

浙江大学

面向对象程序设计期末设计报告

选课系统设计

姓名/学号:

李蕴哲 3140105619

傅子谦 3120102373

刘迪一 3140105434

曾钰文 3140105136

组号: 35

第 35 组面向对象程序设计期末设计报告

李蕴哲 3140105619

傅子谦 3120102373

刘迪一 3140105434

曾钰文 3140105136

本次 OOP 的期终作业，我们选择了选课系统这一题目，根据基本要求最终合作实现了一个简单的选课系统。

该系统的设计思路是基于 QT 和 MySQL，由前端界面接收输入，交由内核进行信息重整，再与后端数据库进行交互获得结果后，返回到前端界面进行显示。程序的设计被分为了三个部分：一个是界面部分，即根据功能需求构建起完整的符合需求的界面架构即基本的界面运转逻辑；一个是内核部分，即在接收界面的输入及信号后调度不同的管理模块对信息进行重整；最后一个数据库连接单元部分，即根据内核提供的信息与任务与数据库进行交互。

通过本次作业，我们充分锻炼了团队协作能力与利用 C++ 及第三方库进行编程的能力。尽管最终的成果还存在着许多不足，但是也能够体现出我们的努力。

一、 程序背景介绍

1. 程序背景及基本要求

选课系统面向学生、老师、管理员三种不同角色，要求具有不同的独立界面。学生能够通过系统选课、查看课程安排，教师能够查看课程安排、选课情况、录入成绩等，管理员能够利用系统管理课程、学生和老师的信息。各个部分的增删改查功能需较为完备。

2. 开发平台

Qt 5: Qt 是一个跨平台的 C++ 图形用户界面应用程序开发框架，用户能够使用其开发 GUI 程序。本次期终作业主要是借助 Qt Creator 3.6.1 进行开发，基于 Qt 5.6.0 及 MSVC 2013 64 bit 的构建套件。

MySQL: MySQL 是一个关系型数据库管理系统, 用户能够借助其进行数据存储、管理与使用。本次期终作业主要是基于 MySQL Server 5.7 进行数据库交互。

二、 程序设计与架构

1. 总体框架设计

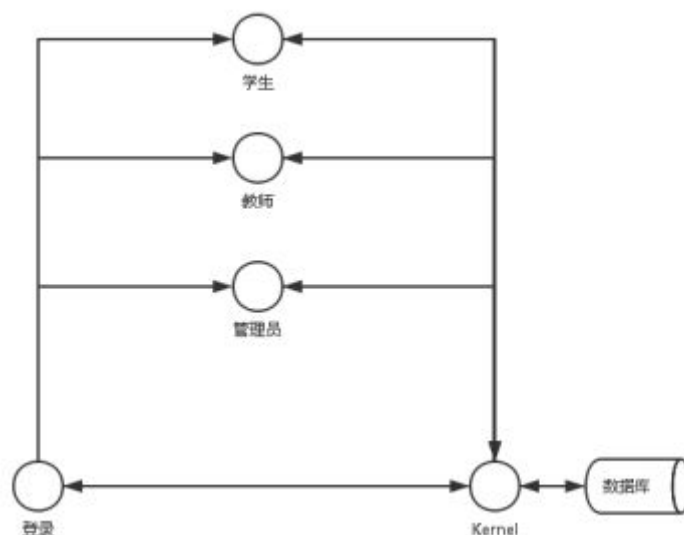


Figure 1 总体结构设计

我们的总体框架设计如上图。由于我们的程序设计基于 MySQL 数据库, 因此我们将程序分为了数据库部分和程序自身两个部分。Qt 提供了 QSqlQuery 这个库可以用于 Qt 程序与数据库的交互, 因此我们可以将数据库部分剥离为数据库设计, 然后在程序自身部分对 QSqlQuery 根据我们的需求进行封装。因此程序自身部分也就分为了界面部分、内核部分和数据库连接单元 (即 QSqlQuery 封装部分) 三个部分。上图中的“登录”、“学生”、“教师”、“管理员”分别对应独立的界面部分, “Kernel”则对应内核部分, 数据库连接单元部分则不直接体现在其中, 但是我们将其与内核部分组合在了一起, 由一个全局内核对象进行从界面输入到数据库输入、从数据库输出到界面输出的操作。

2. 数据库设计

数据库的设计主要是根据系统需求设计对应的数据表存储信息, 我们的设计

如下，其中下划线部分表示该表的主键：

- Course (courseid, coursename, dept, description)
课程 (课程编号, 名称, 专业, 介绍)
- Student (id, name, grade, class, phone, email)
学生 (编号, 姓名, 年级, 班级, 电话, 邮箱)
- Teacher (id, name, prof, phone, emai, dept)
教师 (编号, 姓名, 职称, 电话, 邮箱, 专业)
- Dept (deptname)
专业 (名称)
- Account (userid, passwd, rol)
账户 (编号, 密码, 角色) (userid 同其他表的 id)
- Admin (id, name, phone, email, dept)
管理员 (编号, 姓名, 电话, 邮箱, 专业)
- Takes(courseid,coursename,stuid,stuname,teaid,teaname,day,starttime,endtime,pos,score)
上课记录 (课程编号, 课程名, 学生编号, 学生姓名, 教师编号, 教师姓名, 周几, 开始节数, 结束节数, 上课地点, 分数)
- Teaches(courseid,coursename,teaid,teaname,capacity,currentstu,day,starttime,endtime,pos,dept)
授课记录 (课程编号, 课程名, 教师编号, 教师姓名, 最大容量, 现有容量, 周几, 开始节数, 结束节数, 上课地点, 专业)
- Newcourse (courseid, teaid, coursename, dept, description)
- 申请课程 (课程编号, 申请教师编号, 名称, 专业, 介绍)

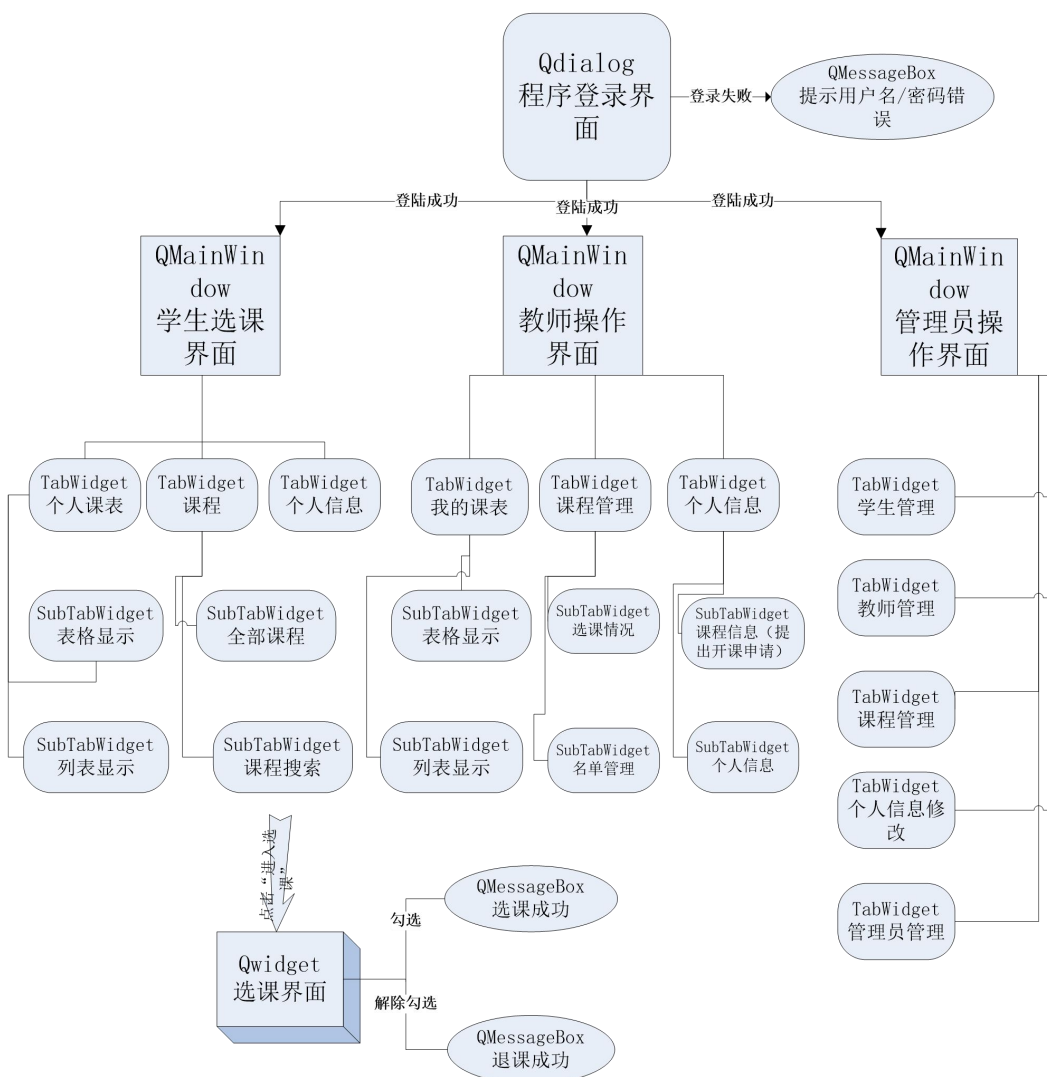
其中 takes 和 teaches 由于是对于不同角色所选的不同课程，因此主键是由课程编号与角色编号组合而成，这样能够表示唯一的记录。当然由于此处在设计申请课程时没有考虑到“申请课程”与“申请授课”的不同，导致我们不能在系统中直接申请授课，而需要后台在数据库添加，下面是数据库中的外键依赖：

- Student.id, Teacher.id, Admin.id 依赖于 Account.id (级联更新, 级联删除)
- Course.dept, Teacher.dept, Admin.dept, Newcourse.dept, Teaches.dept 依赖于 Dept.deptname (级联更新, 级联删除)

- Take.courseid, Teaches.courseid 依赖于 Course.courseid（级联更新，级联删除）
- Takes.stuid 依赖于 Student.id（级联更新，级联删除）
- Teaches.teaid, Takes.teaid, Newcourse.teaid 依赖于 Teacher.id（级联更新，级联删除）

基于这种外键依赖和主键设定，我们在删除学生/教师/管理员时可以一并删除账户与具体信息，在删除课程时可以一并删除该课的上课记录、授课记录和申请课程记录，诸如此类的删除与更新都能保证一致性。但同时其他的一些属性的一致性并不能完全保证。具体的属性数据类型定义不在此显示，详情可参见附的源代码中的 CourseSystem.sql。

3. 界面设计



在设计界面时，我将选课管理系统的功能根据使用者身份的不同分为三块——分别是学生选课模块，教师操作模块和管理员操作模块。这三个模块因为是所有界面的主要承载体，因此采用 `QMainWindow` 绘制。考虑到登录功能的需求，我又在这三个模块之前设计了继承 `QDialog` 类的登录界面，根据登录时选择的身份选择需要显示的模块，并在输入的用户名和密码错误的时候使用 `QMessageBox` 弹出错误信息对话框。

而考虑到界面设计的整体性和简洁性，并借鉴了浙江大学教务管理系统的设计，在这三个模块下面的功能界面我都采用 `QTabWidget`（即标签界面）来绘制。首先我将学生选课模块分成三个主要部分，个人课表，课程和个人信息。在个人课表下，是两个子 `tab`，分别是表格显示和列表显示，采用两种方式对已经选择的课程进行显示以应对不同的需求。而在课程模块下，分为全部课程和课程搜索两个部分。全部课程下，学生可以浏览当前开设的全部课程，而在课程搜索功能下，学生可以填写课程名，课程编号，上课时间等信息，有针对性的搜索需要的课程，并通过点击“进入选课”按钮，打开一个由 `QWidget` 类定义的选课窗口，显示该课程开设的全部教学班，并在这个选课窗口里通过勾选/取消勾选 `checkBox` 来进行选课和退课操作，每次操作都会有 `QMessagebox` 提供提示信息。而个人信息界面主要是显示学生个人信息并提供修改服务。

教师操作界面同样分成三个部分，分别是个人课表，课程管理，个人信息。在个人课表部分，与学生选课界面相似，针对教师当前开课采用表格和列表两种方式进行显示。在课程管理界面下分两个子 `tab`，分别是选课情况和名单管理，在选课情况下显示教师当前开设课程的容量情况，而在名单管理下显示当前选课的学生，并提供可编辑的表格，和“成绩录入”按钮来完成成绩录入的功能。而在个人信息模块下，教师可以修改自己的个人信息，并为新设课程提出申请。在管理员模块，则是提供了学生，教师，管理员，课程的增删功能。其中学生、管理员、教师提供注册功能。而课程则可以通过教师申请审核和直接添加两种方式进行增加。

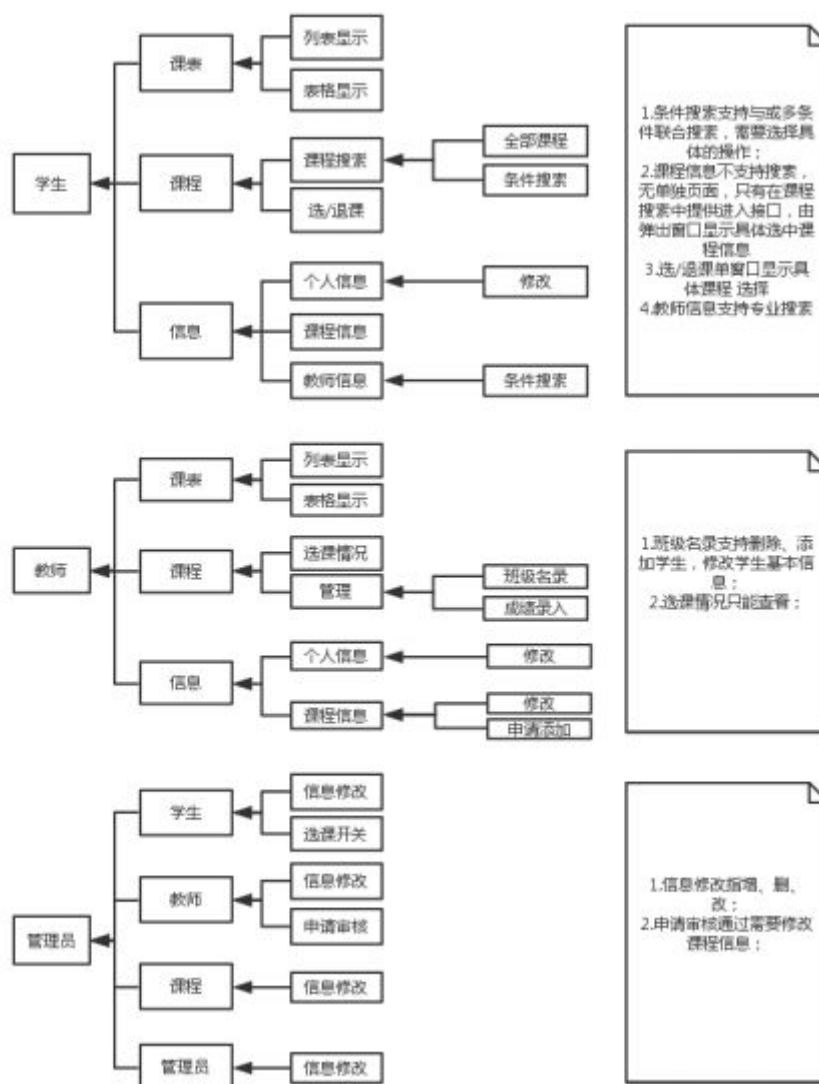
整个界面程序的设计参考了部分浙江大学教务管理系统的设计方式，也实现了教务管理系统根据身份不同而产生的不同功能需要，较好完成了选课管理系统对界面的需求。

4. 代码逻辑模块设计

代码主要位于 `cskernel.h` 中，其中 `kernel` 是整体功能的中心部分，根据不同的要求调用不同的 `manager` 进行处理。

数据库连接单元由 `myquery.h`, `myinsert.h`, `myupdate.h` 和 `mydelete.h` 组成，分别对应数据库的增、删、改、查。

三、 程序功能说明



详细功能于视频中有完整演示与描述。

四、 总结

本次期末设计的完成较为曲折，更多的是由于我（李蕴哲）个人原因导致全组的工作被拖延，也错过了现场验收的机会。

通过本次设计我们锻炼了团队合作的能力，和利用 C++ 与第三方类库编程的能力。由于在最初的整体设计中有一些没考虑到的地方，使得后续的工作也遇到了诸多的困难，并砍掉了一些预想的功能以及一些逻辑上的检查，因此最终的结果也是存在不少问题。但是总体来讲我们还是尽力去完成了这次的作业，最终的结果也能部分的体现出我们的努力。

五、 小组分工

总体架构设计：李蕴哲、傅子谦、刘迪一、曾钰文

数据库设计：李蕴哲

界面架构设计：傅子谦

代码逻辑模块设计：

界面逻辑模块：傅子谦

内核模块：李蕴哲、刘迪一、曾钰文

数据库连接单元模块：李蕴哲、刘迪一、曾钰文

报告撰写：李蕴哲、傅子谦、刘迪一、曾钰文

视频录制：李蕴哲