|  |  |
| --- | --- |
| 总评成绩 |  |

数据库课程设计

说明书

学生成绩管理系统

学生姓名： 王浩 学号： 1707004716

学 院： 大数据学院

专 业： 计算机科学与技术

指导教师： 杨晓雯

2020年1月

目 录

1. 引言 …………………………………………………………………………… 1
   1. 编写目的 ……………………………………………………………… 1
   2. 项目背景及意义 ……………………………………………………… 1
2. 需求分析 ……………………………………………………………………… 2
   1. 功能需求分析 ………………………………………………………… 2
   2. 设计分析 ……………………………………………………………… 2
3. 应用环境 ……………………………………………………………………… 3
   1. 系统运行网络环境 …………………………………………………… 4
   2. 系统软硬件环境 ……………………………………………………… 4
4. 概念结构设计 ………………………………………………………………… 5
   1. 学生实体结构设计 …………………………………………………… 5
   2. 教师实体结构设计 ……………………………………………………… 5
   3. 管理员实体结构设计 …………………………………………………… 6
   4. 班级实体结构设计 ……………………………………………………… 6
   5. 课程实体结构设计 …………………………………………………… 6
   6. 系统整体结构设计 ……………………………………………………… 7
5. 逻辑结构设计 ……………………………………………………………… 8
   1. 学生逻辑结构设计 …………………………………………………… 8
   2. 教师逻辑结构设计 ……………………………………………………… 9
   3. 课程逻辑结构设计 …………………………………………………… 9
   4. 成绩逻辑结构设计 …………………………………………………… 10
   5. 管理员逻辑结构设计 ………………………………………………… 10
   6. 班级逻辑结构设计 …………………………………………………… 11
   7. 班级课程逻辑结构设计 ……………………………………………… 11
   8. 班级教师逻辑结构设计 ……………………………………………… 11
   9. 考试逻辑结构设计 …………………………………………………… 12
   10. 授课逻辑结构设计 …………………………………………………… 12
6. 物理结构设计 ……………………………………………………………… 13
   1. 存取方法的设计 ……………………………………………………… 13
7. 概要设计 …………………………………………………………………… 14
   1. 系统总体设计概述 …………………………………………………… 14
   2. 系统功能模块划分 …………………………………………………… 14
   3. 系统功能模块清单 …………………………………………………… 14
8. 详细设计 …………………………………………………………………… 17
   1. 数据库连接模块详细设计 …………………………………………… 17
   2. 查询个人信息功能模块详细设计 …………………………………… 18
   3. 修改个人信息功能模块详细设计 …………………………………… 18
   4. 查询成绩功能模块详细设计 ………………………………………… 18
   5. 查询考试功能模块详细设计 ………………………………………… 22
   6. 查询课程功能模块详细设计 ………………………………………… 23
9. 测试 ………………………………………………………………………… 25
   1. 功能测试 ……………………………………………………………… 25
      1. 修改信息功能测试 …………………………………………… 25
      2. 修改密码功能测试 …………………………………………… 26
      3. 查询个人成绩功能测试 ……………………………………… 29
      4. 修改考试信息功能测试 ……………………………………… 31
      5. 修改课程信息功能测试 ……………………………………… 32
      6. 数据库备份功能测试 ………………………………………… 34
10. 结论 ………………………………………………………………………… 35

参考文献 ………………………………………………………………………… 36

**1 引言**

**1.1 编写目的**

学生成绩管理一直以来都是学校教学管理中十分重要又相当复杂的管理工作之一，随着学校的规模不断扩大、学生数量的急剧增长，有关学生的各种信息也在增加，传统的手工处理已经无法满足当下教学和管理的要求。而计算机具有运算速度快、处理能力强等特点，非常适合应用到该领域上。

因此，为了保证流畅、安全、高效的存储大量学生的成绩信息，有必要设计一个学生成绩管理系统。这不但可以使教务人员从复杂的成绩管理中解脱出来，而且对于推动教学的发展也起到非常重要的作用。

**1.2 项目背景及意义**

随着我国教育改革的不断深入以及自学教育的不断扩招，参加考试的人越来越多,考试科目的多样化,使得学校对自学考试成绩的管理越趋繁琐、复杂，工作业务繁杂,工作量大,这种传统的学生管理模式已经暴露出种种弊端:难以统一调配和处理，效率极低，缺乏科学性以及合理性。

随着计算机应用的普及与深入，利用计算机能够对所有自学考试成绩进行统一管理，并进行分析，大大减少教学秘书的工作量,提高工作效率，为教学办公带来了极大的方便。使操作人员只需输入一些简单的汉字、数字,便可以存储、查找、修改、打印学生成绩信息等。学生成绩管系统的开发可实现学生成绩管理的系统化、规范化、自动化、达到提高学生成绩管理效率的目的,本着实用性、通用、开放和安全的原则 ,实现学生信息管理、课程信息管理、学生成绩管理、成绩查询等功能。本着一切从实际出发,充分考虑成绩的内部管理、信息交流等方面的复杂需求,实现成绩的有效管理,真正为学生成绩管理提供一个电子平台。

因此为了做好学生成绩管理工作,提高办事效率,实现全面的、相对集中的办公自动化,开发学生成绩管理系统就成了当务之急,其目的主要为了彻底改变这种繁杂的管理模式,把广大教师从繁重的成绩管理工作中解脱出来、把学校从传统的成绩管理模式中解放出来,实现全面的、相对集中的、智能化的信息综合管理,为学校的教学管理工作带来方便。

**2 需求分析**

**2.1 功能需求分析**

每当考试结束后，管理员开放考试成绩录入功能，教师录入考试成绩，学生可以查询自己的考试成绩。

学生成绩管理系统的用户有管理员，可以进行个人信息查询和修改，修改个人密码、开放考试成绩录入功能，录入、安排课程，安排、查询考试，管理学生成绩、学生、教师等操作；其次还有任课老师，可以查询、修改自己的个人信息，修改个人密码，查询考试信息，录入、查询、修改学生考试成绩；最后，学生也是不可或缺的用户，学生可以登录查询、修改个人信息，修改个人密码，查询各科成绩，查询考试，查询课程等。

**2.2 设计分析**

因为我们做的是学生成绩管理系统，所以数据库中有一个表存储学生各科成绩，有了学生成绩，就要有一个表存储学生信息，一个表存储课程信息，教师能录入学生成绩，所以要有一个教师表，管理员可以管理录入功能的开放与关闭，也可以管理学生教师以及课程的信息，所以要有一个管理员表，每个学生都有一个班级，所以要有一个表来存储班级信息，每个班级都有不同的课程，所以要建立一个班级课程表，每个班级也都有教师，所以要建立一个班级的教师表，教师授课的情况也要知道，所以要建立一个授课表。

综上所述，学生成绩管理系统有11个表：班级表class，班级课程表classcourse，班级教师表classteacher，课程表course，成绩表csgrade，考试表exam，成绩录入功能表gradesystem，管理员表manager，学生表student，教师表teacher，授课表teaching。

1. 班级表class：应包括班级号、班级人数、系别、专业。
2. 班级课程表classcourse：除了班级号、课程号必要字段，还应包括学年以及学期等。
3. 班级教师表classteacher：除了班级教师号、教师号必要字段，还应包括学年以及学期等。
4. 课程表course：应包括课程号、课程名、学时、学分和属性。
5. 成绩表csgrade：应包括学号、课程号、成绩。
6. 考试表exam：除了考试号、考试课程号必要字段，还应包括时间、地点等。
7. 成绩录入功能表gradesystem：用于管理成绩录入功能的开启和关闭，0为关闭，1为开启。
8. 管理员表manager：除了管理员号、姓名、性别、密码必要字段，还应包括年龄、电话、电子邮箱等。
9. 学生表student：除了学号、姓名、性别、班级、密码必要字段，还应包括年龄、电话、电子邮箱等。
10. 教师表teacher：除了教师号、姓名、性别、院系、密码必要字段，还应包括年龄、职称、学历、电话、电子邮箱等。
11. 授课表teaching：除了教师号、课程号、地点、时间、周次，还应包括学年、学期等。

**3 应用环境**

**3.1 系统运行网络环境**

网络结构图如图3-1所示，用户（学生、教师和管理员）可以通过网络登录学生成绩管理系统，进行相应的操作，应用服务器接收到相应操作做出相应响应之后对数据库进行相应操作，并返回进行展示。

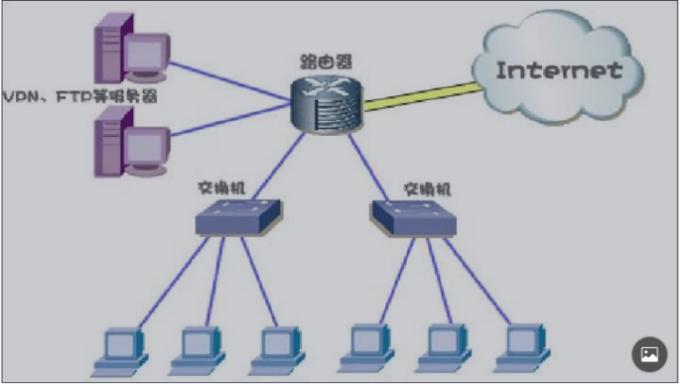


图3-1 系统网络结构图

**3.2 系统软硬件环境**

3.2.1 开发工具

服务器配置要求如表3.1所示。

表3.1 服务器配置

|  |  |
| --- | --- |
| 配置项 | 配置参数 |
| CPU 型号 | Intel(R)Core(TM)i5-7200U |
| CPU 主频 | 2.50GHz |
| 内存容量 | 16.0GB |
| 硬盘容量 | 256G + 448G |
| 操作系统 | Windows 10 专业版 |

系统运行软件环境如下表3.2所示。

表3.2 系统运行软件环境

|  |  |
| --- | --- |
| 环境项 | 环境参数 |
| 操作系统 | 具有Java虚拟机的操作系统均可 |
| 数据库 | MySQL |
| 数据库管理工具 | Navicat Premium 12、Navicat 12 for MySQL |
| 开发工具 | IntelliJ IDEA 2019.2、Eclipse |
| 第三方开发工具包 | mysql-connector-java-8.0.18.jar、jfreechart-1.0.19.jar、jcommon-1.0.23.jar、datepicker.jar、jdatepicker-1.3.4.jar、poi-3.9.jar、poi-ooxml-3.9.jar、poi-ooxml-schemas-3.9.jar |
| 图标设计工具 | Adobe PhotoShop CC 2019 |

**4 概念结构设计**

本章在需求分析的基础之上对学生成绩管理系统的数据库进行设计，设计原则是实体合理、联系精简、范式标准等。学生成绩管理系统数据库有5个实体，分别为学生、教师、管理员、班级和课程。各实体的属性以及实体E-R图如下所述。

**4.1 学生实体结构设计**

学生基本信息包括：学号、姓名、性别、年龄、班级、联系电话、电子邮箱、密码等，其中学号为主码。

学生实体的E-R图如图4-1所示。

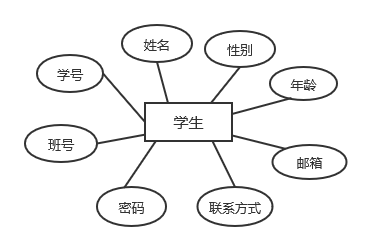


图4-1 学生实体E-R图

**4.2 教师实体结构设计**

教师基本信息包括：教师号、密码、姓名、性别、年龄、系别、学历、职称、职务、联系电话、电子邮箱等，其中教师号为主码。

教师实体的E-R图如图4-2所示。

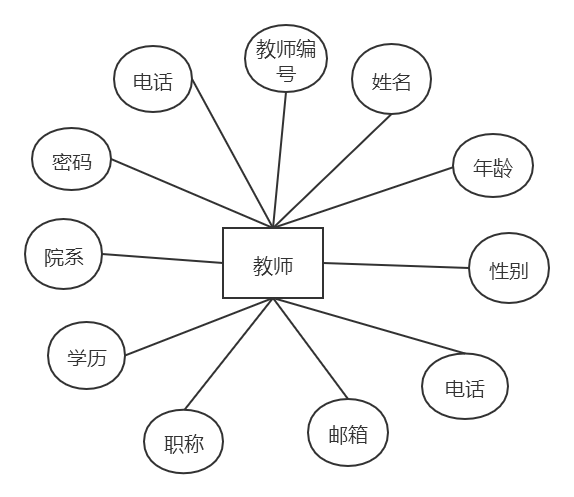


图4-2 教师实体E-R图

**4.3 管理员实体结构设计**

管理员基本信息包括：管理员号、姓名、性别、年龄、密码、联系电话、电子邮箱等，其中管理员号为主码。

管理员实体的E-R图如图4-3所示。

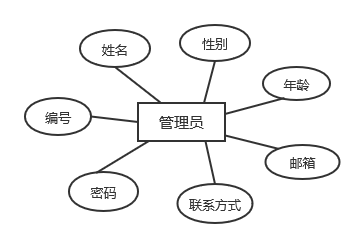


图4-3 管理员实体E-R图

**4.4 班级实体结构设计**

班级基本信息包括：班级号、班级人数、系别、专业等，其中班号为主码。

班级实体的E-R图如图4-4所示。

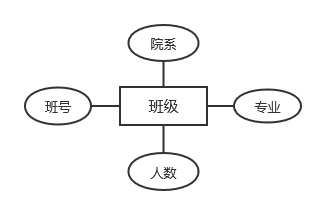


图4-4 班级实体E-R图

**4.5 课程实体结构设计**

课程基本信息包括：课程号、课程名、学时、学分、属性等，其中课程号为主码。

课程实体的E-R图如图4-5所示。

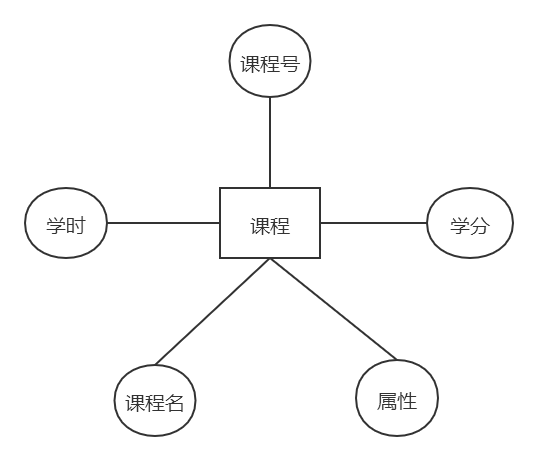


图4-5 课程实体E-R图

**4.6 系统整体结构设计**

系统的实体E-R图如图4-6和4-7所示。



图4-6 系统实体E-R图

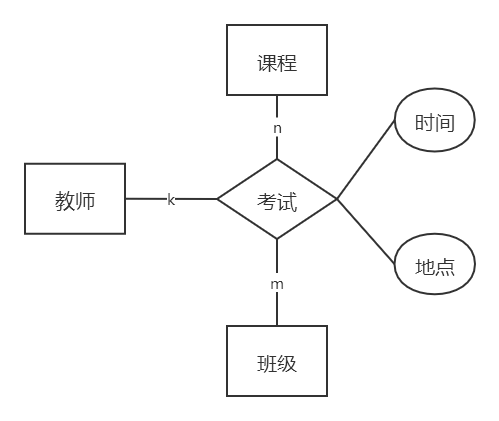


图4-7 系统实体E-R图

**5 逻辑结构设计**

根据E-R图向关系模型的转换原则，一个实体型转换为一个关系模式，实体的属性就是关系的属性。因此按照图3-6、3-7中所画的E-R图，本数据库中的表为：学生表（Student表）、教师表（Teacher表）、课程表（Course表）、成绩表（CSGrade表）、管理员表（Manager表）、班级表（Class表）、班级课程表（ClassCourse表）、班级教师表（ClassTeacher表）和教授课程表（Teaching表）。

根据实体型间联系的转换原则：

（1）一个1:1联系可以转-换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性转换为联系的属性，每个实体的码均是该关系的候选码。如果与某一端实体对应的关系模式合并，则需要在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。

（2）一个1:n联系可以转换为一个独立的关系模式，也可以与n端对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性转换为联系的属性，而关系的码为n端实体的码。

（3）一个m:n联系可以转换为一个关系模式，与该联系相连的各实体的码以及联系本身的属性转换为联系的属性，各实体的码组成关系的码或关系码的一部分。

综上所述，该数据库中的逻辑结构设计如下：

**5.1 学生逻辑结构设计**

学生的逻辑结构设计如表5-1所示。

表5-1 学生表（Student表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 学号 | Sno | varchar | 10 | No | 主码 |
| 姓名 | Sname | varchar | 10 | Yes |  |
| 性别 | Ssex | char | 2 | Yes | 男、女 |
| 年龄 | Sage | int | 2 | Yes |  |
| 班级 | Sclass | varchar | 10 | Yes |  |
| 密码 | Spassword | varchar | 10 | No |  |
| 联系电话 | Stel | varchar | 11 | Yes |  |
| 电子邮箱 | Semail | varchar | 20 | Yes |  |

**5.2 教师逻辑结构设计**

教师的逻辑结构设计如表5-2所示。

表5-2 教师表（Teacher表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 教师号 | Tno | varchar | 10 | No | 主码 |
| 姓名 | Tname | varchar | 10 | Yes |  |
| 性别 | Tsex | char | 2 | Yes | 男、女 |
| 年龄 | Tage | int | 2 | Yes |  |
| 系别 | Tdept | varchar | 20 | No |  |
| 学历 | Tdegree | varchar | 8 | Yes | 本科、硕士、博士、博士后 |
| 职称 | Ttitle | varchar | 4 | Yes | 助教、讲师、副教授、教授 |
| 职务 | Tduty | varchar | 20 | Yes |  |
| 联系电话 | Ttel | varvhar | 11 | No |  |
| 电子邮箱 | Temail | varchar | 20 | Yes |  |
| 密码 | Tpassword | varchar | 10 | No |  |

**5.3 课程逻辑结构设计**

课程的逻辑结构设计如表5-3所示。

表5-3 课程表（Course表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 课程号 | Cno | varchar | 10 | No | 主码 |
| 课程名 | Cname | varchar | 50 | No |  |
| 学时 | Cperiod | int | 2 | Yes |  |
| 学分 | Ccredit | double | 4 | Yes |  |
| 属性 | Cattribute | char | 4 | No | |  | | --- | | 必修、专业选修、公共选修 | |

**5.4 成绩逻辑结构设计**

成绩的逻辑结构设计如表5-4所示。

表5-4 成绩表（CSGrade表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 学号 | Sno | varchar | 10 | No | 主码 |
| 课程号 | Cno | varchar | 10 | No | 主码 |
| 成绩 | Grade | double | 4 | Yes |  |

**5.5 管理员逻辑结构设计**

管理员的逻辑结构设计如表5-5所示。

表5-5 管理员表（Manager表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 管理员号 | Mno | varchar | 10 | No | 主码 |
| 姓名 | Mname | varchar | 10 | No |  |
| 性别 | Msex | varchar | 2 | No | 男、女 |
| 年龄 | Mage | int | 2 | Yes |  |
| 密码 | Mpassword | varchar | 10 | Yes |  |
| 联系电话 | Mtel | varchar | 11 | No |  |
| 邮箱 | Memail | varchar | 25 | Yes |  |

**5.6 班级逻辑结构设计**

班级的逻辑结构设计如表5-6所示。

表5-6 班级表（Class表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 班级号 | CLno | varchar | 10 | No | 主码 |
| 班级人数 | CLnum | varchar | 11 | Yes |  |
| 学院 | CLdept | char | 50 | Yes |  |
| 专业 | CLmajor | char | 50 | Yes |  |

**5.7 班级课程逻辑结构设计**

班级课程的逻辑结构设计如表5-7所示。

表5-7 班级课程表（ClassCourse表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 班级号 | CLno | varchar | 20 | No | 主码 |
| 课程号 | Cno | varchar | 20 | No | 主码 |
| 学年 | Cyear | varchar | 20 | Yes |  |
| 学期 | Cterm | varchar | 20 | Yes |  |

**5.8 班级教师逻辑结构设计**

班级教师的逻辑结构设计如表5-8所示。

表5-8 班级教师表（ClassTeacher表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 班级号 | CLno | varchar | 20 | No | 主码 |
| 教师号 | Tno | varchar | 20 | No | 主码 |
| 学年 | Cyear | varchar | 20 | Yes |  |
| 学期 | Cterm | varchar | 20 | Yes |  |

**5.9 考试逻辑结构设计**

考试的逻辑结构设计如表5-9所示。

表5-9 考试表（Exam表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 班级号 | Eclno | varchar | 2 | No | 主码 |
| 课程号 | Ecno | varchar | 10 | No | 主码 |
| 考试地点 | Eposition | varchar | 10 | Yes |  |
| 开始时间 | Etime | Datetime | 6 | Yes |  |
| 结束时间 | Endtime | Datetime | 6 | Yes |  |
| 考试编号 | Eno | char | 20 | No | 主码 |

**5.10 授课逻辑结构设计**

教师的逻辑结构设计如表5-10所示。

表5-10 教授课程表（Teaching表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文含义 | 字段名称 | 数据类型 | 长度 | 是否为空 | 备注 |
| 教师号 | Tno | varchar | 10 | No | 主码 |
| 课程号 | Cno | varchar | 10 | No | 主码 |
| 上课地点 | Cposition | varchar | 20 | Yes |  |
| 上课周次 | Cweek | varchar | 20 | Yes |  |
| 上课时间1 | Ctime1 | varchar | 30 | Yes |  |
| 上课时间2 | Ctime2 | varchar | 30 | Yes |  |
| 学年 | Cyear | varchar | 20 | Yes |  |
| 学期 | Cterm | varchar | 20 | Yes |  |

**6 物理结构设计**

为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程，就是数据库的物理设计。确定数据库的物理结构，在关系数据库中主要指存取方法和存储结构。

**6.1 存取方法的设计**

常用的存取方法为索引方法和聚簇方法。B+树索引和hash索引是数据库中经典的存取方法，使用最普遍。

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 属性 | 索引类型 | 存取方法 |
| class表 | 班号 | NORMAL类型 | B+树索引 |
| classcourse表 | 班号、课程号 | NORMAL类型 | B+树索引 |
| Classteacher表 | 教师号、班号 | NORMAL类型 | B+树索引 |
| course表 | 学号、课程号 | NORMAL类型 | B+树索引 |
| csgrade表 | 课程号 | NORMAL类型 | B+树索引 |
| Manager表 | 管理员号 | NORMAL类型 | B+树索引 |
| student表 | 学号、班级号 | NORMAL类型 | B+树索引 |
| teacher表 | 教师号 | NORMAL类型 | B+树索引 |
| teaching表 | 教师号、课程号 | NORMAL类型 | B+树索引 |

**7 概要设计**

**7.1 系统总体设计概述**

经过需求分析，已经明确系统必须要“做什么”，接下来就要确定“怎么做”了。系统总体设计的一个很好的方法就是要按照系统要实现的功能进行模块划分，把联系紧密的功能实现划分到相同模块中，模块之间的划分遵循“高内聚、低耦合”的原则。

**7.2 系统功能模块划分**

该系统应该完成以下两个方面的内容:学生信息资料的管理和学生成绩的管理(包括必修课和选修课),每个内容均需要提供添加、修改和查询的功能。该系统涉及到的用户包括教师、学生和管理员，各类用户的权限不一样。因此，把学生成绩管理系统划分为登录模块、学生模块、教师模块、管理员模块四大模块。模块划分如图7-1所示。

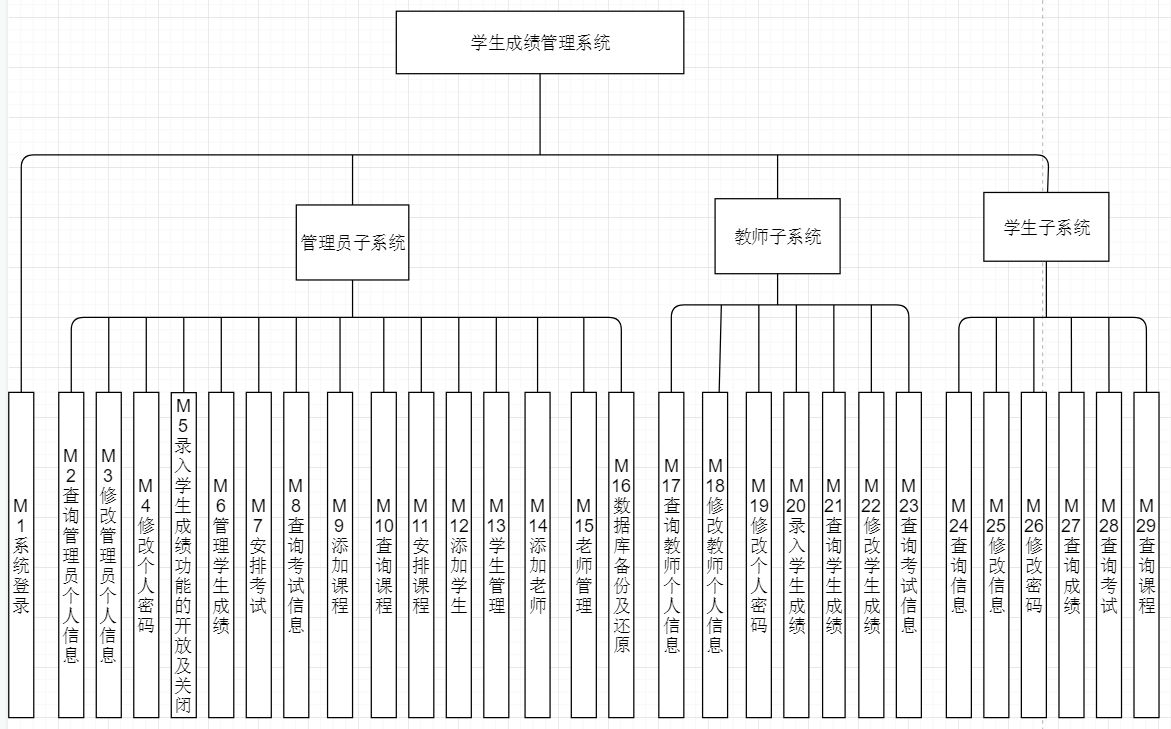


图7-1 系统总功能图

* 1. **系统功能模块清单**

对图7-1所示的系统各个功能模块进行细化：

* M1系统登录：用户根据自己的身份选择相应选项进行登录，进入对应身份的界面。
* M2个人信息查询: 管理员查看个人的信息
* M3个人信息修改：管理员修改个人信息。
* M4修改个人密码：管理员通过输入旧密码、输入新密码和再次输入新密码确认密码来修改自己的密码。
* M5教师录入学生成绩功能的开启和关闭：管理员开启成绩录入功能，教师才能录入成绩，管理员关闭此功能，教师不能再录入成绩。
* M6管理学生成绩：通过学年、学期、班级查询、修改、删除学生成绩，班级成绩情况可以用饼状图显示各科成绩平均分以及所占比重，也可通过学年、学期、学号来查询、修改、删除学生成绩，所查询到的学生成绩表可保存到本地文件夹。
* M7安排考试：管理员通过输入考试编号和考试地点、选择考试开始时间和结束时间、选择已存在的班级和已安排的课程来安排考试。
* M8查询考试：课程号为空时，管理员可通过学年、学期和班级号查询整个班全部课程的考试安排，课程号为已安排课程时，管理员通过学年、学期、班级和课程号查询班级的相应课程的考试安排。
* M9添加课程：管理员通过输入课程编号、课程名称、课程学时、课程学分和选择课程属性来添加课程。
* M10查询课程：管理员选择学院和专业来查询、修改、删除课程，学院、专业为空时查询所有课程，课程可分为已安排课程和未安排课程， 查询到的课程表可导出保存到本地。
* M11安排课程：管理员选择上课班级、学年、学期、科目和没有教课的老师、输入上课地点、上课周次、上课时间来安排课程。
* M12添加学生：管理员输入学生的信息包括学号、姓名、性别、年龄、班级、联系方式和邮箱来添加学生。
* M13学生管理：管理员可通过班级查询整个班级的学生信息，也可通过学号查询某个学生的信息，可修改选中的学生的学号、密码和班级，可删除选中的学生。
* M14添加老师：管理员输入老师的信息包括教师号、姓名、性别、年龄、学院、学历、职称、职务、联系方式和邮箱来添加老师。
* M15老师管理：管理员通过教师号查询老师信息，可修改、删除选中老师的信息，可初始化老师的密码为123456。
* M16数据库备份及还原：管理员进行数据库备份和还原。
* M17查询个人信息：老师查看个人信息。
* M18修改个人信息：老师修改个人信息。
* M19修改密码：老师通过输入旧密码、输入新密码和再次输入新密码确认密码来修改自己的密码。
* M20录入学生成绩：在管理员开启成绩录入功能后，老师选择班级、学生、科目、输入成绩来录入学生成绩。
* M21查询学生成绩：老师通过选择班级和科目来查询整个班级的学生的成绩，整个班级的成绩可以按升序或降序排序、可以求平均成绩，也可通过输入学号并选择课程来查询某个学生某课程的成绩，查询得到的成绩表可以导出到本地保存。
* M22修改学生成绩：老师可以通过输入学生学号和姓名、选择科目和输入成绩来修改学生的成绩。
* M23查询考试信息：老师通过选择科目来查询考试信息。
* M24查询信息：学生查看个人信息。
* M25修改信息：学生修改个人信息。
* M26修改密码：学生通过输入旧密码、输入新密码和再次输入新密码确认密码来修改自己的密码。
* M27查询成绩：学生通过选择学年、学期查询自己的成绩，成绩可以图表显示，查询得到的成绩表可以导出保存到本地。
* M28查询考试：学生可以查询所修课程的考试安排。
* M29查询课程：学生通过选择学年和学期查询课程。

**8 详细设计**

**8.1 数据库连接模块详细设计**

使用mysql-connector-java-8.0.12.jar文件加载连接MySQL数据库的JDBC-数据库驱动，并使用JDBC进行如下操作：

* 与本地数据库建立连接

private static Connection con = null;  
//数据库的url地址;characterEncoding=utf-8解决数据库中文乱码问题  
private static String url = "jdbc:mysql://localhost:3306/student?"  
 + "&useSSL=true&characterEncoding=utf-8&serverTimezone=GMT";  
//驱动的完整包名  
private static String driver = "com.mysql.cj.jdbc.Driver";   
//数据库的用户名  
private static String user = "root";  
//数据库的用户名密码  
private static String password = "212033";

static{  
 try{  
 Class.forName(driver); //加载JDBC-MySQL驱动  
 }catch (Exception e){  
 e.printStackTrace();  
 }try {  
 //获取数据库连接  
 con = DriverManager.getConnection(url, user, password);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
}

* 向已连接的数据库发送SQL语句

try{  
 PreParedStatement preparedsmt = con.prepareStatement(selectLnoSQL);  
 preparedsmt.setString(1, Tno);  
 preparedsmt.executeQuery(); //提交请求  
} catch (SQLException e) {  
 e.printStackTrace();  
}

* 处理SQL语句返回的结果

try{

//获取返回结果  
 ResultSet result = preparedsmt.executeQuery();  
} catch (SQLException e) {  
 e.printStackTrace();  
}

**8.2 查询个人信息功能模块详细设计**

8.2.1 SQL语句设计

* 根据学生学号查询学生个人信息

通过student表中的外码sclass（所属班级号）与class表（班级表）中的主码Clno（班号）条件连接两个表，并根据从登陆界面获取的学号获取到该学生的具体信息，其中包括学生的基本信息、所在院系和所在专业等信息。

select student.\* ,class.CLdept,class.CLmajor from student,class

where student.Sclass = class.CLno and

student.Sno=?;

8.2.2 StuInfoJPanel类设计

* StuInfoJPanel(String sno)方法

使用登陆界面传回的sno（学生学号）查询学生个人信息。并使用Java Swing中的JLabel标签显示每一个对应信息，并将这些JLabel添加到JPanel面板以构建整个个人信息界面。

8.2.3 界面效果



图8-1 查询个人信息界面

**8.3 修改个人信息功能模块详细设计**

8.3.1 SQL语句设计

* 根据学生学号更新学生信息

学生只能修改个人信息中的其他信息。其他必要信息，例如学号、院系、专业等信息需要管理员来进行修改，因此只需要通过update语句更新学生表（student表）,即可更改学生基本信息。

update student set Sname=?,Ssex=?,Sage=?,Stel=?,Semail=? where Sno=?;

8.3.2 StuChangeInfoJPanel类设计

* StuChangeInfoJPanel(String sno)方法

使用登陆界面传回的学生学号查询学生个人信息，并设置到每一个可修改的JTextField输入框上，供学生进行修改，但学号、院系、专业等信息需要管理员来进行修改，需设计这类数据框无法修改。并将获取到的新数据使用更新语句更新学生信息。

8.3.3 界面效果



图8-2 修改个人信息界面

**8.4 查询成绩功能模块详细设计**

8.4.1 SQL语句设计

* 根据学年、学期查询该学生所有科目成绩

通过连接student表（学生表）、course表（课程表）、csgrade表（成绩表）、classcourse（班课表）查询指定学年学期该学生所有科目的成绩，并使用distinct剔除重复的成绩信息。学生只能查看成绩，无法对成绩进行修改。

select distinct Student.Sno,Student.Sname,Course.Cno,Course.Cname,Csgrade.Grade,Ccredit from Student,Course,Csgrade,classcourse

where Student.Sno=Csgrade.Sno and

Csgrade.Cno = Course.Cno and Student.Sclass = classcourse.CLno and course.Cno = classcourse.Cno and Student.Sno = ? and

classcourse.Cyear=? and classcourse.Cterm=?;

8.4.2 GradeSearchJPanel类设计

* StuChangeInfoJPanel(String sno,JFrame main)构造方法

设置JcomboBox学年和学期下拉选择框供学生用户进行选择，使用JTable表格装载查询到的成绩数据，最终使用JscrollPane滑动窗格显示到面板上。

* setTableStyle(JTable table)方法

用于设置table表格的行宽、背景色、字体等一些属性。以保证数据可以更全面的展示给学生用户。界面效果：



图8-3 查询成绩界面

8.4.3 ChartJPanel类设计

使用JFreeChart第三方图表控件，使用柱状图的方式可观的展示当前学生本学期各科目成绩的大致情况，并以JPanel面板的方式添加到当前学生成绩窗口上。

功能演示界面：

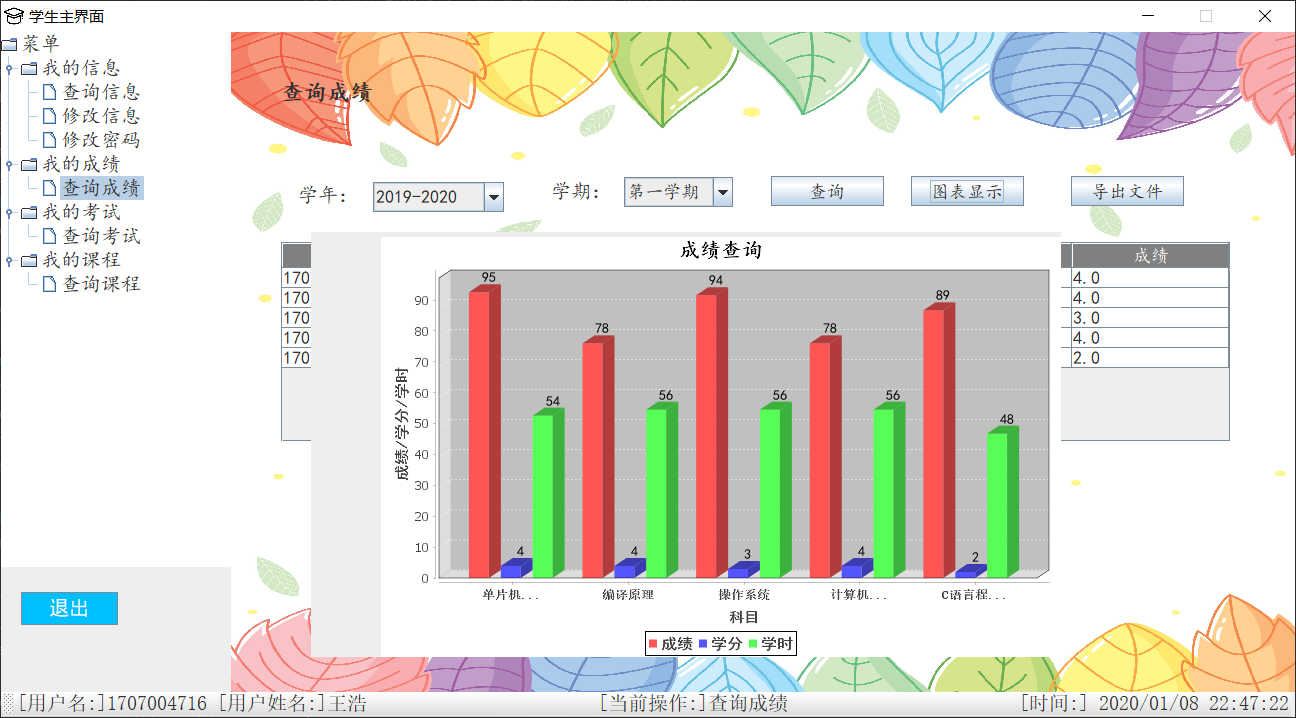


图8-4 查询成绩柱状图显示界面

8.4.4 FileUtil类设计

* TableToFile(JTable table)方法

使用poi-3.9.jar第三方包提供的excel写入方法，将查询得到的查询结果（JTable表格）写入到本地文件，保证查询成绩结果的持久性。并使用FileDialog提示用户输入文件保存位置，保证位置的可选性。

TableModel model = table.getModel();  
*//第一步 创建一个workbook对应一个excel文件*HSSFWorkbook workbook = **new** HSSFWorkbook();  
*//第一步 在workbook中创建一个sheet对应excel中的sheet*HSSFSheet sheet = workbook.createSheet();  
*//第三步 在sheet表中添加表头第0行*HSSFRow row = sheet.createRow(0);  
*//第四步 创建单元格，设置表头***for** (**int** i = 0; i < model.getColumnCount(); i++) {  
 HSSFCell cell = row.createCell(i);  
 cell.setCellValue(model.getColumnName(i));  
}  
*//第五步 写入实体数据***for** (**int** i = 0; i < model.getRowCount(); i++) {  
 HSSFRow row1 = sheet.createRow(i + 1);  
 **for** (**int** j = 0; j < model.getColumnCount(); j++) {

String data = model.getValueAt(i, j).toString()  
 row1.createCell(j).setCellValue(data);  
 }  
}

最终界面演示：



**8.5 查询考试功能模块设计**

8.5.1 SQL语句设计

* 通过学生学号查询该学生所在班级的考试

通过班号连接student表（学生表）和exam表（考试表），通过课程号连接exam表（考试表）和course表（课程表），使用学号查询该学生需要参与的考试。

Select exam.Eclno,exam.Ecno,course.Cname,exam.Eposition,exam.Etime,exam.Endtime

from student,exam,course

where student.Sclass = exam.Eclno and

exam.Ecno = course.Cno and

student.Sno=?;

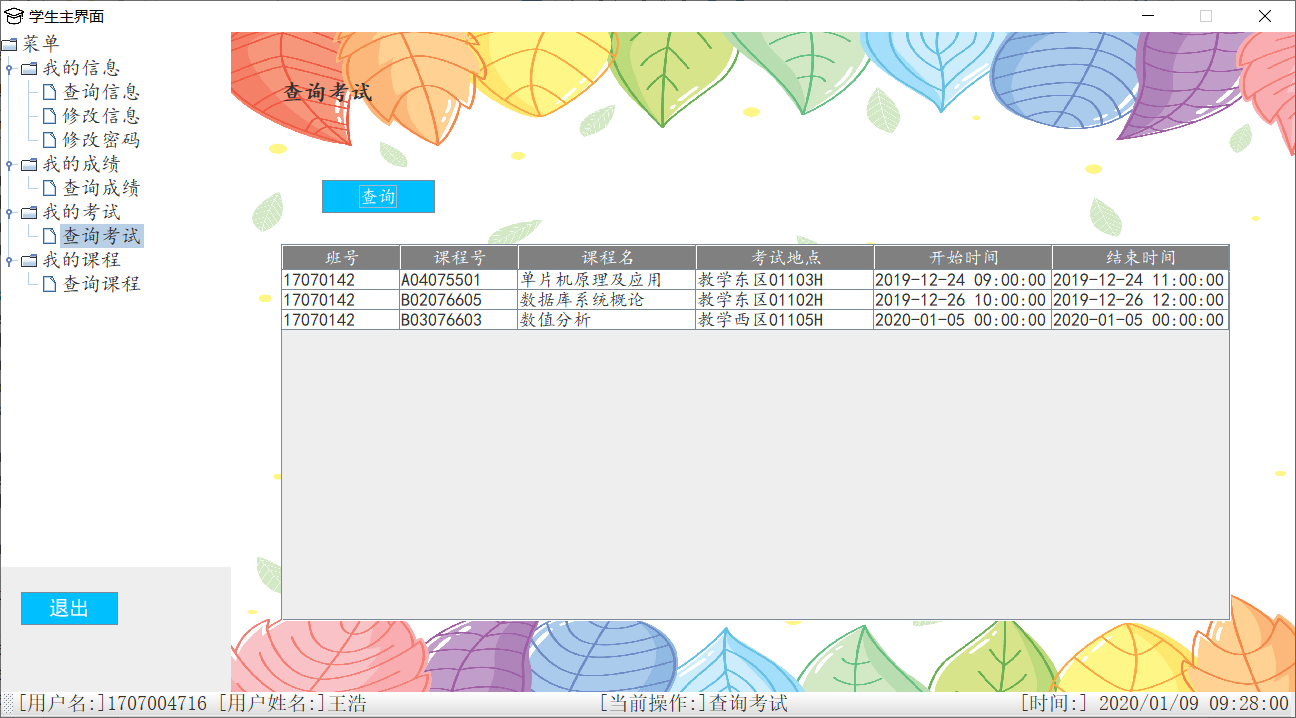
8.5.2 MyExamSearchJPanel类设计

使用二维数据data[][]存储查询得到的结果，使用JTable加载表格标题和查询结果，将JTable添加到JscrollPane当中加载到成绩查询面板上。

Object data[][] = ExamUtil.*getMyExam*(sno);  
Object name[] = {**"班号"**,**"课程号"**,**"课程名"**,**"考试地点"**,**"开始时间"**,**"结束时间"**};  
remove(**table**);  
**table** = **new** JTable(data,name);

JScrollPane jScrollPane = **new** JScrollPane(table);  
jScrollPane.setBounds(50, 212, 950, 377);  
jScrollPane.setViewportView(table);  
add(jScrollPane);  
validate();

最终界面演示：



**8.6 查询课程功能模块设计**

8.6.1 SQL语句设计

* 通过学号查询指定学年学期该学生所上课程

由于该课程必须是该学生所在班级中已经呗授课的课程，通过班号连接student表（学生表）和classcourse表（班级课程表）,确定该学生所在在班级的课程；通过课程号连接classcourse表（班级课程表）和teaching表（授课表）,确定该课程已经安排授课；通过课程号连接teaching表（授课表）和course表（课程表）,确定该课程是已有课程。具体的查询SQL语句如下：

select Cname,Cweek,Cposition,Ctime1,Ctime2,Tname,Cperiod,Ccredit,Cattribute

from student,classcourse,teaching,teacher,course

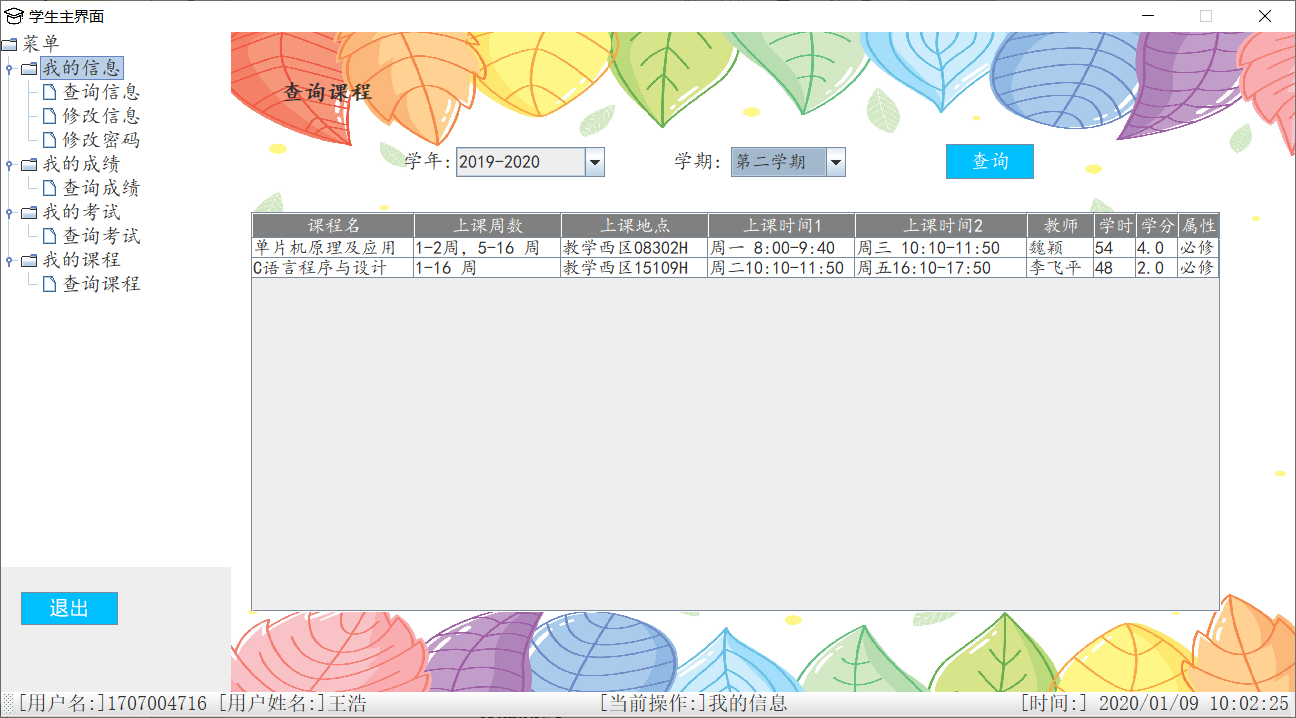
where Sno = ? and Sclass=CLno and classcourse.Cno=teaching.Cno

and teaching.Tno=teacher.Tno and teaching.Cno=course.Cno;

8.6.2 CourseSearchJPanel类设计

设置JcomboBox学年和学期下拉选择框供学生用户进行选择，使用Jtable表格装载查询到的成绩数据，并使用JscrollPane滑动窗格显示到面板上。

功能效果图：



**9 测试**

**9.1 功能测试**

9.1.1 修改信息功能测试

测试目的：学生用户能够修改个人信息，包括除班级、学号、院系、专业以外的其他信息。具体修改信息功能等价类表如下表9.1所示。

表9.1 修改信息功能测试等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入等价类** | **测试用例** | **测试数据** | **期望结果** | **实际结果** |
| 无效等价类 | 可修改信息未作改动 | 原信息：王浩（姓名等）  改后信息：王浩（姓名等） | 无效输入 | 提示未做任何修改 |
| 有效等价类 | 可修改信息有作改动 | 原信息：20（年龄等）  改后信息：21（年龄等） | 有效输入 | 提示修改成功 |

**用例 1：**

* 测试条件：原信息未作任何修改
* 测试用户：学生
* 测试步骤：在修改信息面板，对可修改信息不做任何修改，点击修改按钮
* 预期结果：系统提示未作任何修改
* 实际结果：系统提示没有做作任何修改

具体结果图如图9-1所示。

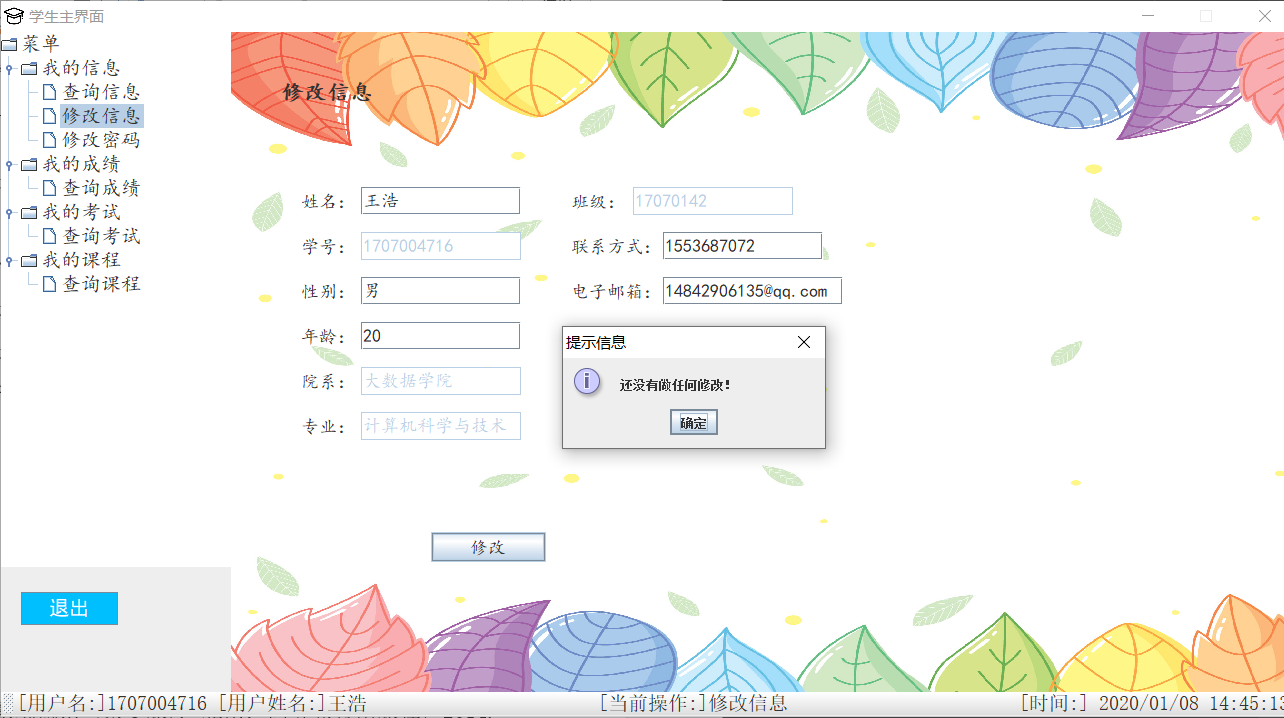


图9-1 修改信息功能测试用例1测试图

**用例 2：**

* 测试条件：原可修改信息被用户修改
* 测试用户：学生
* 测试步骤：在修改信息面板，对可修改信息进行修改修改，点击修改按钮
* 预期结果：系统提示修改成功
* 实际结果：系统提示修改成功

具体结果图如图9-2所示。

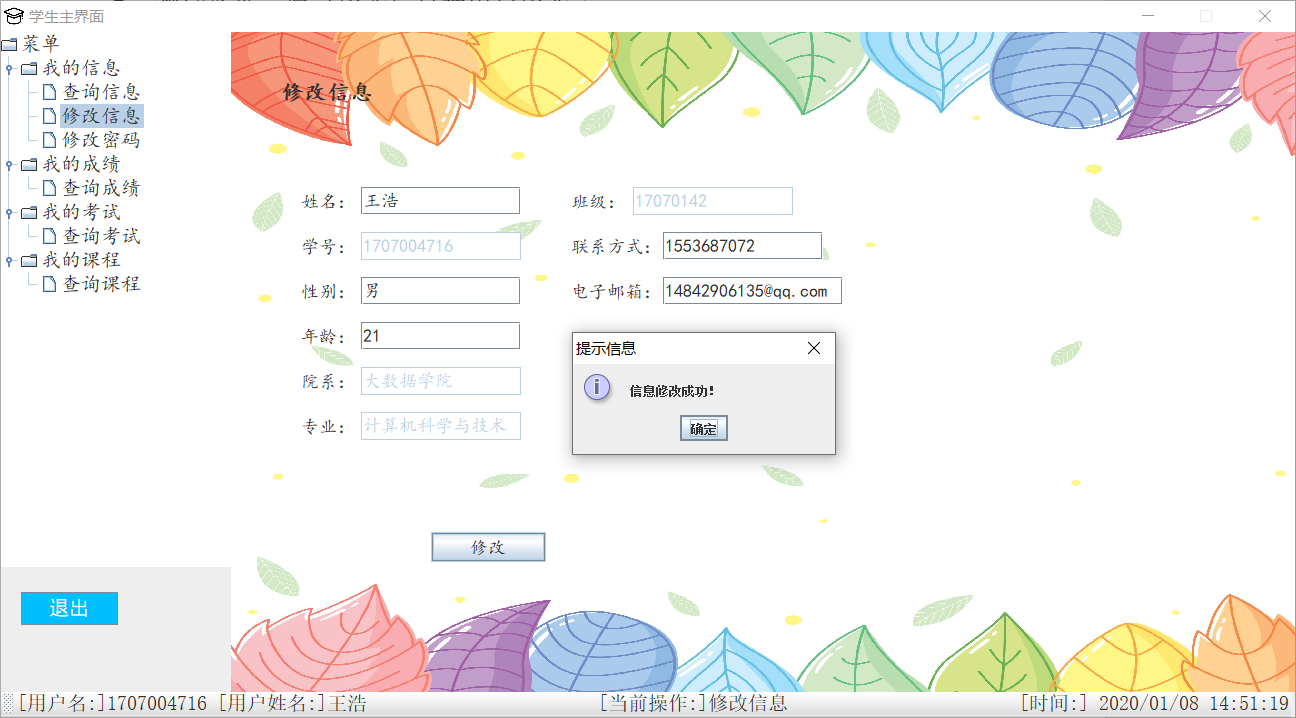


图9-2 修改信息功能测试用例2测试图

9.1.2 修改密码功能测试

测试目的：学生用户能够修改个人的登陆密码。具体修改密码功能等价类表如下表9.2所示。

表9.2 修改密码功能测试等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入等价类** | **测试用例** | **测试数据** | **期望结果** | **实际结果** |
| 无效等价类 | 未输入旧密码 | 旧密码输入为空 | 提示未输入旧密码 | 提示请输入原密码 |
| 旧密码输入有误 | 旧密码：678645  输入旧密码：123456 | 提示旧密码输入有误 | 提示原密码错误，请重新输入 |
| 未输入新密码 | 新密码输入为空 | 提示新密码不能为空 | 提示请输入新密码 |
| 两次新密码输入不一致 | 新密码：12345、  确认密码：123456 | 系统提示：两次输入密码不一致 | 系统提示：两次输入密码不一致 |
| 有效等价类 | 旧密码正确，新密码与确认密码一致 | 旧密码：678645  输入旧密码：678645  新密码：123456  确认密码：123456 | 系统提示：密码修改成功 | 系统提示：密码修改成功 |

**用例 1：**

* 测试条件：原密码为空
* 测试用户：学生
* 测试步骤：在修改密码面板，不输入原密码，并点击修改按钮
* 预期结果：系统提示请输入原密码
* 实际结果：系统提示请输入原密码

具体结果图如图9-3所示。



图9-3修改密码功能测试用例1测试图

**用例 2：**

* 测试条件：原密码输入有误（不正确）
* 测试用户：学生
* 测试步骤：在修改密码面板，输入错误的原密码，并点击修改按钮
* 预期结果：系统提示原密码错误，请重新输入
* 实际结果：系统提示原密码错误，请重新输入

具体结果图如图9-4所示。



图9-4 修改密码功能测试用例2测试图

**用例 3：**

* 测试条件：新密码与确认密码不一致
* 测试用户：学生
* 测试步骤：在修改密码面板，输入两次不同的新密码，并点击修改按钮
* 预期结果：系统提示两次输入密码不一致，请重新输入
* 实际结果：系统提示两次输入密码不一致，请重新输入

具体结果图如图9-5所示。



图9-5 修改密码功能测试用例3测试图

**用例 4：**

* 测试条件：旧密码输入正确，新密码与确认密码一致
* 测试用户：学生
* 测试步骤：在修改密码面板，输入正确的原密码，并保证新密码与确认密码一致，并点击修改按钮
* 预期结果：系统提示更新密码成功
* 实际结果：系统提示更新密码成功，并提示需重新登录以应用更改

具体结果图如图9-6所示。

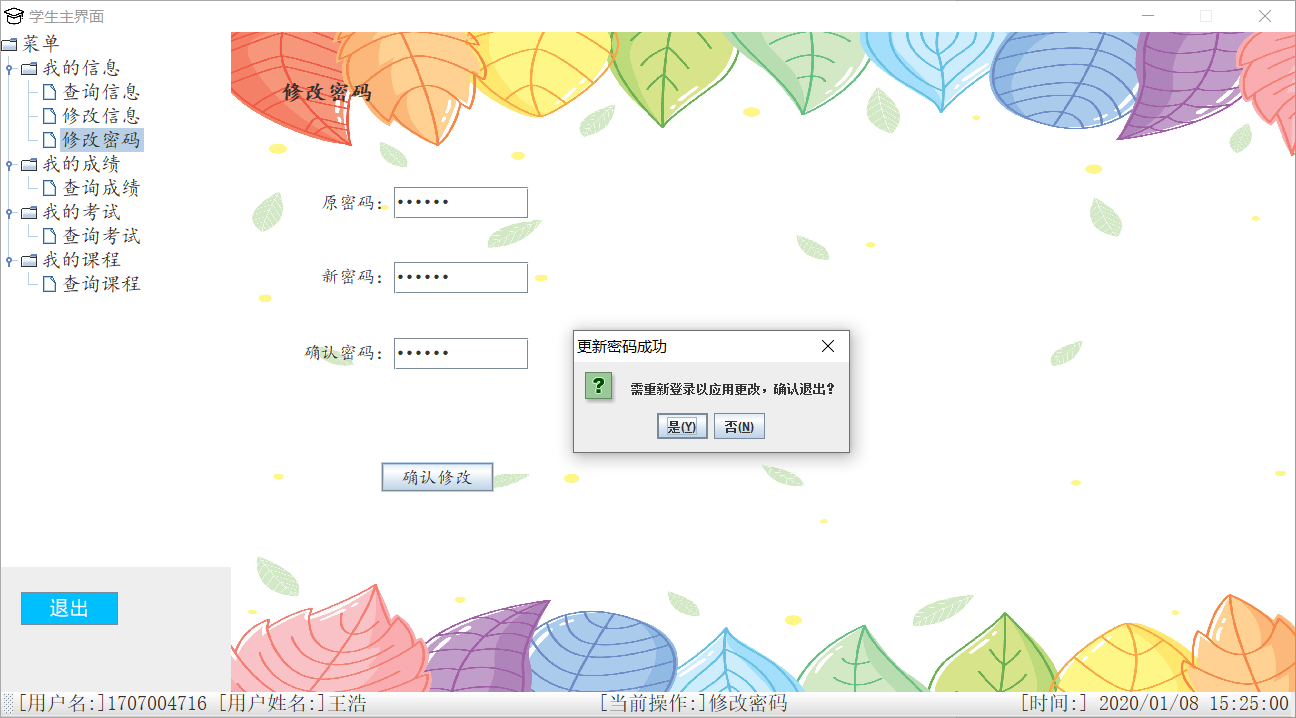


图9-6 修改密码功能测试用例4测试图

9.1.3 查询个人成绩功能测试

测试目的：学生用户能够根据学年和学期查询自己的成绩情况，可通过可视化图标显示出来。具体查询成绩功能等价类表如下表9.3所示。

表9.3 查询成绩功能测试等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入等价类** | **测试用例** | **测试数据** | **期望结果** | **实际结果** |
| 无效等价类 | 未选择学年 | 选择学年：  选择学期： | 系统提示未选择学年 | 系统提示请先选择学年 |
| 未选择学期 | 选择学年：2019-2020  选择学期： | 系统提示未选择学期 | 系统提示请先选择学期 |
| 有效等价类 | 选择对应学年并选择对应学期 | 选择学年：2019-2020  选择学期：第一学期 | 系统展示该学年学期该同学的成绩 | 系统展示该学年学期该同学的成绩并可以通过图表展示 |

**用例 1：**

* 测试条件：未选择学年，并进行查询
* 测试用户：学生
* 测试步骤：不选择学年和学期，并点击查询按钮进行查询
* 预期结果：系统提示系统提示请先选择学年
* 实际结果：系统提示系统提示请先选择学年

具体结果图如图9-7所示。

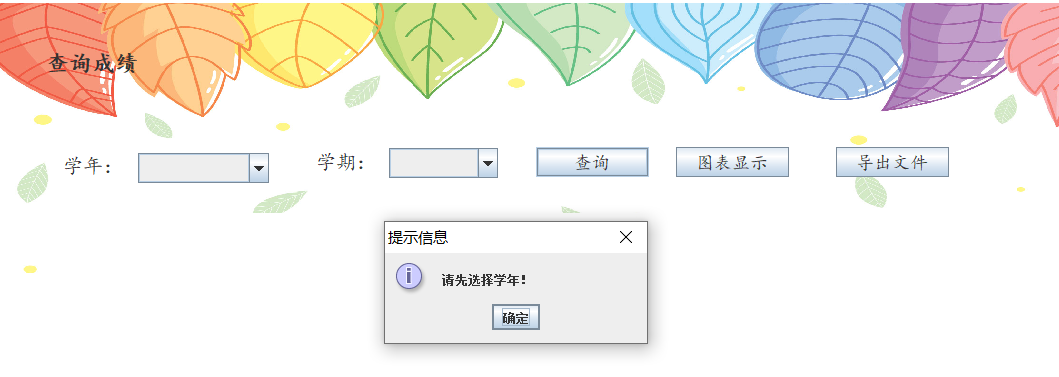


图9-7 成绩查询功能测试用例1测试图

**用例 2：**

* 测试条件：选择对应的学年和学期，并进行查询
* 测试用户：学生
* 测试步骤：选择对应的学年和学期，并点击查询按钮进行查询
* 预期结果：系统提示展示该学年学期该学生各门课程成绩
* 实际结果：系统提示展示该学年学期该学生各门课程成绩，通过图表的方式展示，并可将成绩导出为本地excel文件

具体结果图如图9-8、图9-9所示。



图9-8 成绩查询功能测试用例2测试图

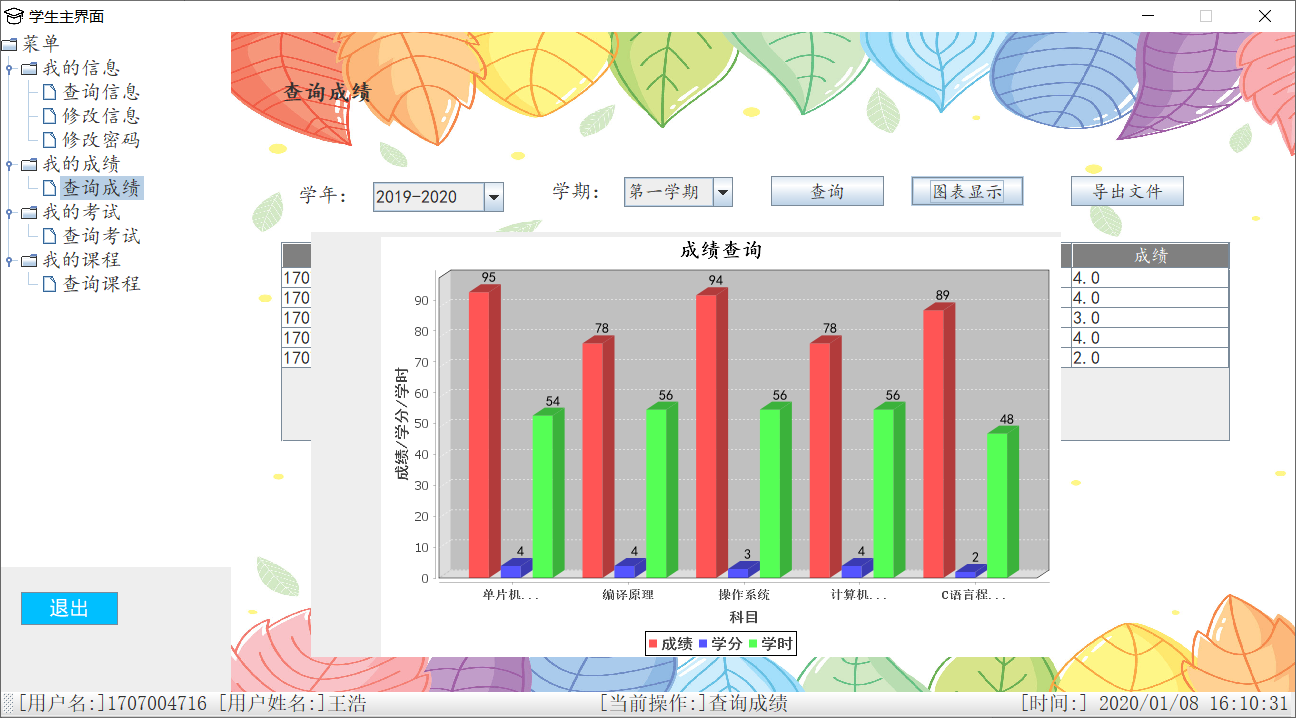


图9-9 成绩查询功能测试用例2测试图

9.1.4 查询考试信息功能测试

测试目的：学生用户能够查询自己所在班级的考试。具体查询考试功能等价类表如下表9.4所示。

表9.4 查询考试功能测试等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入等价类** | **测试用例** | **测试数据** | **期望结果** | **实际结果** |
| 有效等价类 | 点击按钮进行查询 | 无 | 系统展示该学年学期该同学所在班级的考试 | 系统展示该学年学期该同学所在班级的考试 |

**用例 1：**

* 测试条件：点击查询按钮进行查询
* 测试用户：学生
* 测试步骤：点击查询按钮进行查询
* 预期结果：系统展示该学年学期该同学所在班级的考试
* 实际结果：系统展示该学年学期该同学所在班级的考试

具体结果图如图9-10所示。

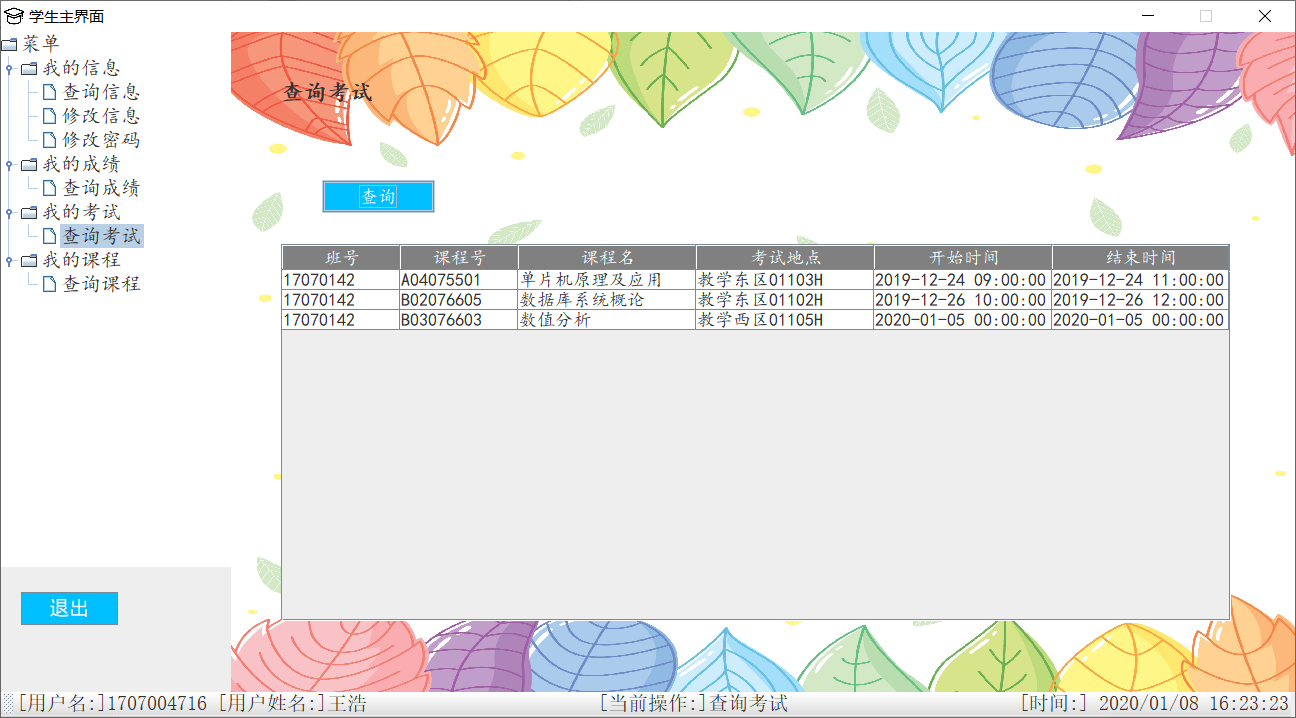


图9-10 考试查询功能测试用例1测试图

9.1.5 查询课程信息功能测试

测试目的：学生用户能够查询自己所在班级指定学年学期的课程。具体查询课程功能等价类表如下表9.5所示。

表9.5 查询课程功能测试等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入等价类** | **测试用例** | **测试数据** | **期望结果** | **实际结果** |
| 无效等价类 | 未选择学年 | 选择学年：  选择学期： | 系统提示未选择学年 | 系统提示请先选择学年 |
| 未选择学期 | 选择学年：2019-2020  选择学期： | 系统提示未选择学期 | 系统提示请先选择学期 |
| 有效等价类 | 选择对应学年并选择对应学期 | 选择学年：2019-2020  选择学期：第一学期 | 系统展示该学年学期该同学所在班级的课程 | 系统展示该学年学期该同学所在班级的课程 |

**用例 1：**

* 测试条件：未选择学年或学期，并进行查询
* 测试用户：学生
* 测试步骤：不选择学年和学期或只选择学年不选择学期，并点击查询按钮进行查询
* 预期结果：系统提示请先选择学年或先选择学期
* 实际结果：系统提示请先选择学年或先选择学期

具体结果图如图9-11、图9-12所示。

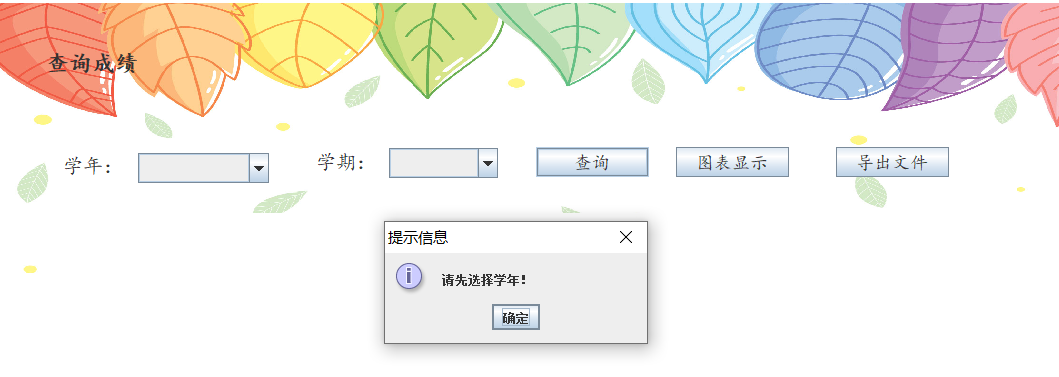


图9-11 课程查询功能测试用例1测试图

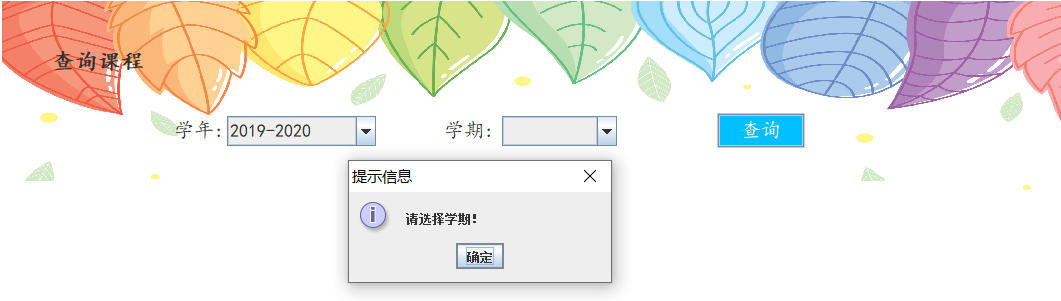


图9-12 课程查询功能测试用例1测试图

**用例 2：**

* 测试条件：选择对应的学年和学期，并进行查询
* 测试用户：学生
* 测试步骤：选择对应的学年和学期，并点击查询按钮进行查询
* 预期结果：系统展示该学年学期该同学所在班级的课程
* 实际结果：系统展示该学年学期该同学所在班级的课程

具体结果图如图9-13所示。

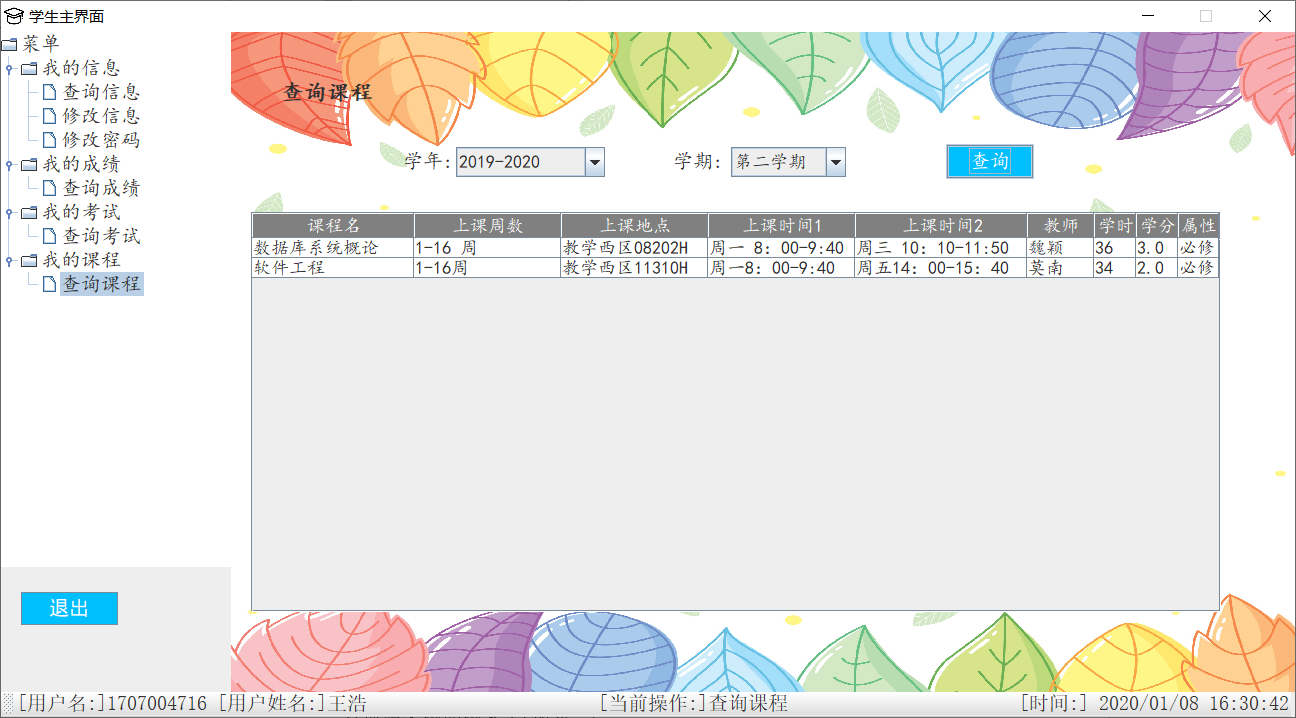


图9-13 课程查询功能测试用例2测试图

9.1.6 数据库备份功能测试

测试目的：管理员可以进行数据库备份，以免发生错误时数据丢失。

**用例 1：**

* 测试条件：将数据库保存到本地进行备份
* 测试用户：管理员
* 测试步骤：登录成功后，点击数据库备份，进行保存
* 预期结果：数据库备份成功
* 实际结果：数据库备份成功，

具体结果图如图9-14所示。

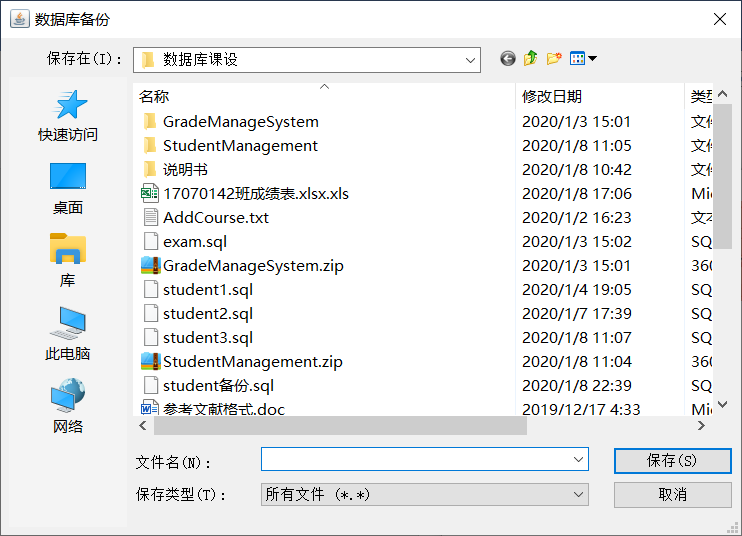




图9-14 数据库备份功能用例1结果图

**10 结论**

本次数据库课程设计，我们小组使用了Java程序设计课程上学到的Java Swing设计我们的系统界面，使用MySQL数据库来存储我们的数据并使用JDBC连接查询我们的数据。在设计过程当中也遇到了一些问题，比如作为作为外码的属性数据类型要一致、外码的主属性表无法删除数据、数据库转储必须输入密码等问题。由于我们一开始需求分析没有做好，导致我们后来做了很大的改动，深刻地体会到了需求分析的重要性。虽然此次学生管理系统可能还存在些许问题，但是一个完整数据库系统被我们构造出来，还是挺有成就感的。

**参 考 文 献**

[1] 王珊,萨师煊. 数据库系统概论（第5版). 北京：高等教育出版社，2014.

[2] 马忠贵. 数据库技术及应用——Microsoft SQL Server 2008+Java. 北京：国防工业出版社, 2012.

[3] 李刚. Java程序员之旅--Java数据库技术详解. 北京：化学工业出版社，2010.

[4] 贾蓉生，胡大源，林金池. Java典型应用彻查1000例-数据库应用基础.

.北京：科学出版社.2009.

[5] 郑阿奇. SQL Server实用教程（第4版）.北京：电子工业出版社，2015.