**哈尔滨理工大学**

**计算机科学与技术学院**

**《DB-Java》**

**课程设计报告**

|  |  |
| --- | --- |
| **题 目** | ：学生信息管理系统 |
| **班 级** | ：网络23-2 班 |
| **专 业** | ：网络工程 |
| **姓名**  **（学 号）** | 黄策（2304020210）  陈子权（2304020205） |
| 翟昆洋（2206040305）  饶蕊（2206040115） |
| **指导教师** | ：王莉莉 |
| **日期** | ：2024年12月30日 |

**目录**

[一、需求分析 1](#_Toc29407)

[一、系统开发背景 1](#_Toc28240)

[二、系统开发意义 1](#_Toc5372)

[三、 系统开发功能 3](#_Toc10605)

[二、系统设计 7](#_Toc6251)

[（一）系统中的数据定义 7](#_Toc13685)

[（二） 系统的概要设计 9](#_Toc4989)

[（三） 系统的详细设计 11](#_Toc16478)

[（四） 数据库设计 13](#_Toc29049)

[（五）系统的核心算法 15](#_Toc20704)

[三、系统编码及运行 21](#_Toc19640)

[（一）系统开发涉及的软件 21](#_Toc29820)

[开发工具 21](#_Toc12023)

[（二）系统运行界面及结果......................................................24](#_Toc7965)

[五、总结 35](#_Toc4376)

[二、系统项目开发过程中的收获 37](#_Toc20561)

[三、未来专业学习的目标 38](#_Toc32215)

[附录 41](#_Toc29605)

## 一、需求分析

### （一）系统开发背景

在当今数字化时代，教育领域的规模不断扩大，学生数量持续增长，各类教育机构，包括中小学、高校以及各类培训学校等，所涉及的学生信息愈发繁杂多样。传统的学生信息管理方式主要依赖人工记录和纸质文档保存，这种模式在面对日益增长的信息量时，暴露出诸多弊端。

随着学校的发展，招生范围的拓宽，一个普通学校每年入学的新生可达数百人甚至上千人，加上老生信息的动态更新，如成绩变化、奖惩记录、课程选修等，使得学生信息的总量呈爆炸式增长。以高校为例，一个院系的学生专业、课程、导师、实践活动等信息交织在一起，人工处理这些数据不仅效率低下，而且极易出错。教师在统计学生成绩、查询学业进展时，往往需要翻阅大量纸质档案，耗费大量时间精力，还可能因人为疏忽导致成绩录入错误、信息更新不及时等问题，影响对学生学业的准确评估。

从学校管理层面看，不同部门之间对于学生信息的共享需求迫切。教务处需要学生的课程注册、考试安排信息来组织教学活动；学生处关注学生的奖惩、心理健康、宿舍安排等生活方面；财务处则侧重学费收缴、奖助学金发放明细。然而，传统手工管理模式下，信息分散于各部门，形成“信息孤岛”，部门间沟通协调成本高，难以实现高效协同办公。一旦学生办理转学、休学等手续，信息变更需要在多个部门依次传递，稍有延误就会造成管理混乱，给学生和家长带来困扰。

此外，教育信息化趋势促使学校与家长、教育主管部门之间的信息交互日益频繁。家长期望实时了解孩子在校学习、生活表现；教育部门需要精准的学生数据进行教育政策制定、资源分配评估等宏观调控。陈旧的管理手段无法满足这些快速、准确传递信息的需求，阻碍教育管理现代化进程，开发一套全面、高效的学生信息管理系统迫在眉睫。

### （二）系统开发意义

1.提高管理效率

（1）数据录入便捷化：学生信息管理系统为学校各部门提供统一的数据录入端口，工作人员只需按照标准化格式，一次性输入学生基本信息、学业信息、缴费信息等各类数据，系统自动存储并分类整理。例如招生季，招生办录入新生入学资料后，系统立即将相关信息同步至教务处、学生处等部门，无需重复劳动，节省大量人力与时间成本，相较于传统人工录入分发，效率可提升数倍。

（2） 快速查询与统计：教师想要了解某学生的历年成绩、课程完成情况，只需在系统检索栏输入学生学号或姓名，几秒钟内即可获取详尽报告，包括成绩趋势分析、挂科补考记录等；管理人员能迅速统计各年级、各专业学生人数分布、男女比例、生源地情况等，为教学资源调配、招生计划制定提供精准依据，克服以往人工翻阅档案、手工计算的缓慢低效。

2.提升信息准确性

（1） 减少人为错误：系统内置数据校验功能，像成绩录入时，自动判断数据格式是否正确、是否超出合理分值范围，杜绝因手误写错成绩；学生个人信息修改遵循严格审批流程，修改痕迹全程留档，保证信息变更的合法性、准确性，避免人工记录随意涂改、信息前后矛盾等问题，确保学生档案的严谨性。

（2） 实时数据更新：学生选课、退课操作实时反馈在系统课程安排模块；获得新荣誉、受到处分等情况一经记录，各关联部门看到的都是最新状态，保证各方依据同一数据源工作，消除因信息滞后造成的决策偏差，为学业评价、评优评先、纪律管理等提供可靠支撑。

3.优化教育服务

（1） 面向学生个性化服务：学生登录系统可自主查询课表、考试安排、成绩排名，根据学业预警提示合理规划学习，选修感兴趣课程；还能查看奖助学金申请进度、宿舍水电使用情况等生活信息，实现校园生活自主管理，提升学习体验与自我规划能力。

（2） 加强家校沟通：系统设置家长端入口，家长能远程了解孩子出勤、课堂表现、作业完成情况，定期接收学业报告，即便身处异地也能深度参与孩子教育过程，配合学校督促学业，形成家校共育合力，促进学生健康成长。

4.辅助教育决策

（1）数据分析支持：通过对历年学生成绩数据挖掘，分析课程难度系数、各专业通过率，为教学改革提供量化依据，如发现某专业多门课程不及格率偏高，学校可针对性优化课程内容、改进教学方法；从生源质量、毕业去向等大数据洞察学校吸引力、人才培养成效，助力调整招生策略、专业设置，提升学校竞争力。

（2） 资源合理配置：依据系统反映的学生选课热度、实验室设备使用频率，合理安排教室、实验室开放时间，调配师资力量，避免资源闲置或过度紧张，实现教育资源效益最大化，保障教学活动有序开展。

综上所述，开发学生信息管理系统是适应教育现代化发展、提升学校管理水平与服务质量的关键举措，对推动教育机构高效运转、助力学生成长成才具有深远意义，为教育信息化蓝图添上浓墨重彩的一笔。

### （三）系统开发功能

本学生信息管理系统依托架构构建而成，采用 MySQL 作为后端核心数据库，对学生、班级、院系、课程以及成绩等多维度海量数据进行高效存储与管理。在数据库设计层面，除构建严谨的数据表结构以保障数据规范化存储外，还充分运用视图、存储过程和触发器等高级特性，深度优化数据查询、更新及关联操作，全方位提升数据处理效能。

用户成功登录系统后，即可访问一系列精准且实用的信息查询与维护功能。

1. 查看学生信息：在该功能模块下，系统以结构化表格形式呈现详细信息，涵盖用户自身及其同学的关键资料。其中，姓名与学号作为主键，具备唯一性，精准对应每位学生个体；性别、出生日期等基本属性清晰展示；联系方式字段提供如手机号码、电子邮箱等多元信息，确保沟通渠道畅通，满足日常交流及紧急联络需求。

2. 查看班级信息：聚焦班级维度，系统全面展示班级综合信息。班级名称作为标识具备唯一性，班主任信息明确班级管理责任人，班级实时人数动态反映人员变动情况。

3. 查看院系信息：切换至院系视角，系统呈现丰富且专业的学术资料。院系名称精准定位学科领域，专业方向列表详细列举各专业代码、名称及培养方向，为学业规划提供指引；

4. 查看课程信息：进入课程信息查询板块，每门课程详情精准呈现。课程名称明确知识核心主题，授课教师信息关联教师个人资料，展示其专业资质与教学专长；

5. 查看成绩信息：专为学生设计的成绩查询功能，精准反映学业成效。科目名称与考试成绩一一对应，直观呈现知识掌握程度；绩点换算依据标准化算法生成，客观衡量学习质量。同时，系统提供成绩排名、课程平均分对比等分析数据，辅助学生精准定位自身学习水平；还可按需生成成绩趋势图，以可视化方式展现学习轨迹变化，便于自我评估与调整学习策略。

6. 修改用户信息：考虑到个人信息动态更新需求，系统为用户赋予自主修改权限。用户可对联系方式字段进行编辑，确保信息准确性，便于接收通知；

在用户平台上，用户能够执行登录操作，并且能够查看学生信息等其他功能。用户在用户流程中的具体实现分析如下所示。用户用例图请参见图1-1。

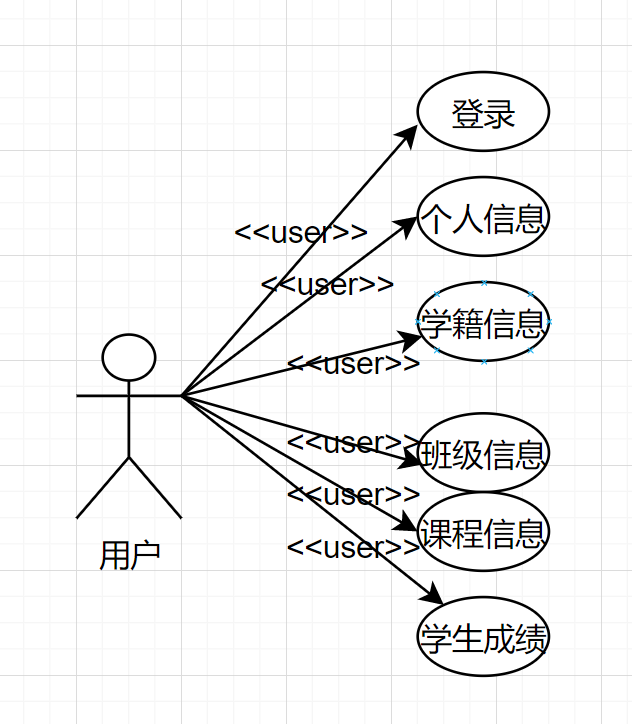


图1-1 用户用例图

管理员在系统中扮演关键的管理角色，拥有高级权限，负责维护系统数据生态的完整性、准确性与时效性，保障系统平稳高效运行。

1. 学生信息录入：新增学生信息时，管理员操作界面呈现一张精细的表单，涵盖必填项如姓名、学号、身份证号，用于精准识别学生身份；入学时间、生源地、高考成绩等信息提供生源背景与入学起点参考；兴趣爱好、特长技能等拓展字段用于丰富学生画像。尤为突出的是，系统支持批量导入功能，面对招生旺季海量新生数据录入需求，管理员只需依照标准化模板整理 Excel 文件，一键上传后，系统自动启动解析、校验程序，严格比对数据格式与逻辑规则，高效完成批量数据入库，大幅缩减人力与时间成本。

2. 班级信息录入：搭建班级信息架构时，班级名称、入学年份作为关键标识，确保班级唯一性识别；班主任人选关联至教师信息库，明确育人主体责任；班级人数字段初始设定后，伴随人员增减操作实时自动更新。此外，管理员还可录入班级特色文化描述、班费收支管理规则等拓展信息，为班级数字化管理增添内涵，助力班级规范化、特色化发展。

3. 院系信息录入：塑造院系信息全景时，管理员需精准录入院系名称、简称、成立时间等基础信息，详细阐述各专业方向设置详情，包括专业代码、培养目标、学位授予类型等核心要素；通过上传院系宣传资料、知名教授简介文档及图片，以多媒体形式全方位展现院系专业实力与学术风采，提升院系知名度与吸引力，为招生、科研合作等工作奠定基础。

4. 课程信息录入：构建课程知识蓝图环节，课程编号作为唯一标识符，贯穿课程全生命周期管理；课程名称精准概括知识核心要点，授课教师选取关联教师专业资质库，确保教学专长与课程匹配；课时规划兼顾理论讲授、实践操作环节合理分配，学分设定依据课程难度系数、学业价值科学权衡。同时，系统支持关联教材选用信息、上传课程大纲文档，为教学活动有序开展提供完备的基础资料。

5. 成绩信息录入：录入成绩过程支持多样化操作模式，以适配不同场景需求。手工单个录入功能用于应对少量特殊成绩调整，如补考、重修成绩录入；批量导入功能适用于期末考试、阶段性测验等大规模数据处理场景，大幅提升录入效率；还可与智能阅卷系统对接，实现成绩实时采集，减少人工干预误差。录入过程中，系统自动校验成绩格式，实时核算总分、平均分、排名等统计指标，成绩提交瞬间同步触发学业预警机制，依据预设规则筛查学业风险学生，为精准帮扶提供依据，护航学生学业成长。

6. 删除用户：在系统运营过程中，针对毕业离校、退学等学籍状态变更导致无需系统账号的学生群体，管理员依循审慎流程启动删除操作。删除前，系统强制多重确认学生身份信息，比对学籍档案、核实离校手续办理情况；关键数据如学习成绩、奖惩记录等自动备份至历史数据库，确保数据可追溯；操作全程系统留痕，详细记录删除时间、操作人员、删除原因等信息，保障数据清理流程规范、安全，维护系统用户数据库的精准性与精简性。

在管理员平台上，管理员能够执行登录和用户管理操作，并且能够修改学生信息等其他功能。用户在用户流程中的具体实现分析如下所示。管理员用例图请参见图1-2。

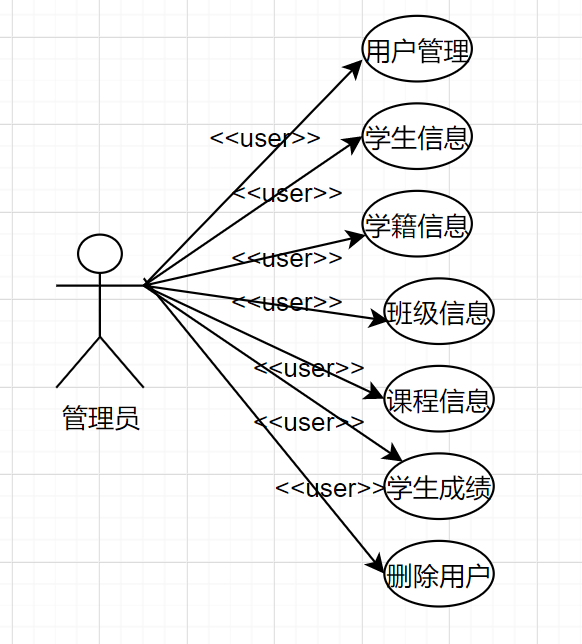
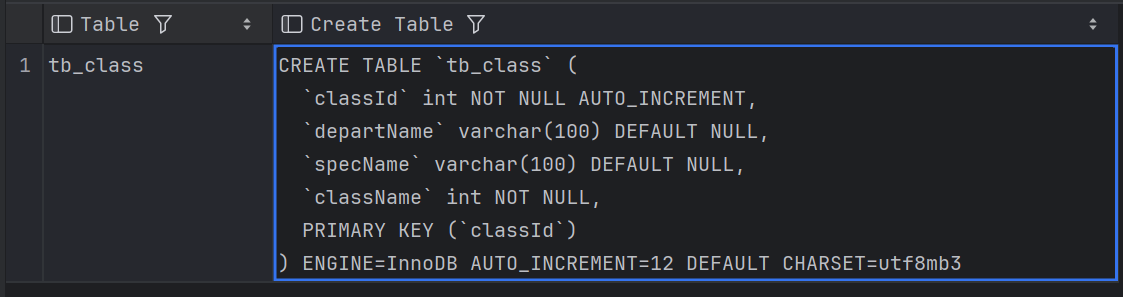


图1-2 管理员用例图

## 二、系统设计

### （一）系统中的数据定义

1.tb\_class表中数据定义如图2-1所示：

 图2-1 tb\_calss表中数据定义

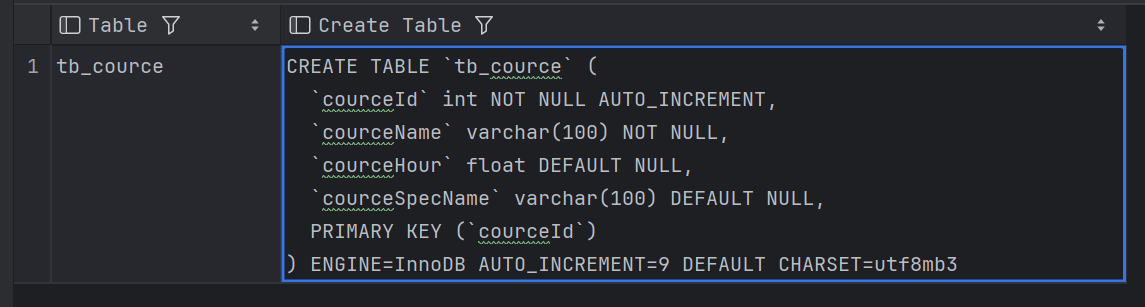
2.tb\_course表中数据定义如图2-2所示：  


图2-2 tb\_course表中数据定义

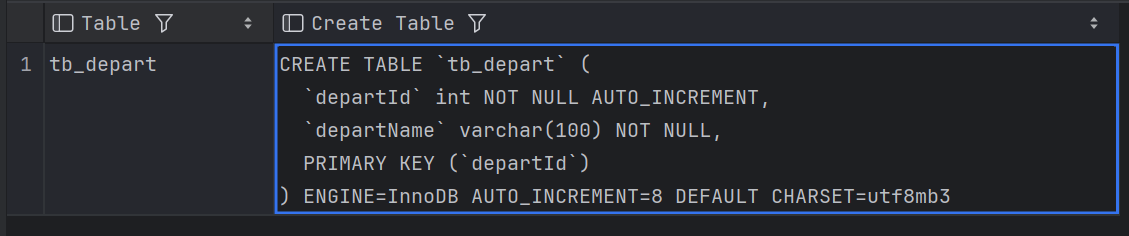
3.tb\_depart表中数据定义如图2-3所示：  


图2-3 tb\_depart表中数据定义

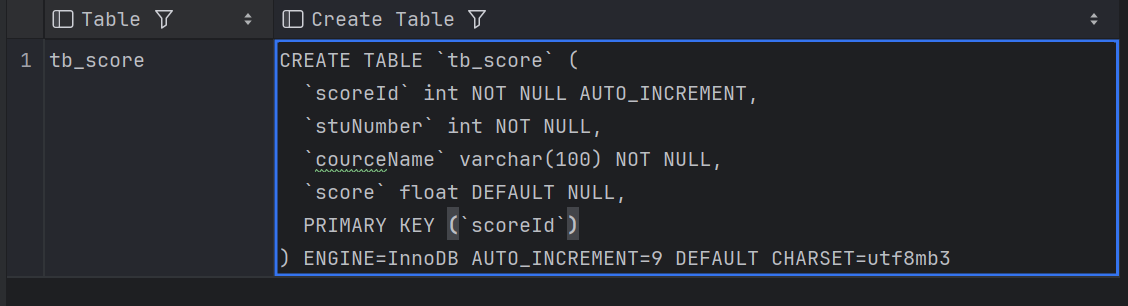
4.tb\_score表中数据定义如图2-4所示：  


图2-4 tb\_score表中数据定义

5.tb\_spec表中数据定义如图2-5所示：

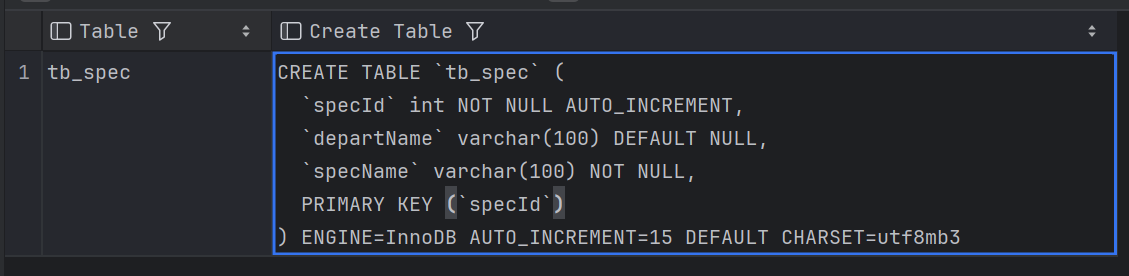


图2-5 tb\_spec表中数据定义

6. tb\_user表中数据定义如图2-6所示：

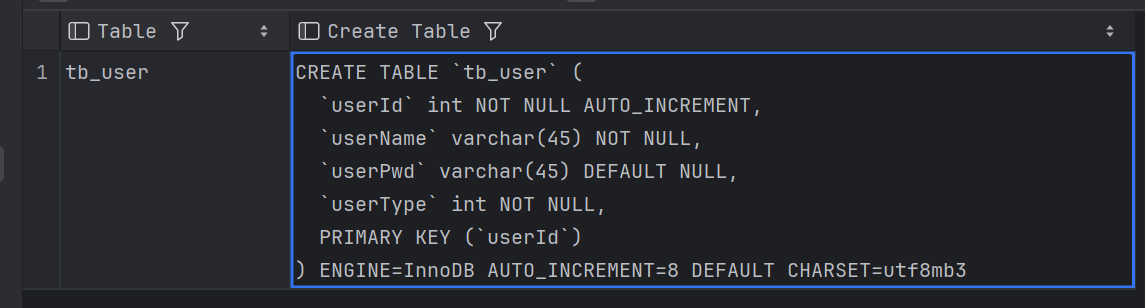


图2-6 tb\_user表中数据定义

7.tb\_student表中数据定义如图2-7所示：

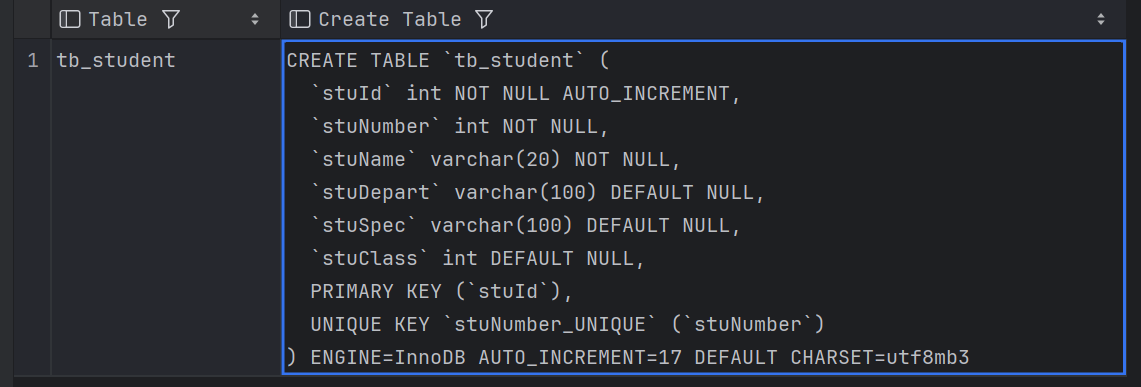


图2-7 tb\_student表中数据定义

### 系统的概要设计

在当今数字化快速发展的时代，一个高效、精准且贴合教育管理需求的系统设计至关重要。本系统设计的核心目的，便是全方位、深入地剖析系统涵盖的所有功能结构，从而为后续开发人员顺利开展开发工作、完美实现系统功能筑牢根基，铺设好前行的道路。

历经前期严谨且细致的需求调查、深入的分析以及精心的整理流程后，最终确定的总体需求凝聚着各方智慧，主要涵盖了六大关键模块。首先是学生管理模块，它如同精密齿轮组中的核心部件，关联着学生从入学到毕业全流程的信息管控；班级管理模块负责统筹班级架构、班委设置等班级生态事务；院系管理着眼于宏观的院系资源调配、专业布局；课程管理精心编排课表、调配师资与教室资源；用户管理精准掌控系统使用者的各类权限与账号细节；

系统整体角色架构简洁而明晰，划分为两个相辅相成的部分。一方面是作为系统的管理员，从系统后台维护到数据的最终审核，都由其把关；另一方面是广大用户群体，其中学生作为主要的用户构成，除具备基础的浏览信息权限外，还被赋予了诸多专属权利，他们能够便捷查询和精细管理自己的账户信息，随时追踪学业进程中的成绩动态，精准核实学籍信息等，全方位保障自身学习权益，确保系统服务精准触达需求。该系统功能结构图直观展现了各模块与角色间的协同关系。系统功能结构图如图2-8所示。

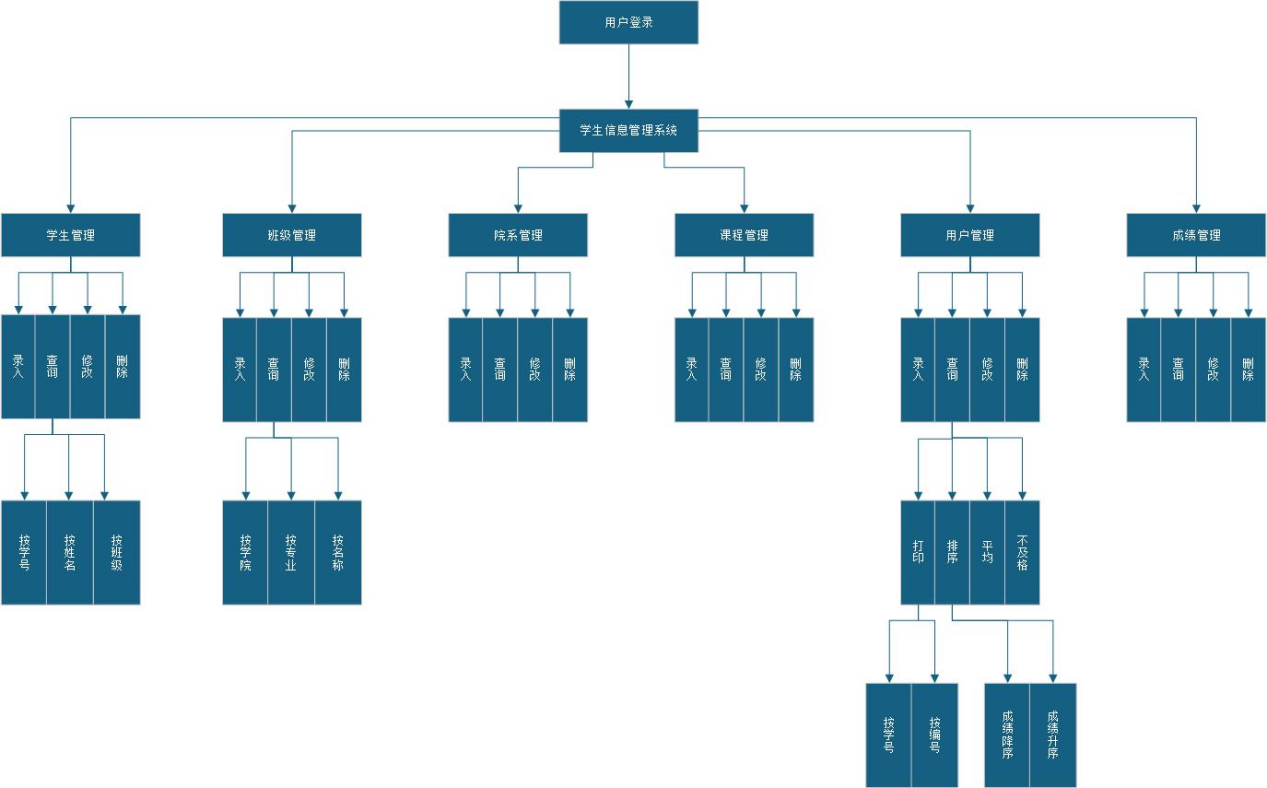


图2-8 系统功能结构图

表单字段校验处理设计：

1.出错信息类型

A 未输入必填项。用户未填写系统要求的必填字段。常见的必填项包括姓名、学号、密码等。

B 应输入合法字符或数字却输入非法字符或数字。用户输入的数据类型不符合系统要求。常见的错误包括数字字段中输入了字母、特殊字符。

C 对空数据表进行更改或删除操作。用户尝试对没有任何记录的数据表执行修改或删除操作。此类错误通常出现在数据库操作中，尤其是更新、删除操作时。

D 向要求唯一值的关键字段添加重复值。用户在要求唯一值的字段中输入了重复数据，违反了唯一性约束。

2.出错处理对策

对于错误A，系统要求用户输入非空值。实时提示：在表单字段旁边实时显示错误信息，提醒用户该字段为必填项。视觉提示：通过表单字段旁的红色星号或提示框，使必填项易于辨识。

对于错误B，系统要求用户输入合法字符。实时校验：在用户输入时，通过前端实时校验，立即提示非法字符或数字输入。详细错误信息：不仅提示“输入错误”，还需给出具体指导。

对于错误C，系统提示数据表无记录可更改或删除。数据校验：在执行更改或删除操作时，先校验数据表中是否有记录。如果为空，提示用户：“数据表无记录可更改或删除”。操作限制：对于空数据表，可以禁用修改或删除按钮，避免用户错误操作。

对于错误D，系统要求用户输入非重复值。即时验证：在输入时，通过AJAX请求实时验证字段的唯一性，若发现重复值，立即提示用户。明确提示信息：明确告知用户哪个字段存在重复，并给出有效的解决方案。

### 系统的详细设计

**1.功能模块设计**

该模块是为所有用户登录设计的，如学生登录后只能进行自己的普通功能操作(如个人信息修改），管理员和超级管理员登录后有不同的权限，管理员不能超越权限。超级管理员能对整个系统的数据进行管理，主要是用户的登录权限以及用户登录后在系统里的操作权限，用户登录界面如图2-9所示，用户注册界面如图2-10所示。

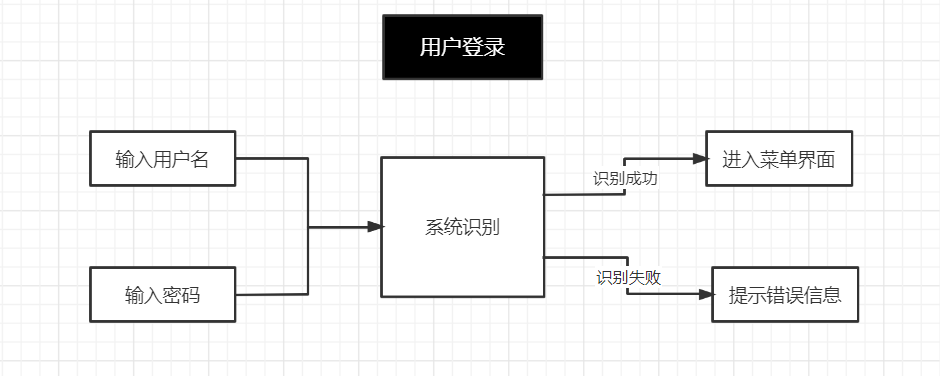


图2-9 用户登录流程图

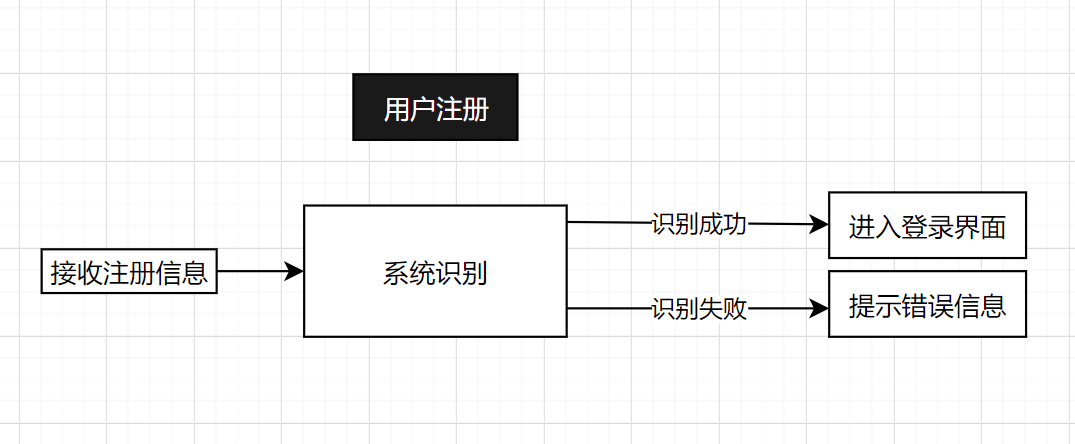


图2-10用户注册流程图

对于学生管理模块，学生录入相关信息数据，点击录入按钮，依次填写要录入的成绩数据，点击提交按钮，将数据提交至数据库，然后刷新成绩数据页面，每条数据右边有删除和编辑按钮，来完成相应的删除和更新功能。

对于班级管理模块，学生录入班级数据，点击班级录入按钮，依次填写要录入的班级数据，点击提交按钮，将数据提交至数据库，然后刷新班级数据页面，每条数据右边有删除和编辑按钮，来完成相应的删除和更新功能。

对于院系管理模块，学生录入院系数据，点击院系录入按钮，依次填写要录入的院系数据，点击提交按钮，将数据提交至数据库，然后刷新院系数据页面，每条数据右边有删除和编辑按钮，来完成相应的删除和更新功能。

对于课程管理模块，学生录入课程数据，点击课程录入按钮，依次填写要录入的课程数据，点击提交按钮，将数据提交至数据库，然后刷新课程数据页面，每条数据右边有删除和编辑按钮，来完成相应的删除和更新功能。

对于成绩管理模块，学生录入成绩数据，点击成绩录入按钮，依次填写要录入的成绩数据，点击提交按钮，将数据提交至数据库，然后刷新成绩数据页面，每条数据右边有删除和编辑按钮，来完成相应的删除和更新功能。

**2.系统维护设计**

学生信息管理系统是个现代化的高度集成的综合信息系统，系统主要服务对象为用户，最大化在页面中提供多的信息，因此在系统维护上，数据量较大，整体的维护设计如下：

（1）代码的维护：建立代码版本控制系统，如使用 Git，为每次代码更新留下详细记录，包括修改人、修改时间、修改原因，以便追溯问题源头。以及安排专业代码审查小组，定期审查核心模块代码，依据代码规范、可读性、可扩展性标准优化，提升代码质量，预防潜在隐患。

（2）功能增加：功能升级在必要的前提下，结合用户反馈与行业发展趋势，精准调研新功能需求，经严谨评估、规划、设计后融入系统，避免冗余功能堆砌。

（3）数据维护：根据安全性等需求，定期对数据库进行人工备份。同时制定数据恢复演练计划，确保备份数据有效性。

### 数据库设计

数据库E-R图2-11和图2-12所示：

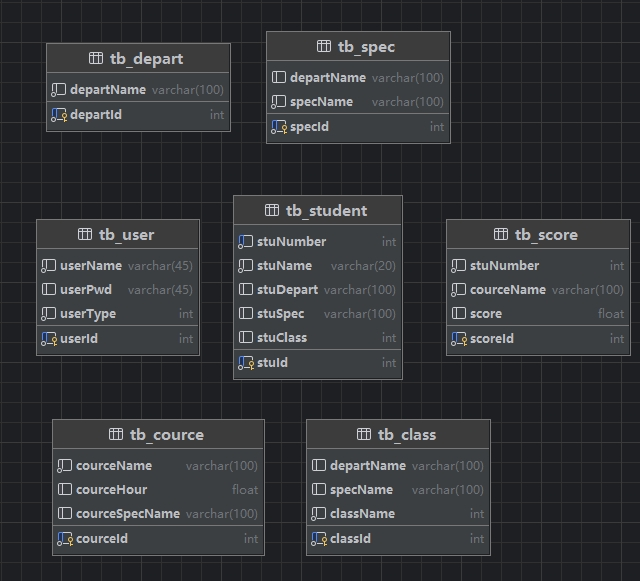


图2-11 数据库E-R图

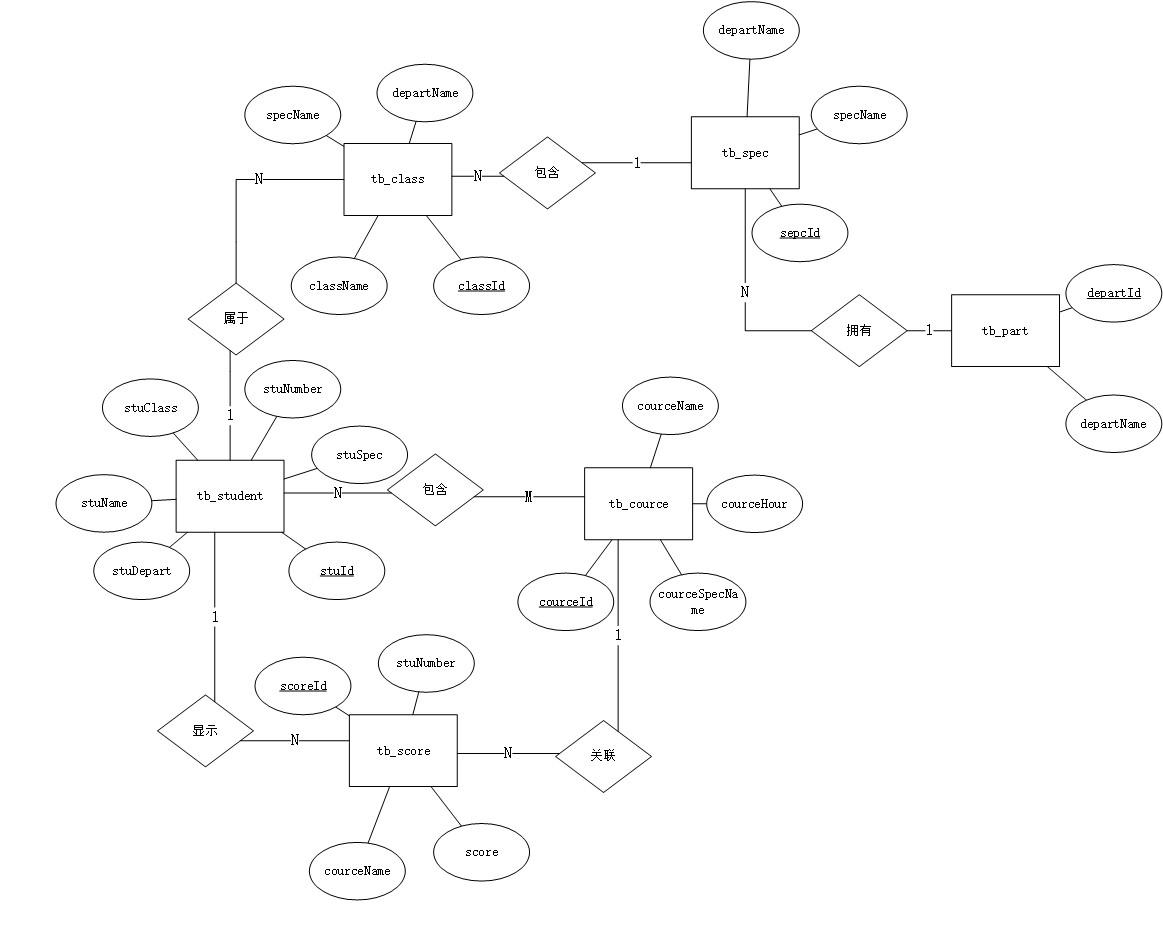


图2-12 数据库E-R图

**实体及其属性**

1. tb\_depart(学院表)  
   tb\_depart 用于记录学院或部门的信息，每个部门有一个唯一的 ID（departId）以及对应的部门名称（departName） ，如图2-13所示。

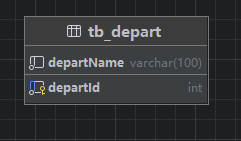


图2-13 学院表

2.tb\_spec (专业表)  
tb\_spec 表用于记录各个专业的信息。每个专业都有一个唯一的 specId，并且与特定的学院或部门（departName）相关联，专业名称（specName）也被存储。如图2-14所示。

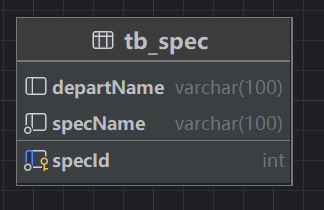


图2-14专业表

3.tb\_class (班级表)  
tb\_class 表用于记录各个班级的信息。每个班级都有一个唯一的 classId，并且与所属学院（departName）和专业（specName）相关联。班级名称（className）用来标识具体的班级编号或名称。如图2-15所示。

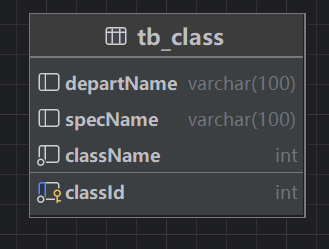


图2-15 班级表

4.tb\_user (用户表)  
tb\_user 表用于管理系统中的所有用户信息，每个用户都有一个唯一的 userId，并且与用户名（userName）、密码（userPwd）和用户类型（userType）相关联。userType 字段可以用来区分不同权限的用户。  
tb\_user 表中的数据可以用来实现用户认证和授权，确保不同用户在系统中的操作权限符合其角色。例如，管理员可以访问和管理所有数据，而普通用户只能访问自己的数据。  
该表支持系统的身份验证和安全控制，确保用户在登录时能够通过用户名和密码进行身份验证，同时根据 userType 控制其可执行的操作和访问的资源。如图2-16所示。

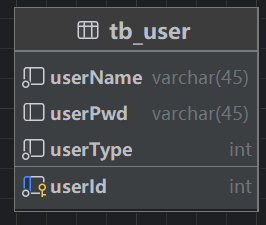


图2-16 用户表

1. tb\_student (学生表)  
   记录学生基本信息：tb\_student 表是用于存储学生的基本信息的核心表。每个学生都有一个唯一的 stuId 和 stuNumber，用来标识学生。

学生信息管理：tb\_student 表为系统提供了学生信息管理的基础。它可以帮助学校管理员或系统用户管理学生的个人信息、查询学生的基本情况。

查询与操作：

可以通过学号（stuNumber）查询特定学生的详细信息。

可以按学院（stuDepart）、专业（stuSpec）或班级（stuClass）对学生进行分组查询，方便进行统计、管理等操作。如图2-17所示。

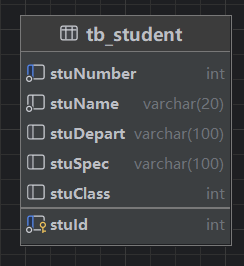


图2-17 学生表

6.tb\_cource (课程表)  
tb\_cource 表用于记录系统中开设的所有课程信息。每个课程都有一个唯一的 courceId，并且与课程名称（courceName）、课程学时数（courceHour）以及适用的专业（courceSpecName）相关联。如图2-18所示。

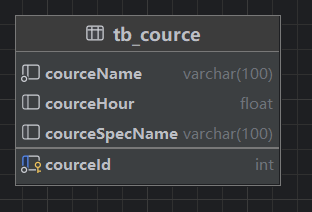


图2-18 课程表

7.tb\_score (成绩表)  
记录学生成绩：tb\_score 表的核心功能是存储学生在不同课程中的成绩。每条记录代表一个学生在某门课程的成绩，且每条记录都有一个唯一的 scoreId。

成绩查询与统计：通过 tb\_score 表，系统可以查询学生在某门课程中的成绩，或者根据学生学号（stuNumber）查询该学生在所有课程中的成绩。也可以根据课程名称（courceName）查询所有选修该课程的学生的成绩。

成绩管理与分析：tb\_score 表对于成绩的管理非常关键，尤其是在需要进行成绩汇总、统计、排序等操作时。它可以支持生成学生成绩单，计算总成绩、平均分等。如图2-19所示。

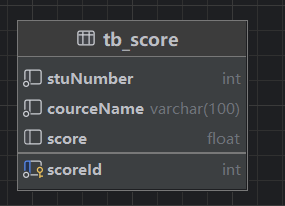
****

图2-19 成绩表

**实体之间的关系**

1.tb\_depart → tb\_spec

一个学院（tb\_depart）包含多个专业（tb\_spec）。  
一个专业（tb\_spec）只属于一个学院（tb\_depart）。

关系类型：一对多（1:N）

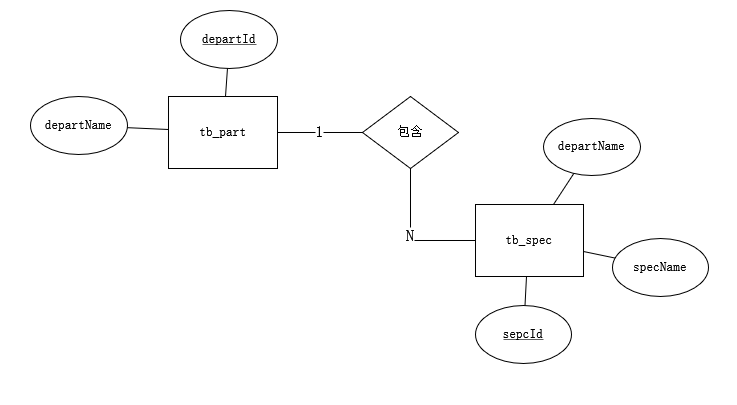
该关系如图2-20所示。  


图2-20 tb\_depart → tb\_spec关系

2.tb\_spec → tb\_class

一个专业（tb\_spec）包含多个班级（tb\_class）。  
一个班级（tb\_class）只对应一个专业（tb\_spec）。

关系类型：一对多（1:N）

该关系如图2-21所示。

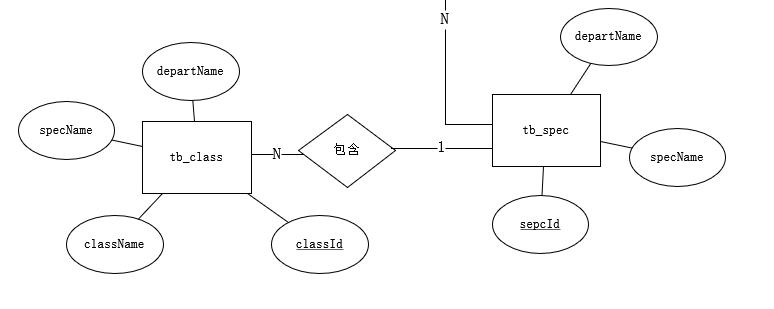


图2-21 tb\_spec → tb\_class关系

3.tb\_student → tb\_class

一个学生（tb\_student）属于一个班级（tb\_class）。  
一个班级（tb\_class）里可以有多个学生（tb\_student）。

关系类型：多对一（N:1）

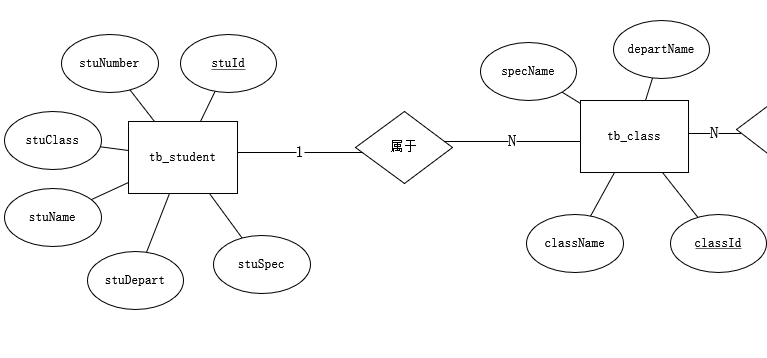
该关系如图2-22所示。  


图2-22 tb\_student → tb\_class关系

4.tb\_student → tb\_score

一个学生（tb\_student）可以有多门课程成绩（tb\_score）。  
一门课程成绩（tb\_score）只能显示一个学生的成绩（tb\_student）。  
关系类型：一对多（1:N）

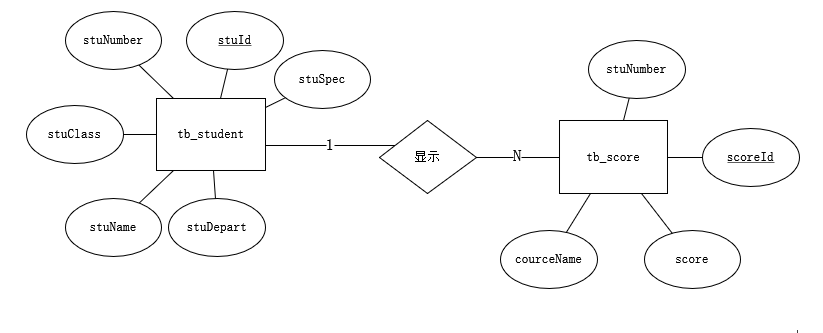
该关系如图2-23所示。  


图2-23 tb\_student → tb\_score关系

5.tb\_cource → tb\_score

一个课程（tb\_cource）可以关联多条成绩记录（tb\_score）。  
一条成绩（tb\_score）只关联一个课程（tb\_cource）。

关系类型：一对多（1:N）

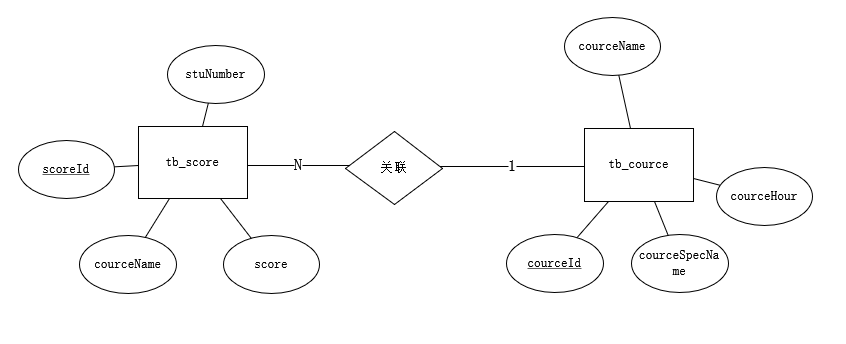
该关系如图2-24所示。  


图2-24 tb\_cource → tb\_score关系

6.tb\_student ↔tb\_cource

一个学生（tb\_student）可以学多门课程（tb\_cource）

一个课程（tb\_cource）可以包含多个学生（tb\_student）

关系类型：多对多（N:M）

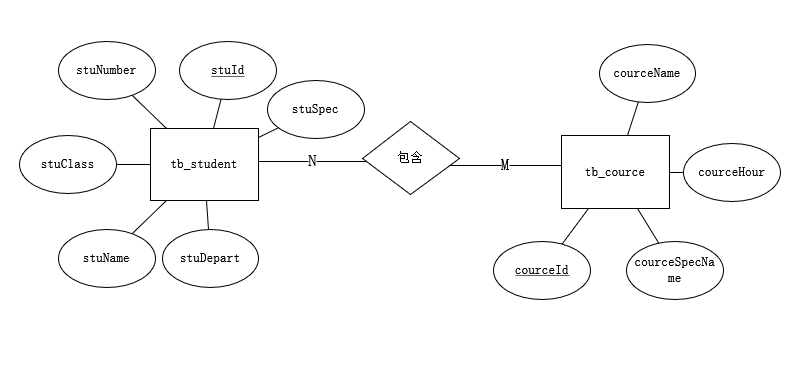
该关系如图2-25所示。  


图2-25 tb\_student ↔tb\_cource关系

### （五）系统的核心算法

**核心算法 1：学生信息管理算法**

可以实现学生信息的添加、修改、删除和查询。并确保数据正确存储至数据库。管理学生信息通过输入操作类型（添加、修改、删除、查询），然后根据操作类型执行相应逻辑：  
 定义输入校验函数

校验学号是否唯一

校验年龄是否在合法范围

校验性别是否有效

如果操作类型为 "添加"：

输入学生信息（学号、姓名、年龄、性别、班级等）

检查输入合法性

构造SQL语句：INSERT INTO students VALUES (...)

执行SQL语句并检查执行结果

如果操作类型为 "修改"：

输入学生ID和需要修改的信息

构造SQL语句：UPDATE students SET ... WHERE id=...

执行SQL语句并检查执行结果

如果操作类型为 "删除"：

输入学生ID

构造SQL语句：DELETE FROM students WHERE id=...

执行SQL语句并检查执行结果

如果操作类型为 "查询"：

输入查询条件（如班级或学号）

构造SQL语句：SELECT \* FROM students WHERE ...

执行SQL语句并返回查询结果  
 错误处理和日志记录

捕获并记录数据库异常  
 提示用户操作结果   
 返回操作成功或失败信息

最后返回操作结果。  
该操作的流程图如图2- 26所示。

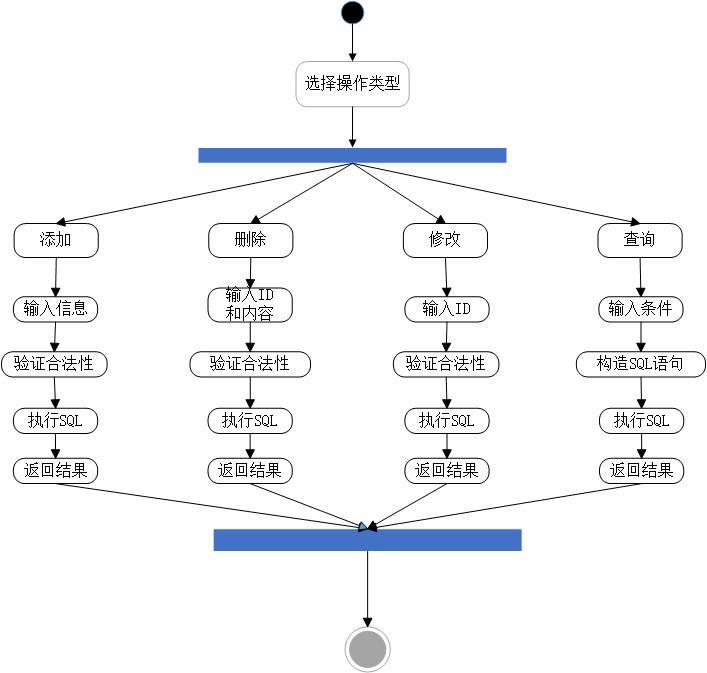


图2-26 课程管理算法流程图

**核心算法 2：课程管理算法**

可以实现课程的添加、修改、删除及课程查询功能。实现管理课程信息先输入操作类型（添加、修改、删除、查询），然后根据操作类型执行相应逻辑：

如果操作类型为 "添加"：

输入课程信息（课程ID、名称、学分、讲师等）

检查输入合法性

构造SQL语句：INSERT INTO courses VALUES (...)

执行SQL语句并检查执行结果

如果操作类型为 "修改"：

输入课程ID和需要修改的信息

构造SQL语句：UPDATE courses SET ... WHERE id=...

执行SQL语句并检查执行结果

如果操作类型为 "删除"：

输入课程ID

构造SQL语句：DELETE FROM courses WHERE id=...

执行SQL语句并检查执行结果

如果操作类型为 "修改"：

输入课程ID和需要修改的信息

构造SQL语句：UPDATE courses SET ... WHERE id=...

执行SQL语句并检查执行结果

如果操作类型为 "删除"：

输入课程ID

构造SQL语句：DELETE FROM courses WHERE id=...

执行SQL语句并检查执行结果

如果操作类型为 "查询"：

输入查询条件（如课程名或讲师名）

构造SQL语句：SELECT \* FROM courses WHERE ...

执行SQL语句并返回查询结果

最后返回操作结果。

**核心算法 3：班级信息管理算法**

实现班级信息的增删改查功能。管理班级信息先输入操作类型（添加、修改、删除、查询），然后根据操作类型执行相应逻辑：

如果操作类型为 "添加"：

输入班级信息（班级ID、名称、人数等）

检查输入合法性

构造SQL语句：INSERT INTO classes VALUES (...)

执行SQL语句并检查执行结果

如果操作类型为 "修改"：

输入班级ID和需要修改的信息

构造SQL语句：UPDATE classes SET ... WHERE id=...

执行SQL语句并检查执行结果

如果操作类型为 "删除"：

输入班级ID

构造SQL语句：DELETE FROM classes WHERE id=...

执行SQL语句并检查执行结果

如果操作类型为 "查询"：

输入查询条件（如班级名或人数范围）

构造SQL语句：SELECT \* FROM classes WHERE ...

执行SQL语句并返回查询结果

最后返回操作结果。

**核心算法 4：综合查询算法**

可以根据学生、班级、课程三者关系进行综合查询。综合查询先输入查询条件（如学生ID、班级ID或课程ID），然后构造联合查询SQL语句，通过 JOIN 操作将 students、classes 和 courses 三张表连接起来。采用条件筛选（where）过滤符合条件的记录：

SELECT s.name, c.name, cl.name

FROM students s

JOIN classes cl ON s.class\_id = cl.id

JOIN courses c ON cl.id = c.class\_id

WHERE ...

接着执行SQL语句并获取查询结果，最后返回结果。  
对结果进行处理并返回给用户。如果查询到数据，返回学生姓名、班级名称和课程名称等信息。

如果查询无结果，返回“未找到符合条件的记录”。该流程如图2- 27所示。

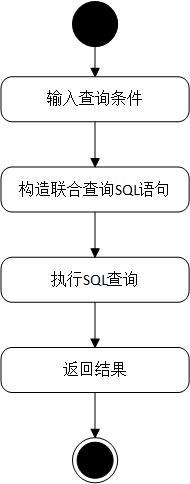


图2-27 综合查询算法流程图

**核心算法 5：用户登录验证算法**

可以验证用户身份，根据用户名和密码允许或拒绝访问系统。进行用户登录验证时，先输入用户名和密码，然后构造SQL查询语句：

SELECT \* FROM users WHERE username = ? AND password = ?

接着执行SQL语句并获取查询结果。如果查询结果为空，返回 "用户名或密码错误"。否则，验证用户权限级别（如管理员、普通用户）并返回 "登录成功" 和用户权限信息。

**核心算法 6：异常处理算法**

在操作数据库时处理各种异常，例如SQL语句错误、数据库连接失败等。

异常处理，当尝试执行操作（如SQL查询、数据库更新）时，如果操作成功，返回操作结果。如果操作失败，程序捕获异常，打印异常信息，返回友好的错误提示（如 "数据库连接失败，请稍后重试"）。

## 三、系统编码及运行

### （一）系统开发涉及的软件

**1.开发工具**

IntelliJ IDEA支持多种编程语言和框架，其中广泛用于Java应用程序的开发。也可用于不同平台项目的开发，其包含以下几个特点：IntelliJ IDEA提供了智能编码辅助功能，需进行简单选择完成简单的代码自动化编写，使开发人员能够更好的完成编码。

MySQL是一种流行的开源关系型数据库管理系统。它使用结构化查询语言（SQL）进行数据库管理。许多网站和应用程序用它来存储和管理数据，像用户信息、订单数据等。例如一个电商网站，商品信息、用户订单详情和会员账户数据都可以存储在MySQL数据库中。

draw.io是一款在线的绘图工具，能创建多种类型的图表。它有简单易用的界面，支持绘制流程图、组织结构图、UML图等多种图表。用户可以通过简单的拖拽操作添加图形元素和连接线，方便团队协作进行项目规划、流程设计等工作。

git是一个分布式版本控制系统。它可以记录文件的修改历史，方便开发者追踪代码变化。在软件开发中广泛应用，多人协作开发项目时，git能让开发者在自己的分支上工作，然后将修改合并到主分支。比如一个大型软件项目，不同开发人员可以同时开发不同功能模块，最后通过git整合代码。

JDK（Java Development Kit）是Java开发工具包。它包含了编译、运行Java程序的各种工具和Java运行时环境。开发人员用JDK编写Java程序，其中的编译器（javac）把Java源代码转换成字节码，Java虚拟机（JVM）则执行这些字节码。

1. **开发技术**

（1）编程语言与开发框架

使用Java作为编程语言，具有跨平台性、面向对象的特性，适合开发中小型管理系统。代码中使用了 Java 的核心类库以及常见的面向对象设计方法（如类、接口、继承等）。

开发框架使用Java Swing实现了系统的图形用户界面（GUI）。Swing 提供了一系列组件（如 JFrame、JPanel、JButton 等）用于创建用户交互界面。

（2）数据库与持久化技术

系统使用 MySQL 作为数据库，存储用户、数据管理等模块的核心数据。MySQL 具有性能稳定、支持大规模数据存储的优势。

使用JDBC（Java Database Connectivity）作为Java 与数据库交互的核心技术，用于执行 SQL 查询、数据插入、更新和删除操作。通过 JDBC 实现了与 MySQL 的连接，使用 DriverManager 和 Connection 类完成数据库操作。

（3） 核心算法与逻辑实现

对于输入校验与错误处理，通过条件语句和正则表达式对用户输入的数据进行校验，确保数据完整性和准确性。代码实现了对数据库中数据的创建（Create）、读取（Read）、更新（Update）和删除（Delete）功能。使用 JDBC 的 PreparedStatement 避免了 SQL 注入漏洞，并提升了数据库操作的安全性。最后用户身份验证通过比对数据库中的账号与密码字段，验证用户身份，并控制系统访问权限。

（4）系统架构与模块化设计

分层架构：表示层（GUI）： 使用 Swing 开发用户界面，展示数据并接受用户输入；业务逻辑层： 实现了各功能模块的业务逻辑（如登录验证、数据操作等）；数据访问层（DAL）： 使用 JDBC 实现与数据库的交互。

模块化设计：代码中将功能划分为多个模块（如登录模块、数据管理模块、查询模块），便于代码的维护与扩展。

（5）测试与调试

测试数据生成与验证：

编造测试数据，用于验证系统功能是否正常运行，确保系统逻辑正确性。

异常处理：

使用 try-catch 块捕获运行时异常，避免系统崩溃，并提供用户友好的错误提示。

### 系统运行界面及结果

**1.用户登录界面：**当程序运行时，弹出一个登录界面，可以登录用户或者注册用户，如果登录用户名或密码不对，注册时用了已有的用户名，则会显示错误。如图3-1所示。

图3-1系统登录界面

**2.系统界面：**当用户登录成功时会进入主界面，里面有学生管理，班级管理，院系管理，课程管理，成绩管理，用户管理，关于，退出功能，可以进行录入、查询、修改删除的操作。如图3-2所示。

图3-2 系统界面

**3.添加用户界面：**  
当管理员想要添加用户时，可以点用户管理按钮然后添加用户，如果用户名已存在或者两次输入的密码错误，则显示错误。如图3-3所示。

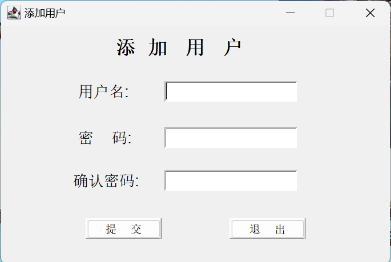


图3-3 添加用户界面

**4.学生信息录入界面：**  
当管理员想要录入学生信息时，可以点学生管理按钮，然后录入学号，姓名，所属学院，所属专业，所在班级。如图3-4所示。

图3-4 学生信息录入界面

1. **学生信息管理界面：**学生和管理员都可以查看学生信息管理界面，可以按学号，姓名，班级进行查询学生的信息，不同的是，只有管理员可以进行修改和删除。如图3-5所示。



图3-5 学生信息管理界面

**6.班级信息管理界面：**用户和管理员都可以进入班级信息管理这个界面，可以按学院查询，按专业查询，按名称查询，但只有管理员可以进行删除和修改。如图3-6所示。

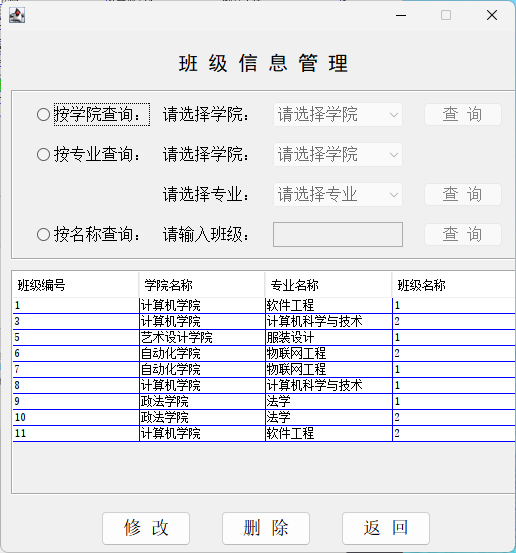


图3-6 班级信息管理界面

**7.院系信息管理界面：**用户和管理员都可以查看院系的信息，但是只有管理员可以修改和删除院系信息。如图3-7所示。

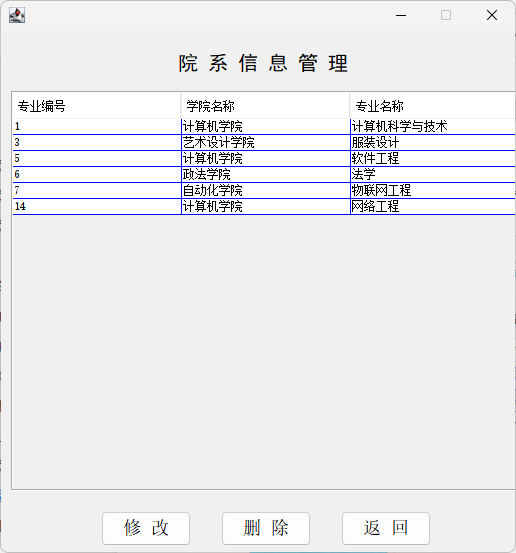


图3-7 院系管理界面

**8.课程管理界面：**用户和管理员都可以查看课程信息，但只有管理员可以修改和删除。如图3-8所示。



图3-8 课程管理界面

**9.用户密码修改界面：**当用户想要修改密码时，可以点用户管理按钮，然后进行用户密码修改，如果两次输入的密码不同时会提示错误。如图3-9所示。

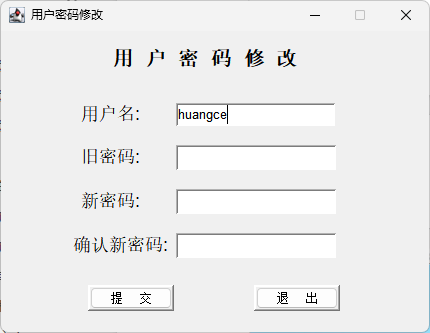


图3-9 用户密码修改界面

**10.关于信息界面：**

界面信息中可以看该系统名称，及版本号，该系统的作者和在github链接。如图3-10所示。



图3-10 关于信息界面

### 系统测试

**1.主要测试数据及测试结果**

（1）用户功能模块测试

测试验证了学生信息管理系统的主要功能，包括学生信息查看、班级信息查看、成绩查询和用户密码修改等，系统整体运行稳定，功能实现符合预期。测试中发现SQL查询条件与数据库字段不匹配的问题，导致院系信息查询结果为空，已通过修正查询语句解决。系统整体表现良好，功能逻辑完善，后续可进一步加强边界条件测试和查询优化以提升鲁棒性与用户体验。测试结果如表3-1用户功能测试表所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试功能 | 测试数据 | 预期结果 | 实际结果 | Bug 描述 | 解决方法 |
| 001 | 查看学生信息 | 用户A登录，查询学生学号 20230001 | 显示学号为20230001的学生详细信息 | 正常显示 | 无 | 无 |
| 002 | 查看班级信息 | 用户B登录，查询班级名称为 "计算机科学2023" 的信息 | 显示该班级学生人数及详细信息 | 正常显示 | 无 | 无 |
| 003 | 查看班级信息 | 用户B登录，查询班级名称为 "计算机科学2023" 的信息 | 返回该学生所有课程及其成绩 | 正常显示 | 无 | 无 |
| 004 | 查看成绩信息 | 用户C查询学号 20230001 的成绩信息 | 提示密码修改成功 | 正常显示 | 无 | 无 |
| 005 | 查看院系信息 | 用户D修改当前用户密码为 password123 | 显示院系包含的所有专业、班级等信息 | 返回结果为空 | 查询条件错误，未与数据库字段匹配 | 修正SQL查询语句，确保字段名与数据库一致。 |

表3-1用户功能测试表

1. 管理员功能模块测试

测试验证了学生信息管理系统的管理员功能模块，包括学生信息录入、班级信息录入、用户删除、成绩录入和学生信息删除等功能。大部分功能运行正常，但发现部分问题。整体系统功能稳定，逻辑较完善，但需加强数据一致性和约束条件设计。测试结果如表3-2管理员功能测试表所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试功能 | 测试数据 | 预期结果 | 实际结果 | Bug 描述 | 解决方法 |
| 101 | 学生信息录入 | 添加学生：学号20230002，姓名张三，班级计算机科学2023 | 提示录入成功学2023 | 提示录入成功 | 无 | 无 |
| 102 | 班级信息录入 | 添加班级：班级名称软件工程2023，所属专业软件工程 | 提示录入成功 | 提示"重复班级名称" | 数据库字段未设置唯一约束 | 修改数据库表，增加className字段的唯一约束条件。 |
| 103 | 删除用户 | 删除用户：用户ID为1001 | 提示删除成功 | 提示"外键约束，无法删除" | 用户被成绩表或学生表引用 | 删除前增加级联操作或检查外键依赖，并允许管理员确认删除操作。 |
| 104 | 成绩信息录入 | 添加成绩：学号20230001，课程高等数学，成绩90 | 提示录入成功 | 成绩未成功录入 | 数据库成绩表未对学号+课程设置唯一约束 | 增加stuNumber和courceName联合唯一索引，避免重复数据。 |
| 105 | 删除学生信息 | 删除学生：学号20230003 | 提示删除成功 | 正常删除 | 无 | 无 |

表3-2 管理员功能测试表

**2.Bug 总结与解决方法**

（1）java与mysql连接测试连接时，显示密码错误。解决方法如下：

在db包中将url更改为正确的路径，将user更改为正确的用户名，将pwd改为数据库的密码。

String url = "jdbc:mysql://localhost:3306/stuManagerDB";

String user = "root";

String pwd = "123456";  
将url（路径），user（用户名），pwd（密码）改为正确的时候，接下来在IDEA中下载MySQL驱动。  
下一步进行驱动加载：  
Class.forName("com.mysql.cj.jdbc.Driver");  
如图3-11所示。

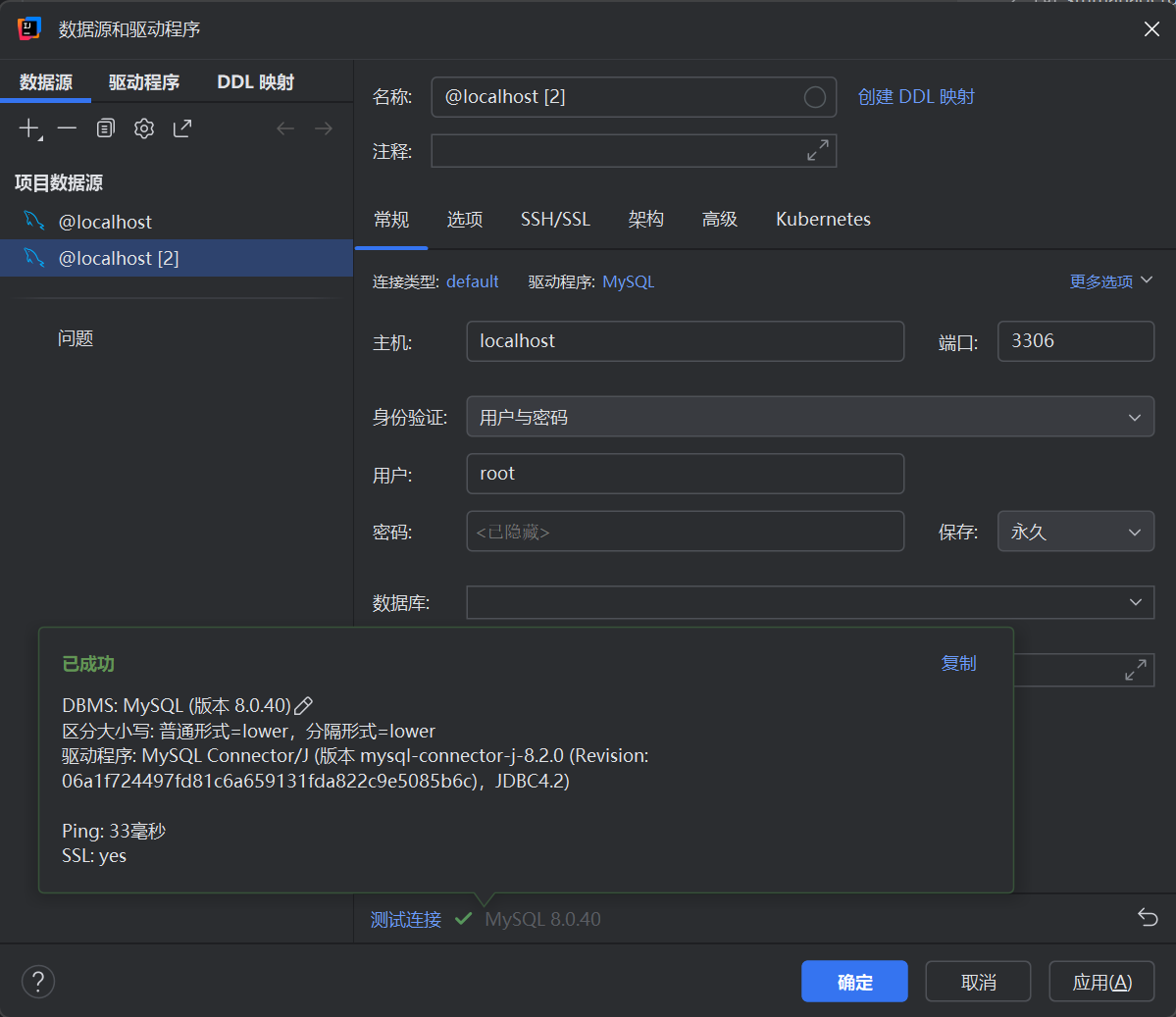


图3-11 驱动加载页面

（2）进入登陆界面后，不管是用admin直接登录还是注册新用户，IDEA都会报NullPointException错误。解决方法如下：

在DB中添加了admin数据或者其他用户数据，再次测试时仍然抛出空指针异常错误，接下来测试java与db是否连接，测试结果为java与db连接未成功，分析问题通常是由于数据库连接或相关依赖未正确配置导致的，在mysql官网上下载mysql-connector-j-9.0.0.jar，并且添加到项目模块依赖。界面如图3-3-2所示。

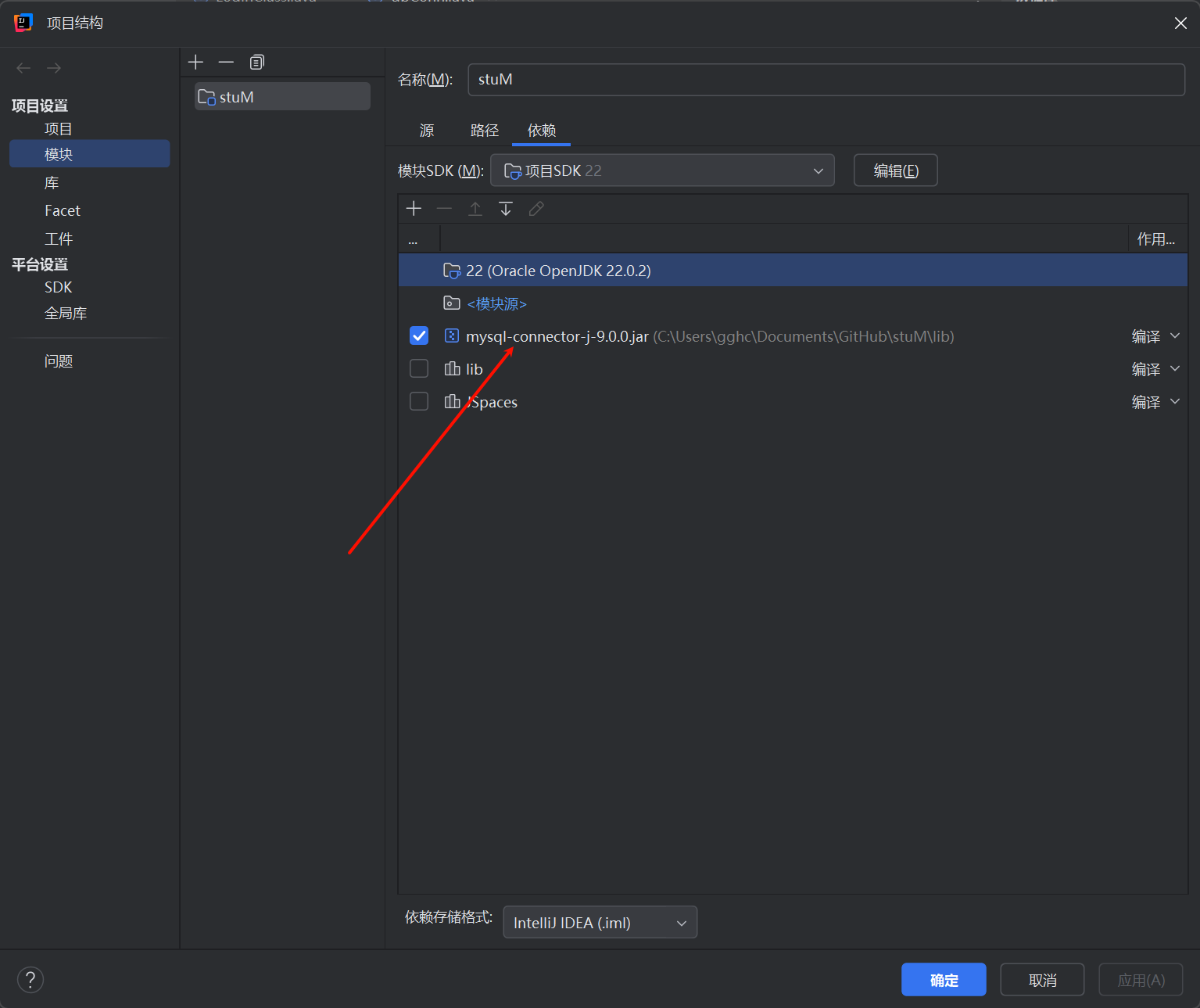


图3-3-2 添加到项目模块页面

（3）用户模块查询条件错误。查询院系信息时条件与数据库字段不匹配，返回结果为空。解决方法： 检查SQL语句字段名，确保与数据库一致。然后在查询接口中增加输入参数的验证与提示。

（4）班级信息重复录入。班级名称未设置唯一约束，可能导致重复数据录入。解决方法：修改tb\_class表，在className字段上添加唯一约束。然后在后端逻辑中添加唯一性校验，防止重复提交。

（5）用户删除外键约束冲突。用户删除时，存在外键约束，导致删除失败。解决方法： 增加删除确认操作提示用户检查依赖。然后在数据库设计时添加级联删除规则（如ON DELETE CASCADE）。

（6）成绩信息重复录入。成绩信息录入时，学号与课程未设置唯一约束，导致重复成绩记录。解决方法： 修改tb\_score表，设置stuNumber和courceName为联合唯一键。然后在后端逻辑中校验是否存在重复记录。

通过测试发现并修复Bug，完善了系统的稳定性和功能完整性，提高了开发质量。未来在开发中，将注重数据一致性、约束条件和接口设计的健壮性，避免类似问题的发生。

## 五、总结

一、学生信息系统实现的功能

（一）用户权限管理

1. 多角色设定：系统管理员不仅能对各类用户账号进行创建、删除、修改密码等基础操作，还能灵活配置系统参数，保障系统稳定高效运行；学生作为系统的核心关注对象，被赋予了有限但实用的权限，仅限查看个人学业详情、课程安排，以及进行选课等自主性操作，充分保护个人隐私；

（二）学生信息管理

1. 全面信息录入：录入学生的基本信息，方便学校对学生对信息采集为学校全方位了解学生提供了“全景画像”，更方便学校实行管理。

2. 灵活信息查询与修改：支持多维度查询方式，按学号查询瞬间定位个体，姓名查询配备模糊检索功能。当学生个人信息发生变动，如家庭住址搬迁、手机号码更换，表中内容会自动修改，确保信息时效性。

（三）班级信息管理

1.班级基础信息维护：详细记录班级名称、班级编号、所属院系、入学年份、班主任等核心信息，为学校教学组织架构奠定基石。清晰的班级归属与责任人明确，方便任务下达、沟通协调，确保班级管理有条不紊。

2. 班级动态管理：实时追踪班级人数变化、学生流动情况，如转学进出、分班合班动态，自动更新相关数据，让学校能精准把握教育资源分配节奏，依据班级实际情况灵活调整师资、教室等资源。

（四）院系信息管理

1. 院系架构呈现：完整勾勒出院系层级，从学院名称、学院代码，到下属各系、专业设置，再到系主任、专业负责人等关键人物信息。

2. 院系数据统计与分析：能够对院系内学生数量、教师配备、专业招生趋势等数据进行深度挖掘，以可视化图表输出，为院系发展规划、资源争取提供数据硬支撑，助力院系在学科建设、人才培养上扬帆远航。

（五）课程管理

1. 课程基本信息录入：课程名称、课程代码、学分设置、授课教师、上课时间、上课地点、课程大纲等信息逐一录入系统。精准的课程信息是学生选课、教师授课、教学排期的关键指引，确保教学秩序井然。

2. 选课管理优化：在选课窗口期，系统实时显示课程余量、选课学生名单，结合学生专业限制、学分要求，智能引导学生合理选课，避免冲突，实现课程资源的科学配置，提升学生学习满意度。

（六）成绩管理

成绩多元录入：教师能将成绩准确无误地录入系统，录入时自动校验数据格式、分值范围，确保成绩真实性。

### 二、系统项目开发过程中的收获

1. 技术技能进阶

在开发过程中，深入学习并熟练运用多种编程语言及其框架。后端开发选用 Java 的 Web 开发框架，掌握其路由配置、数据库连接、业务逻辑处理等核心技术，实现系统高效稳定的服务器端功能；通过前后端的协同开发，打通数据交互通道，对全栈开发流程有了清晰认知，能独立完成从界面设计到数据存储读取的完整功能模块开发。

精通关系型数据库（如 MySQL、Oracle）的设计原则，根据系统需求设计合理的表结构，建立学生、课程、成绩、教师等实体间的关联关系（如通过外键实现一对多、多对多关系），优化查询语句以提高数据检索效率，掌握数据库索引、视图、存储过程等高级特性的运用，解决数据一致性、完整性、并发控制等问题，确保在大数据量存储与频繁读写操作下系统性能不受影响，同时学会数据库的日常维护操作，如数据备份、优化、迁移等，保障数据安全可靠。

2. 项目管理与协作素养

经过两周的课程学习，收集整理系统需求，撰写详细的需求规格说明书，将模糊的业务需求转化为具体的功能模块、数据指标、操作流程，运用思维导图、流程图等工具梳理系统架构与业务逻辑，确定项目的整体范围、开发周期、里程碑节点，为后续开发提供明确指导，避免盲目开发导致返工。

开发团队成员在项目推进过程中，建立高效的沟通机制，每日站会汇报工作进度、遇到的问题，及时协调资源解决；通Git进行代码版本管理，分支开发、合并代码，解决代码冲突，确保团队成员并行工作不冲突；协同测试人员完成系统测试，根据反馈的 BUG 及时修复，在不断的交互协作中，学会理解他人工作难点，发挥自身优势，提升团队凝聚力与战斗力。

将开发任务分解到个人、天，跟踪任务进度，及时发现并解决项目延误风险，通过设置代码审查环节、编写单元测试用例、进行集成测试、性能测试等多阶段测试流程，严格把控系统质量，确保交付的系统功能完备、性能稳定、安全可靠，满足学校教学管理实际需求。

3. 问题解决与应变思维

开发中遭遇诸多技术挑战，如大数据量下系统响应迟缓，通过数据库优化（查询缓存、索引优化、分库分表）、代码算法优化（采用异步处理、缓存技术）等措施解决。

### 三、未来专业学习的目标

1. 技术深耕方向

人工智能与大数据在教育领域应用：随着教育信息化的深入发展，计划学习人工智能技术中的机器学习、深度学习算法，将其应用于学生学习行为分析，如通过分析学生的课堂表现、作业完成情况、考试成绩走势等多源数据，预测学生的学习困难、辍学风险，为教师提前干预提供精准依据；探索大数据技术在教育管理中的运用，构建教育数据仓库，整合校内校外教育资源数据，挖掘数据价值，辅助学校进行专业设置优化、教学资源配置决策，实现智能化教育管理。

移动开发与跨平台技术：考虑到师生对移动端便捷访问系统的需求日益增长，学习移动开发技术，如 Android、iOS 原生开发，以及跨平台开发框架（如 Flutter、React Native），将学生信息管理系统拓展到移动端，实现随时随地查询信息、提交作业、进行考勤签到等功能，提升系统的可用性与覆盖面，打造一体化的智慧教育移动端生态。

2. 实践创新与行业引领

参与开源教育项目与竞赛：积极投身开源教育软件社区，参与开源学生信息管理项目的开发、维护，与全球开发者交流协作，吸收先进理念与技术，提升个人在行业内的知名度；参加各类教育信息化相关竞赛，如全国大学生计算机设计大赛、“互联网 +”创新创业大赛的教育赛道等，将创新想法付诸实践，以赛促学，在竞赛中打磨项目，争取获得优异成绩，为未来职业发展积累荣誉资本。

推动教育信息化标准制定与推广：关注教育行业信息化标准的发展动态，结合自身实践经验，参与地方乃至国家教育信息化标准的制定研讨，如学生信息数据标准、教育管理系统功能规范等，为行业规范化发展贡献力量；在学校、区域内推广先进的教育信息化理念与技术应用，通过举办讲座、培训等方式，提升教育从业者的信息化素养，引领教育行业向数字化、智能化方向迈进。

综上所述，我们小组成员通过参与学生信息管理系统的开发，在技术、项目管理、问题解决等多方面收获颇丰，而面向未来，以技术深耕、跨学科融合、实践创新为路径，设定明确的专业学习目标，力求在教育信息化领域持续发光发热，为教育事业的发展添砖加瓦。

附录

**java:**

**Class:**

**ClassAddFrame:**

package Class;

import java.awt.Dimension;

import java.sql.ResultSet;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.\*;

import javax.swing.JLabel;

import javax.swing.JTextField;

import db.dbConn;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\*

\* Title: 课程添加

\* Description: 实现课程添加的模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*

\*/

public class ClassAddFrame extends JFrame {

JPanel contentPane;

JLabel jLabel1 = new JLabel();

JLabel jLabel2 = new JLabel();

JLabel jLabel3 = new JLabel();

JTextField jTextField1 = new JTextField();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

JComboBox jComboBox1 = new JComboBox();

String xueyuan, zhuanye;

dbConn conn = new dbConn();

JLabel jLabel4 = new JLabel();

JComboBox jComboBox2 = new JComboBox();

public ClassAddFrame() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(465, 280));

setTitle("班级录入");

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel1.setText("班 级 信 息 录 入");

jLabel1.setBounds(new Rectangle(136, 20, 212, 25));

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel2.setText("班级名称:");

jLabel2.setBounds(new Rectangle(80, 80, 90, 20));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel3.setText("所属学院:");

jLabel3.setBounds(new Rectangle(80, 130, 90, 20));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField1.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField1.setBounds(new Rectangle(180, 80, 180, 25));

jButton1.setBounds(new Rectangle(102, 223, 96, 29));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("提 交");

jButton1.addActionListener(new AddClassFrame\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(265, 221, 96, 31));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setToolTipText("");

jButton2.setText("退 出");

jButton2.addActionListener(new AddClassFrame\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(106, 258, 262, 90));

jOptionPane1.setLayout(null);

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setEditable(true);

jComboBox1.setBounds(new Rectangle(180, 130, 180, 25));

jComboBox1.addActionListener(new AddClassFrame\_jComboBox1\_actionAdapter(this));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel4.setText("所属专业：");

jLabel4.setBounds(new Rectangle(80, 180, 90, 20));

jComboBox2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox2.setEditable(true);

jComboBox2.setBounds(new Rectangle(180, 180, 180, 25));

contentPane.add(jLabel1);

contentPane.add(jOptionPane1);

contentPane.add(jComboBox1);

contentPane.add(jLabel3);

contentPane.add(jLabel2);

contentPane.add(jTextField1);

contentPane.add(jLabel4);

contentPane.add(jComboBox2);

contentPane.add(jButton2);

contentPane.add(jButton1);

jComboBox1.addItem("请选择学院");

// 选择学院

try {

ResultSet rs = conn.getRs("select \* from tb\_depart ");

while (rs.next()) {

String xibu = rs.getString("departName");

jComboBox1.addItem(xibu);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

jComboBox2.setEnabled(false);

}

// 选择专业

public void zhuanye() {

jComboBox2.removeAllItems();

jComboBox2.addItem("请选择专业");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where departName='" + String.valueOf(jComboBox1.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("specName");

jComboBox2.addItem(zhy);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 存下选定学院到xueyuan

public void xueYuan() {

String sel = String.valueOf(jComboBox1.getSelectedItem());

try {

ResultSet rs = conn.getRs("select \* from tb\_depart where departName='" + sel + "'");

while (rs.next()) {

xueyuan = rs.getString("departName");

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 存下选定专业到zhuanye

public void zhuanYe() {

String sel = String.valueOf(jComboBox2.getSelectedItem());

try {

ResultSet rs = conn.getRs("select \* from tb\_spec where specName='" + sel + "'");

while (rs.next()) {

zhuanye = rs.getString("specName");

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 提交

public void jButton1\_actionPerformed(ActionEvent e) {

// 异常判断

if (this.jTextField1.getText().trim().length() == 0) {

jOptionPane1.showMessageDialog(this, "班级名称不能为空。", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jComboBox1.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择所属学院！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jComboBox2.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择所属专业！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

xueYuan();

zhuanYe();

try {

boolean classname = false;

ResultSet rs = conn.getRs("select className from tb\_class where specName= '"

+ String.valueOf(jComboBox2.getSelectedItem()) + "' ");

while (rs.next()) {

if (jTextField1.getText().trim().equals(rs.getString("className").trim())) {

classname = true;

}

}

if (classname) {

jOptionPane1.showMessageDialog(this, "该班级名称已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

conn.getUpdate("insert into tb\_class (className,specName, departName) values ('"

+ jTextField1.getText().trim() + "','" + zhuanye + "', '" + xueyuan + "')");

jOptionPane1.showMessageDialog(this, "恭喜您班级信息录入成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

} catch (Exception ce) {

System.out.println(ce);

}

}

}

// 选择学院后触发专业可选

public void jComboBox1\_actionPerformed(ActionEvent e) {

zhuanye();

jComboBox2.setEnabled(true);

}

}

class AddClassFrame\_jComboBox1\_actionAdapter implements ActionListener {

private ClassAddFrame adaptee;

AddClassFrame\_jComboBox1\_actionAdapter(ClassAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox1\_actionPerformed(e);

}

}

class AddClassFrame\_jButton1\_actionAdapter implements ActionListener {

private ClassAddFrame adaptee;

AddClassFrame\_jButton1\_actionAdapter(ClassAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class AddClassFrame\_jButton2\_actionAdapter implements ActionListener {

private ClassAddFrame adaptee;

AddClassFrame\_jButton2\_actionAdapter(ClassAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**ClassChange:**

package Class;

import java.awt.Dimension;

import java.sql.ResultSet;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.\*;

import javax.swing.JLabel;

import javax.swing.JTextField;

import db.dbConn;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\*

\* Title: 班级信息修改

\* Description: 班级信息修改模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class ClassChange extends JFrame {

JPanel contentPane;

JLabel jLabel1 = new JLabel();

JLabel jLabel2 = new JLabel();

JLabel jLabel3 = new JLabel();

JTextField jTextField1 = new JTextField();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

JComboBox jComboBox1 = new JComboBox();

dbConn conn = new dbConn();

JLabel jLabel4 = new JLabel();

JComboBox jComboBox2 = new JComboBox();

String find;

String zhuanye, xueyuan;

public ClassChange(String find) {

this.find = find;

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(465, 280));

setTitle("班级修改");

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel1.setText("班 级 信 息 修 改");

jLabel1.setBounds(new Rectangle(136, 20, 212, 25));

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel2.setText("班级名称:");

jLabel2.setBounds(new Rectangle(80, 80, 90, 20));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel3.setText("所属学院:");

jLabel3.setBounds(new Rectangle(80, 130, 90, 20));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField1.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField1.setBounds(new Rectangle(180, 80, 180, 25));

jButton1.setBounds(new Rectangle(102, 223, 96, 29));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("修 改");

jButton1.addActionListener(new ClassU\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(265, 221, 96, 31));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setToolTipText("");

jButton2.setText("退 出");

jButton2.addActionListener(new ClassU\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(106, 258, 262, 90));

jOptionPane1.setLayout(null);

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setBounds(new Rectangle(180, 130, 180, 25));

jComboBox1.addActionListener(new ClassU\_jComboBox1\_actionAdapter(this));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel4.setText("所属专业：");

jLabel4.setBounds(new Rectangle(80, 180, 90, 20));

jComboBox2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox2.setEditable(true);

jComboBox2.setBounds(new Rectangle(180, 180, 180, 25));

contentPane.add(jLabel1);

contentPane.add(jOptionPane1);

contentPane.add(jComboBox1);

contentPane.add(jLabel3);

contentPane.add(jLabel2);

contentPane.add(jTextField1);

contentPane.add(jLabel4);

contentPane.add(jComboBox2);

contentPane.add(jButton2);

contentPane.add(jButton1);

jComboBox1.addItem("请选择学院");

// 选择学院

try {

ResultSet rs = conn.getRs("select \* from tb\_depart ");

while (rs.next()) {

String xibu = rs.getString("departName");

jComboBox1.addItem(xibu);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

jComboBox2.setEnabled(false);

xianshi();

}

// 选择专业

public void zhuanye() {

jComboBox2.removeAllItems();

jComboBox2.addItem("请选择专业");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where departName='" + String.valueOf(jComboBox1.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("specName");

jComboBox2.addItem(zhy);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 存下选定学院到xueyuan

public void xueYuan() {

String sel = String.valueOf(jComboBox1.getSelectedItem());

try {

ResultSet rs = conn.getRs("select \* from tb\_depart where departName='" + sel + "'");

while (rs.next()) {

xueyuan = rs.getString("departName");

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 存下选定专业到zhuanye

public void zhuanYe() {

String sel = String.valueOf(jComboBox2.getSelectedItem());

try {

ResultSet rs = conn.getRs("select \* from tb\_spec where specName='" + sel + "'");

while (rs.next()) {

zhuanye = rs.getString("specName");

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 修改

public void jButton1\_actionPerformed(ActionEvent e) {

// 异常判断

if (this.jTextField1.getText().trim().length() == 0) {

jOptionPane1.showMessageDialog(this, "班级名称不能为空。", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jComboBox1.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择所属学院！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jComboBox2.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择所属专业！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

xueYuan();

zhuanYe();

try {

// 不允许同一个专业有两个一样的班级

boolean classname = false;

ResultSet rs = conn.getRs("select className from tb\_class where specName='"

+ String.valueOf(jComboBox2.getSelectedItem()) + "'");

while (rs.next()) {

if (jTextField1.getText().trim().equals(rs.getString("className").trim())) {

classname = true;

}

}

if (classname) {

jOptionPane1.showMessageDialog(this, "该班级名称已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

conn.getUpdate(

"update tb\_class set className='" + jTextField1.getText().trim() + "',specName='" + zhuanye

+ "', departName='" + xueyuan + "' where classId='" + Integer.valueOf(find) + "'");

jOptionPane1.showMessageDialog(this, "恭喜您班级信息修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

} catch (Exception ce) {

System.out.println(ce);

}

}

}

// 选择触发

public void jComboBox1\_actionPerformed(ActionEvent e) {

zhuanye();

jComboBox2.setEnabled(true);

}

// 选定行后在修改界面显示选定默认值

public void xianshi() {

if (find == null)

return;

else {

jComboBox2.setEnabled(true);

try {

ResultSet rs = conn.getRs("select \* from tb\_class where classId='" + Integer.valueOf(find) + "'");

while (rs.next()) {

jComboBox1.setSelectedItem(String.valueOf(rs.getString("departName")));

jComboBox2.setSelectedItem(String.valueOf(rs.getString("specName")));

jTextField1.setText(rs.getString("className").trim());

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

}

}

class ClassU\_jComboBox1\_actionAdapter implements ActionListener {

private ClassChange adaptee;

ClassU\_jComboBox1\_actionAdapter(ClassChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox1\_actionPerformed(e);

}

}

class ClassU\_jButton1\_actionAdapter implements ActionListener {

private ClassChange adaptee;

ClassU\_jButton1\_actionAdapter(ClassChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class ClassU\_jButton2\_actionAdapter implements ActionListener {

private ClassChange adaptee;

ClassU\_jButton2\_actionAdapter(ClassChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**ClassManager:**

package Class;

import java.awt.\*;

import javax.swing.JFrame;

import com.borland.jbcl.layout.XYLayout;

import db.dbConn;

import com.borland.jbcl.layout.\*;

import stuManager.MainFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.ResultSet;

import javax.swing.table.DefaultTableModel;

import java.util.\*;

/\*\*

\*

\* Title: 课程管理

\* Description: 实现课程管理模块，包含课程查询，同时是课程修改和删除的入口。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class ClassManager extends JFrame {

XYLayout xYLayout1 = new XYLayout();

JLabel jLabel1 = new JLabel();

JPanel jPanel1 = new JPanel();

XYLayout xYLayout2 = new XYLayout();

JRadioButton jRadioButton1 = new JRadioButton();

JRadioButton jRadioButton2 = new JRadioButton();

JRadioButton jRadioButton3 = new JRadioButton();

ButtonGroup buttonGroup1 = new ButtonGroup();

JLabel jLabel2 = new JLabel();

JLabel jLabel3 = new JLabel();

JLabel jLabel4 = new JLabel();

JComboBox jComboBox1 = new JComboBox();

JComboBox jComboBox2 = new JComboBox();

JTextField jTextField1 = new JTextField();

JLabel jLabel5 = new JLabel();

JComboBox jComboBox3 = new JComboBox();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JButton jButton3 = new JButton();

JScrollPane jScrollPane1 = new JScrollPane();

JTable jTable1 = new JTable();

JButton jButton4 = new JButton();

JButton jButton5 = new JButton();

JButton jButton6 = new JButton();

dbConn conn = new dbConn();

Object[][] arrData = {};

String[] arrField = { "班级编号", "学院名称", "专业名称", "班级名称" };

JOptionPane jOptionPane1 = new JOptionPane();

DefaultTableModel model = new DefaultTableModel();

String sql, find;

int intRow;

public ClassManager() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

getContentPane().setLayout(xYLayout1);

xYLayout1.setWidth(530);

xYLayout1.setHeight(540);

jLabel1.setFont(new java.awt.Font("黑体", Font.PLAIN, 20));

jLabel1.setText("班 级 信 息 管 理");

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel2.setText("请选择学院：");

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel3.setText("请选择学院：");

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel4.setText("请输入班级：");

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox2.setEditable(false);

jComboBox2.addActionListener(new ClassManager\_jComboBox2\_actionAdapter(this));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jRadioButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jRadioButton1.setText("按学院查询：");

jRadioButton1.addItemListener(new ClassManager\_jRadioButton1\_itemAdapter(this));

jRadioButton1.addActionListener(new ClassManager\_jRadioButton1\_actionAdapter(this));

jPanel1.setBorder(BorderFactory.createEtchedBorder());

jPanel1.setLayout(xYLayout2);

jRadioButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jRadioButton2.setText("按专业查询：");

jRadioButton2.addItemListener(new ClassManager\_jRadioButton2\_itemAdapter(this));

jRadioButton2.addActionListener(new ClassManager\_jRadioButton2\_actionAdapter(this));

jRadioButton3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jRadioButton3.setText("按名称查询：");

jRadioButton3.addItemListener(new ClassManager\_jRadioButton3\_itemAdapter(this));

jRadioButton3.addActionListener(new ClassManager\_jRadioButton3\_actionAdapter(this));

jLabel5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel5.setText("请选择专业：");

jComboBox3.setEnabled(false);

jComboBox3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setText("查 询");

jButton1.addActionListener(new ClassManager\_jButton1\_actionAdapter(this));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton2.setText("查 询");

jButton2.addActionListener(new ClassManager\_jButton2\_actionAdapter(this));

jButton3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton3.setText("查 询");

jButton3.addActionListener(new ClassManager\_jButton3\_actionAdapter(this));

jScrollPane1.setBorder(BorderFactory.createEtchedBorder());

jButton4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton4.setText("修 改");

jButton4.addActionListener(new ClassManager\_jButton4\_actionAdapter(this));

jButton5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton5.setText("删 除");

jButton5.addActionListener(new ClassManager\_jButton5\_actionAdapter(this));

jButton6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton6.setText("返 回");

jButton6.addActionListener(new ClassManager\_jButton6\_actionAdapter(this));

jPanel1.add(jRadioButton2, new XYConstraints(20, 50, 125, 25));

jPanel1.add(jRadioButton3, new XYConstraints(20, 130, 125, 25));

jPanel1.add(jRadioButton1, new XYConstraints(20, 10, 125, 25));

buttonGroup1.add(jRadioButton1);

buttonGroup1.add(jRadioButton2);

buttonGroup1.add(jRadioButton3);

jPanel1.add(jLabel3, new XYConstraints(150, 50, 100, 25));

jPanel1.add(jLabel4, new XYConstraints(150, 130, 100, 25));

jPanel1.add(jLabel2, new XYConstraints(150, 10, 100, 25));

jPanel1.add(jComboBox1, new XYConstraints(260, 10, 130, 25));

jPanel1.add(jComboBox2, new XYConstraints(260, 50, 130, 25));

jPanel1.add(jTextField1, new XYConstraints(260, 130, 130, 25));

jPanel1.add(jLabel5, new XYConstraints(150, 90, 100, 25));

jPanel1.add(jComboBox3, new XYConstraints(260, 90, 130, 25));

this.getContentPane().add(jLabel1, new XYConstraints(177, 14, 177, 39));

this.getContentPane().add(jPanel1, new XYConstraints(10, 59, 510, 170));

jPanel1.add(jButton1, new XYConstraints(410, 10, 80, 25));

jPanel1.add(jButton2, new XYConstraints(410, 90, 80, 25));

jPanel1.add(jButton3, new XYConstraints(410, 130, 80, 25));

jScrollPane1.getViewport().add(jTable1);

this.getContentPane().add(jScrollPane1, new XYConstraints(10, 239, 510, 225));

this.getContentPane().add(jButton4, new XYConstraints(100, 480, 90, 35));

this.getContentPane().add(jButton5, new XYConstraints(220, 480, 90, 35));

this.getContentPane().add(jButton6, new XYConstraints(340, 480, 90, 35));

jComboBox1.addItem("请选择学院");

jComboBox2.addItem("请选择学院");

// 用户等级判断

// 1为管理员

if (MainFrame.level.equals("1")) {

// 不隐藏任何功能

}

// 2为普通用户

else if (MainFrame.level.equals("2")) {

// 隐藏部分功能

jButton4.setVisible(false);

jButton5.setVisible(false);

}

// 选择学院

try {

ResultSet rs = conn.getRs("select \* from tb\_depart ");

while (rs.next()) {

String xibu = rs.getString("departName");

jComboBox1.addItem(xibu);

jComboBox2.addItem(xibu);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

jComboBox1.setEnabled(false);

jComboBox2.setEnabled(false);

jComboBox3.setEnabled(false);

jTextField1.setEditable(false);

jButton1.setEnabled(false);

jButton2.setEnabled(false);

jButton3.setEnabled(false);

sql = "select \* from tb\_class";

UpdateRecord();

}

// 选择触发

public void jRadioButton1\_actionPerformed(ActionEvent e) {

jComboBox1.setEnabled(true);

jButton1.setEnabled(true);

}

public void jRadioButton2\_actionPerformed(ActionEvent e) {

jComboBox2.setEnabled(true);

jButton2.setEnabled(true);

}

public void jRadioButton3\_actionPerformed(ActionEvent e) {

jTextField1.setEditable(true);

jButton3.setEnabled(true);

}

public void jRadioButton1\_itemStateChanged(ItemEvent e) {

jComboBox1.setEnabled(false);

jButton1.setEnabled(false);

}

public void jRadioButton2\_itemStateChanged(ItemEvent e) {

jComboBox2.setEnabled(false);

jComboBox3.setEnabled(false);

jButton2.setEnabled(false);

}

public void jRadioButton3\_itemStateChanged(ItemEvent e) {

jTextField1.setEditable(false);

jButton3.setEnabled(false);

}

public void jComboBox2\_actionPerformed(ActionEvent e) {

jComboBox3.setEnabled(true);

jcb();

}

public void jButton1\_actionPerformed(ActionEvent e) {

sql = "select \* from tb\_class where departName='" + String.valueOf(jComboBox1.getSelectedItem()) + "'";

UpdateRecord();

}

public void jButton2\_actionPerformed(ActionEvent e) {

sql = "select \* from tb\_class where specName='" + String.valueOf(jComboBox3.getSelectedItem()) + "'";

UpdateRecord();

}

public void jButton3\_actionPerformed(ActionEvent e) {

sql = "select \* from tb\_class where className='" + jTextField1.getText() + "'";

UpdateRecord();

}

// 班级修改

public void jButton4\_actionPerformed(ActionEvent e) {

getM();

if (intRow == -1) {

jOptionPane1.showMessageDialog(this, "请选择要修改的班级！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

ClassChange siadd = new ClassChange(find);

siadd.setLocation(400, 200);

siadd.setSize(465, 310);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

this.dispose();

}

}

// 班级删除

public void jButton5\_actionPerformed(ActionEvent e) {

getM();

if (intRow == -1) {

jOptionPane1.showMessageDialog(this, "请选择要删除的班级！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

try {

conn.getUpdate("delete from tb\_class where classId='" + Integer.valueOf(find) + "'");

jOptionPane1.showMessageDialog(this, "删除成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

UpdateRecord();

}

// 退出

public void jButton6\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 选择专业

public void jcb() {

jComboBox3.removeAllItems();

jComboBox3.addItem("请选择专业");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where departName='" + String.valueOf(jComboBox2.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("specName");

jComboBox3.addItem(zhy);

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 更新表格

public void UpdateRecord() {

Object[][] arrTmp = {};

Vector vec = new Vector(1, 1);

model = new DefaultTableModel(arrTmp, arrField);

jTable1 = new JTable(model);

jScrollPane1.getViewport().add(jTable1, null);

try {

ResultSet rs3 = conn.getRs(sql);

while (rs3.next()) {

vec = new Vector();

vec.add(String.valueOf(rs3.getInt("classId")));

vec.add(rs3.getString("departName").trim());

vec.add(rs3.getString("specName"));

vec.add(rs3.getString("className"));

model.addRow(vec);

}

rs3.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

jScrollPane1.getHorizontalScrollBar();

jTable1.setGridColor(Color.blue);

jTable1.setDragEnabled(true);

jTable1.setSelectionForeground(Color.red);

jTable1.setSelectionBackground(Color.green);

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.setRowSelectionAllowed(true);

jTable1.setShowVerticalLines(true);

}

// 获取选定的行

public void getM() {

intRow = jTable1.getSelectedRow();

if (intRow == -1) {

return;

}

try {

find = model.getValueAt(intRow, 0).toString().trim();

} catch (Exception e) {

e.printStackTrace();

}

}

}

class ClassManager\_jButton6\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jButton6\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton6\_actionPerformed(e);

}

}

class ClassManager\_jButton5\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jButton5\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton5\_actionPerformed(e);

}

}

class ClassManager\_jButton4\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jButton4\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton4\_actionPerformed(e);

}

}

class ClassManager\_jButton3\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jButton3\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton3\_actionPerformed(e);

}

}

class ClassManager\_jButton2\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jButton2\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

class ClassManager\_jButton1\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jButton1\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class ClassManager\_jRadioButton1\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jRadioButton1\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jRadioButton1\_actionPerformed(e);

}

}

class ClassManager\_jRadioButton1\_itemAdapter implements ItemListener {

private ClassManager adaptee;

ClassManager\_jRadioButton1\_itemAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void itemStateChanged(ItemEvent e) {

adaptee.jRadioButton1\_itemStateChanged(e);

}

}

class ClassManager\_jRadioButton2\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jRadioButton2\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jRadioButton2\_actionPerformed(e);

}

}

class ClassManager\_jRadioButton2\_itemAdapter implements ItemListener {

private ClassManager adaptee;

ClassManager\_jRadioButton2\_itemAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void itemStateChanged(ItemEvent e) {

adaptee.jRadioButton2\_itemStateChanged(e);

}

}

class ClassManager\_jRadioButton3\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jRadioButton3\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jRadioButton3\_actionPerformed(e);

}

}

class ClassManager\_jRadioButton3\_itemAdapter implements ItemListener {

private ClassManager adaptee;

ClassManager\_jRadioButton3\_itemAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void itemStateChanged(ItemEvent e) {

adaptee.jRadioButton3\_itemStateChanged(e);

}

}

class ClassManager\_jComboBox2\_actionAdapter implements ActionListener {

private ClassManager adaptee;

ClassManager\_jComboBox2\_actionAdapter(ClassManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox2\_actionPerformed(e);

}

}

**Cource:**

**CourceAddFrame:**

package Cource;

import java.awt.Dimension;

import java.sql.ResultSet;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.\*;

import javax.swing.JLabel;

import javax.swing.JTextField;

import db.dbConn;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\*

\* Title: 课程录入

\* Description: 课程录入模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class CourceAddFrame extends JFrame {

JPanel contentPane;

JLabel jLabel1 = new JLabel();

int FS, spid, FC;

JLabel jLabel3 = new JLabel();

JLabel jLabel4 = new JLabel();

String IS;

JTextField jTextField2 = new JTextField();

JTextField jTextField3 = new JTextField();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

JLabel jLabel6 = new JLabel();

JComboBox jComboBox1 = new JComboBox();

dbConn sta = new dbConn();

public CourceAddFrame() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(482, 300));

setTitle("课程录入");

jLabel1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jLabel1.setText("课 程 录 入");

jLabel1.setBounds(new Rectangle(178, 17, 126, 25));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel3.setText("课程名称：");

jLabel3.setBounds(new Rectangle(75, 70, 81, 21));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel4.setText("学 分：");

jLabel4.setBounds(new Rectangle(75, 170, 77, 21));

jTextField2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField2.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField2.setText("");

jTextField2.setBounds(new Rectangle(190, 70, 184, 26));

jTextField3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jTextField3.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField3.setText("");

jTextField3.setBounds(new Rectangle(190, 170, 94, 26));

jButton1.setBounds(new Rectangle(103, 240, 90, 29));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("提 交");

jButton1.addActionListener(new CourceAddFrame\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(277, 240, 90, 29));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setText("退 出");

jButton2.addActionListener(new CourceAddFrame\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(75, 261, 262, 90));

jOptionPane1.setLayout(null);

jLabel6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel6.setText("所属专业：");

jLabel6.setBounds(new Rectangle(75, 120, 82, 21));

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setBounds(new Rectangle(190, 120, 160, 25));

contentPane.add(jLabel1);

contentPane.add(jButton1);

contentPane.add(jButton2);

contentPane.add(jOptionPane1);

contentPane.add(jTextField3);

contentPane.add(jLabel4);

contentPane.add(jLabel3);

contentPane.add(jTextField2);

contentPane.add(jComboBox1);

contentPane.add(jLabel6);

jComboBox1.addItem("请选择");

// 选择专业

try {

ResultSet rs = sta.getRs("select \* from tb\_spec");

while (rs.next()) {

String xibu = rs.getString("specName");

jComboBox1.addItem(xibu);

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 录入

public void InC() {

// 异常判断

if (jTextField2.getText().length() == 0) {

jOptionPane1.showMessageDialog(this, "课程名称不能为空！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jComboBox1.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择课程所属系部！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jTextField3.getText().length() == 0) {

jOptionPane1.showMessageDialog(this, "课程学分不能为空！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

try {

boolean name = false;

ResultSet rs = sta.getRs("select courceName from tb\_cource");

while (rs.next()) {

if (jTextField2.getText().trim().equals(rs.getString("courceName").trim())) {

name = true;

}

}

if (name) {

jOptionPane1.showMessageDialog(this, "课程名称已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

sta.getUpdate("insert into tb\_cource (courceName,courceHour,courceSpecName) values ('"

+ jTextField2.getText().trim() + "','" + Float.valueOf(jTextField3.getText().trim()) + "','"

+ String.valueOf(jComboBox1.getSelectedItem()) + "')");

jOptionPane1.showMessageDialog(this, "课程信息提交成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

rs.close();

} catch (Exception ce) {

System.out.println("--------" + ce);

}

}

}

public void jButton1\_actionPerformed(ActionEvent e) {

InC();

}

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

}

class CourceAddFrame\_jButton1\_actionAdapter implements ActionListener {

private CourceAddFrame adaptee;

CourceAddFrame\_jButton1\_actionAdapter(CourceAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class CourceAddFrame\_jButton2\_actionAdapter implements ActionListener {

private CourceAddFrame adaptee;

CourceAddFrame\_jButton2\_actionAdapter(CourceAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**CourceChange:**

package Cource;

import java.awt.Font;

import javax.swing.JFrame;

import com.borland.jbcl.layout.XYLayout;

import db.dbConn;

import com.borland.jbcl.layout.\*;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JComboBox;

import javax.swing.JButton;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

import javax.swing.\*;

/\*\*

\*

\* Title: 课程修改

\* Description: 课程修改模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class CourceChange extends JFrame {

XYLayout xYLayout1 = new XYLayout();

JLabel jLabel1 = new JLabel();

JLabel jLabel2 = new JLabel();

JLabel jLabel3 = new JLabel();

JLabel jLabel5 = new JLabel();

JTextField jTextField1 = new JTextField();

JComboBox jComboBox1 = new JComboBox();

JTextField jTextField2 = new JTextField();

JButton jButton1 = new JButton();

JButton jButton3 = new JButton();

dbConn sta = new dbConn();

JOptionPane jOptionPane1 = new JOptionPane();

int csid;

int kefind;

public CourceChange(int find) {

kefind = find;

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

getContentPane().setLayout(xYLayout1);

xYLayout1.setWidth(500);

xYLayout1.setHeight(350);

jLabel1.setFont(new java.awt.Font("新宋体", Font.BOLD, 20));

jLabel1.setText("课 程 修 改 ");

jButton3.addActionListener(new CourceManager\_jButton3\_actionAdapter(this));

jButton1.addActionListener(new CourceManager\_jButton1\_actionAdapter(this));

jTextField2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton3.setText("退 出");

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setText("修 改");

jLabel5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel5.setText("学 分：");

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel3.setText("所属专业：");

this.getContentPane().add(jComboBox1, new XYConstraints(200, 130, 140, 30));

this.getContentPane().add(jTextField2, new XYConstraints(200, 180, 140, 30));

this.getContentPane().add(jLabel2, new XYConstraints(85, 80, 80, 30));

this.getContentPane().add(jLabel5, new XYConstraints(85, 180, 80, 30));

this.getContentPane().add(jLabel3, new XYConstraints(85, 130, 80, 30));

this.getContentPane().add(jTextField1, new XYConstraints(200, 80, 140, 30));

this.getContentPane().add(jLabel1, new XYConstraints(158, 21, -1, 49));

this.getContentPane().add(jButton1, new XYConstraints(130, 230, 90, 30));

this.getContentPane().add(jButton3, new XYConstraints(295, 230, 90, 30));

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel2.setText("课程名称：");

this.setTitle("课程修改与删除");

jComboBox1.addItem("请选择");

// 选择专业

try {

ResultSet rs = sta.getRs("select \* from tb\_spec");

while (rs.next()) {

String xibu = rs.getString("specName");

jComboBox1.addItem(xibu);

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

setF();

}

// 选定行后在修改界面显示选定默认值

public void setF() {

if (kefind < 0)

return;

else {

try {

ResultSet rs = sta.getRs("select \* from tb\_cource where courceId='" + Integer.valueOf(kefind) + "'");

while (rs.next()) {

jComboBox1.setSelectedItem(String.valueOf(rs.getString("courceSpecName")));

jTextField1.setText(rs.getString("courceName").trim());

jTextField2.setText(rs.getString("courceHour").trim());

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

}

// 退出

public void jButton3\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 修改

public void jButton1\_actionPerformed(ActionEvent e) {

String keCName, zyName, tName, xueFen;

keCName = jTextField1.getText().trim();

zyName = String.valueOf(jComboBox1.getSelectedItem());

xueFen = jTextField2.getText().trim();

try {

sta.getUpdate("update tb\_cource set courceSpecName='" + zyName + "', courceName='" + keCName

+ "',courceHour='" + Float.valueOf(xueFen) + "' where courceId='" + kefind + "'");

jOptionPane1.showMessageDialog(this, "课程修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} catch (Exception a) {

System.out.println(a.getMessage());

}

}

}

class CourceManager\_jButton1\_actionAdapter implements ActionListener {

private CourceChange adaptee;

CourceManager\_jButton1\_actionAdapter(CourceChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class CourceManager\_jButton3\_actionAdapter implements ActionListener {

private CourceChange adaptee;

CourceManager\_jButton3\_actionAdapter(CourceChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton3\_actionPerformed(e);

}

}

**CourceManager:**

package Cource;

import java.awt.Font;

import com.borland.jbcl.layout.XYLayout;

import db.dbConn;

import com.borland.jbcl.layout.\*;

import stuManager.MainFrame;

import javax.swing.\*;

import java.sql.ResultSet;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.table.DefaultTableModel;

import java.util.\*;

import java.awt.\*;

/\*\*

\*

\* Title: 课程管理

\* Description: 课程管理模块，实现课程查询，同时也是课程修改、删除的入口。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class CourceManager extends JFrame {

XYLayout xYLayout1 = new XYLayout();

JLabel jLabel1 = new JLabel();

JScrollPane jScrollPane1 = new JScrollPane();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

XYLayout xYLayout2 = new XYLayout();

JTable jTable1 = new JTable();

ButtonGroup buttonGroup1 = new ButtonGroup();

dbConn sta = new dbConn();

String sql;

Object[][] arrData = {};

String[] arrField = { "课程编号", "专业名称", "课程名称", "课程学分" };

DefaultTableModel model = new DefaultTableModel();

int intRow;

static int find;

JOptionPane jOptionPane1 = new JOptionPane();

JButton jButton6 = new JButton();

public CourceManager() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

getContentPane().setLayout(xYLayout1);

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel1.setText("课 程 管 理");

jScrollPane1.setBorder(BorderFactory.createEtchedBorder());

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton1.setText("修 改");

jButton1.addActionListener(new CourceF\_jButton1\_actionAdapter(this));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton2.setText("返 回");

jButton2.addActionListener(new CourceF\_jButton2\_actionAdapter(this));

xYLayout1.setWidth(550);

xYLayout1.setHeight(560);

this.setTitle("课程管理");

jButton6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton6.setText("删 除");

jButton6.addActionListener(new CourceF\_jButton6\_actionAdapter(this));

jScrollPane1.getViewport().add(jTable1);

this.getContentPane().add(jLabel1, new XYConstraints(208, 13, 135, 43));

this.getContentPane().add(jButton1, new XYConstraints(74, 480, 100, -1));

this.getContentPane().add(jButton2, new XYConstraints(374, 480, 100, -1));

this.getContentPane().add(jScrollPane1, new XYConstraints(18, 60, 490, 400));

this.getContentPane().add(jButton6, new XYConstraints(224, 480, 100, -1));

// 用户等级判断

// 1为管理员

if (MainFrame.level.equals("1")) {

// 不隐藏任何功能

}

// 2为普通用户

else if (MainFrame.level.equals("2")) {

// 隐藏部分功能

jButton1.setVisible(false);

jButton6.setVisible(false);

}

sql = "select \* from tb\_cource ";

UpdateRecord();

}

// 查询显示

public void UpdateRecord() {

Object[][] arrTmp = {};

Vector vec = new Vector(1, 1);

model = new DefaultTableModel(arrTmp, arrField);

jTable1 = new JTable(model);

jScrollPane1.getViewport().add(jTable1, null);

try {

ResultSet rs = sta.getRs(sql);

int i = 1;

while (rs.next()) {

vec = new Vector();

vec.add(String.valueOf(rs.getString("courceId")));

vec.add(String.valueOf(rs.getString("courceSpecName")));

vec.add(String.valueOf(rs.getString("courceName")));

vec.add(String.valueOf(rs.getString("courceHour")));

model.addRow(vec);

}

rs.close();

} catch (Exception ce) {

System.out.println(ce);

}

jScrollPane1.getHorizontalScrollBar();

jTable1.setGridColor(Color.blue);

jTable1.setDragEnabled(true);

jTable1.setSelectionForeground(Color.red);

jTable1.setSelectionBackground(Color.green);

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.setRowSelectionAllowed(true);

jTable1.setShowVerticalLines(true);

}

// 获取选定的行

public void getM() {

intRow = jTable1.getSelectedRow();

if (intRow == -1) {

jOptionPane1.showMessageDialog(this, "请选择要修改的课程！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

return;

}

try {

find = Integer.parseInt(model.getValueAt(intRow, 0).toString());

System.out.println(find);

} catch (Exception e) {

e.printStackTrace();

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 修改

public void jButton1\_actionPerformed(ActionEvent e) {

getM();

if (find > 0) {

CourceChange siadd = new CourceChange(find);

siadd.setLocation(400, 200);

siadd.setSize(465, 310);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

this.dispose();

}

}

// 删除

public void jButton6\_actionPerformed(ActionEvent e) {

getM();

if (intRow == -1) {

jOptionPane1.showMessageDialog(this, "请选择要删除的课程！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

try {

sta.getUpdate("delete from tb\_cource where courceId='" + Integer.valueOf(find) + "'");

jOptionPane1.showMessageDialog(this, "删除成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

UpdateRecord();

}

}

class CourceF\_jButton1\_actionAdapter implements ActionListener {

private CourceManager adaptee;

CourceF\_jButton1\_actionAdapter(CourceManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class CourceF\_jButton6\_actionAdapter implements ActionListener {

private CourceManager adaptee;

CourceF\_jButton6\_actionAdapter(CourceManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton6\_actionPerformed(e);

}

}

class CourceF\_jButton2\_actionAdapter implements ActionListener {

private CourceManager adaptee;

CourceF\_jButton2\_actionAdapter(CourceManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**db:**

**dbConn:**

package db;

import java.sql.\*;

/\*\*

\*

\* Title: 数据库连接

\* Description: 数据库连接模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class dbConn {

public dbConn() {

try {

jbInit();

} catch (Exception e) {

e.printStackTrace();

}

}

private Statement conn() {

try {

// 加载驱动

Class.forName("com.mysql.cj.jdbc.Driver");

// 数据库名称，管理员账号、密码

String url = "jdbc:mysql://localhost:3306/stuManagerDB";

String user = "root";

String pwd = "123456";

// 连接

Connection con = null;

con = DriverManager.getConnection(url, user, pwd);

Statement stat = con.createStatement();

return stat;

} catch (ClassNotFoundException ex) {

return null;

} catch (SQLException ex1) {

return null;

}

}

// 查询数据库

public ResultSet getRs(String sql) {

try {

Statement stat = conn();

ResultSet rs = stat.executeQuery(sql);

System.out.println(rs);

return rs;

} catch (SQLException ex) {

System.err.println("------------" + ex.getMessage());

return null;

}

}

// 更新数据库

public int getUpdate(String sql) {

try {

Statement stat = conn();

int i = stat.executeUpdate(sql);

return i;

} catch (Exception ex) {

System.out.println(">>>>>>>>" + ex.getMessage());

return -1;

}

}

private void jbInit() throws Exception {

conn();

}

}

**Depart：**

**DepartAddFrame:**

package Depart;

import java.sql.ResultSet;

import java.awt.\*;

import javax.swing.\*;

import db.dbConn;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\*

\* Title: 院系信息录入

\* Description: 院系信息录入模块，实现对学院、专业的录入。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class DepartAddFrame extends JFrame {

JPanel contentPane;

JLabel jLabel1 = new JLabel();

JLabel jLabel3 = new JLabel();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

JComboBox jComboBox1 = new JComboBox();

String xueyuan, zhuanye;

dbConn conn = new dbConn();

JLabel jLabel4 = new JLabel();

JComboBox jComboBox2 = new JComboBox();

public DepartAddFrame() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(465, 280));

setTitle("院系录入");

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel1.setText("院 系 信 息 录 入");

jLabel1.setBounds(new Rectangle(136, 20, 212, 25));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel3.setText("所属学院:");

jLabel3.setBounds(new Rectangle(80, 80, 90, 20));

jButton1.setBounds(new Rectangle(102, 180, 96, 29));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("提 交");

jButton1.addActionListener(new AddDepartFrame\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(265, 180, 96, 31));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setToolTipText("");

jButton2.setText("退 出");

jButton2.addActionListener(new AddDepartFrame\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(106, 258, 262, 90));

jOptionPane1.setLayout(null);

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setEditable(true);

jComboBox1.setBounds(new Rectangle(180, 80, 180, 25));

jComboBox1.addActionListener(new AddDepartFrame\_jComboBox1\_actionAdapter(this));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel4.setText("所属专业：");

jLabel4.setBounds(new Rectangle(80, 130, 90, 20));

jComboBox2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox2.setEditable(true);

jComboBox2.setBounds(new Rectangle(180, 130, 180, 25));

contentPane.add(jLabel1);

contentPane.add(jOptionPane1);

contentPane.add(jComboBox1);

contentPane.add(jLabel3);

contentPane.add(jLabel4);

contentPane.add(jComboBox2);

contentPane.add(jButton2);

contentPane.add(jButton1);

jComboBox1.addItem("请选择学院");

// 选择学院

try {

ResultSet rs = conn.getRs("select \* from tb\_depart ");

while (rs.next()) {

String xibu = rs.getString("departName");

jComboBox1.addItem(xibu);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

jComboBox2.setEnabled(false);

}

// 选择专业

public void zhuanye() {

jComboBox2.removeAllItems();

jComboBox2.addItem("请选择专业");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where departName='" + String.valueOf(jComboBox1.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("specName");

jComboBox2.addItem(zhy);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 存下选定学院到xueyuan

public void xueYuan() {

String sel = String.valueOf(jComboBox1.getSelectedItem());

try {

ResultSet rs = conn.getRs("select \* from tb\_depart where departName='" + sel + "'");

while (rs.next()) {

xueyuan = rs.getString("departName");

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 存下选定专业到zhuanye

public void zhuanYe() {

String sel = String.valueOf(jComboBox2.getSelectedItem());

try {

ResultSet rs = conn.getRs("select \* from tb\_spec where specName='" + sel + "'");

while (rs.next()) {

zhuanye = rs.getString("specName");

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 添加院系

public void jButton1\_actionPerformed(ActionEvent e) {

// 异常判断

if (jComboBox1.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择所属学院！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jComboBox2.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择所属专业！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

xueYuan();

zhuanYe();

try {

// 对学院、专业分别存在与否的四种情况做判断处理

boolean departExist = false, specExist = false;

ResultSet rs\_departExist = conn.getRs("select \* from tb\_spec where departName= '"

+ String.valueOf(jComboBox1.getSelectedItem()) + "' ");

while (rs\_departExist.next()) {

departExist = true;

}

ResultSet rs\_specExist = conn.getRs("select \* from tb\_spec where specName= '"

+ String.valueOf(jComboBox2.getSelectedItem()) + "' ");

while (rs\_specExist.next()) {

specExist = true;

}

if (departExist && specExist) {

jOptionPane1.showMessageDialog(this, "该学院与专业已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (!departExist && specExist) {

jOptionPane1.showMessageDialog(this, "该专业在其他学院已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (departExist && !specExist) {

conn.getUpdate("insert into tb\_spec (departName,specName) values ('"

+ String.valueOf(jComboBox1.getSelectedItem()) + "','"

+ String.valueOf(jComboBox2.getSelectedItem()) + "')");

jOptionPane1.showMessageDialog(this, "恭喜您院系录入成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (!departExist && !specExist) {

conn.getUpdate("insert into tb\_spec (departName,specName) values ('"

+ String.valueOf(jComboBox1.getSelectedItem()) + "','"

+ String.valueOf(jComboBox2.getSelectedItem()) + "')");

conn.getUpdate("insert into tb\_depart (departName) values ('"

+ String.valueOf(jComboBox1.getSelectedItem()) + "') ");

jOptionPane1.showMessageDialog(this, "恭喜您院系录入成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

} catch (Exception ce) {

System.out.println(ce);

}

}

}

// 选择触发

public void jComboBox1\_actionPerformed(ActionEvent e) {

zhuanye();

jComboBox2.setEnabled(true);

}

}

class AddDepartFrame\_jComboBox1\_actionAdapter implements ActionListener {

private DepartAddFrame adaptee;

AddDepartFrame\_jComboBox1\_actionAdapter(DepartAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox1\_actionPerformed(e);

}

}

class AddDepartFrame\_jButton1\_actionAdapter implements ActionListener {

private DepartAddFrame adaptee;

AddDepartFrame\_jButton1\_actionAdapter(DepartAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class AddDepartFrame\_jButton2\_actionAdapter implements ActionListener {

private DepartAddFrame adaptee;

AddDepartFrame\_jButton2\_actionAdapter(DepartAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**DepartChange:**

package Depart;

import java.sql.ResultSet;

import java.awt.\*;

import javax.swing.\*;

import db.dbConn;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\*

\* Title: 院系修改

\* Description: 院系信息修改模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class DepartChange extends JFrame {

JPanel contentPane;

JLabel jLabel1 = new JLabel();

JLabel jLabel3 = new JLabel();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

JComboBox jComboBox1 = new JComboBox();

dbConn conn = new dbConn();

JLabel jLabel4 = new JLabel();

JComboBox jComboBox2 = new JComboBox();

String find;

String zhuanye, xueyuan;

public DepartChange(String find) {

this.find = find;

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(465, 280));

setTitle("院系修改");

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel1.setText("院 系 信 息 修 改");

jLabel1.setBounds(new Rectangle(136, 20, 212, 25));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel3.setText("所属学院:");

jLabel3.setBounds(new Rectangle(80, 100, 90, 20));

jButton1.setBounds(new Rectangle(102, 223, 96, 29));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("修 改");

jButton1.addActionListener(new DepartU\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(265, 221, 96, 31));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setToolTipText("");

jButton2.setText("退 出");

jButton2.addActionListener(new DepartU\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(106, 258, 262, 90));

jOptionPane1.setLayout(null);

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setEditable(true);

jComboBox1.setBounds(new Rectangle(180, 100, 180, 25));

jComboBox1.addActionListener(new DepartU\_jComboBox1\_actionAdapter(this));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel4.setText("所属专业：");

jLabel4.setBounds(new Rectangle(80, 150, 90, 20));

jComboBox2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox2.setEditable(true);

jComboBox2.setBounds(new Rectangle(180, 150, 180, 25));

contentPane.add(jLabel1);

contentPane.add(jOptionPane1);

contentPane.add(jComboBox1);

contentPane.add(jLabel3);

contentPane.add(jLabel4);

contentPane.add(jComboBox2);

contentPane.add(jButton2);

contentPane.add(jButton1);

jComboBox1.addItem("请选择学院");

// 选择学院

try {

ResultSet rs = conn.getRs("select \* from tb\_depart ");

while (rs.next()) {

String xibu = rs.getString("departName");

jComboBox1.addItem(xibu);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

jComboBox2.setEnabled(false);

xianshi();

}

// 选择专业

public void zhuanye() {

jComboBox2.removeAllItems();

jComboBox2.addItem("请选择专业");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where departName='" + String.valueOf(jComboBox1.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("specName");

jComboBox2.addItem(zhy);

}

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 存下选定学院到xueyuan

public void xueYuan() {

String sel = String.valueOf(jComboBox1.getSelectedItem());

try {

ResultSet rs = conn.getRs("select \* from tb\_depart where departName='" + sel + "'");

while (rs.next()) {

xueyuan = rs.getString("departName");

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 存下选定专业到zhuanye

public void zhuanYe() {

String sel = String.valueOf(jComboBox2.getSelectedItem());

try {

ResultSet rs = conn.getRs("select \* from tb\_spec where specName='" + sel + "'");

while (rs.next()) {

zhuanye = rs.getString("specName");

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 修改

public void jButton1\_actionPerformed(ActionEvent e) {

// 异常判断

if (jComboBox1.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择所属学院！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jComboBox2.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择所属专业！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

xueYuan();

zhuanYe();

try {

// 对学院、专业分别存在与否的四种情况做判断处理

boolean departExist = false, specExist = false;

ResultSet rs\_departExist = conn.getRs("select \* from tb\_spec where departName= '"

+ String.valueOf(jComboBox1.getSelectedItem()) + "' ");

while (rs\_departExist.next()) {

departExist = true;

}

ResultSet rs\_specExist = conn.getRs("select \* from tb\_spec where specName= '"

+ String.valueOf(jComboBox2.getSelectedItem()) + "' ");

while (rs\_specExist.next()) {

specExist = true;

}

if (departExist && specExist) {

jOptionPane1.showMessageDialog(this, "该学院与专业已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (!departExist && specExist) {

jOptionPane1.showMessageDialog(this, "该专业在其他学院已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (departExist && !specExist) {

conn.getUpdate("delete from tb\_spec where specId='" + Integer.valueOf(find) + "'");

conn.getUpdate("insert into tb\_spec (departName,specName) values ('"

+ String.valueOf(jComboBox1.getSelectedItem()) + "','"

+ String.valueOf(jComboBox2.getSelectedItem()) + "')");

jOptionPane1.showMessageDialog(this, "恭喜您院系信息修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (!departExist && !specExist) {

conn.getUpdate("delete from tb\_spec where specId='" + Integer.valueOf(find) + "'");

conn.getUpdate("insert into tb\_spec (departName,specName) values ('"

+ String.valueOf(jComboBox1.getSelectedItem()) + "','"

+ String.valueOf(jComboBox2.getSelectedItem()) + "')");

conn.getUpdate("insert into tb\_depart (departName) values ('"

+ String.valueOf(jComboBox1.getSelectedItem()) + "') ");

jOptionPane1.showMessageDialog(this, "恭喜您院系信息修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

} catch (Exception ce) {

System.out.println(ce);

}

}

}

// 选择触发

public void jComboBox1\_actionPerformed(ActionEvent e) {

zhuanye();

jComboBox2.setEnabled(true);

}

// 选定行后在修改界面显示选定默认值

public void xianshi() {

if (find == null)

return;

else {

jComboBox2.setEnabled(true);

try {

ResultSet rs = conn.getRs("select \* from tb\_spec where specId='" + Integer.valueOf(find) + "'");

while (rs.next()) {

jComboBox1.setSelectedItem(String.valueOf(rs.getString("departName")));

jComboBox2.setSelectedItem(String.valueOf(rs.getString("specName")));

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

}

}

class DepartU\_jComboBox1\_actionAdapter implements ActionListener {

private DepartChange adaptee;

DepartU\_jComboBox1\_actionAdapter(DepartChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox1\_actionPerformed(e);

}

}

class DepartU\_jButton1\_actionAdapter implements ActionListener {

private DepartChange adaptee;

DepartU\_jButton1\_actionAdapter(DepartChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class DepartU\_jButton2\_actionAdapter implements ActionListener {

private DepartChange adaptee;

DepartU\_jButton2\_actionAdapter(DepartChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**DepartManager:**

package Depart;

import java.awt.\*;

import com.borland.jbcl.layout.\*;

import db.dbConn;

import stuManager.MainFrame;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

import javax.swing.table.DefaultTableModel;

import java.util.\*;

/\*\*

\*

\* Title: 院系管理

\* Description: 院系管理模块，可以对院系信息进行查询，是修改和删除的入口。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class DepartManager extends JFrame {

XYLayout xYLayout1 = new XYLayout();

JLabel jLabel1 = new JLabel();

ButtonGroup buttonGroup1 = new ButtonGroup();

JScrollPane jScrollPane1 = new JScrollPane();

JTable jTable1 = new JTable();

JButton jButton4 = new JButton();

JButton jButton5 = new JButton();

JButton jButton6 = new JButton();

dbConn conn = new dbConn();

Object[][] arrData = {};

String[] arrField = { "专业编号", "学院名称", "专业名称" };

JOptionPane jOptionPane1 = new JOptionPane();

DefaultTableModel model = new DefaultTableModel();

String sql, find;

int intRow;

public DepartManager() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

getContentPane().setLayout(xYLayout1);

xYLayout1.setWidth(530);

xYLayout1.setHeight(540);

jLabel1.setFont(new java.awt.Font("黑体", Font.PLAIN, 20));

jLabel1.setText("院 系 信 息 管 理");

jScrollPane1.setBorder(BorderFactory.createEtchedBorder());

jButton4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton4.setText("修 改");

jButton4.addActionListener(new DepartManager\_jButton4\_actionAdapter(this));

jButton5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton5.setText("删 除");

jButton5.addActionListener(new DepartManager\_jButton5\_actionAdapter(this));

jButton6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton6.setText("返 回");

jButton6.addActionListener(new DepartManager\_jButton6\_actionAdapter(this));

this.getContentPane().add(jLabel1, new XYConstraints(177, 14, 177, 39));

jScrollPane1.getViewport().add(jTable1);

this.getContentPane().add(jScrollPane1, new XYConstraints(10, 60, 510, 400));

this.getContentPane().add(jButton4, new XYConstraints(100, 480, 90, 35));

this.getContentPane().add(jButton5, new XYConstraints(220, 480, 90, 35));

this.getContentPane().add(jButton6, new XYConstraints(340, 480, 90, 35));

// 用户等级判断

// 1为管理员

if (MainFrame.level.equals("1")) {

// 不隐藏任何功能

}

// 2为普通用户

else if (MainFrame.level.equals("2")) {

// 隐藏部分功能

jButton4.setVisible(false);

jButton5.setVisible(false);

}

// 选择专业表

sql = "select \* from tb\_spec";

// 刷新

UpdateRecord();

}

// 修改院系

public void jButton4\_actionPerformed(ActionEvent e) {

getM();

if (intRow == -1) {

jOptionPane1.showMessageDialog(this, "请选择要修改的院系！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

DepartChange siadd = new DepartChange(find);

siadd.setLocation(400, 200);

siadd.setSize(465, 310);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

this.dispose();

}

}

// 删除院系

public void jButton5\_actionPerformed(ActionEvent e) {

getM();

if (intRow == -1) {

jOptionPane1.showMessageDialog(this, "请选择要删除的院系！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

try {

conn.getUpdate("delete from tb\_spec where specId='" + Integer.valueOf(find) + "'");

jOptionPane1.showMessageDialog(this, "删除成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

UpdateRecord();

}

// 退出

public void jButton6\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 更新表格

public void UpdateRecord() {

Object[][] arrTmp = {}; // 设定表格的字段

Vector vec = new Vector(1, 1);

model = new DefaultTableModel(arrTmp, arrField);

jTable1 = new JTable(model);

jScrollPane1.getViewport().add(jTable1, null);

try {

ResultSet rs3 = conn.getRs(sql);

while (rs3.next()) {

vec = new Vector();

vec.add(String.valueOf(rs3.getInt("specId")));

vec.add(rs3.getString("departName"));

vec.add(rs3.getString("specName"));

model.addRow(vec);

}

rs3.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

jScrollPane1.getHorizontalScrollBar();

jTable1.setGridColor(Color.blue);

jTable1.setDragEnabled(true);

jTable1.setSelectionForeground(Color.red);

jTable1.setSelectionBackground(Color.green);

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.setRowSelectionAllowed(true);

jTable1.setShowVerticalLines(true);

}

// 获取所选的行

public void getM() {

intRow = jTable1.getSelectedRow();

if (intRow == -1) {

return;

}

try {

find = model.getValueAt(intRow, 0).toString().trim();

} catch (Exception e) {

e.printStackTrace();

}

}

}

class DepartManager\_jButton6\_actionAdapter implements ActionListener {

private DepartManager adaptee;

DepartManager\_jButton6\_actionAdapter(DepartManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton6\_actionPerformed(e);

}

}

class DepartManager\_jButton5\_actionAdapter implements ActionListener {

private DepartManager adaptee;

DepartManager\_jButton5\_actionAdapter(DepartManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton5\_actionPerformed(e);

}

}

class DepartManager\_jButton4\_actionAdapter implements ActionListener {

private DepartManager adaptee;

DepartManager\_jButton4\_actionAdapter(DepartManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton4\_actionPerformed(e);

}

}

**Score:**

**ScoreAddFrame:**

package Score;

import java.sql.ResultSet;

import java.awt.\*;

import javax.swing.\*;

import db.dbConn;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\*

\* Title: 成绩录入

\* Description: 成绩录入模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class ScoreAddFrame extends JFrame {

JPanel contentPane;

JLabel jLabel1 = new JLabel();

int FS, spid, FC;

JLabel jLabel3 = new JLabel();

JLabel jLabel4 = new JLabel();

String IS;

JTextField jTextField2 = new JTextField();

JTextField jTextField3 = new JTextField();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

JLabel jLabel6 = new JLabel();

JComboBox jComboBox1 = new JComboBox();

dbConn sta = new dbConn();

public ScoreAddFrame() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(482, 300));

setTitle("成绩录入");

jLabel1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jLabel1.setText("成 绩 录 入");

jLabel1.setBounds(new Rectangle(178, 17, 126, 25));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel3.setText("学生学号：");

jLabel3.setBounds(new Rectangle(75, 70, 81, 21));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel4.setText("成 绩：");

jLabel4.setBounds(new Rectangle(75, 170, 77, 21));

jTextField2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField2.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField2.setText("");

jTextField2.setBounds(new Rectangle(190, 70, 184, 26));

jTextField3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jTextField3.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField3.setText("");

jTextField3.setBounds(new Rectangle(190, 170, 94, 26));

jButton1.setBounds(new Rectangle(103, 240, 90, 29));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("提 交");

jButton1.addActionListener(new ScoreAddFrame\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(277, 240, 90, 29));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setText("退 出");

jButton2.addActionListener(new ScoreAddFrame\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(75, 261, 262, 90));

jOptionPane1.setLayout(null);

jLabel6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel6.setText("课 程：");

jLabel6.setBounds(new Rectangle(75, 120, 82, 21));

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setBounds(new Rectangle(190, 120, 160, 25));

contentPane.add(jLabel1);

contentPane.add(jButton1);

contentPane.add(jButton2);

contentPane.add(jOptionPane1);

contentPane.add(jTextField3);

contentPane.add(jLabel4);

contentPane.add(jLabel3);

contentPane.add(jTextField2);

contentPane.add(jComboBox1);

contentPane.add(jLabel6);

jComboBox1.addItem("请选择");

try {

// 选择课程

ResultSet rs = sta.getRs("select \* from tb\_cource");

while (rs.next()) {

String xibu = rs.getString("courceName");

jComboBox1.addItem(xibu);

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

// 录入

public void InC() {

// 异常判断

if (jTextField2.getText().length() == 0) {

jOptionPane1.showMessageDialog(this, "学生学号不能为空！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jComboBox1.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择课程！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jTextField3.getText().length() == 0) {

jOptionPane1.showMessageDialog(this, "成绩不能为空！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

try {

boolean name = false;

ResultSet rs = sta.getRs("select stuNumber from tb\_score where courceName='"

+ String.valueOf(jComboBox1.getSelectedItem()) + "'");

while (rs.next()) {

if (jTextField2.getText().trim().equals(rs.getString("stuNumber").trim())) {

name = true;

}

}

if (name) {

jOptionPane1.showMessageDialog(this, "课程名称已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

sta.getUpdate("insert into tb\_score (stuNumber,score,courceName) values ('"

+ jTextField2.getText().trim() + "','" + Float.valueOf(jTextField3.getText().trim()) + "','"

+ String.valueOf(jComboBox1.getSelectedItem()) + "')");

jOptionPane1.showMessageDialog(this, "课程信息提交成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

rs.close();

} catch (Exception ce) {

System.out.println("--------" + ce);

}

}

}

// 录入

public void jButton1\_actionPerformed(ActionEvent e) {

InC();

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

}

class ScoreAddFrame\_jButton1\_actionAdapter implements ActionListener {

private ScoreAddFrame adaptee;

ScoreAddFrame\_jButton1\_actionAdapter(ScoreAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class ScoreAddFrame\_jButton2\_actionAdapter implements ActionListener {

private ScoreAddFrame adaptee;

ScoreAddFrame\_jButton2\_actionAdapter(ScoreAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**ScoreChange:**

package Score;

import java.awt.Font;

import com.borland.jbcl.layout.\*;

import db.dbConn;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

import javax.swing.\*;

/\*\*

\*

\* Title: 成绩修改

\* Description: 成绩修改模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class ScoreChange extends JFrame {

XYLayout xYLayout1 = new XYLayout();

JLabel jLabel1 = new JLabel();

JLabel jLabel2 = new JLabel();

JLabel jLabel3 = new JLabel();

JLabel jLabel5 = new JLabel();

JTextField jTextField1 = new JTextField();

JComboBox jComboBox1 = new JComboBox();

JTextField jTextField2 = new JTextField();

JButton jButton1 = new JButton();

JButton jButton3 = new JButton();

dbConn sta = new dbConn();

JOptionPane jOptionPane1 = new JOptionPane();

int csid;

int kefind;

public ScoreChange(int find) {

kefind = find;

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

getContentPane().setLayout(xYLayout1);

xYLayout1.setWidth(500);

xYLayout1.setHeight(350);

jLabel1.setFont(new java.awt.Font("新宋体", Font.BOLD, 20));

jLabel1.setText("成 绩 修 改 ");

jButton3.addActionListener(new ScoreManager\_jButton3\_actionAdapter(this));

jButton1.addActionListener(new ScoreManager\_jButton1\_actionAdapter(this));

jTextField2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton3.setText("退 出");

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setText("修 改");

jLabel5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel5.setText("学 分：");

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel3.setText("课程名称：");

this.getContentPane().add(jComboBox1, new XYConstraints(200, 130, 140, 30));

this.getContentPane().add(jTextField2, new XYConstraints(200, 180, 140, 30));

this.getContentPane().add(jLabel2, new XYConstraints(85, 80, 80, 30));

this.getContentPane().add(jLabel5, new XYConstraints(85, 180, 80, 30));

this.getContentPane().add(jLabel3, new XYConstraints(85, 130, 80, 30));

this.getContentPane().add(jTextField1, new XYConstraints(200, 80, 140, 30));

this.getContentPane().add(jLabel1, new XYConstraints(158, 21, -1, 49));

this.getContentPane().add(jButton1, new XYConstraints(130, 230, 90, 30));

this.getContentPane().add(jButton3, new XYConstraints(295, 230, 90, 30));

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel2.setText("学生学号：");

this.setTitle("成绩修改");

jComboBox1.addItem("请选择");

// 选择课程

try {

ResultSet rs = sta.getRs("select \* from tb\_cource");

while (rs.next()) {

String xibu = rs.getString("courceName");

jComboBox1.addItem(xibu);

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

setF();

}

// 选定行后在修改界面显示选定默认值

public void setF() {

if (kefind < 0)

return;

else {

try {

ResultSet rs = sta.getRs("select \* from tb\_score where scoreId='" + Integer.valueOf(kefind) + "'");

while (rs.next()) {

jComboBox1.setSelectedItem(String.valueOf(rs.getString("courceName")));

jTextField1.setText(rs.getString("stuNumber").trim());

jTextField2.setText(rs.getString("score").trim());

}

rs.close();

} catch (Exception ce) {

System.out.println("++++++++" + ce);

}

}

}

// 退出

public void jButton3\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 修改

public void jButton1\_actionPerformed(ActionEvent e) {

String keCName, zyName, tName, xueFen;

keCName = jTextField1.getText().trim();

zyName = String.valueOf(jComboBox1.getSelectedItem());

xueFen = jTextField2.getText().trim();

try {

sta.getUpdate("update tb\_score set courceName='" + zyName + "', stuNumber='" + keCName + "',score='"

+ Float.valueOf(xueFen) + "' where scoreId='" + kefind + "'");

jOptionPane1.showMessageDialog(this, "课程修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} catch (Exception a) {

System.out.println(a.getMessage());

}

}

}

class ScoreManager\_jButton1\_actionAdapter implements ActionListener {

private ScoreChange adaptee;

ScoreManager\_jButton1\_actionAdapter(ScoreChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class ScoreManager\_jButton3\_actionAdapter implements ActionListener {

private ScoreChange adaptee;

ScoreManager\_jButton3\_actionAdapter(ScoreChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton3\_actionPerformed(e);

}

}

**ScoreManager:**

package Score;

import com.borland.jbcl.layout.\*;

import db.dbConn;

import stuManager.MainFrame;

import javax.swing.\*;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.table.DefaultTableModel;

import java.util.\*;

import java.awt.\*;

/\*\*

\*

\* Title: 成绩管理

\* Description: 成绩管理模块，可以查询成绩，是成绩修改和删除的入口。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class ScoreManager extends JFrame {

XYLayout xYLayout1 = new XYLayout();

JLabel jLabel1 = new JLabel();

JPanel jPanel1 = new JPanel();

JScrollPane jScrollPane1 = new JScrollPane();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

XYLayout xYLayout2 = new XYLayout();

JTable jTable1 = new JTable();

ButtonGroup buttonGroup1 = new ButtonGroup();

JButton jButton\_print = new JButton();

JButton jButton\_sort = new JButton();

JButton jButton\_average = new JButton();

JButton jButton\_failure = new JButton();

dbConn sta = new dbConn();

String sql;

Object[][] arrData = {};

String[] arrField = { "成绩编号", "学生学号", "姓名", "课程", "分数" };

DefaultTableModel model = new DefaultTableModel();

int intRow;

static int find;

int i\_sortDesc = 0, i\_print = 0;

JOptionPane jOptionPane1 = new JOptionPane();

JButton jButton6 = new JButton();

public ScoreManager() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

getContentPane().setLayout(xYLayout1);

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel1.setText("成 绩 管 理");

jScrollPane1.setBorder(BorderFactory.createEtchedBorder());

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton1.setText("修 改");

jButton1.addActionListener(new ScoreF\_jButton1\_actionAdapter(this));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton2.setText("返 回");

jButton2.addActionListener(new ScoreF\_jButton2\_actionAdapter(this));

jPanel1.setLayout(xYLayout2);

jPanel1.setBorder(BorderFactory.createEtchedBorder());

xYLayout1.setWidth(550);

xYLayout1.setHeight(560);

jButton\_print.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton\_print.setText("打 印");

jButton\_print.addActionListener(new ScoreManager\_jButton\_print\_actionAdapter(this));

jButton\_sort.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton\_sort.setText("排 序");

jButton\_sort.addActionListener(new ScoreManager\_jButton\_sort\_actionAdapter(this));

jButton\_average.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton\_average.setText("平 均");

jButton\_average.addActionListener(new ScoreManager\_jButton\_average\_actionAdapter(this));

jButton\_failure.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton\_failure.setText("不及格");

jButton\_failure.addActionListener(new ScoreManager\_jButton\_failure\_actionAdapter(this));

this.setTitle("成绩管理");

jButton6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jButton6.setText("删 除");

jButton6.addActionListener(new ScoreF\_jButton6\_actionAdapter(this));

jScrollPane1.getViewport().add(jTable1);

this.getContentPane().add(jLabel1, new XYConstraints(208, 13, 135, 43));

this.getContentPane().add(jButton1, new XYConstraints(74, 470, 100, -1));

this.getContentPane().add(jButton2, new XYConstraints(374, 470, 100, -1));

this.getContentPane().add(jPanel1, new XYConstraints(17, 68, 515, 63));

this.getContentPane().add(jScrollPane1, new XYConstraints(18, 150, 515, 300));

this.getContentPane().add(jButton6, new XYConstraints(224, 470, 100, -1));

jPanel1.add(jButton\_print, new XYConstraints(10, 15, 80, 25));

jPanel1.add(jButton\_sort, new XYConstraints(147, 15, 80, 25));

jPanel1.add(jButton\_average, new XYConstraints(287, 15, 80, 25));

jPanel1.add(jButton\_failure, new XYConstraints(419, 15, 80, 25));

// 用户等级判断

// 1为管理员

if (MainFrame.level.equals("1")) {

// 不隐藏任何功能

}

// 2为普通用户

else if (MainFrame.level.equals("2")) {

// 隐藏部分功能

jButton1.setVisible(false);

jButton6.setVisible(false);

}

// 设置默认查找所有成绩

sql = "select \* from tb\_score ";

// 更新

UpdateRecord();

}

// 查询并更新表格

public void UpdateRecord() {

Object[][] arrTmp = {}; // 设定表格的字段

Vector vec = new Vector(1, 1);

model = new DefaultTableModel(arrTmp, arrField);

jTable1 = new JTable(model);

jScrollPane1.getViewport().add(jTable1, null);

try {

ResultSet rs = sta.getRs(sql);

int i = 1;

while (rs.next()) {

vec = new Vector();

vec.add(String.valueOf(rs.getString("scoreId")));

vec.add(String.valueOf(rs.getString("stuNumber")));

ResultSet rsTemp = sta.getRs("select stuName from tb\_student where stuNumber='"

+ String.valueOf(rs.getString("stuNumber")) + "'");

while (rsTemp.next()) {

vec.add(String.valueOf(rsTemp.getString("stuName")));

}

vec.add(String.valueOf(rs.getString("courceName")));

vec.add(String.valueOf(rs.getString("score")));

model.addRow(vec);

}

rs.close();

} catch (Exception ce) {

System.out.println(ce);

}

jScrollPane1.getHorizontalScrollBar();

jTable1.setGridColor(Color.blue);

jTable1.setDragEnabled(true);

jTable1.setSelectionForeground(Color.red);

jTable1.setSelectionBackground(Color.green);

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.setRowSelectionAllowed(true);

jTable1.setShowVerticalLines(true);

}

// 选定所选的行

public void getM() {

intRow = jTable1.getSelectedRow();

if (intRow == -1) {

jOptionPane1.showMessageDialog(this, "请选择要修改的课程！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

return;

}

try {

find = Integer.parseInt(model.getValueAt(intRow, 0).toString());

System.out.println(find);

} catch (Exception e) {

e.printStackTrace();

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 单次点击打印对学号进行升序排序

// 双次点击打印对学号进行升序排序

public void jButton\_print\_actionPerformed(ActionEvent e) {

i\_print++;

if (i\_print % 2 == 1) {

sql = "select \* from tb\_score order by scoreId, stuNumber";

} else if (i\_print % 2 == 0) {

sql = "select \* from tb\_score order by stuNumber, scoreId";

}

UpdateRecord();

}

// 单次点击排序对成绩进行降序排序

// 双次点击排序对成绩进行升序排序

public void jButton\_sort\_actionPerformed(ActionEvent e) {

i\_sortDesc++;

if (i\_sortDesc % 2 == 1) {

sql = "select \* from tb\_score order by score desc, stuNumber";

} else if (i\_sortDesc % 2 == 0) {

sql = "select \* from tb\_score order by score, stuNumber";

}

UpdateRecord();

}

// 平均分

public void jButton\_average\_actionPerformed(ActionEvent e) {

float f\_average = 0;

int averageBig = 0, averageSmall = 0;

try {

// 获得平均分

ResultSet rs\_average = sta.getRs("select avg(score) as scoreAverage from tb\_score");

while (rs\_average.next()) {

f\_average = Float.valueOf(rs\_average.getString("scoreAverage"));

}

// 获得大于等于平均分的人数

ResultSet rs\_averageBig = sta.getRs("select \* from tb\_score where score>='" + f\_average + "'");

while (rs\_averageBig.next()) {

averageBig++;

}

// 获得小于平均分的人数

ResultSet rs\_averageSmall = sta.getRs("select \* from tb\_score where score<'" + f\_average + "'");

while (rs\_averageSmall.next()) {

averageSmall++;

}

} catch (SQLException e1) {

System.out.println(e1);

}

// 显示平均分

jOptionPane1.showMessageDialog(this,

"平均分 = " + f\_average + "\n大于等于平均分有 " + averageBig + "人\n小于平均分有 " + averageSmall + "人", "提示",

JOptionPane.INFORMATION\_MESSAGE, null);

}

// 不及格

public void jButton\_failure\_actionPerformed(ActionEvent e) {

int totalNumber = 0, failureNumber = 0;

try {

// 获取总人数

ResultSet rs\_total = sta.getRs("select \* from tb\_score");

while (rs\_total.next()) {

totalNumber++;

}

// 获取低于60分（不及格）人数

ResultSet rs\_failure = sta.getRs("select \* from tb\_score where score<'" + 60 + "'");

while (rs\_failure.next()) {

failureNumber++;

}

} catch (SQLException e1) {

System.out.println(e1);

}

// 显示不及格率

jOptionPane1.showMessageDialog(this, "不及格率 = " + (float) failureNumber / totalNumber \* 100 + "%\n不及格人数有 "

+ failureNumber + "人\n总人数有 " + totalNumber + "人", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

// 修改课程

public void jButton1\_actionPerformed(ActionEvent e) {

getM();

if (find > 0) {

ScoreChange siadd = new ScoreChange(find);

siadd.setLocation(400, 200);

siadd.setSize(465, 310);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

this.dispose();

}

}

// 删除课程

public void jButton6\_actionPerformed(ActionEvent e) {

getM();

if (intRow == -1) {

jOptionPane1.showMessageDialog(this, "请选择要删除的课程！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

try {

sta.getUpdate("delete from tb\_score where scoreId='" + Integer.valueOf(find) + "'");

jOptionPane1.showMessageDialog(this, "删除成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

UpdateRecord();

}

}

class ScoreF\_jButton1\_actionAdapter implements ActionListener {

private ScoreManager adaptee;

ScoreF\_jButton1\_actionAdapter(ScoreManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class ScoreF\_jButton6\_actionAdapter implements ActionListener {

private ScoreManager adaptee;

ScoreF\_jButton6\_actionAdapter(ScoreManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton6\_actionPerformed(e);

}

}

class ScoreF\_jButton2\_actionAdapter implements ActionListener {

private ScoreManager adaptee;

ScoreF\_jButton2\_actionAdapter(ScoreManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

class ScoreManager\_jButton\_print\_actionAdapter implements ActionListener {

private ScoreManager adaptee;

ScoreManager\_jButton\_print\_actionAdapter(ScoreManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton\_print\_actionPerformed(e);

}

}

class ScoreManager\_jButton\_sort\_actionAdapter implements ActionListener {

private ScoreManager adaptee;

ScoreManager\_jButton\_sort\_actionAdapter(ScoreManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton\_sort\_actionPerformed(e);

}

}

class ScoreManager\_jButton\_average\_actionAdapter implements ActionListener {

private ScoreManager adaptee;

ScoreManager\_jButton\_average\_actionAdapter(ScoreManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton\_average\_actionPerformed(e);

}

}

class ScoreManager\_jButton\_failure\_actionAdapter implements ActionListener {

private ScoreManager adaptee;

ScoreManager\_jButton\_failure\_actionAdapter(ScoreManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton\_failure\_actionPerformed(e);

}

}

**Student:**

**StudentChange:**

package Student;

import java.awt.\*;

import javax.swing.\*;

import db.dbConn;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

/\*\*

\*

\* Title: 学生信息修改

\* Description: 学生信息修改模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class StudentChange extends JFrame {

JPanel contentPane;

JLabel jLabel2 = new JLabel();

JLabel jLabel3 = new JLabel();

JLabel jLabel4 = new JLabel();

JLabel jLabel7 = new JLabel();

JLabel jLabel8 = new JLabel();

JTextField jTextField1 = new JTextField();

JTextField jTextField2 = new JTextField();

JComboBox jComboBox4 = new JComboBox();

JComboBox jComboBox5 = new JComboBox();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

dbConn con = new dbConn();

Object clas[] = {};

JLabel jLabel10 = new JLabel();

JComboBox jComboBox1 = new JComboBox();

dbConn conn = new dbConn();

JOptionPane jOptionPane\_LoginFeedback = new JOptionPane();

String number, name, sdepart, sspec, sclass;

int find;

public StudentChange(int find) {

this.find = find;

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(592, 500));

setTitle("学生信息修改");

jLabel2.setFont(new java.awt.Font("Dialog", Font.BOLD, 23));

jLabel2.setText("学 生 信 息 修 改");

jLabel2.setBounds(new Rectangle(197, 17, 231, 25));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel3.setText("学 号:");

jLabel3.setBounds(new Rectangle(50, 74, 90, 22));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel4.setText("姓 名:");

jLabel4.setBounds(new Rectangle(50, 124, 90, 22));

jLabel7.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel7.setText("所属学院:");

jLabel7.setBounds(new Rectangle(50, 174, 90, 22));

jLabel8.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel8.setText("所属专业:");

jLabel8.setBounds(new Rectangle(297, 174, 90, 22));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField1.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField1.setBounds(new Rectangle(150, 74, 195, 23));

jTextField2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField2.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField2.setText("");

jTextField2.setBounds(new Rectangle(150, 124, 195, 23));

jComboBox4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox4.setBorder(BorderFactory.createLoweredBevelBorder());

jComboBox4.setBounds(new Rectangle(150, 174, 125, 25));

jComboBox4.addActionListener(new StudentUp\_jComboBox4\_actionAdapter(this));

jComboBox5.setEnabled(false);

jComboBox5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox5.setBorder(BorderFactory.createLoweredBevelBorder());

jComboBox5.setBounds(new Rectangle(390, 174, 125, 25));

jComboBox5.addActionListener(new StudentUp\_jComboBox5\_actionAdapter(this));

jButton2.setBounds(new Rectangle(375, 274, 90, 27));

jButton1.setBounds(new Rectangle(147, 274, 90, 27));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("提 交");

jButton1.addActionListener(new StudentUp\_jButton1\_actionAdapter(this));

jButton2.setText("退 出");

jButton2.addActionListener(new StudentUp\_jButton2\_actionAdapter(this));

jLabel10.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel10.setText("所在班级：");

jLabel10.setBounds(new Rectangle(47, 224, 93, 29));

jComboBox1.setEnabled(false);

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setBounds(new Rectangle(150, 224, 125, 25));

contentPane.add(jLabel2);

contentPane.add(jLabel8);

contentPane.add(jComboBox5);

contentPane.add(jLabel7);

contentPane.add(jComboBox4);

contentPane.add(jLabel3);

contentPane.add(jTextField1);

contentPane.add(jLabel4);

contentPane.add(jTextField2);

contentPane.add(jButton1);

contentPane.add(jButton2);

contentPane.add(jLabel10);

contentPane.add(jComboBox1);

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

this.jComboBox4.addItem("请选择学院");

// 选择学院

try {

ResultSet rs = con.getRs("select \* from tb\_depart ");

while (rs.next()) {

String xibu = rs.getString("departName");

jComboBox4.addItem(xibu);

}

} catch (Exception ce) {

System.out.println(ce);

}

showstu();

jComboBox5.setEnabled(false);

jComboBox1.setEnabled(false);

}

// 选择学院后触发

public void jComboBox4\_actionPerformed(ActionEvent e) {

jcb();

jComboBox5.setEnabled(true);

}

// 选择专业后触发

public void jComboBox5\_actionPerformed(ActionEvent e) {

classAdd();

jComboBox1.setEnabled(true);

}

// 修改

public void jButton1\_actionPerformed(ActionEvent e) {

// 异常判断

if (jTextField1.getText().trim().equals("")) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "请输入学生学号！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else if (jTextField2.getText().trim().equals("")) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "请输入学生姓名！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else if (jComboBox4.getSelectedIndex() == 0) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "请选择学生所在学院！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else if (jComboBox5.getSelectedIndex() == 0) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "请选择学生所属专业！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else if (jComboBox1.getSelectedIndex() == 0) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "请选择学生所在班级！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else {

try {

findDepart();

findSpec();

findClass();

if (find == Integer.parseInt(jTextField1.getText().trim())) {

// 不修改学号的情况

number = jTextField1.getText().trim();

name = jTextField2.getText().trim();

instu();

} else {

// 修改学号的情况

boolean Num = false;

ResultSet rs = conn.getRs("select \* from tb\_student");

while (rs.next()) {

if (jTextField1.getText().trim().equals(rs.getString("stuNumber").trim())) {

Num = true;

}

}

rs.close();

if (Num) {

// 判断学号是否存在

jOptionPane\_LoginFeedback.showMessageDialog(this, "学生学号已存在，请重新输入！", "提示",

JOptionPane.INFORMATION\_MESSAGE, null);

} else {

number = jTextField1.getText().trim();

name = jTextField2.getText().trim();

instu();

}

}

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 查找学院

public void findDepart() {

try {

ResultSet rs = conn.getRs(

"select \* from tb\_depart where departName='" + String.valueOf(jComboBox4.getSelectedItem()) + "' ");

while (rs.next()) {

sdepart = rs.getString("departName");

}

rs.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

// 查找专业

public void findSpec() {

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where specName='" + String.valueOf(jComboBox5.getSelectedItem()) + "' ");

while (rs.next()) {

sspec = rs.getString("specName");

}

rs.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

// 查找班级

public void findClass() {

try {

ResultSet rs = conn.getRs(

"select \* from tb\_class where className='" + String.valueOf(jComboBox1.getSelectedItem()) + "' ");

while (rs.next()) {

sclass = rs.getString("className");

}

rs.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

// 选择专业

public void jcb() {

jComboBox5.removeAllItems();

jComboBox5.addItem("请选择专业");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where departName='" + String.valueOf(jComboBox4.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("specName");

jComboBox5.addItem(zhy);

}

rs.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

// 选择班级

public void classAdd() {

jComboBox1.removeAllItems();

jComboBox1.addItem("请选择班级");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_class where specName='" + String.valueOf(jComboBox5.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("className");

jComboBox1.addItem(zhy);

}

rs.close();

} catch (Exception e) {

System.out.println(e);

}

}

// 修改

public void instu() {

try {

if (0 < conn.getUpdate("update tb\_student set stuNumber = '" + number + "', stuName = '" + name

+ "', stuDepart = '" + sdepart + "', stuSpec = '" + sspec + "', stuClass = '" + sclass

+ "' where stuNumber = '" + find + "'")) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "学生信息修改成功！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else {

System.err.printf("修改 tb\_student 表中 stuNumber = %d 的记录失败\n", find);

jOptionPane\_LoginFeedback.showMessageDialog(this, "学生信息修改失败！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

}

// 显示

public void showstu() {

if (find < 0) {

return;

} else {

try {

ResultSet rs = con.getRs("select \* from tb\_student where stuNumber='" + find + "' ");

while (rs.next()) {

number = rs.getString(2);

name = rs.getString(3);

sdepart = rs.getString(4);

sspec = rs.getString(5);

sclass = rs.getString(5);

}

} catch (Exception ce) {

System.out.println(ce);

}

jTextField1.setText(number);

jTextField2.setText(name);

jComboBox4.setSelectedItem(sdepart);

jComboBox5.addItem(sspec);

jComboBox1.addItem(sclass);

}

}

}

class StudentUp\_jButton2\_actionAdapter implements ActionListener {

private StudentChange adaptee;

StudentUp\_jButton2\_actionAdapter(StudentChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

class StudentUp\_jButton1\_actionAdapter implements ActionListener {

private StudentChange adaptee;

StudentUp\_jButton1\_actionAdapter(StudentChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class StudentUp\_jComboBox5\_actionAdapter implements ActionListener {

private StudentChange adaptee;

StudentUp\_jComboBox5\_actionAdapter(StudentChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox5\_actionPerformed(e);

}

}

class StudentUp\_jComboBox4\_actionAdapter implements ActionListener {

private StudentChange adaptee;

StudentUp\_jComboBox4\_actionAdapter(StudentChange adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox4\_actionPerformed(e);

}

}

**StudentInfoAddFrame:**

package Student;

import java.awt.\*;

import javax.swing.\*;

import db.dbConn;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

/\*\*

\*

\* Title: 学生信息录入

\* Description: 学生信息录入模块

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class StudentInfoAddFrame extends JFrame {

JPanel contentPane;

JLabel jLabel1 = new JLabel();

JLabel jLabel2 = new JLabel();

JLabel jLabel4 = new JLabel();

JLabel jLabel6 = new JLabel();

JLabel jLabel7 = new JLabel();

JLabel jLabel8 = new JLabel();

JTextField jTextField1 = new JTextField();

JTextField jTextField2 = new JTextField();

JComboBox jComboBox4 = new JComboBox();

JComboBox jComboBox5 = new JComboBox();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

dbConn con = new dbConn();

Object clas[] = {};

JLabel jLabel10 = new JLabel();

JComboBox jComboBox1 = new JComboBox();

dbConn conn = new dbConn();

JOptionPane jOptionPane1 = new JOptionPane();

JOptionPane jOptionPane\_LoginFeedback = new JOptionPane();

String number, name, sdepart, sspec, sclass;

public StudentInfoAddFrame() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(592, 500));

setTitle("学生信息录入");

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 23));

jLabel1.setForeground(SystemColor.inactiveCaptionText);

jLabel1.setText("学 生 信 息 录 入");

jLabel1.setBounds(new Rectangle(196, 16, 232, 25));

jLabel2.setFont(new java.awt.Font("Dialog", Font.BOLD, 23));

jLabel2.setText("学 生 信 息 录 入");

jLabel2.setBounds(new Rectangle(197, 17, 231, 25));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel4.setText("学 号:");

jLabel4.setBounds(new Rectangle(50, 74, 90, 22));

jLabel6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel6.setText("姓 名:");

jLabel6.setBounds(new Rectangle(50, 124, 90, 22));

jLabel7.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel7.setText("所属学院:");

jLabel7.setBounds(new Rectangle(50, 174, 90, 22));

jLabel8.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel8.setText("所属专业:");

jLabel8.setBounds(new Rectangle(297, 174, 90, 22));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField1.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField1.setBounds(new Rectangle(150, 74, 195, 23));

jTextField2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jTextField2.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField2.setText("");

jTextField2.setBounds(new Rectangle(150, 124, 195, 23));

jComboBox4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox4.setBorder(BorderFactory.createLoweredBevelBorder());

jComboBox4.setBounds(new Rectangle(150, 174, 125, 25));

jComboBox4.addActionListener(new StudentInfoAddFrame\_jComboBox4\_actionAdapter(this));

jComboBox5.setEnabled(false);

jComboBox5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox5.setBorder(BorderFactory.createLoweredBevelBorder());

jComboBox5.setBounds(new Rectangle(390, 174, 125, 25));

jComboBox5.addActionListener(new StudentInfoAddFrame\_jComboBox5\_actionAdapter(this));

jButton1.setText("提 交");

jButton1.setBounds(new Rectangle(147, 274, 90, 27));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.addActionListener(new StudentInfoAddFrame\_jButton1\_actionAdapter(this));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jButton2.setText("退 出");

jButton2.setBounds(new Rectangle(375, 274, 90, 27));

jButton2.addActionListener(new StudentInfoAddFrame\_jButton2\_actionAdapter(this));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jLabel10.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel10.setText("所在班级：");

jLabel10.setBounds(new Rectangle(50, 224, 93, 29));

jComboBox1.setEnabled(false);

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 16));

jComboBox1.setBounds(new Rectangle(150, 224, 125, 25));

jOptionPane\_LoginFeedback.setBounds(new Rectangle(28, 263, 262, 90));

jOptionPane\_LoginFeedback.setLayout(null);

contentPane.add(jLabel2);

contentPane.add(jLabel1);

contentPane.add(jLabel8);

contentPane.add(jComboBox5);

contentPane.add(jLabel7);

contentPane.add(jComboBox4);

contentPane.add(jTextField1);

contentPane.add(jLabel4);

contentPane.add(jLabel6);

contentPane.add(jTextField2);

contentPane.add(jButton1);

contentPane.add(jButton2);

contentPane.add(jLabel10);

contentPane.add(jComboBox1);

contentPane.add(jOptionPane\_LoginFeedback);

this.jComboBox4.addItem("请选择学院");

// 选择学院

try {

ResultSet rs = con.getRs("select \* from tb\_depart ");

while (rs.next()) {

String xibu = rs.getString("departName");

jComboBox4.addItem(xibu);

}

} catch (Exception ce) {

System.out.println(ce);

}

jComboBox5.setEnabled(false);

jComboBox1.setEnabled(false);

}

// 选择学院触发

public void jComboBox4\_actionPerformed(ActionEvent e) {

jcb();

jComboBox5.setEnabled(true);

}

// 选择专业触发

public void jComboBox5\_actionPerformed(ActionEvent e) {

classAdd();

jComboBox1.setEnabled(true);

}

// 学生信息录入

public void jButton1\_actionPerformed(ActionEvent e) {

// 异常判断

if (jTextField1.getText().trim().equals("")) {

jOptionPane1.showMessageDialog(this, "请输入学生学号！", "提 示", jOptionPane1.INFORMATION\_MESSAGE);

} else if (jTextField2.getText().trim().equals("")) {

jOptionPane1.showMessageDialog(this, "请输入学生姓名！", "提 示", jOptionPane1.INFORMATION\_MESSAGE);

} else if (jComboBox4.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择学生所在学院！", "提 示", jOptionPane1.INFORMATION\_MESSAGE);

} else if (jComboBox5.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择学生所属专业！", "提 示", jOptionPane1.INFORMATION\_MESSAGE);

} else if (jComboBox1.getSelectedIndex() == 0) {

jOptionPane1.showMessageDialog(this, "请选择学生所在班级！", "提 示", jOptionPane1.INFORMATION\_MESSAGE);

} else {

try {

findDepart();

findSpec();

findClass();

boolean Num = false;

ResultSet rs = conn.getRs("select \* from tb\_student");

while (rs.next()) {

if (jTextField1.getText().trim().equals(rs.getString("stuNumber").trim())) {

Num = true;

}

}

rs.close();

if (Num) {

jOptionPane1.showMessageDialog(this, "学生学号已存在，请重新输入！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

number = jTextField1.getText().trim();

name = jTextField2.getText().trim();

instu();

}

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 查询学院

public void findDepart() {

try {

ResultSet rs = conn.getRs(

"select \* from tb\_depart where departName='" + String.valueOf(jComboBox4.getSelectedItem()) + "' ");

while (rs.next()) {

sdepart = rs.getString("departName");

}

rs.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

// 查询专业

public void findSpec() {

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where specName='" + String.valueOf(jComboBox5.getSelectedItem()) + "' ");

while (rs.next()) {

sspec = rs.getString("specName");

}

rs.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

// 查询班级

public void findClass() {

try {

ResultSet rs = conn.getRs(

"select \* from tb\_class where className='" + String.valueOf(jComboBox1.getSelectedItem()) + "' ");

while (rs.next()) {

sclass = rs.getString("className");

}

rs.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

// 选择专业

public void jcb() {

jComboBox5.removeAllItems();

jComboBox5.addItem("请选择专业");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_spec where departName='" + String.valueOf(jComboBox4.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("specName");

jComboBox5.addItem(zhy);

}

rs.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

// 选择班级

public void classAdd() {

jComboBox1.removeAllItems();

jComboBox1.addItem("请选择班级");

try {

ResultSet rs = conn.getRs(

"select \* from tb\_class where specName='" + String.valueOf(jComboBox5.getSelectedItem()) + "' ");

while (rs.next()) {

String zhy = rs.getString("className");

jComboBox1.addItem(zhy);

}

rs.close();

} catch (Exception e) {

System.out.println(e);

}

}

// 学生信息录入

public void instu() {

try {

if (0 < conn.getUpdate("insert into tb\_student (stuNumber,stuName,stuDepart,stuSpec,stuClass) values ('"

+ number + "','" + name + "','" + sdepart + "','" + sspec + "','" + sclass + "')")) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "学生信息录入成功！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else {

System.err.printf("修改 tb\_student 表中 stuNumber = %d 的记录失败\n", number);

jOptionPane\_LoginFeedback.showMessageDialog(this, "学生信息录入失败！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

}

}

class StudentInfoAddFrame\_jButton2\_actionAdapter implements ActionListener {

private StudentInfoAddFrame adaptee;

StudentInfoAddFrame\_jButton2\_actionAdapter(StudentInfoAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

class StudentInfoAddFrame\_jButton1\_actionAdapter implements ActionListener {

private StudentInfoAddFrame adaptee;

StudentInfoAddFrame\_jButton1\_actionAdapter(StudentInfoAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class StudentInfoAddFrame\_jComboBox5\_actionAdapter implements ActionListener {

private StudentInfoAddFrame adaptee;

StudentInfoAddFrame\_jComboBox5\_actionAdapter(StudentInfoAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox5\_actionPerformed(e);

}

}

class StudentInfoAddFrame\_jComboBox4\_actionAdapter implements ActionListener {

private StudentInfoAddFrame adaptee;

StudentInfoAddFrame\_jComboBox4\_actionAdapter(StudentInfoAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox4\_actionPerformed(e);

}

}

**StudentManager:**

package Student;

import javax.swing.table.DefaultTableModel;

import java.util.\*;

import javax.swing.JFrame;

import javax.swing.JLabel;

import java.awt.\*;

import com.borland.jbcl.layout.XYLayout;

import db.dbConn;

import com.borland.jbcl.layout.\*;

import stuManager.MainFrame;

import javax.swing.JComboBox;

import javax.swing.JPanel;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

/\*\*

\*

\* Title: 学生信息管理

\* Description: 学生信息管理模块，可以分别按学号、姓名、班级查询，是修改、删除学生信息的入口。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class StudentManager extends JFrame {

String sql, zhy;

JLabel jLabel1 = new JLabel();

XYLayout xYLayout1 = new XYLayout();

JLabel jLabel2 = new JLabel();

JComboBox jComboBox1 = new JComboBox();

JPanel jPanel1 = new JPanel();

JLabel jLabel3 = new JLabel();

JButton jButton1 = new JButton();

JPanel jPanel2 = new JPanel();

XYLayout xYLayout2 = new XYLayout();

JLabel jLabel4 = new JLabel();

JTextField jTextField2 = new JTextField();

JButton jButton2 = new JButton();

ButtonGroup buttonGroup1 = new ButtonGroup();

JPanel jPanel4 = new JPanel();

JLabel jLabel6 = new JLabel();

XYLayout xYLayout4 = new XYLayout();

JTextField jTextField3 = new JTextField();

JButton jButton4 = new JButton();

XYLayout xYLayout5 = new XYLayout();

JTextField jTextField1 = new JTextField();

JButton jButton5 = new JButton();

JButton jButton6 = new JButton();

JScrollPane jScrollPane1 = new JScrollPane();

JTable jTable1 = new JTable();

int intRow;

dbConn conn = new dbConn();

String[] arrField = { "学号", "姓名", "学院", "专业", "班级" };

DefaultTableModel model = new DefaultTableModel();

Object[][] arrData = {};

static int find;

JButton jButton7 = new JButton();

JOptionPane jOptionPane\_LoginFeedback = new JOptionPane();

public StudentManager() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

getContentPane().setLayout(xYLayout1);

jLabel1.setFont(new java.awt.Font("新宋体", Font.BOLD, 27));

jLabel1.setText("学生信息管理");

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jLabel2.setText("请选择查询方式：");

this.setTitle("学生信息管理");

jPanel1.setBorder(BorderFactory.createEtchedBorder());

jPanel1.setLayout(xYLayout5);

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jLabel3.setText("请输入学号：");

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jButton1.setText("查询");

jButton1.addActionListener(new StudentC\_jButton1\_actionAdapter(this));

jPanel2.setBorder(BorderFactory.createEtchedBorder());

jPanel2.setLayout(xYLayout2);

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jLabel4.setText("请输入学生姓名：");

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jButton2.setText("查询");

jButton2.addActionListener(new StudentC\_jButton2\_actionAdapter(this));

jPanel4.setBorder(BorderFactory.createEtchedBorder());

jPanel4.setLayout(xYLayout4);

jLabel6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jLabel6.setText("请输入要查询的班级：");

jButton4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jButton4.setText("查询");

jButton4.addActionListener(new StudentC\_jButton4\_actionAdapter(this));

jComboBox1.addActionListener(new StudentC\_jComboBox1\_actionAdapter(this));

jComboBox1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jTextField3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jTextField2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jButton5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jButton5.setText("修改");

jButton5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(ActionEvent e) {

jButton5\_actionPerformed(e);

}

});

jButton6.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jButton6.setText("返回");

jButton6.addActionListener(new StudentC\_jButton6\_actionAdapter(this));

jScrollPane1.setBorder(BorderFactory.createEtchedBorder());

jTable1.setCellSelectionEnabled(true);

jButton7.setFont(new java.awt.Font("Dialog", Font.PLAIN, 20));

jButton7.setText("删除");

jButton7.addActionListener(new StudentC\_jButton7\_actionAdapter(this));

this.getContentPane().add(jPanel2, new XYConstraints(150, 125, 510, 50));

jPanel2.add(jButton2, new XYConstraints(381, 8, 85, 27));

this.getContentPane().add(jPanel4, new XYConstraints(150, 125, 510, 50));

jPanel4.add(jButton4, new XYConstraints(388, 8, 89, 26));

ShowPanel();

jComboBox1.addItem("请选择");

jComboBox1.addItem("按学号查询");

jComboBox1.addItem("按姓名查询");

jComboBox1.addItem("按班级查询");

jPanel1.add(jTextField1, new XYConstraints(164, 8, 149, 25));

jPanel1.add(jButton1, new XYConstraints(360, 8, 80, 29));

jPanel1.add(jLabel3, new XYConstraints(29, 8, 125, 26));

jPanel2.add(jLabel4, new XYConstraints(21, 8, 164, 31));

jPanel2.add(jTextField2, new XYConstraints(206, 8, 132, 27));

jPanel4.add(jLabel6, new XYConstraints(16, 8, 201, 30));

jPanel4.add(jTextField3, new XYConstraints(212, 8, 152, 28));

this.getContentPane().add(jLabel1, new XYConstraints(321, 12, 180, 38));

this.getContentPane().add(jComboBox1, new XYConstraints(370, 76, -1, 30));

this.getContentPane().add(jLabel2, new XYConstraints(169, 74, 161, 32));

this.getContentPane().add(jPanel1, new XYConstraints(150, 125, 510, 50));

jScrollPane1.getViewport().add(jTable1, null);

this.getContentPane().add(jScrollPane1, new XYConstraints(20, 193, 759, 325));

this.getContentPane().add(jButton5, new XYConstraints(225, 545, 95, 34));

this.getContentPane().add(jButton6, new XYConstraints(525, 545, 95, 34));

this.getContentPane().add(jButton7, new XYConstraints(375, 545, 95, 34));

// 用户等级判断

// 1为管理员

if (MainFrame.level.equals("1")) {

// 不隐藏任何功能

}

// 2为普通用户

else if (MainFrame.level.equals("2")) {

// 隐藏部分功能

jButton5.setVisible(false);

jButton7.setVisible(false);

}

// 查找所有学生

sql = "select \* from tb\_student";

// 更新显示

UpdateRecord();

}

// 更新显示

public void UpdateRecord() {

Object[][] arrTmp = {}; // 设定表格的字段

Vector vec = new Vector(1, 1);

model = new DefaultTableModel(arrTmp, arrField);

jTable1 = new JTable(model);

jScrollPane1.getViewport().add(jTable1, null);

try {

ResultSet rs = conn.getRs(sql);

while (rs.next()) {

vec = new Vector();

vec.add(rs.getString("stuNumber").trim());

vec.add(rs.getString("stuName").trim());

vec.add(rs.getString("stuDepart").trim());

vec.add(rs.getString("stuSpec").trim());

vec.add(rs.getString("stuClass").trim());

model.addRow(vec);

}

} catch (Exception e) {

e.printStackTrace();

}

jScrollPane1.getHorizontalScrollBar();

jTable1.setGridColor(Color.blue);

jTable1.setDragEnabled(true);

jTable1.setSelectionForeground(Color.red);

jTable1.setSelectionBackground(Color.green);

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.setRowSelectionAllowed(true);

jTable1.setShowVerticalLines(true);

}

// 查询方式

public void ShowPanel() {

jPanel1.setVisible(false);

jPanel2.setVisible(false);

jPanel4.setVisible(false);

if (jComboBox1.getSelectedIndex() == 1)

jPanel1.setVisible(true);

else if (jComboBox1.getSelectedIndex() == 2)

jPanel2.setVisible(true);

else if (jComboBox1.getSelectedIndex() == 3)

jPanel4.setVisible(true);

}

// 获取选定的行

public void getM() {

intRow = jTable1.getSelectedRow();

if (intRow == -1)

return;

try {

find = Integer.parseInt(model.getValueAt(intRow, 0).toString());

System.err.println(find);

} catch (Exception e) {

e.printStackTrace();

}

}

// 退出

public void jButton6\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 按学号查询

public void jButton1\_actionPerformed(ActionEvent e) {

if (!jTextField1.getText().trim().equals("")) {

sql = "select \* from tb\_student where stuNumber='" + jTextField1.getText().trim() + "'";

UpdateRecord();

} else

jOptionPane\_LoginFeedback.showMessageDialog(this, "请输入要查询的学生学号！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

// 选择触发

public void jComboBox1\_actionPerformed(ActionEvent e) {

ShowPanel();

}

// 按姓名查询

public void jButton2\_actionPerformed(ActionEvent e) {

if (!jTextField2.getText().trim().equals("")) {

sql = "select \* from tb\_student where stuName='" + jTextField2.getText().trim() + "'";

UpdateRecord();

} else

jOptionPane\_LoginFeedback.showMessageDialog(this, "请输入要查询的学生姓名！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

// 按班级查询

public void jButton4\_actionPerformed(ActionEvent e) {

if (!jTextField3.getText().trim().equals("")) {

sql = "select \* from tb\_student where stuClass='" + jTextField3.getText().trim() + "'";

UpdateRecord();

} else

jOptionPane\_LoginFeedback.showMessageDialog(this, "请输入要查询的班级！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

// 修改

public void jButton5\_actionPerformed(ActionEvent e) {

getM();

if (intRow != -1) {

StudentChange siadd = new StudentChange(find);

siadd.setLocation(400, 200);

siadd.setSize(592, 350);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

this.dispose();

} else

jOptionPane\_LoginFeedback.showMessageDialog(this, "请选择要修改的信息！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

// 触发删除

public void jButton7\_actionPerformed(ActionEvent e) {

getM();

if (intRow != -1) {

delstu();

} else

jOptionPane\_LoginFeedback.showMessageDialog(this, "请选择要删除的信息！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

// 删除

public void delstu() {

try {

if (0 < conn.getUpdate("delete from tb\_student where stuNumber='" + find + "'")) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "学生信息删除成功！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else {

System.err.printf("删除 tb\_student 表中 stuNumber = %d 的记录失败\n", find);

jOptionPane\_LoginFeedback.showMessageDialog(this, "学生信息删除失败！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

} catch (Exception ce) {

System.out.println(ce.getMessage());

}

}

}

class StudentC\_jButton7\_actionAdapter implements ActionListener {

private StudentManager adaptee;

StudentC\_jButton7\_actionAdapter(StudentManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton7\_actionPerformed(e);

}

}

class StudentC\_jButton4\_actionAdapter implements ActionListener {

private StudentManager adaptee;

StudentC\_jButton4\_actionAdapter(StudentManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton4\_actionPerformed(e);

}

}

class StudentC\_jButton2\_actionAdapter implements ActionListener {

private StudentManager adaptee;

StudentC\_jButton2\_actionAdapter(StudentManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

class StudentC\_jButton1\_actionAdapter implements ActionListener {

private StudentManager adaptee;

StudentC\_jButton1\_actionAdapter(StudentManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class StudentC\_jButton6\_actionAdapter implements ActionListener {

private StudentManager adaptee;

StudentC\_jButton6\_actionAdapter(StudentManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton6\_actionPerformed(e);

}

}

class StudentC\_jComboBox1\_actionAdapter implements ActionListener {

private StudentManager adaptee;

StudentC\_jComboBox1\_actionAdapter(StudentManager adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jComboBox1\_actionPerformed(e);

}

}

**stuManager:**

**LoginClass:**

package stuManager;

import java.awt.Toolkit;

import java.io.File;

import javax.swing.SwingUtilities;

import javax.swing.UIManager;

import java.awt.Dimension;

/\*\*

\*

\* Title: 登录类

\* Description: 登陆模块，程序的入口类

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class LoginClass {

boolean packFrame = false;

public LoginClass() {

LoginFrame frame = new LoginFrame();

if (packFrame) {

frame.pack();

} else {

frame.validate();

}

// 调整窗体大小

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

Dimension frameSize = frame.getSize();

if (frameSize.height > screenSize.height) {

frameSize.height = screenSize.height;

}

if (frameSize.width > screenSize.width) {

frameSize.width = screenSize.width;

}

frame.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);

frame.setVisible(true);

}

// 主方法

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

} catch (Exception e) {

e.printStackTrace();

}

new LoginClass();

}

});

}

}

LoginFrame:

package stuManager;

import javax.swing.JFrame;

import java.awt.Dimension;

import javax.swing.JPanel;

import java.awt.\*;

import javax.swing.JLabel;

import javax.swing.JTextField;

import User.UserAddFrame;

import db.dbConn;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.net.URL;

import java.sql.ResultSet;

/\*\*

\*

\* Title: 登录模块

\* Description: 登录模块的可视化实现

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class LoginFrame extends JFrame {

JPanel contentPane;

URL url = getClass().getResource("/img/bg.jpg");

ImageIcon loginBg = new ImageIcon(url);

JLabel jLabel\_welcomeTitle = new JLabel();

JLabel jLabel\_systemTitle = new JLabel();

JLabel jLabel\_userName = new JLabel();

JLabel jLabel\_pwd = new JLabel();

JTextField jTextField\_userName = new JTextField();

JButton jButton\_Login = new JButton();

JButton jButton\_quit = new JButton();

JLabel jLabel\_systemTitle2 = new JLabel();

JLabel jLabel\_welcomeTitle2 = new JLabel();

JOptionPane jOptionPane\_LoginFeedback = new JOptionPane();

JPasswordField jPasswordField\_pwd = new JPasswordField();

JPanel jPanel\_bgBox = new JPanel();

JLabel jLabel\_bg = new JLabel();

CardLayout cardLayout1 = new CardLayout();

static String level, name;

dbConn conn = new dbConn();

public LoginFrame() {

try {

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(504, 344));

setTitle("欢迎登录学生管理系统");

jLabel\_welcomeTitle.setFont(new java.awt.Font("Dialog", Font.BOLD, 25));

jLabel\_welcomeTitle.setToolTipText("");

jLabel\_welcomeTitle.setText("欢 迎 登 录 学 生");

jLabel\_welcomeTitle.setBounds(new Rectangle(128, 19, 246, 30));

jLabel\_systemTitle.setFont(new java.awt.Font("Dialog", Font.BOLD, 25));

jLabel\_systemTitle.setToolTipText("");

jLabel\_systemTitle.setText("管 理 系 统");

jLabel\_systemTitle.setBounds(new Rectangle(173, 61, 164, 29));

jLabel\_userName.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel\_userName.setText("用户名:");

jLabel\_userName.setBounds(new Rectangle(95, 123, 77, 25));

jLabel\_pwd.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel\_pwd.setText("密 码:");

jLabel\_pwd.setBounds(new Rectangle(95, 172, 74, 21));

jTextField\_userName.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jTextField\_userName.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField\_userName.setBounds(new Rectangle(188, 122, 212, 27));

jButton\_Login.setBounds(new Rectangle(124, 243, 93, 30));

jButton\_Login.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jButton\_Login.setBorder(BorderFactory.createRaisedBevelBorder());

jButton\_Login.setText("登 录");

jButton\_Login.addActionListener(new LoginFrame\_jButton\_Login\_actionAdapter(this));

jButton\_quit.setBounds(new Rectangle(286, 243, 93, 30));

jButton\_quit.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jButton\_quit.setBorder(BorderFactory.createRaisedBevelBorder());

jButton\_quit.setText("注 册");

jButton\_quit.addActionListener(new LoginFrame\_jButton\_SignUp\_actionAdapter(this));

jLabel\_systemTitle2.setFont(new java.awt.Font("Dialog", Font.BOLD, 25));

jLabel\_systemTitle2.setForeground(Color.blue);

jLabel\_systemTitle2.setToolTipText("");

jLabel\_systemTitle2.setText("管 理 系 统");

jLabel\_systemTitle2.setBounds(new Rectangle(173, 62, 164, 29));

jLabel\_welcomeTitle2.setFont(new java.awt.Font("Dialog", Font.BOLD, 25));

jLabel\_welcomeTitle2.setForeground(Color.blue);

jLabel\_welcomeTitle2.setToolTipText("");

jLabel\_welcomeTitle2.setText("欢 迎 登 录 学 生");

jLabel\_welcomeTitle2.setBounds(new Rectangle(128, 20, 246, 30));

jOptionPane\_LoginFeedback.setBounds(new Rectangle(28, 263, 262, 90));

jOptionPane\_LoginFeedback.setLayout(null);

jPasswordField\_pwd.setBorder(BorderFactory.createLoweredBevelBorder());

jPasswordField\_pwd.setBounds(new Rectangle(187, 168, 213, 26));

jPanel\_bgBox.setBounds(new Rectangle(-15, 0, 529, 375));

jPanel\_bgBox.setLayout(cardLayout1);

jLabel\_bg.setIcon(loginBg);

contentPane.add(jTextField\_userName);

contentPane.add(jLabel\_userName);

contentPane.add(jLabel\_pwd);

contentPane.add(jButton\_Login);

contentPane.add(jButton\_quit);

contentPane.add(jLabel\_welcomeTitle);

contentPane.add(jLabel\_systemTitle);

contentPane.add(jLabel\_welcomeTitle2);

contentPane.add(jLabel\_systemTitle2);

contentPane.add(jPasswordField\_pwd);

contentPane.add(jPanel\_bgBox);

jPanel\_bgBox.add(jLabel\_bg, "jLabel8");

contentPane.add(jOptionPane\_LoginFeedback);

}

// 添加用户模块

public void jButton\_SignUp\_actionPerformed(ActionEvent e) {

UserAddFrame siadd = new UserAddFrame();

siadd.setLocation(400, 200);

siadd.setSize(469, 315);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 登录

public void jButton\_Login\_actionPerformed(ActionEvent e) {

boolean cheng = false;

// 异常判断

if (jTextField\_userName.getText().length() == 0) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "用户名不能为空！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

} else {

try {

ResultSet rs = conn

.getRs("select \* from tb\_user where userName='" + jTextField\_userName.getText() + "'");

while (rs.next()) {

if (rs.getString("userPwd").trim().equals(this.jPasswordField\_pwd.getText())) {

level = rs.getString("UserType").trim();

name = jTextField\_userName.getText().trim();

MainFrame main = new MainFrame(level, jTextField\_userName.getText());

main.setLocation(200, 40);

main.setSize(900, 650);

main.setVisible(true);

main.setResizable(false);

main.validate();

this.dispose();

cheng = false;

break;

} else {

cheng = true;

}

}

if (cheng) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "用户名或密码错误！", "提 示",

jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

rs.close();

} catch (Exception ce) {

System.out.println(ce);

}

}

}

}

class LoginFrame\_jButton\_Login\_actionAdapter implements ActionListener {

private LoginFrame adaptee;

LoginFrame\_jButton\_Login\_actionAdapter(LoginFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton\_Login\_actionPerformed(e);

}

}

class LoginFrame\_jButton\_SignUp\_actionAdapter implements ActionListener {

private LoginFrame adaptee;

LoginFrame\_jButton\_SignUp\_actionAdapter(LoginFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton\_SignUp\_actionPerformed(e);

}

}

**MainFrame:**

package stuManager;

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.net.URL;

import javax.swing.\*;

import Class.ClassAddFrame;

import Class.ClassManager;

import Cource.CourceAddFrame;

import Cource.CourceManager;

import Depart.DepartAddFrame;

import Depart.DepartManager;

import Score.ScoreAddFrame;

import Score.ScoreManager;

import Student.StudentInfoAddFrame;

import Student.StudentManager;

import User.UserAddFrame;

import User.UserDelete;

import User.UserPasswordFrame;

/\*\*

\*

\* Title: 主界面

\* Description: 主界面模块，有对学生、班级、院系、课程、成绩、用户的录入与管理的菜单

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class MainFrame extends JFrame {

JPanel contentPane;

public static String level;

static String name;

URL url = getClass().getResource("/img/bg.jpg");

ImageIcon mainBg = new ImageIcon(url);

JLabel jLabel\_welcomeTitle = new JLabel();

JLabel jLabel\_welcomeTitle2 = new JLabel();

JLabel jLabel\_help1 = new JLabel();

JLabel jLabel\_help2 = new JLabel();

JLabel jLabel\_help3 = new JLabel();

JMenuBar jMenuBar = new JMenuBar();

JMenu jMenu\_stuManager = new JMenu();

JMenuItem jMenuItem\_stuInfoSignUp = new JMenuItem();

JMenuItem jMenuItem\_stuInfoInquire = new JMenuItem();

JMenu jMenu\_classManager = new JMenu();

JMenuItem jMenuItem\_classInput = new JMenuItem();

JMenuItem jMenuItem\_classManager = new JMenuItem();

JMenu jMenu\_departManager = new JMenu();

JMenuItem jMenuItem\_departInput = new JMenuItem();

JMenuItem jMenuItem\_departManager = new JMenuItem();

JMenu jMenu\_courceManager = new JMenu();

JMenuItem jMenuItem\_courceInput = new JMenuItem();

JMenuItem jMenuItem\_courceManager = new JMenuItem();

JMenu jMenu\_scoreManager = new JMenu();

JMenuItem jMenuItem\_scoreInput = new JMenuItem();

JMenuItem jMenuItem\_scoreManager = new JMenuItem();

JMenu jMenu\_userManager = new JMenu();

JMenuItem jMenuItem\_userSignUp = new JMenuItem();

JMenuItem jMenuItem\_userChangePwd = new JMenuItem();

JMenuItem jMenuItem\_userDelete = new JMenuItem();

JMenu jMenu\_about = new JMenu();

JMenuItem jMenuItem\_about = new JMenuItem();

JMenu jMenu\_quit = new JMenu();

JMenuItem jMenuItem\_quit = new JMenuItem();

JMenuItem jMenuItem\_reboot = new JMenuItem();

JPanel jPane\_box = new JPanel();

JLabel jLabe\_mainBgBox = new JLabel();

BorderLayout borderLayout = new BorderLayout();

JOptionPane jOptionPane\_LoginFeedback = new JOptionPane();

public MainFrame(String level, String name) {

this.level = level;

this.name = name;

try {

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

this.setJMenuBar(jMenuBar);

setSize(new Dimension(911, 698));

setTitle("欢迎登录学生管理系统");

jLabel\_welcomeTitle.setText("欢 迎 使 用 学 生 信 息 管 理 系 统");

jLabel\_welcomeTitle.setBounds(new Rectangle(180, 130, 600, 33));

jLabel\_welcomeTitle.setFont(new java.awt.Font("Dialog", Font.BOLD, 30));

jLabel\_welcomeTitle.setForeground(Color.lightGray);

jLabel\_welcomeTitle2.setFont(new java.awt.Font("Dialog", Font.BOLD, 30));

jLabel\_welcomeTitle2.setText("欢 迎 使 用 学 生 信 息 管 理 系 统");

jLabel\_welcomeTitle2.setBounds(new Rectangle(181, 131, 608, 33));

jLabel\_help1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel\_help1.setText("您可以对：");

jLabel\_help1.setBounds(new Rectangle(500, 340, 400, 30));

jLabel\_help2.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel\_help2.setText("学生、班级、院系、课程、成绩");

jLabel\_help2.setBounds(new Rectangle(500, 370, 400, 30));

jLabel\_help3.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel\_help3.setText("进行录入、查询、修改、删除的操作");

jLabel\_help3.setBounds(new Rectangle(500, 400, 400, 30));

jMenu\_stuManager.setText("学生管理");

jMenuItem\_stuInfoSignUp.setText(" 录 入");

jMenuItem\_stuInfoSignUp.addActionListener(new MainFrame\_jMenuItem\_stuInfoSignUp\_actionAdapter(this));

jMenuItem\_stuInfoInquire.setText(" 管 理");

jMenuItem\_stuInfoInquire.addActionListener(new MainFrame\_jMenuItem\_stuInfoInquire\_actionAdapter(this));

jMenu\_classManager.setText("班级管理");

jMenuItem\_classInput.setText(" 录 入");

jMenuItem\_classInput.addActionListener(new MainFrame\_jMenuItem\_classInput\_actionAdapter(this));

jMenuItem\_classManager.setText(" 管 理");

jMenuItem\_classManager.addActionListener(new MainFrame\_jMenuItem\_classManager\_actionAdapter(this));

jMenu\_departManager.setText("院系管理");

jMenuItem\_departInput.setText(" 录 入");

jMenuItem\_departInput.addActionListener(new MainFrame\_jMenuItem\_departInput\_actionAdapter(this));

jMenuItem\_departManager.setText(" 管 理");

jMenuItem\_departManager.addActionListener(new MainFrame\_jMenuItem\_departManager\_actionAdapter(this));

jMenu\_courceManager.setText("课程管理");

jMenuItem\_courceInput.setText(" 录 入");

jMenuItem\_courceInput.addActionListener(new MainFrame\_jMenuItem\_courceInput\_actionAdapter(this));

jMenuItem\_courceManager.setText(" 管 理");

jMenuItem\_courceManager.addActionListener(new MainFrame\_jMenuItem\_courceManager\_actionAdapter(this));

jMenu\_scoreManager.setText("成绩管理");

jMenuItem\_scoreInput.setText(" 录 入");

jMenuItem\_scoreInput.addActionListener(new MainFrame\_jMenuItem\_scoreInput\_actionAdapter(this));

jMenuItem\_scoreManager.setText(" 管 理");

jMenuItem\_scoreManager.addActionListener(new MainFrame\_jMenuItem\_scoreManager\_actionAdapter(this));

jMenu\_userManager.setText("用户管理");

jMenuItem\_userSignUp.setText("添加用户");

jMenuItem\_userSignUp.addActionListener(new MainFrame\_jMenuItem\_userSignUp\_actionAdapter(this));

jMenuItem\_userChangePwd.setText("修改密码");

jMenuItem\_userChangePwd.addActionListener(new MainFrame\_jMenuItem\_userChangePwd\_actionAdapter(this));

jMenuItem\_userDelete.setEnabled(true);

jMenuItem\_userDelete.setText("删除用户");

jMenuItem\_userDelete.addActionListener(new MainFrame\_jMenuItem\_userDelete\_actionAdapter(this));

jMenu\_about.setText("关于");

jMenuItem\_about.setText(" 关 于");

jMenuItem\_about.addActionListener(new MainFrame\_jMenuItem\_about\_actionAdapter(this));

jMenu\_quit.setText("退出");

jMenuItem\_quit.setText(" 退 出");

jMenuItem\_quit.addActionListener(new MainFrame\_jMenuItem\_quit\_actionAdapter(this));

jMenuItem\_reboot.setText("重新启动");

jMenuItem\_reboot.addActionListener(new MainFrame\_jMenuItem\_reboot\_actionAdapter(this));

jPane\_box.setBounds(new Rectangle(-6, 0, 900, 700));

jPane\_box.setLayout(borderLayout);

jLabe\_mainBgBox.setIcon(mainBg);

contentPane.add(jLabel\_welcomeTitle2);

contentPane.add(jLabel\_welcomeTitle);

contentPane.add(jLabel\_help1);

contentPane.add(jLabel\_help2);

contentPane.add(jLabel\_help3);

contentPane.add(jPane\_box);

jPane\_box.add(jLabe\_mainBgBox, java.awt.BorderLayout.NORTH);

jMenuBar.add(jMenu\_stuManager);

jMenuBar.add(jMenu\_classManager);

jMenuBar.add(jMenu\_departManager);

jMenuBar.add(jMenu\_courceManager);

jMenuBar.add(jMenu\_scoreManager);

jMenuBar.add(jMenu\_userManager);

jMenuBar.add(jMenu\_about);

jMenuBar.add(jMenu\_quit);

jMenu\_stuManager.add(jMenuItem\_stuInfoSignUp);

jMenu\_stuManager.add(jMenuItem\_stuInfoInquire);

jMenu\_classManager.add(jMenuItem\_classInput);

jMenu\_classManager.add(jMenuItem\_classManager);

jMenu\_departManager.add(jMenuItem\_departInput);

jMenu\_departManager.add(jMenuItem\_departManager);

jMenu\_courceManager.add(jMenuItem\_courceInput);

jMenu\_courceManager.add(jMenuItem\_courceManager);

jMenu\_scoreManager.add(jMenuItem\_scoreInput);

jMenu\_scoreManager.add(jMenuItem\_scoreManager);

jMenu\_userManager.add(jMenuItem\_userSignUp);

jMenu\_userManager.add(jMenuItem\_userDelete);

jMenu\_userManager.add(jMenuItem\_userChangePwd);

jMenu\_about.add(jMenuItem\_about);

jMenu\_quit.add(jMenuItem\_quit);

jMenu\_quit.add(jMenuItem\_reboot);

// 用户等级判断

// 1为管理员

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

// 不隐藏任何功能

}

// 2为普通用户

else if (level.equals("2")) {

// 隐藏部分功能

this.jMenuItem\_userSignUp.setVisible(false);

this.jMenuItem\_userDelete.setVisible(false);

this.jMenuItem\_stuInfoSignUp.setVisible(false);

this.jMenuItem\_classInput.setVisible(false);

this.jMenuItem\_departInput.setVisible(false);

this.jMenuItem\_courceInput.setVisible(false);

this.jMenuItem\_scoreInput.setVisible(false);

}

}

// 关于信息

public void jMenuItem\_about\_actionPerformed(ActionEvent e) {

jOptionPane\_LoginFeedback.showMessageDialog(this, "学生信息管理系统\n版本号：1.0\nauthor：黄策 陈子权 翟昆洋 饶蕊\ngithub：https://github.com/dsaDadas11/stuM\n\n ^\_^",

"关于", jOptionPane\_LoginFeedback.INFORMATION\_MESSAGE);

}

// 退出

public void jMenuItem\_quit\_actionPerformed(ActionEvent e) {

System.exit(0);

}

// 重启

public void jMenuItem\_reboot\_actionPerformed(ActionEvent e) {

this.dispose();

LoginFrame login = new LoginFrame();

login.setLocation(400, 200);

login.setSize(504, 344);

login.setVisible(true);

login.setResizable(false);

login.validate();

}

// 学生录入模块

public void jMenuItem\_stuInfoSignUp\_actionPerformed(ActionEvent e) {

StudentInfoAddFrame siadd = new StudentInfoAddFrame();

siadd.setLocation(400, 200);

siadd.setSize(592, 350);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 学生管理模块

public void jMenuItem\_stuInfoInquire\_actionPerformed(ActionEvent e) {

StudentManager siadd = new StudentManager();

siadd.setLocation(250, 60);

siadd.setSize(800, 620);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 班级录入模块

public void jMenuItem\_classInput\_actionPerformed(ActionEvent e) {

ClassAddFrame siadd = new ClassAddFrame();

siadd.setLocation(400, 200);

siadd.setSize(465, 310);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 班级管理模块

public void jMenuItem\_classManager\_actionPerformed(ActionEvent e) {

ClassManager siadd = new ClassManager();

siadd.setLocation(400, 100);

siadd.setSize(530, 560);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 院系录入模块

public void jMenuItem\_departInput\_actionPerformed(ActionEvent e) {

DepartAddFrame siadd = new DepartAddFrame();

siadd.setLocation(400, 200);

siadd.setSize(465, 310);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 院系管理模块

public void jMenuItem\_departManager\_actionPerformed(ActionEvent e) {

DepartManager siadd = new DepartManager();

siadd.setLocation(400, 100);

siadd.setSize(530, 560);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 课程录入模块

public void jMenuItem\_courceInput\_actionPerformed(ActionEvent e) {

CourceAddFrame siadd = new CourceAddFrame();

siadd.setLocation(400, 200);

siadd.setSize(482, 320);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 课程管理模块

public void jMenuItem\_courceManager\_actionPerformed(ActionEvent e) {

CourceManager siadd = new CourceManager();

siadd.setLocation(400, 100);

siadd.setSize(530, 560);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 成绩录入模块

public void jMenuItem\_scoreInput\_actionPerformed(ActionEvent e) {

ScoreAddFrame siadd = new ScoreAddFrame();

siadd.setLocation(400, 200);

siadd.setSize(482, 320);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 成绩管理模块

public void jMenuItem\_scoreManager\_actionPerformed(ActionEvent e) {

ScoreManager siadd = new ScoreManager();

siadd.setLocation(400, 100);

siadd.setSize(550, 560);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 添加用户模块

public void jMenuItem\_userSignUp\_actionPerformed(ActionEvent e) {

UserAddFrame siadd = new UserAddFrame();

siadd.setLocation(400, 200);

siadd.setSize(469, 315);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 删除用户模块

public void jMenuItem\_userDelete\_actionPerformed(ActionEvent e) {

UserDelete siadd = new UserDelete();

siadd.setLocation(400, 200);

siadd.setSize(444, 280);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

// 修改密码模块

public void jMenuItem\_userChangePwd\_actionPerformed(ActionEvent e) {

UserPasswordFrame siadd = new UserPasswordFrame(level, name);

siadd.setLocation(400, 200);

siadd.setSize(444, 340);

siadd.setVisible(true);

siadd.setResizable(false);

siadd.validate();

}

}

class MainFrame\_jMenuItem\_courceManager\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_courceManager\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_courceManager\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_scoreManager\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_scoreManager\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_scoreManager\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_stuInfoInquire\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_stuInfoInquire\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_stuInfoInquire\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_classManager\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_classManager\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_classManager\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_classInput\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_classInput\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_classInput\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_departManager\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_departManager\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_departManager\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_departInput\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_departInput\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_departInput\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_courceInput\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_courceInput\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_courceInput\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_scoreInput\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_scoreInput\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_scoreInput\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_userSignUp\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_userSignUp\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_userSignUp\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_userChangePwd\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_userChangePwd\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_userChangePwd\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_userDelete\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_userDelete\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_userDelete\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_about\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_about\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_about\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_quit\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_quit\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_quit\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_stuInfoSignUp\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_stuInfoSignUp\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_stuInfoSignUp\_actionPerformed(e);

}

}

class MainFrame\_jMenuItem\_reboot\_actionAdapter implements ActionListener {

private MainFrame adaptee;

MainFrame\_jMenuItem\_reboot\_actionAdapter(MainFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jMenuItem\_reboot\_actionPerformed(e);

}

}

**User:**

**UserAddFrame:**

package User;

import java.awt.Dimension;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.\*;

import javax.swing.JLabel;

import javax.swing.JTextField;

import db.dbConn;

import javax.swing.\*;

import java.sql.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\*

\* Title: 用户添加

\* Description: 用户添加模块，只对管理员显示。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class UserAddFrame extends JFrame {

JPanel contentPane;

String level;

JLabel jLabel1 = new JLabel();

JLabel jLabel2 = new JLabel();

JLabel jLabel3 = new JLabel();

JLabel jLabel4 = new JLabel();

JTextField jTextField1 = new JTextField();

JPasswordField jPasswordField1 = new JPasswordField();

JPasswordField jPasswordField2 = new JPasswordField();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

public UserAddFrame() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(469, 315));

setTitle("添加用户");

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 23));

jLabel1.setText("添 加 用 户");

jLabel1.setBounds(new Rectangle(134, 12, 198, 27));

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel2.setText("用户名:");

jLabel2.setBounds(new Rectangle(90, 66, 68, 22));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel3.setText("密 码:");

jLabel3.setBounds(new Rectangle(90, 118, 75, 27));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel4.setText("确认密码:");

jLabel4.setBounds(new Rectangle(84, 167, 89, 29));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jTextField1.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField1.setBounds(new Rectangle(191, 65, 155, 24));

jPasswordField1.setBorder(BorderFactory.createLoweredBevelBorder());

jPasswordField1.setBounds(new Rectangle(191, 119, 155, 24));

jPasswordField2.setBorder(BorderFactory.createLoweredBevelBorder());

jPasswordField2.setText("");

jPasswordField2.setBounds(new Rectangle(191, 169, 155, 24));

jButton1.setBounds(new Rectangle(99, 225, 89, 25));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("提 交");

jButton1.addActionListener(new AddAdminFrame\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(267, 225, 89, 25));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setText("退 出");

jButton2.addActionListener(new AddAdminFrame\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(0, 233, 262, 90));

jOptionPane1.setLayout(null);

contentPane.add(jLabel2);

contentPane.add(jLabel3);

contentPane.add(jLabel4);

contentPane.add(jTextField1);

contentPane.add(jPasswordField2);

contentPane.add(jPasswordField1);

contentPane.add(jButton1);

contentPane.add(jButton2);

contentPane.add(jLabel1);

contentPane.add(jOptionPane1);

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 添加

public void jButton1\_actionPerformed(ActionEvent e) {

boolean zhi = false;

// 异常判断

if (jTextField1.getText().length() == 0) {

jOptionPane1.showMessageDialog(this, "用户名不能为空！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jPasswordField2.getText().trim().equals(jPasswordField1.getText().trim())) {

zhi = true;

} else {

jOptionPane1.showMessageDialog(this, "密码确认错误！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

if (zhi) {

try {

// 查询用户名是否存在

boolean name = false;

dbConn conn = new dbConn();

ResultSet rs = conn.getRs("select userName from tb\_user");

while (rs.next()) {

if (jTextField1.getText().trim().equals(rs.getString("userName").trim())) {

name = true;

}

}

if (name) {

jOptionPane1.showMessageDialog(this, "该用户名已经存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

// 不存在则添加

conn.getUpdate("insert into tb\_user (userName,userPwd,userType) values ('"

+ jTextField1.getText().trim() + "','" + jPasswordField2.getText().trim() + "','2')");

jOptionPane1.showMessageDialog(this, "恭喜您添加用户成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

rs.close();

} catch (Exception ce) {

System.out.println(ce);

}

}

}

}

class AddAdminFrame\_jButton1\_actionAdapter implements ActionListener {

private UserAddFrame adaptee;

AddAdminFrame\_jButton1\_actionAdapter(UserAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class AddAdminFrame\_jButton2\_actionAdapter implements ActionListener {

private UserAddFrame adaptee;

AddAdminFrame\_jButton2\_actionAdapter(UserAddFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**UserDelete:**

package User;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.Rectangle;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

import javax.swing.BorderFactory;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import db.dbConn;

/\*\*

\*

\* Title: 用户删除

\* Description: 用户删除模块，只对管理员显示。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class UserDelete extends JFrame {

JPanel contentPane;

String level;

JLabel jLabel1 = new JLabel();

JLabel jLabel2 = new JLabel();

JTextField jTextField1 = new JTextField();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

dbConn conn = new dbConn();

public UserDelete() {

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(469, 315));

setTitle("删除用户");

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 23));

jLabel1.setText("删 除 用 户");

jLabel1.setBounds(new Rectangle(134, 12, 198, 27));

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel2.setText("用户名:");

jLabel2.setBounds(new Rectangle(90, 100, 68, 22));

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jTextField1.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField1.setBounds(new Rectangle(191, 100, 155, 24));

jButton1.setBounds(new Rectangle(99, 180, 89, 25));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("确 认");

jButton1.addActionListener(new UserDeleteFrame\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(267, 180, 89, 25));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setText("退 出");

jButton2.addActionListener(new UserDeleteFrame\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(0, 233, 262, 90));

jOptionPane1.setLayout(null);

contentPane.add(jLabel2);

contentPane.add(jTextField1);

contentPane.add(jButton1);

contentPane.add(jButton2);

contentPane.add(jLabel1);

contentPane.add(jOptionPane1);

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 删除用户

public void jButton1\_actionPerformed(ActionEvent e) {

boolean isAdmin = false, noUser = true;

// 异常判断

if (jTextField1.getText().length() == 0) {

jOptionPane1.showMessageDialog(this, "用户名不能为空！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

try {

ResultSet rs = conn

.getRs("select \* from tb\_user where userName='" + jTextField1.getText().trim() + "'");

while (rs.next()) {

// 判断输入的用户名是否存在

noUser = false;

// 判断删除的用户是否管理员（无法删除管理员用户）

if (1 == Integer.valueOf(rs.getString("userType").trim())) {

isAdmin = true;

}

}

if (noUser) {

jOptionPane1.showMessageDialog(this, "该用户名不存在！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (isAdmin) {

jOptionPane1.showMessageDialog(this, "无法删除管理员账号！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

conn.getUpdate("delete from tb\_user where userName='" + jTextField1.getText().trim() + "'");

jOptionPane1.showMessageDialog(this, "恭喜您删除用户成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

rs.close();

} catch (Exception ce) {

System.out.println(ce);

}

}

}

}

class UserDeleteFrame\_jButton1\_actionAdapter implements ActionListener {

private UserDelete adaptee;

UserDeleteFrame\_jButton1\_actionAdapter(UserDelete adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class UserDeleteFrame\_jButton2\_actionAdapter implements ActionListener {

private UserDelete adaptee;

UserDeleteFrame\_jButton2\_actionAdapter(UserDelete adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**UserPasswordFrame:**

package User;

import java.awt.Dimension;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.\*;

import javax.swing.JLabel;

import javax.swing.JTextField;

import db.dbConn;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

/\*\*

\*

\* Title: 用户密码修改

\* Description: 用户密码修改模块，管理员和普通用户都可见，管理员可修改所有用户密码，普通用户只能修改本用户密码。

\*

\* @author 黄策，陈子权，翟昆洋，饶蕊

\*/

public class UserPasswordFrame extends JFrame {

JPanel contentPane;

String level;

String name;

JLabel jLabel1 = new JLabel();

JLabel jLabel2 = new JLabel();

JLabel jLabel3 = new JLabel();

JLabel jLabel4 = new JLabel();

JLabel jLabel5 = new JLabel();

JTextField jTextField1 = new JTextField();

JPasswordField jPasswordField1 = new JPasswordField();

JPasswordField jPasswordField2 = new JPasswordField();

JPasswordField jPasswordField3 = new JPasswordField();

JButton jButton1 = new JButton();

JButton jButton2 = new JButton();

JOptionPane jOptionPane1 = new JOptionPane();

dbConn conn = new dbConn();

public UserPasswordFrame(String level, String name) {

this.level = level;

this.name = name;

try {

jbInit();

} catch (Exception exception) {

exception.printStackTrace();

}

}

private void jbInit() throws Exception {

contentPane = (JPanel) getContentPane();

contentPane.setLayout(null);

setSize(new Dimension(444, 340));

setTitle("用户密码修改");

jLabel1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));

jLabel1.setText("用 户 密 码 修 改");

jLabel1.setBounds(new Rectangle(112, 15, 204, 24));

jLabel2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel2.setText("用户名:");

jLabel2.setBounds(new Rectangle(80, 72, 74, 23));

jLabel3.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel3.setText("旧密码:");

jLabel3.setBounds(new Rectangle(80, 115, 74, 23));

jLabel4.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel4.setText("新密码:");

jLabel4.setBounds(new Rectangle(80, 159, 74, 23));

jLabel5.setFont(new java.awt.Font("Dialog", Font.PLAIN, 18));

jLabel5.setText("确认新密码:");

jLabel5.setBounds(new Rectangle(72, 203, 101, 23));

jTextField1.setEnabled(false);

jTextField1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jTextField1.setBorder(BorderFactory.createLoweredBevelBorder());

jTextField1.setBounds(new Rectangle(175, 72, 159, 23));

jPasswordField1.setBorder(BorderFactory.createLoweredBevelBorder());

jPasswordField1.setBounds(new Rectangle(175, 114, 160, 25));

jPasswordField2.setBorder(BorderFactory.createLoweredBevelBorder());

jPasswordField2.setText("");

jPasswordField2.setBounds(new Rectangle(175, 158, 160, 25));

jPasswordField3.setBorder(BorderFactory.createLoweredBevelBorder());

jPasswordField3.setText("");

jPasswordField3.setBounds(new Rectangle(175, 202, 160, 25));

jButton1.setBounds(new Rectangle(87, 254, 86, 26));

jButton1.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jButton1.setBorder(BorderFactory.createRaisedBevelBorder());

jButton1.setText("提 交");

jButton1.addActionListener(new MAPasswordFrameFrame\_jButton1\_actionAdapter(this));

jButton2.setBounds(new Rectangle(253, 254, 86, 26));

jButton2.setFont(new java.awt.Font("Dialog", Font.PLAIN, 13));

jButton2.setBorder(BorderFactory.createRaisedBevelBorder());

jButton2.setText("退 出");

jButton2.addActionListener(new MAPasswordFrameFrame\_jButton2\_actionAdapter(this));

jOptionPane1.setBounds(new Rectangle(-8, 84, 262, 90));

jOptionPane1.setLayout(null);

contentPane.add(jLabel1);

contentPane.add(jLabel2);

contentPane.add(jLabel3);

contentPane.add(jLabel4);

contentPane.add(jLabel5);

contentPane.add(jPasswordField3);

contentPane.add(jPasswordField1);

contentPane.add(jPasswordField2);

contentPane.add(jButton2);

contentPane.add(jButton1);

contentPane.add(jTextField1);

contentPane.add(jOptionPane1);

this.jTextField1.setText(name);

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

jTextField1.setEnabled(true);

}

}

// 退出

public void jButton2\_actionPerformed(ActionEvent e) {

this.dispose();

}

// 修改密码

public void jButton1\_actionPerformed(ActionEvent e) {

boolean a = false;

boolean b = true;

// 异常判断

if (jTextField1.getText().trim() == null || jTextField1.getText().trim().length() == 0

|| jTextField1.getText().trim().length() > 20) {

jOptionPane1.showMessageDialog(this, "用户名不能为空且最大长度为20个字符！", "提 示", JOptionPane.INFORMATION\_MESSAGE, null);

} else if (jPasswordField2.getText().trim().compareTo(jPasswordField3.getText().trim()) != 0) {

jOptionPane1.showMessageDialog(this, "新密码确认错误！", "提 示", JOptionPane.INFORMATION\_MESSAGE, null);

} else {

try {

// 对原密码进行验证

ResultSet rs = conn.getRs("select userName,userPwd from tb\_user");

while (rs.next()) {

if (jTextField1.getText().trim().equals(rs.getString("userName").trim())

&& jPasswordField1.getText().trim().equals(rs.getString("userPwd").trim())) {

a = true;

b = false;

break;

}

}

if (b) {

jOptionPane1.showMessageDialog(this, "用户名或密码错误！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

}

if (a) {

// 修改密码

conn.getUpdate("update tb\_user set userPwd= '" + jPasswordField2.getText().trim()

+ "' where userName=('" + jTextField1.getText().trim() + "') ");

jOptionPane1.showMessageDialog(this, "恭喜您修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE, null);

jTextField1.setText("");

jPasswordField1.setText("");

jPasswordField2.setText("");

jPasswordField3.setText("");

}

rs.close();

} catch (java.sql.SQLException ce) {

System.out.println(ce);

}

}

}

}

class MAPasswordFrameFrame\_jButton1\_actionAdapter implements ActionListener {

private UserPasswordFrame adaptee;

MAPasswordFrameFrame\_jButton1\_actionAdapter(UserPasswordFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton1\_actionPerformed(e);

}

}

class MAPasswordFrameFrame\_jButton2\_actionAdapter implements ActionListener {

private UserPasswordFrame adaptee;

MAPasswordFrameFrame\_jButton2\_actionAdapter(UserPasswordFrame adaptee) {

this.adaptee = adaptee;

}

public void actionPerformed(ActionEvent e) {

adaptee.jButton2\_actionPerformed(e);

}

}

**database:**

create database stumanagerdb;

use stumanagerdb;

-- MySQL dump 10.13 Distrib 5.7.12, for Win64 (x86\_64)

--

-- Host: localhost Database: stumanagerdb

-- ------------------------------------------------------

-- Server version 5.7.16-log

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `tb\_class`

--

DROP TABLE IF EXISTS `tb\_class`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tb\_class` (

`classId` int(11) NOT NULL AUTO\_INCREMENT,

`departName` varchar(100) DEFAULT NULL,

`specName` varchar(100) DEFAULT NULL,

`className` int(11) NOT NULL,

PRIMARY KEY (`classId`)

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tb\_class`

--

LOCK TABLES `tb\_class` WRITE;

/\*!40000 ALTER TABLE `tb\_class` DISABLE KEYS \*/;

INSERT INTO `tb\_class` VALUES (1,'计算机学院','软件工程',1),(3,'计算机学院','计算机科学与技术',2),(5,'艺术设计学院','服装设计',1),(6,'自动化学院','物联网工程',2),(7,'自动化学院','物联网工程',1),(8,'计算机学院','计算机科学与技术',1),(9,'政法学院','法学',1),(10,'政法学院','法学',2),(11,'计算机学院','软件工程',2);

/\*!40000 ALTER TABLE `tb\_class` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tb\_cource`

--

DROP TABLE IF EXISTS `tb\_cource`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tb\_cource` (

`courceId` int(11) NOT NULL AUTO\_INCREMENT,

`courceName` varchar(100) NOT NULL,

`courceHour` float DEFAULT NULL,

`courceSpecName` varchar(100) DEFAULT NULL,

PRIMARY KEY (`courceId`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tb\_cource`

--

LOCK TABLES `tb\_cource` WRITE;

/\*!40000 ALTER TABLE `tb\_cource` DISABLE KEYS \*/;

INSERT INTO `tb\_cource` VALUES (1,'大学英语',2.5,'法学'),(2,'大学物理',1.5,'物联网工程'),(3,'高等数学',5.5,'计算机科学与技术'),(4,'C语言程序设计',3,'软件工程'),(5,'数据结构',3,'软件工程'),(6,'线性代数',1.5,'物理学'),(7,'大学语文',1.5,'服装设计'),(8,'服装市场考察',2.5,'服装设计');

/\*!40000 ALTER TABLE `tb\_cource` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tb\_depart`

--

DROP TABLE IF EXISTS `tb\_depart`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tb\_depart` (

`departId` int(11) NOT NULL AUTO\_INCREMENT,

`departName` varchar(100) NOT NULL,

PRIMARY KEY (`departId`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tb\_depart`

--

LOCK TABLES `tb\_depart` WRITE;

/\*!40000 ALTER TABLE `tb\_depart` DISABLE KEYS \*/;

INSERT INTO `tb\_depart` VALUES (1,'计算机学院'),(3,'艺术设计学院'),(4,'自动化学院'),(5,'政法学院'),(7,'物理学院');

/\*!40000 ALTER TABLE `tb\_depart` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tb\_score`

--

DROP TABLE IF EXISTS `tb\_score`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tb\_score` (

`scoreId` int(11) NOT NULL AUTO\_INCREMENT,

`stuNumber` int(11) NOT NULL,

`courceName` varchar(100) NOT NULL,

`score` float DEFAULT NULL,

PRIMARY KEY (`scoreId`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tb\_score`

--

LOCK TABLES `tb\_score` WRITE;

/\*!40000 ALTER TABLE `tb\_score` DISABLE KEYS \*/;

INSERT INTO `tb\_score` VALUES (1,3001,'C语言程序设计',95),(3,3002,'高等数学',59),(5,3004,'高等数学',96),(7,3001,'数据结构',97),(8,3006,'大学物理',100);

/\*!40000 ALTER TABLE `tb\_score` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tb\_spec`

--

DROP TABLE IF EXISTS `tb\_spec`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tb\_spec` (

`specId` int(11) NOT NULL AUTO\_INCREMENT,

`departName` varchar(100) DEFAULT NULL,

`specName` varchar(100) NOT NULL,

PRIMARY KEY (`specId`)

) ENGINE=InnoDB AUTO\_INCREMENT=15 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tb\_spec`

--

LOCK TABLES `tb\_spec` WRITE;

/\*!40000 ALTER TABLE `tb\_spec` DISABLE KEYS \*/;

INSERT INTO `tb\_spec` VALUES (1,'计算机学院','计算机科学与技术'),(3,'艺术设计学院','服装设计'),(5,'计算机学院','软件工程'),(6,'政法学院','法学'),(7,'自动化学院','物联网工程'),(14,'计算机学院','网络工程');

/\*!40000 ALTER TABLE `tb\_spec` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tb\_student`

--

DROP TABLE IF EXISTS `tb\_student`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tb\_student` (

`stuId` int(11) NOT NULL AUTO\_INCREMENT,

`stuNumber` int(11) NOT NULL,

`stuName` varchar(20) NOT NULL,

`stuDepart` varchar(100) DEFAULT NULL,

`stuSpec` varchar(100) DEFAULT NULL,

`stuClass` int(11) DEFAULT NULL,

PRIMARY KEY (`stuId`),

UNIQUE KEY `stuNumber\_UNIQUE` (`stuNumber`)

) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tb\_student`

--

LOCK TABLES `tb\_student` WRITE;

/\*!40000 ALTER TABLE `tb\_student` DISABLE KEYS \*/;

INSERT INTO `tb\_student` VALUES (1,3001,'黄章','计算机学院','软件工程',1),(4,3003,'马云','计算机学院','计算机科学与技术',2),(6,3002,'马化腾','计算机学院','软件工程',1),(7,3004,'周鸿祎','自动化学院','物联网工程',2),(8,3005,'雷军','自动化学院','物联网工程',2),(9,3006,'李彦宏','物理学院','物理学',1),(10,3007,'刘强东','政法学院','法学',1),(11,3008,'丁磊','物理学院','物理学',1),(13,3009,'张朝阳','艺术设计学院','服装设计',1);

/\*!40000 ALTER TABLE `tb\_student` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `tb\_user`

--

DROP TABLE IF EXISTS `tb\_user`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tb\_user` (

`userId` int(11) NOT NULL AUTO\_INCREMENT,

`userName` varchar(45) NOT NULL,

`userPwd` varchar(45) DEFAULT NULL,

`userType` int(11) NOT NULL,

PRIMARY KEY (`userId`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `tb\_user`

--

LOCK TABLES `tb\_user` WRITE;

/\*!40000 ALTER TABLE `tb\_user` DISABLE KEYS \*/;

INSERT INTO `tb\_user` VALUES (1,'admin','',1),(4,'test','bbb',2),(5,'abc','123',2);

/\*!40000 ALTER TABLE `tb\_user` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;