

分 类 号: TP311  
研究生学号: 201253H362

单位代码: 10183  
密 级: 公 开



# 吉 林 大 学

## 硕士学位论文

(专业学位)

基于 Java Web 的学生成绩信息管理系统  
Student Performance Information Management  
System based on JAVA WEB

作 者 姓 名: 杨宝林

类 别: 工程硕士

领域(方向): 软件工程

指 导 教 师: 张 猛 教授

培 养 单 位: 计算机科学与技术学院

2015 年 5 月

基于 Java Web 的学生成绩信息管理系统

**Student Performance Information Management  
System based on JAVA WEB**

作者姓名：杨宝林

领域（方向）：软件工程

指导教师：张猛 教授

类别：工程硕士

答辩日期：2015 年 5 月 16 日

未经本论文作者的书面授权，依法收存和保管本论文书面版本、电子版本的任何单位和个人，均不得对本论文的全部或部分内容进行任何形式的复制、修改、发行、出租、改编等有碍作者著作权的商业性使用（但纯学术性使用不在此限）。否则，应承担侵权的法律责任。

### 吉林大学硕士学位论文原创性声明

本人郑重声明：所呈交的硕士学位论文，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：杨琳

日期：2015年6月1日

## 摘 要

## 基于 Java Web 的学生成绩信息管理系统

当今世界，如果要评判一个国家的发展速度与进步水平，那么这个国家对教育的重视程度将会是一个重要的考量因素。随着我国经济水平的不断提高，国家政府对教育事业也越来越重视，而随着受教育人数的不断增加，高校对学生各个层次的管理变得越来越困难，需要我们思考如何高效简便的解决学生的管理工作。

传统的学生管理模式，不仅需要耗费大量的人力物力，还极其容易出现错误，特别是在学生的成绩管理这一方面，传统的管理方法已经无法适应庞大的学生人数和繁复的科目划分。基于以上理论基础，本次研究旨在设计一个基于 Java Web 开发技术的网上学生成绩管理系统，方便学生随时随地查看自己的成绩，修改自己的学生信息，也方便教师录入学生成绩以及管理员维护系统。

基于 Java Web 的学生成绩管理系统的研究与开发，是采用 Java Web，在网络上构建一个动态的成绩管理过程。本次研究采用 B/S 三层架构的模式，即浏览器和服务端架构，使用 Struts 开源框架实现 MVC 三层模式进行开发。在数据库连接方面采用 JDBC-ODBC 桥驱动程序，采用的关系数据库为 SQL Server 2005，整体开发环境我们选择的是 Eclipse。

随着社会发展，各大教育机构使用电子管理系统已经越来越普遍，网上管理省却许多不必要的麻烦，而且提高了管理效率，降低了管理工作中的错误率。相信本次研究所开发的学生成绩管理系统，会在未来的高校管理工作中拥有长足的发展空间。

**关键词：**

**Java Web， 管理系统， 成绩， Struts， 数据库**

## **Abstract**

### **Student Performance Information Management System based on JAVA WEB**

In today's world, if you want to judge the development of the level of assessment of a country and the progress, so the importance of education will be an important consideration. As China's economy continues to improve, the state government is more and more attention to education, and with the increasing number of education, management of students of different levels of colleges and universities is becoming more and more difficult, we need to think about how to solve the simple and efficient student management work.

The traditional student management mode, not only need to spend a lot of manpower and material resources, but also very prone to error, especially the performance management in the students on the one hand, the traditional management methods have been unable to adapt to the number of the students in the huge and complicated subjects. Based on the above theory, the aim of this study was to design an online student performance management system based on Java Web development technology, which is convenient for students to check their grades whenever and wherever possible, modify student information, but also is convenient for teachers to input student achievement as well as the administrator for the maintenance of the system.

Research and development of student achievement management system Java based on Web, is the use of Java Web, the construction of performance management in a dynamic process in the network. This study uses B/S three layer structure mode, namely the browser and server architecture, the use of open source Struts framework for the realization of the MVC three layer model for development. In the database connection using JDBC-ODBC bridge driver, using the relational database SQL Server 2005 integrated development environment, we choose Eclipse.

With the development of society, the educational institutions to use electronic management system has been more and more popular online management, save a lot of unnecessary trouble, but also improve the management efficiency, reduce management errors. That student achievement management system for the development of this institute, will have great development space in the future management of colleges and universities.

**Keywords:**

**Java Web, management system, performance, Struts , Database**

---

第一章 绪论.....	1
1.1 研究背景.....	1
1.2 研究现状.....	3
1.3 研究意义与目标.....	4
1.4 全文组织结构.....	5
1.5 本章小结.....	6
第二章 基础知识及相关设计原理.....	7
2.1 Java 相关技术简介.....	7
2.1.1 Java 简介.....	7
2.1.2 JSP 技术简介.....	8
2.1.3 JDBC 简介.....	9
2.2 B/S 模式分析.....	10
2.3 Struts 框架.....	11
2.4 Servlet 技术.....	12
2.5 开发工具介绍.....	13
2.5.1 Eclipse 介绍.....	13
2.5.2 SQL Server 2005 介绍.....	14
2.6 本章小结.....	14
第三章 开发的可行性与需求分析.....	16
3.1 可行性分析.....	16
3.1.1 技术可行性.....	16

3.1.2 经济可行性.....	17
3.1.3 操作可行性.....	17
3.1.4 时间可行性.....	17
3.1.5 法律可行性.....	18
3.2 需求分析.....	18
3.2.1 功能需求.....	18
3.2.2 运行需求.....	20
3.2.3 数据需求.....	20
3.3 本章小结.....	21
第四章 系统分析与总体设计.....	22
4.1 系统结构设计.....	22
4.2 系统功能模块设计.....	22
4.3 系统流程设计.....	24
4.3.1 普通用户流程设计.....	24
4.3.2 超级用户流程设计.....	26
4.4 数据库设计.....	27
4.5 本章小结.....	31
第五章 系统的详细设计与实现.....	33
5.1 连接数据库.....	33
5.2 系统界面的搭建与登录的实现.....	35
5.2.1 系统界面的设计与实现.....	35
5.2.2 登录功能实现.....	37



5.3 学生功能实现.....	40
5.4 教师功能实现.....	43
5.4.1 成绩功能实现.....	43
5.4.2 其他信息查询.....	48
5.5 管理员功能实现.....	52
5.6 本章小结.....	56
第六章 总结与展望.....	57
6.1 工作总结.....	57
6.2 研究展望.....	57
参考文献.....	59
作者简介.....	61
致    谢.....	62

## 第一章 绪论

一个国家的发展速度与发展水平取决于这个国家对教育的重视程度,随着我国经济水平的不断提高,国家政府对教育事业也越来越重视,而随着受教育人数的不断增加,中国大学开始不断地扩招,随之而来的是高校对学生各个层次的管理变得越来越困难,因此高效简便的解决学校的学生管理工作成为高校业务发展的一个新的课题。

在传统的 student 管理中,需要校方投入大量的人力和物力,效率低下不说,还极其容易出现错误,特别是在学生的成绩管理这一方面,传统的管理方法已经无法适应庞大的学生人数和繁复的科目划分,在此基础上,本次研究应运而生。本次研究旨在基于 Java Web 的技术基础上,设计一个网上学生成绩管理系统,方便学生随时随地查看自己的成绩,修改自己的学生信息,也方便教师录入学生成绩,方便管理员维护系统。

### 1.1 研究背景

在我国,教育是影响国家进步和人民幸福的重要指标,特别是大学教育,随着这几年国家发展速度加快,大学变得平民化,越来越多的人进入大学进行学习生活,在提高了基础教育的水平的同时,这种学生人数急速增加的现象也为学校的正常运行和学生的管理带来了非常多的不便,校方通常需要配备更多的人手来管理学校各个方面的正常运行,而且在管理工作中也会额外增加许多支出,为学校的正常运转造成了很大的困难,也为广大教育者提供了一个改进教育管理系统的机会。

从国外的发展来看,西方许多大学早就引入了电子管理系统的概念,学生可以通过网络在任何时间任何地点访问学校的管理系统网站,浏览学校发布的学术信息,或者查看自己的课程与考试成绩,这种方便的管理方式给了国内教育从业人员很大的思索空间,这种网上的学生管理系统大大节省了人力物力,学生可以在家查看系统上的信息,老师在批改完试卷后,也可以在家或者公共场所直接上传成绩,而不必回到办公室,至于学校的管理人员,这种系统更是极大的减轻了

他们的工作压力,他们只需要提供正确的身份校验就可以以管理员的身份进入系统,发布学校的最新动态,查看学生的信息,并且可以根据实际情况及时的添加或者删除学生的个人信息,这种网上学生管理系统的出现为学校的日常管理提供了一个便利,轻松,高效的途径。

在国内的大学生中,我们经常可以看到这样一个情形,就是学生在期末考试结束离开学校后,老师才批改完试卷,进行成绩的登记和录入,那么这时由于学生已经离开了学校,无法查询到自己的考试成绩,忐忑不安。而网上学生成绩系统的实现,使得同学们可以在任何有网络的地方登陆学校的网站,进行有效的身份验证,从而查看到考试成绩。更重要的是如果实现了这种网上管理系统,我们就可以做到随时随地发布重要信息,为同学和老师掌握第一手的学校学术娱乐信息提供了有效的途径。

在研究技术方面,随着科学水平的不断提升,电子化信息化产业进入了一个飞速发展的时代,计算机技术已经强大到可以处理日常生活中的纷繁复杂的管理工作。本次研究基于 Java Web 的基本设计思想,采用 Jsp,在 B/S 模式和 Struts 架构的基础上,完成系统的设计与开发。

本次开发所设计的管理系统适合于各种网络应用,在开发工具的选择上,我们使用 Eclipse 作为开发工具, Eclipse 是一款开源的集成开发工具,能够处理大部分 Java 语言的 Web 程序编辑工作,是现今做得最好的一款多语言集成开发环境。在使用 Eclipse 来开发之前,首先需要配置 JRE,并且接入 Tomcat 软件,实现虚拟服务器的连接,方便我们实现网页端的程序编辑,并对部分功能进行测试与实现。

关系数据库我们选择 SQL Server 2005, SQL Server 2005 的特点是适合小型数据的管理,适合条目清晰,表格连通明显的数据库表格文件,同时大多数与之相关的开发工具都是免费的。

SQL Server 2005 通过协议层传递 TDS,然后分析和执行 SQL 命令,在查询优化器和执行器的工作下将数据发送到存储引擎,并最终进行提取, SQL Server 2005 具有一个数据缓冲池,方便我们暂时存储已经具有查询结果的结果集。图 2-4 所体现的是 SQL Server 2005 所涉及的服务器与数据库交互流程。

本次系统的设计与实现力求完成一个全面的,高效的,简洁的,并且通用的

学生成绩管理系统。特别是针对系统的通用性，由于本次设计并不是针对某一所高校或者某一个学院，而是针对具有相同特点的一系列学生管理工作而设计的，所以必须对学生管理工作中出现问题的普遍性加以把握，在系统的可移植性和复用上面做好前期准备，以便于后期系统的维护和更改，同时也为更多教育机构使用本系统打好基础。

## 1.2 研究现状

从国内大学教育的发展轨迹来看，我国高等教育已经进入一个急速扩招的时代，几乎每一所大学，每一个专业都已经吸纳了超过需求几倍之多的学生，我们抛开人才以后进入社会以后的问题不谈，单从学校教育和管理这么多的学生这个角度出发，就迫切的需要为教育事业的从业者和管理者提供一种新的管理思路，并且为这种新的管理思路给予理论支持，并为这种新的管理思路的实现做好技术上的工作准备。

长久以来，学生的成绩管理是管理工作中的重点与难点，重要性在于学生的成绩关系到一直以来学习水平的检验，与各种评奖评优挂钩，关乎到学生各个方面的利益，甚至关乎到改变学生人生轨迹的机会的把握，而困难性在于，现今学生成绩早已经不是上世纪人们所理解的单纯的分数，而是需要根据科目，专业的不同进行划分，合理的进行加权工作，并在接收各方面综合评比和学生全面信息的基础上，通过一套精密的算法计算出一个最为合理的“成绩”。这种思想的提出，为学生之间的公平竞争提供了最为直接的保障，但也为技术上的实现造成了困难，单纯使用人力去实现这样一套算法，并且在算法的实施过程中不断地去更改和维护最终得到的结果，这实在是太困难了，几乎不可实现，在这个基础上，我们需要设计的系统必须实现以下几个特点：

（1）全面，系统必须涵盖普通学校成绩管理的各个方面，分别从教师和学生角度出发，去探究所出现的一切可能，比如学生重名，分数不规律等，另外在技术允许的情况下要先于教师对一门成绩进行自动的整理和分析，最终提供给老师和同学一部分直观且清晰的数字，让他们对于某一门考试的成绩有一个全面的认识；

(2) 高效, 系统必须建立在一个高效率的概要设计的基础上, 需要首先做好学生, 教师和管理员三个身份的用例图, 一切从实际出发, 避免一些华而不实的功能, 为学生和老师提供一个最为高效的运行方式, 此外也要考虑管理员的日常维护工作, 需要在设计时就为管理员的一系列操作设计好入口;

(3) 简便, 我们必须认识到除了计算机和软件专业人员之外, 学校中许多专业的同学和老师并没有很强的计算技术知识, 所以我们设计的系统必须方便用户的操作, 在一些重要的功能中提供最为直接的图形界面, 特别是在连接数据库的时候, 要仔细设计数据库表格, 尽可能多的联系表格与表格之间的关系, 考虑使用者的感受;

(4) 通用, 上文中已经说过, 本次设计并不是针对某一所高校或者某一个学院, 而是针对具有相同特点的一系列学生管理工作而设计的, 所以必须对学生管理工作中出现问题的普遍性加以把握, 在系统的可移植性和复用上面做好前期准备, 以便于后期系统的维护和更改, 同时也为更多教育机构使用本系统打好基础。

### 1.3 研究意义与目标

本次学生成绩系统的设计与实现, 旨在打破原有的传统教育机构管理方式, 为教育工作者提供一种管理学生新的思路与方法, 并且实现这种新的方法。从每一个用户的角度出发, 实实在在的考虑到传统管理方式的不足与缺憾, 从而在新的网上管理系统中避免这些问题。从实际出发, 我们为将要开始实现的系统提出以下目标:

(1) 本次研究所设计的管理系统应当具备实用性, 避免以往设计工作中华而不实的一面, 同时应该充分考虑系统的通用性, 为以后系统的升级和系统将来在多个教育机构中的重复使用做好前期工作;

(2) 正确有效的设计数据库表格, 方便教师和学生的理解, 同时防止数据库在存储过程中出现错误;

(3) 提供全面的用户身份验证方法, 对不同等级的用户提供不同的身份验证, 从而实现系统根据不同等级的用户提供不同查询方法和查询结果的目的;

(4) 系统应该留好外接设备的入口, 包括打印机传真机等办公设备, 以便

于对成绩查询结构及时的进行打印汇总或者传真；

(5) 对学生信息和成绩要严格保密，同为学生等级的用户不能查询除了自己以外的学生信息，更不能对这些信息进行更改，以免造成信息录入错误，系统管理混乱等结果；

(6) 仅仅在需要外接信息交互的情况下才让用户对系统进行内部访问，否则不允许非管理员身份的人进入系统内部；

随着社会发展，各大中小学校中使用电子管理系统已经越来越普遍了，网上管理的优势非常明显，能够在日常教育机构管理中省却许多不必要的麻烦。相信本次研究所涉及的学生成绩管理系统，在未来的高效管理工作中将会有长足的发展空间。

## 1.4 全文组织结构

全文一共分为六章，除了本章绪论外，其余五章安排如下：

第二章：基础知识及相关设计原理，主要介绍了本次开发的基础知识和所使用的相关技术，分别从编程语言，组件与工具，设计模式等角度分析了本次开发的实现前期准备，最后介绍了本次开发所使用的开发工具；

第三章：开发的可行性与需求分析，主要从技术，经济，操作，时间和法律五个角度分析了本次开发的可行性，从功能，数据等方面对本次开发做出系统的开发需求；

第四章：系统分析与总体设计，对要实现的系统架构，事务处理流程，具体功能模块的实现做出详细的说明，对数据库的实现理论，数据库设计，实体属性设计和实体信息表设计做出说明；

第五章：系统的详细设计与实现，详细介绍了本次研究的实现部分，对每一个功能部分做出介绍，介绍了使用的关键类与函数，每一项功能，分别从关键代码和页面展示两个角度介绍，全面叙述了开发的整个流程；

第六章：总结与展望，主要总结了研究工作，同时针对目前研究工作中存在的一些不足提出了相应的研究展望。

## 1.5 本章小结

绪论部分主要向读者介绍了当前教育机构管理系统的发展背景,国内外学生管理系统的发展趋势以及在该领域的一些基本的研究状况,通过绪论读者能够对本次研究有一个简单的认识,同时也能够了解到本次研究的现实背景,以及本文的章节安排。本章起着开篇和总体指导的作用。

## 第二章 基础知识及相关设计原理

### 2.1 Java 相关技术简介

Java，是一门我们所熟知的编程语言，它最先由 Sun 公司推出，是一门跨多平台，具有高移植性的面向对象编程语言。使用 Java 语言进行开发最大的特点和优势在于 Java 所编辑程序的多平台运用能力，即 Java 所编辑的程序可以在多个平台上进行使用。现今生活中 PC 与手机充斥着我们的生活，而大多数的手机游戏或者手机应用程序都是使用 Java 语言编写的。Java 编辑的程序，无视所根植的操作系统，或者是硬件上的差别，Windows 系统上可以使用的程序，到了 Linux 系统上照样可以使用，Java 程序可以在任何平台上运行，可以说是通用的语言之一<sup>[1]</sup>。

接下来的部分，我们将对 Java 语言以及一些相关的技术做出介绍，让读者对本次研究中所涉及的基础知识有一个简单的了解。

#### 2.1.1 Java 简介

Java 语言最初在 1995 年由美国 Sun 公司推出，是一种极富创造力和想象力的面向对象高级程序设计语言。它的设计者是有 Java 之父之称的 Sun 研究院院士，James Gosling 博士。Java 具有很强的可移植性，使用 Java 语言进行开发最大的特点和优势在于 Java 所编辑程序可以在多种系统或者平台上运行。现今生活中随处可见高科技终端设备，而人们日常生活中接触到的大多数的手机游戏或者手机应用程序都是使用 Java 语言编写的。Java 编辑的程序，无视所根植的操作系统，或者是硬件上的差别<sup>[2]</sup>，Windows 系统上可以使用的程序，到了 Linux 系统上照样可以使用，Java 程序可以在任何平台上运行。

Java 的编译过程类似于 C 语言，采用的是一种编译和解释相交加的过程。Java 程序所使用的关键字或者定义的一些类文字非常贴近日常生活中的使用，开发人员很容易掌握这种语言编辑模式，也很容易进行连锁开发，同时，Java 中有



着太多封装好的功能类，我们可以直接使用，为开发人员省去了许多烦恼。

Java 程序的编译过程可以通过下面这个图来理解。在一开始，程序代码在编译器中进行编译，所生成的是一系列字节码，这些字节码被送到 Java 虚拟机(Java Virtual Machine JVM)，Java 虚拟机将字节码转化为机器码，然后机器码在操作系统上开始运行<sup>[3]</sup>，Java 程序运行的整个过程非常类似于 C 或者 C++，但实际上在编译和回收机制处理等方面，Java 和 C 又有着显著的不同，这一特性在网络端程序编写方面体现的非常明显，图 2-1 展示了具体的编译和执行过程。

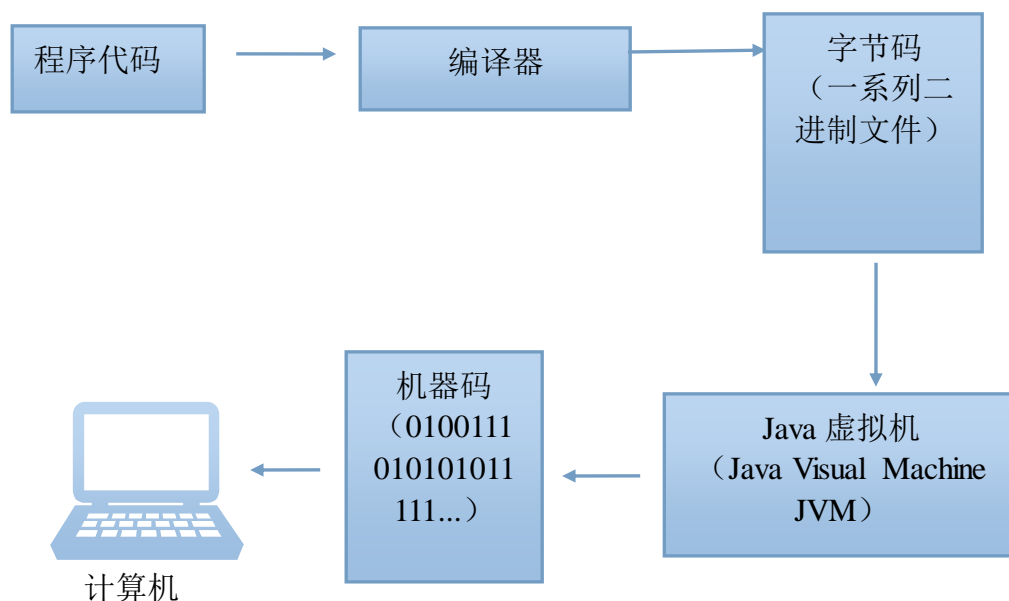


图 2-1 java 程序的编译和执行过程

### 2.1.2 JSP 技术简介

JSP(Java Server Pages)技术实现了存在于 HTML 语法和思维中的 Java 扩张，JSP 技术与 Servlet 一致，都是本地编写，服务器端执行，从这个角度讲，就不难理解 JSP 技术的高移植性，从使用上来讲，JSP 技术是在用 Java 语言来编写一种类 XML 的文件（包括 tags 和 scriptlets）<sup>[4]</sup>，这一点与现今存在的 ASP 技术很像，众所周知，ASP 技术就是一种本地编写，服务器端执行的技术，这种技术在 JSP 出现之前广泛运用于 HTML 的语法和思维中，而 JSP 比 ASP 优秀的地方，就是它轻量化了网页编写的整个工程，省去了很多不必要的代码的编写，大量节省了开发人员的时间和精力。

JSP 编写一个非常好的地方在于，不同于其他技术所实现的，网页返回的是

一个相当复杂的地址或其他信息，JSP 技术所实现的是返回一个 HTML 文本，首先从文件量级上就减轻了传送的压力，其次因为只传送一个 HTML 文本，就没有跨平台执行的压力了，所以从这个角度来说 JSP 技术更适合于当今的网络发展。

JSP 技术最伟大之处，就是脱离了认为网页编写和网页逻辑必需捆绑实现固化思维，将网页逻辑与网页的设计与显示分别开来，从模块和组件的角度分化和重组程序，支持基于组件的动态网页开发与设计，让存在在服务器端的各式各样的应用开发变得快速且简便。当一个 Web 端的服务器开始处理基于 JSP 技术的网页处理请求时，第一件事是访问该动态网页的主程序部分，包括程序模块和各种组件，然后在这个基础上，进一步实现设计与编写动态网页所需要的各种功能和技术<sup>[5]</sup>。

JSP 相比于现阶段其他技术的优势在于：

(1) Java 更出色，更人性化，程序在移植过程中，除了系统之外，源代码部分不需要做出任何改变。

(2) JSP 具有强大的可伸缩性<sup>[6]</sup>。从编写实现一个小小的 Jar 文件，就可以跨平台的运行 Servlet，JSP 技术，到通过联通多台服务器一起来实现进行的集群式开发和多负载联合均衡。从一台服务器进行的事务处理，到多台服务器共同进行应用级的事务处理，甚至是消息处理，系统维护这种较为复杂的技术。

(3) JSP 汇集了无数优秀的开发人员，他们共同奋斗已经得到了一大批高效实用的开发工具，更加可贵的是这其中的许多开发工具我们免费就能得到，在 JSP 实现开发的时间里，已经有许多开发工具得到测试并顺利的运行在多个环境与平台之上。

### 2.1.3 JDBC 简介

JDBC (Java Data Base Connectivity)，是一个 Java 程序连接数据库的应用接口，使用 JDBC 技术，我们实现本地数据存储到网络端的目的，同时使用 JDBC-ODBC 桥技术，我们可以用编辑好的一种访问程序访问多种关系数据库，这个特点在我们保持程序的复用性，即当不同教育机构使用同一个管理系统来连接不同数据库时，程序同样可以运行。

JDBC 中常用的类和接口有 **Connection** 接口，它代表所执行程序与何种关系数据库进行连接，并在连接后执行相应的 SQL 语句。**Statement** 接口，我们在连接数据库成功后，使用这个接口来发送 SQL 语句。**PreparedStatement** 接口，用于动态执行 SQL 语句。**DriverManager** 类，这个类属于 JDBC 的管理层，方便我们动态管理数据库。**ResultSet** 接口，用于临时安置和存储一些已经经过查询得到的结果集。

对于开发人员，作为 API 来讲，JDBC 可以说是为程序的开发与实现提供了一个业界认可的标准的接口，JDBC 强大的包容性使其使用了现有的 SQL 标准的同时，还仍然支持与其它类型数据库连接的标准（比如 ODBC 之间的桥接）。

JDBC 实现了以上所述的这些高效，系统的目标，使程序连接数据库进入统一的轨道，统一的模式，实现了一种高性能的接口。

## 2.2 B/S 模式分析

所谓的客户级别应用程序，指的是我们在系统中为了实现用户与数据直接交互所设定的一个模块。而服务器管理程序，在这里主要负责高效快速的对各方面资源进行系统的管理，打个比方，比如服务器管理程序就会实现对一个系统用户信息数据库的管理，服务器级别管理程序还有一个重要的功能，那就是当存在多个客户同时访问服务器上的一个资源时，也就是出现我们所说的死锁，或者饥饿现象时，我们的服务器管理程序可以对用户所访问的公共资源进行最大程度的优良化管理。中间组织者，它的功能是当发生应用程序与数据库的交互时，可以很好地连接客户应用程序与服务器管理程序。

而我们这里所说的 B/S 模式，是作为一种在网络互联的基础上，叠加用户层的管理，所形成的新兴的 MIS 系统平台<sup>[7]</sup>。从原理的角度讲，B/S 模式是把刚才所介绍的 C/S 模式中，那种统一的服务器管理层，分成了一个或几个服务器，或者是服务器与应用程序的结合，从而形成了一种用户与服务器传送信息的三层次系统管理平台。

三层系统管理平台的第一层，指的是用户与网络所连接的接口部分，我们日常所实现的应用程序，在这里将会被集成显示在浏览器中，在这个显示过程中，HTML 文件会把单调的程序变成我们日常所见到的五光十色的网页，我们日常所

浏览的信息也大多来源于此。与此同时，我们会发现日常生活中我们所接触的网页不仅仅是在显示信息，我们还通过网页在交互许多重要信息，比如在线填写一些表格，在网上传送文字交流思想等等，那么这就涉及到信息往服务器端的提交，而这个提交过程，就要使用到接下来将要介绍的第二层。

第二层中，WEB 服务器将根据用户上一层所做出的行为来提供响应，这个响应过程是这样来进行的，首先服务器会根据用户要求生成一系列的网页代码，而这个网页代码中已经包含了用户所要求的结果，将这个生成的网页代码发送至用户浏览器，用户便得到了这一次交互所需要的信息。需要注意的是，如果用户首次提出的要求中包括对数据库的访问，那么 WEB 服务器还需要与数据库合作完成用户的需求，其中涉及的技术过于复杂，我们在这里不做解释。

第三层是数据库服务器，数据库的管理过程类似于以上介绍的 C/S 模式，都是基于对并发任务的协调。

通过以上的介绍，我们不难发现 B/S 模式相较于 C/S 模式有着很强的适应性，更加能够适应现今复杂的网络环境。事实也是如此，现今大多数企业都已经开始设计并研发自己的 B/S 模式下运行方案，由于学生管理系统客户量极其庞大，而且客户分级复杂，客户的差异性小，所以需要使用一种多级访问，多级控制的管理系统，便于各个级别的把关和监控。B/S 模式是我们本次研究的最佳选择，基于 B/S 模式设计的学生成绩管理系统将大大减轻管理环节的任务。

## 2.3 Struts 框架

通常我们所说的 Struts 框架，指的是在 2004 年由 Apache 开源软件联盟所开发的一个开软软件。Struts 框架的英文全称是 Jakarta Struts Framework。

Struts 框架的理解与 MVC 设计模式有着密不可分的关系，在基于 Struts 框架的设计与开发中，首先显而易见的是它是面向对象设计的，其次它在使用 MVC 的思想时充分考虑了编写代码的弹性<sup>[8]</sup>。比如在 Struts 框架中，所有代码被形象的划分为以下三个部分，Model，Controller 和 View。在 Model 部分的代码，我们可以将它们形容为是处理事务与应用范围内的相关逻辑，它们的主要作用在于对范围内的对象状态进行管理，并提供一个加以运用的模板；

Controller 部分的作用是连接 Model 与 View，也就是说它将接收到来自 View

部分的数据，资料和信息，经过整理和运算再发送到 Model 部分中去，完成一次信息的交互，也就是完成了一个流程控制过程；

至于 View 部分，则主要负责收集并展示一部分我们收集的信息和材料，并且在与用户产生信息交互时，可以获取信息输入者的资料。View 是整个框架的信息展示部分。

Struts 是一个基于弹性控制的控制层，它将 MVC “分离显示逻辑和业务逻辑”的综合思想运用的恰到好处，体现了它动态的一面，可以说 Struts 是 MVC 的一种框架展示，它符合 MVC 的多种特性，并且在和 J2EE 技术相结合后产生了多层系的扩展与提升，许多结构和特性都随着核心技术的发展有了显著地改变。图 2-2 展示了 Struts 框架的工作原理。

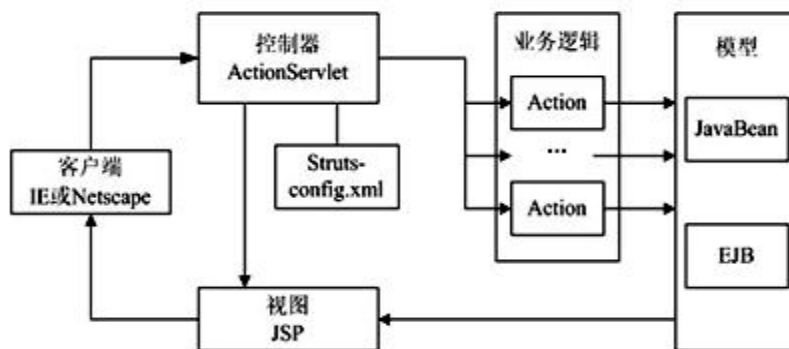


图 2-2 Struts 框架的工作原理

## 2.4 Servlet 技术

Servlet 指的是我们编辑运行在服务器上的一组程序，使用 servlet 程序来实现远端服务器的使用，并使用 Servlet 技术来进行数据库存储和远端数据的各种处理，这种程序往往是微型的，片段化的，每一个独立的类或者函数构成一小组程序，这一组程序可能只有一种功能，依靠这一组小程序解决一个问题，这些程序被整合成一些独立的代码文件，与网页捆绑在一起，进行编译和使用<sup>[9]</sup>。Servlet 的使用一般都是在打开网页的客户端上，用户打开网页，与网页捆绑在一起的 Servlet 程序开始运行，进行一些数据计算，图形操作或者是定位等功能，当用户关闭网页时，Servlet 程序进行自动保存，记录一定的用户信息和操作流程，随着网页自动关闭。图 2-3 形象的反映了 Servlet 的实现过程。

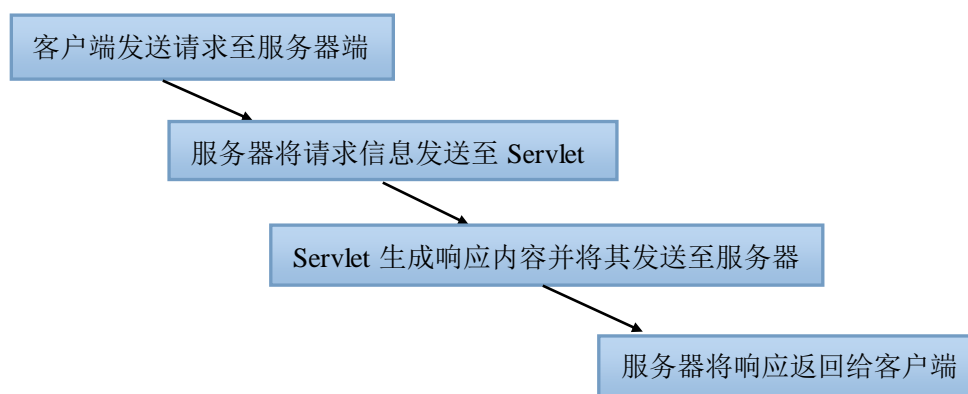


图 2-3 Servlet 的实现过程

Servlet 的生命周期如下，首先是实例化，即通过 Servlet 容器加载一部分组件，然后成功创建一个 Servlet 实例；

其次是初始化，开始运行 `init()` 方法，`init()` 方法用于初始化 Servlet 实例，这个过程首先建立一个 `ServletConfig` 接口，通过这个接口向 Servlet 发送我们即将初始化的一些配置文件信息，是名称--数值的形式。接着 Servlet 再通过 `ServletConfig` 接口连接到 Servlet 容器，交互信息，最终通过 `getServletConfig` 类来获取 `ServletConfig` 的实现对象；

然后是服务过程，通过实现 Servlet 组件中的 `GenericServlet` 类的 `Service` 方法，我们可以直接处理提交的用户请求，然后用 `ServletRequest` 接口和 `ServletResponse` 接口，分别实现请求的发送和响应。

最后是释放过程，执行 `Destory` 方法，来释放掉 Servlet 创建的实例。

## 2.5 开发工具介绍

### 2.5.1 Eclipse 介绍

Eclipse 是一款开源的集成开发工具，能够处理大部分 Java 语言的 Web 程序编辑工作，Eclipse 是现今做得最好的一款多语言集成开发环境。在使用 Eclipse 来开发之前，首先需要配置 JRE，并且接入 Tomcat 软件，实现虚拟服务器的连接，方便我们实现网页端的程序编辑，并对部分功能进行测试与实现。

在使用 Eclipse 作为开发环境时，JDK(Java Development Kit)的查阅为我们减轻了许多困难，许多类的使用和继承都变得一目了然，对于一些类有时候我们需

要重写其中的一些功能，这就需要我们查找它的父类，而有了 JDK，这个查找过程将变得十分轻松。

## 2.5.2 SQL Server 2005 介绍

SQL Server 2005 的特点是适合小型数据的管理，适合条目清晰，表格连通明显的数据库表格文件，同时大多数与之相关的开发工具都是免费的。

SQL Server 2005 通过协议层传递 TDS，然后分析和执行 SQL 命令，在查询优化器和执行器的工作下将数据发送到存储引擎，并最终进行提取，SQL Server 2005 具有一个数据缓冲池，方便我们暂时存储已经具有查询结果的结果集。图 2-4 所体现的是 SQL Server 2005 所涉及的服务器与数据库交互流程。

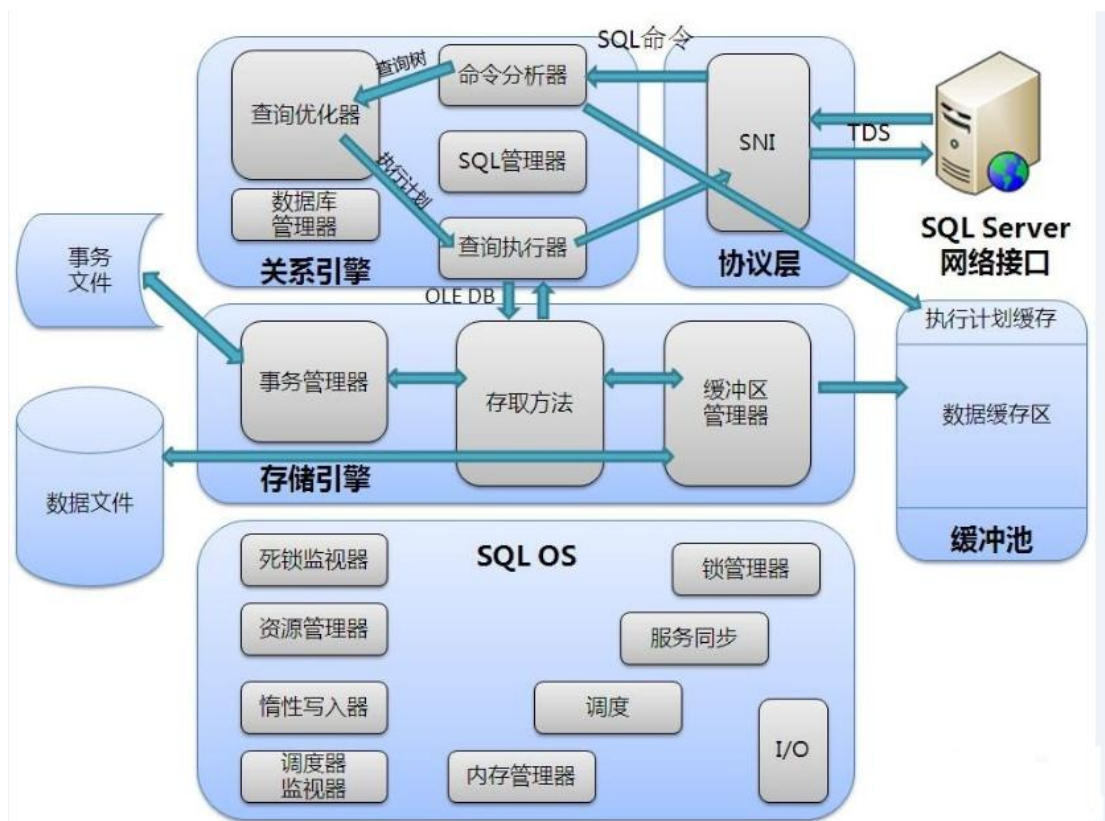


图 2-4 SQL Server 2005 所涉及的服务器与数据库交互流程

## 2.6 本章小结

本章主要描述了本次开发的基础知识部分以及相应的开发技术与开发环境。

从 Java 语言及相关技术的介绍，再到 B/S 模式分析和相关知识浏览，再到 Struts 框架简介和 Servlet 技术的介绍，最后是此次开发的工具介绍。通过这些知识的了解让我们对此次开发有了一个更加具象的认识。通过阅读本章读者会了解相关技术概念，为理解后面章节打下基础。



## 第三章 开发的可行性与需求分析

### 3.1 可行性分析

可行性分析是一次开发中非常重要的一环,它并不是去解决什么技术上的问题或者设计上的困难,而是从各个方面考虑本次研究的整个过程是否可行,这个过程中我们将会考虑到项目从技术上来讲的难点与攻克,将要使用的技术有哪些以及如何从开发技术的可行性上来说明整个项目开发是能够完成的。除了技术上的可行性,我们还将说明本次开发周期经济上是可行的,也就是说开发过程的整个费用预算在我们的控制范围之内,或者说开发系统将来的经济价值是非常可观的。操作可行性是指本次所开发的学生成绩管理系统便于操作,交付给使用人员后是能够在预计内掌握基本使用原理的。时间可行性是指开发的整个过程符合我们的日期计划,而法律可行性指的是开发过程不涉及任何法律侵权问题,不冒用或者抄袭他人的科学成果。

从直观上来看,可行性分析似乎在做着一些无用功,但事实并不是这样,可行性的分析研究是一次开发过程的根本。下面我们将从技术,经济,操作,时间和法律五个角度剖析本次学生成绩管理系统的可行性。

#### 3.1.1 技术可行性

技术上的可行性指的是我们在本次系统开发中将使用什么技术,基于什么平台,将会套用什么框架以及将会采用什么设计模式,在基础知识介绍部分我们已经进行了相关说明,本次开发是基于 Java Web 的一次网络管理系统开发,所使用的语言是 Java,在开发过程中主要基于 Struts 框架,采用 Jsp 技术进行服务器端的开发,并且整个开发思路采用 MVC 设计模式,弹性管理代码,对代码部分的划分采用 Model-Controller-View 三个部分。

Eclipse 是现今做得最好的一款多语言集成开发环境,用户可以通过准确的接口在 Eclipse 上实现多种语言程序的互联。在使用 Eclipse 来开发之前,首先需要配置 JRE,并且接入 Tomcat 软件,实现虚拟服务器的连接,方便我们实现网页端的程序编辑,并对部分功能进行测试与实现。而选用 SQL Server 2005 是因

为 SQL Server 2005 适合小型数据的管理，适合条目清晰，表格连通明显的数据库表格文件，同时大多数与之相关的开发工具都是免费的。

### 3.1.2 经济可行性

经济可行性指我们的开发过程符合我们之前的开发预算，整个开发过程的费用支出都有明确记录，并且符合开发过程的要求。并且在系统完成后，所开发的学生成绩管理系统有着非常好的前景，具有很高的经济价值，这一点在系统的开发之前也在我们的考虑范围之内，综合以上考虑，我们认定本次管理系统的设计与开发从经济上来讲是可行的。

### 3.1.3 操作可行性

一个系统设计的成功与否，一个非常重要的考量因素就是使用者的使用感受，从学生管理系统的角度考虑，我们必须认识到非专业人员即使是高校教育人员也很难对内部系统进行调试，出现问题时也很难解决，那么在设计之初就应该明确，设计的整个系统是封装好的，系统的内部运行和功能模块并不对外界展示，而且除了管理员其他等级的用户不能随意查看系统内部的信息，包括系统升级记录，功能更改记录等。

另外一点，系统交给使用人员后，要保证系统使用简便，操作便捷，大部分功能都已经通过设计好的图形界面来实现，数据输入也要设计的灵活完整，不产生歧义，方便教师和学生登记和检查自己的信息。整个系统必须是简洁的，高效的，使用过程中要做好不同等级用户的使用权限管理。从以上这些部分考虑，我们可以认定本次开发从操作上来讲是可行的。

### 3.1.4 时间可行性

整个系统在开发之前我们已经做了详细的工作计划，整个开发将按照时间表来进行，从而充分的保证整个开发计划的时间可行性。从 2014 年 4 月到 6 月，我们将要完成系统的需求分析，同时也要学习相关知识技术，积累一定的项目经验。从 2014 年 7 月到 12 月，这一阶段我们将逐步完善系统的各个模块功能，并

进行相应的测试，撰写中期检查报告。在 2015 年 1 月至 2 月这两个月里，将要总结前几阶段的工作并对系统设计的不足和缺陷提出展望，同时开始撰写论文。而在 2015 年 3 月，整个开发过程进入尾声，我们将完成论文，并完成提交审阅阶段。

### 3.1.5 法律可行性

从法律的角度来讲，整个开发工作具有以下几个特性：

- (1) 开发所使用的所有技术资料都是合法得到的；
- (2) 整个开发过程中不存在知识产权问题；
- (3) 开发过程未抄袭任何网站，不存在侵犯版权问题；
- (4) 开发过程全部来自于大家合作，原始劳动，不涉及承担所谓的法律责任。

综上所述，此次学生成绩管理系统的开发过程是可行的。

## 3.2 需求分析

本节中我们将主要为大家介绍本文的需求分析部分，需求分析是指在对所开发的项目有了一个具体的了解后，开始对所要实现的功能做出一定的分析，并围绕功能的实现，程序的运行，数据设计以及安全运行等方面给出一定的实现目标，方便接下来的开发过程。

### 3.2.1 功能需求

本次系统开发的功能需求基于学生，教师和管理员三个等级，学生主要查询自己的个人信息，包括考试成绩，教师可以查看，录入和修改管辖学生的某一科成绩，管理员负责维护系统运行，可以查看某一科成绩的各种情况，并进行导出和打印。

功能需求的用例图是用来形象的说明所要实现的功能，以下分别是三个等级用户的功能用例图。

- 学生
- 1, 验证个人信息, 登录系统;
  - 2, 查询个人基本信息: 学号, 班级, 课程表等;
  - 3, 查看成绩;
  - 4, 查询自己的所有成绩, 并得到平均分, 总分等指标;
  - 5, 查看某一科目的年级平均成绩, 或班级平均成绩;
  - 6, 查询自己的绩点等相关指标;
  - 7, 打印和下载自己的成绩;
  - 8, 退出系统;

- 教师
- 1, 验证信息, 登录系统;
  - 2, 查看成绩;
  - 3, 录入某一科成绩;
  - 4, 修改或更新某一科成绩;
  - 5, 查询某一科目的平均成绩, 以及该科目的最低最高分;
  - 6, 统计某科成绩的附加情况;
  - 7, 打印和下载统计结果;
  - 8, 退出系统;

- 管理员
- 1, 验证信息, 登录系统;
  - 2, 查看成绩;
  - 3, 查询某一科目的平均成绩, 以及该科目的最低最高分;
  - 4, 录入某一科成绩;
  - 5, 修改或更新某一科成绩;
  - 6, 统计某科成绩的附加情况;
  - 7, 更改或添加已经发布的功能;
  - 8, 添加公告信息;
  - 9, 退出系统;

### 3.2.2 运行需求

硬件条件：酷睿 CPU 及以上；

1G RAM 及以上；

10G 硬盘及以上；

打印机；

局域网；

软件条件：Eclipse；

SQL Server 2005；

IE 8.0 及以上；

Windows XP 及以上；

### 3.2.3 数据需求

数据需求是指存储在数据库文件中的数据设定，在本次学生成绩管理系统设计中，需要设定的数据有不同等级用户的登陆用户名和密码，学生的学院，系别，年级，班级和课程表信息，某一门科目的名称，标号，学时，上课老师，分数（及对应绩点），以及学分等，对于已经录入的一门成绩，还要记录该考试成绩的考试类型，等级划分，考试时间等信息，对于管理员，还应当记录管理员真实姓名，所任职务，联系方式以及上次登录系统时间等，以便于记录系统维护信息<sup>[10]</sup>。

数据需求体现在 ER 图中 ER 图完全由开发人员设置，这样就充分的体现了人性化的思想，数据库的存储契合于人类的理解，也就方便存储和升级，以后也可以在这个基础上轻松添加或删除数据<sup>[11]</sup>，还有一点是 ER 图与计算机存储结构无关，设置的时候较为简单，不需要考虑数据结构如何建立在系统中，如何去存储。

数据库的建立十分复杂与繁琐，在这里我们先给出对数据需求的分析，在接下来的章节中我们将会从 ER 图，实体属性表，实体信息表和数据关系图等各个方面来讲解数据库的确立。

### 3.3 本章小结

本章主要描述了开发的学生管理系统的研究可行性和需求分析，可行性方面包括系统开发的技术可行性，经济可行性，操作可行性，时间可行性和法律可行性，从每一个方面来印证本次开发过程是可行的。需求分析方面我们主要分析了本次设计的功能需求，运行需求和数据需求，具象的给出了每一个功能模块的实现用例，具体的实现过程将在接下来的章节中进行描述，本章为将来系统的开发工作打下了基础。

## 第四章 系统分析与总体设计

### 4.1 系统结构设计

按照不同用户等级来划分，管理员具有系统管理，学生信息管理的权限，教师具有成绩信息管理，成绩统计和授课信息管理的权限，而学生具有成绩及相关信息查询的权限，在管理员对系统进行管理时，可以添加，删改用户，并且对系统信息和功能进行修改，而学生等级的用户，可以查询自身相关信息，不能查看别人信息，也不能修改自己的信息。图 4-1 为学生成绩管理系统结构图。

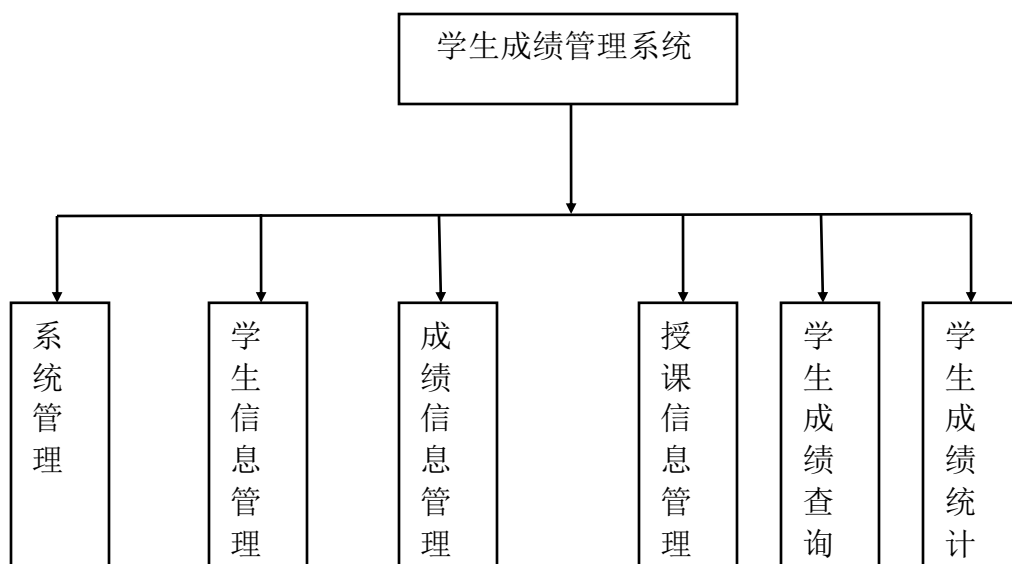


图 4-1 学生成绩管理系统结构图

### 4.2 系统功能模块设计

在功能模块设计中，用户被分为普通用户和超级用户，超级用户就是之前我们所说的系统管理员，这个等级的用户可以进行系统管理，包括信息发布和系统功能的删改，而普通用户可以是学生或者老师，学生可以查看自己的系统信息，查询成绩，老师可以完成成绩录入，更改，查询成绩，统计成绩信息并打印，管理所教课程等工作。每一个等级的用户都需要进行身份验证进入系统，但系统内

部不针对普通用户开放，内部的管理与维护也将由超级用户来完成，且普通用户没有权利更改系统信息<sup>[12]</sup>。

在系统管理员，教师和学生中，用户分级显得尤为重要，且对于教育管理系统，数据管理和查看是很重要的，不同等级的用户只能查看或者更改一部分数据。数据的更改对于管理员是没有限制权限的，管理员作为超级用户具有所有权限，教师对部分成绩具有更改权限。

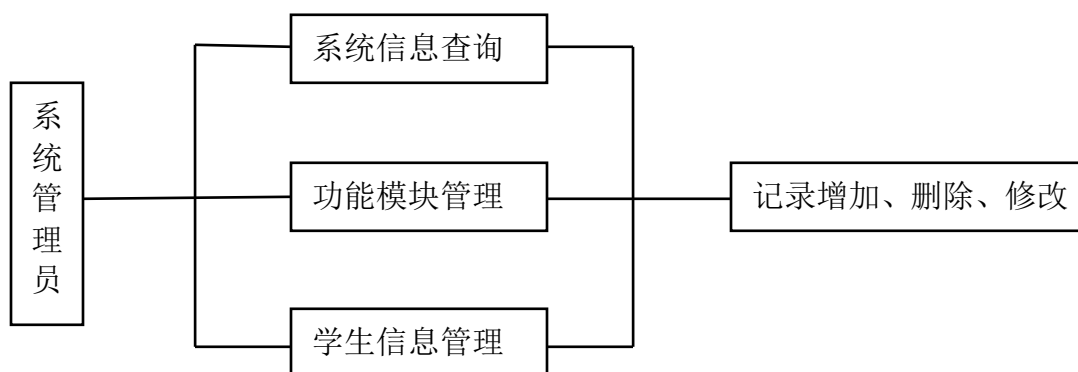


图 4-2 管理员功能模块图

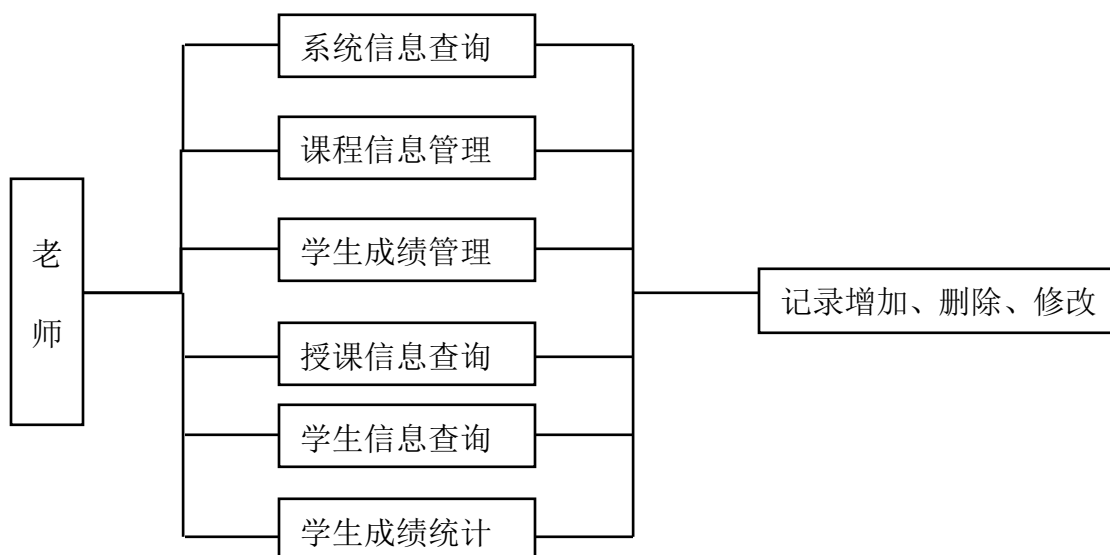


图 4-3 教师功能模块图



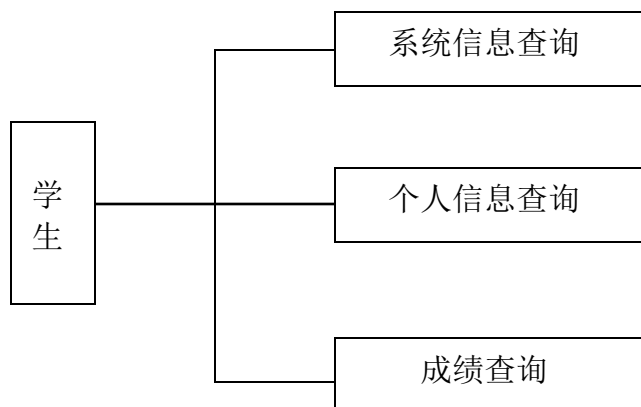


图 4-4 学生功能模块图

## 4.3 系统流程设计

流程设计是指模拟不同等级的用户进入系统,根据设定好的步骤来对系统进行使用,按照要求测验系统可能被使用到的每一项功能,并考虑可能出现的所有情况,成功的或者失败的<sup>[13]</sup>。流程模拟过程需要设定好进入系统的用户等级,本次系统中用户等级可以简单地分为普通用户(教师,学生)和超级用户(管理员),接着进入系统后,需要在每一个信息输入环节进行一次判定。流程设计应当遵循“进入系统-----退出系统”的黑盒测试原理<sup>[14]</sup>,模拟这期间发生的所有动作,本节将分别对普通用户和超级用户两个等级给出具体的流程设计<sup>[15]</sup>。

### 4.3.1 普通用户流程设计

普通等级用户包括教师和学生,他们的系统使用流程分为以下几个步骤,首先是系统登录,验证身份信息,如果验证成功则进入系统,分别进入系统设定的教师页面和学生页面,若验证失败则返回上一步,进入系统后,系统显示菜单栏,教师选择将要执行的功能,开始功能使用过程,使用完毕后可以选择注销本次登陆,然后退出系统<sup>[16]</sup>。

图 4-5 为大家详细描述了普通用户的功能使用流程。

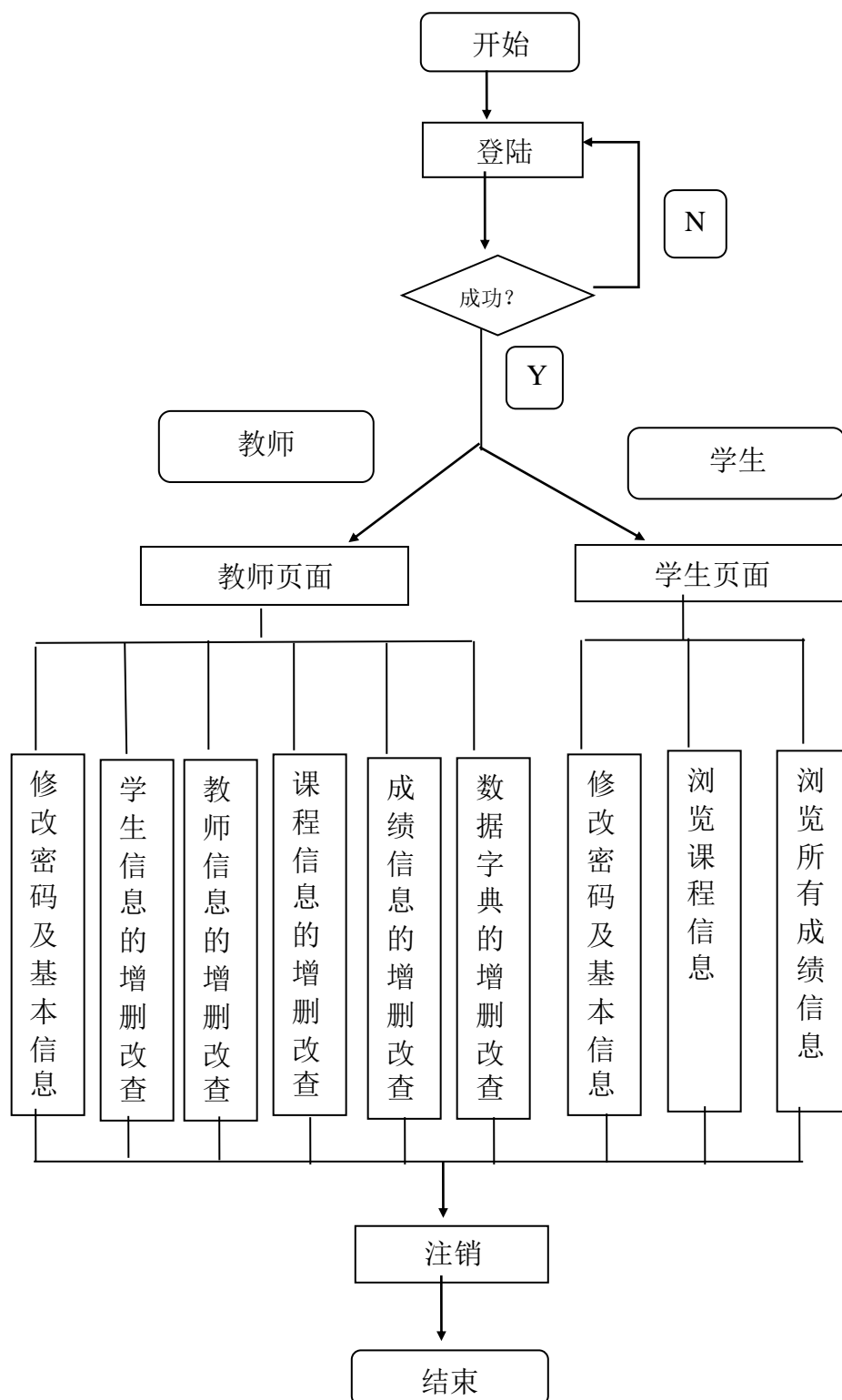


图 4-5 普通用户流程图

### 4.3.2 超级用户流程设计

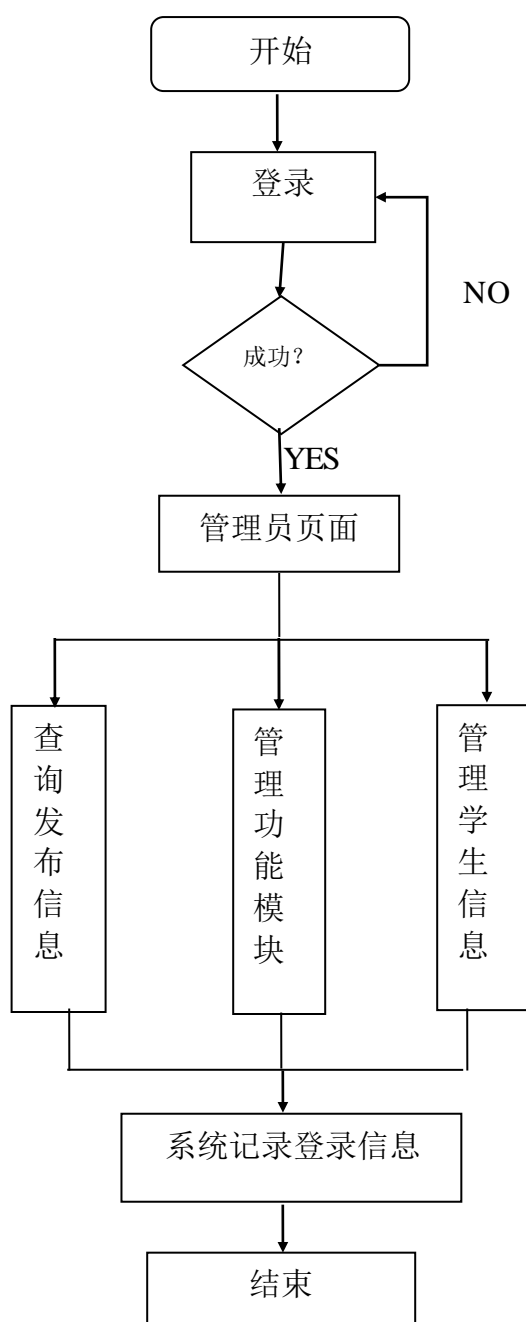


图 4-6 超级用户的流程图

管理员用户的系统登录流程与普通用户相同，进入系统后，直接跳入管理员界面，管理员选择将会要执行的功能，然后系统调用功能模块，功能执行完毕后退出系统。与普通用户不同，系统将会对管理员的登录信息进行记录，比如管理员的登录系统时间，登陆 IP 地址和登录之后的行为等，这是因为管理员用户具

有更改系统和升级系统的权限,如果系统出现错误需要对上一层的维护记录进行查看,方便系统维护,而且因为管理员可以发布系统信息,为了对这一行为进行监管,需要对其发布信息的时间,内容进行记录<sup>[17]</sup>。图 4-6 是超级用户的流程图。

## 4.4 数据库设计

所谓实体联系模型,指的是在这种关系模拟方式中,我们首先选取好需要统计和录入的信息,根据信息的数量和彼此之间的关系画出 ER 图,接着转换为需要保存的数据模型<sup>[18]</sup>。这样做的好处是,ER 图完全由开发人员设置,这样就充分的体现了人性化的思想,数据库的存储契合于人类的理解,也就方便存储和升级,以后也可以在这个基础上轻松添加或删除数据,还有一点是 ER 图与计算机存储结构无关,设置的时候较为简单,不需要考虑数据结构如何建立在系统中,如何去存储。

但 ER 图也有它的不足,那就是 ER 图本身是由工作人员人工绘制的,他只是说明了数据与数据,结构与结构之间的关系,这种模拟化的数据存储流程非常不利于计算机运算,很可能会浪费存储资源,而且有很多信息是人类需要的,但计算机并不需要存储,如果存储太多这样的冗余信息,会使得数据库系统运行非常缓慢,计算机处理过程也十分盲目,如果一开始的 ER 图设计不合理,数据存储关系不明确,后期更改将会非常困难<sup>[19]</sup>。

首先根据系统设定制作出开发数据库系统的 ER 图,每一部分都设置主键,教师 ID,管理员 ID 和课程号是三大主类的主键,另外的数据元素包括教师和管理员的姓名,以及登录密码等。学生信息储存在学籍信息部分,这样设计主要是便于整个系统的管理,在学籍信息中,学生姓名,学号等信息也是需要记录的,而对于成绩信息,成绩所属课程,课程号和成绩是需要被记录的,设置班级信息主要是统计班级人数后,便于教师统计一门学科的班级总分或者平均分,也就方便输出这门成绩的其他相关信息,其他信息都包括在绘制的 ER 图中,图 4-7 是本次系统的 ER 图。

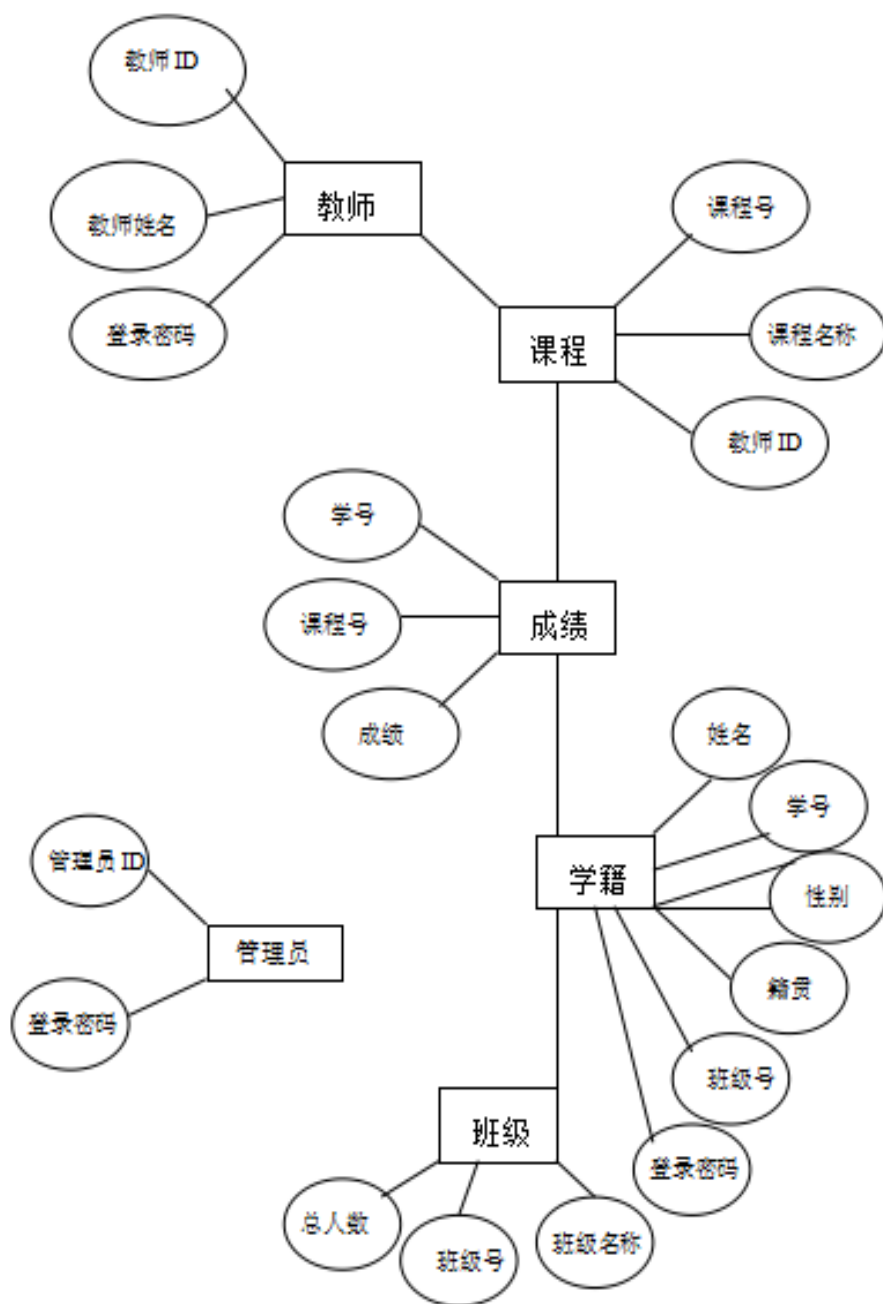


图 4-7 学生成绩管理系统 ER 图

上图仅仅是本系统中的总数据 ER 图，还有许多分类的实体 ER 图，在这里不多做解释，ER 图的展示目的是说明数据的存储关系。

根据 ER 图可以很明显的看出个部分数据之间的关系，我们知道管理员具有管理功能模块的权限，但是这一特性在数据设定中无法体现，所以管理员的数据实体仅仅是单独出来的一块，管理员权限的独立性同样体现在他对数据的操作的权限上，因为我们可以确定管理员可以查看所有信息，也正因为如此管理员的身份验证才需要进一步加强，避免有人利用管理员的身份进入系统，恶意篡改信息。

所以在实现过程中应当确定管理员的数据是独立的,完备的,不受其他数据限制,这样才能最大程度上保证管理员身份的安全<sup>[20]</sup>,也就最大程度上保证系统的安全性。下面给出数据库表格的具体设计方案,根据之前列出的属性管理,我们可以设计六个数据实体,分别是教师,课程,学籍,成绩,班级和管理员,数据实体将构成整个数据库结构,是数据库表格的目录,同时要知道实体的属性应该是简介而且说明问题的,表 4-1 是本系统的实体属性表。

表 4-1 实体属性表

实体	属性
教师	教师 ID, 教师姓名, 登录密码
课程	课程号, 课程名称, 教师 ID
管理员	管理员 ID, 登录密码
班级	班级号, 班级名称, 所属学院, 总人数
学籍	学号, 姓名, 性别, 班级号, 籍贯, 登录密码
成绩	学号, 课程号, 成绩

根据实体属性,创建实体的信息表,下表 4-2 至表 4-7 分别是这六个实体的具体信息设置表格。

表 4-2 教师信息表

字段名	类型	空值	约束条件
教师 ID	varchar(8)	not null	主键
教师姓名	varchar(8)	not null	
登录密码	varchar(8)	not null	

表 4-3 课程信息表

字段名	类型	空值	约束条件
课程号	varchar(8)	not null	主键
课程名称	varchar(12)	not null	
教师 ID	varchar(8)	not null	外键

表 4-4 管理员信息表

字段名	类型	空值	约束条件
管理员 ID	varchar(10)	not null	主键
登录密码	varchar(10)	not null	

表 4-5 成绩信息表

字段名	类型	空值	约束条件
学号	varchar(15)	not null	主键，外键
课程号	varchar(8)	not null	主键，外键
成绩	smallint	not null	

表 4-6 班级信息表

字段名	类型	空值	约束条件
班级号	varchar(8)	not null	主键
班级名称	varchar(8)	not null	
所属学院	varchar(6)		
班级人数	nchar(4)		

表 4-7 学籍信息表

字段名	类型	空值	约束条件
学号	int	not null	主键
姓名	varchar(30)	not null	
性别	char(2)		
班级号	varchar(30)	not null	外键
籍贯	char(10)		
登录密码	varchar(30)	not null	

在这里需要对各个实体信息表中的主键确认进行说明，我们分别明确了教师 ID，课程号，管理员 ID 等信息作为各个实体表的主键，这不仅仅是因为这些信息相对其他信息比较重要，在后期数据库建立和数据库维护中我们知道，数据库的连接十分重要，有时候一项信息的更改会导致与这项信息相关联的许多信息发生变化，比如在教室实体信息表中，我们确立的主键是教师姓名，那么当这名教师改了名字，或者学校里有同名同姓的教师，数据库就需要进行大面积的更改和维护，因为教师信息变化后课程信息也变了，而这种变化所带来的工作量将是巨大的。所以通过上面的描述大家可以知道，所确立的主键应当是轻易不会发生改变的，简单明了的，并且与其他表格有着唯一的联系的，比如个人编号，编号极少出现重复，而且储存简便，易于分解和传递，最重要的是根据序号的排列和择选可以很好地区分学生团体的各种级别，事实证明在数据库建立中编号是最好的

主键选择。

数据库关系图形象的描述了实体之间的联系，其中的钥匙代表的就是各实体的主键，而实体与实体之间的钥匙代表了通过上一个实体信息来确认下一个实体的主键，这种确认由于存在层级关系，便于管理，也便于工作人员分级掌握人员情况，实现信息的节制与保密，最终得到的系统中的数据关系图如下图 4-8。



图 4-8 数据关系图

## 4.5 本章小结

在这一章节中，我们着重介绍了本次开发的结构，功能模块，事务处理流程和数据库的设计，开发结构明确了开发的具体部署，并且根据开发结构来设定功能模块，在事务处理流程中我们分为普通等级用户和超级用户两个层面，分别给出具体的流程图来分析程序的运行过程。最后给出了全面的数据库设计，首先从



ER 图的设定，再到具体实体及属性的确定，再到确认每一个实体的具体信息表，最后给出实现数据库文件的 SQL 语句，并展示了系统的数据关系表。

## 第五章 系统的详细设计与实现

### 5.1 连接数据库

数据库连接我们使用 JDBC-ODBC 桥驱动程序, 目的在于使用 JDBC 连接技术来实现多接口的数据库连接, JDBC-ODBC 桥技术旨在通过 JDBC 接口连接数据库, 借用 ODBC 的多兼容性来实现多个数据库的连接。由于 ODBC 已经可以连接多个关系数据库, 所以使用 JDBC-ODBC 桥驱动程序可以间接地实现多个关系数据库的连接。以下是数据库连接类代码:

```
package cn.system.manage.admin;
import java.util.ArrayList;
import cn.system.manage.tools.PageBean;
public class adminMgr {
    /**
     * adminMgrImpl
     */
    private adminMgrImpl impl;
    public adminMgr(){
        /**
         * Constructor
         */
        impl = new adminMgrImpl();
    }
    public boolean existedadmin( admin admin ){
        return impl.existedadmin( admin );
    }
    /**
     * Get Method: get all objects split by search
     * @param
     * @return
     */
    public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
        return impl.getadminListSplitBySearch(page,adminSearch);
    }
    public ArrayList getadminList( admin admin ){
        return impl.getadminList( admin );
    }
}
```

```

    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);
    }
    public int getMaxadminIdByadminname(String adminname) {
        return impl.getMaxadminIdByadminname(adminname);
    }
}
public class adminMgr {
    /**
     * adminMgrImpl
     */
    private adminMgrImpl impl;
    /**
     * Constructor
     */
    public adminMgr(){
        impl = new adminMgrImpl();
    }
    /**
     * check existed
     * @param
     * @return boolean
     */
    public boolean existedadmin( admin admin ) {
        return impl.existedadmin( admin );
    }
    /**
     * Get Method: get all objects split by search
     * @param
     * @return
     */
    public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
        return impl.getadminListSplitBySearch(page,adminSearch);
    }
    /**
     * Get Method: get all objects by condition
     * @param
     * @return
     */
    public ArrayList getadminList( admin admin ){
        return impl.getadminList( admin );
    }
    /**

```

```

    * Get Method: get all objects split
    * @param
    * @return
    */
    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);
    }
    /**
    * 得到最大的 admin id
    */
    public int getMaxadminIdByadminname(String adminname) {
        return impl.getMaxadminIdByadminname(adminname);
    }
}

```

在数据库连接代码中,我们首先明确 JDBC-ODBC 桥的设立需要导入几个连接数据库的包,也就是 cn.system.manage.tools.PageBean, 这里 PageBean 的导入为我们省去了几个常用类的麻烦,最终返回的是 impl.getMaxadminIdByadminname 类型的变量,这里的 impl 是一个引用类型,定义上来看是一个 char 类型的返回变量,需要在使用时重新转换为数据库地址。

## 5.2 系统界面的搭建与登录的实现

### 5.2.1 系统界面的设计与实现

系统界面搭建,目的在于设计一个简介实用的界面,用户根据不同的身份登录系统,系统根据输入的登录名来确认对方用户等级,从而跳入不同的用户管理界面,具体实现代码如下:

```

    public boolean getadminListSplitBySearch(PageBean page,adminSearch
    adminSearch){
        return impl.getadminListSplitBySearch(page,adminSearch);
    }
    public ArrayList getadminList( admin admin ){
        return impl.getadminList( admin );
    }
    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);
    }
    public int getMaxadminIdByadminname(String adminname) {

```



```

        op.setTrueName(rst.getString("trueName"));
        op.setUseFlag(rst.getInt("use_flag"));
    }
}

```

以上代码包括在 Login\_manager 包中，分别从 Java.swing 中导入菜单，菜单项与框架三个包，来搭建系统界面，界面按钮的新建使用 JMenuBar()来实现，具体形式是 JMenuBar jMenuBar1 = new JMenuBar();需要说明 MenuBar 和 MenuItem 的区别，我们知道菜单中包括的东西是很多的，不一定只有按钮，还有类似于下拉菜单，选项等项目，这些不在按钮中的选项都算作 MenuItem，此外，MenuBar 在动作添加方面只能选择点击，而 MenuItem 则可以进行多种动作的选择和概括，在为 MenuItem 添加 Label 时，可以根据界面效果进行统一处理，比如添加统一的前缀，返回统一目录等等，但对于 MenuBar 则需要逐个操作，因为 Bar 相对于 Item 是独立存在的，不能因为其相关性而联系。图 5-1 为成绩管理系统登陆界面展示。



图 5-1 成绩管理系统登陆界面展示

### 5.2.2 登录功能实现

在实现登录功能时，具体的程序模块划分为 Login()和 LoginMgr(), Login()主要用于设定登陆功能中的主要接口，而 LoginMgr()负责对其进行实现，需要注意的是 Login()中 this 返回值的使用，需要标注明确 this 返回的是什么对象，以

便以后调用，另外还有一点，当需要在登录过程中访问数据库，我们使用的 `DBAccess.getDBAccess()` 方法，通常 `getDBAccess()` 返回的是一个地址，也就是这个函数访问的数据库信息表的具体位置，但是比如像姓名这种信息，在多个数据库表格中均有包括，取地址时并不能很好的执行，还需要对应访问密码和具体表格信息，这里就需要 `DBAccess.uniDBAccess()` 的使用，这个函数返回数据所在表格的具体信息，来得到一个唯一的访问量，使用 `DBAccess.uniDBAccess()` 可以很好地避免重复信息提取错误的问题。

以下是登陆功能的具体实现代码，图 5-2 为用户登录后主界面展示，需要说明的是三个等级的用户登陆效果相同，只是登录名和登录类别处有不同，在这里不再对每个等级的登陆效果作展示。

```
public Operator login(String adminname,String password)
{
    Operator op=null;
    StringBuffer sql=new StringBuffer("");
    //执行查询
    sql.append("select * from admins where adminname =")
        .append(adminname)
        .append(" and password=")
        .append(password)
        .append("");
    System.out.println(">>>>>>>>>>>>>>> sql:" + sql.toString());
    DBAccess dao = DBAccess.getDBAccess();
    ResultSet rst= dao.query(sql.toString());
    append(",score")
    .append(",examdate")
    .append(",status")
    .append(",use_flag")
    .append(")")
    .append(" values");
    sqlBufValues
    .append("'" + scores.getStudentid())
    .append(", " + scores.getCourseid())
    .append(", '" + scores.getExamtype() + "'")
    .append(", " + scores.getScore())
    .append(", '"+scores.getExamdate()+"'")
    .append(", " + scores.getStatus())
    .append(", " + scores.getUse_flag())
    try{
```

```

        while(rst.next()){
            op = new Operator();
            op.setId(rst.getInt("id"));
            op.setadminname(adminname);
            op.setPassword(password);
            op.setadminType(rst.getString("admintype"));
            op.setTrueName(rst.getString("true name"));
            op.setUseFlag(rst.getInt("use_flag"));
        }
    }
    catch(Exception ex)
    {
        op=null;
        ex.printStackTrace();
    }
    finally
    {
        dao=null;
    }
    return op;
}
}

```

以上代码解释了登录过程中的数据库访问问题,即通过什么方法来访问数据库。我们使用 `DBAccess.getDBAccess()`方法,是因为通常 `getDBAccess()`返回的是一个地址,也就是这个函数访问的数据库信息表的具体位置,但是比如像姓名这种信息,在多个数据库表格中均有包括,取地址时并不能很好的执行,还需要对应访问密码和具体表格信息,这里就需要 `DBAccess.uniDBAccess()`的使用, `DBAccess.uniDBAccess()`是一个基于 `DataBaseAPI` 下的方法,用于返回一个进入数据库的变量,通过这个可以进入数据库的变量来传递信息,这个函数返回数据所在表格的具体信息,来得到一个唯一的访问量,使用 `DBAccess.uniDBAccess()`可以很好地避免重复信息提取错误的问题。

以上代码中还有一个比较突出的问题,那就是 `op` 的使用,通常在异常处理机制中我们会使用 `op` 来传递异常信息,然后捕获 `op`,通过 `ex.printStackTrace` 来分析和解决异常,而以上代码中我们添加 `dao` 这个表达变量,在 `finally{}`中我们用 `dao` 来确认 `op` 的信息然后向异常处理机制传递 `dao`,不在传递 `op`,来统一异常处理,使得每一个部分的代码最终都返回一个 `dao` 形式的异常处理问题,统一



解决，省去了很多麻烦。

图 5-2 为用户登录主界面效果展示。



图 5-2 用户登录主界面展示

### 5.3 学生功能实现

学生功能包括验证个人信息，登录系统；查询个人基本信息：学号，班级，课程表等；查看成绩；查询自己的所有成绩，并得到平均分，总分等指标；查看某一科目的年级平均成绩，或班级平均成绩；查询自己的绩点等相关指标；打印和下载自己的成绩以及退出系统。实现以上功能的具体划分需要借鉴数据库设计的一部分思想，然后在设计的时候充分运用封装的思想，按照模块的划分来实现代码。

通过 `sqlBuf.append()` 来实现信息提取时，要注意 `append()` 这个函数的定义，`append` 在 `JDK` 中的定义中有 `toStringCache=null`，这是来源于 `Java` 源码中 `cache` 的定义问题，许多人会觉得 `cache` 不会为 `null`，但信息提取时如果 `String` 无意义，并不代表不进行提取，只不过不提取相应字符串的功能，也就是 `toStringCache` 值为 `null`，但这个字符串还是需要保存的。再添加数据库信息时，使用 `StringBuffer sqlBufValues = new StringBuffer` 实现，`sqlBufValues` 是我们新添加的数据信息，同样的道理，还是需要调用 `append` 功能，比如在添加学生学号这一信息时，就可以用 `.append(“”+ student.getCode(学生学号)+“”)` 来实现。在获取

到的信息与数据库信息进行匹配时，我们用获取信息 = dao.query(sqlBuf.toString())来进行匹配校验，其他的具体实现过程，比如查询，删除，更改，还有信息的综合处理等，在这里不做一一赘述。以下为具体实现代码：

```

package cn.system.manage.student;
import java.util.ArrayList;
import cn.system.manage.tools.PageBean;
public class StudentMgr {
    public StudentMgr(){
        impl = new StudentMgrImpl();
    }
    append(",student")
    .append(",exam")
    .append(",class")
    .append(",sex")
    .append(")")
    .append(" address(");
    sqlBufValues
    .append("'" + scores.getStudentid())
    .append(", " + scores.getCourseid())
    public boolean existedStudent( Student student ) {
        return impl.existedStudent( student );
    }
    op = new Operator();
        op.setId(rst.getInt("id"));
        op.setadminname(adminname);
        op.setPassword(password);
        op.setadminType(rst.getString("adminclass"));
        op.setTrueName(rst.getString("adminname"));
        op.setUseFlag(rst.getInt("use_flag"));

    public boolean addStudent( Student student ){
        return impl.addStudent( student );
    }
    finally {
        dao = null;
    }
    return existed;
    }

    return impl.getStudentListSplitBySearch(page,studentSearch);}
    public ArrayList getStudentList( Student student ){

```

```
return impl.getStudentList( student );  
}
```

以上接口代码中可以发现 `impl.getStudentListSplitBySearch(page,studentSearch)` 这个返回值，由于代码的层叠使用，使得这个返回值变得异常复杂且不利于理解，这是因为首先当 `page,studentSearch` 返回一个地址时，我们的目的已经达到了，但返回的地址不能够用于系统识别和更改，我们需要将这个地址通过一个方法来解析，即 `.getStudentListSplitBySearch`，通过 `.getStudentListSplitBySearch` 的调用，首先在格式上统一了返回 `char` 类型，便于系统的修改，同时由于 `impl` 类的限定，我们使用统一标识的变量便于内部的访问和更改，所以这么做。

同时在 `getStudentList( student )` 中也是使用了这个思想，取出的是一个 `char` 类型变量，但这里有一点不同，那就是 `getStudentList` 虽然取出的仍然是 `char` 类型的变量，却是被 `List` 类限定过的，即取出的 `char` 是有格式的，这是因为在 `student` 数据库中所存储的信息较多，而且联系较为紧密，使用 `getStudentList` 可以很好地将这些信息进行约束和管理，同时由于学生数目较多，而且各方面信息差别比较大，所以使用一个 `List` 来控制变量存储的格式和大小，可以很好地节省程序运行时间和成本，也方便工作人员统计，还有一点那就是使用 `getStudentList` 可以实现信息的分解保存，信息查看的时候也可以做到分节浏览，便于我们的工作人员保护学生信息。

图 5-3 与 5-4 为部分学生功能的实现效果展示。

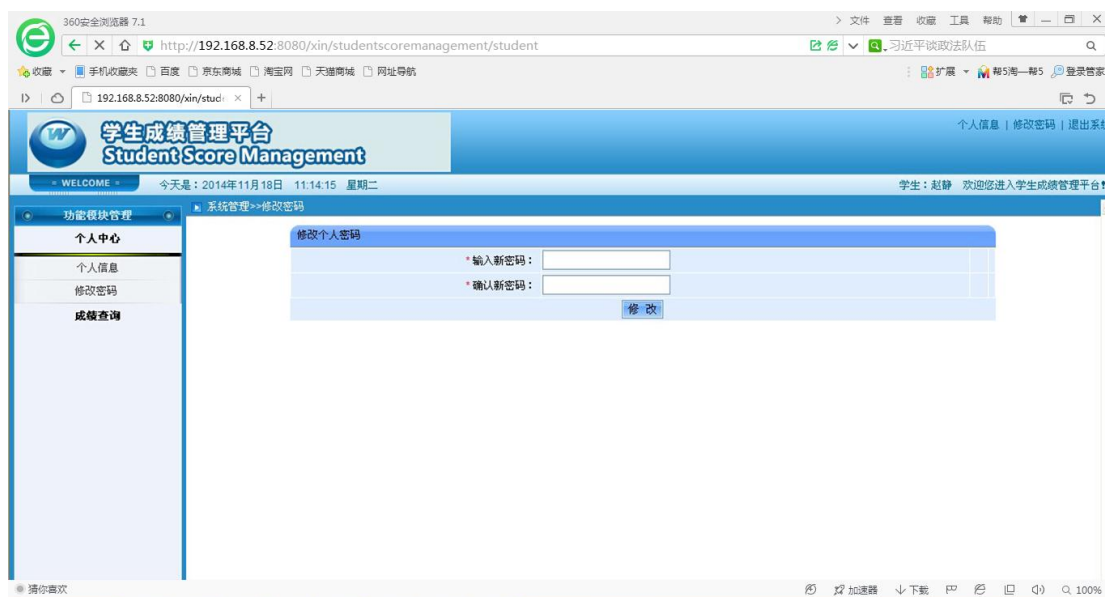


图 5-3 学生信息查询效果展示



图 5-4 单科成绩查询界面展示

## 5.4 教师功能实现

教师页面中我们可以进行有关成绩的一系列操作，包括成绩录入，修改，查询等，除此之外，教师还可以查询学生相关信息，查询班级，所教课程信息等，从功能的重要性和实现功能的主次区分来看，我们将这些功能分为成绩功能和其他信息查询功能两个部分。

### 5.4.1 成绩功能实现

在有关成绩处理的一系列功能，包括确定范围的成绩查询，成绩录入和更改等。以下为具体实现代码。

```
package cn.system.manage.student;
import java.util.ArrayList;
import cn.system.manage.tools.PageBean;
public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
    return impl.getadminListSplitBySearch(page,adminSearch);
}
public ArrayList getadminList( admin admin ){
    return impl.getadminList( admin );
}
```

```

    }
    .append("coursecode")
    .append(",courseName")
    .append(",coursescore")
    .append(",use_flag")
    .append(",teacherName")
    public class StudentMgr {
    public StudentMgr(){
    impl = new StudentMgrImpl();
    }
    public boolean existedStudent( Student student ) {
    return impl.existedStudent( student );
    }
    public boolean addStudent( Student student ){
    return impl.addStudent( student );
    }
    finally {
    dao = null;
    }
    return existed;
    }

    return impl.getStudentListSplitBySearch(page,studentSearch);}
    public ArrayList getStudentList( Student student ){
    return impl.getStudentList( student );
    }
}

```

成绩查询功能中，查询和匹配功能与学生信息查询和匹配相同，不同的是如何去实现确定范围的成绩查询，比如某个班的成绩，又比如某一分值到某一分值之间成绩的查询，这里代码实现时我们主要考虑使用容器的思想，即不同范围仅仅是不同的容器定义，所在容器内部的成绩则是我们需要输出的结果，具体实现时我们使用类 `ArrayList` `getScoresListBySearch(ScoresSearch scoresSearch)` 来实现，`scoresSearch` 即为我们确定的成绩容器范围输出，查询结果分页显示时，使用 `pageBean` 函数实现，程序输出一个 `pageBean.getPageSize()`，即所显示信息的大小，然后通过这个 `getPageSize()` 所得到的数据（比如 1-15），和我们一页之间显示的数据量的比较，再去设定页面数与页面号，页面号由函数 `pageBean.getPageNo()` 来设定。成绩录入所使用的仍然是 `append()` 函数，使用方法为 `append(""+添加信息+"")`，添加完成后使用 `this.updateData()` 来实现数据的更新。至于成绩修改和删除具体思想都相似，每次关于成绩操作如果需要进

行批量操作，需要引进 `scoresIDs[]` 量的概念，即成绩的序号，通过引进成绩序号这一概念来实现成绩的统计信息处理，从而帮助教师更广泛和全面的获取成绩信息，除此之外 `scoresIDs[]` 的引入也方便我们批量管理成绩，便于我们批量添加或删除成绩。

图 5-5 为成绩查询界面展示效果图。



图 5-5 成绩查询界面展示

在成绩录入，修改和删除部分由于涉及到对现有数据的更改，使用类 `boolean editScores( Scores scores)` 来实现，`Scores` 是一个已经存在的变量类型，要注意通常在 `Scores` 的累加过程中会出现溢出，这是因为 `Scores` 定义时容量为 `char(4)`，解决方法是设置一个 `StringBuffer(100)`，来首先完成分数的累加，然后在变量转换过程中完成 `int to char` 的转换，从而得到一个符合大小的分数，存入数据库中，另外在 `.append()` 方法的调用中，需要注意字段的顺序和分别，比如分数概念中就十分容易混淆单科分数和总分数等信息，需要我们留意并加上相应的区别符号，还有 `.append` 的顺序直接决定了数据存储和读取的顺序是否正确，在这里同样采用 `List` 的思想，首先将这几个 `.append` 字段作为变量保存在一个 `List` 中，然后根据这个 `List` 来统一生成，这样即加快了工作进度，又保证了正确性。以下为成绩录入修改和删除的实现部分的关键代码：

```
package cn.system.manage.admin;
import java.util.ArrayList;
import cn.system.manage.tools.PageBean;
```

```

public class adminMgr {
    /**
     * adminMgrImpl
     */
    private adminMgrImpl impl;
    public adminMgr(){
        /**
         * Constructor
         */
        impl = new adminMgrImpl();
    }
    public boolean existedadmin( admin admin ) {
        return impl.existedadmin( admin );
    }
    /**
     * Get Method: get all objects split by search
     * @param
     * @return
     */
    }
    public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
        return impl.getadminListSplitBySearch(page,adminSearch);
    }
    public ArrayList getadminList( admin admin ){
        return impl.getadminList( admin );
    }
    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);
    }
    public int getMaxadminIdByadminname(String adminname) {
        return impl.getMaxadminIdByadminname(adminname);
    }
    }
    /**
     * @param scores 记录的值对象
     public Operator login(String adminname,String password)
    {
        Operator op=null;
        StringBuffer sql=new StringBuffer("");
        //执行查询
        sql.append("select * from admins where adminname =")
        .append(adminname)
        .append(" and password=")

```







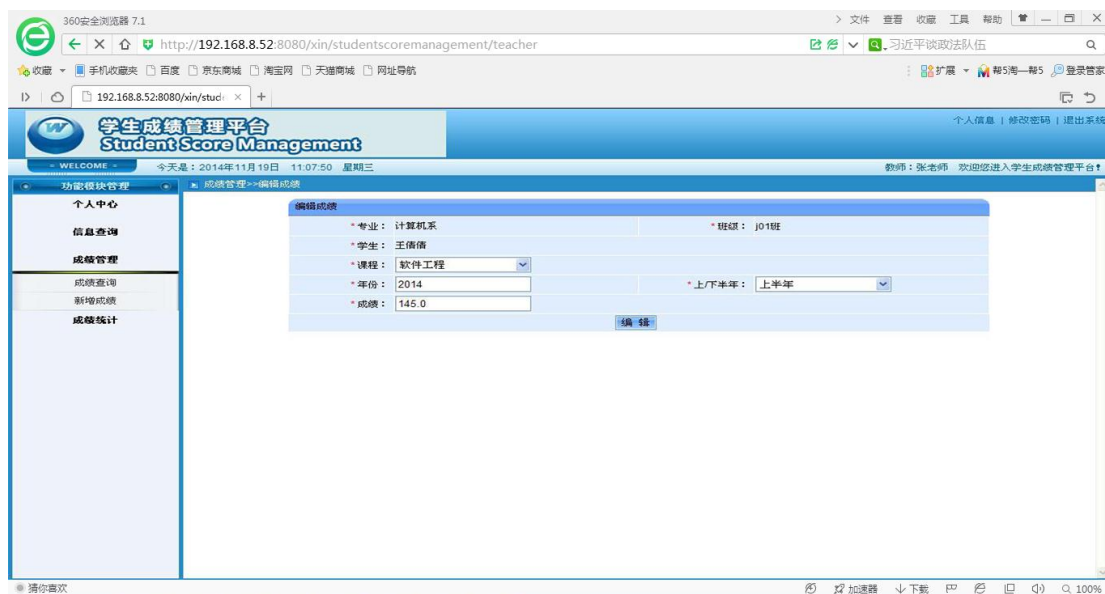


图 5-7 成绩修改界面展示

### 5.4.2 其他信息查询

这一部分功能包括课程和班级信息的查询，课程主要是该教师所教的课程信息，而班级信息包括自己所教课程的上课班级，和自己录入成绩的有关班级。具体实现关键代码如下：

```
package cn.system.manage.admin;
import java.util.ArrayList;
import cn.system.manage.tools.PageBean;
public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
    return impl.getadminListSplitBySearch(page,adminSearch);
}
public ArrayList getadminList( admin admin ){
    return impl.getadminList( admin );
}
public boolean getUserListSplit(PageBean page){
    return impl.getUserListSplit(page);
}
public int getMaxadminIdByadminname(String adminname) {
    return impl.getMaxadminIdByadminname(adminname);
}
}
public class adminMgr {
    private adminMgrImpl impl;
```

```

    public adminMgr(){
        impl = new adminMgrImpl();
    public boolean existedadmin( admin admin ){
        return impl.existedadmin( admin );
    }
    public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
/**
 * Constructor
 */

    return impl.getadminListSplitBySearch(page,adminSearch);
}
    public ArrayList getadminList( admin admin ){
        return impl.getadminList( admin );
    }
/**
 * adminMgrImpl
 */

    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);
    }
    public int getMaxadminIdByadminname(String adminname) {
        return impl.getMaxadminIdByadminname(adminname);
    }
}
/**
 * @param
 * @return
 */

public class adminMgr {
private adminMgrImpl impl;
public adminMgr(){
    impl = new adminMgrImpl();
}
    public boolean existedadmin( admin admin ){
        return impl.existedadmin( admin );
    }
    public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
        return impl.getadminListSplitBySearch(page,adminSearch);
    }
    public ArrayList getadminList( admin admin ){

```

```

        return impl.getadminList( admin );
    }
    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);
    }
    public int getMaxadminIdByadminname(String adminname) {
        return impl.getMaxadminIdByadminname(adminname);
    }
}
public class adminMgr {
    /**
     * adminMgrImpl
     */
    private adminMgrImpl impl;
    /**
     * Constructor
     */
    public adminMgr(){
        impl = new adminMgrImpl();
    }
    append(",score")
    .append(",examdate")
    .append(",status")
    .append(",use_flag")
    .append(")")
    .append(" values(");
    sqlBufValues
    .append(", " + scores.getCourseid())
    .append(", " + scores.getExamtype() + " ")
    .append(", " + scores.getScore())
    .append(", "+scores.getExamdate()+ " ")
    .append(", " + scores.getStatus())
    .append(", " + scores.getUse_flag())
    public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
        return impl.getadminListSplitBySearch(page,adminSearch);
    }
    public ArrayList getadminList( admin admin ){
        return impl.getadminList( admin );
    }
    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);
    }
}

```

```

/**
 * 得到最大的 admin id
 */
public int getMaxadminIdByadminname(String adminname) {
    return impl.getMaxadminIdByadminname(adminname);
}
}

```

通过 `sqlBuf.append()` 来实现信息提取时, 要注意 `append()` 这个函数的定义, `append` 在 JDK 中的定义中有 `toStringCache=null`, 这是来源于 Java 源码中 `cache` 的定义问题, 许多人会觉得 `cache` 不会为 `null`, 但信息提取时如果 `String` 无意义, 并不代表不进行提取, 只不过不提取相应字串的功能, 也就是 `toStringCache` 值为 `null`, 但这个字串还是需要保存的。再添加数据库信息时, 使用 `StringBuffer sqlBufValues = new StringBuffer` 实现, `sqlBufValues` 是我们新添加的数据信息, 同样的道理, 还是需要调用 `append` 功能, 比如在添加学生学号这一信息时, 就可以用 `.append(“”+ student.getCode(学生学号)+“”)` 来实现。

图 5-8 和 5-9 分别是教师页面下班级和课程信息查询结果展示。

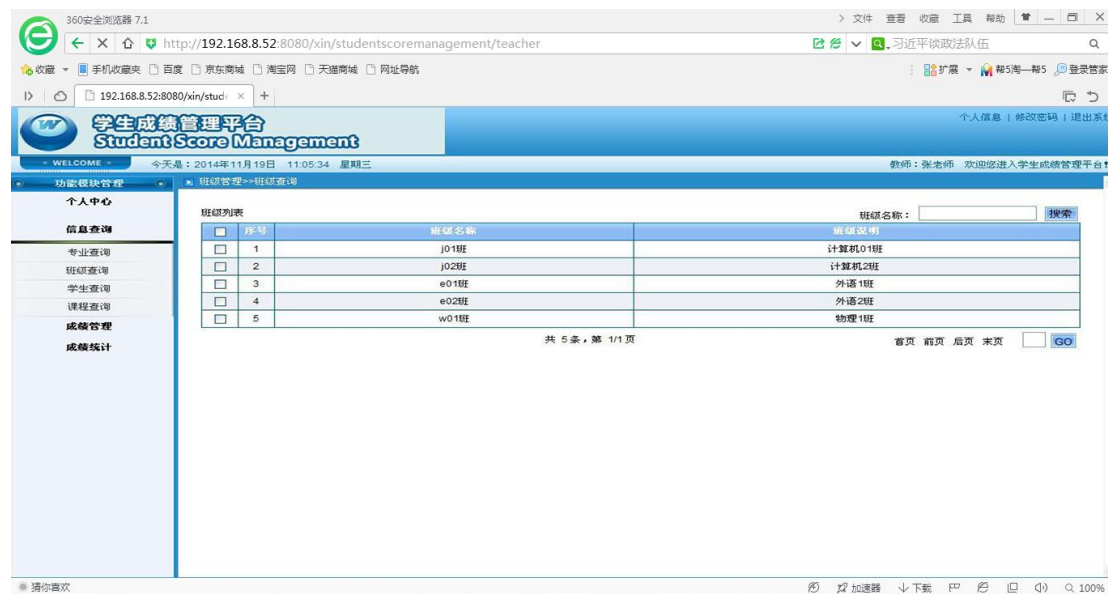


图 5-8 班级信息查询界面展示



图 5-9 课程信息查询界面展示

## 5.5 管理员功能实现

在系统功能与数据库分布设计中,管理员是一个独立于其他用户和功能的等级,如果说的更明确一些,就是管理员具有以上所描述的所有功能,在此基础上还可以对这些功能模块进行更改,并且在系统主页发布信息。

管理员属于超级用户,这也就意味着管理员身份的用户没有了权限的限制,反而需要加强对其身份的验证,所以在代码实现部分我们不设置信息的分层保密等功能,加大身份验证的程度,从而最大程度上保证管理员对系统的正确使用和系统不受到管理员特殊权限的影响。

由于实现功能的特殊性,管理员功能接口设计颇为复杂,需要再次引入 ArrayList 容器的概念,在功能的修改上,管理员具有独立于其他用户的功能,比如添加修改班级,专业和教师学生人员信息等,实现过程中 admin() 和 adminMgr() 分别实现功能接口和具体实现,以下为 admin() 具体实现的接口代码。

```
package cn.system.manage.admin;  
import java.util.ArrayList;  
import cn.system.manage.tools.PageBean;  
public class adminMgr {  
    /**  
     * adminMgrImpl  
     */  
}
```

```

public class StudentMgr {
public StudentMgr(){
    impl = new StudentMgrImpl();
}
public boolean existedStudent( Student student ) {
return impl.existedStudent( student );
}
public boolean addStudent( Student student ){
return impl.addStudent( student );
}
private adminMgrImpl impl;
/**
 * Constructor
 */
public adminMgr(){
    impl = new adminMgrImpl();
}
/**
 * check existed
 * @param
 * @return boolean
 */
public boolean existedadmin( admin admin ) {
    return impl.existedadmin( admin );
}
/**
 * Get Method: get all objects split by search
 * @param
 * @return
 */
public boolean getadminListSplitBySearch(PageBean page,adminSearch
adminSearch){
    return impl.getadminListSplitBySearch(page,adminSearch);
}
/**
 * Get Method: get all objects by condition
 * @param
 * @return
 */
public ArrayList getadminList( admin admin ){
    return impl.getadminList( admin );
}
/**
 * Get Method: get all objects split

```

```

    * @param
    * @return
    */
    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);
    }
    /**
    * 得到最大的 admin id
    */
    public int getMaxadminIdByadminname(String adminname) {
        return impl.getMaxadminIdByadminname(adminname);
    }
}

.....

```

在这一部分的接口代码中最重要的是信息存在性的验证,由于管理员身份的特殊性,我们要防止其他人冒充管理员的身份进入系统,对信息进行恶意的篡改,以 adminMgr 这一信息的存在性验证为例,在 impl = new adminMgrImpl()访问中,通过 bool checkexisted 来检验 adminMgrImpl 信息是否存在,然后将这个 bool 值传递到下一函数中进行比对,如果结果为 true,则进行身份验证,否则不予验证。其他部分的信息也是采用这一思想来完成。

在 adminMgr()中,由于管理员具有读取任意数据库资料的权限,为此需要引进 Java 异常与回收机制中的 SQL Exception,即数据库异常。此外在实现功能时,我们只为管理员设定 Insert,Update,Delete 三个功能,具体针对什么对象去执行 Insert,Update,Delete,或者说对什么数据去执行,我们都不在程序中实现,而是之间使用容器 ArrayList 的思想,为三个功能的对象设置容器,不关心每一个 Obj 的数据甚至格式,而是直接去处理。处理过程与上文中的所描述基本一致,即采用先录入后查看的方式,对每一个数据进行相对应的比对,然后得到结果,每一个结果与结果之间都是相联系的,以保证数据格式一致,查看的时候方便人员操作,更方便将来数据的修改和删除,这种思想将大大减轻系统后期的维护代价和升级费用。以下为 adminMgr()具体功能实现关键代码:

```

package cn.system.manage.admin;
import java.util.ArrayList;
import cn.system.manage.tools.PageBean;
public class adminMgr {
    private adminMgrImpl impl;

```

```

public adminMgr(){
    impl = new adminMgrImpl();
}
public boolean existedadmin( admin admin ){
    return impl.existedadmin( admin );
}
public ArrayList getadminList( admin admin ){
    return impl.getadminList( admin );
}
public boolean getUserListSplit(PageBean page){
    return impl.getUserListSplit(page);
}
public int getMaxadminIdByadminname(String adminname) {
    return impl.getMaxadminIdByadminname(adminname);
}
public class adminMgr {
    private adminMgrImpl impl;
    public adminMgr(){
        impl = new adminMgrImpl();
    }
    public boolean existedadmin( admin admin ){
        return impl.existedadmin( admin );
    }
    public boolean getUserListSplit(PageBean page){
        return impl.getUserListSplit(page);    }}

```

图 5-10 与 5-11 分别显示了管理员修改班级，修改课程和修改专业信息的界面展示。

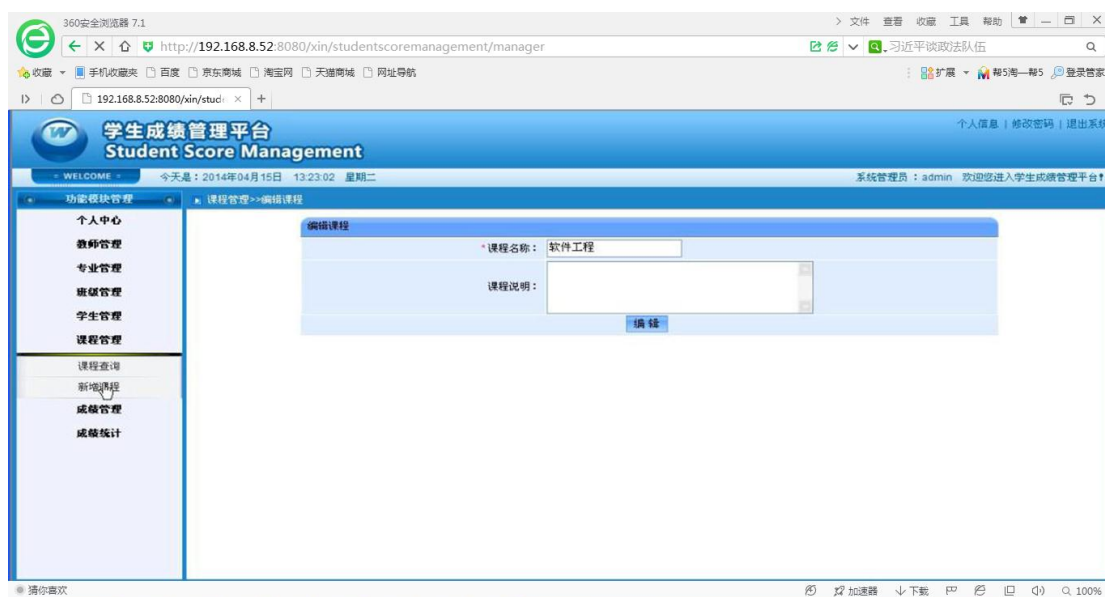


图 5-10 管理员修改班级界面展示 1



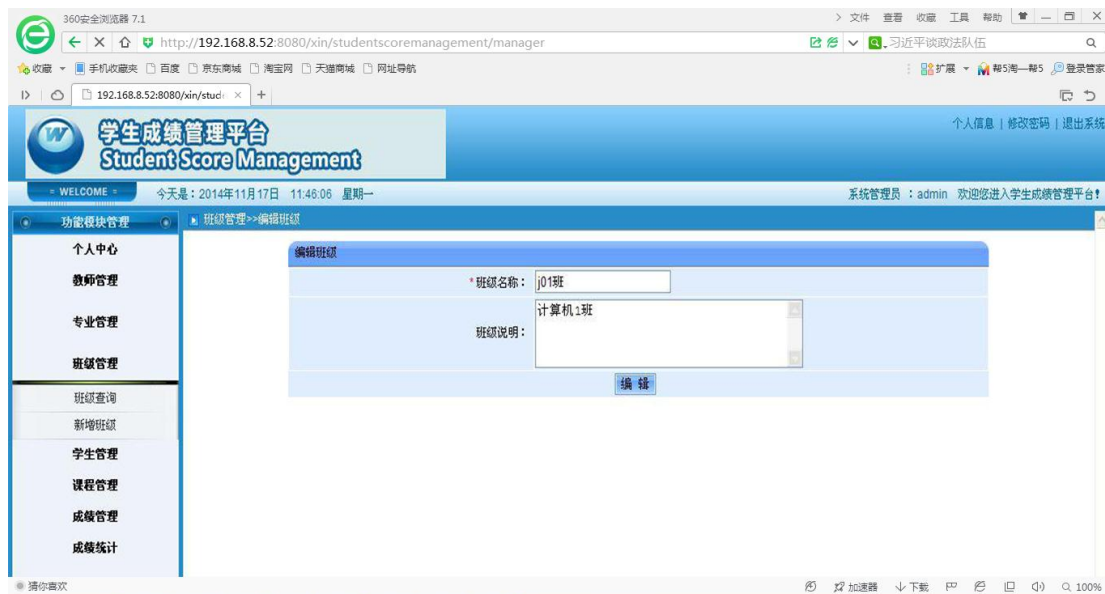


图 5-11 管理员修改课程界面展示 2

## 5.6 本章小结

本章是本此研究中最重要的一章，具体对从连接数据库到完成界面，再到学生，教师，管理员各部分功能模块的设计与实现。在学生功能中完成了本次研究的重点，即对成绩的管理与查询，而在教师功能模块中则实现了成绩的录入，删除和修改，最后在对管理员功能的设定中，我们明确了管理员特使的管理权限和功能。本章节中，对每一个功能的实现，我们都从关键代码和界面展示两个角度来进行说明，力求论文的严谨和完整，同时对部分关键代码从使用方法和使用思想的校对进行了说明。

## 第六章 总结与展望

### 6.1 工作总结

在本次研究所实现的基于 Java Web 的学生成绩信息管理系统过程中, 我们首先介绍了研究的现实意义, 研究背景与现状, 和研究的技术的发展等方面知识, 让本次开发过程具有实现的价值。

接下来我们说明了本次开发所使用的相关技术和基本原理知识, 其中着重介绍了开发语言和相关技术的使用, 以及开发设计模式和数据库连接技术, 最后介绍了开发工具, 这部分知识的介绍让开发人员对将要学习和探索的方向有了一个具象的认识, 也让开发从实现的技术意义上站得住脚。

接下来我们分析了开发系统的可行性与需求分析, 可行性方面, 我们着重从开发技术, 操作, 时间, 经济和开发法律等方面给出分析, 确定本次开发是可行的。然后做出系统的需求分析, 对系统的主要架构, 事务的处理流程, 和将要实现功能划分等方面作出详细说明。

最后, 我们用两章的篇幅, 对系统的总体设计, 数据库的分析与实现以及系统的详细设计作出说明, 并对相应功能界面做出了展示, 这一部分是论文的核心, 整个实现过程以及部分关键代码的使用, 我们都做了详细说明, 也分别对各种用户的不同功能界面做出了展示。

从背景知识到需求分析再到系统设计与代码实现, 再到部分功能测试与效果展示, 本次开发过程力求从每一个开发环节进行详细说明, 保证系统开发的完整与高效。

### 6.2 研究展望

本次研究还有着许多的不足之处, 比如没有考虑服务器的分级处理, 对庞大的学生登录现象没有做好准备; 用户信息采集不明确, 对学生的许多重要信息(包括民族, 住址等)都没在数据库中做好保存; 界面设计不够美观, 许多功能菜单

和信息显示界面处理的不简洁，整体界面效果不够人性化等等，在接下来的工作中，我们将会对以上不足做出改进，争取让开发的学生成绩信息管理系统逐步完善化。

## 参考文献

- [1]BruceEckel. Java 编程思想[M]. 机械工业出版社, 2003 年 10 月:23-322.
- [2]曾建潮. 软件工程[M].武汉理工大学出版社, 2008. 17-100
- [3]明日科技. Java 从入门到精通[M]. 清华大学出版社, 2012 年 9 月:1-256.\
- [4]张白一,崔尚森.面向对象程序设计——Java[M].西安: 电子科技大学出版社,2004.2-10
- [5]梁庆林. 三足鼎立各领风骚——动态网页技术 ASP、PHP、JSP 浅析[J]. 《科技信息（学术版）》, 2007, (24): 180-181
- [6]孙媛.JSP 是电子商务技术发展之根[J]. 《计算机软件与应用》, 2010, (7): 1-1
- [7]JOHNSON R, HOELLER J, ARENDSSEN A, et al. Java/J2EE application framework reference document. V1.1. 2004.
- [8]陈天河等.Struts,Hibernate,Spring 集成开发宝典[M].北京: 电子工业出版社,2007.39-72
- [9]马朝晖等.Java 教程.(第 6 版)[M].北京: 人民邮电出版社.2012.1-277
- [10]张萍生,何升编. SQL Server 数据库应用技术[M].清华大学出版社, 2005. 1-379
- [11]沈洁.数据库设计入门经典[M].清华大学出版社.2007 年 3 月.47-125
- [12]苏茂芳. 高职院校学生工作管理平台的设计与实现[D]. 湖南大学, 2013. 2-17
- [13]段艳超. 基于 UML 建模技术的学生成绩管理系统的设计与实现[J]. 硅谷, 2008, (7):
- [14]古乐, 史九林. 软件测试技术概论[M].清华大学出版社, 2009. 244-268
- [15]慈明亮. 基于互联网的教师成绩录入系统[J]. 《科技情报开发与经济》, 2013, (11): 166-167
- [16]张健. 网络考试系统结构分析[J]. 《软件导刊》, 2010, (11): 129-131
- [17]林建科. 基于知识地图的知识集成方法和系统研究[D].浙江大学, 2012. 2-50
- [18]何玉洁.数据库设计教程[M].机械工业出版社.2005 年 1 月.238-255
- [19]谷庆华等.基于 java 语言实现数据库访问[J]. 《计算机技术与发展》, 2008
- [20]洪承煜, 陈浩, 杨尚琴. 数据库统一接口模型设计与实现[J]. 《网络新媒体技

参考文献

术》, 2012, (3): 49-53

## 作者简介

作者姓名：杨宝林

工作单位：吉林省松原市地方税务局

邮政编码：138000

## 致 谢

很荣幸能够在吉林大学计算机科学与技术学院学习,能够在学院领导和诸位老师的帮助下攻读硕士学位,在吉林大学的这三年学习时光,将会成为我一生中最宝贵的财富。

首先要感谢我的导师张猛教授,张猛老师在本次的系统设计与实现工作中,给予了本人太多的帮助与指导。在设计之处,张猛老师仔细询问了开发工作的现实意义与准备情况,对可能发生的情况作出说明,并为开发工作提供了各方面的准备帮助,开发工程中,老师定期询问工作进展,对遇到的困难详细分析,对不懂的知识悉心解释,帮助我们排除了在开发过程中所遇到了一系列阻碍,最后,张猛老师又对系统的测试工作作出指导,对系统的使用给出自己的意见和建议,是我们在后期系统维护和完善工作中有的放矢,再次感谢张猛教授对本人的栽培与帮助!

其次,要感谢教研室其他老师和同学对我的帮助与照顾,每一次他们对开发工作的总结与建议都给了我巨大的启发,使我在工作中少走了许多弯路。一次良好的开发过程离不开一个强大的团队,可以说教研室的力量是本次开发工作完成的前提条件。

最后,对每一次帮助过我的领导,老师,同学和朋友再一次致以真诚的谢意!