目录

[1 绪论 1](#_Toc5908)

[1.1 课题研究背景及意义 1](#_Toc14000)

[1.1.1 选题背景 1](#_Toc5331)

[1.1.2 选题意义 2](#_Toc30511)

[1.2 主要研究内容 2](#_Toc32447)

[2需求分析 4](#_Toc11606)

[2.1 功能需求分析 4](#_Toc7721)

[2.2 非功能需求分析 5](#_Toc12442)

[3 系统设计 6](#_Toc31350)

[3.1 功能设计 6](#_Toc18825)

[3.1.2详细设计 6](#_Toc1488)

[3.2 数据库设计 9](#_Toc1577)

[3.2.1 概念结构设计 9](#_Toc4229)

[3.2.2 逻辑结构设计 10](#_Toc3694)

[3.2.3 物理结构设计 10](#_Toc16424)

[4 系统实现 11](#_Toc24904)

[4.1 系统实现环境 11](#_Toc28912)

[4.2 功能实现展示 11](#_Toc26380)

[4.2.1 数据库连接 12](#_Toc23280)

[4.2.2 主界面实现 12](#_Toc27723)

[5 总结 17](#_Toc25108)

[5.1 本系统的优点 17](#_Toc28763)

[5.2 缺点与改进思路 17](#_Toc28428)

# 

# **1 绪论**

1.1 课题研究背景及意义

1.1.1 选题背景

近年来，随着大学扩招政策的实施，大量学生涌入校园，这一趋势在给高等教育带来活力的同时，也带来了一系列管理上的挑战。其中，不乏有冒名顶替者，严重威胁到了教育的公平性和严肃性。与此同时，随着信息时代的到来，大学生们对自身信息与资料的管理意识逐渐增强，他们希望能够更加方便地查询和修改个人在校信息。然而，当前部分学校的管理系统在面对这些新需求时显得力不从心，亟需进行改革与升级。对此，需要着手使用更易上手的工具来开发一个更加安全易用的管理系统。

就传统的在校大学生个人学籍信息管理系统来说，可能存在以下问题：

1.数据孤岛与集成性差

不同的成绩与信息相互分离，如现在校园里广泛推行的“第二课堂”，其对应的二课积分不归属于普通课业成绩的管辖，分散在不同的部门。系统缺乏与其他外部系统如招生系统、就业系统等的数据互通，容易导致数据重复或丢失。由人工整理各项信息，效率较计算机系统低且容易出差错。部分UI与交互逻辑较为落后，学生需要查询的时候难以掌握正确的查询方法，容易查错或查不到自己的信息。且查询路径较为单一并有限制，基本都需要经过校园网，不能做到随时随地的查询。

2.功能繁杂

对于现行的系统，除开最主要的功能外，额外添加了许多不必要的功能，如成绩分析等，这些功能对于一个用来管理学生学籍的系统来说，是额外的负担。过多的功能，意味着更高的服务器负载，和更加拥挤的网络访问，这只会降低使用者对系统的满意度以及后期系统的维护程度。

3.技术架构落后，扩展性差

早期的管理系统往往部署在单机上或者小型局域网内，数据查询受限于单机或局域网服务器的稳定性，且一般使用Access甚至于早已停止维护的FoxPro（最新版本发布于2007年），虽然小型数据库可能速度快，处理性能好，但是面对海量数据，有极大可能因数据量过于庞杂而导致进程相应缓慢甚至崩溃。难以向已经成型的数据表中添加新的字段、添加新的模块，功能升级受到很大阻碍。

4.数据安全难以保证

总的来说，早期的学生学籍管理系统存在诸多缺点，这些缺点严重制约了学校信息化水平与服务水平的提升，促使新一代管理系统的出现与发展。所以就传统的信息管理系统而言，新的系统与程序可以克服以上弊端，推进高校数字教育与资源管理的现代化，智能化，为学生与高校信息管理人员提供更加便利的服务。

1.1.2 选题意义

在当前大学扩招政策的影响下，高等教育机构迎来了前所未有的学生数量增长，这不仅为学术研究和校园活动注入了新的活力，同时也对学校的日常管理和运作提出了更为严峻的挑战。特别是在学生身份验证和信息管理方面，一些潜在的问题逐渐浮现，成为亟待解决的难题。

一方面，随着学生数量的激增，冒名顶替入学的事件时有发生，这种行为不仅破坏了教育的公平性和严肃性，也损害了真正努力学生的权益，挫伤了他们的积极性。因此，高效、准确的学生身份验证核查，对于维护教育公正、保障学生权益具有极其重要的意义。另一方面，随着信息技术的快速发展和普及，大学生们对于个人信息的管理和隐私保护意识日益增强。他们期望能够便捷地查询和修改个人在校信息，同时保证这些信息的安全性和保密性。然而，当前许多学校的管理系统在设计上较为落后，无法满足这些新的需求，导致学生在使用过程中遇到诸多不便，甚至可能面临信息泄露的风险。

针对上述问题，开发一个更加安全、易用且功能全面的学生信息管理系统显得尤为重要。这不仅有助于提升学校的管理效率和服务质量，还能够更好地保障学生的信息安全和隐私权益。同时，通过引入更先进的技术和工具，可以降低系统的操作难度，使得管理人员和学生都能够更加轻松地掌握和使用该系统，从而进一步推动高等教育的信息化进程。

综上所述，本课题的选题意义在于通过开发一个安全易用的学生信息管理系统，有效应对大学扩招带来的管理挑战，维护教育公平性和学生信息安全，提升学校的管理水平和服务质量，为高等教育的可持续发展贡献力量。

1.2 主要研究内容

本项目围绕学生学籍管理系统的开发与应用展开，主要内容包括：系统需求分析，通过对学校学籍管理业务流程和用户需求的调研，明确系统功能和性能要求；系统架构设计，构建合理的系统架构，确定系统的技术选型和模块划分；功能模块开发，实现学生信息管理、成绩管理、学籍异动等核心功能；数据库设计与优化，设计高效、稳定的数据库结构，确保数据的存储、查询和更新效率；系统测试与部署，对系统进行全面测试，确保系统稳定运行，并完成系统的部署和推广应用。通过以上研究内容的实施，旨在开发出一套满足学校实际需求、具有良好实用性和扩展性的学生学籍管理系统。

# 2需求分析

在软件开发生命周期中，明确使用者的需求是至关重要的，一份详实的需求规格说明书对于后期的软件开发与实现有很大的帮助。此阶段需要充分调查学生学籍管理系统使用者对本系统的功能需求，明确实现该系统的各种软硬件环境，然后在此基础上实现之前调查的功能，以下将从功能需求与非功能需求两个方面来对本系统进行需求分析。

2.1 功能需求分析

本程序仅面向管理员方开放使用，仅设置管理员一位用户。此模块包括“添加学生信息”“查看学生信息”“管理学籍状态”三种功能。管理员的可以完成以上四种操作。管理员角色用例图如图2-1所示。



图2-1 管理员用例图

2.2 非功能需求分析

在软件生命周期的需求分析阶段，还需要分析系统的一些非功能需求，如表2-1所示。

表2-1 非功能性需求分析表

|  |  |
| --- | --- |
| 需求名称 | 需求分析 |
| 用户操作需求 | 本系统需要具有清晰的业务逻辑流程，，简单易用的用户界面，可以使用户很快上手。 |
| 安全性需求 | 本系统需要控制不同的用户权限，建立安全保障机制，要求如下：   1. 未注册用户无法查询系统中的信息； 2. 不同等级的用户不能越级操作或访问数据。 |
| 系统稳定性需求 | 本系统必须具有可靠的质量，较快的响应以及稳定的性能。在使用时，要求能够稳定运行，要避免卡退、死机等现象的发生。 |
| 数据完整性需求 | 本系统需要切实保障数据的完整性与一致性，具体要求如下：  1、所有的记录信息要完整且准确，主键均不能为空；  2、相同数据在不同表中要满足前后一致的要求。 |

# **3 系统设计**

## 3.1 功能设计

#### 3.1.1 总体设计

在总体设计阶段，要明确划分出系统的物理元素，同时确定系统由哪些模块组成，明确每个模块之间的互相联系。本学生学籍管理系统在实际设计中，主要根据三类角色分别来进行模块设计与开发，模块之间有关联性，每个模块分工明确，以便后续进行详细设计与整体的功能测试。通过对使用本平台的用户 角色的划分，构建出的本系统整体功能模块图如图3-1所示。

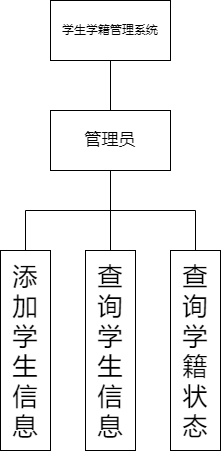


图3-1 系统功能模块图

3.1.2详细设计

在总体的基本模块设计完成后，需要对本系统进行详细设计，将模块的功能转化成详细的结构化描述。以下将对管理员业务流程的核心功能进行详细叙述，并加以流程图来辅助说明。

本学生学籍管理系统后台只有一个系统管理员用户，进入程序后直接以管理员身份进行管理操作，可实现对学生信息和学生学籍状态的管理。

1. 添加学生信息

在本系统中，管理员可以自行添加指定学生的个人信息，按照以下流程进行录入。如图3-2所示。



图3-2 添加学生信息业务流程图

1. 查看信息

在本功能模块中，管理员可以直接查看所有学生的信息。如图3-3所示。



图3-3 查看学生信息业务流程图

1. 管理学籍状态

本功能模块可以实现对指定学号的学生的学籍状态修改。如图3-4所示。

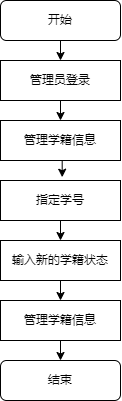


图3-4 管理学籍状态业务流程图

3.2 数据库设计

学生学籍管理系统的数据关系需设计为二维表，存储于关系型数据库中。良好的数据库设计有助于系统平台的高效运行和数据的一致性。在本系统设计中，我们尽可能满足第三范式（3NF），以消除冗余、降低维护成本并增强数据完整性。

3.2.1 概念结构设计

在概念设计阶段，将系统需求抽象为概念模型。系统采用 E-R 模型 表达管理员端管理对象之间的关系。主要实体为管理员、学生和学生信息。整体E-R图如图3-5所示。

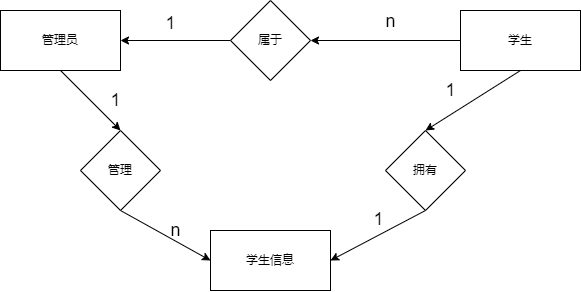


图3-5 系统总体E-R图

以下为实体的说明：

（1）学生信息实体属性图

学生信息的实体属性图包括学生学号、姓名、性别、成绩、学籍状态与所属班级。学生信息的属性图如图3-6所示。

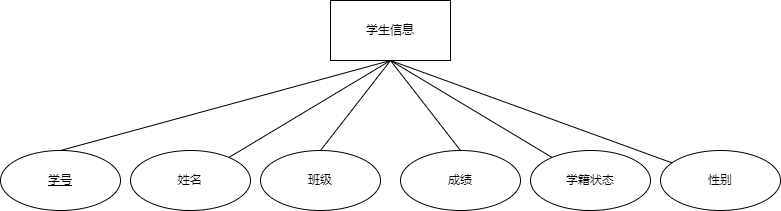


图3-6 学生信息实体属性图

（2）管理员实体属性图

管理员实体属性包括管理员ID、姓名、所属部门和联系方式。对应的属性图如图3-7所示。

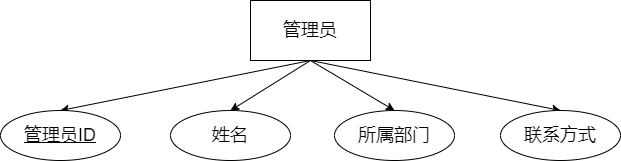


图3-7 管理员实体属性图

（3）学生实体属性图

学生实体属性包括姓名、学号、所属班级和性别。如图3-8所示。

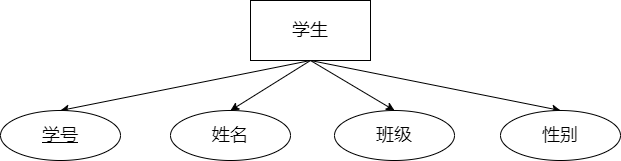


图3-8 学生实体属性图

3.2.2 逻辑结构设计

由概念设计的E-R图得到以下关系模式：

学生信息模式（id， name, class, grade, Total, graduation\_statues, gender）；

管理员模式（adminID, name, dept, contact）；

学生模式（id, name, class, gender）；

3.2.3 物理结构设计

有逻辑结构设计的关系模式经过优化，得到以下表（表3-1）。

表3-1 学生信息模式表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 是否为空 | 说明 |
| id | Varchar | 否 | 主键，学生学号 |
| name | Varchar | 否 | 姓名 |
| class | Varchar | 否 | 所在班级 |
| grade | Varchar | 否 | 年级 |
| Total | Varchar | 否 | 成绩 |
| graduation\_statues | Varchar | 否 | 学籍状态 |
| gender | Enum | 否 | 性别 |

表3-2 管理员模式表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 是否为空 | 说明 |
| adminID | Varchar | 否 | 主键，教师工号 |
| name | Varchar | 否 | 姓名 |
| dept | Varchar | 否 | 所属部门 |
| contact | Varchar | 否 | 联系方式 |

表3-3 学生模式表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 是否为空 | 说明 |
| id | Varchar | 否 | 主键，学生学号 |
| name | Varchar | 否 | 姓名 |
| class | Varchar | 否 | 所在班级 |
| gender | Enum | 否 | 性别 |

# 4 系统实现

本部分将对系统的具体实现进行详细的展示，以便对本系统有一个直观的印象。

4.1 系统实现环境

对于本系统的实现，主要分为硬件实现环境与软件实现环境。本系统开发硬件环境信息如表4-1所示，软件环境信息如表4-2所示。

表4-1 系统开发硬件环境信息表

|  |  |
| --- | --- |
| 说明 | 参数 |
| CPU | Intel(R) Core(TM) i7-12650H [CPU@2.30GHz](mailto:CPU@2.30GHz) |
| 内存 | 32.0GiB |
| 网卡 | 10-1000M |

表4-2 系统开发硬件环境信息表

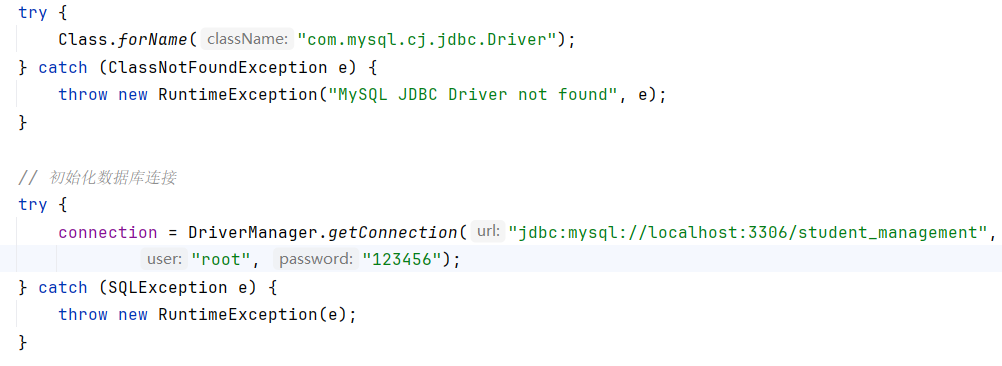
|  |  |
| --- | --- |
| 说明 | 参数 |
| 操作系统 | Windows 10 |
| 数-据库选择 | MySQL9.3.0 |
| 开发工具 | IntelliJ IDEA Community 2024.3 |
| 开发语言 | Java |
| 服务器 | Tomcat9.0 |

4.2 功能实现展示

此部分将会对本系统的具体实现功能进行分类展示，并加以部分代码的说明。

4.2.1 数据库连接

首先需要在MySQL数据库中创建数据库student，并在配置文件中实现数据可以连接配置。核心代码如下：



4.2.2 主界面实现

进入本系统后，会直接展示主界面。主界面包含一个“操作”按键。点按“按键”后会显示一个菜单栏，包含本系统目前所有的功能入口。主界面如图4-1所示。

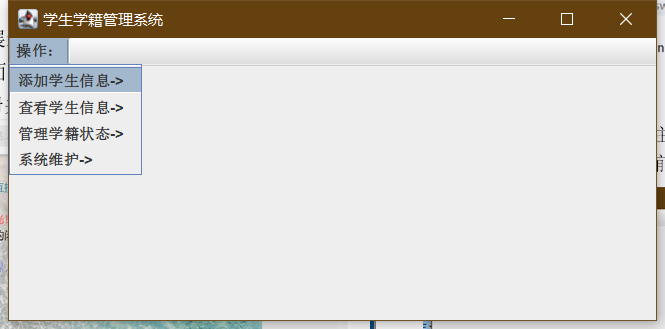


图4-1 主界面图

（1）添加学生信息

该功能模块可以进行学生基本信息的添加，包括姓名、总成绩、年级、学号、学籍状态与性别，除性别外不限制输入数值类型与长度。添加学生信息模块如图4-2所示。

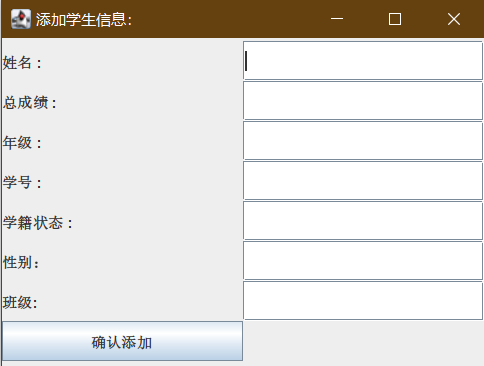


图4-2 添加学生信息模块

（2）查看学生信息

在本模块用户可以直接查看所有学生信息，学生信息按照学号顺序排列。如图4-3所示。

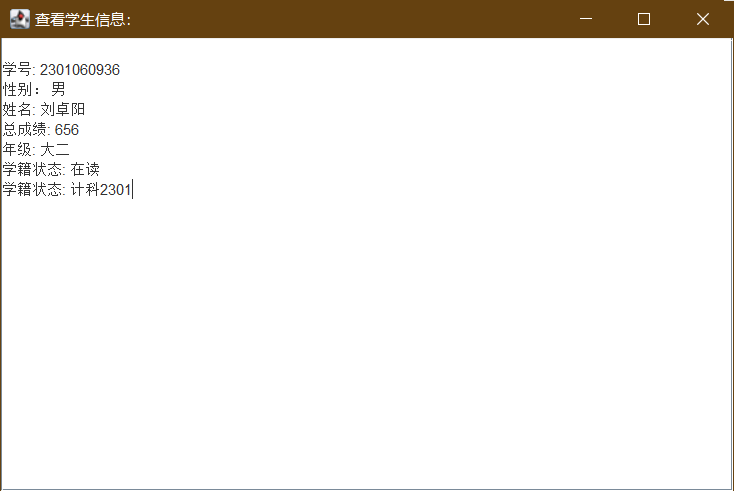


图4-3 查看学生信息模块

（3）管理学籍状态

在本模块，用户可以根据学生学号更新对应学生的新学籍状态，因学号具有唯一性，故无需担心因同名导致的一系列问题。更改后的数据将会同步保存至对应数据库中。如图4-4所示。

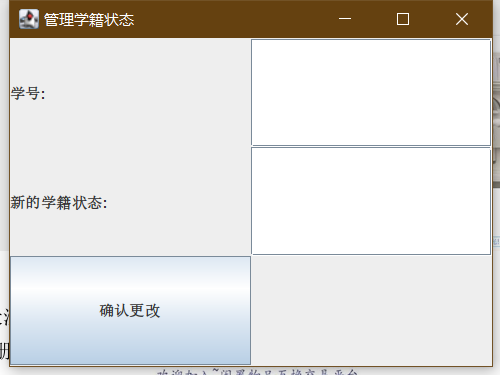


图4-4 管理学籍状态模块

（4）系统维护

本模块可以实现对于系统的升级维护，一般状态下不予开放。如图4-5所示。



图4-5 系统维护模块

4.2.3 用户功能实现

（1）添加指定的学生信息

进入页面可以添加指定的学生信息，同时支持重复学号检测，出现提示“Duplicate entry'xxxxxxxxx' for key'student.PRIMARY”即表示当前学号重复。功能如图4-6所示。

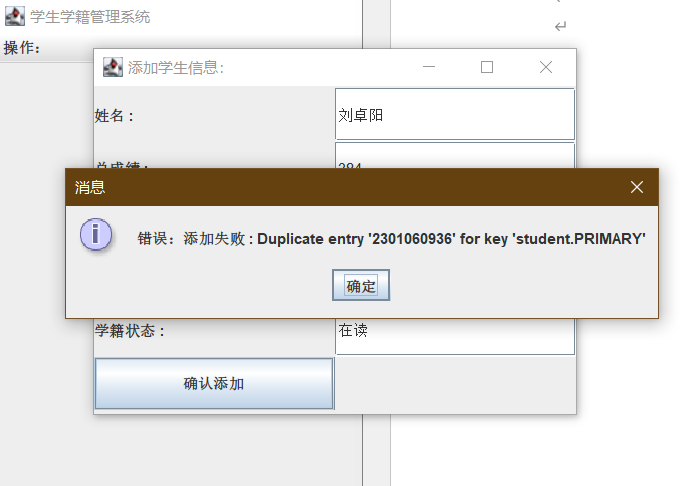


图4-6 添加指定学生信息

（2）查看学生信息

进入页面可以查看所有学生信息，信息按照学号先后顺序排列。如图4-7所示。

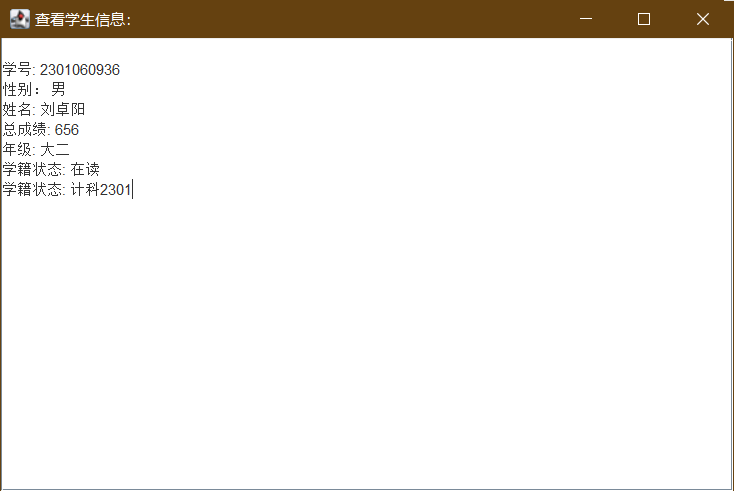


图4-7 查看学生信息

（3）管理学生学籍状态

进入本页面可以对指定学号的学生的学籍状态进行更改，支持不同语言和不同状态，可更改部分信息。如图4-8所示。



图4-8 管理学籍状态

# 5 总结

5.1 本系统的优点

本系统是在分析现行流行的学籍管理系统上新开发的系统，相较于前者，本系统更加的轻便，更便于使用者上手。简洁的操作界面与中文提示更加友好。为了与“学籍管理”更加切题，本系统较现行系统剔除了诸多与主题无关的功能，突出强调了本系统的主要任务——学籍管理。此外，本系统只有一位管理员，只能供一位管理员使用，使数据安全更有保障。

5.2 缺点与改进思路

缺点：一，本系统在开发时，仅仅强调了教务处管理员的方面，可能存在部分学生也有查询自己成绩与学籍状态的要求，这要求无可厚非，所有可以考虑后期添加适合学生用户的一个学籍查询系统。

二，本系统开发时为了紧扣“学籍管理”的主题，功能大幅精简，仅保留了基本的增加、修改和查询功能，没有删除信息的功能，日后优化系统时可以考虑将删除部分信息的功能添加进去。

改进思路：一，深入走访不同学校的教务处，综合分析共同需求，定向分析不同的需求。

二，为确保数据、程序与账户安全，可以添加一道身份验证机制，使用哈希算法加密数据库内的数据与账户数据。

参考文献