

系统设计描述文档

1. 系统概述

- **系统名称：**学生成绩管理系统
- **开发语言：**Python
- **数据库：**SQLite
- **目的：**实现学生信息、成绩管理、异常数据监控、统计分析等功能，为教师、学生和管理员提供简便的操作界面和高效的数据管理方式。

2. 系统架构设计

系统架构可以采用分层结构设计，主要包含以下几层：



图一 系统架构设计图

用户界面层：提供 Tkinter 界面，供用户进行数据输入和查询操作。

逻辑处理层：管理业务逻辑，处理用户请求（如身份验证、数据输入、异常检测等）。

数据访问层：与 SQLite 数据库交互，完成数据的存储和检索。

模块架构图：

用户界面层：ui.py（包含 LoginApp 和 StudentManagementApp 类）

逻辑处理层：data_export_import.py、settings.py、student_management.py（具体模块处理）

数据访问层：database.py（提供数据库连接、表的创建和数据操作接口）

3. 数据库设计

系统的数据库包含以下几张表：

3.1 Users 表

描述：用于存储用户的基本信息和角色。

字段：

UserID：用户 ID，主键。

StudentID：学生 ID。

Name：用户名。

Password：用户密码。

Role：用户角色（'teacher' 或 'student'）。

3.2 Students 表

描述：存储学生的基本信息。

字段：

StudentID：学生 ID，主键。

Name：学生姓名。

Gender：性别。

Age：年龄。

Class：所在班级。

EnrollmentDate：入学日期。

3.3 Courses 表

描述：存储课程的基本信息。

字段：

CourseID：课程 ID，主键。

CourseName: 课程名称。

CourseDescription: 课程描述。

3.4 Scores 表

描述: 记录学生的成绩信息。

字段:

ScoreID: 成绩 ID, 主键。

StudentID: 关联 Students 表的外键。

CourseID: 关联 Courses 表的外键。

RegularGrade: 平时成绩。

MidtermGrade: 期中成绩。

FinalGrade: 期末成绩。

3.5 ExceptionDataHandling 表

- 描述: 存储异常数据检测的结果。
- 字段:

Id: 异常记录 ID, 主键。

DataType: 异常数据类型。

ExceptionRule: 异常规则描述。

DetectedAt: 检测时间。

StudentID: 学生 ID。

ScoreID: 成绩 ID。

Description: 异常描述。

3.6 Settings 表

- 描述: 存储系统配置设置。
- 字段:

SettingID: 设置 ID, 主键。

Name: 配置项名称。

Value: 配置项的值。

4. 模块功能说明

4.1 用户管理模块

功能: 通过 LoginApp 类实现用户登录, 判断用户角色并跳转到相应的界面。



图二 用户登录界面图

- **主要操作:**

用户登录验证。

用户角色区分 (教师、学生)。

登录成功后, 显示对应的管理界面。

4.2 学生管理模块

功能: 管理学生的基本信息, 包括添加、编辑、删除和查询。

类: StudentManagementApp

主要操作:

显示学生基本信息。

学生信息的增删改查功能。

tk

学生管理

成绩管理

导入/导出数据

异常数据处理

所有学生成绩

姓名:

性别:

年龄:

班级:

入学日期:

用户名:

密码:

添加学生

查看学生列表

删除学生

	ID	姓名	性别	年龄	班级	入学日期
1		吴亿凡	男	22	22网络3班	2016年9月1日
3		赵一鸣	男	18	人工智能3班	2024年8月
4		王世凯	男	16	机械人工智能3班	2024年9月
5		小美	女	18	22计算机科学与技术	2024年6月
11		xah	女	18	23网络班	2023年9月
13		老得好	女	18	22网络3班	2024年9月
14		cxl	男	18	22计算机科学与技术	2024年6月

图三 学生管理模块示意图

4.3 成绩管理模块

功能：管理学生成绩，包括成绩录入、成绩编辑、成绩查询。

类：StudentManagementApp

主要操作：

教师可以录入学生的平时成绩、期中成绩和期末成绩。

查询特定学生的成绩记录。

tk

学生管理

成绩管理

导入/导出数据

异常数据处理

所有学生成绩

选择学生:

成绩ID:

平时成绩:

期中成绩:

期末成绩:

录入成绩

检测异常数据

ID	学生ID	成绩ID	异常描述	检测时间
----	------	------	------	------

图四 成绩管理模块示意图

4.4 异常数据检测模块

- 功能：通过 check_for_anomalies 方法，检测成绩数据中的异常值。
- 类：StudentManagementApp
- 主要操作：
 - 自动识别成绩中的异常数据（例如超过 100 分或低于 0 分）。

- 将异常数据记录到 ExceptionDataHandling 表。



图五 异常数据监测模块示意图

图四 异常数据监测模块示意图

4.5 数据导入导出模块

- 功能：通过 DataExportImport 类实现数据的导入和导出。
- 类：DataExportImport
- 主要操作：
 - 导出学生信息和成绩信息到 CSV 文件。
 - 从 CSV 文件导入学生和成绩数据。



图六 数据导入导出模块图

4.6 系统设置模块

- 功能：管理系统的基本配置。
- 类：Settings
- 主要操作：
 - 系统配置的读取和保存。
 - 系统配置的重置。

5. 用户界面设计

5.1 登录界面

- 功能：输入用户名和密码，登录系统。
- 控件：
 - 用户名输入框。
 - 密码输入框。
 - 登录按钮。

5.2 学生管理界面（教师）

- 功能：教师登录后进入的界面，可管理学生信息。
- 控件：
 - 学生信息显示表。
 - 学生信息增删改查的按钮。

5.3 成绩管理界面（教师）

- 功能：教师查看、录入或编辑学生成绩。
- 控件：
 - 成绩录入和编辑表单。
 - 查询成绩的输入框。

5.4 学生成绩查看界面（学生）

- 功能：学生登录后进入的界面，查看自己的成绩。
- 控件：
 - 显示学生个人的课程成绩信息。

6. 主要流程说明

6.1 用户登录流程

1. 用户在登录界面输入用户名和密码。
2. 系统查询 Users 表，验证用户名和密码。
3. 根据用户角色显示教师界面或学生界面。

6.2 学生成绩录入流程

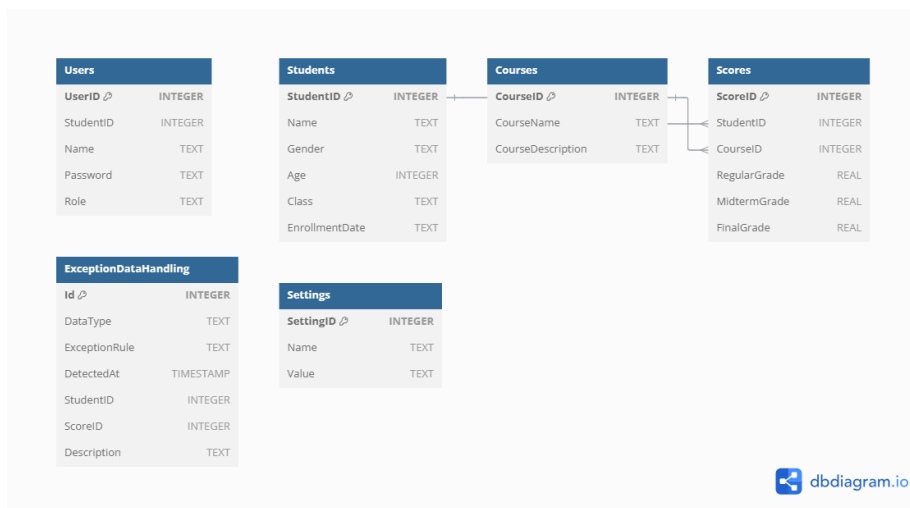
1. 教师选择学生并输入课程成绩。
2. 系统验证成绩的合理性（例如，分数在 0~100 之间）。
3. 将成绩存储到 Scores 表中。

6.3 异常数据检测流程

1. 系统定期或按需检测 Scores 表中的成绩数据。

2. 识别异常数据并记录到 `ExceptionDataHandling` 表中。

7. 数据库表结构



图七 数据表结构示意图

8. 系统约束和假设

假设所有用户数据都存储在 `SQLite` 中，不支持分布式操作。

系统不考虑大规模并发访问，仅支持单机版的操作。

使用 `Python` 的 `Tkinter` 界面库提供用户界面。