

四川轻化工大学

# 课程设计书

科目 数据库系统课程设计

学院 计算机科学与工程学院

专业 软件工程

班级 2019 级 6 班

题目 线上考试系统

教师 梁 兴 建

学生 郑小菁 杨佳佳 杨顺利

# 在线考试系统设计

## 摘要

包含内容写清楚用什么构建数据库，用什么平台编写代码，系统包含哪些功能，哪些功能已经实现，系统设计的流程什么，文档主要叙述了什么

本系统主要基于 Java 编程语言，以 Eclipse 为开发工具进行开发，利用 Mysql 作为后台的数据库，利用 WINDOWS 10 作为系统平台进行整个系统的开发。本系统包含用户注册及登录功能、用户个人中心、题库管理、试卷管理、考试安排等功能。已经实现用户注册及登录功能、用户个人中心、题库管理、试卷管理、考试安排这几个功能模块。

本系统设计的流程为：首先根据需求分析，建立数据库，再通过 Java Swing 进行界面以及功能的实现。

文档主要叙述了本项目软件开发的方法，可以为本项目的相关专题计划制定提供指导和参考，供项目组全体成员阅读。

### 应用程序分工情况：

完成人	完成内容	工作量比例
郑小菁	题库信息、试卷信息、班级信息	50%
杨顺利	注册登录、个人中心	35%
杨佳佳	考试信息	15%

## 目 录

<b>1、项目开发目的与目标.....</b>	<b>- 1 -</b>
1.1 设计目的.....	- 1 -
1.2 设计目标.....	- 1 -
<b>2、需求分析.....</b>	<b>- 1 -</b>
2.1 主要功能简介.....	- 1 -
2.2 功能模块分析.....	- 2 -
<b>3、概念结构设计.....</b>	<b>- 4 -</b>
3.1 数据实体及相关的联系分析.....	- 4 -
3.2 E-R 模型设计.....	- 5 -
<b>4、逻辑结构设计.....</b>	<b>- 6 -</b>
4.1 关系模式建立.....	- 7 -
4.2 数据表结构.....	- 7 -
<b>5、应用系统设计与实现.....</b>	<b>- 12 -</b>
5.1 数据库操作类的设计.....	- 12 -
5.2 题库功能模块实现.....	- 13 -
5.3 试卷功能模块实现.....	- 15 -
5.4 考试功能模块实现.....	- 18 -
5.5 班级信息功能模块实现.....	- 18 -
5.6 登录注册功能模块实现.....	- 18 -
5.7 个人中心功能模块实现.....	- 18 -
<b>6、设计总结.....</b>	<b>- 31 -</b>
<b>7、设计体会.....</b>	<b>- 32 -</b>
<b>附录 A、操作手册.....</b>	<b>- 33 -</b>

# 新闻信息系统设计

## 1、项目开发目的与目标

### 1.1 设计目的

考试可以反映出学生的学习效果和教师教学的成就和不足。如果充分发挥考试的功能，则能促进教学的发展。随着我国教育信息化系统的建设，计算机和网络辅助教学和考试作为一种新的手段，已显示出它本身独特的优势，充分利用现有的计算机和网络资源开发出一套具有在线考试及批阅功能的系统，具有不可低估的作用。

### 1.2 设计目标

在线考试系统是由本小组开发的在线考试平台。目标是为了方便考生们考试，教师阅卷，管理员管理而开发的系统。该考试系统包括了考生考试，用户管理，考试管理，试卷管理，题库管理等多个功能。定位于节省考试过程中的成本，方便管理员对考试的管理。

## 2、需求分析

### 2.1 主要功能简介

在网络技术飞速发展的今天，互联网已进入千家万户，社会进入网络时代，计算机网络已经成为社会发展的强大动力。网络不仅给我们带来的是无穷的信息，也带来了更多便利。不仅企业、政府的正常工作离不开网络，教育事业同样需要网络。教育的网络化代表了教育的一个重要的方面，并且已经开始成为现代教育的一个主要特征，对教育的发展形成新的推动力。随着 Internet 的迅速发展和广泛普及，建立在其上的远程教育成为现代教育技术未来发展方向之一，在线考试作为远程教育的一个子系统也成为重要的研究领域。

本系统主要由后台用户管理和前台学生、教师管理两部分组成，其中后台用户管理包含学生管理、教师管理等相关模块，前台功能里的教师管理包含题库管理、试卷管理、考试安排、用户注册及登录、教师个人中心等相关功能；前台功能里的学生管理包含用户注册及登录、我的考试、学生个人中心等相关功能。本系统的功能模块如图 2-1 所示。

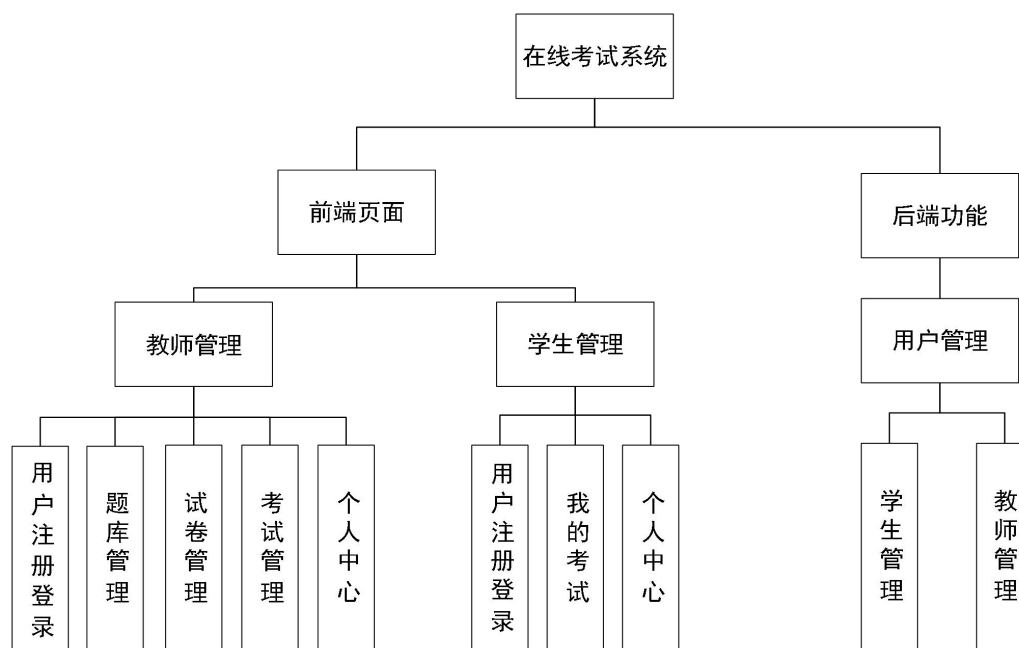


图 2-1 系统功能模块

## 2.2 功能模块分析

### 1)后台管理功能分析

a)教师用户管理：包含显示教师信息，修改教师信息和调整激活状态三个子功能。

显示教师信息，显示全部已注册教师的信息（包含激活状态，工号，姓名，年龄，已绑定的手机号，已发布的考试），并可以通过工号和已发布的考试进行搜索。修改教师信息,可以修改教师已绑定手机号，可以进行密码重置。

调整激活状态，若要禁止该教师使用此考试系统，通过调整教师的激活状态实现。

b)学生用户管理：包含显示学生信息，修改学生信息和调整激活状态三个子功能。

显示学生信息：显示全部已注册学生的信息（包含激活状态，学号，姓名，年龄，班级，已绑定的手机号，已参加的考试），并可以通过学号和班级进行搜索。

修改学生信息：可以修改学生已绑定手机号，可以进行密码重置。

调整激活状态：若要禁止该学生使用此考试系统，通过调整学生的激活状态实现。

### 2)前台管理功能分析

a)前台教师管理模块：主要包含注册/登录，题库管理，考试管理，试卷管理，个人中心几个版块。

注册：教师进入登录界面点击注册按钮，选择学校和身份，输入工号，然后设置密码，并通过手机号发送验证码进行注册；

登录：允许用户输入账号（或手机号）和密码并进行验证，若账号或密码错误则发

出提示，若忘记密码则可通过手机号发送验证码找回密码。

**题库管理：**包含新建题库，编辑题库，删除题库和搜索题库四个子功能。

**新建题库：**点击新建题库弹出新建界面，输入题库名称，若已存在该题库，则在名称下给出提示“该题库已存在”，若确认创建题库则点击“导入题目”进入编辑界面，若不想导入则点击“创建并退出”，若想改名则点击“上一步”；

**编辑题库：**可以修改题库名称，也可以向题库中导入题目，导入题目可以通过文件导入和手动添加实现；

**删除题库：**点击删除，若其他试卷引用该题库里的试题则给出提示，若继续删除则点击确认删除，否则取消；

**搜索题库：**在搜索框输入题库名称模糊搜索相关题库。并附带删除，编辑，浏览等功能。

**考试管理：**包含发布考试，查看考试，参考人员管理，批改试卷，统计分析，试卷选择六个子功能。

**发布考试：**可以命名考试，添加参考人员，选择试卷，设置开考时间和考试时长；  
**查看考试：**在搜索框输入考试名称对相关考试进行模糊检索，在搜索结果中显示每一个考试的状态比如“已发布”，“未发布”，“已完成”。若已完成则可查看该堂考试的统计分析结果和参考学生的详细成绩；  
**参考人员管理：**可以选择参考班级，并可以对学参考情况进行编辑；  
**批改试卷：**可以选择手动阅卷或自动阅卷，将考生的答案与题库中题目的答案进行匹配；  
**统计分析：**统计出考生的最高、最低分以及平均分布情况；  
**试卷选择：**教师选择用于本次考试的试卷。

**试卷管理：**包含创建试卷，编辑试卷两个子功能。

**创建试卷：**可以命名试卷，设置命题人，试卷总分以及考试时长，在题库中自己的需求引入试题，选择需要引用的题库，选择题型和设置限定条件（题目数量，题目分值），设置完毕后在其后显示总分；  
**编辑试卷：**可以对试卷进行重命名，试题更改，试卷删除操作。

**个人中心：**包含个人信息编辑，账号设置，退出登录，注销账户，关于我们。

**个人信息编辑：**可以在其中修改头像，昵称，性别；  
**账号设置：**可以修改密码和变更手机号；  
**退出登录：**退出当前账号进入登录界面；  
**注销账户；**关于我们：可以看到系统版本，选择更新以及我们的团队信息。

**b)前台学生管理模块：**主要包含注册/登录，我的考试，个人中心几个版块。

**注册：**学生进入登录界面点击注册按钮，选择学校和身份，输入学生号，然后设置

密码，并通过手机号发送验证码进行注册；

登录功能：允许用户输入账号（或手机号）和密码并进行验证，若账号或密码错误则发出提示，若忘记密码则可通过手机号发送验证码找回密码。

我的考试：包含未考，进入答题，已考三个子功能。

未考：学生可以在此界面下查看教师已发布但还未开始的考试；

进入答题：通过点击对应考试开始考试。在考试期间可以答题，未作答题目高亮显示；查看所有的题目编号并可通过点击题目编号进入对应题目；确认答题无误后可点击交卷按钮，若考生还有未作答的题目则发出提示-“你还有未作答题目，是否确认交卷？”若考生不交卷则返回考试页面显示所有题目编号，若考生点击确认则交卷；若考生作答完所有题目则请考生确认是否继续交卷，考生确认则交卷，反之返回答题页面。距离考试结束还有 15 分钟时弹出提示框提醒考生“距离考试结束还有十五分钟，请注意考试时间！”，若考试结束考生依旧没有交卷则系统自动交卷；

已考：可以查看自己已经参考过的考试，考试题目（包含题目对错）以及得分情况。

个人中心：包含个人信息编辑，账号设置，退出登录，注销账户，关于我们。

个人信息编辑：可以在其中修改头像，昵称，性别；账号设置：可以修改密码和变更手机号；退出登录：退出当前账号进入登录界面；注销账户；关于我们：可以看到系统版本，选择更新以及我们的团队信息。

### 3、概念结构设计

#### 3.1 数据实体及相关的联系分析

根据需求分析，本系统主要包含实体如下：

- 1)管理员：管理员 ID、姓名、密码、手机号
- 2)题库：题库名称、题库 ID
- 3)题目：题目 ID、题干、答案、分数
- 4)教师：教师 ID、教师工号、姓名、性别、手机号、密码、激活状态
- 5)学生：学生 ID、学号、姓名、性别、手机号、密码、激活状态
- 6)班级信息：班级 ID、班级名称
- 7)课程：课程 ID、课程号、课程名称
- 8)试卷：试卷 ID、试卷名称、启用状态、考试时长、总分
- 9)考试信息：考试 ID、考试名称、学生成绩

本系统实体之间的联系如下：

- 1)一个管理员对应多条班级信息，一个班级信息可以由多个管理员管理（m:n）
- 2)一个教师对应多条班级信息，一个班级信息可以由多个管理员管理（1:n）
- 3) 一个教师可以创建多个题库，一个题库只能由一个教师创建（1:n）
- 4)一个题库可以组合出多份试卷，一份试卷只能来源于一个题库（1:n）
- 5)一个课程对应多个题库，一个题库只针对一门课程（1:n）
- 6)一份试卷对应多个考试信息，一个考试对应一份试卷（1:n）
- 7)一个学生对应多个考试信息，一个考试信息对应多个学生（m:n）
- 8)一个学生对应多个班级信息，一个班级信息只针对一个学生（1:n）
- 9)一个题库可以包含多个题目，一个题目只来自一个题库（1:n）

### 3.2 E-R 模型设计

概念模型是现实世界到信息世界的第一层抽象，确定实体属性关系等。概念模型其中的一种模型是实体—联系模型即 E-R 模型，它是用 E-R 图来描述现实世界的概念模型。

E-R 图中使用矩形表示实体，使用菱形表示联系。E-R 图显示了要建立的模型的数据内部及数据之间的相互关系。本系统采用 E-R 模型进行概念结构设计，学生实体图如图 3-1 所示，教师实体图如图 3-2 所示，试卷实体图如图 3-3 所示，考试实体图如图 3-4 所示，管理员实体图如图 3-5 所示，课程实体图如图 3-6 所示，班级实体图如图 3-7 所示，题库实体图如图 3-8 所示，题目实体图如图 3-9 所示，合并后的 E-R 图如 3-10 所示。

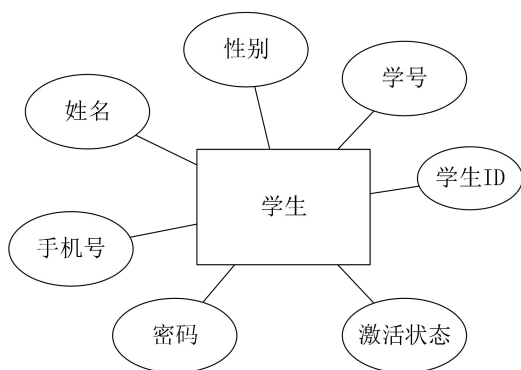


图 3-1 学生实体图

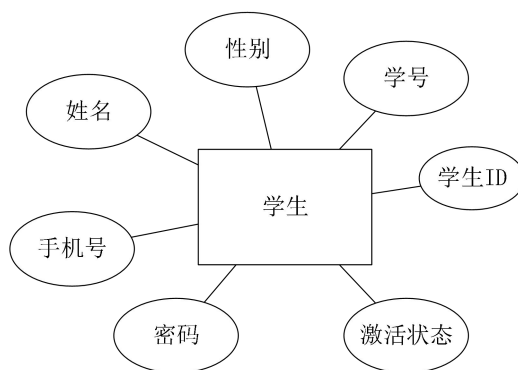


图 3-2 教师实体图



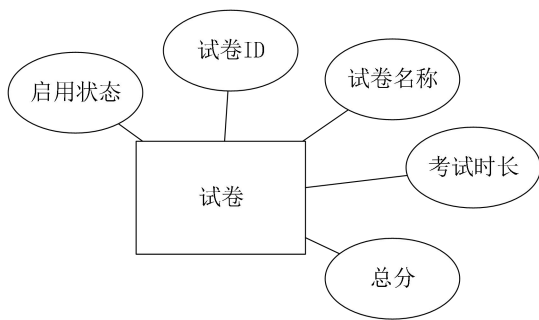


图 3-3 试卷实体图

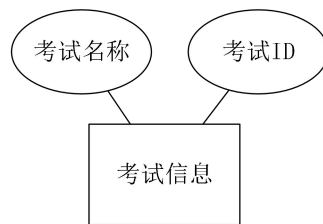


图 3-4 考试信息实体图

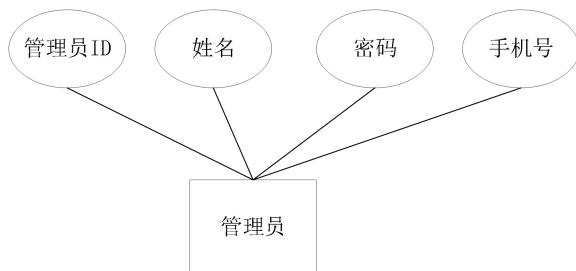


图 3-5 管理员实体图

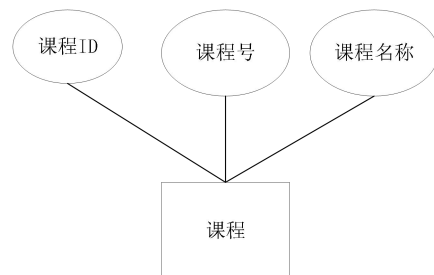


图 3-6 课程实体图

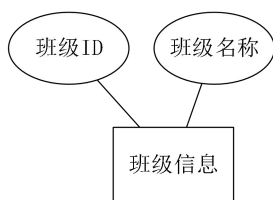


图 3-7 班级实体图

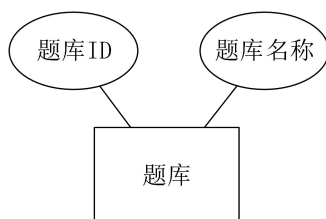


图 3-8 题库实体图

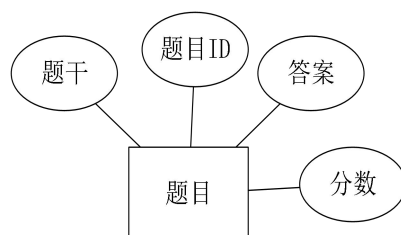


图 3-9 题目实体图

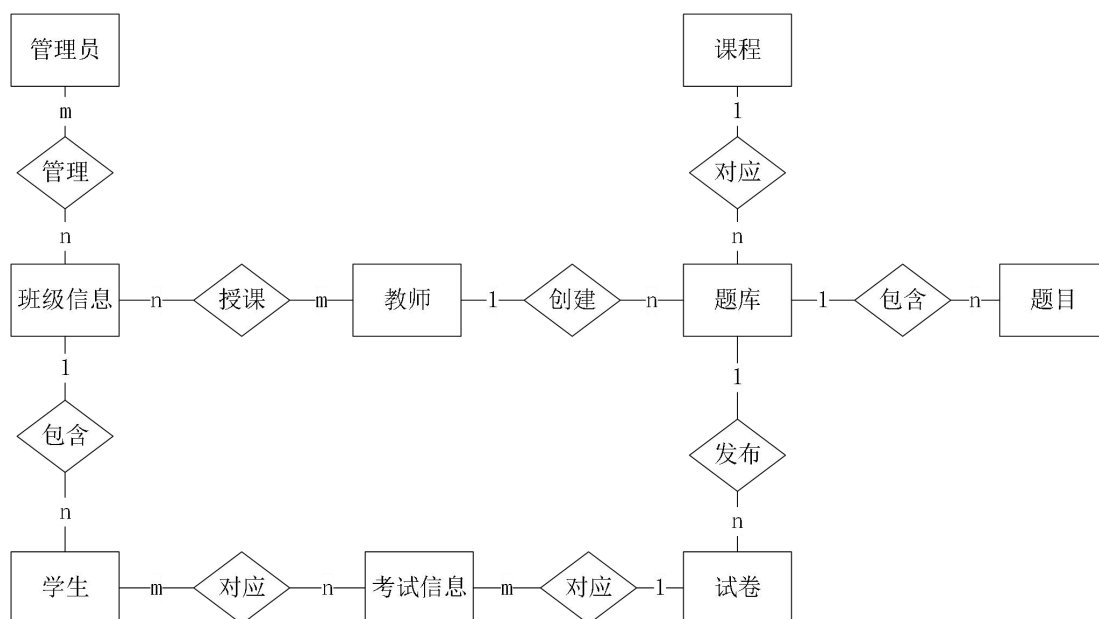


图 3-10 系统总体 E-R 图

## 4、逻辑结构设计

### 4.1 关系模式建立

概念模型已经确定了实体以及属性联系等，逻辑结构设计主要是针对 E-R 模型转化为关系模式，根据转化规则，逻辑模型设计结果如下，其中划线的属性为主码。

- 1) 学生信息表(学生 ID、学号、姓名、性别、手机号、密码、激活状态、班级 ID)
- 2) 教师信息表(教师 ID、教师工号、姓名、性别、手机号、密码、激活状态)
- 3) 试卷信息表(试卷 ID、试卷名称、总分、考试时长、启用状态、题库 ID)
- 4) 考试信息表(考试 ID、考试名称、试卷 ID)
- 5) 管理员信息表(管理员 ID、管理员账号、姓名、性别、密码、手机号)
- 6) 课程信息表(课程 ID、课程号、课程名称)
- 7) 班级信息表(班级 ID、班级名称)
- 8) 题库信息表(题库 ID、题库名称、教师 ID、课程 ID)
- 9) 题目信息表(题目 ID、题干、答案、分数、题库 ID)
- 10) 管理员\_班级信息关联表(管理员\_班级信息关联 ID、管理员 ID、班级 ID)
- 11) 学生\_考试信息关联表(学生\_考试信息关联 ID、学生 ID、学生成绩、考试 ID)
- 12) 教师\_班级信息关联表(班级信息\_教师关联 ID、班级 ID、教师 ID)

### 4.2 数据表结构

通过关系的完整性分析，本系统可以分为：学生信息表、教师信息表、试卷信息

表、考试信息表、管理员信息表、课程信息表、班级信息表、题库信息表、题目信息表、管理员\_班级信息关联表、学生\_考试信息关联表、教师\_班级信息关联表。

学生信息表（Stu\_Info）：用于记录学生的注册信息和个人信息，添加管理员 ID 为外键，激活状态默认为未激活，密码默认 123456，主要数据结构如表 4-1 所示：

表 4-1 学生信息表（Stu\_Info）

字段名	类型	长度	约束条件	备 注
StuID	int		主键，自增	学生 ID
Sno	char	11		学号
SName	varchar	12	非空	姓名
SSex	char	1	非空	性别
SPWD	varchar	16	非空，默认 123456	密码
IsActive	tinyint	1	非空，默认 0	是否激活（0：未激活，1：已激活）
SPhone	int	11	非空	手机号
ClassID	int			班级 ID

教师信息表（Teacher\_Info）：用于记录教师的注册信息和个人信息，添加管理员 ID 为外键，激活状态默认为未激活，密码默认 123456，主要数据结构如表 4-2 所示：

表 4-2 教师信息表（Teacher\_Info）

字段名	类型	长度	约束条件	备 注
TeacherID	int	11	主键，自增	教师 ID
Tno	char	11		教师工号
TeacherName	varchar	12	非空	姓名
TeacherSex	char	1	非空	性别
TeacherPhone	int	11	非空	手机号
IsActive	tinyint	1	非空，默认 0	是否激活（0：未激活，1：已激活）
TeacherPWD	varchar	16	非空，默认 123456	密码

试卷信息表（TestPaper\_Info）：用于记录试卷信息，试卷名称必须唯一，添加教师 ID 和测试 ID 作为外键，试卷启用状态默认为未启用，主要数据结构如表 4-3 所示：

表 4-3 试卷信息表 (TestPaper\_Info)

字段名	类型	长度	约束条件	备 注
TPID	int	11	自增，主键	试卷 ID
TPName	varchar	20	唯一，非空	试卷名称
TestTime	int	11	非空	考试时长
TotalScore	int	11	非空	总分
IsActive	tinyint	1	非空，默认 0	是否启用（0：未启用，1：已启用）
QBID	int		外键	题库 ID

考试信息表 (Question\_Info)：用于记录每一次测试使用的是哪一份试卷，需要增加试卷 ID 作为外键，主要数据结构如表 4-4 所示：

表 4-4 考试信息表 (Question\_Info)

字段名	类型	长度	约束条件	备 注
TestID	int	11	自增，主键	测试 ID
TestName	vachar	15		考试名称
TestPaperID	int	11	外键	试卷 ID

管理员信息表 (Admin\_Info)：用于记录管理员信息，密码默认 123456，主要数据结构如表 4-5 所示：

表 4-5 管理员信息表 (Admin\_Info)

字段名	类型	长度	约束条件	备 注
AdminID	int	11	主键，自增	管理员 ID
AdminNo	vachar	11	唯一，非空	管理员账号
AdminName	varchar	12	非空	姓名
AdminSex	char	1	非空	性别
AdminPWD	varchar	16	非空，默认 123456	密码
AdminPhone	int	11	非空	手机号

课程信息表 (Course\_Info)：用于记录课程信息，课程名称必须唯一，主要数据结构如表 4-6 所示：

表 4-6 课程信息表 (Course\_Info)

字段名	类型	长度	约束条件	备 注
CourseID	int	11	自增, 主键	课程 ID
Cno	char	11		课程号
CourseName	varchar	20	唯一, 非空	课程名称

班级信息表 (Class\_Info)：用于记录班级信息，添加教师 ID 和学生 ID 作为外键，主要数据结构如表 4-7 所示：

表 4-7 班级信息表 (Class\_Info)

字段名	类型	长度	约束条件	备 注
ClassID	int	11	自增, 主键	班级 ID
ClassName	varchar	15		班级名称

题库信息表 (QuestionBank\_Info)：用于记录是哪个老师创建的题库，记录该题库是针对哪个课程，需要添加教师 ID 和课程 ID 作为外键，题库名必须唯一，主要数据结构如表 4-8 所示：

表 4-8 题库信息表 (QuestionBank\_Info)

字段名	类型	长度	约束条件	备 注
QbID	int	11	自增, 主键	题库 ID
QbName	varchar	20	唯一, 非空	题库名称
TeacherID	int	11	外键	教师 ID
CourseID	int	11	外键	课程 ID

题目信息表 (Problem\_Info)：用于记录每个题库的题目信息，添加题库 ID 作为外键，主要数据结构如表 4-9 所示：

表 4-9 题目信息表 (Problem\_Info)

字段名	类型	长度	约束条件	备 注
ProblemID	int	11	自增, 主键	题目 ID
problem	varchar	128	非空	题干
answer	varchar	128	非空	答案
score	int	2	非空	分数
QbID	int		外键	题库 ID

管理员\_班级信息关联表（Admin\_Class\_Relation）：用于管理员和班级信息之间的联系，主要数据结构如表 4-10 所示：

表 4-10 管理员\_班级信息关联表（Admin\_Class\_Relation）

字段名	类型	长度	约束条件	备 注
Admin_Class ID	int	11	自增，主键	管理员_班级关联 ID
AdminID	int	11	外键	管理员 ID
ClassID	int	11	外键	班级 ID

学生\_考试信息关联表（Stu\_Test\_Relation）：用于学生和考试之间的联系，主要数据结构如表 4-11 所示：

表 4-11 学生\_考试信息关联表（Stu\_Test\_Relation）

字段名	类型	长度	约束条件	备 注
Stu_Test ID	int	11	自增，主键	学生_考试信息关联 ID
StuID	int	11	外键	学生 ID
TestID	int	11	外键	考试 ID
Score	int			考试成绩

教师\_班级信息关联表（Class\_Teacher\_Relation）：用于存储教师和班级之间的联系，主要数据结构如表 4-12 所示：

表 4-12 教师\_班级信息关联表（Class\_Teacher\_Relation）

字段名	类型	长度	约束条件	备 注
Class_Teacher ID	int	11	自增，主键	班级信息_教师关联 ID
TeacherID	int	11	外键	教师 ID
ClassID	int	11	外键	班级 ID

## 5、应用系统设计与实现

### 5.1 数据库操作类的设计

#### 1、数据库连接的方法

```
public static Connection conection = null;
// Java-JDBC 操作-数据库访问
private String user = "root"; //数据库用户名
private String password = "7802667cxdbc"; //数据库密码
private String url = "jdbc:mysql://localhost:3306/net_exam"; //访问数据库的地址
private String driver = "com.mysql.cj.jdbc.Driver";

public CommonaJdbc(){
    getCon();
    try {
        Class.forName(driver); //加载驱动
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}

public Connection getCon(){
    try {
        conection = DriverManager.getConnection(url, user, password);
    } catch (SQLException e){
        e.printStackTrace();
    }

    return conection;
}
```

#### 2、插入、修改、删除记录的方法（也可以分成三个方法）

##### (1) 插入与修改记录的方法

```
public boolean InsertOrUpdateObject_bank(Obj_gradeinfo objgradeinfo,String
```

```

eventname)          //题目的插入与修改
    public boolean InsertOrUpdateObject_paper(Obj_testpaper objgradeinfo,String
eventname)          //试卷的插入与修改
    public boolean InsertOrUpdate_Obj_gradeinfo_sub(Obj_gradeinfo_sub object)
        //学生成绩的插入与修改
(2) 删除记录的方法
    public boolean DeleteObject(String deleteSql)
3、查询提取多条数据（一个记录集）的方法
    public void buildTable()

```

## 5.2 题库功能模块实现

教师登录以后，可以通过上方的菜单栏中的“题库信息”，并选择相应的题库，进入题库窗口，该窗口展示了该题库里的所有题目信息，教师可选定对应题目，对题目的内容、答案、分数进行修改，以及对题目进行删除、添加功能。该页面的设计效果如图 5-1 所示。



图 5-1 题库信息界面

书籍列表实现思路：点击书籍管理以后，请求得到数据库的所有书籍信息并在书籍



管理界面显示。在书籍管理界面管理员用户可以根据可视化的按钮提示实现书籍管理操作如分类查看、修改、更新、删除等。具体实现过程如下：

### 1、题库信息查看功能的实现：

(1) 在 buildTable() 方法下定义并编写查询该题库下所有题目的 SQL 语句，SQL 语句如下：

```
String sqlStr = "SELECT problemid,problem,answer,score FROM problem_info,  
questionbank_info "+ "WHERE questionbank_info.Obid=problem_info.obID AND  
Obname='"+ eventname + "'";
```

(2) 提取调用connection对象的execQuery(String)方法，并将查询的结果一一加入到DefaultTableModel表格模型中，代码如下：

```
rs = connection.prepareStatement(sqlStr).executeQuery();  
while (rs.next()) {  
    Vector vdata = new Vector();  
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {  
        vdata.addElement(rs.getObject(i));  
        tableModel.addRow(vdata); // 将集合添加到表格模型中  
    }  
}
```

### 2、题目信息修改功能的实现：

(1) 创建一个题库实体 Obj\_bankinfo，来存储修改的题目的信息，代码如下：

```
appstu.model.Obj_bankinfo objbankinfo = new appstu.model.Obj_bankinfo();
```

(2) 首先通过validateID(String id, String tname, String idvalue)方法查询该题目编号是否存在，若存在，则执行修改的题目SQL语句，代码如下：

```
if (validateID("problemid", "problem_info", objbankinfo.getProblemid().trim())) {  
    sqlStatement = "Update problem_info set problemid = " +  
    objbankinfo.getProblemid() + ",problem = " + objbankinfo.getProblem()  
    + ",answer=" + objbankinfo.getAnswer() + ",score=" +  
    objbankinfo.getScore() + " where problemid = " +  
    objbankinfo.getProblemid().trim() +;  
}
```

### 3、题目的增添功能的实现：

(1) 创建一个题库实体 Obj\_bankinfo，来存储增添的题目的信息，代码如下：

```
appstu.model.Obj_bankinfo objbankinfo = new appstu.model.Obj_bankinfo();
```

(2) 首先通过validateID(String id, String tname, String idvalue)方法查询该题目编号是否存在，若不存在，则执行增加题目的SQL语句，代码如下：

```
if (validateID("problemid", "problem_info", objbankinfo.getProblemid().trim())){  
    修改的SQL语句}  
else{
```

```

        sqlStatement = "Insert into problem_info(problemid,problem,answer,score,obid)
        values (" + objgradeinfo.getProblemid() + "," + objgradeinfo.getProblem() + ","
        + objgradeinfo.getAnswer()+"," + objgradeinfo.getScore()+ "," + id + ")";
    }

```

#### 4、题目的删除功能的实现：

(1) 读取题目编号文本框中的内容，获得该题目的编号，根据编号编写删除SQL语句，SQL语句如下：

```

String sqlDel = "delete from problem_info where problemid = '" +
jTextField0.getText().trim() + "'";

```

(2) 并通过删除记录方法DeleteObject（），将该题目删除，代码如下：

```

public boolean DeleteObject(String deleteSql) {
    infoStr = "删除";
    return AdapterObject(deleteSql);
}

```

### 5.3 试卷功能模块实现

教师登录以后，可以通过上方的菜单栏点击“试卷信息”，并选择相应的试卷，进入试卷窗口，该窗口展示了该试卷里的所有相关信息，教师可通过下方的文本框，对试卷的名称、考试时长、总分、以及题库的选择进行修改，也可进行添加试卷以及删除试卷操作。该页面的设计效果如图 5-2 所示。

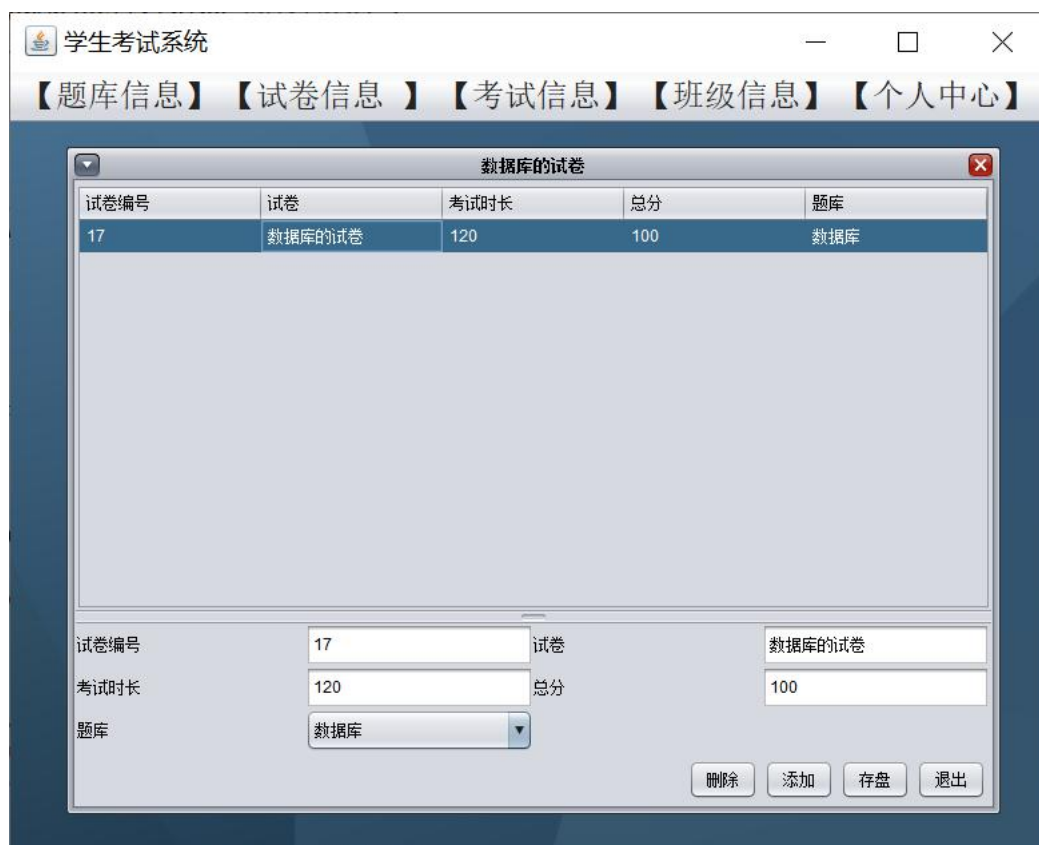


图 5-2 试卷信息界面

### 1、试卷信息查看功能的实现：

(1) 在 buildTable () 方法下定义并编写查询该试卷信息的 SQL 语句， SQL 语句如下：

```
String sqlStr = "SELECT tpid,tpname,testtme,totalscore,Obname FROM
testpaper_info,questionbank_info "
+ "WHERE testpaper_info.qbid=questionbank_info.obid and
tpname='" + eventname + "'";
```

(2) 提取调用 connection 对象的 execQuery(String)方法，并将查询到试卷信息的结果加入到 DefaultTableModel 表格模型中，代码如下：

```
rs = connection.prepareStatement(sqlStr).executeQuery();
while (rs.next()) {
    Vector vdata = new Vector();
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {
        vdata.addElement(rs.getObject(i));
    }
    tableModel.addRow(vdata); // 将集合添加到表格模型中
}
```

### 2、试卷信息修改功能的实现：

(1) 创建一个试卷实体 Obj\_testpaper，来存储修改的题目的信息，代码如下：

```
appstu.model.Obj_testpaper objgradeinfo = new appstu.model.Obj_testpaper();
```

(2) 首先通过 validateID(String id, String tname, String idvalue)方法查询该试卷编号是否存在, 若存在, 则执行修改试卷的 SQL 语句, 代码如下:

```
if (validateID("tpid", "testpaper_info", objgradeinfo.gettpid().trim())) {  
    sqlStatement = "Update testpaper_info set tpid = " + objgradeinfo.gettpid()  
    + ",tpname=" + objgradeinfo.gettpname() + ",testtme=" +  
    objgradeinfo.gettesttme() + ",totalscore=" + objgradeinfo.gettotalscore()+  
    ",qbid=" + id  
    + " where tpid = " + objgradeinfo.gettpid().trim() + """;  
}
```

### 3、增添试卷功能的实现:

(1) 创建一个题库实体 Obj\_bankinfo, 来存储增添的题目的信息, 代码如下:

```
appstu.model.Obj_bankinfo objbankinfo = new appstu.model.Obj_bankinfo();
```

(2) 首先通过validateID(String id, String tname, String idvalue)方法查询该试卷编号是否存在, 若不存在, 则执行添加试卷SQL语句, 代码如下:

```
if (validateID("problemid", "problem_info", objbankinfo.getProblemid().trim())){  
    修改的SQL语句}  
else {  
    sqlStatement = "Insert into  
    testpaper_info(tpid,tpname,testtme,totalscore,qbid) values (" +  
    objgradeinfo.gettpid() + "," + objgradeinfo.gettpname() + "," +  
    objgradeinfo.gettesttme()+"," + objgradeinfo.gettotalscore()+"," + id+  
    ")";  
}
```

(3) 同时新创建一个JMenuItem, 将新添加的题目加入到“试卷信息”的菜单项中, 已到达刷新的效果, 代码如下:

```
JMenuItem newtest = new JMenuItem(objgradeinfo.gettpname());  
设置菜单项  
AppMain._jMenu[1].add(newtest);
```

### 4、试卷的删除功能的实现:

(1) 读取试卷编号文本框中的内容, 获得该试卷的编号, 根据编号编写删除SQL语句, SQL语句如下:

```
String sqlDel = "delete from testpaper_info where tpid = " +  
jTextField0.getText().trim() + "";
```

(2) 并通过删除记录方法DeleteObject(), 将该试卷删除, 代码如下:

```
public boolean DeleteObject(String deleteSql) {  
    infoStr = "删除";  
    return AdapterObject(deleteSql);  
}
```

(3) 将“试卷信息”下的所有菜单项全部删除, 再重新导入试卷, 以达到刷新的功能, 代码如下:

```

AppMain._jMenu[1].removeAll();
String sqlSelect = "select tpname from testpaper_info";
appstu.util.RetrieveObject retrieve2 = new appstu.util.RetrieveObject();
vdata2 = retrieve2.getObjectRow(sqlSelect);
JMenuItem[] _jMenuItem1 = new JMenuItem[vdata2.size()];
for(int j = 0,num=vdata2.size(); j < vdata2.size();j++) {
    AppMain._jMenu[1].add(_jMenuItem1[j]);
}

```

## 5.4 考试信息功能模块实现

教师登录以后，可以通过上方的菜单栏点击“考试信息”，进入考试信息窗口，选择班级并选择对应的考试，该窗口展示了所有班级里学生考试的信息，教师可通过下方的文本框，对选中的考生的成绩进行修改，也可添加考生的考试信息，以及删除考生的考试信息的操作。该页面的设计效果如图 5-3 所示。

学号	姓名	性别	班级	考试科目	成绩
1	李 小	男	软件191	数据结构综合测试	90
19102	张 三	女	软件191	数据结构综合测试	88
19103	杨 阳	男	软件191	数据结构综合测试	90
19105	王 川	男	软件191	数据结构综合测试	30
19106	李 四	女	软件191	数据结构综合测试	100

图 5-3 考试信息界面

### 1、班级考试信息的查询

(1) 首先获取两个下拉列表当前的信息代码如下：

```

String cid = classid[jComboBox2.getSelectedIndex()];
String tid = examkindid[jComboBox1.getSelectedIndex()];

```

(2) 根据获得到的班级和考试编写查询班级信息的SQL语句，代码如下：

```

String sqlStr = "SELECT stu_info.sno,sname,ssex,classname,testname,score from
stu_info,class_info,question_info,stu_test_relation where " +

```

```
"class_info.classid="+cid+" and stu_info.classid=class_info.classid and
stu_test_relation.testid="+tid+" and "+"stu_info.stuid=stu_test_relation.stuid and
question_info.testid=stu_test_relation.testid";
```

(3) 提取调用connection对象的execQuery(String)方法，并将查询的结果一一加入到DefaultTableModel表格模型中，代码如下：

```
rs = connection.prepareStatement(sqlStr).executeQuery();
while (rs.next()) {
    Vector vdata = new Vector();
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {
        vdata.addElement(rs.getObject(i));
    }
    tableModel.addRow(vdata); // 将集合添加到表格模型中
}
```

## 2、修改学生成绩功能的实现：

(1) 创建一个学生考试信息实体 Obj\_gradeinfo\_sub，来存储修改的学生成绩的信息，代码如下：

```
appstu.model.Obj_gradeinfo_sub objgradeinfo = new
appstu.model.Obj_gradeinfo_sub();
```

(2) 首先通过 validateID(String id, String tname, String idvalue)方法查询该学生成绩的编号是否存在，若存在，则执行修改该学生成绩的 SQL 语句，代码如下：

```
if (validateID("stu_testid", "stu_test_relation", id)) {
    sqlStatement = "UPDATE stu_test_relation SET
score="+object.getscore().trim()+" WHERE "+"stu_testid="+id;
}
```

## 3、增添试卷功能的实现：

(1) 创建一个学生考试信息实体 Obj\_gradeinfo\_sub，来存储增添的学生成绩的信息，代码如下：

```
appstu.model.Obj_bankinfo objbankinfo = new appstu.model.Obj_bankinfo();
```

(2) 首先通过validateID(String id, String tname, String idvalue)方法查询该学生考试成绩编号是否存在，若不存在，则执行添加学生成绩SQL语句，代码如下：

```
if (validateID("problemid", "problem_info", objbankinfo.getProblemid().trim())){
    修改的SQL语句}
else {
    sqlStatement="Insert into stu_test_relation(stuid,testid,score)
values(""+sid+"",""+tid+"",""+object.getscore().trim()+"");";
    infoStr = "添加学生成绩信息";
}
```

## 4、删除学生成绩功能的实现：

(1) 读取试卷编号文本框中的内容，获得该学生的学号和考试的科目，根据学生的学号和考试科目编写删除SQL语句，SQL语句如下：

```
String sqlDel="delete from stu_test_relation where stuid = "" + sid + "" and
testid="" +id+""";
```

(2) 并通过删除记录方法DeleteObject()，将该学生的成绩删除，代码如下：

```
public boolean DeleteObject(String deleteSql) {  
    infoStr = "删除";  
    return AdapterObject(deleteSql);  
}
```

## 5.5 班级信息功能模块实现

教师登录以后，可以通过上方的菜单栏点击“班级信息”，点击对应的班级，进入该班级的信息窗口，该窗口展示了该班级所有学生的信息。由于这是老师进行查看的界面，所以不能对学生的任何信息进行修改。该页面的设计效果如图 5-4 所示。



学号	姓名	性别	班级	电话
1	李 小	男	软件191	17360323424
19102	张 三	女	软件191	13608252123
19103	杨 阳	男	软件191	17883382358
19104	卢 艳	女	软件191	15882519412
19105	王 川	男	软件191	18909097456
19106	李 四	女	软件191	19883289336
19107	张海洋	男	软件191	13541925687
19108	袁大丽	女	软件191	15982545756
19109	郑 风	男	软件191	13795684224

图 5-4 考试信息界面

### 1、查询班级中学生信息功能的实现：

(1) 在 buildTable() 方法下定义并编写查询该班级下所有学生的 SQL 语句，SQL 语句如下：

```
String sqlStr = "select sno,sname,ssex,classname,sphone from stu_info,class_info  
"+ "where classname='"+eventname+"' and stu_info.ClassID=class_info.classid;";
```

(2) 提取调用connection对象的execQuery(String)方法，并将查询的结果一一加入到DefaultTableModel表格模型中，代码如下：

```
rs = connection.prepareStatement(sqlStr).executeQuery();  
while (rs.next()) {  
    Vector vdata = new Vector();  
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {  
        vdata.addElement(rs.getObject(i));  
    }  
    tableModel.addRow(vdata); // 将集合添加到表格模型中
```

}

## 5.6 注册登录功能模块实现

用户进入程序后通过页面上的注册按钮进入注册页面,注册界面收集用户基本信息后将其信息存入数据库中,也可以直接在登录页面输入账号密码进行登录。用户登录的主页面设计效果如图 5-1 所示,用户注册界面如图 5-2 所示,。



图 5-1 用户登录页面

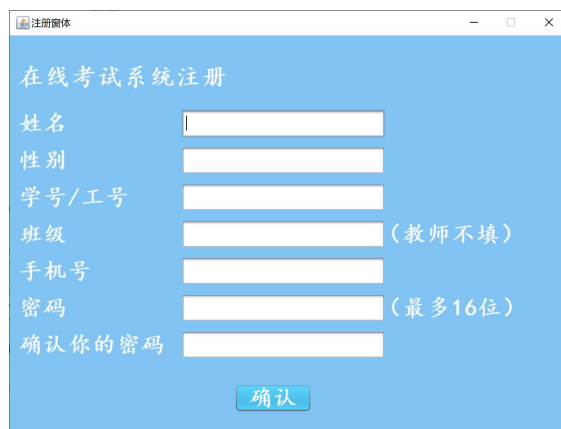


图 5-2 用户注册页面

用户登录实现思路:用户选择身份并且输入账号密码后在数据库中查询对应账户并且校对密码,如果正确则直接进入主页面,否则提示“账号或密码错误”。具体实现过程如下:

### 1、用户账户密码查询校对账户密码功能的实现:

(1) 定义并编写 SQL 查询语句, SQL 语句如下:  
if(身份为教师){

```
    sql = "select * from teacher_info where TeacherNo = ? and TeacherPWD = ?"; //定义查询预处理语句
```

```
}
```

```
else {
```

```
    sql = "select * from stu_info where SNo = ? and SPWD = ?"; //定义查询预处理语句
```

```
}
```

(2) 调用conn对象的prepareStatement(String)方法,将上述查询语句作为参数执行查询操作,并且将查到的结果存到对象user里面。

```
PreparedStatement statement = conn.prepareStatement(sql); //实例化PreparedStatement对象
```

```
statement.setString(1, userName); //设置预处理语句参数
```

```
statement.setString(2, passWord);
```

```
ResultSet rest = statement.executeQuery(); //执行预处理语句
```

```
while(rest.next()){
```



```

if(flag==1) {
    user.setId(rest.getInt(1));           //应用查询结果设置对象属性
    user.setUserNo(rest.getString(2));
    user.setUserName(rest.getString(3));
    user.setUserSex(rest.getString(4));
    user.setPWD(rest.getString(7));
    user.setIsActive(1);
    user.setUserPhone(rest.getString(5));
}
else {
    user.setId(rest.getInt(1));           //应用查询结果设置对象属性
    user.setUserNo(rest.getString(2));
    user.setUserName(rest.getString(3));
    user.setUserSex(rest.getString(4));
    user.setPWD(rest.getString(5));
    user.setIsActive(1);
    user.setUserPhone(rest.getString(7));
    user.setClassID(rest.getInt(8));
}
}

```

(3) 将上面查到的user对象的密码和用户输入的密码校对，如果正确则登录成功，进入主页面，否则登录失败，给出提示信息“用户名或密码错误”。

```

enterButton.addActionListener(new ActionListener() {      //按钮的单击事件
    public void actionPerformed(ActionEvent e) {
        UserDao userDao = new UserDao();                  //创建保存有
操作数据库类对象
        User user =
userDao.getUser(userNameTextField.getText(),passwordField.getText(),flag); //以用户添加
的用户名与密码为参数调用查询用户方法
        if(user.getId()>0){                                //判断用户编号是否大于0
            Session.setUser(user);                          //设置Session对象的User属性
值
            //RemoveButtonFrame frame = new RemoveButtonFrame();
//创建主窗体对象
            //frame.setVisible(true);                      //显示主窗体
//
JOptionPane.showMessageDialog(getContentPane(), "登录成功");
PCenter center=new PCenter();
center.setVisible(true);
Enter.this.dispose();                                    //销毁登录窗体
        }
        else{                                              //如果用户输入的用户名与密
码错误
            JOptionPane.showMessageDialog(getContentPane(), "用户名或密
码错误"); //给出提示信息

```

```

        userNameTextField.setText("");           //用户名文本框设置为空
        passwordField.setText("");             //密码文本框设置为空
    }
}
});

```

用户注册实现思路：用户输入基本用户信息，先核对用户的前后密码是否一致，若不一致则给出错误提示：“密码前后不一致”，若密码前后一致则先将基本信息先保存到一个user对象中，再使用statement对象的executeUpdate(String)方法执行插入语句。

（1）定义插入语句。

```

if(身份是教师) {
    sql = "insert
teacher_info(TeacherName,TeacherPWD,TeacherNo,TeacherSex,isActive,TeacherPhone)"

    +"values("+user.getUserName()+",""+user.getPWD()+",""+user.getUserNo()+",""+user.
getUserSex()+",""+1+"",""+user.getUserPhone()+")";
}
else {
    sql = "insert stu_info(SName,SPWD,Sno,SSex,isActive,SPhone,ClassID)"

    +"values("+user.getUserName()+",""+user.getPWD()+",""+user.getUserNo()+",""+user.
getUserSex()+",""+1+"",""+user.getUserPhone()+",""+user.getClassID()+")";
}

```

（2）核对用户的前后密码，密码相同则保存。

```

User user=new User();
if(passwordField1.getText().compareTo(passwordField2.getText())==0) {
    user.setPWD(passwordField2.getText());
    user.setUserName(userNameTextField.getText());
    user.setUserNo(userNoTextField.getText());
    user.setUserSex(userSexTextField.getText());
    user.setUserPhone(userPhoneTextField.getText());
    if(userClassTextField.getText()==null) {
        user.setFlag(1);
    }
    else {
        user.setFlag(0);
    }
}

```

```

user.setClassID(Integer.parseInt(userClassTextField.getText()));

```

（3）使用statement对象的executeUpdate(String)方法执行插入语句。

```

Statement statement = conn.createStatement();    //实例化用来执行sql语句的对象
// 增删改： executeUpdate(String sql)==>int, 影响的行数
int rows = statement.executeUpdate(sql);
//释放资源
if (statement != null) {
    statement.close();
}

```

```

    }
    if (conn != null) {
        conn.close();
    }
}

```

### 5.7个人中心功能模块实现

用户通过点击导航栏的个人中心查看个人信息，界面如图5-3所示；还可以点击信息面板上的信息修改进入修改信息界面，如图5-4所示。



图5-3 个人信息展示界面



图5-4 个人信息修改界面

查询信息实现思路：从数据库中根据用户的账号查询出用户的基本个人信息。并且放到对应的文本框中，文本框设为不可编辑。

(1) 查询用户个人信息并且插入到文本框中。

```

private BackgroundPanel PersonalCenter() {
    // TODO Auto-generated method stub
    if(contentPane==null) {
        Session s=new Session();
        contentPane = new BackgroundPanel();// 创建登录面板对象
        contentPane.setOpaque(false);// 面板透明
        contentPane.setImage(getToolkit().getImage(
            getClass().getResource("login.jpg")));// 设置面板背景图片
        contentPane.setLayout(null);

        JLabel Label=new JLabel("个人信息展示");
        Label.setBounds(10, 0, 400, 100);
        Label.setFont(new Font("楷体",Font.BOLD,30));
        Label.setForeground(new Color(255,255,255));
        //
        proLabel.setSize(80, 80);
        contentPane.add(Label);
        //姓名 userNameTextField
        JLabel userNameLabel = new JLabel("姓名");
        userNameLabel.setForeground(new Color(255,255,255));
        userNameLabel.setBounds(10, 90, 250, 35);
        userNameLabel.setFont(new Font("楷体",Font.BOLD,28));
    }
}

```

```

contentPane.add(userNameLabel);

userNameTextField = new JTextField(s.getUser().getUserName());
userNameTextField.setBounds(110, 90, 250, 35);
userNameTextField.setEditable(false);
contentPane.add(userNameTextField);
//性别 userSexTextField
JLabel userSexLabel = new JLabel("性别");
userSexLabel.setForeground(new Color(255,255,255));
userSexLabel.setBounds(10, 135, 250, 35);
userSexLabel.setFont(new Font("楷体",Font.BOLD,28));
contentPane.add(userSexLabel);

userSexTextField = new JTextField(s.getUser().getUserSex());
userSexTextField.setBounds(110, 135, 250, 35);
userSexTextField.setEditable(false);
contentPane.add(userSexTextField);
//学号 userNoTextField
if(s.getUser().getFlag()==1) {
    JLabel userNoLabel = new JLabel("工号");
    userNoLabel.setForeground(new Color(255,255,255));
    userNoLabel.setBounds(10, 180, 250, 35);
    userNoLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userNoLabel);

    //userClassTextField
    JLabel userClassLabel = new JLabel("您的身份是一个教师");
    userClassLabel.setForeground(new Color(255,255,255));
    userClassLabel.setBounds(10, 225, 250, 35);
    userClassLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userClassLabel);
}
else {
    JLabel userNoLabel = new JLabel("学号");
    userNoLabel.setForeground(new Color(255,255,255));
    userNoLabel.setBounds(10, 180, 250, 35);
    userNoLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userNoLabel);

    //userClassTextField
    JLabel userClassLabel = new JLabel("班级");
    userClassLabel.setForeground(new Color(255,255,255));
    userClassLabel.setBounds(10, 225, 250, 35);
    userClassLabel.setFont(new Font("楷体",Font.BOLD,28));
}

```

```

        contentPane.add(userClassLabel);

        userClassTextField = new
JTextField(Integer.toString(s.getUser().getClassID()));
        userClassTextField.setBounds(110, 225, 250, 35);
        userClassTextField.setEditable(false);
        contentPane.add(userClassTextField);
    }
    userNoTextField = new JTextField(s.getUser().getUserNo());
    userNoTextField.setBounds(110, 180, 250, 35);
    userNoTextField.setEditable(false);
    contentPane.add(userNoTextField);

    //userPhoneTextField
    JLabel userPhoneLabel = new JLabel("手机号");
    userPhoneLabel.setForeground(new Color(255,255,255));
    userPhoneLabel.setBounds(10, 270, 250, 35);
    userPhoneLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userPhoneLabel);

    userPhoneTextField = new JTextField(s.getUser().getUserPhone());
    userPhoneTextField.setBounds(110, 270, 250, 35);
    userPhoneTextField.setEditable(false);
    contentPane.add(userPhoneTextField);

    modify=new JButton("修改信息");
    modify.setBounds(135, 325, 155, 35);
    modify.setFont(new Font("楷体",Font.BOLD,28));
    modify.setForeground(new Color(255,255,255));
    modify.setBackground(new Color(51,161,201));
    contentPane.add(modify);

    });
}
return contentPane;
}

```

修改信息实现思路：从数据库中根据用户的账号查询出用户的基本个人信息。并且放到对应的文本框中，将可修改的信息文本框设为可编辑。编辑完成后点击确认修改，然后从文本框中提取信息。如果原密码为空则证明不更改密码，读入文本框所有信息（密码置为原密码）对原信息进行修改。如果原密码非空说明用户需要修改密码，则核对原密码是否正确。不正确则发出提示信息“原密码错误”并且不予修改。正确则重新读入所有文本框的信息，对原信息进行修改。

(1) 将用户信息查询出来并放到文本框中并且设置可修改的为可编辑。

```

Session s=new Session();
    contentPane = new BackgroundPanel();// 创建登录面板对象
    contentPane.setOpaque(false);// 面板透明
    contentPane.setImage(getToolkit().getImage(
        getClass().getResource("login.jpg")));// 设置面板背景图片
    contentPane.setLayout(null);

    JLabel Label=new JLabel("个人信息展示");
    Label.setBounds(10, 0, 400, 100);
    Label.setFont(new Font("楷体",Font.BOLD,30));
    Label.setForeground(new Color(255,255,255));
//
    proLabel.setSize(80, 80);
    contentPane.add(Label);
    //姓名 userNameTextField
    JLabel userNameLabel = new JLabel("姓名");
    userNameLabel.setForeground(new Color(255,255,255));
    userNameLabel.setBounds(10, 90, 250, 35);
    userNameLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userNameLabel);

    userNameTextField = new JTextField(s.getUser().getUserName());
    userNameTextField.setBounds(210, 90, 250, 35);
    contentPane.add(userNameTextField);
    //性别 userSexTextField
    JLabel userSexLabel = new JLabel("性别");
    userSexLabel.setForeground(new Color(255,255,255));
    userSexLabel.setBounds(10, 135, 250, 35);
    userSexLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userSexLabel);

    userSexTextField = new JTextField(s.getUser().getUserSex());
    userSexTextField.setBounds(210, 135, 250, 35);
    contentPane.add(userSexTextField);
    //学号 userNoTextField
    if(s.getUser().getFlag()==1) {
        JLabel userNoLabel = new JLabel("工号");
        userNoLabel.setForeground(new Color(255,255,255));
        userNoLabel.setBounds(10, 180, 250, 35);
        userNoLabel.setFont(new Font("楷体",Font.BOLD,28));
        contentPane.add(userNoLabel);
    }
    else {
        JLabel userNoLabel = new JLabel("学号");
        userNoLabel.setForeground(new Color(255,255,255));

```

```

        userNoLabel.setBounds(10, 180, 250, 35);
        userNoLabel.setFont(new Font("楷体",Font.BOLD,28));
        contentPane.add(userNoLabel);
    }
    userNoTextField = new JTextField(s.getUser().getUserNo());
    userNoTextField.setBounds(210, 180, 250, 35);
    userNoTextField.setEditable(false);
    contentPane.add(userNoTextField);
    //userClassTextField
    JLabel userClassLabel = new JLabel("班级");
    userClassLabel.setForeground(new Color(255,255,255));
    userClassLabel.setBounds(10, 225, 250, 35);
    userClassLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userClassLabel);

    userClassTextField = new
JTextField(Integer.toString(s.getUser().getClassID()));
    userClassTextField.setBounds(210, 225, 250, 35);
    userClassTextField.setEditable(false);
    contentPane.add(userClassTextField);
    //userPhoneTextField
    JLabel userPhoneLabel = new JLabel("手机号");
    userPhoneLabel.setForeground(new Color(255,255,255));
    userPhoneLabel.setBounds(10, 270, 250, 35);
    userPhoneLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userPhoneLabel);

    userPhoneTextField = new JTextField(s.getUser().getUserPhone());
    userPhoneTextField.setBounds(210, 270, 250, 35);
    contentPane.add(userPhoneTextField);

    //passwordField1
    JLabel userPWDLLabel = new JLabel("请输入原密码");
    userPWDLLabel.setForeground(new Color(255,255,255));
    userPWDLLabel.setBounds(10, 315, 250, 35);
    userPWDLLabel.setFont(new Font("楷体",Font.BOLD,28));
    contentPane.add(userPWDLLabel);

    passwordField1 = new JPasswordField(16);
    passwordField1.setBounds(210, 315, 250, 35);
    contentPane.add(passwordField1);

    JLabel remarkLabel2 = new JLabel("（不修改则留空）");
    remarkLabel2.setForeground(new Color(255,255,255));

```

```

remarkLabel2.setBounds(450, 315, 250, 35);
remarkLabel2.setFont(new Font("楷体",Font.BOLD,28));
contentPane.add(remarkLabel2);
//passwordField2
JLabel userPWDLable2 = new JLabel("请输入新密码");
userPWDLable2.setForeground(new Color(255,255,255));
userPWDLable2.setBounds(10, 360, 250, 35);
userPWDLable2.setFont(new Font("楷体",Font.BOLD,28));
contentPane.add(userPWDLable2);

passwordField2 = new JPasswordField(16);
passwordField2.setBounds(210, 360, 250, 35);
contentPane.add(passwordField2);

JLabel remarkLabel3 = new JLabel("（最多16位）");
remarkLabel3.setForeground(new Color(255,255,255));
remarkLabel3.setBounds(450, 360, 250, 35);
remarkLabel3.setFont(new Font("楷体",Font.BOLD,28));
contentPane.add(remarkLabel3);

modify=new JButton("确认修改");
modify.setBounds(245, 425, 165, 35);
modify.setFont(new Font("楷体",Font.BOLD,28));
modify.setForeground(new Color(255,255,255));
modify.setBackground(new Color(51,161,201));
contentPane.add(modify);

```

（2）查询出现有的信息并且保存为临时变量。

```

Session s=new Session();
    User user=new User();
    String PWD0=s.getUser().getPWD();
    String PWD1=passwordField1.getText();
    String PWD2=passwordField2.getText();
    String userName=userNameTextField.getText();
    String userSex=userSexTextField.getText();
    String userNo=userNoTextField.getText();
    int classID=Integer.parseInt(userClassTextField.getText());

```

（3）如果原密码为空则证明不更改密码，读入文本框所有信息（密码置为原密码）对原信息进行修改。如果原密码非空说明用户需要修改密码，则核对原密码是否正确。不正确则发出提示信息“原密码错误”并且不予修改。正确则重新读入所有文本框的信息，对原信息进行修改。

```

modify.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

```



```

// TODO Auto-generated method stub

if(PWD1==null) {
    PWD2=PWD0; //没输入原密码则表示不修改密码
}
else if(PWD1==null || PWD1.compareTo(PWD0)==0) {
    user.setPWD(PWD2);
    user.setUserName(userName);
    user.setUserNo(userNo);
    user.setUserSex(userSex);
    user.setUserPhone(userPhoneTextField.getText());
    if(classID==0) {
        user.setFlag(1);
    }
    else {
        user.setFlag(0);

        user.setClassID(Integer.parseInt(userClassTextField.getText()));
    }
    try {
        new User_modify().modify_user(user);
        JOptionPane.showMessageDialog(getContentPane(), "修改信
息成功");

        s.setUser(user);
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
        JOptionPane.showMessageDialog(getContentPane(), "修改信
息失败");
    }

}
else if(PWD1.compareTo(PWD0)!=0) {
    JOptionPane.showMessageDialog(getContentPane(), "原密码错");
    System.out.print(PWD0);
}
}

```

## 6、设计总结

此在线考试系统完成了用户注册及登录功能、用户个人中心、题库管理、试卷管理、考试安排、班级信息查看这几个功能，基本上实现了在线考试系统的核心功能。在窗口方面，界面简约、操作简单，易上手，非常地适合新用户。在差错处理方面，各个功能均配只有各自特有的差错处理模式（例如弹窗警告，文本框的文本提示信息等），给用户及时的反馈，非常的人性。

不足方面

1. 时间较为仓促，还有很多功能尚未实现；
2. 在添加、修改试卷时，若用户想要添加或修改试卷的题库，则应该直接选择现有题库，而不是用户直接输入题库的名称。
3. 由于知识认知上的不充足，只实现了一些比较简单的功能，如插入以及查找功能。
4. 虽然界面做出来了，但还是有很多需要改进之处，比如界面的美观性以及交互性的改善，没有来的及做。
5. 不足之处在于界面不太美观，还有学生和教师信息是分开管理的，查询、插入、修改信息时需要分别判断是教师还是学生，代码量大且较为冗余。
6. 有些功能的设计可能考虑不到位，只是在自己的意识下进行了设计已经实现，并没有根据客户的需求去改进，只是一个简单的系统，到一个成熟的系统还需要不断地改进完善后才能实现。

## 7、设计体会

郑小菁：设计体会：在最开始设计过程中，遇到了数据连接的问题一直无法连接到数据库，后来经过查询资料，添加数据库驱动，解决了该问题，在设计时，一边做用户，一边做开发者，发现其中的问题，并解决问题，比如在添加、修改试卷是，若用户想要添加或修改试卷的题库，则应该直接选择现有题库，而不是用户直接输入题库的名称。所以做是设计，不仅要功能完善，更要达到功能使用的合理性，这也算是本次课程设计的收获。

杨佳佳：感谢能有这次实践机会，通过这次课程设计让我了解到理论与实际相结合的重要性，只有理论知识是远远不够的，只有把所学的理论知识与实践结合起来，才能真正发挥出自身的价值，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。在本次设计的过程中也遇到了许多问题，可以说是困难重重，让我体会到了做一个完整应用项目的难度。过程远没有之前想象的那么简单。通过这次课程设计让我深刻地认识到自身的不足点，基础是非常重要的，有了基础才能更加深入学习其他的东西，也暴露出自己动手敲代码的能力不足，想要在这条道路上走下去就必须自己下去刻苦练习实践。

杨顺利：通过本次课程设计，我深刻认识到自己的不足，也为自己上个假期没有提前学习 php 或者 jsp 感到遗憾。自己之前也接触过一些项目，也跟着别人写过一点东西，但是真的要自己独立构思和创作，却就像从未接触过这个东西一样，根据这一点，我也深刻认识到我平时只体验观看的快感，而不经常动手实践的弊端。本次课程设计第二点收获在于我对系统设计的流程更加清楚了。

# 附录 A、操作手册

在 eclipse 里面按照 Java Application 运行即可启动平台，首先进入登录页面，如图 A-1 所示。



图A-1 登录页面

若是还没有注册可以进行注册操作，示例如图 A-2 所示。



图A-2 注册操作

注册完成后即返回登录界面。输入账号密码登录后进入平台首页，首页如图 A-3 所示。



图A-3 平台首页

点击题库信息，可以查看题库的信息运行效果如图 A-4 所示。



图A-4 题库信息页

在题库信息这里可以增加每个题库里的题目，先点击添加，写完后再点击存盘即可。如图 A-5 所示。



图 A-5 添加题目页面

点击试卷信息，可以查看试卷信息。效果如图 A-6 所示。

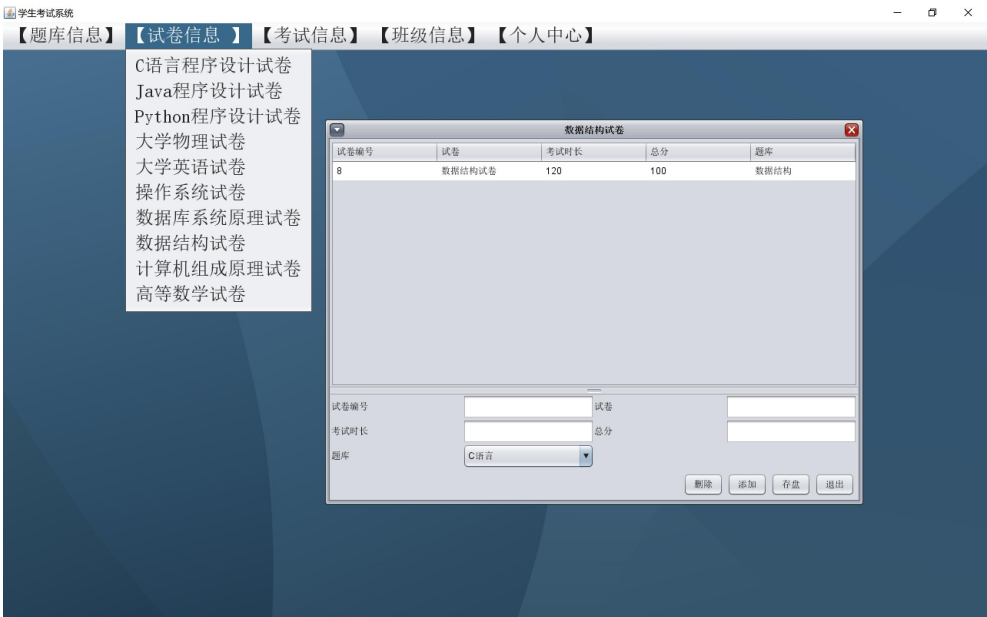


图 A-6 查看试卷信息

点击下面的添加按钮即可开始添加试卷，输入试卷名称，考试时长，总分以及选择题库之后可以点击存盘进行保存操作。如图 A-7 所示。



图 A-7 添加试卷

点击考试信息，可以查询对应学生的考试成绩。如图 A-8 所示。



图 A-8 学生成绩查询

点击下面的添加按钮即可开始添加学生成绩，输入学号，成绩以及考试科目之后可以点击存盘进行保存操作。如图 A-9 所示。



图 A-9 添加学生成绩

点击班级信息可以查询对应班级学生的详细信息，如图 A-10 所示。



图 A-10 查看学生信息



点击个人中心可以查看用户的个人信息（教师的班级显示为0）。如图 A-11 所示。



图 A-11 查看个人信息

点击修改信息，可以对个人信息进行修改。如图 A-12 所示。

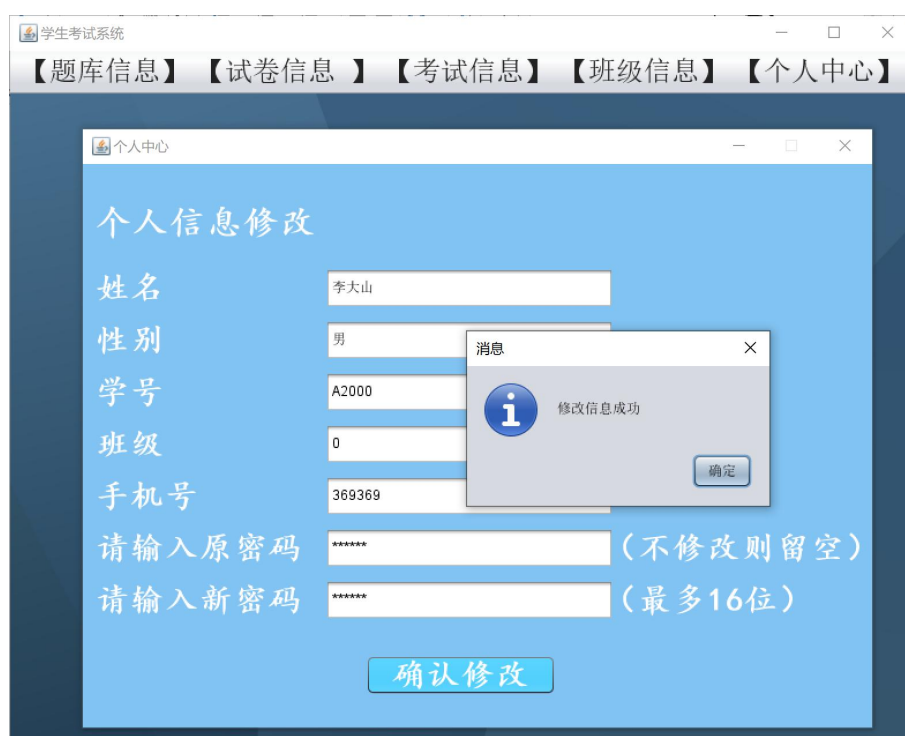


图 A-12 修改个人信息

