**《数据库开发课程设计》日志**

**班级：计科2301**

**学号：2301060936**

**姓名：刘卓阳**

**时间：2025年6月23日**

**完成任务：**

完成任务：项目启动与需求调研

项目伊始，我通过查阅高校信息化现状与相关政策文档，深入了解学生学籍管理系统存在的痛点。随着大学扩招和教育信息化的推进，传统管理系统在数据集成性、操作便利性、安全稳定性方面都面临挑战，特别是“数据孤岛”“功能单一”“系统老旧”等问题普遍存在，亟待改革。

我从角色角度出发，构思系统应覆盖的主要用户群体：学生、教务人员、管理员，并针对他们的典型操作需求（如成绩查询、学籍变更、权限分配）进行了初步功能构思。此外，我还列出一些前期要关注的非功能要素，包括安全性、访问控制、响应时间和异常处理机制。

为确保系统开发目标明确，我整理了一份开发任务清单和时间安排草案，计划用十天完成从需求分析到功能实现再到文档总结的完整闭环开发周期。项目目标不仅是做出一个能用的系统，更要确保它结构清晰、逻辑完备、体验良好。

**时间：2025年6月24日**

**完成任务：**需求分析与系统规划

这一天我重点梳理了系统的功能需求与非功能需求。基于前一日的调研成果，我制定了详细的功能模块清单，并按照用户角色分类：学生模块包括信息查询、成绩打印、学籍异动申请；教务模块涵盖成绩录入、毕业审核、学籍审批；管理员模块则包含用户管理、系统维护与运行监控。

随后，我编写了一份需求规格说明文档，明确了每一个模块下的核心功能流程。与此同时，我考虑到后期系统扩展的可能性，因此在设计上尽量保持模块独立、接口通用，避免耦合过高造成维护困难。

在非功能方面，我规定了系统必须满足的技术性指标：清晰的UI交互流程、良好的数据完整性机制、安全的权限体系，以及较高的系统稳定性。考虑到系统可能面临高并发查询或大数据表处理，我也提出后期可以考虑优化缓存机制或引入分层架构。

**时间：2025年6月25日**

**完成任务：**技术架构搭建与技术选型

今天我正式启动代码工程，并搭建整个系统的技术架构。我选择Java作为主要开发语言，因其跨平台稳定且生态丰富，且我已具备一定使用经验。为了实现界面直观的桌面应用，我决定使用Swing进行图形界面开发。数据库层则选择MySQL，结合MySQL Connector/J实现Java与数据库的连接与数据交互。

在代码目录结构方面，我采用了MVC设计思想，分为模型（Model）、视图（View）和控制器（Controller）三层，这样有利于代码管理与后期维护。为确保不同模块解耦清晰，我提前制定了模块接口规范。

我还配置了项目的开发工具（如IDEA、Navicat）和版本管理工具（Git），以便高效编写与调试代码。项目初始化完成后，我开始撰写数据库连接类与UI启动界面的雏形，为后续开发铺好技术基础。

**时间：2025年6月26日**

**完成任务：** 数据库建模与连接测试

今天专注于数据库设计与搭建。我详细设计了系统所需的核心数据表，包括用户表、学生信息表、成绩表、学籍申请表、系统日志表等，并为每张表定义了字段名、数据类型、主键外键约束和默认值等属性，确保数据结构合理且具备扩展性。

我特别注意到数据之间的关联性和一致性：如学生ID在多个表中作为外键出现，必须保持引用完整，成绩表必须与课程表联动，申请记录需关联到学生和审批人。通过ER图和数据关系图，我全面可视化了数据结构逻辑。

随后，我在MySQL中建库建表，并利用JDBC测试数据库连接功能，编写了简单的测试用例，对增删查改进行基本测试，确保系统后端能稳定与数据库通信。为后续操作的稳定性和安全性，我还配置了数据库连接池相关参数。

**时间：2025年6月27日**

**完成任务：** 界面结构搭建与交互初稿

今天我转向前端，开始利用Swing设计系统的用户界面。从登录页面入手，我设置了用户类别选择（学生/教师/管理员）、身份验证输入框，并添加样式控制逻辑。为提升用户体验，我调整了按钮大小、字体配色，并设计界面响应动画，避免出现“卡顿”或“无反应”的尴尬体验。

随后，我搭建了主菜单界面，按角色不同显示不同功能板块，采用卡片式布局结构，避免页面堆叠造成视觉混乱。同时我设计了通用顶部导航栏和底部提示条，规范操作路径，让用户更容易理解如何使用系统。

界面调试过程中，我不断从用户角度出发优化交互体验，加入了清晰的控件标签和操作提示，确保即便是新手用户也能快速上手。

**时间：2025年7月1日**

**完成任务：** 学生模块功能开发与数据交互实现

今天我聚焦在学生角色的核心模块开发，尤其是个人信息查询、修改与学籍申请等功能。我首先编写了学生登录后的主页逻辑，确保登录后根据身份正确加载对应模块。信息查询功能通过查询数据库中的学生表与成绩表实现了动态数据展示，同时使用 JTable 组件将数据以表格形式清晰地展现。

在“学籍变动申请”功能中，我设计了一个带下拉分类和申请原因输入框的界面，用户可选择休学、转专业等申请类型并提交至数据库的申请表中。我编写了数据验证逻辑，防止提交空字段或重复申请，并加入了提交结果反馈（成功/失败）提示框。

我还优化了 Swing 表单中的布局，让界面在不同分辨率下都能居中显示，不发生错位。此外，学生模块部分功能加入了操作日志记录机制，方便管理员后续查看用户行为数据。

**时间：2025年7月2日**

**完成任务：**教务与管理员模块功能构建

本日我专注于剩余两类用户角色的功能开发。教务人员方面，完成了成绩录入与修改模块，利用表格编辑与提交功能，支持批量处理学生成绩。同时实现了学业预警机制，当学生某一课程成绩不及格或平均分低于阈值时，系统可自动标记预警提示。

在毕业资格审核功能中，我根据“学分达标+无挂科记录”设定审核条件，系统可批量判定毕业状态，并支持人工审核确认。我还添加了信息注册功能，供教务人员对新生、在校生、毕业生分别进行信息录入。

管理员模块主要实现了用户权限配置、数据库参数设置、系统监控日志查看等功能。我加入了账号启用/禁用逻辑，确保非授权用户无法登录系统。同时实现了数据库备份接口，以保证数据的可恢复性与安全性。

**时间：2025年7月3日**

**完成任务：**系统集成与流程测试

今天是整个系统整合调试的关键阶段。我把前几天开发的学生、教务、管理员模块进行全面联调，确保各角色可以无缝切换登录并按权限访问指定功能区。我设计了多个模拟使用场景（如学生提交休学申请→教务审批→管理员查看日志），并逐步执行操作进行流程测试。

我还设置了错误数据测试，例如字段为空、非法值提交、权限越界访问等，以验证系统的稳定性与鲁棒性。在测试过程中，我发现了个别提交按钮未成功刷新数据的问题，也发现学生身份修改成功后界面未及时提示成功。通过逐条 debug，我修复了这些逻辑漏洞，并统一了前端交互响应风格。

在最终测试结果中，系统各模块之间能稳定运行，流程逻辑清晰，操作体验良好。

**时间：2025年7月4日**

**完成任务：**系统安全强化与用户体验优化

今天我重点处理系统的安全性与用户体验细节问题。在权限管理上，我加入了拦截器设计，系统在操作前必须校验当前用户的身份与操作权限。例如，学生不能访问教务数据，管理员拥有所有访问权限。未登录用户尝试访问界面时会自动跳转至登录页。

我还优化了异常处理机制，比如数据库连接失败、操作失败、提交无效输入等情况，系统都会弹窗提示明确原因，提升用户信任感。

此外，我对界面加载性能进行调优，减少不必要的数据加载操作，在数据量增大情况下保持良好响应速度。我也统一了界面配色、字体与按钮样式，提升整体视觉一致性。

为防止数据意外丢失，我在退出系统时加入了自动保存状态与操作日志写入，便于管理者后续审计与问题追踪。

**时间：2025年7月5日**

**完成任务：**文档整理与成果归纳总结

作为项目的收尾阶段，我对整个开发流程与技术实现进行了全面总结。我编写了项目开发说明文档，内容涵盖系统架构、数据库设计、功能模块说明、用户操作手册及测试用例结果。同时附上了系统结构图、流程图与关键模块代码注释，方便未来进行维护与功能拓展。

我整理了项目在开发中遇到的困难（如权限绕过漏洞、JDBC连接超时）与相应解决方案，并归纳出开发经验，如“提前设计数据结构的重要性”、“模块测试的分阶段策略”。

最后，我针对后续系统优化提出建议，如支持移动端访问、增加数据统计图表、对接学校外部系统接口等。整个项目不仅让我收获了技术实战能力，也锻炼了我项目规划与问题解决的综合能力。