

学生实习报告

课程编号： B01051

课程名称：计算机综合课程设（二）

学 号： 32001014

姓 名： 李琛

专业班级： 计算机2001班

所在学院： 计算学院

报告日期： 2022 年 9 月 8 日

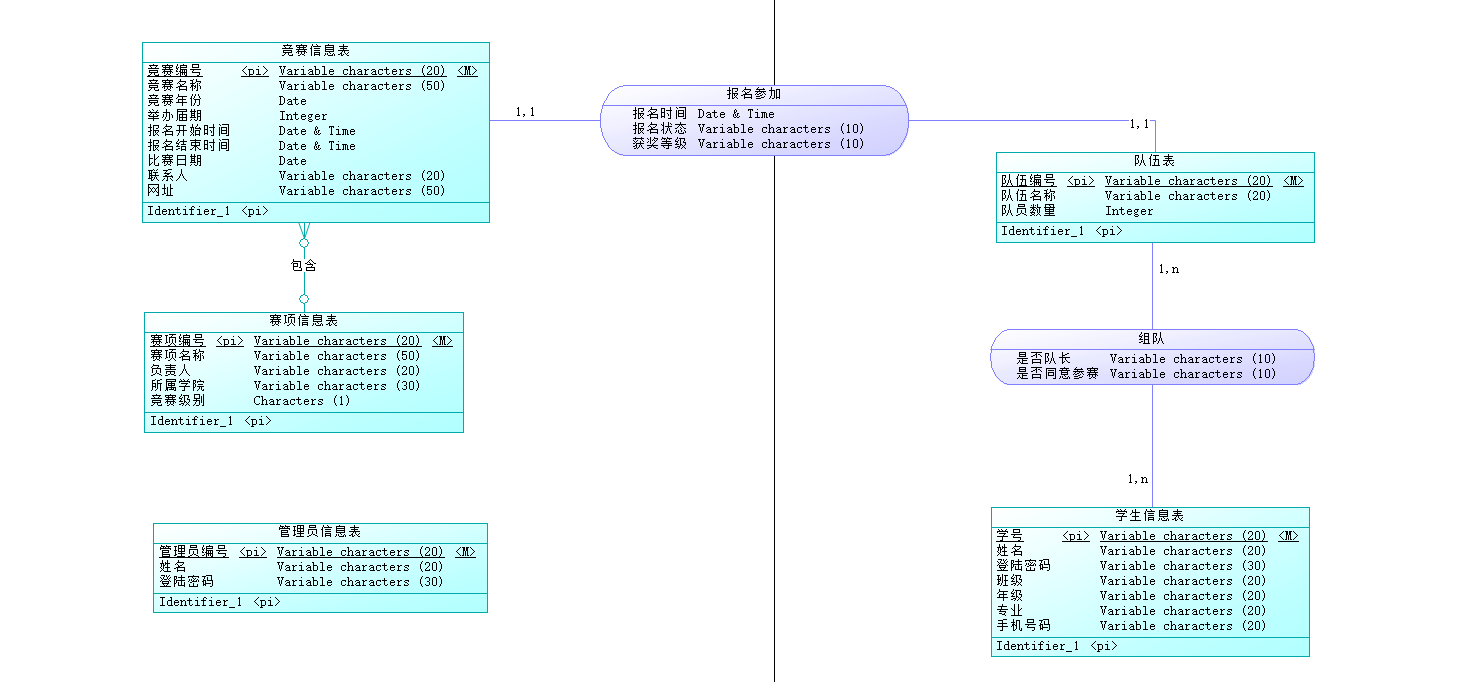
|  |  |  |  |
| --- | --- | --- | --- |
| 实习时间 | 20 22 －20 23 学年 □ 秋季学期 □ 春季学期 √ 暑期短学期  自 2022年 8 月 26 日 至 2022年 9 月 9 日 | | |
| 实习类型 | □ 认识（社会调查）实习  √ 教学（课程、专题、金工）实习  □ 毕业（综合）实习 | 实习地点 |  |
| 实习单位评语： | | | |
| 实习单位（盖章）：  年 月 日 | | | |
| 实习指导（带队）教师评语： | | | |
| 实习总评成绩：  教师（签名）：  年 月 日 | | | |

注：学生的实习总结等文档附在本封面之后

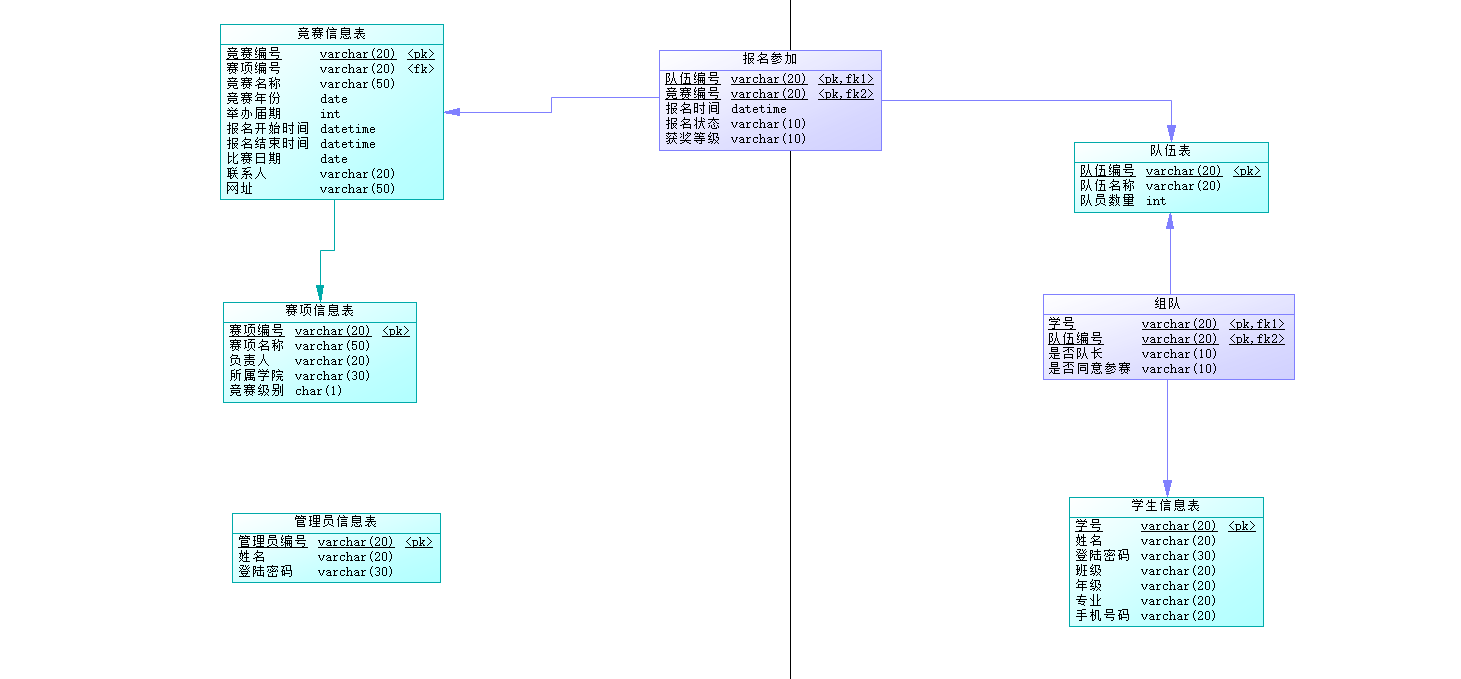
**实习总结**

**数据库设计部分：**

完善后的 **CompetitionSystem概念模型**



完善后的 **CompetitionSystem物理模型**



**系统设计部分：**

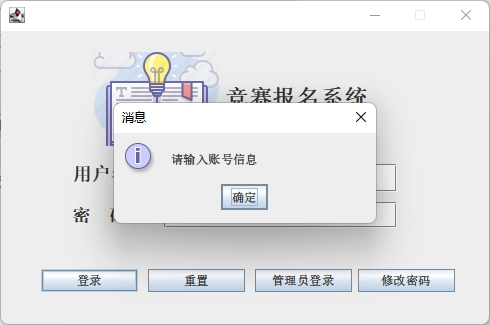
1. **登录模块：**

**学生登录：**

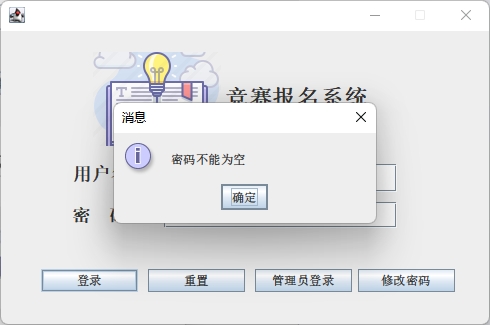
以学生登陆页面作为系统起始页面，在该页面上有登录、重置、管理员登录和修改密码选项。



若没有输入账号密码，则会出现以下提示信息：



若只输入了账号，则会出现以下提示信息：

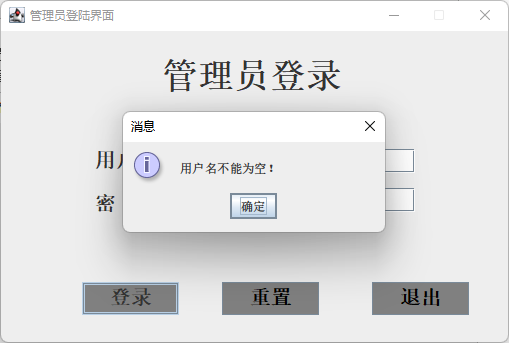


**管理员登录：**

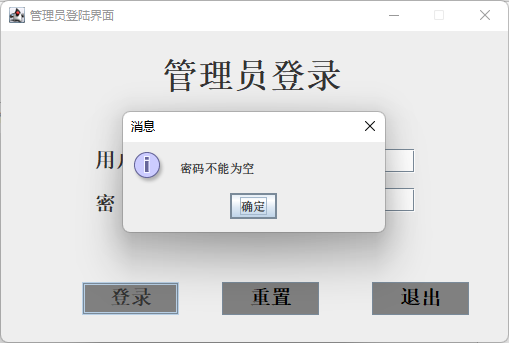
点击管理员登录选项即会弹出管理员登陆界面，管理员界面有登录重置和退出三个功能选项



未输入管理员账号密码时，会出现以下提示：

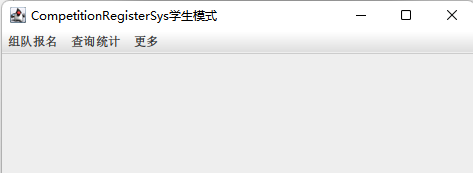


当只输入账号没有输入密码时，会出现以下提示：



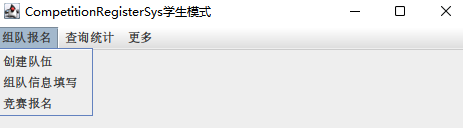
**二、学生系统主界面**

学生系统主页面主要分为三个功能面板：组队报名，查询统计以及更多



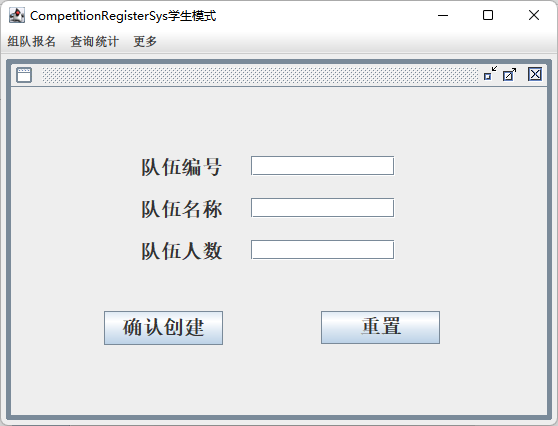
**组队报名：**

该模块内有三个功能



**创建队伍：**

**输入队伍编号，队伍名称，队伍人数进行队伍的创建**

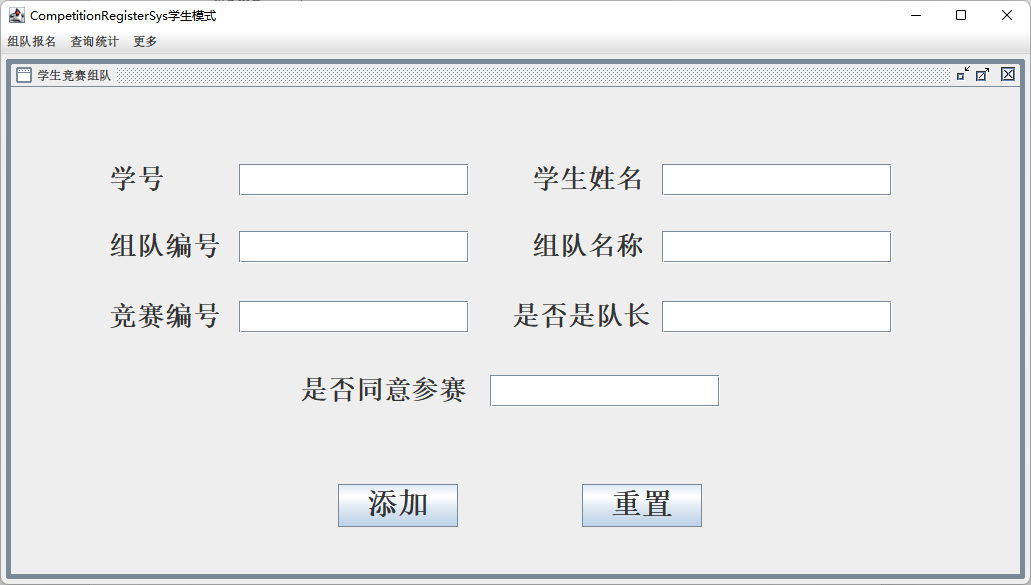


**组队信息填写：**

在该表格中填写组队信息

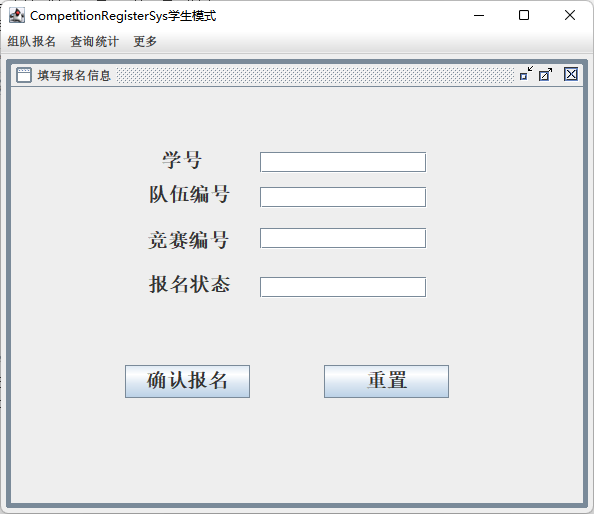
若为队长则输入‘是’，若不是则输入‘否’

若同意参赛则输入‘是’，若不同意参赛则输入‘否’

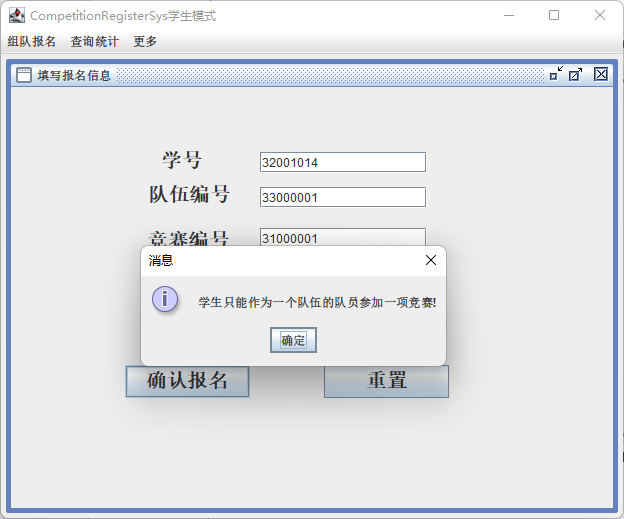


在进行添加时，应注意添加现有同学的学号、现有队伍的编号和现有竞赛的编号，否则会出现添加失败的提示

**竞赛队伍报名：**



应注意在一项竞赛中，学生只能参加一个队伍进行报名，如果参加同一个竞赛的多个队伍进行报名，则会出现学生稚嫩作为一个队伍的队员参加一项竞赛的提示



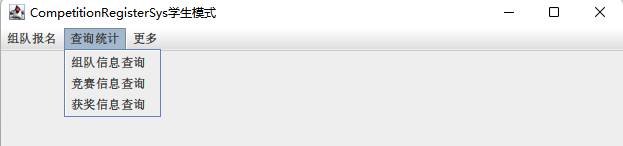
**查询统计：**

查询统计内有三个功能，组队信息查询，竞赛信息查询，获奖信息查询

组队信息查询供学生参看已经组队的同学和队伍信息

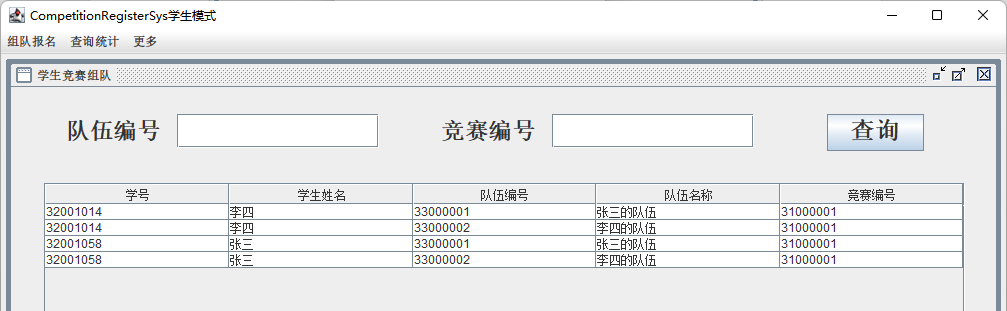
竞赛信息查看供学生参看现存的竞赛

获奖信息查询供学生查询队伍和个人的获奖情况



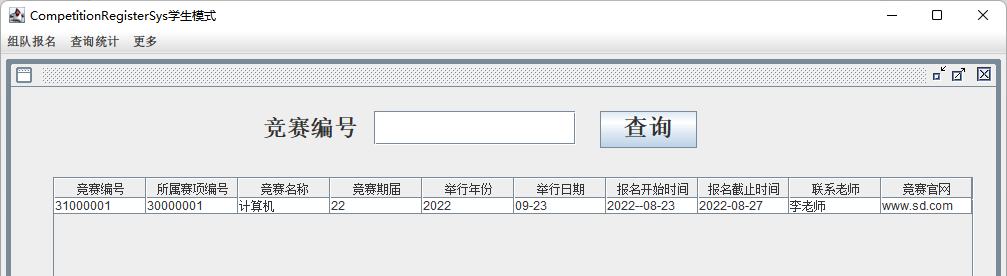
**组队信息查询：**

可以根据队伍编号和竞赛编号分别查询队伍和队员信息



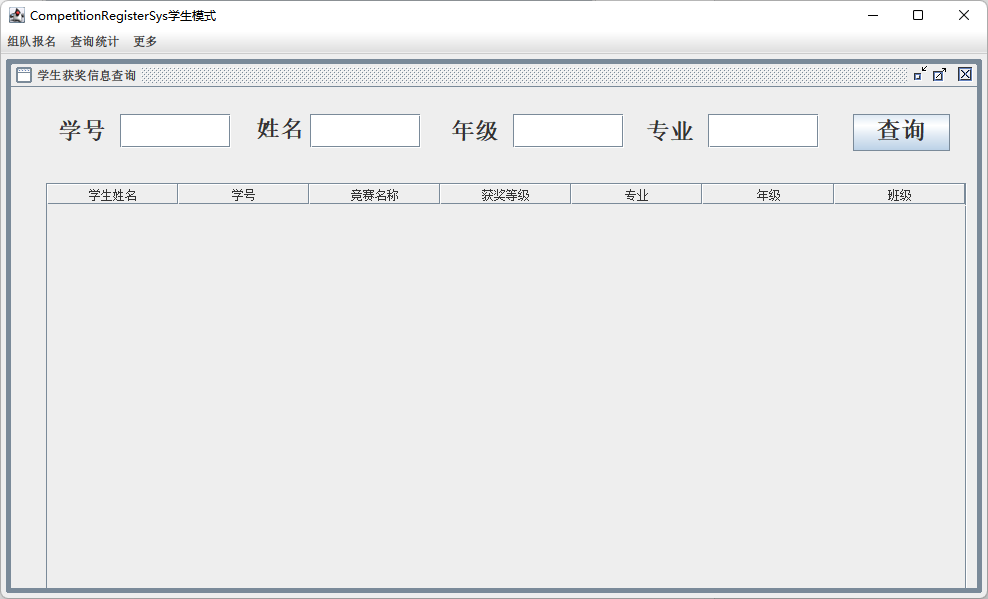
**竞赛信息查询：**

可以根据竞赛编号从而查询特定的竞赛信息

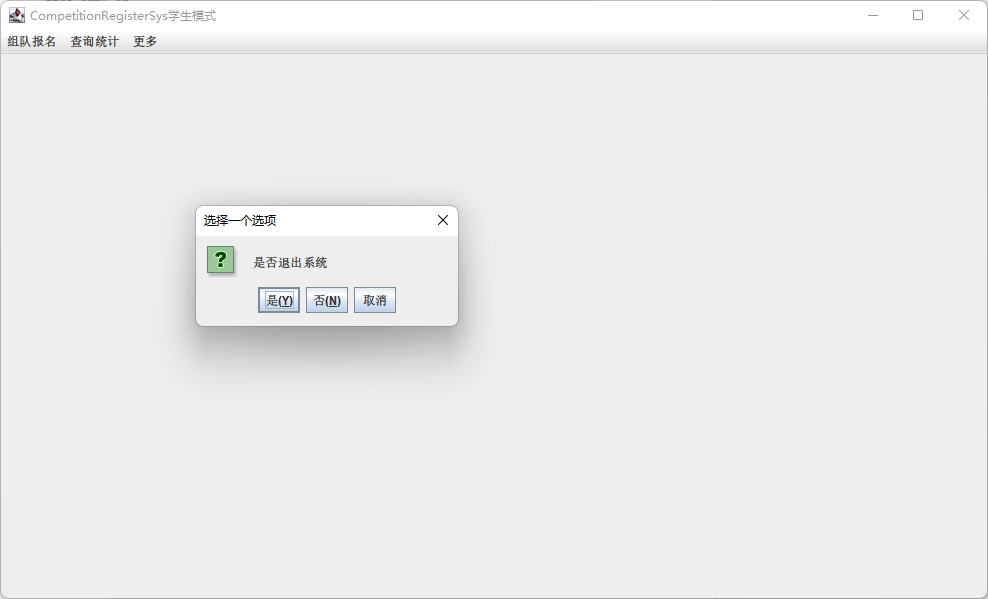


**获奖信息查询：**

**可以根据学号、姓名、年级、专业四个信息分别进行查询**



**安全退出功能：**



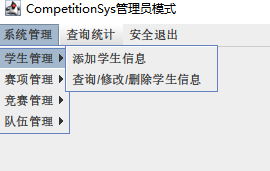
**管理员系统主页面：**

管理员主页面分为四个功能模块：学生管理、赛项管理、竞赛管理、队伍管理

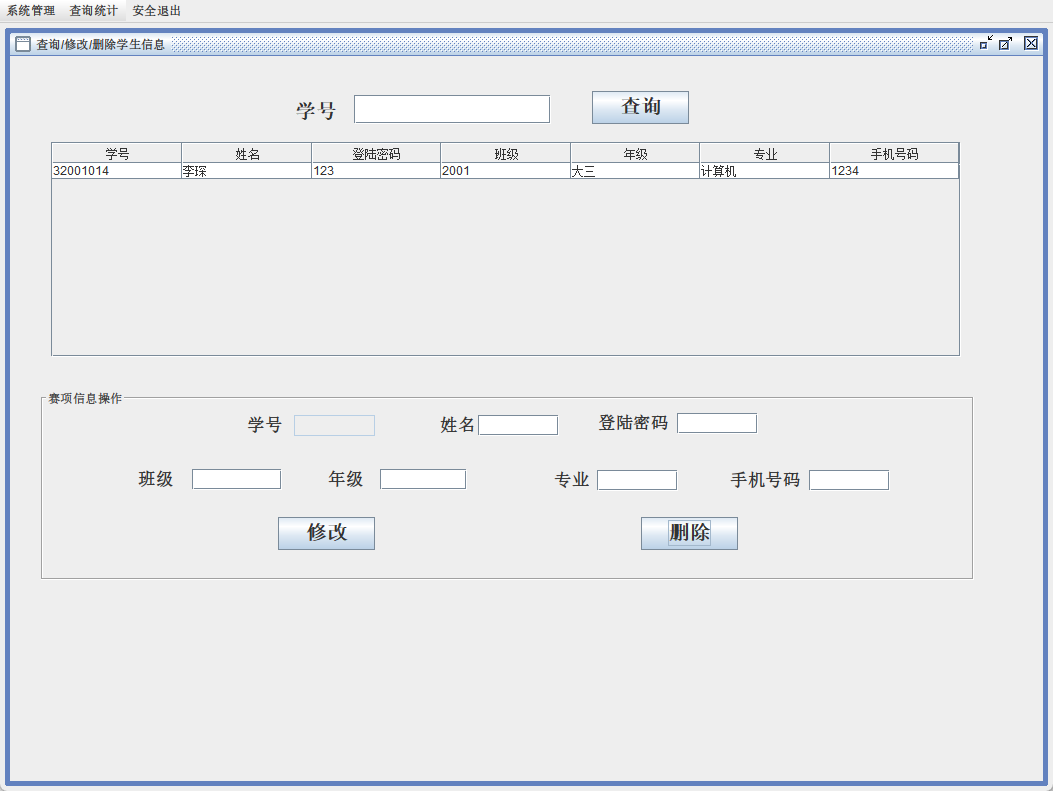
每一个管理中都分别由增加信息、修改信息、查询信息、删除信息的操作



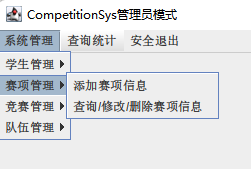
**学生管理：**



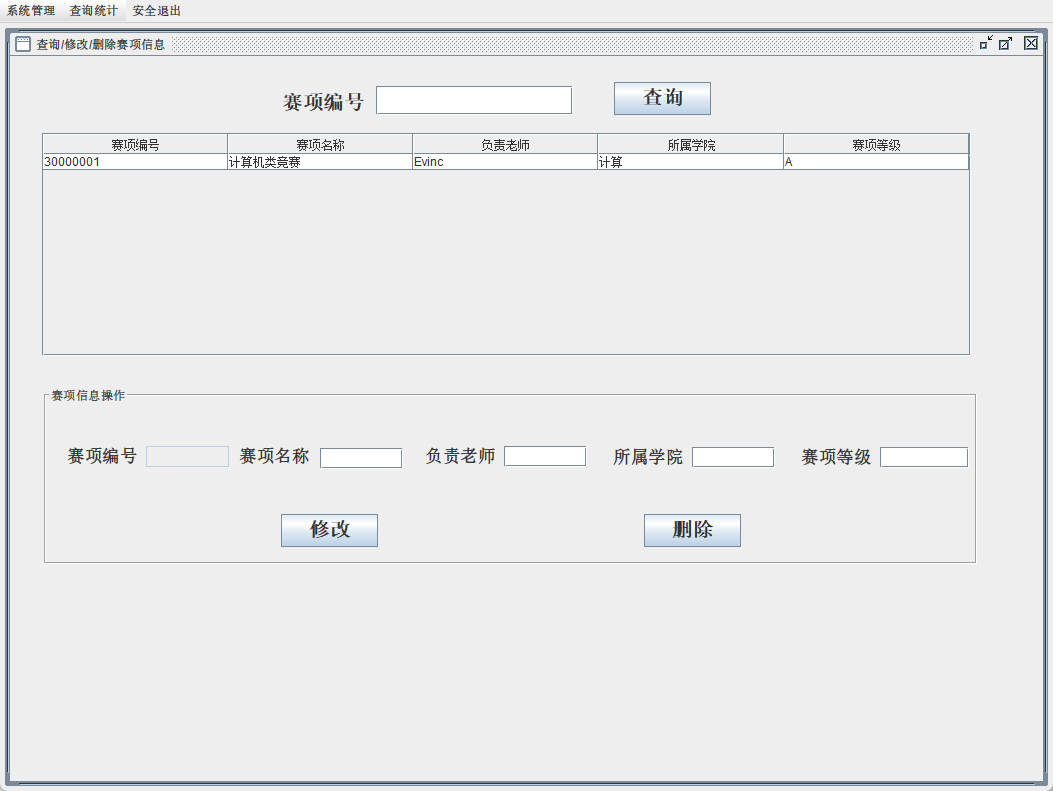




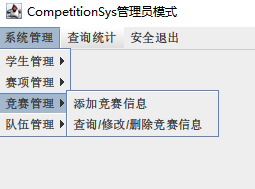
**赛项管理：**



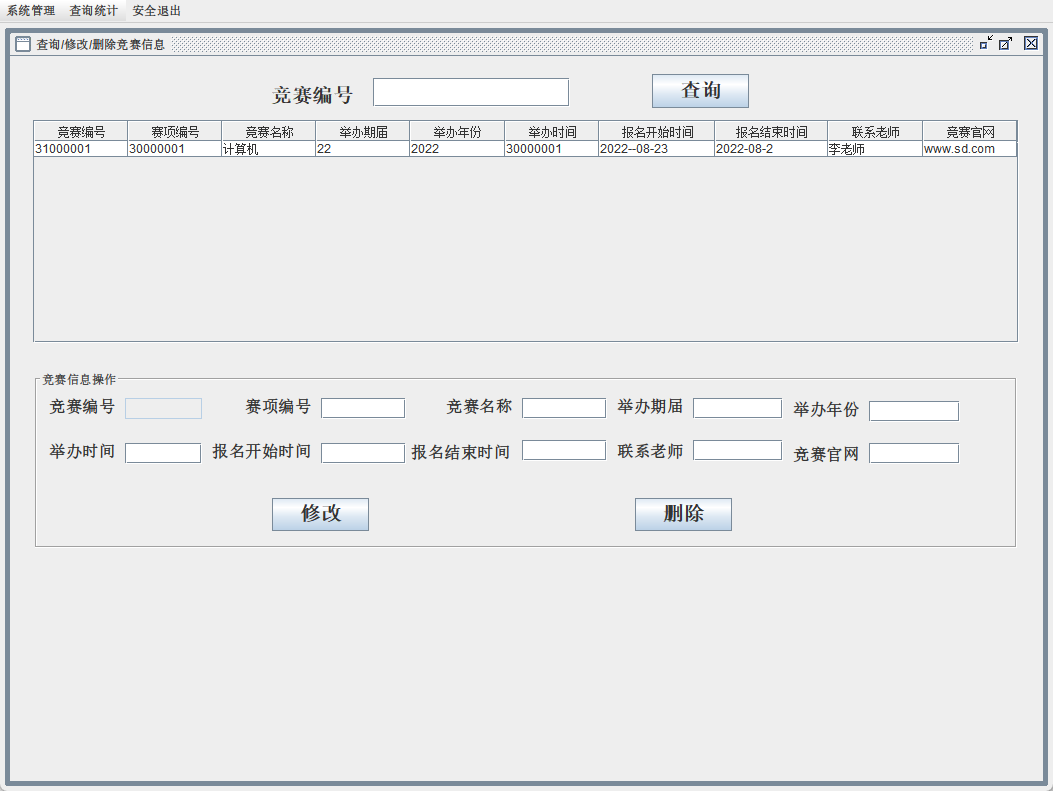




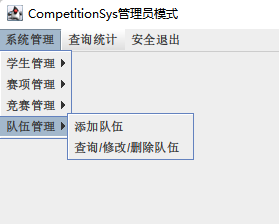
**竞赛管理：**

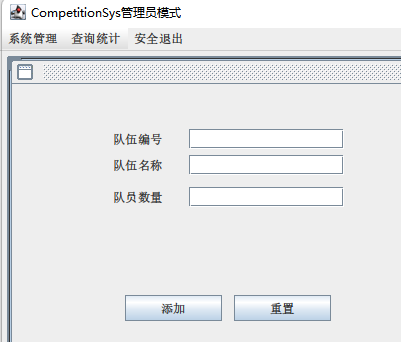


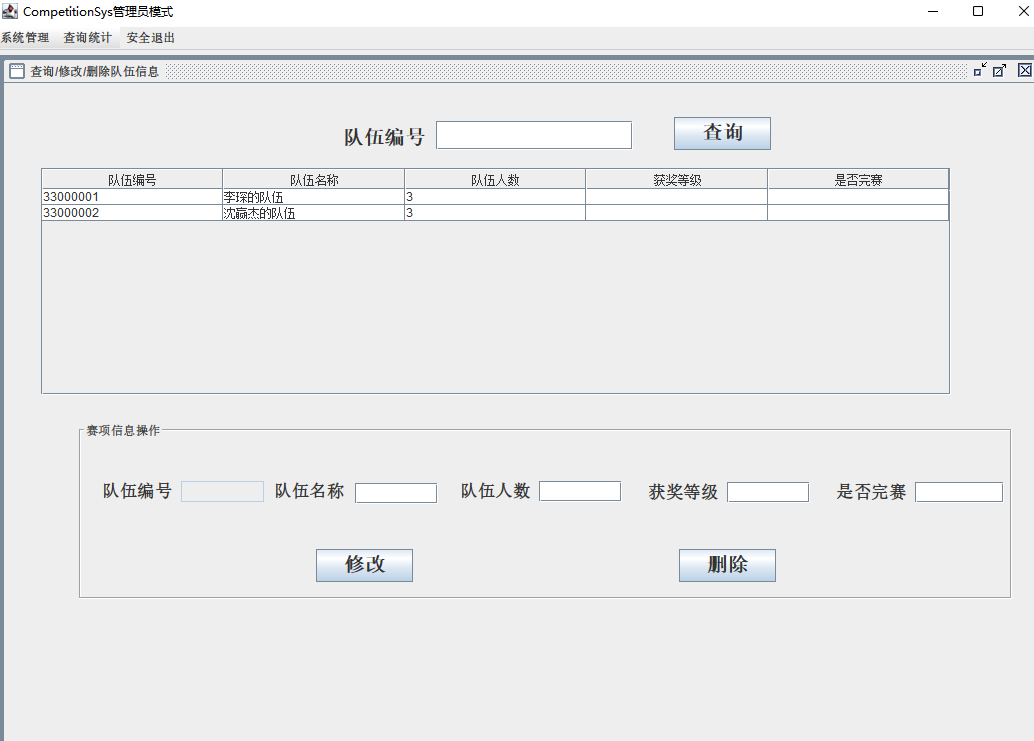




**队伍管理：**







**三、项目设计过程反思：**

**应该在一开始就在Navicat中把所有的操作模拟一遍，因为这个系统本质上就是对数据库的操作**

* **管理对象之间的关系就是表的定义和表之间的关系,**
* **管理对象的信息就是表中的属性信息 ,**
* **对信息的增删改查就是sql语句**

在数据库中全部演示过, 都觉得可以了之后, 然后根据MVC构建模式构建model view control

model就是用Java对于数据库中表的呈现

再写control, 也就是要做的一系列对数据的操作, 也就是sql语句

最后再写view, 根据功能学习一下怎么把数据显示到页面山, 把页面设计一下就好

而这次写的过程恰好把关键的第二三步反过来, 先弄的界面, 界面弄好了才去弄得功能

原因是用了windowsBuilder这个插件, 对于这个东西完全不熟悉, 以为这个的操作才是重点, 在界面设计上花费了比较多的时间

事实是功能没有想好, 界面弄得也很慢

**一切都是要把功能放在第一位**,然后再去考虑怎么在界面上显示的问题

当然也要注意 显示的形式 和 功能展示结果的形式 的一致性