

南阳理工学院·计算机与软件学院

**《Python程序开发课程设计》报告**

实验管理系统的设计与实现

专 业： 软工智能

班 级： 24软工智能一班

学 号： 2415929709

姓 名： 袁鑫晨

指导老师： 郭东恩

报告日期： 2022年06月xx日

南阳理工学院

Nanyang Institute of Technology

Python程序开发课程设计·成绩评定表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 考核方式 | 评分项目 | 课程目标 | 总分 | 得分 | 合计 | 占比 |
| 材料考核 | 课程设计报告 | 课程目标1 |  |  | 43 | 50% |
| 课程目标3 |  |  |
| 实践考核 | 课程设计答辩 | 课程目标1 |  |  | 48 | 50% |
| 课程目标2 |  |  |
| 最终成绩： | | |  | | | |
| 指导老师签字： | | |  | | | |

**1.材料考核环节成绩评分标准**

指导教师根据问题分析与定义，逻辑结构设计和详细设计的正确性等综合实训报告情况，按照百分制评定成绩，占总评成绩的50%。评分标准如5所示。

表5 材料考核评分标准

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **课程目标** | **考核指标** | **等级** | | | | |
| **优秀** | **良好** | **中等** | **及格** | **不及格** |
| **课程目标1：**根据给定的复杂工程问题，能够选择合适的函数和模块等，设计高效的程序流程方案并开发实现，分析、解释实验结果，判断程序结果是否符合预期，给出合理有效的结论。 | 函数、模块、技术方案、算法描述、实验结果分析及优化改进思路 | 数学模型正确、技术方案可行、程序流程描述完整；运行结果完全正确，能够对结果的进行准确、全面的分析评价，有明确的优化改进思路。 | 数学模型正确、技术方案可行、程序流程描述比较完整；运行结果正确，能够对结果的进行准确的分析评价，提出了一定的优化改进思路。 | 数学模型比较正确、技术方案可行、程序流程描述比较完整；运行结果正确，能够对结果进行分析评价。 | 数学模型基本正确、技术方案基本可行、程序流程步骤描述基本完整；运行结果基本正确，能够对结果的进行一定程度的分析评价。 | 数学模型、技术方案、程序流程描述等存在多处错误或缺失；运行结果不正确，不能对结果进行正确的分析评价。 |
| **课程目标3：**具备分析和描述软件开发过程的能力，能够系统规范的撰写软件开发设计文档，设计功能流程、简述实现过程、验证实现结果。 | 综合复杂性问题的分析和处理 | 能正确使用测试数据验证程序，测试结果分析全面且正确。 | 能正确使用测试数据验证程序，测试结果分析不够全面。 | 能正确使用测试数据验证程序，测试结果分析部分正确。 | 能使用测试数据验证程序，分析测试结果。 | 不能使用测试数据验证程序，无分析测试结果。 |

表6 实践考核成绩评分标准

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **课程目标** | **考核指标** | **等级** | | | | |
| **优秀** | **良好** | **中等** | **及格** | **不及格** |
| **课程目标1：**根据给定的复杂工程问题，能够选择合适的函数和模块等，设计高效的程序流程方案并开发实现，分析、解释实验结果，判断程序结果是否符合预期，给出合理有效的结论。 | 函数、模块、技术方案、算法描述、实验结果分析及优化改进思路 | 数学模型正确、技术方案可行、程序流程描述完整；运行结果完全正确，能够对结果的进行准确、全面的分析评价，有明确的优化改进思路。 | 数学模型正确、技术方案可行、程序流程描述比较完整；运行结果正确，能够对结果的进行准确的分析评价，提出了一定的优化改进思路。 | 数学模型比较正确、技术方案可行、程序流程描述比较完整；运行结果正确，能够对结果进行分析评价。 | 数学模型基本正确、技术方案基本可行、程序流程步骤描述基本完整；运行结果基本正确，能够对结果的进行一定程度的分析评价。 | 数学模型、技术方案、程序流程描述等存在多处错误或缺失；运行结果不正确，不能对结果进行正确的分析评价。 |
| **课程目标2：**具备依据Python课程设计实际问题的需求，运用所学知识对具体实际项目进行设计的能力。 | 程序质量，程序结构、代码规范性、可读性、测试样例的正确性。 | 程序运行稳定且功能完善。程序结构清晰，代码规范，可读性强。测试数据正确。 | 程序运行稳定且正确，功能全。程序结构清晰，代码规范，可读性强，测试数据正确。 | 程序运行稳定且正确，功能不够完善。代码规范，可读性强，测试数据正确。 | 程序运行稳定且正确。代码不规范，测试数据基本正确。 | 程序运行不稳定，测试数据不确。 |

实验室管理系统的设计与实现

软件工程 袁鑫晨

Design and Implementation of the  
XXXX System

Software Engineering Major Yuan Xinchen

目 录

[1 绪论 1](#_Toc19840)

[1.1 研究背景 1](#_Toc19103)

[1.2 研究目的与意义 1](#_Toc22268)

[1.3 国内外研究现状 1](#_Toc3109)

[1.4 研究内容 1](#_Toc28060)

[2 相关技术简介 1](#_Toc31983)

[3 需求描述与分析 1](#_Toc31800)

[4 系统设计 1](#_Toc9443)

[5 编码与实现 1](#_Toc4927)

[6 测试 1](#_Toc15170)

[总结与展望 2](#_Toc9568)

[参考文献 3](#_Toc20031)

# 绪论

## 研究背景

随着高校实验教学改革的深入和科研项目的持续增加，实验室资源管理复杂度显著提升。传统的人工管理方式在用户权限分配、实验任务跟踪、数据存储与查询等方面暴露出效率低下、易出错等问题。同时，多角色协同管理（管理员、教师、学生）对系统的权限分离和功能差异化提出了更高要求。在此背景下，开发一套基于 PyQt5 和 SQLite3 的实验室管理系统，实现信息化、规范化管理，成为提升实验室运行效率的迫切需求。

## 研究目的与意义

### 研究目的

设计并实现一个跨平台的实验室管理系统，满足三端用户（管理员、教师、学生）的核心功能需求，包括权限分离的用户管理、全流程实验任务管理、结构化数据存储与高效查询。

### 研究意义

通过信息化手段优化实验室管理流程，降低人工成本；实现数据的集中存储与安全共享，为教学科研提供数据支撑；培养软件开发实践能力，掌握 PyQt5 界面设计与 SQLite3 数据库交互的核心技术。

## 国内外研究现状

### 国外研究现状

发达国家较早实现实验室管理信息化，如美国高校普遍采用集成化管理系统（如 LabVantage），支持设备监控、实验预约、数据追溯等功能，技术成熟但成本较高。

### 国内研究现状

国内高校实验室管理系统多为定制化开发，以 Java 或 C# 为主要技术栈，近年来 Python 凭借简洁性和跨平台优势逐渐被采用。现有系统存在功能同质化、扩展性不足等问题，尤其在轻量级桌面应用领域存在优化空间。

## 研究内容

### 多角色权限管理

实现管理员、教师、学生三端独立界面与功能权限控制。​

### 实验任务全流程管理

支持任务发布、作业提交、成绩评定的闭环管理。​

### 数据持久化存储

基于 SQLite3 设计关系型数据库，实现用户、任务、作业数据的结构化存储。​

### 跨平台 GUI 开发

使用 PyQt5 构建响应式界面，通过 QSS 实现主题化样式设计。

# 相关技术简介

## Python 编程语言

Python 是一种高级、通用且解释型的编程语言，以其简洁易读、功能强大、开源免费等显著优势而闻名。在本系统的开发中，Python 扮演着核心角色，贯穿整个系统的开发过程，用于实现系统的业务逻辑处理和数据交互。

## PyQt5 框架

PyQt5 是 Qt 5 框架的 Python 绑定库，提供 500+GUI 组件，支持跨 Windows/macOS/Linux 平台开发。核心特性包括：​

**信号与槽机制:**解耦界面交互与业务逻辑（如登录按钮点击信号连接数据库查询槽函数）。​

**布局管理：**通过 QVBoxLayout/QHBoxLayout 实现响应式布局，适配不同屏幕尺寸。​

**模型 - 视图架构：**QTableView 结合 QSqlTableModel 实现数据库数据的实时渲染，支持表格编辑与筛选。

## QSS（Qt Style Sheets）

Qt Style Sheets（QSS）是 Qt 的界面美化语言，语法兼容 CSS，支持元素选择器、伪状态和属性设置。本系统典型应用：

\* {

background-color: #ffffff;

color: #808080;

}

QTextBrowser {

border: none; /\* 隐藏边框 \*/

background-color: transparent; /\* 背景颜色 \*/

color: #808080; /\* 文字颜色 \*/

border-radius: 10px; /\* 圆角 \*/

padding: 5px; /\* 内边距 \*/

}

QPushButton {

background-color: #e9e7ef;

border-radius: 10px;

padding: 5px;

border: none; /\* 移除边框 \*/

outline: none; /\* 移除焦点轮廓 \*/

}

QPushButton:hover {

background-color: #c0ebd7;

}

QPushButton:pressed {

background-color: #88ada6;

}

QPushButton:focus {

border: none; /\* 确保焦点时没有边框 \*/

}

QLineEdit {

background-color: #e9e7ef;

border: none; /\* 去掉边框 \*/

border-radius: 10px; /\* 圆角 \*/

padding: 5px;

color: #3d3b4f; /\* 输入字体颜色 \*/

border-bottom: 2px solid #3d3b4f; /\* 底部下划线 \*/

}

QLineEdit:hover {

background-color: #d0d7e5; /\* 鼠标悬停时的背景颜色 \*/

}

QLineEdit:focus {

background-color: #b8c0d0; /\* 获得焦点时的背景颜色 \*/

border-bottom: 2px solid #88ada6; /\* 获得焦点时的下划线颜色 \*/

}

## SQLite 数据库

轻量级嵌入式关系型数据库，具备以下优势：

**单文件存储**：数据存储于单个.db文件，无需服务器配置，适合中小规模数据管理。

**ACID 特性**：支持事务处理，确保数据一致性（如用户删除操作同时更新关联表记录）。

**Python 原生支持**：通过sqlite3模块实现无缝交互，支持参数化查询防止 SQL 注入。

## 面向对象编程（OOP）

系统采用 OOP 思想设计核心组件：​

**DatabaseManager 类**：封装数据库连接、表结构初始化、CRUD 操作，提供统一数据访问接口。

**BaseWindow 基类**：定义三端界面公共方法（如权限校验、数据刷新），提升代码复用性。​

**Model-View 分离**：界面组件（View）与数据模型（Model）通过信号通信，降低模块耦合度。​

# 需求描述与分析

## 功能需求

### 用户角色与权限

|  |  |
| --- | --- |
| 角色 | 核心功能 |
| **管理员** | 用户信息管理（增删改查）、权限分配、系统配置 |
| **教师** | 实验任务发布与管理、学生作业评分、成绩统计分析 |
| **学生** | 任务查看与提交、成绩查询、作业进度 |

## 非功能需求

**性能**：数据库查询响应时间≤1 秒（数据量≤10 万条），界面切换延迟≤0.5 秒。​

**安全性**：用户密码加密存储（采用 SHA 算法），防止 SQL 注入与越权访问。​

**扩展性**：预留设备管理、耗材申领接口，支持通过配置文件新增功能模块。​

# 系统设计

# 编码与实现

# 测试

总结与展望

参考文献