩以及查看课表等。针对这些测试，主要是本着提高程序的稳定性和检查程序是否存在隐患，查询成绩测试用例如下表7-11所示。学生可以初始选择课程，学生初修选课测试用例如下图7-12所示。针对于学生必修课没有通过，开设重修选课，而选修课不存在重修选课，学生重修选课测试用例如下图7-13所示。学生查看课表信息如下图7-14所示。

表7-11 学生查询成绩测试用例

|  |  |
| --- | --- |
| 用例名称 | 学生查询成绩 |
| 测试输入： 无任何输入，自行选择不同学期 | |
| 预期结果：学生可以查看初修成绩和重修成绩，而当教师将成绩录入之后，学生才可以查看到，重修成绩也是如此。初始状态可以查看所有初始成绩，而当选择某一个学期时，只显示这个学期的成绩。 | |
| 实际结果：无任何异常错误，查看成绩成功。 | |

表7-12 学生初修选课测试用例

|  |  |
| --- | --- |
| 用例名称 | 学生初修选课 |
| 测试输入： 无任何输入，自行选择课程 | |
| 预期结果： 根据时间的设置，可以查看下学期可修课程，学生只能选选修课，而必修课是教师选完课程时默认选择的，学生不需要选择；而学生首先可以看到所有符合条件的课程，然后必修课不可以选择，选修课选择了后，若该生没有选择该课程，则选课成功，否则选课失败。 | |
| 实际结果：无任何异常错误。 | |

表7-13 学生重修选课测试用例

|  |  |
| --- | --- |
| 用例名称 | 学生重修选课 |
| 测试输入： 无任何输入 | |
| 预期结果： 学生可以重修前几个学期的未通过的选修课，根据不同学期，查看前几个学期的所开设的课程，若学生所有的课程都通过了，则显示不存在重修课程，若学生已经选了该课程，则选课失败，反之，则学生重修选课成功。 | |
| 实际结果：无任何异常错误。 | |

表7-14 学生查看当前课表测试用例

|  |  |
| --- | --- |
| 用例名称 | 学生查看当前课表 |
| 测试输入： 无任何输入 | |
| 预期结果： 学生查看当前课表的信息，如果学生在本学期没有要上的课程，那么提示本学期暂无课程，否则正确显示课程信息。 | |
| 实际结果：无任何异常错误。 | |

# 结 论

在过去一个月的时间里，毕业设计的过程并不是一路顺利的，从开始论文选题到完成毕业设计，让我懂得需求分析、数据库设计和方案设计的重要性。功能测试和系统测试来检测自己设计的不足，同时需要对系统进行优化。

毕业设计让我经历了软件开发的全过程，经过不断的实践，综合教务管理系统按时完成了。综合教务管理系统从实际需求出发，完成选课退选、学生信息管理和教师信息管理等功能，正确使用软件工程方法进行设计与研究，通过查找资料让我们学习了在学校没有学到的东西——比如MyBatis、JavaScript等技术，同时也需对系统设计进行全方位考虑。而这次毕业设计让我认识到了很多，也让我们更好的将大学我们所学的知识运用到实践中。大学培养了我们解决实际问题的能力，对系统进行调试，发现了自己系统存在错误，对于错误，自我在网上查找资料和寻求老师的帮助解决，然后对系统不断进行完善。本系统虽然按时完成了，但是难免存在缺陷，比如成绩分析表的打印，是否毕业以至能否选重修课，成绩的计算未能动态设计，根据实际情况部分选修课可以重修以及不能将课程合并选择。此次设计开发，需要与实际结合，才能设计得更好。

毕业设计整个过程，让我更好的认识到自己知识不足，需要在开发中学习，在学习中开发，同时要远看整个过程，不能仅局限于目前。开发是一个持续的过程，只有在开发中不断的完善系统，才能让系统更加稳定。

# 致 谢

即将告别大学四年生活的我们，无论在专业学习方面还是为人处理方面，都得到了老师和同学们的帮助，而他们的帮助是永远的，将会影响我们今后的人生。在此我对给予我帮助的朋友们表示衷心的感谢。

我非常感谢我的指导老师—程凤娟老师，在过去几个月里从开题到毕业论文的完成，老师给予我专业的帮助和技术的指导，在我无法进行设计时，给我指引了方向，让我在这段时间中坚持完成开发，在设计完成时，给我提出新的需求，不断完善系统。我的毕业论文并不是那么完美，但是对我的人生重大的影响。

# 参 考 文 献

[1] 疯狂软件.Spring+MyBatis企业级应用实战.电子工业出版社.2017

[2] 徐涛.深入理解Bootstrap.机械工业出版社.2014

[3] 黄艳秀.基于mybatis的面向数据库自动生成技术.河南科技大学,2014

[4] [David M. Kroenke,David J. Auer](http://hwopac.haut.edu.cn/opac/openlink.php?author=(%E7%BE%8E)David+M.+Kroenke%EF%BC%8C(%E7%BE%8E)David+J.+Auer%E8%91%97).数据库处理.电子工业出版社.2016

[5] 范新灿.JSP综合项目开发案例精编.高等教育出版社.2013

[6] 林学良.JSP&Servlet学习笔记.清华大学出版社.2012

[7] 刘继承.JAVA程序设计及实验.清华大学出版社.2012

[8] Metsker S J.Java框架设计.第一版.电子工业出版社.2012

[9] 刘勇军.Java Web核心编程技术.电子工业出版社.2014

[10] 陈强.精通Java开发技术.清华大学出版社.2014

[11] 石正喜.MySQL数据库实用教程.北京师范大学出版社.2014

[12] 崔洋.MySQL数据库应用从入门到精通.中国铁道出版社.2013

[13] Buell.Data structures using Java. Jones&Bartlett Learning.2013

[14] Martin Ngobye.Computing Static Slice for Java Programs.2012

[15] Nicholas S.Williams. Professional Java for Web Applications:Featuring Websockets,Spring Framework,JPA Hibernate,and Spring Security.Wrox Press,USA,2014

[16] 陶伟. 基于J2EE的高校综合教务管理系统的设计与实现[D].山东大学,2014

[17] 郭丽娟. 基于J2EE的教务综合管理系统的设计与实现[D].吉林大学,2017

[18] 魏春茹. 基于J2EE的综合教务管理系统设计与实现[D].电子科技大学,2014

[19] 于虹博. 基于SSH架构的高校选课系统的设计与实现[D].黑龙江大学,2016

[20] 朱其现. 基于JSP的教务管理系统的设计与实现[D].厦门大学,2013

[21] 牛雪. 高校教务管理系统设计与实现[D].天津大学,2016