

基于 QT 的学生课程管理系统的设计与实现

张世博 白俊鸽

成都锦城学院计算机与软件学院 四川 成都 611731

摘要：课程管理工作是一个事业性的教育单元不可或缺的组成部分，随着信息化和计算机科学技术的进步和飞速发展，利用电脑来实现课程管理势在必行。本篇论文主要介绍学生对于课程管理的系统从设计分析、研究到开发的整个过程，在研究和开发的过程中，主要利用 VS19 和 QT 过设计和实现了对于课程、老师、学生的资料导入，修改等各项基本功能，具备了检索迅速、查询方便、可靠度高等优势，增强了对教务处的管理工作。

【关键词】学生课程管理系统；QT；C++

1 引言

学生课程管理数据库系统是完成对于常的教学和工作过程中学生考试成绩档案的一种数字化管理需求所需要的。针对教务、课堂教学上的各种服务和资料进行管理，使得数据的管理更加智能化。与此同时，减轻了工作人员日常的工作量，缩小了人力开支，提高效率，学生资料管理变得更为简便，为教育行业做出了突出的贡献，在未来的未来市场的竞争力强大，需求量非常大。

2 开发环境介绍

QT 为支持跨平台的图形界面应用框架，给专业开发人员提供构建可交互式的图形用户界面时所必须要做的所有功能。QT 是完全面向对象的，很容易进行扩展，并且实现了真正的组件编程。^[1]

3 主要功能描述

学生课程管理系统以管理学生成绩信息为目标，主要功能包括：

成绩录入、查询、修改、删除、单科成绩排名、查看学生信息等。

(1) 成绩录入：每次考试结束，都可以添加学生成绩信息并且支持正确格式的一键导入成绩的功能。

(2) 查询：查询单个学生和多个学生的成绩和基本信息，支持模糊匹配学生姓名查询。

(3) 修改：包括修改学生某一门课的成绩和个人的基本信息。

(4) 删除：包括删除学生的成绩和学生的基本信息。

(5) 排名：支持按学生成绩的高低顺序排列，也可以支持按学生学号进行排序。

(6) 查看的学生信息。

4 界面设计

用户登录窗体中放置了 Qt 的标签控件 QLabel，显示登录系统名字：学生课程管理系统，使用三个 QRadioButton 的作用是选择管理员、教师、学生的登录身份，主界面使用 QLabel 显示用户的个人信息和 QTableWidgetItem 显示分别显示教师和学生信息，QPushButton 负责和用户交互实现功能和页面跳转，使用 QLineEdit 获取用户输入以用来添加教师和学生、课程、收集学生评价，使用 QToolButton 嵌入在 QTableWidgetItem 中提供对某一行的修改、删除功能整体采用无边框设计，

以达到美化、简单的的好处。

5 功能实现

5.1 界面控件

根据我们实地考察整理出的系统需求，分为个人信息展示页面和功能选项页面和课程数据显示页面^[2]，个人信息页面使用标签进行个人信息的展示具体展示姓名，专业，年级，已选课程总学分。功能选项页面主要使用按钮和用户交互，三种角色登录分别显示不同的控件。学生显示选课按钮，教师端则在 QTableWidgetItem（以下简称 QW）进行交互，管理员端则显示三个按钮负责课程、教师、学生的信息的导入。使用 QW 关联模型显示数据，模型负责数据的组装而 QW 负责数据的显示，使用工具按钮作为表格中的按钮进行功能交互。使用按钮作为程序的关闭按钮，隐藏 QT 默认的边框使用无边框布局，因为在程序关闭时需要保存数据，并且 QT 默认的边框不美观，所以使用按钮重写了一个程序关闭功能^[3]。



图一：学生课程管理系统登录界面

5.2 设计思路及具体功能

作为一个学生课程管理系统，最基本的需求就是课程、教师、学生信息的导入，如何让用户有一个最简单的操作体验是本应用最大的追求，所以完善的设计思路是迫切需要的。由于本系统需要三种身份使用所以我设计了三个类支持此需求，每个类都独立又互相关联。彼此的连续是基础的课程数据^[3]，而且在界面的角度来看只是所需功能不一样。所以在此基础上我设计了一个“学生身份选择系统”（以下简称 SES）公共界面处理三种不同身份的登录，如图 1，当登录时候会附带一个登录身份 Role^[4]传递给 SES 类进行不同界面的判断处理和

初始化。

额外需要一个课程类进行课程信息的相关操作。由于每个身份登录所需要的数据是完整的,所以不论每个身份登录,程序会默认装载全体数据,所有的信息都保存在 QVector (以下简称 QV) 中,比如当管理员需要添加课程时,会生成课程类的一个对象,获取用户输入后将此课程对象添加到 QV 中,添加学生、教师类似。当教师需要对所授课程进行成绩导入时,会找到对应学生对象的课程 QV 中添加成绩。当学生查看自己成绩时,系统会查询学生的课程 QV, 查找到教师导入的成绩从而显示自己所有课程的成绩。当每个身份操作数据时系统会记录该操作的时间以及操作了什么数据,方便管理员进行查看。当程序结束时系统会按照用户、教师、学生、课程的顺序将内存中的数据存储到对应的数据文件中,比如当存储学生的信息时会首先将打开学生的数据文件,会依次写入学生的相关数据,并依据课程数在另起一行写入课程的课程号和课程班级。该程序的设计思路简介明了,代码简单^[5],易懂。主要设计了 4 个类分别是管理员、教师、学生和课程来进行数据的交互。使用文本文件存储数据。

5.3 数据存取功能

在程序启动时会去检测数据文件是否正确,若不正确会自动修正系统所需要的文件,若正确会按照行读取数据文件,会顺序的装载用户数据和教师数据、学生数据。若哪一个环节出现问题都会导致程序不能正常工作,所以本程序不支持手动修改数据文件。在装载用户数据时会分别读取用户的编号和密码以及个人信息。装载到内存当中,当装载教师信息时会载入教师的编号 姓名 所属学院和所授课程数,程序会根据所授课程数去读取课程信息,课程信息包含课程编号、课程名、课程班级、选课学生数和对应学生的信息。当装载学生信息时会载入学生的学号、姓名、学院、专业、班级和所选课程的总学分、所选课程数量,程序会按照课程数量读取对应的课程信息,包括课程编号、成绩以及对教师的评价。当程序被关闭时会自动存储数据,会将用户数据、教师数据、学生数据进行更新。当用户对数据进行修改时会产生记录,记录在日志文件里。方便用户查看操作记录。

5.4 打开数据文件功能

由于本学生课程管理系统是一个本地学生课程管理系统,必不可少的功能自然是打开本地数据文件。其中批量导入学生成绩需要选择文件功能,在打开文件按钮的点击响应函数中,先设置一个 QString 类型的变量来接收使用 getOpenFileName() (以下简称 OFN) 函数获取到的需要打开的文件名,为了避免选中非正确文件打开造成程序崩溃,使用 OFN 时,通过文件后缀名筛选出文本文件,这样用户在打开文件的窗口中的文件夹里就不会看到非文本文件,从而避免了非文本文件造成本学生课程管理系统的崩溃^[5]。

6 测试及结论

项目完成之后,对本学生课程管理系统采用黑盒测试的方式进行了功能性测试。

登录界面分别使用三种身份登录,分别正常显示对应界面以及显示正确的个人信息,该登录功能正常。当

使用学生身份正常进入界面后可以正常看到管理员创建的课程详情,可以进行选课和退课,当老师进行成绩的导入后可以对老师进行评价查看自己的成绩和排名等信息。当点击对老师进行评价时,可以正确的与用户进行交互保存评价。结论:该功能正常。当使用教师身份正常进入界面后,可以正常查看管理员分配的课程详情,可以看到学生列表,可以进行单个的成绩导入也可以进行批量成绩的导入,当成绩完成导入后,系统会自动计算平均成绩、及格率、平均绩点等信息,并且教师能够正常的进行退课。点击导入成绩按钮后,选中本地文件夹中需要的成绩表,成绩成功导入,结论:该功能正常。当使用管理员身份登录系统时,正常显示对应的控件。结论:该功能正常。当点击课程管理时,数据显示界面能正常显示课程信息列表,可以进行课程的增加,可以课程余量和此课程被学生选了多少,课程包含学生的个人信息和成绩,以及学生对此课程的评价,结论:该功能正常。当使用教师管理时, QTableWidgetItem 界面能正常显示所有的教师的信息,可以添加或修改删除信息,还可以查看教师所授课程情况,结论:该功能正常。当点击学生管理时,数据显示界面能正常显示学生信息列表,可以添加或修改删除学生的信息。还可以查看学生所选的课程情况,包含学生个人信息和选课成绩等信息,结论:该功能正常。点击关闭按钮后,系统能够正确的保存系统记录的信息,在重启程序之后能够正常的读取信息,结论:该功能正常。

总结:该学生课程管理系统的基础功能均能正常使用。

7 结语

本文介绍了如何使用 QT 开发一款学生课程管理系统,该系统具有所占内存小,功能比较完善、界面简洁的特点,但相对于多数学生课程管理系统还存在许多不足,比如可以修改为数据库存储可以提高数据的安全些,但是同时也会带来开发成本的提高但是这是利大于弊的。开发成本低,代码简单易懂,非常适用于 QT 初学者学习练习。

【参考文献】

- [1] 黄橡丽,周艳聪,孟巍. 数据结构在学生课程管理系统设计中的应用[J]. 数字技术与应用, 2019, 37(11):140-141.
- [2] 任荣君. 基于计算机技术的跨平台学生信息多终端管理系统设计[J]. 无线互联科技, 2021, 18(09):63-64.
- [3] 蔡倩芸. 高校发展型学生资助信息管理系统构建研究[J]. 黑龙江科学, 2021, 12(09):144-145.
- [4] 周清松,付星宇. 高校学生请假管理系统的设计与开发[J]. 电脑知识与技术, 2021, 17(13):63-66+85.
- [5] 段振虎. 基于 Qt 纯代码编写软件的方法实例[J]. 电脑知识与技术, 2020, 16(27):72-74+84.