

2014-2015-(1)-C++

《C++》

软件实习报告

成 绩:

评阅人:

学院: 计算机科学与技术

专业: 计算机科学与技术

年级: 2013 级

班级: 4 班

姓名: 梁振铎

学号: 3013216096

日期: 2015 年 10 月

1. 问题描述： 建立图书管理系统

要求该系统能够实现：借书、还书、读者维护、图书维护等功能。

馆藏图书建有检索索引目录，保存相关收藏的各种信息，包括：编号、书名、作者、出版社、出版日期，每种图书有 1-10 册不等。馆中各类图书允许借阅离馆，图书馆需了解图书的借阅者姓名、借阅者联系电话、借阅日期。

借阅图书被归还后，首先要经过检查，正常情况归库以备再借；轻微破损经修补后归库；严重破损或被借阅者遗失时，联系借阅者赔偿，此书淘汰。

程序应可显示出某册图书的当前状态（库存、借出、修补、淘汰）。为方便读者检索，系统可以用任意字串为关键字在所有馆藏图书的名称索引内进行检索（图书名称内包含关键字字串时即为选中），并显示结果。

为提高图书周转，规定每人最多借阅图书 30 天，系统每日检查被借图书，给出超期借阅者名单以便联系。当显示读者信息时，应该能够显示其所借图书，以及每本书离归还还有多少天。为了解图书需求状况，系统可对图书借阅历史进行分析，对某一专业的所有借阅者所借图书，按每本图书的被借次数给出排序结果。

2. 问题分析：

图书管理系统的开发环境为 Qt 5.5，编译器使用 MinGW，数据库使用 MySQL 5.6。整体系统可以分用户交互界面和后端处理模型两个部分。

用户交互界面是通过 Qt Creator 的 UI 设计器设计完成，并将用户界面封装为一个类的形式在后端主程序当中对类进行实例化。对于用户 UI 界面与后端主程序通讯的方式，使用了 Qt 的信号槽机制（SIGNAL & SLOT）来实现，对于 UI 界面中的每一个交互操作都会形成一个 SIGNAL，并通过 connect 触发主程序的对应 SLOT 来执行。通过基于 GUI 的 UI 界面可以增强对用户的操作友好性，相比网页应用操作更为流畅便捷，相对于 Console 应用更加直观易用。

后端处理模型负责与数据库的通讯、界面交互以及后端数据处理等工作。在数据库通讯方面使用了 QODBC 连接 MySQL ODBC 驱动。界面交互时，可根据用户的不同操作，通过向数据库发送查询语句获取相应信息，并经过对数据的加工处理后返回给用户。

3. 数据模型：

(1) 对象模型

本程序在头文件 public.h 中定义了 `class TJUL_CLASS { }`；类用作基本对象的类，对象 TJUL 声明在主函数中。

(2) 全局变量

全局变量声明在头文件 public.h 中：

```
class TJUL_CLASS {  
private:  
    bool is_connect_database;  
    int num_borrowed;  
    int num_limit;
```

```
public:
    bool isLogin;
    bool isAdmin;
    QString stuid;
    QString name;
};
```

其中 `is_connect_database` 表示数据库连接状态, `num_borrowed` 表示当前用户借书量, `num_limit` 表示当前用户借书上限, `isLogin` 表示当前登录状态, `isAdmin` 表示当前用户是否为管理员, `stuid` 表示当前用户学号, `name` 表示当前用户用户名。

(3) 数据库结构

数据库名为 `dbms`, 其中包含以下 6 张表:

表 **book**: 用于存放书目信息, 包括 ISBN、书名、类型、出版社、出版年份、作者、数量, 其中 ISBN 作为主键;

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
isbn	CHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		None	UTF-8	utf8_general_ci
title	CHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	UTF-8	utf8_general_ci
type	CHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	UTF-8	utf8_general_ci
pub_press	CHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	UTF-8	utf8_general_ci
pub_year	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None		
author	CHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	UTF-8	utf8_general_ci
num_total	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None		

表 **user**: 用于存放用户信息, 包括学号、姓名、密码、联系电话、借书数量、借书上限、学院、用户类别, 其中学号作为主键;

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
stuid	CHAR	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		None	UTF-8	utf8_general_ci
name	CHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	UTF-8	utf8_general_ci
password	CHAR	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	UTF-8	utf8_general_ci
telnum	CHAR	16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	UTF-8	utf8_general_ci
num_borrowed	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None		
num_limit	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None		
school	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None		
isadmin	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None		

表 **item**: 用于存放单册图书信息, 包括书号、ISBN、图书状态, 其中书号作为主键;

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
id	CHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		None	UTF-8	utf8_general_ci
isbn	CHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		None	UTF-8	utf8_general_ci
status	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None		

表 **record**: 用于存放借书记录信息, 包括流水号、书号、学号、图书状态、借书时间、应还时间、还书时间、逾期时间, 其中流水号作为主键;

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
uuid	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		auto_in...		
id	CHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		None	UTF-8	utf8_gene
stuid	CHAR	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	MUL	NULL	None	UTF-8	utf8_gene
status	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None		
time_borrow	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		CURR...	None		
time_deadline	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0000-...	None		
time_return	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None		
overtime	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None		

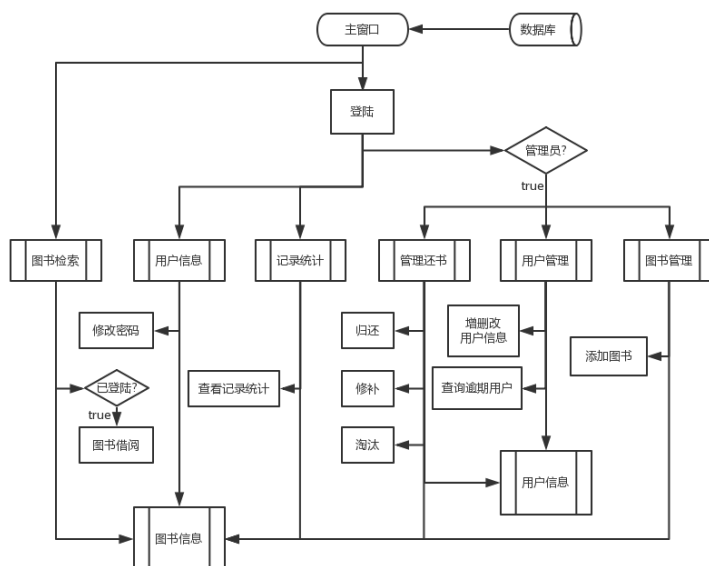
表 school: 用于存放学院信息，包括学院代码、学院名称，其中学院代码作为主键；

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
code	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		None		
school	CHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	UTF-8	utf8_gene

表 rank: 用于存放每册书在各学院借阅次数排名信息，包括 ISBN、各学院代号，其中 ISBN 作为主键；

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
isbn	CHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		None	UTF-8	utf8_gene
s201	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s202	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s203	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s204	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s205	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s206	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s207	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s208	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s209	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s210	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s211	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s212	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s213	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s214	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s216	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s218	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		
s226	INT	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None		

4. 算法描述:



主函数 main.cpp

```
#include <QApplication>
#include <QSplashScreen>
#include <iostream>
#include "public.h"
#include "mainwindow.h"

using namespace std;

TJUL_CLASS TJUL;

int main(int argc, char *argv[]) {
    TJUL.initialize_database();
    QApplication::addLibraryPath(".");
    QApplication app(argc, argv);
    QSplashScreen *splash=new QSplashScreen;
    splash->setPixmap(QPixmap(":/images/TJU_logo.png"));
    splash->show();
    for(int i=0;i<5000;i++){ //延时显示启动画面
        splash->repaint();
    }
    MainWindow w;
    w.show();
    splash->finish(&w);
    delete splash;
    return app.exec();
}

void TJUL_CLASS::user_logout() { //用户注销
    isLogin = 0;
    isAdmin = 0;
    stuid = "";
    name = "";
    num_borrowed = 0;
    num_limit = 0;
}

bool TJUL_CLASS::initialize_database() { //连接数据库
    QSqlDatabase db = QSqlDatabase::addDatabase("QODBC");
    db.setDatabaseName("Driver={MySQL ODBC 5.3 Unicode
Driver};SERVER=127.0.0.1;DATABASE=dbms;UID=root;Pwd=password");

    if (!db.open()) {
        cerr << "Error: Couldn't connect to database server!" << endl;
    }
}
```

```

        cerr << db.lastError().text().toString() << endl;
        is_connect_database = 0;
        return 0;
    }
    else {
        is_connect_database = 1;
    }

    return 1;
}

bool TJUL_CLASS::check_DatabaseConnect() { //检查数据库连接状态
    return is_connect_database;
}

bool TJUL_CLASS::check_isLogin() { //检查用户登陆状态
    return isLogin;
}

bool TJUL_CLASS::check_isAdmin() { //检查是否为管理员
    return isAdmin;
}

bool TJUL_CLASS::modify_user_book(int num) { //判断借书状态
    if (!isLogin)
        return 0;

    if (num_borrowed + num > num_limit)
        return 0;

    if (num > 0) {
        num_borrowed ++;
        QSqlQuery query("UPDATE user SET num_borrowed = num_borrowed + 1 WHERE
stuid = " + stuid);
    }
    else {
        num_borrowed --;
        QSqlQuery query("UPDATE user SET num_borrowed = num_borrowed - 1 WHERE
stuid = " + stuid);
    }

    return 1;
}

```

```

void TJUL_CLASS::set_number(int _num_borrowed, int _num_limit) { //全局变量赋值
    if (_num_borrowed >= 0)
        num_borrowed = _num_borrowed;

    if (_num_limit >= 0)
        num_limit = _num_limit;
}

```

主窗口 mainwindow.cpp



```

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <ctime>
#include <cstdlib>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    if (!TJUL.check_DatabaseConnect()) {
        QMessageBox::warning(this, tr("Connection Error"), tr("Couldn't connect
to database server!\nPlease check your network connection and try again.));
        this->close();
    }
    ui->login_stuid_label->setFocus();

    inst_user_Dialog = new user_Dialog;
    inst_book_Dialog = new book_Dialog;

```

```

    inst_rank_Dialog = new rank_Dialog;
    inst_item_Dialog = new item_Dialog;
    inst_book_return_Dialog = new book_return_Dialog;
    inst_user_manage_Dialog = new user_manage_Dialog;
    inst_book_manage_Dialog = new book_manage_Dialog;
    connect(ui->login_password_label, SIGNAL(returnPressed()), ui->login_action_button, SIGNAL(clicked()), Qt::UniqueConnection);
    connect(this, SIGNAL(signal_change_login_status()), this, SLOT(onsignal_change_login_status()));
    connect(this, SIGNAL(signal_load_user_dialog()), inst_user_Dialog, SLOT(onsignal_load_user_dialog()));
    connect(inst_book_Dialog, SIGNAL(signal_load_item(QString)), inst_item_Dialog, SLOT(onsignal_load_item(QString)));
    connect(inst_book_Dialog, SIGNAL(signal_show_dialog(int)), this, SLOT(onsignal_show_dialog(int)));
    connect(inst_rank_Dialog, SIGNAL(signal_load_item(QString)), inst_item_Dialog, SLOT(onsignal_load_item(QString)));
    connect(inst_rank_Dialog, SIGNAL(signal_show_dialog(int)), this, SLOT(onsignal_show_dialog(int)));
    connect(inst_book_manage_Dialog, SIGNAL(signal_load_item(QString)), inst_item_Dialog, SLOT(onsignal_load_item(QString)));
    connect(inst_book_manage_Dialog, SIGNAL(signal_show_dialog(int)), this, SLOT(onsignal_show_dialog(int)));

    MainWindow::onsignal_change_login_status();
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_login_action_button_clicked()
{
    QString stuid, password;
    stuid = ui->login_stuid_label->text();
    password = ui->login_password_label->text();

    if (stuid == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("学号不能为空! "));
        ui->login_stuid_label->setFocus();
        return;
    }
}

```



```

if (password == "") {
    QMessageBox::warning(this, tr("WARNING"), tr("密码不能为空! "));
    ui->login_password_label->setFocus();
    return;
}

QString query("SELECT stuid, password, name, num_borrowed, num_limit,
isadmin FROM user WHERE stuid = '" + stuid + "'");

if (query.next()) { //用户存在
    if (query.value(1) == password) { //登录成功
        TJUL.isLogin = 1;
        TJUL.stuid = stuid;
        TJUL.name = query.value(2).toString();
        TJUL.name.replace("\r", "");
        TJUL.isAdmin = query.value(5).toInt();
        TJUL.set_number(query.value(3).toInt(), query.value(4).toInt());
        QMessageBox::information(this, tr("SUCCESS"), tr("欢迎 %1 进入天津大学图
书馆! ").arg(TJUL.name));
        ui->login_stuid_label->setText("");
        ui->login_password_label->setText("");
        emit signal_change_login_status();
    }
    else { //密码错误
        QMessageBox::warning(this, tr("ERROR"), tr("密码与账号不匹配, 请重试!
"));
        ui->login_password_label->setText("");
        ui->login_password_label->setFocus();
    }
}

else { //用户不存在
    QMessageBox::warning(this, tr("ERROR"), tr("该用户不存在! "));
    ui->login_stuid_label->setText("");
    ui->login_password_label->setText("");
    ui->login_stuid_label->setFocus();
    return;
}

}

void MainWindow::on_main_logout_Button_clicked() {
    if (TJUL.check_isLogin()) {
        TJUL.user_logout();
        MainWindow::onsignal_change_login_status();
    }
}

```

```

    }
}

void MainWindow::onsignal_change_login_status(){
    if (TJUL.check_isLogin()) {
        ui->login_groupBox->hide();
        char *t = (char*)malloc(sizeof(char)*30);
        time_t now;
        time(&now);
        strftime(t, 30, "%Y年%m月%d日 %H:%M:%S", localtime(&now));
        ui->main_user_status_label->setText(tr("账号: %1 【%2】 登录时间:
%3").arg(TJUL.stuid).arg(TJUL.name).arg(t));
        ui->main_userinfo_Button->setEnabled(true);
        ui->main_ranklist_Button->setEnabled(true);
        if (TJUL.isAdmin) { //开启管理员权限
            ui->main_returnBook_Button->setEnabled(true);
            ui->main_manageBook_Button->setEnabled(true);
            ui->main_manageUser_Button->setEnabled(true);
        }
    }
    else { //关闭登录权限
        ui->login_groupBox->show();
        ui->main_user_status_label->setText("");
        ui->main_userinfo_Button->setEnabled(false);
        ui->main_ranklist_Button->setEnabled(false);
        ui->main_returnBook_Button->setEnabled(false);
        ui->main_manageBook_Button->setEnabled(false);
        ui->main_manageUser_Button->setEnabled(false);
    }
}

void MainWindow::on_main_userinfo_Button_clicked()
{
    inst_user_Dialog->show();
    emit signal_load_user_dialog();
}

void MainWindow::on_main_bookSearch_Button_clicked()
{
    inst_book_Dialog->show();
}

void MainWindow::on_main_ranklist_Button_clicked()
{

```

```

        inst_rank_Dialog->show();
    }

void MainWindow::onsignal_show_dialog(int dialog_id)
{
    switch (dialog_id) {
        case 1: inst_item_Dialog->show();
    }
}

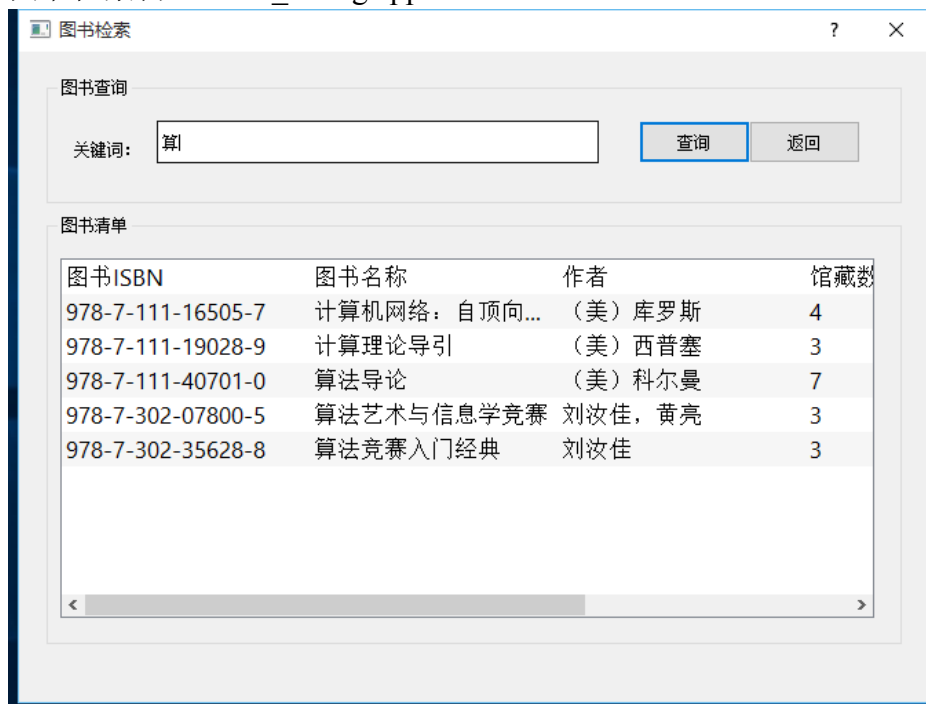
void MainWindow::on_main_returnBook_Button_clicked()
{
    inst_book_return_Dialog->show();
}

void MainWindow::on_main_manageUser_Button_clicked()
{
    inst_user_manage_Dialog->show();
}

void MainWindow::on_main_manageBook_Button_clicked()
{
    inst_book_manage_Dialog->show();
}

```

图书检索窗口 book_dialog.cpp



```
#include "book_dialog.h"
```

```
#include "ui_book_dialog.h"
```

```
book_Dialog::book_Dialog(QWidget *parent) :  
    QDialog(parent),  
    ui(new Ui::book_Dialog)  
{  
    ui->setupUi(this);  
    line=1;  
    ui->keyword_edit->setText("");  
    booklistModel=new QStandardItemModel(0,5,this);  
    booklistModel->insertRow(0);  
    booklistModel->setData(booklistModel->index(0,0), tr("图书ISBN"));  
    booklistModel->setData(booklistModel->index(0,1), tr("图书名称"));  
    booklistModel->setData(booklistModel->index(0,2), tr("作者"));  
    booklistModel->setData(booklistModel->index(0,3), tr("馆藏数量"));  
    booklistModel->setData(booklistModel->index(0,4), tr("出版社"));  
    ui->borrowedbookview->setModel(booklistModel);  
}
```

```
book_Dialog::~book_Dialog()  
{  
    delete ui;  
}
```

```
void book_Dialog::on_return_Button_clicked()  
{  
    for(int i=line;i>0;i--)  
        booklistModel->removeRow(i);  
    ui->keyword_edit->setText("");  
    this->close();  
}
```

```
void book_Dialog::on_search_Button_clicked()  
{  
    int i;  
    for(i=line;i>0;i--)  
        booklistModel->removeRow(i);  
    QString keyword = ui->keyword_edit->text();  
  
    QSqlQuery query("SELECT isbn, title, author, num_total, pub_press FROM book  
WHERE title LIKE '%" + keyword + "%' ORDER BY isbn");  
  
    for (i=1;query.next();i++) {  
        booklistModel->insertRow(i);
```

```

        book_isbn[i] = query.value(0).toString();
        booklistModel->setData(booklistModel->index(i,0),
query.value(0).toString());
        booklistModel->setData(booklistModel->index(i,1),
query.value(1).toString());
        booklistModel->setData(booklistModel->index(i,2),
query.value(2).toString());
        booklistModel->setData(booklistModel->index(i,3),
query.value(3).toString());
        booklistModel->setData(booklistModel->index(i,4),
query.value(4).toString());
    }
    if(i==1)
        QMessageBox::warning(this, tr("ERROR"), tr("未检索到符合条件的记录!"));
    line=i-1;
}

void book_Dialog::on_borrowedbookview_clicked(const QModelIndex &index)
{
    if (index.row() == 0) return;
    emit signal_load_item(book_isbn[index.row()]);
    emit signal_show_dialog(1);
}

```

用户信息窗口 user_dialog.cpp

用户信息

姓名: 梁振铎 原始密码:

学号: 3013216096 新密码:

单位: 计算机学院 确认密码:

当前持有: 1

借书上限: 7

当前已借

图书编号	书目	借阅时间	剩余天数
978-7-111-21382-6(1)	Java编程思想	2015-10-30T19:49:30	26

```

#include "user_dialog.h"
#include "ui_user_dialog.h"

user_Dialog::user_Dialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::user_Dialog)
{
    ui->setupUi(this);
}

user_Dialog::~user_Dialog()
{
    delete ui;
}

void user_Dialog::on_return_Button_clicked()
{
    this->close();
}

void user_Dialog::on_change_Button_clicked()
{
    QString old_password,new_password,confirm_password;
    old_password = ui->old_password->text();
    new_password = ui->new_password->text();
    confirm_password = ui->confirm_password->text();
    QSqlQuery query("SELECT password FROM user WHERE stuid = "+TJUL.stuid);
    query.next();
    if(old_password!=query.value(0)){
        QMessageBox::warning(this, tr("ERROR"), tr("原始密码错误! "));
        ui->old_password->setText("");
        ui->old_password->setFocus();
        return;
    }
    if(new_password==""){
        QMessageBox::warning(this, tr("WARNING"), tr("密码不能为空! "));
        ui->new_password->setFocus();
        return;
    }
    if(confirm_password!=new_password){
        QMessageBox::warning(this, tr("WARNING"), tr("请重新确认密码! "));
        ui->confirm_password->setText("");
        ui->confirm_password->setFocus();
    }
}

```

```

        return;
    }
    QSqlQuery("UPDATE user SET password = " + new_password + " WHERE stuid="+
TJUL.stuid);
    QMessageBox::warning(this, tr("SUCCESS"), tr("密码修改成功! "));
    ui->new_password->setText("");
    ui->old_password->setText("");
    ui->confirm_password->setText("");
    ui->old_password->setFocus();
}

void user_Dialog::onsignal_load_user_dialog()
{
    QSqlQuery query_user("SELECT user.name, user.stuid, school.school,
user.num_borrowed, user.num_limit FROM user LEFT JOIN school ON school.code =
user.school WHERE stuid = '" + TJUL.stuid + "'");
    if (!query_user.next()) {
        QMessageBox::warning(this, tr("ERROR"), tr("未检索到该记录! "));
        this->close();
        return;
    }

    QString label_html;
    label_html = tr("<p>姓名: %1</p><p>学号: %2</p><p>单位: %3</p><p>当前持有:
%4</p><p>借书上限:
%5</p>").arg(query_user.value(0).toString()).arg(query_user.value(1).toString()
).arg(query_user.value(2).toString()).arg(query_user.value(3).toString()).arg(q
uery_user.value(4).toString());

    ui->user_info_label->setText(label_html);

    QSqlQuery(tr("UPDATE record SET overtime = TIMESTAMPDIF(DAY, NOW(),
time_deadline) WHERE stuid = %1 AND status = 0").arg(TJUL.stuid));
    QSqlQuery query_record(tr("SELECT record.id, title, time_borrow, overtime
FROM record LEFT JOIN item ON (item.id = record.id) LEFT JOIN book ON
(book.isbn = item.isbn) WHERE (record.stuid = %1 AND record.status=0) ORDER BY
record.id DESC").arg(TJUL.stuid));

    booklistModel=new QStandardItemModel(0,4,this);
    booklistModel->insertRow(0);
    booklistModel->setData(booklistModel->index(0,0), tr("图书编号"));
    booklistModel->setData(booklistModel->index(0,1), tr("书目"));
    booklistModel->setData(booklistModel->index(0,2), tr("借阅时间"));

```

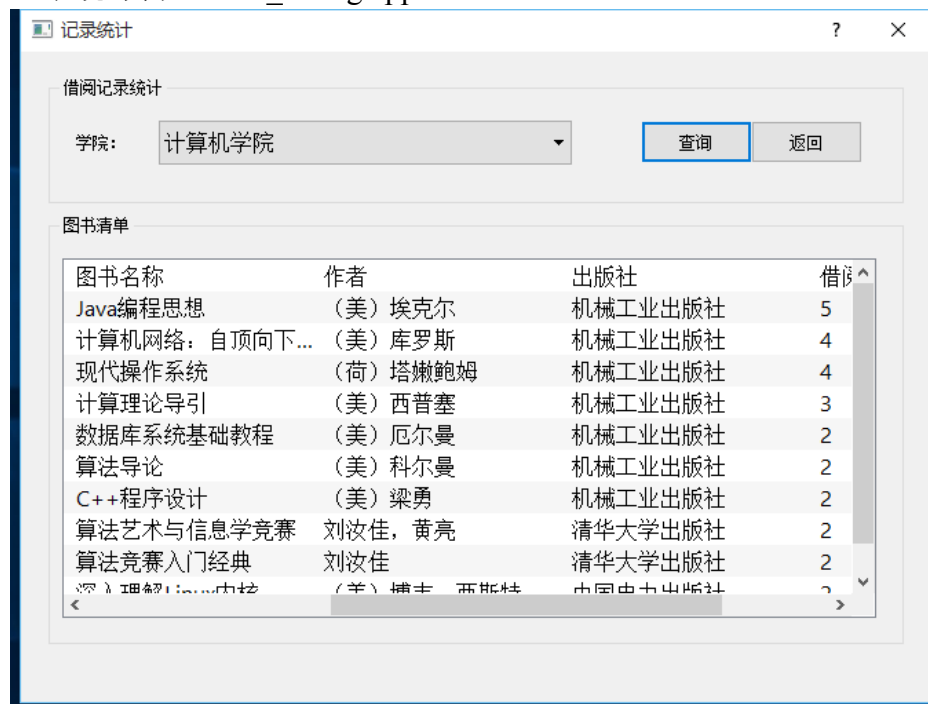
```

booklistModel->setData(booklistModel->index(0,3), tr("剩余天数"));

for(int i=1;query_record.next();i++) {
    booklistModel->insertRow(i);
    booklistModel->setData(booklistModel->index(i,0),
query_record.value(0).toString());
    booklistModel->setData(booklistModel->index(i,1),
query_record.value(1).toString());
    booklistModel->setData(booklistModel->index(i,2),
query_record.value(2).toString());
    booklistModel->setData(booklistModel->index(i,3),
query_record.value(3).toInt() < 0 ? tr("已过期") :
query_record.value(3).toString());
}
ui->borrowedbookview->setModel(booklistModel);
}

```

记录统计窗口 rank_dialog.cpp



```

#include "rank_dialog.h"
#include "ui_rank_dialog.h"

```

```

rank_Dialog::rank_Dialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::rank_Dialog)
{
    ui->setupUi(this);
    line=1;
}

```



```

booklistModel=new QStandardItemModel(0,6,this);
booklistModel->insertRow(0);
booklistModel->setData(booklistModel->index(0,0), tr("排名"));
booklistModel->setData(booklistModel->index(0,1), tr("ISBN"));
booklistModel->setData(booklistModel->index(0,2), tr("图书名称"));
booklistModel->setData(booklistModel->index(0,3), tr("作者"));
booklistModel->setData(booklistModel->index(0,4), tr("出版社"));
booklistModel->setData(booklistModel->index(0,5), tr("借阅次数"));
ui->bookrankview->setModel(booklistModel);
}

rank_Dialog::~rank_Dialog()
{
    delete ui;
}

void rank_Dialog::on_return_Button_clicked()
{
    for(int i=line;i>0;i--)
        booklistModel->removeRow(i);
    ui->school_select->setCurrentIndex(-1);
    this->close();
}

void rank_Dialog::on_search_Button_clicked()
{
    int i;
    for(i=line;i>0;i--)
        booklistModel->removeRow(i);

    QString school=ui->school_select->currentText();
    if (school == "" ) { //未选择学院
        QMessageBox::warning(this, tr("WARNING"), tr("请选择学院! "));
        return;
    }

    QSqlQuery query_depart("SELECT code FROM school WHERE school =
''+school+\"\\r\"");
    query_depart.next();
    QString tmp=query_depart.value(0).toString();
    QSqlQuery query("SELECT book.isbn, book.title, book.author, book.pub_press,
rank.s"+tmp+" FROM book LEFT JOIN rank ON rank.isbn = book.isbn WHERE
rank.s"+tmp+">0 ORDER BY rank.s" + tmp + " DESC");

```

```

        for(i=1;query.next();i++) {
            booklistModel->insertRow(i);
            book_isbn[i] = query.value(0).toString();
            booklistModel->setData(booklistModel->index(i,0), i);
            booklistModel->setData(booklistModel->index(i,1),
            query.value(0).toString());
            booklistModel->setData(booklistModel->index(i,2),
            query.value(1).toString());
            booklistModel->setData(booklistModel->index(i,3),
            query.value(2).toString());
            booklistModel->setData(booklistModel->index(i,4),
            query.value(3).toString());
            booklistModel->setData(booklistModel->index(i,5),
            query.value(4).toString());
        }
        line=i-1;
    }
}

void rank_Dialog::on_bookrankview_clicked(const QModelIndex &index)
{
    if (index.row() == 0) return;
    emit signal_load_item(book_isbn[index.row()]);
    emit signal_show_dialog(1);
}

```

还书窗口 book_return_dialog.cpp

管理还书

管理选项

学号: 3013216104 查询单用户

单册编号: 单册归还

图书编号	书目	借阅时间	归还时间
978-7-111-40701-0(1)	算法导论	2014-12-30T19:50:09	未归还
978-7-111-21382-6(2)	Java编程思想	2015-10-30T19:50:03	未归还
978-7-111-19028-9(1)	计算理论导引	2015-08-30T16:51:32	2015-
978-7-302-35628-8(1)	算法竞赛入门经典	2015-09-30T16:51:24	2015-
978-7-111-42505-2(0)	C++程序设计	2015-10-30T16:51:00	2015-
978-7-111-16505-7(3)	计算机网络：自顶向...	2015-10-23T22:21:38	2015-
978-7-111-25544-4(0)	现代操作系统	2015-10-23T22:21:30	2015-
978-7-302-35628-8(0)	算法竞赛入门经典	2015-10-23T22:21:25	2015-

```
#include "book_return_dialog.h"
```

```

#include "ui_book_return_dialog.h"

book_return_Dialog::book_return_Dialog(QWidget *parent):
    QDialog(parent),
    ui(new Ui::book_return_Dialog)
{
    ui->setupUi(this);

    QStandardItemModel* booklistModel=new QStandardItemModel(0,4,this);
    booklistModel->insertRow(0);
    booklistModel->setData(booklistModel->index(0,0), tr("图书编号"));
    booklistModel->setData(booklistModel->index(0,1), tr("书目"));
    booklistModel->setData(booklistModel->index(0,2), tr("借阅时间"));
    booklistModel->setData(booklistModel->index(0,3), tr("归还时间"));

    ui->borrowedbookview->setModel(booklistModel);
}

book_return_Dialog::~book_return_Dialog()
{
    delete ui;
}

void book_return_Dialog::on_search_Button_clicked()
{
    if (ui->stuid_label->text() == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请输入学号! "));
        return;
    }

    QString stuid = ui->stuid_label->text();
    last_stuid = stuid;

    book_return_Dialog::load_book_list(last_stuid);
}

void book_return_Dialog::action_book_return(QString id)
{
    QSqlQuery query_item("SELECT uuid, item.status, stuid, item.isbn FROM item
LEFT JOIN record ON (record.status = 0 AND record.id = item.id) WHERE item.id =
''+ id +''");

    if (!query_item.next()) {
        QMessageBox::warning(this, tr("ERROR"), tr("未检索到该记录! "));
    }
}

```

```

        return;
    }

    if (query_item.value(1).toInt() == 1) {
        QMessageBox::warning(this, tr("ERROR"), tr("该册图书已在馆! "));
        return;
    }

    QMessageBox box;
    box.setWindowTitle("CHOOSE");
    box.setText(tr("请选择要对该书进行的操作: "));
    box.setStandardButtons(QMessageBox::Save | QMessageBox::Discard |
QMessageBox::Cancel);
    box.setButtonText(QMessageBox::Save, QString("修补"));
    box.setButtonText(QMessageBox::Discard, QString("淘汰"));
    box.setButtonText(QMessageBox::Cancel, QString("归还"));
    int msg_ret = box.exec();
    if (msg_ret == QMessageBox::Cancel) {
        if(query_item.value(1).toInt() == 0) { //检测图书状态
            QSqlQuery(tr("UPDATE record SET status = 1, time_return = NOW() WHERE
uuid = %1").arg(query_item.value(0).toInt()));
            QSqlQuery(tr("UPDATE user SET num_borrowed = num_borrowed - 1 WHERE
stuid = %1").arg(query_item.value(2).toString()));
            QSqlQuery(tr("UPDATE item SET status = 1 WHERE id = '%1'").arg(id));

            QMessageBox::information(this, tr("SUCCESS"), tr("图书已归还入库! "));
        }
        else if(query_item.value(1).toInt() == -1) { //图书归还入库
            QSqlQuery(tr("UPDATE item SET status = 1 WHERE id = '%1'").arg(id));

            QMessageBox::information(this, tr("SUCCESS"), tr("图书已修补完成入库!
"));
        }
    }
    else if (msg_ret == QMessageBox::Save) { //图书修补
        QSqlQuery(tr("UPDATE record SET status = 1, time_return = NOW() WHERE
uuid = %1").arg(query_item.value(0).toInt()));
        QSqlQuery(tr("UPDATE user SET num_borrowed = num_borrowed - 1 WHERE
stuid = %1").arg(query_item.value(2).toString()));
        QSqlQuery(tr("UPDATE item SET status = -1 WHERE id = '%1'").arg(id));

        QMessageBox::information(this, tr("SUCCESS"), tr("图书已登记修补! "));
    }
    else if (msg_ret == QMessageBox::Discard) { //图书淘汰

```

```

        QSqlQuery(tr("UPDATE record SET status = 1, time_return = NOW() WHERE
uuid = %1").arg(query_item.value(0).toInt()));
        QSqlQuery(tr("UPDATE user SET num_borrowed = num_borrowed - 1 WHERE
stuid = '%1'").arg(query_item.value(2).toString()));
        QSqlQuery(tr("UPDATE book SET num_total = num_total - 1 WHERE isbn =
'%1'").arg(query_item.value(3).toString()));
        QSqlQuery(tr("DELETE FROM item WHERE id = '%1'").arg(id));

        QMessageBox::information(this, tr("SUCCESS"), tr("图书已淘汰! "));
    }
    book_return_Dialog::load_book_list(last_stuid);
}

void book_return_Dialog::on_borrowedbookview_clicked(const QModelIndex &index)
{
    if (index.row() < 1 ) return;

    book_return_Dialog::action_book_return(book_id_list[index.row()]);
}

void book_return_Dialog::on_return_Button_clicked()
{
    if (ui->bookId_label->text() == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请填写图书编号! "));
        return;
    }

    book_return_Dialog::action_book_return(ui->bookId_label->text());
}

void book_return_Dialog::load_book_list(QString stuid)
{
    QStandardItemModel* booklistModel=new QStandardItemModel(0,4,this);
    booklistModel->insertRow(0);
    booklistModel->setData(booklistModel->index(0,0), tr("图书编号"));
    booklistModel->setData(booklistModel->index(0,1), tr("书目"));
    booklistModel->setData(booklistModel->index(0,2), tr("借阅时间"));
    booklistModel->setData(booklistModel->index(0,3), tr("归还时间"));
    ui->borrowedbookview->setModel(booklistModel);

    QSqlQuery query_user("SELECT stuid, name, school, num_borrowed, num_limit
FROM user WHERE stuid LIKE '" + stuid + "%'");

    if (!query_user.next()) {

```

```

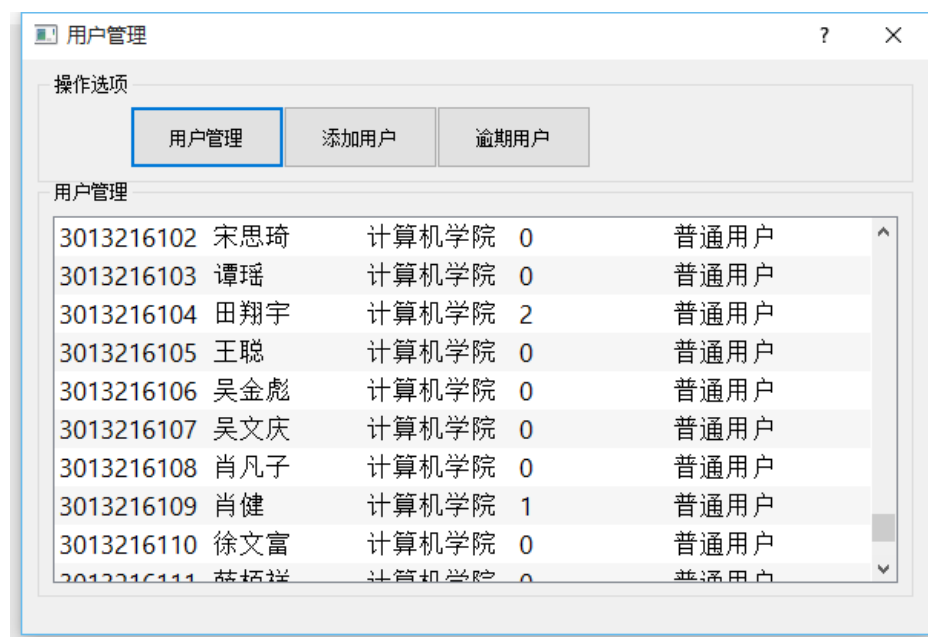
        QMessageBox::warning(this, tr("ERROR"), tr("该用户不存在! "));
        return;
    }

    ui->stuid_label->setText(query_user.value(0).toString());
    QSqlQuery query_record("SELECT record.id, book.title, record.time_borrow,
record.time_return, record.status FROM record LEFT JOIN item ON item.id =
record.id LEFT JOIN book ON book.isbn = item.isbn WHERE record.stuid = '"+ stuid
+" ORDER BY record.status, record.uuid DESC");

    for(int i=1;query_record.next();i++) {
        booklistModel->insertRow(i);
        book_id_list[i] = query_record.value(0).toString();
        booklistModel->setData(booklistModel->index(i,0),
query_record.value(0).toString());
        booklistModel->setData(booklistModel->index(i,1),
query_record.value(1).toString());
        booklistModel->setData(booklistModel->index(i,2),
query_record.value(2).toString());
        booklistModel->setData(booklistModel->index(i,3),
query_record.value(4).toInt() == 0 ? tr("未归还")
:query_record.value(3).toString());
    }
}

```

用户管理窗口 user_manage_dialog.cpp



```

#include "user_manage_dialog.h"
#include "ui_user_manage_dialog.h"

```

```

user_manage_Dialog::user_manage_Dialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::user_manage_Dialog)
{
    ui->setupUi(this);
    user_manage_Dialog::on_guide_user_Button_clicked();
}

user_manage_Dialog::~user_manage_Dialog()
{
    delete ui;
}

void user_manage_Dialog::on_newuser_add_Button_clicked()
{
    QString stuid = ui->newuser_stuid->text();
    QString name = ui->newuser_name->text();
    QString tmp;
    tmp[0]=stuid[4],tmp[1]=stuid[5],tmp[2]=stuid[6];
    int school = tmp.toInt();
    QString password = ui->newuser_password->text();
    QString telnum = ui->newuser_telnum->text();
    int num_limit = ui->newuser_num_limit->currentText().toInt();
    int isAdmin = ui->newuser_isadmin->currentText() == tr("管理员");

    if (stuid == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请填写学号! "));
        ui->newuser_stuid->setFocus();
        return;
    }
    if (name == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请填写姓名! "));
        ui->newuser_name->setFocus();
        return;
    }
    if (password == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请填写密码! "));
        ui->newuser_password->setFocus();
        return;
    }
    if (telnum == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请填写联系电话! "));
        ui->newuser_telnum->setFocus();
    }
}

```

```

        return;
    }

    QSqlQuery query_user(tr("SELECT * FROM user WHERE stuid = %1").arg(stuid));

    if (query_user.next()) {
        QMessageBox::warning(this, tr("ERROR"), tr("该用户已存在! "));
        return;
    }

    QSqlQuery(tr("INSERT INTO user(stuid, name, password, school, num_limit,
num_borrowed, isAdmin, telnum)
VALUES('%1', '%2', '%3', '%4', %5, %6, '%7')").arg(stuid).arg(name).arg(password).a
rg(school).arg(num_limit).arg(isAdmin).arg(telnum));

    ui->newuser_stuid->setText("");
    ui->newuser_name->setText("");
    ui->newuser_password->setText("");
    ui->newuser_telnum->setText("");
    ui->newuser_num_limit->setCurrentIndex(2);

    QMessageBox::information(this, tr("SUCCESS"), tr("新用户添加完成! "));
}

void user_manage_Dialog::on_guide_user_Button_clicked()
{
    QSqlQuery query("SELECT stuid, name, school.school, num_borrowed, isAdmin
FROM user LEFT JOIN school ON (school.code = user.school) ORDER BY stuid");

    QStandardItemModel* listModel=new QStandardItemModel(0,5,this);
    listModel->insertRow(0);
    listModel->setData(listModel->index(0,0), tr("学号"));
    listModel->setData(listModel->index(0,1), tr("姓名"));
    listModel->setData(listModel->index(0,2), tr("单位"));
    listModel->setData(listModel->index(0,3), tr("借书量"));
    listModel->setData(listModel->index(0,4), tr("身份"));

    int i(1);

    while (query.next()) {
        listModel->insertRow(i);
        user_list_stuid[i] = query.value(0).toString();
        listModel->setData(listModel->index(i,0), query.value(0).toString());
    }
}

```



```

        listModel->setData(listModel->index(i,1), query.value(1).toString());
        listModel->setData(listModel->index(i,2), query.value(2).toString());
        listModel->setData(listModel->index(i,3), query.value(3).toString());
        listModel->setData(listModel->index(i,4), query.value(4).toInt() == 1 ?
tr("管理员") : tr("普通用户"));

```

```

        i++;
    }

```

```

    ui->userview->setModel(listModel);
    ui->group_user->show();
    ui->group_newuser->hide();
    ui->group_overuser->hide();
    ui->group_manageuser->hide();
}

```

```

void user_manage_Dialog::on_guide_overuser_Button_clicked() {
    QSqlQuery(tr("UPDATE record SET overtime = TIMESTAMPDIF(DAY,
time_deadline, NOW()) WHERE status = 0"));
    QSqlQuery query_record(tr("SELECT title, name, record.stuid, telnum,
overtime, record.status FROM record LEFT JOIN user ON (user.stuid =
record.stuid) LEFT JOIN item ON (item.id = record.id) LEFT JOIN book ON
(item.isbn = book.isbn) WHERE (record.status = 0 AND overtime > 0) ORDER BY
overtime DESC"));
}

```

```

QStandardItemModel* listModel=new QStandardItemModel(0,5,this);
listModel->insertRow(0);
listModel->setData(listModel->index(0,0), tr("书目"));
listModel->setData(listModel->index(0,1), tr("读者姓名"));
listModel->setData(listModel->index(0,2), tr("学号"));
listModel->setData(listModel->index(0,3), tr("联系电话"));
listModel->setData(listModel->index(0,4), tr("逾期天数"));

```

```

int i(1);
while (query_record.next()) {
    listModel->insertRow(i);
    user_list_stuid[i] = query_record.value(1).toString();
    listModel->setData(listModel->index(i,0),
query_record.value(0).toString());
    listModel->setData(listModel->index(i,1),
query_record.value(1).toString());
    listModel->setData(listModel->index(i,2),
query_record.value(2).toString());
    listModel->setData(listModel->index(i,3),

```

```

query_record.value(3).toString());
        listModel->setData(listModel->index(i,4),
query_record.value(4).toString());

        i++;
    }

    ui->overuserview->setModel(listModel);
    ui->group_user->hide();
    ui->group_newuser->hide();
    ui->group_overuser->show();
    ui->group_manageuser->hide();
}

void user_manage_Dialog::on_guide_newuser_Button_clicked()
{
    ui->group_newuser->show();
    ui->group_overuser->hide();
    ui->group_user->hide();
    ui->group_manageuser->hide();
}

void user_manage_Dialog::on_manageuser_modify_Button_clicked()
{
    QString stuid = ui->manage_stuid->text();
    int num_limit = ui->manage_num_limit->currentText().toInt();
    int isAdmin = ui->manage_isadmin->currentText() == tr("管理员") ? 1 : 0;
    QString name = ui->manage_name->text();
    QString telnum = ui->manage_telnum->text();
    QString password = ui->manage_password->text();

    if (name == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请填写姓名! "));
        ui->newuser_name->setFocus();
        return;
    }

    if (password == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请填写密码! "));
        ui->newuser_password->setFocus();
        return;
    }

    if (telnum == "") {
        QMessageBox::warning(this, tr("WARNING"), tr("请填写联系电话! "));
        ui->newuser_telnum->setFocus();
    }
}

```

```

        return;
    }

    QSqlQuery (tr("UPDATE user SET name = '%1', telnum = '%2', password = '%3',
num_limit = %4, isadmin = %5 WHERE stuid =
'%6'").arg(name).arg(telnum).arg(password).arg(num_limit).arg(isAdmin).arg(stui
d));

    QMessageBox::information(this, tr("SUCCESS"), tr("用户信息修改成功! "));
    user_manage_Dialog::on_guide_user_Button_clicked();
}

void user_manage_Dialog::on_userview_clicked(const QModelIndex &index)
{

    if (index.row() < 1) return;
    QSqlQuery query(tr("SELECT name, password, telnum, num_limit FROM user
WHERE stuid = '%1'").arg(user_list_stuid[index.row()]));

    if (!query.next()) {
        QMessageBox::information(this, tr("ERROR"), tr("未检索到该记录! "));
        return;
    }

    ui->manage_name->setText(query.value(0).toString());
    ui->manage_password->setText(query.value(1).toString());
    ui->manage_telnum->setText(query.value(2).toString());
    ui->manage_num_limit->setCurrentText(query.value(3).toString());
    ui->manage_stuid->setText(tr("%1").arg(user_list_stuid[index.row()]));

    ui->group_manageuser->show();
    ui->group_newuser->hide();
    ui->group_user->hide();
}

void user_manage_Dialog::on_manageuser_delete_Button_clicked()
{
    QString stuid = ui->manage_stuid->text();
    if(TJUL.stuid==stuid){
        QMessageBox::information(this, tr("ERROR"), tr("无法删除已登录用户! "));
        return;
    }

    QSqlQuery(tr("DELETE FROM user WHERE stuid = '%1'").arg(stuid));

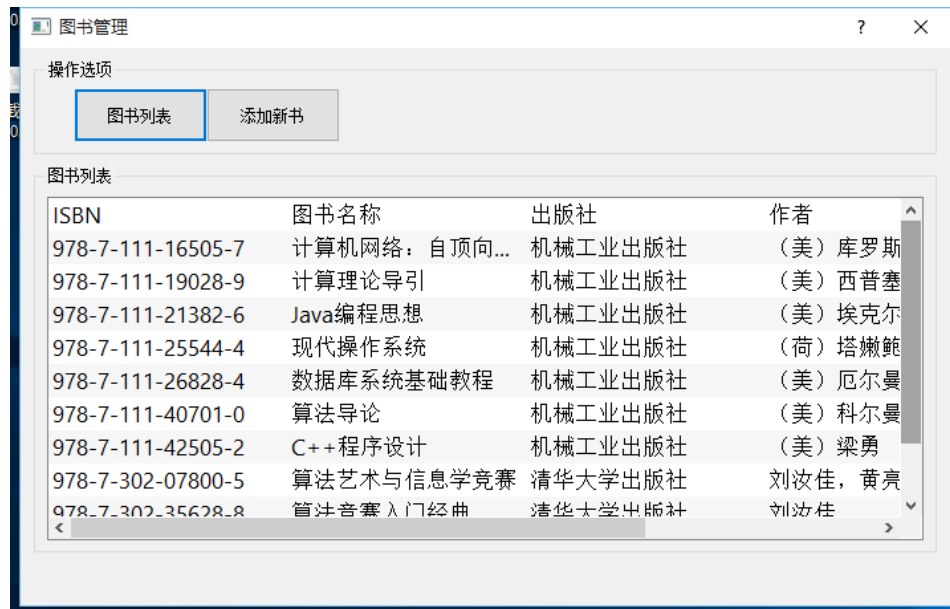
```

```

    QMessageBox::information(this, tr("SUCCESS"), tr("用户 %1 已被成功删除!
").arg(stuid));
    user_manage_dialog::on_guide_user_Button_clicked();
}

```

图书管理窗口 book_manage_dialog.cpp



```

#include "book_manage_dialog.h"
#include "ui_book_manage_dialog.h"

book_manage_dialog::book_manage_dialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::book_manage_dialog)
{
    ui->setupUi(this);
    book_manage_dialog::on_guide_booklist_clicked();
}

book_manage_dialog::~book_manage_dialog()
{
    delete ui;
}

void book_manage_dialog::on_single_add_clicked()
{
    QString isbn = ui->single_isbn->text();
    QString title = ui->single_title->text();
    QString author = ui->single_author->text();
    QString type = ui->single_type->text();
}

```

```

QString press = ui->single_press->text();
double price = ui->single_price->text().toDouble();
int year = ui->single_year->text().toInt();
int num = ui->single_num->currentText().toInt();

if(isbn != ""){
    QSqlQuery query(tr("SELECT title, author, type, pub_press, price,
pub_year, num_total FROM book WHERE isbn = '%1'").arg(isbn));
    if (query.next()) { //图书已存在
        title = query.value(0).toString();
        author = query.value(1).toString();
        type = query.value(2).toString();
        press = query.value(3).toString();
        price = query.value(4).toDouble();
        year = query.value(5).toInt();
        int cur_num = query.value(6).toInt();
        int msg_ret = QMessageBox::information(this, tr("WARNING"), tr("该图书
当前馆藏%1本, 是否继续添加? ").arg(cur_num), QMessageBox::Yes | QMessageBox::No);
        if (msg_ret == QMessageBox::No) return;
        if(num + cur_num > 10) {
            QMessageBox::warning(this, tr("ERROR"), tr("图书超出数量限制, 请重新选
择数量! "));
            ui->single_num->setFocus();
            return;
        }
        for (int i=0;num&&i<10;i++){
            QSqlQuery query_item(tr("SELECT * FROM item WHERE id =
'%1(%2)'").arg(isbn).arg(i));
            if(!query_item.next()){
                QSqlQuery(tr("INSERT INTO item(id, isbn, status)
VALUES('%1(%2)', '%3', 1)").arg(isbn).arg(i).arg(isbn));
                num--;
            }
        }
        QSqlQuery(tr("UPDATE book SET num_total = num_total + %1 WHERE isbn =
'%2'").arg(ui->single_num->currentText()).arg(isbn));
        QMessageBox::information(this, tr("SUCCESS"), tr("图书添加成功! "));

        ui->single_num->setCurrentIndex(0);
        ui->single_price->setText("");
        ui->single_press->setText("");
        ui->single_year->setText("");
        ui->single_title->setText("");
        ui->single_type->setText("");
    }
}

```

```

        ui->single_author->setText("");
        ui->single_isbn->setText("");
        return;
    }
}

//图书不存在
if (title == "") {
    QMessageBox::warning(this, tr("WARNING"), tr("请填写书目! "));
    ui->single_title->setFocus();
    return;
}
if (isbn == "") {
    QMessageBox::warning(this, tr("WARNING"), tr("请填写ISBN! "));
    ui->single_isbn->setFocus();
    return;
}
if (author == "") {
    QMessageBox::warning(this, tr("WARNING"), tr("请填写作者! "));
    ui->single_author->setFocus();
    return;
}
if (type == "") {
    QMessageBox::warning(this, tr("WARNING"), tr("请填写类型! "));
    ui->single_type->setFocus();
    return;
}
if (press == "") {
    QMessageBox::warning(this, tr("WARNING"), tr("请填写出版社! "));
    ui->single_press->setFocus();
    return;
}
if (price <= 0 || price > 4096) {
    QMessageBox::warning(this, tr("WARNING"), tr("请填写有效单价! "));
    ui->single_price->setFocus();
    return;
}
if (year < 1895 || year > 2095) {
    QMessageBox::warning(this, tr("WARNING"), tr("请填写有效年份! "));
    ui->single_year->setFocus();
    return;
}
}

```

```

        QSqlQuery(tr("INSERT INTO book(isbn, title, type, pub_press, pub_year,
author, price, num_total) VALUES('%1', '%2', '%3', '%4', %5, '%6', %7,
%8)").arg(isbn).arg(title).arg(type).arg(press).arg(year).arg(author).arg(price)
).arg(num));
        QSqlQuery(tr("INSERT INTO rank(isbn) VALUES('%1')").arg(isbn));
        for (int i=0;i<num;i++)
            QSqlQuery(tr("INSERT INTO item(id, isbn, status)
VALUES('%1(%2)', '%3', 1)").arg(isbn).arg(i).arg(isbn));
        QMessageBox::information(this, tr("SUCCESS"), tr("图书添加成功! "));

        ui->single_num->setCurrentIndex(0);
        ui->single_price->setText("");
        ui->single_press->setText("");
        ui->single_year->setText("");
        ui->single_title->setText("");
        ui->single_type->setText("");
        ui->single_author->setText("");
        ui->single_isbn->setText("");
    }

void book_manage_Dialog::on_guide_booklist_clicked()
{
    QSqlQuery query("SELECT isbn, title, type, pub_press, pub_year, author,
price, num_total FROM book ORDER BY isbn");

    QStandardItemModel* listModel=new QStandardItemModel(0,5,this);
    listModel->insertRow(0);
    listModel->setData(listModel->index(0,0), tr("ISBN"));
    listModel->setData(listModel->index(0,1), tr("图书名称"));
    listModel->setData(listModel->index(0,2), tr("出版社"));
    listModel->setData(listModel->index(0,3), tr("作者"));
    listModel->setData(listModel->index(0,4), tr("数量"));

    for (int i=1;query.next();i++) {
        listModel->insertRow(i);
        book_isbn[i] = query.value(0).toString();
        listModel->setData(listModel->index(i,0), query.value(0).toString());
        listModel->setData(listModel->index(i,1), query.value(1).toString());
        listModel->setData(listModel->index(i,2), query.value(3).toString());
        listModel->setData(listModel->index(i,3), query.value(5).toString());
        listModel->setData(listModel->index(i,4), query.value(7).toString());
    }

    ui->userview->setModel(listModel);

```

```

        ui->group_booklist->show();
        ui->group_add_single->hide();
    }

    void book_manage_Dialog::on_guide_add_single_clicked()
    {
        ui->group_add_single->show();
        ui->group_booklist->hide();
    }

    void book_manage_Dialog::on_userview_clicked(const QModelIndex &index)
    {
        if (index.row() < 1) return;

        emit signal_load_item(book_isbn[index.row()]);
        emit signal_show_dialog(1);
    }

```

5. 实习心得:

图书管理系统是一个常见的数据管理系统。本系统满足了问题描述的所有基本功能，并添加了一些额外的功能。在完成这个系统的过程中，我发现了开发一个完整的项目与开发简单控制台程序的不同，也遇到了许多困难。在开发中我逐渐掌握了 Qt 项目的开发技巧，并熟悉了一些数据库的查询语句，通过实践收获了一些课堂上较少涉及的知识。