

成绩





Java程序设计课程

设计报告

**学院** 电子信息工程学院

**专业** 网络工程

**班级** 23级（本）01班

**学号** 231040600122

**学生姓名** 董诚义

**授课教师** 董敏

**完成日期** 2025年 月 日

# 高校健身房管理系统

## 1需求分析

### 1.1需求概述

高校健身房管理系统是为了方便学校对健身房信息进行统一、规范和高效的管理而设计的。该系统整合了学生会员、健身器材、健身课程的基本信息数据，实现了从录入到管理的全过程信息化管理，满足学校日常教务工作的需求。

系统主要包括健身器材管理、健身课程管理、会员信息管理、管理员信息管理四大模块。目标是取代传统的人工记录和管理方式，通过计算机软件实现对会员、员工、健身器材、课程等核心资源的高效、准确、便捷的管理。

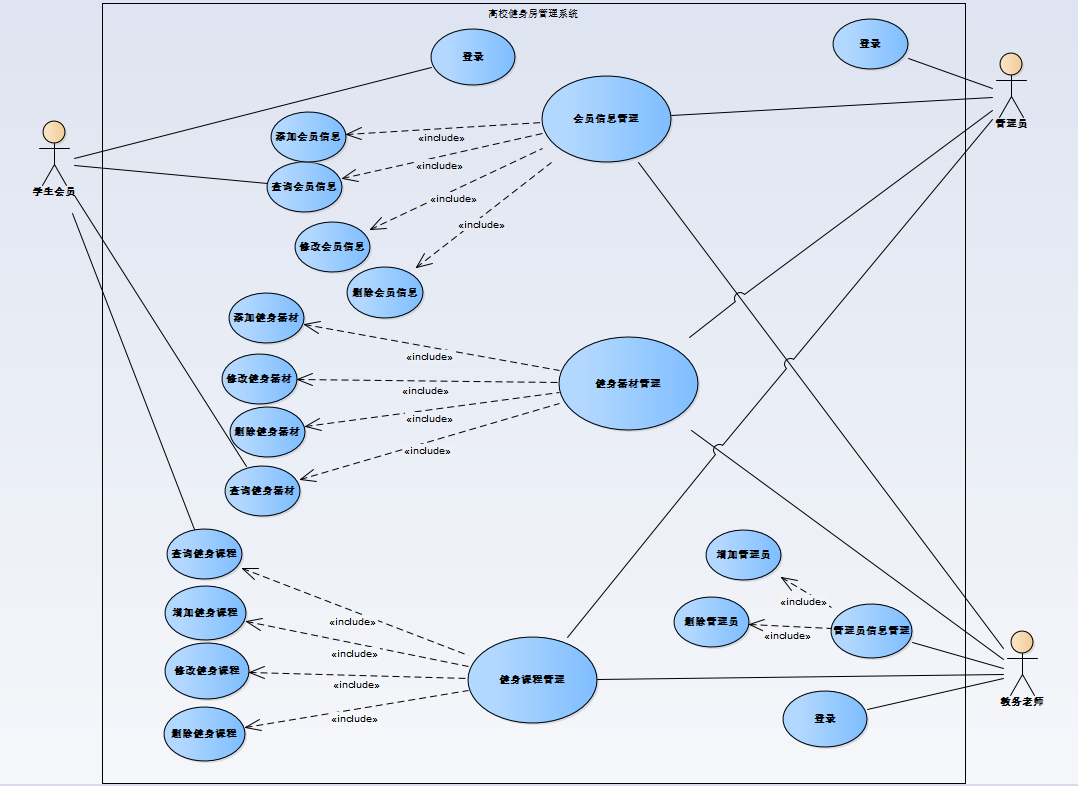
### 1.2功能模型分析

高校健身房管理系统可抽象出三个主要参与者：学生会员、管理员和教务老师。系统的核心用例如下：健身器材管理、健身课程管理、学生信息管理、管理员信息管理。其中，教务老师拥有健身器材管理、健身课程管理、会员信息管理、管理员信息管理的权限，管理员可进行健身器材管理、健身课程管理、会员信息管理，学生仅可登录并查询健身器材、健身课程、自身信息。

系统参与者与对应用例的关系如表11所示，用例视图结构如图11所示。

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用例描述 | 参与者 |
| 登录 | 输入用户名和密码进行登录 | 学生 |
| 查询会员信息 | 学生登录系统后，查看自己的信息，教务老师登录后可以查看所有学生的信息 | 学生、教务老师 |
| 添加会员信息 | 教务老师登录后，可添加会员信息 | 教务老师 |
| 修改会员信息 | 教务老师登录后，可修改会员信息 | 教务老师 |
| 删除会员信息 | 教务老师登录后，可删除会员信息 | 教务老师 |
| 查询健身器材 | 学生登录系统后，查看自己的信息，教务老师登录后可以查看所有学生的信息 | 学生、教务老师 |
| 添加健身器材 | 教务老师登录后，可添加会员信息 | 教务老师 |
| 修改健身器材 | 教务老师登录后，可修改会员信息 | 教务老师 |
| 删除健身器材 | 教务老师登录后，可删除会员信息 | 教务老师 |
| 查询健身课程 | 学生登录系统后，查看自己的信息，教务老师登录后可以查看所有学生的信息 | 学生、教务老师 |
| 添加健身课程 | 教务老师登录后，可添加会员信息 | 教务老师 |
| 修改健身课程 | 教务老师登录后，可修改会员信息 | 教务老师 |
| 删除健身课程 | 教务老师登录后，可删除会员信息 | 教务老师 |

**表1-1高校健身房管理系统用例分析表**



**图1-1高校健身房管理系统用例图**

|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 用户名 | 密码 | 主要权限 |
| 管理员 | admin | admin123 | 查看、管理所有信息 |
| 老师/教练 | Teacher | Teacher123 | 查看和管理课程、查看会员、管理课程报名 |
| 会员 | Member | Member123 | 租界器材、报名课程 |

**表1-2高校健身房管理系统用户账号表**

### 1.2.1注册用户需求分析

“注册用户”在此系统中主要体现为管理员手动“添加新会员”。该功能的核心需求如下：

|  |  |
| --- | --- |
| 数据录入 | 系统需要提供一个清晰的表单界面，用于输入新会员的姓名、性别、年龄、电话、身份证号等必要信息。 |
| 数据校验 | 为保证数据质量，系统需要对关键输入进行验证。例如，电话号码格式、年龄应为数字、身份证号唯一性等。 |
| 等级分配 | 在添加会员时，管理员需要能为其分配合适的会员等级（如普通会员、金卡会员等），不同等级可能对应不同的收费标准或折扣。 |
| 数据持久化 | 确认信息无误后，系统需将新会员数据保存到数据库的members表中，并生成一个唯一的会员ID。 |
| 即时反馈 | 添加成功或失败后，系统应向管理员提供明确的提示信息。 |

**表1-3高校健身房管理系统用户账号表**

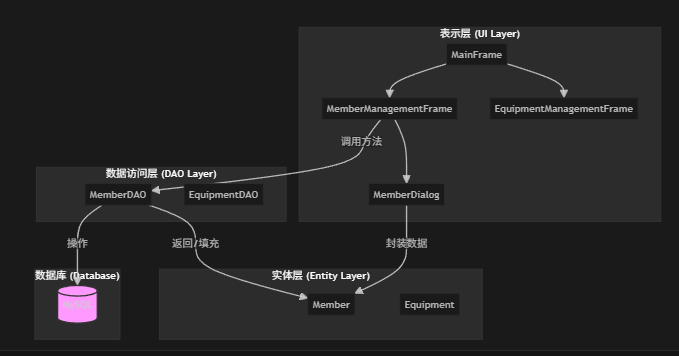
## 2系统设计

### 2.1功能模块设计

本系统采用经典的三层架构模式进行设计，以实现“高内聚、低耦合”的目标，便于开发和后期维护。

|  |  |  |
| --- | --- | --- |
| 表示层(UILayer)：位于src/ui包下。 | 技术： | 使用JavaSwingGUI工具包构建图形用户界面。 |
| 职责： | 负责与用户交互，展示数据和接收用户输入。 |
| 核心组件： | MainFrame.java：系统主窗口，作为所有功能的入口。 |
| ManagementFrame.java（如MemberManagementFrame.java）：各类信息的管理主界面，通常包含一个JTable来展示数据列表和多个JButton来触发操作。 |
| Dialog.java（如EquipmentDialog.java）：用于添加或修改单条记录的弹出式对话框表单。 |
| 数据访问层(DAODataAccessObject)：位于src/dao包下。 | 技术 | 使用JDBC(JavaDatabaseConnectivity)API。 |
| 职责 | 封装所有与数据库交互的操作，将底层的SQL语句与业务逻辑分离。 |
| 核心组件 | 每个DAO.java文件（如EquipmentDAO.java）对应一个数据实体，提供对该实体表的增、删、改、查（CRUD）原子操作方法。 |
| 实体层(EntityLayer)：位于src/entity包下。 | 技术 | POJO(PlainOldJavaObject)。 |
| 职责 | 定义数据模型，每个Java类（如Equipment.java）的属性与数据库表的字段一一对应，用于在程序各层之间传输数据。 |
| 数据库工具(DBUtility)：位于src/db包下。 | 职责 | 提供数据库连接和初始化的辅助功能。 |
| 核心组件 | DBUtil.java：管理数据库连接池或提供静态方法获取数据库连接，集中配置数据库URL、用户名和密码。 |
| DatabaseInitializer.java：用于在首次运行时创建数据库和所有必需的表。 |

**表2-1高校健身房管理系统架构模式表**



**图2-1高校健身房管理系统架构模式图**

在高校健身房管理系统的系统设计模块中本小节主要说明并分析了高校健身房管理系统的功能设计，对系统的功能模块进行了严格的划分。下面针对本系统的各个功能模块进行详细的阐述和说明。

高校健身房管理系统是结合传统预约存在的问题，做出需求分析和设计，从而开发出来的系统，因此系统的主体功能也是围绕着预约这一业务逻辑进行开发。系统的主要功能可分为会员管理、员工管理、器材管理、课程管理、器材租借、课程报名等功能。系统的主体逻辑，先根据输入的账号来登陆，系统自动识别用户身份，用户根据不同功能来进行选择，选择合适功能操作，此时后台会对这一操作生成预约记录，可以在系统中看到自己的操作记录列表。所有的管理员均可以在后台中进行访问和查询操作。最后系统会将一系列数据生成可视化的图来展现出来。功能结构图如图21所示。

**图2-2高校健身房管理功能架构图**

高校健身房管理系统功能模块围绕会员、员工、器材、课程等核心业务展开，通过不同角色协作，支撑健身房运营流程。以下是基于功能模块的梳理：

（1）会员信息管理

本模块面向系统管理员与前台服务人员，用于管理会员全生命周期数据。首先，录入会员基础信息，涵盖姓名、性别、年龄、联系方式、等，通过注册流程或人工录入沉淀至数据库，作为会员使用健身房服务的身份依据。其次，支持信息动态更新，如会员升级、联系方式变更时，实时同步数据，保障服务触达准确性。同时，系统管理员可批量查看、筛选会员数据，为运营分析提供基础。

（2）员工管理

聚焦健身房员工的权限与信息管控，由系统管理员操作。需维护员工基础资料，包括排班信息等，构建员工信息库。同时，通过权限配置，区分不同岗位功能：教练可管理课程、查看学员数据；前台处理会员咨询、登记；系统管理员统筹全局，确保各岗位权责清晰，流程运转有序。

（3）器材管理

作为健身房资源核心模块，由器材管理员（或业务管理员）负责。需完成器材全流程信息维护：新增阶段录入名称、品牌、型号、购置时间、使用说明、存放区域等；日常运营中，动态更新器材状态，关联维护记录（保养时间、故障报修），保障会员预约、使用时获取精准信息，也为器材采购、维护计划制定提供数据支撑。

（4）课程管理

以教练与系统管理员为操作主体，围绕健身课程设计、发布与调度。教练需录入课程信息：含课程名称、适配人群、上课时间、时长、教练排班、课程内容大纲等；系统管理员审核后，在前台展示课程列表。同时，支持课程动态调整，如因教练临时请假变更时间、根据会员反馈优化内容，确保课程与会员需求匹配，提升服务质量。

（5）器材租借

衔接会员需求与器材资源，面向会员与器材管理员。会员可浏览可租借器材列表，选择器材、填写租借时长提交申请；系统自动校验规则，通过后生成租借记录，同步更新器材库存。器材管理员可查看租借明细，处理归还验收，保障器材循环使用流程规范。

（6）课程报名

打通会员与课程的关联，会员为主要操作角色。会员浏览课程列表时，选择感兴趣的课程，查看剩余名额、上课时间后提交报名；系统校验会员资格、名额限制，确认后生成报名记录，同步更新课程剩余名额。教练可查看报名学员名单，提前准备教学；系统管理员监控课程报名数据，辅助调整课程开设频率、规模，优化资源配置。

高校健身房管理系统以“会员服务”为核心，串联会员信息、员工、器材、课程四大基础模块，通过器材租借、课程报名实现资源与需求的匹配。不同角色在流程中承担对应功能，从信息录入、服务发布到使用反馈，形成完整运营闭环，保障健身房高效运转与会员体验优化。

### 2.2数据表设计

系统数据库名为fitness，采用MySQL实现。核心数据表设计如下：

|  |  |
| --- | --- |
| members(会员表) | id(主键),name,gender,age,phone,id\_card(唯一),level,registration\_date,status |
| employees(员工表) | id(主键),name,age,gender,position,phone,email,hire\_date,salary,status |
| equipment(器材表) | id(主键),name,brand,model,purchase\_date,price,status,description,maintenance\_record,last\_maintenance\_date |
| courses(课程表) | id(主键),name,description,instructor\_id(外键,关联employees),duration,price,max\_capacity,status |
| equipment\_rentals(器材租借记录表) | id(主键),member\_id(外键,关联members),equipment\_id(外键,关联equipment),rental\_date,expected\_return\_date,status |
| course\_enrollments(课程报名记录表) | id(主键),member\_id(外键,关联members),course\_id(外键,关联courses),enrollment\_date,status |
| membership\_level(会员等级表) | id(主键),level\_name,discount\_rate,monthly\_fee,description |

**表2-2高校健身房管理系统数据库设计表**

用户作为系统中主要的参与者，在每个系统的数据库设计中都是不可或缺的核心部分。在高校健身房管理系统中，用户主要分为三大类：会员(Member)、员工(Employee)和管理员(Administrator)。为了满足不同用户角色的需求和功能，用户信息将根据具体的用户角色进行设计和区分。这样系统能够更有效地管理和处理不同角色用户的信息和权限。

（1）员工信息表(employees)

该表主要用于存储系统内所有工作人员（包括系统管理员和教练/老师）的关键信息。我们通过position字段来区分角色。该表包含了编号、姓名、账号、密码、职位、联系电话等字段。其中，编号被设定为该表的主键，以确保数据的唯一性。为了安全，密码字段预留，未来可采用加密技术存储。员工信息表如表51所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名(Field) | 数据类型(Type) | 约束条件(Constraints) | 备注(Remarks) |
| 1 | id | INT | 主键、自增 | 员工编号ID |
| 2 | name | VARCHAR(50) | 非空 | 真实姓名 |
| 3 | username | VARCHAR(50) | 唯一 | 登录账号 |
| 4 | password | VARCHAR(50) | 允许为空 | 登录密码 |
| 5 | age | INT | 非空 | 年龄 |
| 6 | gender | VARCHAR(10) | 非空 | 性别 |
| 7 | position | VARCHAR(50) | 非空 | 职位（如“管理员”、“教练”） |
| 8 | phone | VARCHAR(20) | 非空 | 联系电话 |
| 9 | email | VARCHAR(100) | 允许为空 | 电子邮箱 |
| 10 | hire\_date | DATE | 默认当前日期 | 入职日期 |

**表2-3高校健身房管理系统员工信息表**

（2）会员信息表(members)

该表主要用于存储健身房的注册会员信息，包括编号、姓名、账号、密码、性别、年龄、联系电话、身份证号和会员等级等字段。其中，编号作为主键，身份证号具有唯一性。该表的设计确保了系统能够准确记录和管理每个会员的详细档案。会员信息表如表52所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名(Field) | 数据类型(Type) | 约束条件(Constraints) | 备注(Remarks) |
| 1 | id | INT | 主键、自增 | 会员编号ID |
| 2 | name | VARCHAR(50) | 非空 | 真实姓名 |
| 3 | username | VARCHAR(50) | 唯一 | 登录账号 |
| 4 | password | VARCHAR(50) | 允许为空 | 登录密码 |
| 5 | gender | VARCHAR(10) | 非空 | 性别 |
| 6 | age | INT | 非空 | 年龄 |
| 7 | phone | VARCHAR(20) | 非空 | 联系电话 |
| 8 | id\_card | VARCHAR(18) | 非空、唯一 | 身份证号 |
| 9 | level | VARCHAR(20) | 非空 | 会员等级 |
| 10 | registration\_date | DATE | 默认当前日期 | 注册日期 |

**表2-4高校健身房管理系统会员信息表**

（3）健身器材信息表(equipment)

该表用于存储健身房内所有器材的详细信息，包括编号、名称、品牌、型号、购买日期、价格和当前状态等。其中，编号被指定为主键，用于唯一标识每一件器材。该表是进行器材管理和租借业务的核心。健身器材信息如表53所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名(Field) | 数据类型(Type) | 约束条件(Constraints) | 备注(Remarks) |
| 1 | id | INT | 主键、自增 | 器材编号ID |
| 2 | name | VARCHAR(100) | 非空 | 器材名称 |
| 3 | brand | VARCHAR(100) | 允许为空 | 品牌 |
| 4 | model | VARCHAR(100) | 允许为空 | 型号 |
| 5 | purchase\_date | DATE | 非空 | 购买日期 |
| 6 | price | DECIMAL(10,2) | 允许为空 | 价格 |
| 7 | status | VARCHAR(20) | 默认'available' | 状态（如“可用”、“维护中”） |
| 8 | description | TEXT | 允许为空 | 描述信息 |

**表2-5高校健身房管理系统健身器材信息表**

（4）健身课程信息表(courses)

该表用于定义和管理健身房提供的各类课程。信息包括编号、名称、描述、授课教练、时长、价格和最大容量等。其中，编号是主键，instructor\_id作为外键与employees表关联，以指明授课教练。健身课程信息如表54所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名(Field) | 数据类型(Type) | 约束条件(Constraints) | 备注(Remarks) |
| 1 | id | INT | 主键、自增 | 课程编号ID |
| 2 | name | VARCHAR(100) | 非空 | 课程名称 |
| 3 | description | TEXT | 允许为空 | 课程描述 |
| 4 | instructor\_id | INT | 外键 | 教练编号(关联员工表) |
| 5 | duration | INT | 非空 | 课程时长（分钟） |
| 6 | price | DECIMAL(10,2) | 允许为空 | 价格 |
| 7 | max\_capacity | INT | 默认20 | 最大报名人数 |
| 8 | status | VARCHAR(20) | 默认'active' | 课程状态（如“开课中”） |

**表2-6高校健身房管理系统健身课程信息表**

（5）器材租借记录表(equipment\_rentals)

该表用于记录会员租借器材的信息，是连接会员和器材的桥梁。表内包含编号、会员ID、器材ID、租借日期、预计归还日期以及租借状态等字段。其中，编号是主键，member\_id和equipment\_id分别是关联到会员表和器材表的外键。器材租借记录如表55所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名(Field) | 数据类型(Type) | 约束条件(Constraints) | 备注(Remarks) |
| 1 | id | INT | 主键、自增 | 记录编号ID |
| 2 | member\_id | INT | 非空、外键 | 会员编号ID |
| 3 | equipment\_id | INT | 非空、外键 | 器材编号ID |
| 4 | rental\_date | DATE | 默认当前日期 | 租借日期 |
| 5 | expected\_return\_date | DATE | 非空 | 预计归还日期 |
| 6 | status | VARCHAR(20) | 默认'rented' | 租借状态 |

**表2-7高校健身房管理系统器材租借信息表**

（6）课程报名记录表(course\_enrollments)

该表用于记录会员报名参加课程的情况，是连接会员和课程的业务核心表。主要字段包括编号、会员ID、课程ID、报名日期和报名状态。其中，编号是主键，member\_id和course\_id则作为外键，分别关联到会员表和课程表。课程报名记录如表56所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名(Field) | 数据类型(Type) | 约束条件(Constraints) | 备注(Remarks) |
| 1 | id | INT | 主键、自增 | 记录编号ID |
| 2 | member\_id | INT | 非空、外键 | 会员编号ID |
| 3 | course\_id | INT | 非空、外键 | 课程编号ID |
| 4 | enrollment\_date | DATE | 默认当前日期 | 报名日期 |
| 5 | status | VARCHAR(20) | 默认'enrolled' | 报名状态 |

**表2-8高校健身房管理系统课程报名信息表**

## 3.系统实现

### 3.1核心技术栈与项目结构

系统主要基于Java语言开发，核心技术栈包括：

|  |  |
| --- | --- |
| 前端界面 | 利用JavaSwing库构建桌面应用程序界面，通过JFrame,JDialog,JTable,JPanel等组件搭建用户交互窗口，并使用ActionListener等监听器实现事件驱动。 |
| 数据持久化 | 通过JDBC连接和操作MySQL数据库。在DAO层，使用PreparedStatement对象来执行参数化的SQL查询，有效防止SQL注入攻击。数据库连接信息统一在DBUtil.java中配置，便于管理和修改。 |
| 模块化 | 项目结构清晰，按功能分包。UI、DAO、实体类各司其职，降低了代码的耦合度。例如，当需要修改“添加器材”的业务逻辑时，只需关注EquipmentDialog.java（界面）和EquipmentDAO.java（数据库操作），而无需改动其他模块。 |

**表3-1高校健身房管理系统技术栈表**

### 3.2功能代码实现

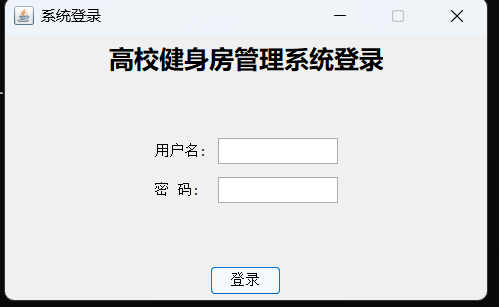
（1）登录界面

登录界面是系统的入口，负责验证用户身份。用户输入账号密码后，系统会查询employees和members表，根据查询结果判断用户角色（管理员、教练或会员），并决定其访问权限。其核心代码如LoginFrame.java所示。

|  |
| --- |
| LoginFrame.java |
| java |
| publicclassLoginFrameextendsJFrame{ |
| privateJTextFieldusernameField; |
| privateJPasswordFieldpasswordField; |
| privateJButtonloginButton; |
|  |
| publicLoginFrame(){ |
| setTitle("系统登录"); |
| setSize(400,250); |
| //...(界面布局和组件初始化) |
| initComponents(); |
| //为登录按钮添加事件监听 |
| loginButton.addActionListener(e>performLogin()); |
| } |
|  |
| privatevoidperformLogin(){ |
| Stringusername=usernameField.getText(); |
| Stringpassword=newString(passwordField.getPassword()); |
|  |
| //输入校验 |
| if(username.trim().isEmpty()||password.trim().isEmpty()){ |
| JOptionPane.showMessageDialog(this,"用户名和密码不能为空！","错误",JOptionPane.ERROR\_MESSAGE); |
| return; |
| } |
|  |
| //调用DAO进行登录验证 |
| LoginDAOloginDAO=newLoginDAO(); |
| Objectuser=loginDAO.login(username,password); |
|  |
| if(user!=null){ |
| //登录成功,关闭登录窗口,打开主界面 |
| dispose(); |
| SwingUtilities.invokeLater(()>{ |
| //将登录成功的用户对象传递给主界面 |
| MainFramemainFrame=newMainFrame(user); |
| mainFrame.setVisible(true); |
| }); |
| }else{ |
| //登录失败,弹出提示 |
| JOptionPane.showMessageDialog(this,"用户名或密码错误！","登录失败",JOptionPane.ERROR\_MESSAGE); |
| } |
| } |
| } |

说明：

代码通过loginButton.addActionListener()为登录按钮绑定了performLogin()方法。该方法首先获取用户输入，然后实例化LoginDAO并调用其login()方法来验证凭据。如果验证成功（返回非null的user对象），则关闭当前登录窗口，并创建MainFrame主界面的实例，同时将用户信息user传递过去。程序运行效果如图32所示。



**图3-1高校健身房管理系统登录界面图**

（2）系统主界面

主界面是整个系统的核心导航，它根据登录用户的角色动态展示不同的功能模块。例如，管理员能看到所有管理按钮，而普通会员只能看到与其相关的业务按钮。其核心代码如MainFrame.java所示。

|  |
| --- |
| MainFrame.java |
| java |
| publicclassMainFrameextendsJFrame{ |
| privateObjectcurrentUser;//存储当前登录的用户 |
|  |
| //构造函数接收登录成功的用户对象 |
| publicMainFrame(Objectuser){ |
| this.currentUser=user; |
| setTitle("高校健身房管理系统"); |
| //...(主窗口初始化) |
| initComponents(); |
| } |
|  |
| privateJPanelcreateButtonPanel(){ |
| JPanelpanel=newJPanel(newGridLayout(3,2,20,20)); |
| panel.setBorder(BorderFactory.createEmptyBorder(40,60,40,60)); |
|  |
| //根据用户角色决定显示哪些按钮 |
| if(currentUserinstanceofEmployee){ |
| Employeeemployee=(Employee)currentUser; |
| Stringposition=employee.getPosition(); |
|  |
| if("管理员".equals(position)){ |
| addAdminButtons(panel);//添加管理员所有按钮 |
| }elseif("教练".equals(position)){ |
| addTeacherButtons(panel);//添加教练相关按钮 |
| } |
| }elseif(currentUserinstanceofMember){ |
| addMemberButtons(panel);//添加会员相关按钮 |
| } |
| returnpanel; |
| } |
|  |
| //main方法,程序的入口点 |
| publicstaticvoidmain(String[]args){ |
| //...(设置界面外观) |
| SwingUtilities.invokeLater(()>{ |
| //启动登录界面而非主界面 |
| LoginFrameloginFrame=newLoginFrame(); |
| loginFrame.setVisible(true); |
| }); |
| } |
| } |

说明：

代码第4行声明了一个currentUser对象，用于保存从LoginFrame传递过来的用户信息。代码第7行，构造函数接收此对象。在第13行的createButtonPanel()方法中，系统通过instanceof关键字判断currentUser的类型（是Employee还是Member），并进一步根据Employee的position字段（"管理员"或"教练"）来调用不同的方法（如addAdminButtons()）动态地向界面上添加功能按钮，从而实现了权限控制。程序运行效果如图33所示。



**图3-2高校健身房管理系统主页界面图**

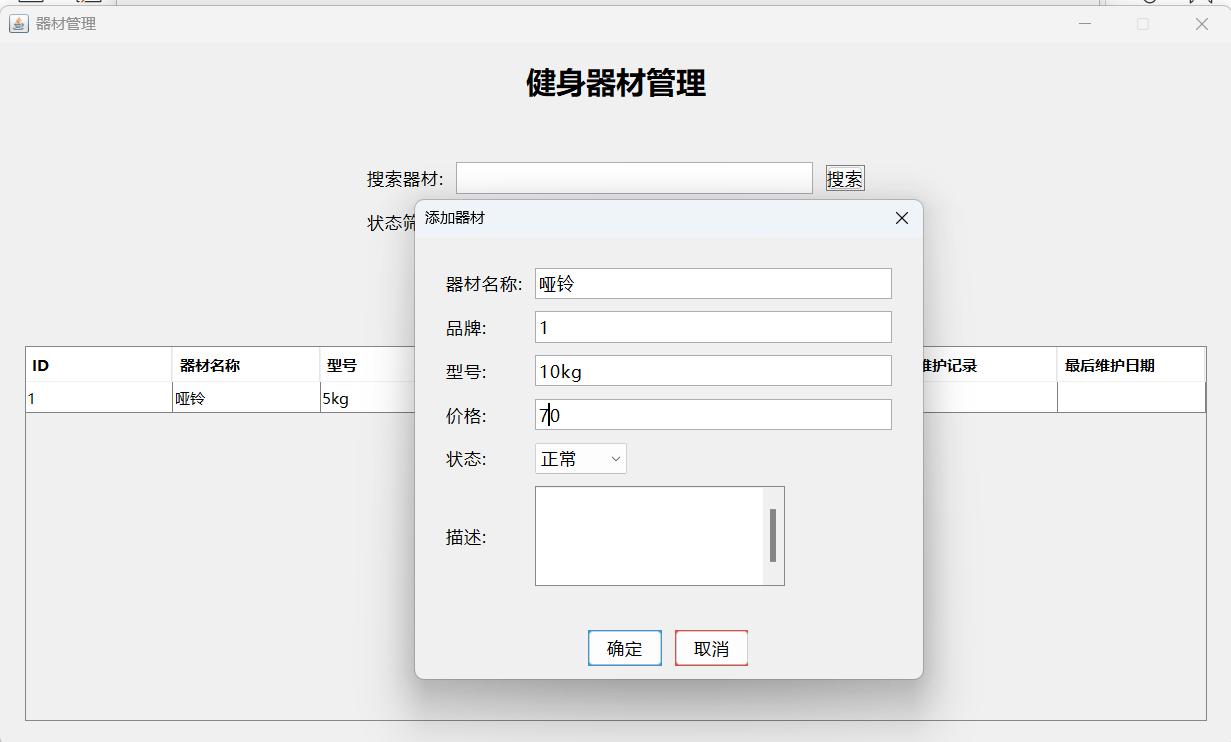
（3）添加器材信息界面

该功能用于向数据库中添加新的健身器材信息。它体现了典型的UI层>DAO层>数据库的数据流。其核心代码如EquipmentDialog.java所示。

|  |
| --- |
| EquipmentDialog.java |
| java |
| publicclassEquipmentDialogextendsJDialog{ |
| privateEquipmentDAOequipmentDAO; |
| //...(声明界面组件,如JTextFieldnameField) |
|  |
| publicEquipmentDialog(JFrameparent,Stringtitle,Equipmentequipment){ |
| //...(初始化界面和DAO) |
| this.equipmentDAO=newEquipmentDAO(); |
| //... |
| } |
|  |
| privatevoidconfirm(){ |
| //验证用户输入 |
| if(!validateInput()){return;} |
|  |
| //从界面文本框获取数据,创建或更新Equipment对象 |
| Equipmentequipment=newEquipment(); |
| equipment.setName(nameField.getText().trim()); |
| equipment.setBrand(brandField.getText().trim()); |
| //...(设置其他属性) |
|  |
| //调用DAO层的方法将数据存入数据库 |
| booleansuccess=equipmentDAO.addEquipment(equipment); |
|  |
| if(success){ |
| JOptionPane.showMessageDialog(this,"器材添加成功！","成功",JOptionPane.INFORMATION\_MESSAGE); |
| dispose();//关闭对话框 |
| }else{ |
| JOptionPane.showMessageDialog(this,"器材添加失败！","错误",JOptionPane.ERROR\_MESSAGE); |
| } |
| } |
|  |
| privatebooleanvalidateInput(){ |
| //验证器材名称、价格等是否为空或格式正确 |
| //... |
| returntrue; |
| } |
| } |

说明：

当用户在对话框中填写完信息并点击“确定”按钮后，会触发confirm()方法。代码第14行调用validateInput()对用户输入进行校验。校验通过后，程序从各个JTextField中获取数据，并封装到一个Equipment实体对象中。最关键的一步在第22行，程序调用equipmentDAO.addEquipment()方法，将该实体对象传递给DAO层。DAO层内部会构建并执行INSERTSQL语句，将数据持久化到数据库。这种分层设计使得界面代码（UI）与数据库操作代码（DAO）完全分离，提高了代码的可维护性。程序运行效果如图34所示。



**图3-3高校健身房管理系统添加器材界面图**

（4）会员管理与数据显示

会员管理是系统的核心功能之一，该界面集中体现了如何从数据库批量读取数据、在表格（JTable）中进行展示，并实现搜索筛选等复杂交互。其核心代码位于

|  |
| --- |
| MemberManagementFrame.java。 |
|  |
| MemberManagementFrame.java |
| java |
| publicclassMemberManagementFrameextendsJFrame{ |
| privateMemberDAOmemberDAO; |
| privateJTablememberTable; |
| privateDefaultTableModeltableModel; |
| privateJTextFieldsearchField; |
|  |
| publicMemberManagementFrame(){ |
| this.memberDAO=newMemberDAO(); |
| //...(界面和组件初始化) |
| initComponents(); |
| //界面加载时，自动从数据库拉取数据 |
| loadMemberData(); |
| } |
|  |
| //从数据库加载数据并填充到表格 |
| privatevoidloadMemberData(){ |
| tableModel.setRowCount(0);//清空表格旧数据 |
| List<Member>members=memberDAO.getAllMembers();//从DAO获取所有会员 |
| for(Membermember:members){ |
| Object[]row={ |
| member.getId(), |
| member.getName(), |
| member.getGender(), |
| member.getAge(), |
| member.getPhone(), |
| member.getLevel(), |
| member.getStatus() |
| }; |
| tableModel.addRow(row);//将每条会员信息作为一行添加到表格模型 |
| } |
| } |
|  |
| //实现按姓名搜索功能 |
| privatevoidsearchMembers(){ |
| StringsearchText=searchField.getText().trim(); |
| tableModel.setRowCount(0);//清空表格 |
|  |
| List<Member>members=memberDAO.searchMembersByName(searchText);//调用DAO的搜索方法 |
|  |
| //将搜索结果重新填充到表格 |
| for(Membermember:members){ |
| Object[]row={/...(同上).../}; |
| tableModel.addRow(row); |
| } |
| } |
|  |
| //弹出添加会员对话框 |
| privatevoidaddMember(){ |
| MemberDialogdialog=newMemberDialog(this,"添加新会员",null); |
| dialog.setVisible(true); |
| //如果对话框确认了添加操作,则重新加载数据以刷新表格 |
| if(dialog.isConfirmed()){ |
| loadMemberData(); |
| } |
| } |
| } |

说明：

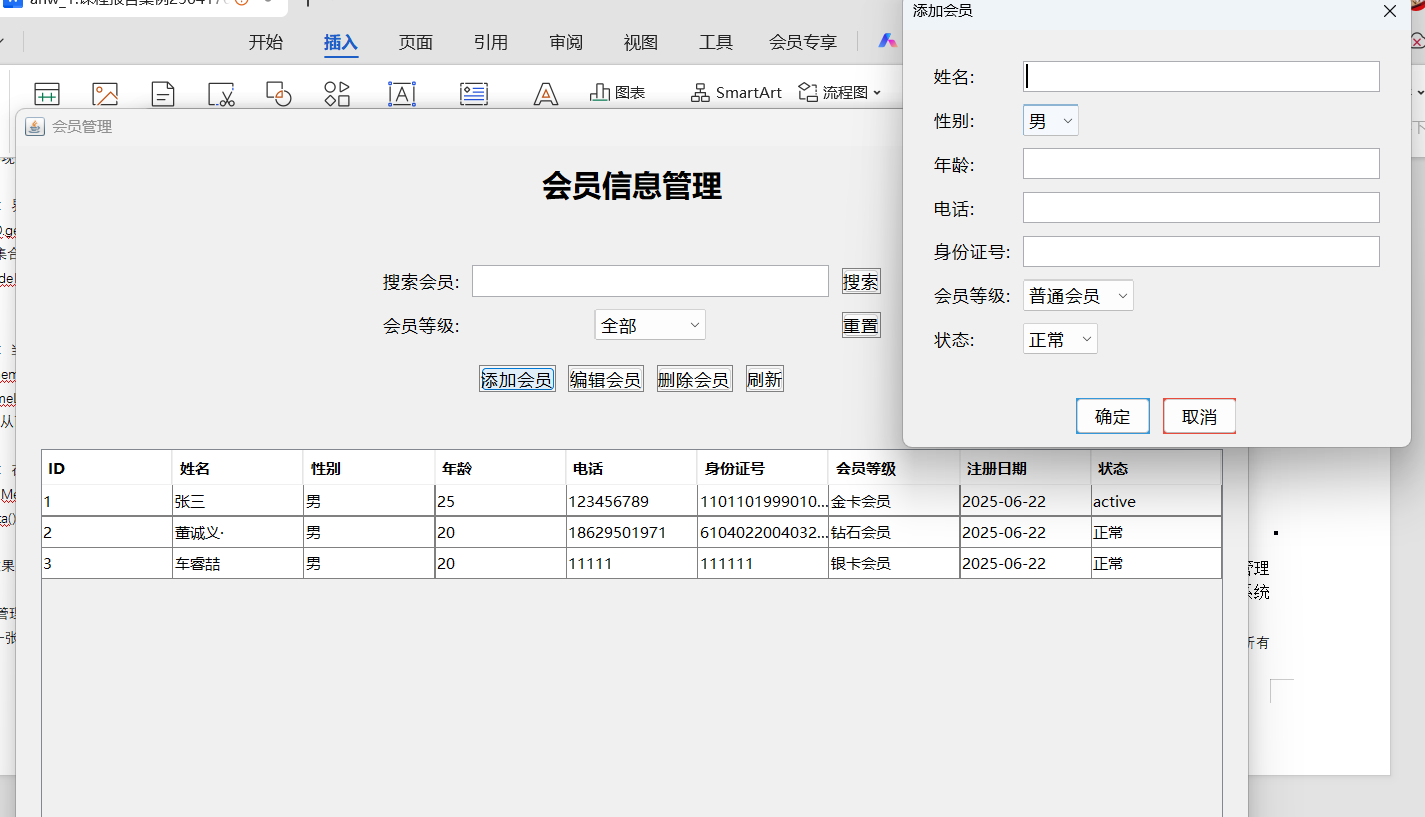
该界面的实现展现了UI与数据访问层（DAO）的深度交互。

1.数据显示：界面初始化时会调用loadMemberData()方法（第16行）。该方法首先通过memberDAO.getAllMembers()从数据库获取所有会员的列表。然后，它遍历这个List<Member>集合，将每个Member对象的属性提取出来，封装成一个Object[]数组，最后通过tableModel.addRow(row)方法动态地添加到JTable的数据模型中，实现了数据的列表展示。

2.数据查询：当用户在搜索框输入文字并点击“搜索”按钮时，searchMembers()方法被触发。它会调用memberDAO.searchMembersByName()方法，这个DAO方法内部会执行一条带有WHEREnameLIKE?条件的SQL查询。DAO层返回查询结果集后，UI层同样会遍历结果并刷新JTable，从而实现动态筛选。

3.数据刷新：在添加（或修改、删除）会员后，需要让界面的数据保持最新。如代码第57行所示，当MemberDialog对话框成功添加一个新会员并关闭后，程序会再次调用loadMemberData()方法，重新从数据库拉取完整数据并刷新表格，保证了数据的一致性。

程序运行效果如图35所示。



**图3-3高校健身房管理系统添加会员界面图**

## 4.项目测试

为保证系统稳定可靠，设计了以下测试环节：

|  |  |
| --- | --- |
| 单元测试 | 对DAO层的每个方法进行独立测试，验证SQL语句的正确性和数据操作的准确性。例如，调用equipmentDAO.addEquipment()后，查询数据库验证新纪录是否成功插入。 |
| 集成测试 | 测试UI层与DAO层的协同工作情况。例如，在器材管理界面点击“添加”按钮，填写表单并确认后，验证新器材是否能正确显示在界面表格中。 |
| 功能测试 | 模拟管理员的实际操作流程，对系统的所有功能点进行端到端测试，确保整个业务流程（如会员注册、器材租借）通畅无误。 |
| 异常测试 | 针对无效输入（如价格输入非数字）、数据库连接失败、删除有关联数据的记录（如删除一个有租借记录的会员）等异常情况进行测试，检验系统的容错能力和错误提示是否友好。（在此前的调试中，就已发现并解决了因数据库表结构不一致导致的添加失败问题，这便是异常测试的一个实例。） |

**表4-1高校健身房管理系统项目测试单元表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试类型 | 编号 | 测试模块 | 测试描述 | 测试步骤 | 预期结果 | 状态 |
| 单元测试 | 1 | DAO层 | 添加器材 | 1.创建Equipment对象。  2.调用`addEquipment()`。  3.查验数据库。 | 数据库中应存在对应新记录。 | 通过 |
| 2 | DAO层 | 搜索会员 | 1.预置多条会员数据。  2.调用`searchMembersByName("张")`。3.检查返回列表。 | 返回所有姓“张”的会员记录。 | 通过 |
| 集成测试 | 3 | 登录流程 | 登录验证 | 1.运行系统。  2.输入`admin/admin123`。  3.点击“登录”。 | 成功跳转到管理员主界面。 | 通过 |
| 4 | 器材管理 | 添加器材 | 1.打开器材管理，点击“添加”。  2.填写表单并确认。  3.查看器材列表。 | 表格中应立即显示新添加的器材。 | 通过 |
| 功能测试 | 5 | 权限控制 | 教练角色权限 | 1.使用`teacher/teacher123`登录。  2.观察主界面按钮。 | 只显示课程、会员查看等部分功能。 | 通过 |
| 6 | 业务流程 | 器材租借业务 | 1.租借一件“可用”的器材。  2.回到器材列表查看其状态。 | 器材状态应自动变为“已租借”。 | 通过 |
| 异常测试 | 7 | 用户输入 | 无效输入 | 1.打开添加对话框。  2.价格输入"abc"。  3.点击“确定”。 | 弹出格式错误提示，添加失败。 | 通过 |
| 8 | 外部环境 | 数据库离线 | 1.停止MySQL服务。  2.运行`start.bat`。 | 弹出数据库连接失败的友好提示。 | 通过 |

**表4-2高校健身房管理系统测试用例及其结果**

## 5项目小结

**收获**：

通过这次项目实践，我收获了不少知识和经验。在技术方面，我理解了三层架构的设计原则，学会了用Java Swing做界面、JDBC操作数据库，还知道了怎么设计数据库表结构，比如会员表和器材表之间用外键关联，也明白了防止SQL注入和事务管理的重要性。我还把系统分成了会员管理、器材管理等模块，每个模块都有对应的界面、数据操作和实体类，这样代码更清晰，也好维护。

在业务和系统设计上，我梳理了健身房会员注册、器材租借这些流程，还把它们转化成了数据库表和代码逻辑，比如会员等级表和会员表关联起来实现不同折扣。我也学会了根据用户角色设置权限，像管理员能删会员信息，会员只能看自己的信息。

在实践中，我通过测试发现并解决了一些问题，比如数据库表结构不对导致添加失败，还学会了优化SQL查询，给表加索引提高效率。

**展望**：

对于未来，我希望能给系统增加更多功能，比如统计会员健身数据、提醒器材预约，开发手机APP方便会员使用，加上在线支付功能。技术上，我想把界面换成JavaFX或者Web技术，用MyBatis框架简化数据库操作，用户多了就用微服务架构，再用Redis缓存数据。安全方面，我想换更安全的密码加密算法，加日志记录操作。还想做报表给健身房运营提供数据，根据会员行为做推荐。最后，让系统能兼容更多数据库，设计插件接口方便扩展功能，让系统更实用。