
忻州职业技术学院计算机系
毕业论文

学生信息管理系统

姓 名

指导教师

专 业 计算机应用技术

班 级 18 计高 3

设计方向 应用系统设计

二〇二一年三月

摘 要

进行学生信息管理是学校的重要工作之一。传统的学生信息管理由于是手工操作，工作量大且容易出错。随着计算机和网络技术的迅速发展，越来越多的院校都拥有了自己的学生信息管理系统，而采用 C/S, B/S 架构的学生信息管理系统进行学生信息的管理则是其趋势。

Java 技术的基础就是核心 Java 平台，它有许多优点，例如：“编写一次、随处运行”的特性、方便存取数据库的 JDBC API 技术以及能够在 Internet 应用中保护数据的安全模式等等。面向对象的体系结构提供丰富的工具类来满足高可用性、高可靠性以及可扩展性的应用需求，为搭建具有可伸缩性、灵活性、易维护性的应用系统提供了良好的机制。

本文基于 Java 技术、采用 Client/Server 结构, 设计并实现了该系统的简化版学生信息管理系统。

论文首先介绍了系统开发涉及的相关技术及开发平台，主要围绕学生信息管理系统实现的相关技术和系统的体系结构，介绍了学生信息管理系统的总体分析与设计，包括需求分析、功能模块分析和数据库设计。最后，利用 Java (JavaFX) +MySQL 模式对简单学生信息管理子系统进行了实现，并作了详细描述，主要采用图片与代码结合的方式详细地叙述了系统功能的实现。

关键词：MySQL 8.0; Docker 20.10.3; Navicat 15 for MySQL; CentOS 8; JavaFX 11.0; Java JDK 15.0

目录

| | |
|----------------------------|----|
| 一 绪论..... | 1 |
| 1.1 课题背景及现状..... | 1 |
| 1.2 开发环境及工具介绍..... | 1 |
| 1.2.1 操作系统及配置..... | 1 |
| 1.2.2 编译环境..... | 1 |
| 1.3 编程语言..... | 3 |
| 二 需求分析..... | 3 |
| 2.1 面向人群..... | 3 |
| 2.2 总体功能描述..... | 3 |
| 2.3 功能描述..... | 3 |
| 2.3.1 普通用户登录实现功能..... | 3 |
| 2.3.2 管理员登录实现功能..... | 3 |
| 三 总体设计..... | 3 |
| 3.1 运行环境..... | 3 |
| 3.2 基本处理流程..... | 4 |
| 3.2.1 用户登录流程图..... | 4 |
| 3.2.2 子模块操作处理流程图..... | 5 |
| 3.3 系统结构..... | 5 |
| 3.3.1 系统结构..... | 5 |
| 3.3.2 功能模块介绍..... | 6 |
| 四 详细设计..... | 6 |
| 4.1 子模块设计..... | 6 |
| 4.1.1 图形用户界面（GUI）模块设计..... | 6 |
| 4.1.2 程序描述..... | 7 |
| 4.2 关键类说明..... | 7 |
| 五 服务器端..... | 8 |
| 5.1 硬件环境..... | 8 |
| 5.2 CentOS 具体设置..... | 8 |
| 六 数据库设计..... | 9 |
| 6.1 数据库设计..... | 9 |
| 6.2 逻辑结构设计..... | 10 |
| 6.3 物理结构设计..... | 10 |
| 七 源码与测试..... | 11 |
| 7.1 操作界面..... | 11 |
| 7.2 关键代码..... | 21 |
| 7.2.1 用户交互（GUI）界面..... | 22 |
| 7.2.2 工具类..... | 22 |
| 八 实验总结..... | 25 |

一 绪论

1.1 课题背景及现状

随着信息技术在学生信息管理上的广泛应用，学生信息管理系统的实施在技术上已逐步成熟。学生信息的管理对于学校管理者来说至关重要，是学校不可缺少的数据资源。特别是近几年，学生数量的增多使学校的管理面临更大压力和挑战。目前大多数学校仍然采用传统的人工管理方式，对学生信息的管理还停留在纸介质的基础上，这既不易于规范化，管理效率也不高，浪费了大量人力物力财力。

作为计算机应用的一部分，使用计算机进行学生信息的管理，有着手工管理无法比拟的优点。本系统针对学校及学生的信息特点，可以对学生信息进行高效率、规范化的管理，大大减轻了人员的工作负担，并避免了人工管理时容易出现的错误。

1.2 开发环境及工具介绍

1.2.1 操作系统及配置

Windows10 专业版 X64，Intel(R) Core(TM) i5-8300H，16.0 GB，CentOS-8.3.2011

1.2.2 编译环境

1.2.2.1 Eclipse IDE for Java Developers (Version: 2020-12 (4.18.0))

Eclipse 是著名的跨平台的自由集成开发环境 (IDE)。最初主要用来 Java 语言开发，通过安装不同的插件 Eclipse 可以支持不同的计算机语言，比如 C++ 和 Python 等开发工具。Eclipse 的本身只是一个框架平台，但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE。

1.2.2.2 VMware® Workstation 16 Pro (Version: 16.1.0 build-17198959)

VMware Workstation (中文名“威睿工作站”)是一款功能强大的桌面虚拟计算机软件,提供用户可在单一的桌面上同时运行不同的操作系统,和进行开发、测试、部署新的应用程序的最佳解决方案。VMware Workstation 可在一部实体机器上模拟完整的网络环境,以及可便于携带的虚拟机器,其更好的灵活性与先进的技术胜过了市面上其他的虚拟计算机软件。对于企业的 IT 开发人员和系统管理员而言, VMware 在虚拟网路,实时快照,拖曳共享文件夹,支持 PXE 等方面的特点使它成为必不可少的工具。

1.2.2.3 Navicat 15 for MySQL

Navicat for MySQL 是一套管理和开发 MySQL 或 MariaDB 的理想解决方案,支持单一程序,可同时连接到 MySQL 和 MariaDB。这个功能齐备的前端软件为数据库管理、开发和维护提供了直观而强大的图形界面,给 MySQL 或 MariaDB 新手以及专业人士提供了一组全面的工具。

1.2.2.4 JavaFX11.0

JavaFX 是一个开源的下一代客户端应用平台,适用于基于 Java 构建的桌面、移动端和嵌入式系统。它是许多个人和公司的共同努力的成果,目的是为开发丰富的客户端应用提供一个现代、高效、功能齐全的工具包。

1.2.2.5 CentOS 8

CentOS 是免费的、开源的、可以重新分发的开源操作系统 [1], CentOS (Community Enterprise Operating System, 中文意思是社区企业操作系统) 是 Linux 发行版之一。

1.2.2.6 MySQL 8.0

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS（Relational Database Management System，关系数据库管理系统）应用软件之一。

1.3 编程语言

Java，SQL。

二 需求分析

2.1 面向人群

学校的学生信息管理人员以及教师。

2.2 总体功能描述

本系统利用 Java 技术实现了学生信息管理系统，具有简单的学生信息管理功能。

实现了以下功能模块：

学生信息管理模块，并能根据登入用户的权限自动展示相关操作。

2.3 功能描述

2.3.1 普通用户登录实现功能

查看所有学生的部分不重要信息。

2.3.2 管理员登录实现功能

对用户进行增删改查，用页面返回处理以及查询结果。

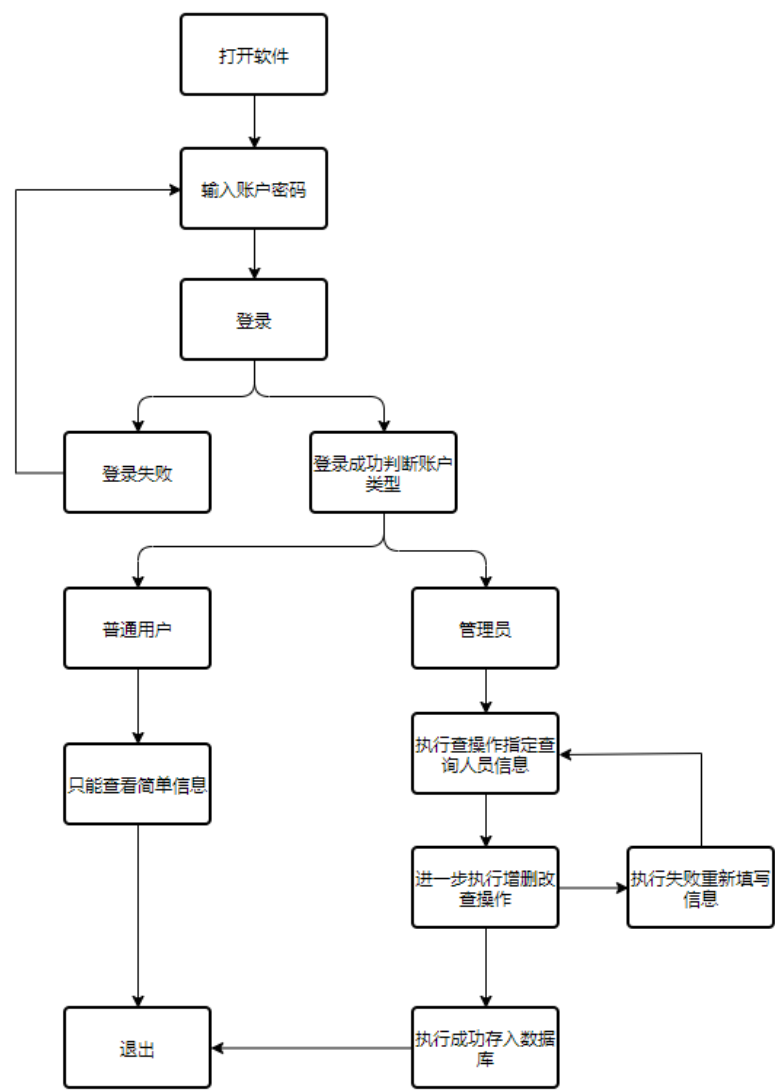
三 总体设计

3.1 运行环境

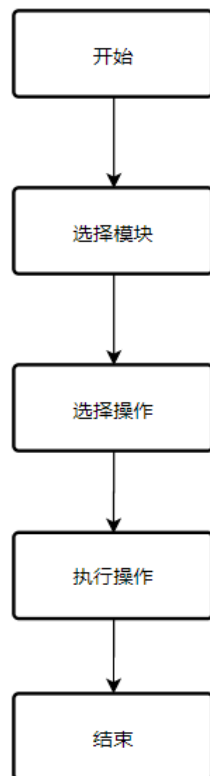
硬件环境：Windows 7 及以上，处理器：Intel Pentium 及以上，内存：2G 以上。

3.2 基本处理流程

3.2.1 用户登录流程图

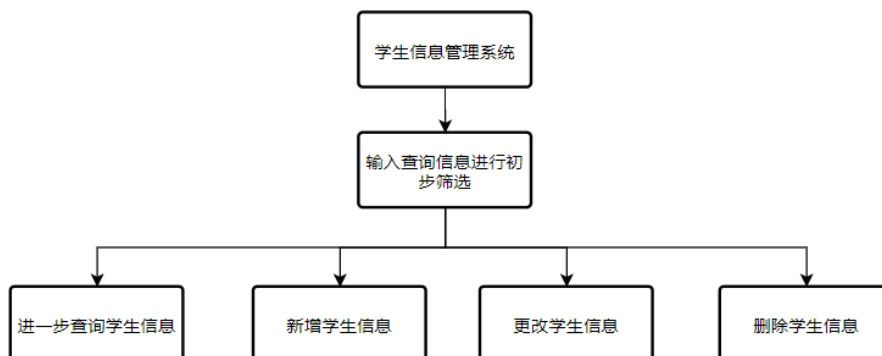


3.2.2 子模块操作处理流程图



3.3 系统结构

3.3.1 系统结构



3.3.2 功能模块介绍

初步查询（筛选）：精确输入学号查询，模糊输入姓名查询，学号姓名同时都有值时，优先采用学号查询。

进一步查询信息：根据初步查询确定的学号信息进行进一步的详细信息查询

新增学生信息：依据数据库学号排序新增学生信息。

更改学生信息：根据初步查询确定的学号信息，更改学生信息。

删除学生信息：根据初步查询确定的学号信息，删除学生信息。

四 详细设计

4.1 子模块设计

4.1.1 图形用户界面（GUI）模块设计

采用 JavaFX 工具包进行设计，用户在前台（GUI）操作，向编写的工具类 c 参发送到服务器数据库中。



4.1.2 程序描述

用户登录程序，建立程序到数据库的连接，进行初步查询确定学号信息，进一步进行学生信息的增删改查操纵。

4.2 关键类说明

Application 包：

Delete.Java：删除学生信息界面

Detailedselect.Java：详细学生信息界面

Insert.Java：新增学生信息界面

Login.Java：登录界面

Update.Java：更新学生信息界面

Window.Java：初步筛选界面

JDBC 包：

Loginutils.Java: 登录建立服务器数据库连接类

Newdelete.Java: 删除学生信息类

Newdetailedselect.Java: 查询学生详细信息类

Newinsert.Java: 新增学生信息类

Newselect.Java: 初步筛选学生信息类

Newupdate.Java: 更新学生信息类

Privilegeutils.Java: 查询登录用户权限类

Selectsno.Java: 新增学生信息确认学号类

Icon 包:

Icon.png 程序 Logo

mix.png 最小化按钮图

quit.png 退出按钮图

五 服务器端

5.1 硬件环境

使用 VMware Workstation 模拟硬件条件:

内存 8GB; 处理器: 内核 8 颗 (4 处理器数量 X2 每个处理器内核); 硬盘: 150GB; 网络适配器: NAT 模式; ISO 文件: CentOS-8.3.2011-x86_64-dvd1_3.iso。

5.2 CentOS 具体设置

Linux 内核版本: 4.18.0。

由于 docker 拉取官方 MySQL8.0 版本的地址在中国大陆并没有服务器, 所以采用了阿里云的 docker 加速器 (加速地址)。

```
{  
  "registry-mirrors": ["https://amxi8109.mirror.aliyuncs.com"]  
}
```

使用 docker 的 3306 端口映射到主机的 3306 端口, 开放主机防火墙的 3306 端口即可与服务器 MySQL 数据库建立连接。

六 数据库设计

6.1 数据库设计

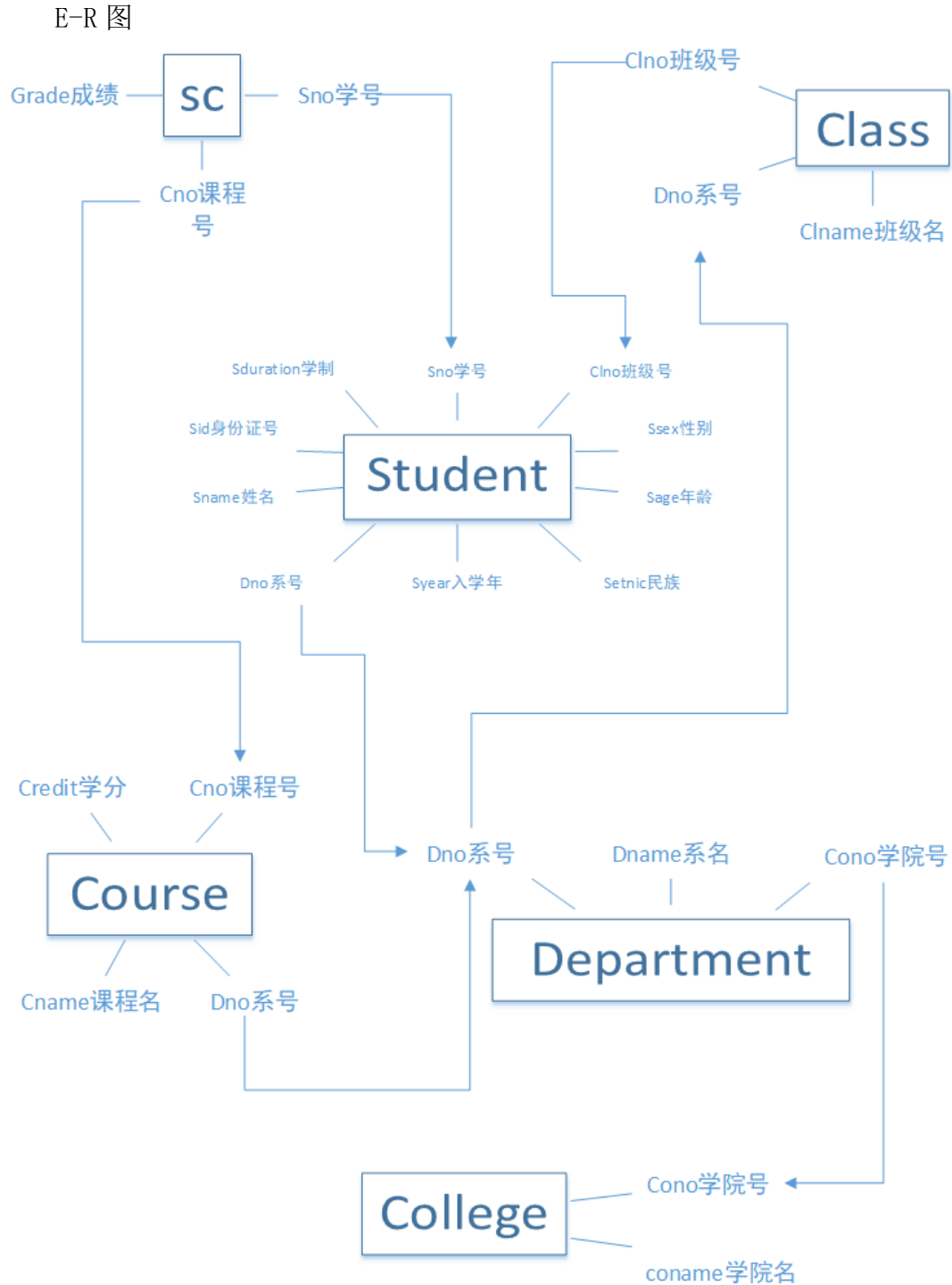
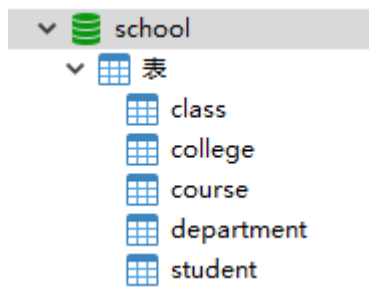


图 6.1 E-R 图

school 数据库中拥有五张数据表：Department（院系），Course（课程），College（学院），Class（班级），Student（学生）。



各个数据表之间的联系如 5.1 E-R 图所示。

6.2 逻辑结构设计

表关系模式（_为主键，#为外键）

Class(Clno, #Dno)

college (cono)

course(cno, #dno)

department(dno, #cono)

student(sno, sid, #clno, #dno)

6.3 物理结构设计

| 表名 | 功能说明 |
|------------|------|
| Department | 院系表 |
| Course | 课程表 |
| College | 学院表 |
| Class | 班级表 |
| Student | 学生表 |

College 表（学院表）

| 列名 | 说明 | 数据类型（精度范围） | 空/非空 | 约束条件 |
|--------|-----|------------|------|------|
| Cono | 学院号 | Char(12) | 非空 | |
| Coname | 学院名 | Char(12) | 非空 | |

Department 表（系表）

| 列名 | 说明 | 数据类型（精度范围） | 空/非空 | 约束条件 |
|-------|-----|------------|------|------|
| Dno | 系号 | Char(12) | 非空 | |
| Dname | 系名 | Char(12) | 非空 | |
| Cono | 学院号 | Char(12) | 非空 | |

Class 表（班级表）

| 列名 | 说明 | 数据类型（精度范围） | 空/非空 | 约束条件 |
|--------|-----|------------|------|------|
| Clno | 班级号 | Char(12) | 非空 | |
| Clname | 班级名 | Char(12) | 非空 | |

| | | | | |
|-----|----|----------|----|--|
| Dno | 系号 | Char(12) | 非空 | |
|-----|----|----------|----|--|

student 表（学生表）

| 列名 | 说明 | 数据类型（精度范围） | 空/非空 | 约束条件 |
|-----------|-----|-------------|------|----------|
| Sno | 学号 | Char(12) | 非空 | |
| Sname | 姓名 | Char(8) | 非空 | |
| Ssex | 性别 | Char(2) | 非空 | ‘男’or‘女’ |
| Sage | 年龄 | Smallint(6) | 非空 | >0 |
| Setnic | 名族 | Char(12) | 非空 | |
| Syear | 入学年 | Char(10) | 非空 | |
| Dno | 系号 | Char(10) | 非空 | |
| Sid | 身份证 | Char(18) | 非空 | |
| Clno | 班级号 | Char(12) | 非空 | |
| Sduration | 学制 | Char(1) | 非空 | |

Course 表（课程表）

| 列名 | 说明 | 数据类型（精度范围） | 空/非空 | 约束条件 |
|--------|-----|------------|------|-------|
| Cno | 课程号 | Char(12) | 非空 | |
| Cname | 课程名 | Char(12) | 非空 | |
| Credit | 学分 | int(6) | 非空 | 0~100 |
| Dno | 系号 | Char(12) | 非空 | |

七 源码与测试

7.1 操作界面

打开软件登录界面

— ×

学生信息
管理系统

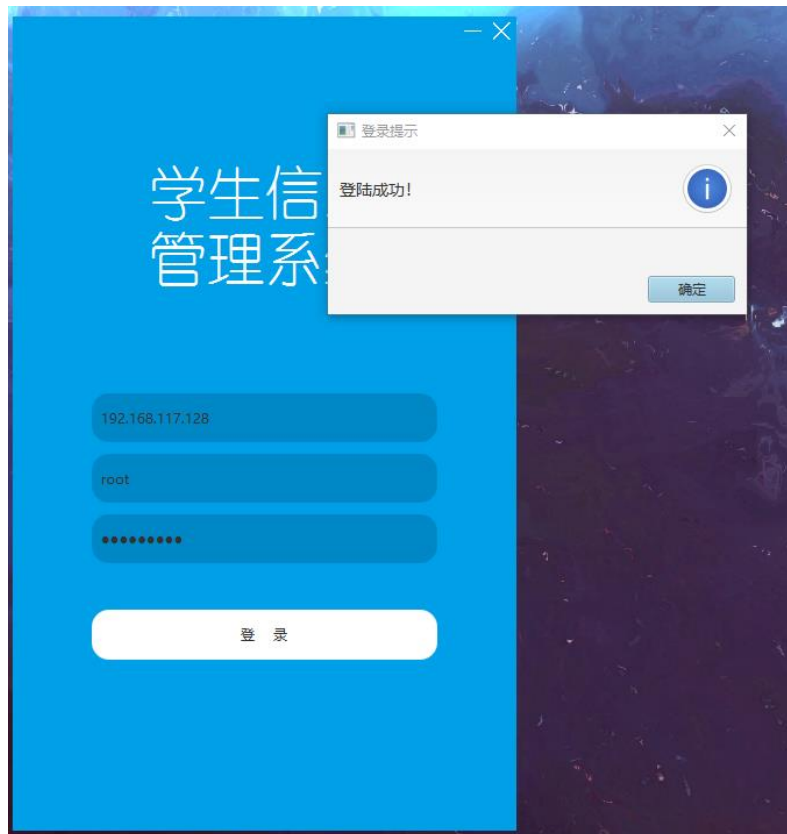
请输入信息系统服务器地址

请输入用户名

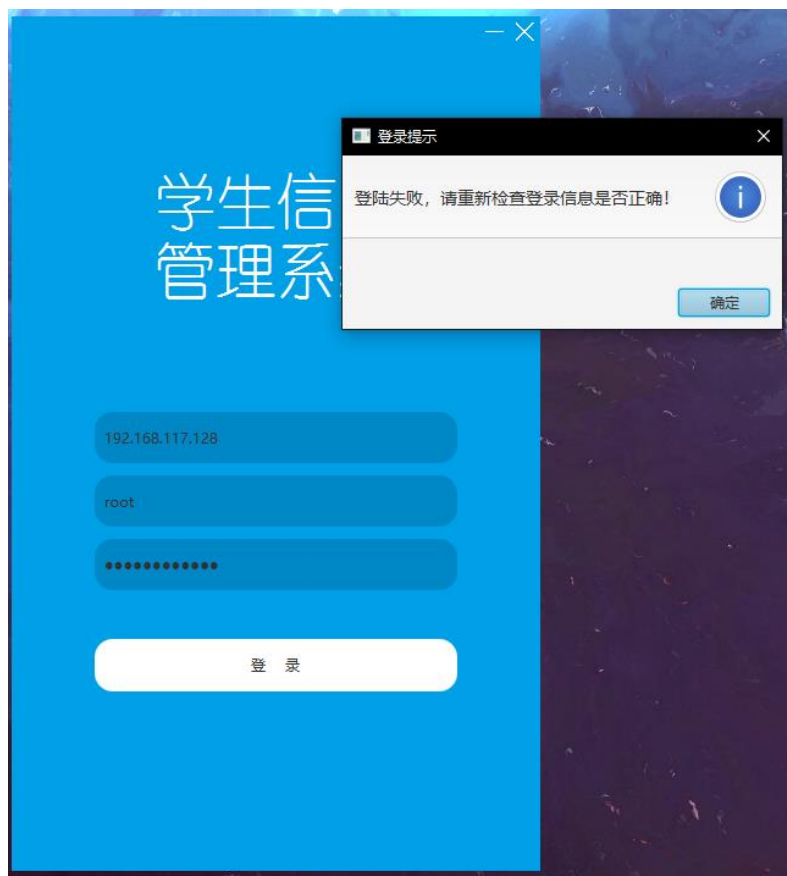
请输入密码

登 录

登录成功界面



登录失败



退出选项



筛选信息界面

管理员登录

— ×

学号

姓名

查询

表中无列

详细查询

增加信息

删除信息

更改信息

普通用户登录

— ×

学号

填写学生学号!

姓名

填写学生姓名!

查询

表中无列

筛选信息

按照学号

— ×

学号20170001

姓名填写学生姓名!

查询

| 学号 | 姓名 | 性别 | 年龄 | 班级 | |
|----------|-----|----|----|----------|--|
| 20170001 | 马保国 | 男 | 18 | 电子科学与技术1 | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

详细查询

增加信息

删除信息

更改信息

按照姓名（有重复）

[illegible]

学号姓名都有信息时，优先按学号查找

[查询详细信息](#)

学号

20170010

姓名

卢本伟

查询

| 学号 | 姓名 | 性别 | 年龄 | 班级 |
|----------|-----|----|----|----------|
| 20170010 | 卢本伟 | 女 | 25 | 电子科学与技术1 |

学号

20170010

姓名

卢本伟

性别

女

年龄

25

民族

汉族

入学年份

2017

身份证号

63211319160506224X

学制

3

所在班级

电子科学与技术1

所在系

电子科学与技术

所在学院

信息与计算机学院

详细查询

增加信息

删除信息

更改信息

修改学生信息

学号

20170010

姓名

卢本伟

查询

| 学号 | 姓名 | 性别 | 年龄 | 民族 | 入学年份 | 系号 | 身份证号 | 班级 | 学制 |
|----------|---------------|----|----|----|------|----|--------------------|----|----|
| 20170010 | <div>修改</div> | | 25 | 汉族 | 2017 | | 63211319160506224X | | 3 |

提交

取消

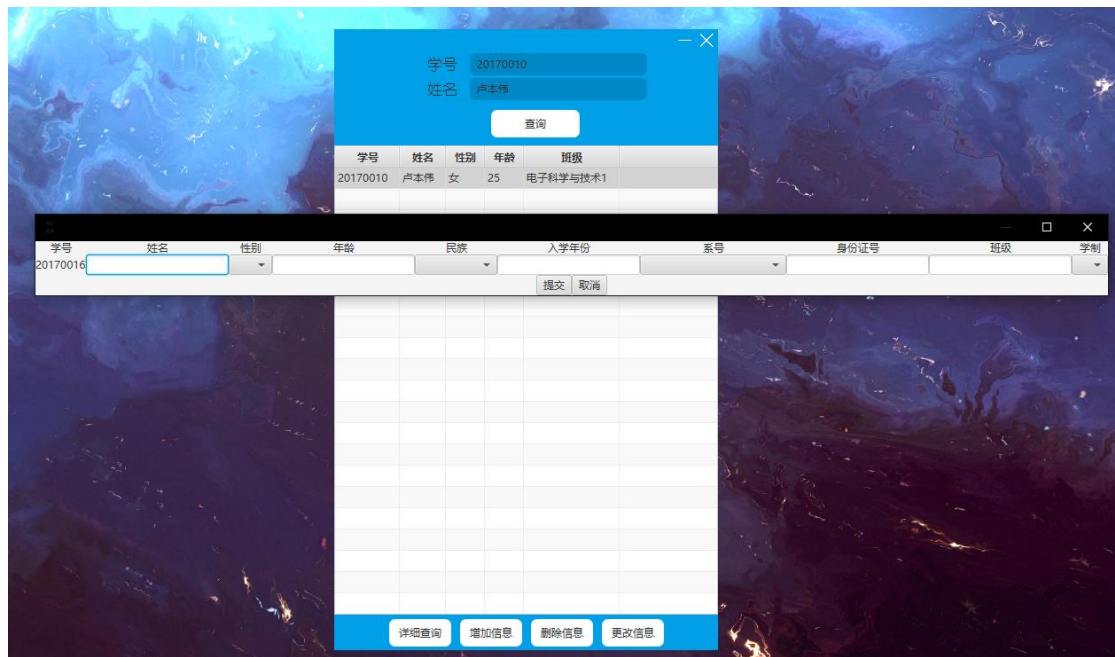
详细查询

增加信息

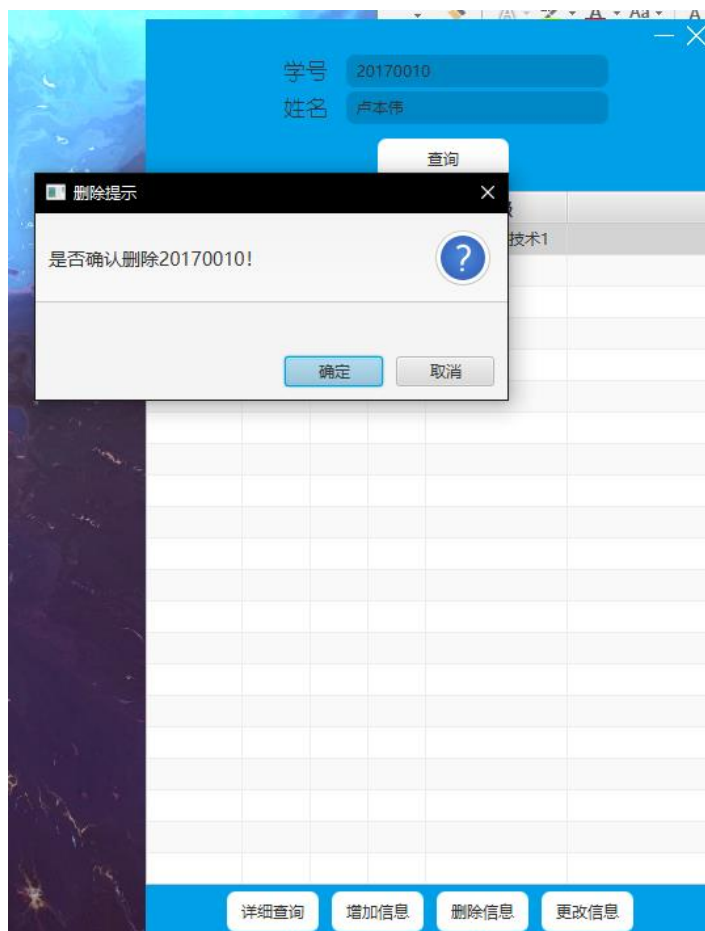
删除信息

更改信息

添加学生信息



删除学生信息



7.2 关键代码

7.2.1 用户交互（GUI）界面

使用 JavaFX 工具包设计的用户界面，具体代码见附件。

7.2.2 工具类

登录

```
public Connection getLogin(String ip,String user,String pass) {

    try {

        String driver = "com.mysql.cj.jdbc.Driver";
        Class.forName(driver);
        this.user = user;
        this.pass = pass;
        this.ip = ip;
        String url = "jdbc:mysql://" + this.ip + "/school?"
            +
            "useSSL=false&useUnicode=true&characterEncoding=utf-8&serverTimezone=
            Asia/Shanghai";

        conn = DriverManager.getConnection(url, user, pass);

        islogin = true;
    }
    catch (Exception e) {
        islogin = false;
    }
    return conn;
}
```

筛选

使用 PreparedStatement 防止 SQL 注入导致异常。

```
public class Newdelete {  
    private int n;  
    private PreparedStatement stat = null;  
    private ResultSet a = null;  
    private String sql = "DELETE FROM student WHERE sno = TRIM(?)";  
    private Connection conn = Loginutils.conn;  
    public boolean isdelete = false;  
  
    public Newdelete(String snoo) {  
        try {  
            stat = conn.prepareStatement(sql);  
            stat.setString(1, snoo);  
            n = stat.executeUpdate();  
            if(n>0) {  
                isdelete = true;  
            }}catch (SQLException e) {  
                // TODO 自动生成的 catch 块  
                e.printStackTrace();  
            }  
        }  
  
        public boolean isdelete() {  
            return isdelete;  
        }  
    }  
}
```

输出列表

使用可观察的 ObservableList 集合中使用有序 HashMap 中定义关键字对应值。

```
private ObservableList<HashMap<String, SimpleStringProperty>> list =
FXCollections.observableArrayList();

public ObservableList<HashMap<String, SimpleStringProperty>> out()
{
    try {
        a.previous();
        while(a.next()) {
            HashMap<String, SimpleStringProperty> map=new
HashMap<String, SimpleStringProperty>();
            map.put("sno", new
SimpleStringProperty(a.getString("sno")));
            map.put("sname", new
SimpleStringProperty(a.getString("sname")));
            map.put("ssex", new
SimpleStringProperty(a.getString("ssex")));
            map.put("sage", new
SimpleStringProperty(a.getString("sage")));
            map.put("cname", new
SimpleStringProperty(a.getString("cname")));

            list.add(map);
        }
    } catch (SQLException e) {
        // TODO 自动生成的 catch 块
        e.printStackTrace();
    }
    return list;
}
```

查询是否管理员账户

使用 SQL 语句中的返回信息进行模糊查询。

```
public void isadmin() {  
  
    if(aaa.contains("root")) {  
        isadmin = true;  
    }else if(aaa.contains("ALL PRIVILEGES")) {  
        isadmin = true;  
    }else {  
        isadmin = false;  
    }  
}
```

等等工具类。

八 实验总结

本次实验遇到了许多问题，由于我对 Java 和 JavaFX 并不是特别熟悉，且 JavaFX 国内教程较少，导致功能不全，并且在 JavaFX 界面设计代码上出现了很多重复的地方，前期没注意加注释的问题，前后零零散散写代码并不规律，导致了代码比较混乱，这个地方我还有很大的改进空间。其次，在本系统的功能设计上，也并不是特别完善，例如：身份证号，年龄，性别的校验，并没有完善好，只是简单的实现了身份证长度的问题；用户界面设计中也并不是特别完善，某些界面节点的位置的设计并不是特别严谨。

虽然在这次课程设计中，我遇到了许许多多的问题，但在这几天的时间里，我还是实现了学生信息管理系统的大致功能。虽然过程比较辛苦，但通过这次的学习与实践，我对 Java 语言及 JavaFX 编写动态页面更加熟悉了，也收获了很多解决此类问题的经验，相信面对以后的课程设计或者实际开发项目，我能更好地去处理解决它。设计作品已上传至本人 GitHub¹公共仓库。

¹ <https://github.com/shenhao5826/Student-information-management-system>