

“语言课程设计” 报告

姓名	韩少婷	程序	满分 优 良 中 及格 不及格
学号	57123118	文档	满分 优 良 中 及格 不及格
专业	网络空间安全	平时	满分 优 良 中 及格 不及格
起始日期	2024 年 8 月	总评	
终止日期	2024 年 9 月		
设计文档			
设计任务	图书室管理系统 该系统需创建和管理以下信息：1、书籍信息：书名、书目编号、作者名、出版日期、出版社、库存册数、登记号数据集；2、每册书的登记信息：登记号、是否借出、借阅日期、借书证号。 系统功能要求如下： 1. 创建和管理描述每本书籍的对象； 2. 创建和管理描述每册书登记信息的对像； 3. 增加和删除书籍； 4. 借书操作：读者提供书籍信息（书名或书目编号），检查该书籍是否可借（有没有没借出的登记号），可借时将某一登录号设置为借出，填入借书证号和借出日期； 5. 还书操作：根据书籍名先找到书，然后找到欲还书的登记号并修改为可借，同时删除借书证信息； 6.基本查询功能； 7. 数据文件读写：文件中包含所有书籍信息、每个书籍的登记信息等数据； 8. 基本信息显示：1）所有书籍信息显示；3）特定书籍的借阅信息（已借出或可借）； 9. 可选功能提升：根据登记号直接还书操作等；		
	一、系统概述 图书室管理系统旨在提供一个自动化的解决方案，用于管理图书室中的书籍和借阅活动。该系统将允许管理员和读者进行高效的图书管理，包括书籍的添加、删除、借阅和归还等等。 二、信息管理 系统需要管理两类关键信息： 1.书籍信息：包括书名、书目编号、作者名、出版日期、出版社、库存册数和登记信息集。这些信息构成了每本书的详细档案，便于管理和检索。 2.借阅登记信息：每册书的借阅状态，包括登记号、借阅状态（是否借出）、借阅日期和借书证号。这些信息帮助跟踪每本书的借阅情况。 三、功能要求 系统需实现以下核心功能： 1.创建和管理书籍对象：允许管理员添加新书籍到系统中，并管理这些书籍的详细信息。 2.创建和管理借阅登记信息：为每册书创建借阅记录，并管理这些记录，包括更新借阅状态。 3.增加和删除书籍：管理员可以添加新书到库存或从库存中删除不再需要的书籍。 4.借书操作：读者可以通过书名或书目编号查找书籍，系统检查书籍的可借状态，并在可借的情况下实现借阅操作并更新借阅信息。 5.还书操作：读者归还书籍时，系统根据书名找到书籍，然后更新相应的借阅登记信息，标记为可借并清除借书证信息。可选功能提升：提供根据登记号直接进行还书的功能，增加系统的灵活性和用户		
任务分析			

友好性。

6.基本查询功能：提供搜索功能，允许用户根据书名或书目编号查询书籍基本信息。

7.数据文件读写：系统应能读写包含所有书籍信息和借阅登记信息的数据文件，确保数据的持久化存储。

8.基本信息显示：系统应能显示所有书籍的详细信息和特定书籍的借阅状态。

四、系统设计考虑

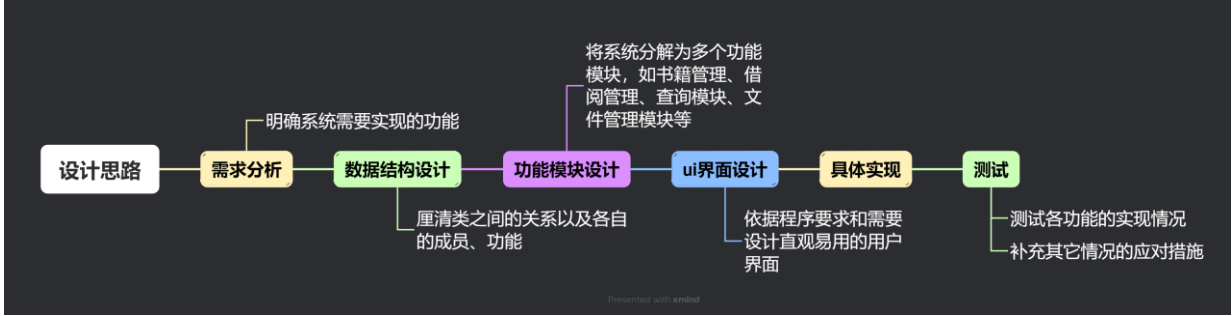
用户界面：设计直观易用的用户界面，分别满足管理员和读者的操作需求。

数据安全：确保书籍和借阅信息的安全，防止未授权访问和数据泄露。

性能要求：系统应能快速响应用户操作，尤其是在借书和还书高峰期。

可扩展性：系统设计应考虑未来可能的扩展，如增加新的功能或支持更多的用户。

一、设计思路：



二、设计过程：

(一) 需求分析

- 1、目的：本文档旨在详细描述图书室管理系统的需求，以便能够设计和实现一个功能齐全、用户友好的图书室管理工具。
- 2、范围：本程序将涵盖书籍信息管理、借阅登记管理、借书与还书操作、查询功能以及数据文件的读写。
- 3、功能需求
- 3.1 书籍信息管理
- 创建和管理书籍对象：系统应能够创建和管理每本书籍的对象，包括书名、书目编号、作者名、出版日期、出版社、库存册数、登记信息。
- 3.2 借阅登记管理
- 创建和管理借阅登记信息：系统应能够创建和管理每册书的登记信息，包括登记号、是否借出、借阅日期、借书证号。
- 3.3 书籍管理
- 增加和删除书籍：系统应允许管理员增加新书籍到库存中，以及删除不再需要的书籍记录。
- 3.4 借书操作
- 借书流程：读者提供书籍信息（书名或书目编号），系统检查书籍是否可借，可借时更新登记信息，记录借书证号和借出日期。
- 3.5 还书操作
- 还书流程：根据书籍名找到书籍，然后找到欲还书的登记号，更新为可借状态，并删除借书证信息。
- 3.6 查询功能
- 基本查询：系统应提供基本查询功能，允许用户根据书名、作者名或其他关键词搜索书籍。

设计思路与流程

3.7 数据文件读写

数据存储：系统应能够读写包含所有书籍信息和登记信息的数据文件。

3.8 信息显示

显示所有书籍信息：系统应能够显示所有书籍的详细信息。

显示特定书籍借阅信息：系统应能够显示特定书籍的借阅状态（已借出或可借）。

3.9 可选功能

直接还书操作：系统可提供根据登记号直接进行还书操作的功能。

4、非功能需求

性能需求：系统应能够快速响应用户操作，查询和更新操作应在几秒内完成。

可用性需求：系统应提供直观的用户界面，方便用户操作。

5、假设和依赖

假设：用户和管理员都将遵守图书室的借阅规则。

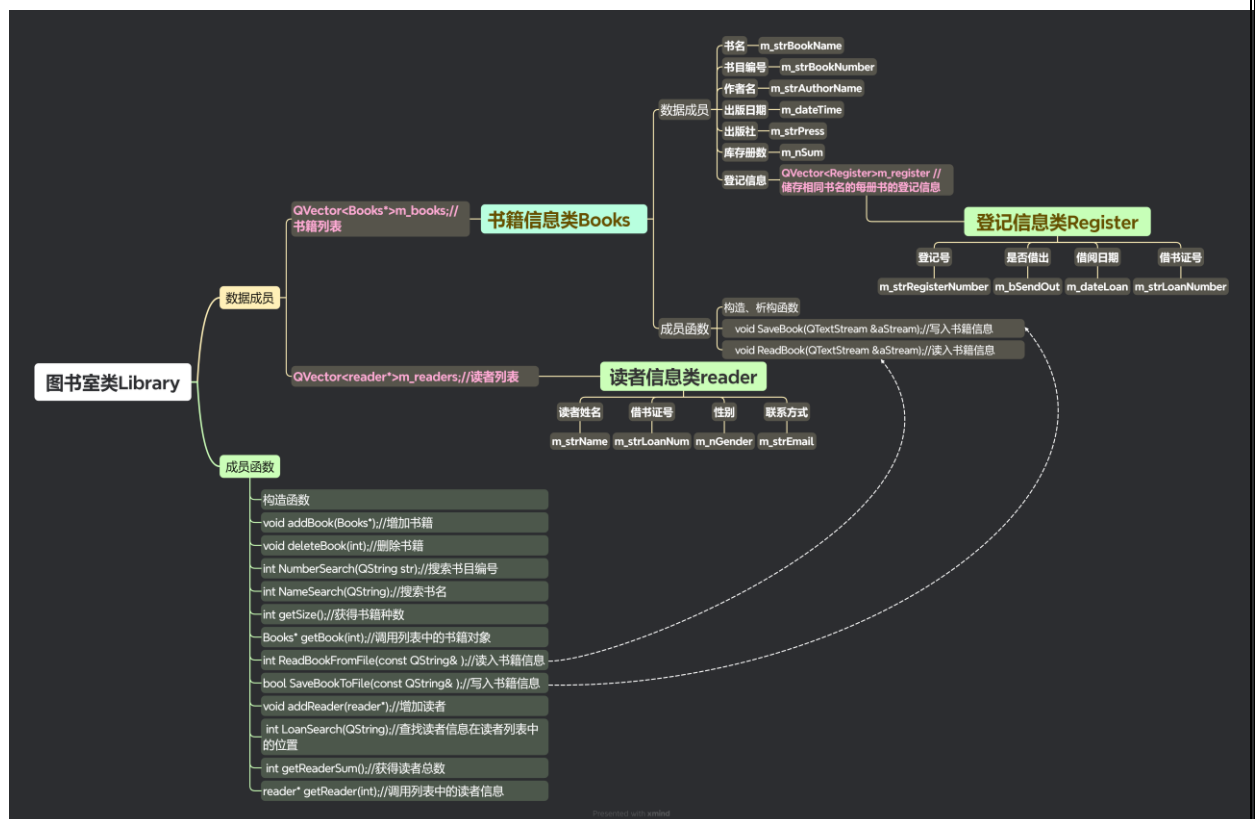
（二）数据结构设计

在这一部分，最主要的类有四个：Library 类，Books 类，Register 类，reader 类。几者的关系如图所示，Library 类存放书籍列表，可以通过其成员函数对书籍列表进行增加、删除等等操作，管理书籍列表；Books 类包含书籍的具体信息，如书名、书目编号等等。

图书室往往存放着相同书名的多册书籍，每册书都有各自的具体登记信息，所以又以由 Register 类型的实例对象组成的数组来储存、管理这些信息。

为了管理读者信息，我又增加了读者信息类 reader，来明确借阅书籍的去向。

几个类层层链接，使用时通过各自的成员函数层层调用来访问、更改每个层次的数据，得以妥善地储存图书室书籍的信息。



1 类之间的关系以及各自成员信息

最后，我在 MainWindow.h 中声明一个 Library 类的实例对象 **m_InfoTable** 来管理、调用以上所有的信息。并通过各个槽函数及成员函数来实现数据与 ui 界面的联接。

除此之外，还有各个随 ui 界面创建的类，主要功能都是通过其成员函数读入各自 ui 界面中可编辑控件的输入内容并返回，以供下一步操作。

包括 addBook, deleteBook, registerNumber, loanNum, addReader, table 类。

```
1  #ifndef MAINWINDOW_H
2  #define MAINWINDOW_H
3  #include "QTableWidget"
4  #include <QMainWindow>
5  #include "books.h"
6  #include <QString>
7  #include <QVector>
8  #include "Library.h"
9  QT_BEGIN_NAMESPACE
10 namespace Ui {
11     class MainWindow;
12 }
13 QT_END_NAMESPACE
14
15 class MainWindow : public QMainWindow
16 {
17     Q_OBJECT
18
19 public:
20     MainWindow(QWidget *parent = nullptr);
21     ~MainWindow();
22
23 protected:
24     void deleteBooksTable(int); //从表格中移除被删除书籍
25     void addBooksTable(Books&); //在表格中添加新书籍
26     void addLoanTable(Books&, int); //新建借阅记录显示
27
28 private slots:
29     void on_actionz_triggered(); //增加书籍槽函数
30     void on_actions_triggered(); //删除书籍槽函数
31     void on_action_cha_triggered(); //查询书籍信息槽函数
32     void on_actionj_triggered(); //借书槽函数
33     void on_action_c_triggered(); //查询借阅信息槽函数
34     void on_actionh_triggered(); //还书槽函数
35     void on_actiondakai_triggered(); //读取文件槽函数
36     void on_actionbaocun_triggered(); //保存文件槽函数
37
38 private:
39     Ui::MainWindow *ui;
40     Library m_InfoTable; //数据库
41 };
42 #endif // MAINWINDOW_H
```

3 MainWindow.h 内容

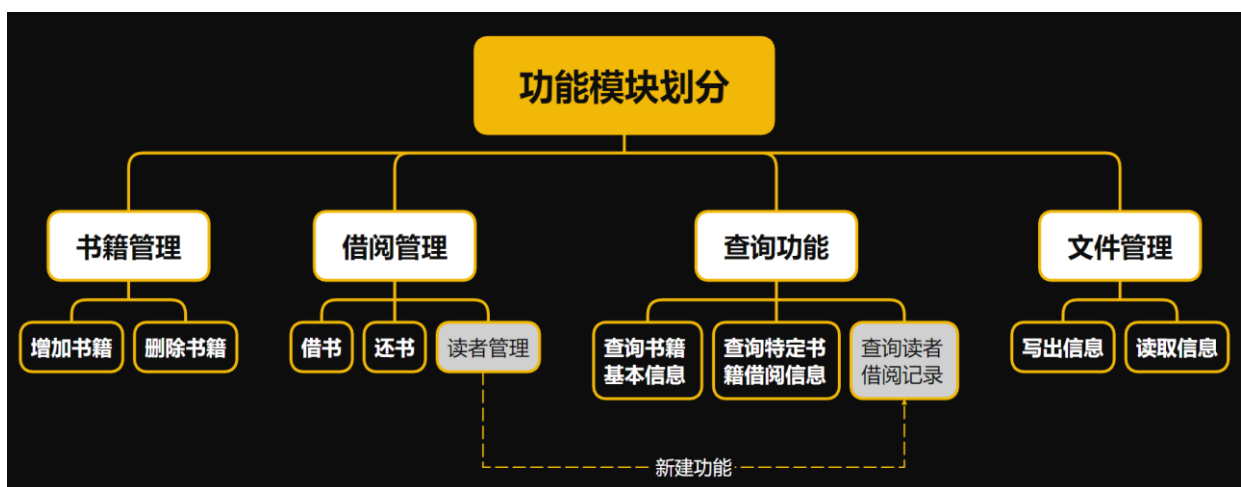
头文件

- addbook.h
- addreader.h
- books.h
- deletebook.h
- library.h
- loanum.h
- mainwindow.h
- reader.h
- register.h
- registernumber.h
- table.h

2 其它类

(三) 功能模块设计

根据程序需求，为了方便操作，我将系统分解为多个功能模块，分别是书籍管理、借阅管理、查询模块、文件管理模块，为了完善系统，我又新增了【读者管理】和【查询读者借阅记录】两个功能：

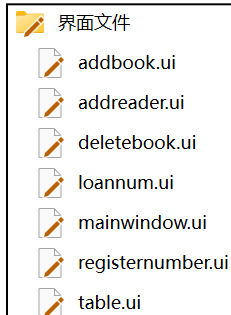


4 功能模块

(四) ui 界面设计

Qt 主窗口界面中,我主要设置了菜单栏、工具栏、状态栏等基本内容,另外的信息显示区域用于满足程序关于显示信息的需求,可以分别显示书籍的基本信息和具体书籍的借阅信息记录,以及系统中的读者信息。

其他的 ui 界面,包括 addBook, deleteBook, table, addReader, registerNumber, loanNum, 界面,会随用户点击相应按钮时弹出,包含主窗口一共 7 个 ui 界面。

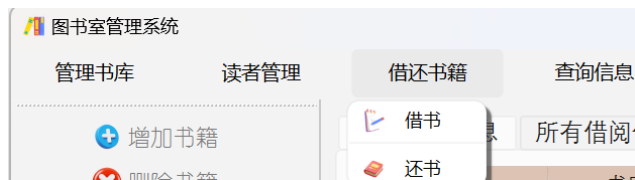


5 ui 界面



6 主界面框架

其中,菜单栏遵循前文的功能模块划分,细分为五个部分。为了便于用户使用,在工具栏展开了具体的功能选项,与菜单栏的选项互通。

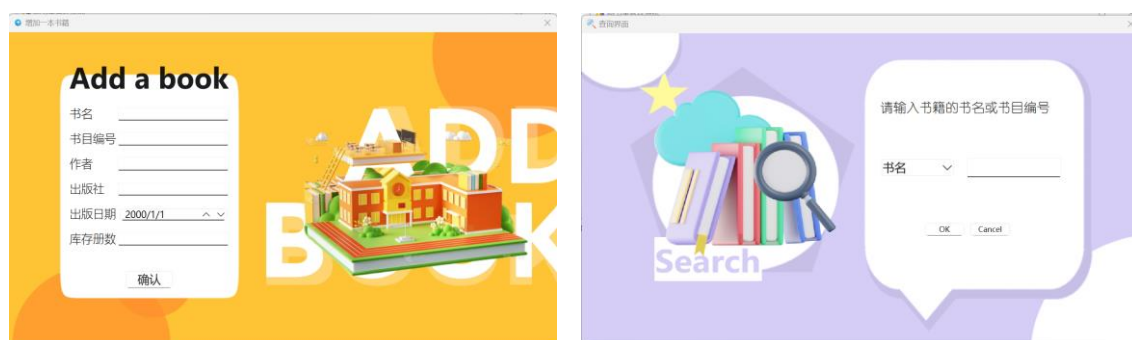


7 菜单栏

为了追求程序的实用性和美观性,我设计了各个 ui 界面的样式和功能,以下是各个 ui 界面的示意图,大部分 ui 都以可编辑的控制如 lineEdit、dateEdit 等和按钮组构成,用以读入用户输入的数据。其中 Table 界面由一个 tableWidget 控件填充,用于显示信息。



8 各个 ui 界面的触发顺序及其实现的功能



9 addBook.ui 和 deleteBook.ui

点击菜单栏或工具栏【添加书籍】按钮，弹出 addBook 窗口，输入书名、书目编号、作者、出版社、出版时间、库存册数，随后点击“确认”读入书籍基本信息，并在 mainWindow 的 tabWidget 中的 tableWidget 控件中新建一行打印书籍信息。

点击菜单栏或工具栏【删除书籍】按钮，弹出 deleteBook 窗口，输入书名或书目编号，点击“OK”读入书名或书目编号进行查找，然后将 mainWindow 的 tabWidget 中的 tableWidget 控件中对应的书籍行移除。



点击菜单或工具栏【添加读者】按钮，弹出 addReader 窗口，填入借书证号、姓名、性别、联系方式，点击“OK”读入读者信息，并在 mainWindow 的 tabWidget 中的 tableWidget_3 控件中新建一行打印书籍信息。

10 addReader.ui (左图)

由于 deleteBook 窗口具有查找功能，很多功能都会调用到它，用来确认用户要找的书是否存在、是书籍列表中的哪一本。

点击菜单栏或工具栏【借书】按钮，弹出 deleteBook 窗口，点击“OK”读入书名或书目编号进行查找，然后弹出 loanNum 窗口，点击“OK”读入借书证号和借阅日期。并在 mainWindow

的 tabWidget 中的 tableWidget_2 控件中新建一行打印借阅信息。



11 loanNum. ui



12 registerNumber. ui

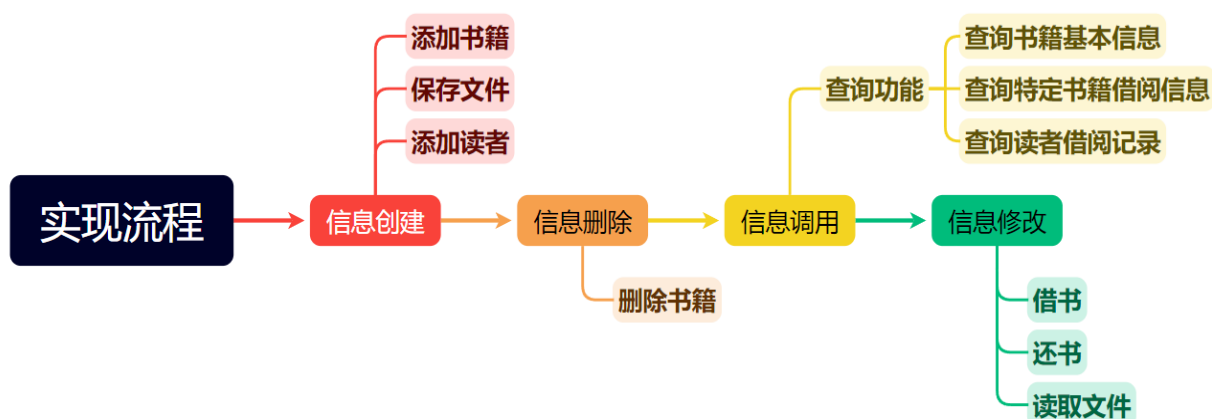
点击菜单栏或工具栏【还书】按钮，弹出 registerNumber 窗口，点击“OK”读入书籍的登记号，并在 mainWindow 的 tabWidget 中的 tableWidget_2 控件中对应书册行将书籍状态改为“可借阅”。

点击菜单栏或工具栏【查询书籍基本信息】按钮，弹出 deleteBook 窗口，点击“OK”读入书名或书目编号进行查找，随后弹出 table 窗口，其中的 tableWidget 控件对目标书籍的信息进行显示。

点击菜单栏或工具栏【查询特定书籍借阅信息】按钮，弹出 registerNumber 窗口，输入书籍登记号，点击“OK”读入书籍的登记号，会弹出信息提示对话框显示书籍借阅状态。

（五）具体实现

我的实现流程：首先设计 MainWindow 主窗口的 ui 界面，确定了所有功能的触发按钮。随后逐个填充槽函数内容，遵循信息的 **创建/删除→调用→更改** 顺序，依次搭建【增加书籍】、【保存文件】、【添加读者】、【删除书籍】、【查询书籍基本信息】、【查询特定书籍借阅信息】、【查询读者借阅记录】、【借书】、【还书】、【读取文件】一应功能，并根据需要在各个类中**创建成员函数**、**设计新的 ui 界面**等来满足需求。



13 功能实现流程

1、信息创建

1.1 增加书籍

填充【增加书籍】按钮的槽函数时，我创建一个 addBook 界面的对象，弹出窗口读入书籍信息，然后创建临时的 Books 类型的指针对象 tempbook 来依次接受各项信息，通

过一个自定义在 Library 类的函数 addBook 来将 tempbook 指针添加到书籍列表中。随后调用 MainWindow 类的函数 addBooksTable, 传入 tempbook 的值, 这个函数的作用是在主窗口的信息显示区域【所有书籍信息】栏新建一行显示书籍信息, 使得新创建的信息可视化。

1.2 保存文件

这里我新建了一系列 QString 对象, 分别赋值它们文件的路径、窗口标题、文件格式, 并调用 getSaveFileName 函数得到文件名赋值给一个 QString 对象。

这里作了一次看文件是否为空的判断, 避免程序继续一段进行不必要的运行。

为了完成进一步的操作, 这里调用了 Library 类的成员函数 SaveBookToFile, 将文件名传入这个函数。在这个函数中, 返回对象是一个布尔类型的值, 我新建了一个 QFile 对象并以传入的文件名初始化, 作判断文件是否是以只写、以文本输入两种模式同时打开, 若不是返回 false; 随后依次写入书籍信息, 然后返回 true。写入的过程中又要进一步调用 Books 类的成员函数 SaveBook 访问书籍信息, 通过这个函数才能接触到书籍的具体信息, 并将它们依次写出。

1.3 添加读者

首先新建 addReader 类的对象弹出窗口读入读者的具体信息, 创建 reader 类的指针对象 tempreader 来接收读者信息并调用 MainWindow 的成员函数 addReadTable 来在主窗口信息显示区域【所有读者信息】栏新建行显示新创建的读者信息, 来使新创建的信息可视化。

然后我调用 reader 类的成员函数 addReader, 传入 tempreader 将其添加到读者列表。

2、信息删除

2.1 删除书籍

首先新建 DeleteBook 类的对象弹出窗口, 读者选择输入书名或书目编号, 借用下拉框控件 comboBox 的函数 currentText 获得用户选中的项目是哪一项目的信息, 然后相应地调用搜索函数 NameSearch 或 NumberSearch, 这两个函数都以 int 类的值为返回对象, 通过遍历寻找目标书籍, 返回目标书籍在书籍列表中的位置序号, 如果书籍不存在于书籍列表中就返回-1。随后在删除书籍的槽函数中作判断: 返回的值是否为-1。若不是则将位置序号传给函数 deleteBooksTable 函数, 这个函数的作用是在主窗口信息显示区域【所有书籍信息】栏找到对应书籍一行执行移除, 不再显示; 同时这个函数还会调用 Library 类的成员函数 deleteBook 把这本书从书籍列表中移除。删除后弹出消息提示窗口。若前面的返回值是-1, 即没有找到这本书, 也设置相应的消息提示窗口弹出。

3、信息调用

3.1 查询书籍基本信息

首先新建 DeleteBook 的对象弹出查询窗口, 接收目标书籍的书名或书目编号, 通过对应的搜索函数进行查找, 确认返回值是否为-1, 如果找到了目标书籍, 新建 table 的对象, 调用 table 类的成员函数 ShowTable, 该函数作用是在 table 窗口中的 tableWidget 控件中创建表格显示该书籍的基本信息。最后使该 table 窗口弹出, 为用户显示对应书籍的信息。

如果没有查询到目标书籍, 设置对应的消息提示对话框弹出。

3.2 查询特定书籍借阅信息

图书室储存着同书名的多本书籍。为了查询每一册各自的借阅信息, 就要调用 Books

类对象的数据成员 `m_register`，即登记信息列表，该书名下的每一册书对应登记信息列表中的一个成员。这个时候书名和书目编号都一样，要区分不同册的书要通过每册书特有的登记号。

新建 `registerNumber` 对象弹出窗口，供用户输入目标书册的书籍登记号，读入后槽函数遍历寻找对应登记号的书册。找到后访问该书册的 `sendOut` 数据成员，得到书册是否借出的信息，弹出对应的信息提示对话框，告知用户借阅信息。如果没有找到目标书册也设置相应的对话框弹出提示用户。

3.3 查询读者借阅记录

这个功能用于查看某个读者所借阅的书册及其信息。每个读者都有独一无二的借书证号，通过这个数据信息来区别和访问不同的读者信息。

新建 `LoanNum` 的对象弹出窗口，供用户输入借书证号，读入后传给函数 `Library` 类的成员函数 `LoanSearch`，这个函数会遍历读者列表，找到借书证号对应的读者，返回其在读者列表中的位置序号。

得到位置序号后，设置消息提示对话框弹出，提示用户现在查看的是哪位用户的借阅信息。随后遍历书籍列表，每本书册的登记信息中都有借阅该书册的借书证号，找到借书证号与用户输入的借书证号一致的书册，调用 `table` 类的成员函数 `ShowLendTable` 在 `table` 窗口的 `tableWidget` 控件中创建行显示借阅信息，新建 `table` 的对象，弹出该窗口，为用户显示信息。

如果遍历后没有找到对应的读者，设置弹出消息对话框，提示用户。

4、信息修改

4.1 借书

新建 `DeleteBook` 对象弹出查询窗口，调用对应搜索函数。找不到目标书籍则设置消息对话框弹出，提示用户未找到。

若找到目标书籍，进一步新建 `LoanNum` 对象，弹出窗口读入用户输入的借书证号和借阅日期。此时作判断，看借书证号对应的读者是否已存在于读者列表中，如果不存在提示读者先新建读者信息到读者列表中，然后再继续借书；如果该读者已存在，遍历书籍列表，查看该书名下是否存在 `sendOut` 数据成员为 0、即未借出的书册，将其借给该读者，并修改对应的登记信息，例如将 `sendOut` 修改为 1、赋值给借书证号信息。同时调用 `MainWindow` 类的成员函数 `addLoanTable`，在主窗口信息显示区域【所有借阅信息】栏新建行显示借阅信息。随后弹出对话框提示读者借书成功。

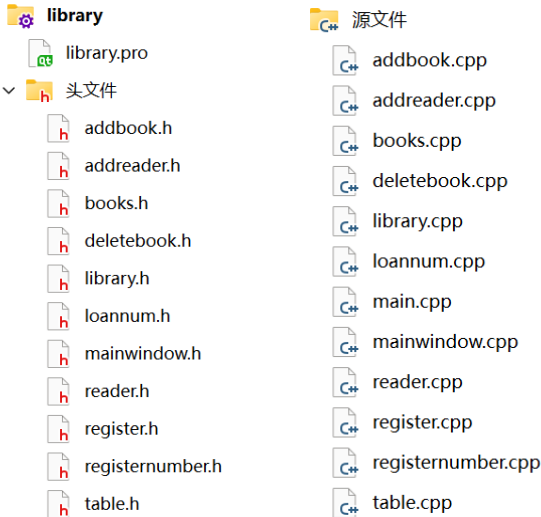
4.2 还书

这里进行题目给出的功能提升，输入登记号直接完成还书操作。

新建 `registerNumber` 对象，弹出窗口供用户输入登记号。随后遍历查询，找到目标书籍后在【所有借阅信息】栏遍历，查看是否存在该书的借阅记录，若不存在则设置消息窗口弹出提示用户；若存在，将书籍状态改为“可借阅”，同时清空借书证号和借阅日期信息。然后弹出消息提示对话框提示。

4.3 读取文件

首先设定好文件的一应信息，通过 `getOpenFileName` 函数得到文件名，然后作判断：该文件是否为空。不为空则调用 `Library` 类的函数 `ReadBookFromFile`，传入文件名，该函数作判断：文件是否存在，以及是否以只读、以文本方式同时打开。随后开始逐个读取文件中的信息。读取具体信息时还要调用 `Books` 类的成员函数 `ReadBook`，然后将新书添加到书籍列表。`ReadBookFromFile` 返回一个 `int` 类型的值，是读取时得到的文档中新

	<p>书的总数。返回后作判断:这个值是否大于零? 如果大于零,继续循环调用 addBooksTable 函数在主窗口【所有书籍信息】栏添加行显示新书信息。</p> <p>(六) 测试 具体见下方【测试报告】。</p>
功能介绍	<p>(1) 创建和管理描述每本书籍的对象、描述每册书登记信息的对象, 即 Books 类和 Register 类。并以 MainWindow 的数据成员 m_InfoTable 来管理、调用这些信息。</p> <p>(2) 增加书籍: 在书籍列表中加入新的 Books 对象, 并在主窗口信息显示区域“所有书籍信息”栏新建一行表格显示书籍信息。</p> <p>(3) 删除书籍: 在书籍列表中删除对应的目标书籍 Books 对象, 并在主窗口信息显示区域“所有书籍信息”栏移除对应书籍的一行信息。</p> <p>(4) 借书: 通过搜索用户输入的书籍的书名或者书目编号, 程序能够判断该书是否存在以及是否还有库存, 若书籍可借则进一步读取用户的借书证号和借阅日期, 更改书籍对应书册的登记信息, 设置为借出。同时主窗口信息显示区域“所有借阅信息”栏新建一行打印借阅信息。</p> <p>(5) 还书: 根据题目要求进行了功能提升, 直接通过书籍登记号进行还书。同时主窗口信息显示区域“所有借阅信息”栏更改书籍借阅状态。</p> <p>(6) 查询: 实现了基本的查询功能。用户可以通过输入书籍书名或书目编号来查询书籍是否存在, 若存在则显示其基本信息; 也可以输入书籍的登记号查询具体某一册书的借阅状态。</p> <p>(7) 信息显示: 主窗口信息显示区域包含“所有书籍信息”和“所有借阅信息”两栏, 可以分别显示相应的具体信息。</p> <p>(8) 数据文件读写: 可以通过点击“保存文件”按钮来写入书籍列表所有书籍的具体信息; 也可以通过点击“读取文件”按钮来读入文件中的书籍信息并新建相应的 Books 对象添加到书籍列表中, 同时主窗口信息显示区域“所有书籍信息”栏打印。</p>
文件与对象描述	<div><div></div><p>所有的头文件已经在前文介绍过了, 以及随之创建的.cpp 文件。这里有几个重要的类对象: MainWindow.h 创建的 Library 对象 m_InfoTable, 这是整个程序的数据总表, 通过这个对象程序得以调用和修改所有的数据。</p><p>Library m_InfoTable; // *总数据调用库</p></div>

而在 Library 类中，又定义了两个类型分别为 Books 和 reader 的 QVector 数组 **m_books** 和 **m_readers**，用来有序地存放书籍和读者信息。

```
protected:
    QVector<Books*>m_books;//书籍列表
    QVector<reader*>m_readers;//读者列表
};
```

在 Books 类中，又有 Register 类型的 QVector 数组，用来存放同一书名下不同书册的登记信息。

```
QVector<Register>m_register;//登记信息：包含登记号
```

其余的头文件都是随 ui 界面创建的，它们的对象多在 MainWindow 的槽函数中被临时创建，用来弹出对应的 ui 界面窗口。

1.创建和管理书籍对象

1.1 添加书籍

标题：成功添加新书籍

前提条件：程序已运行

测试步骤：

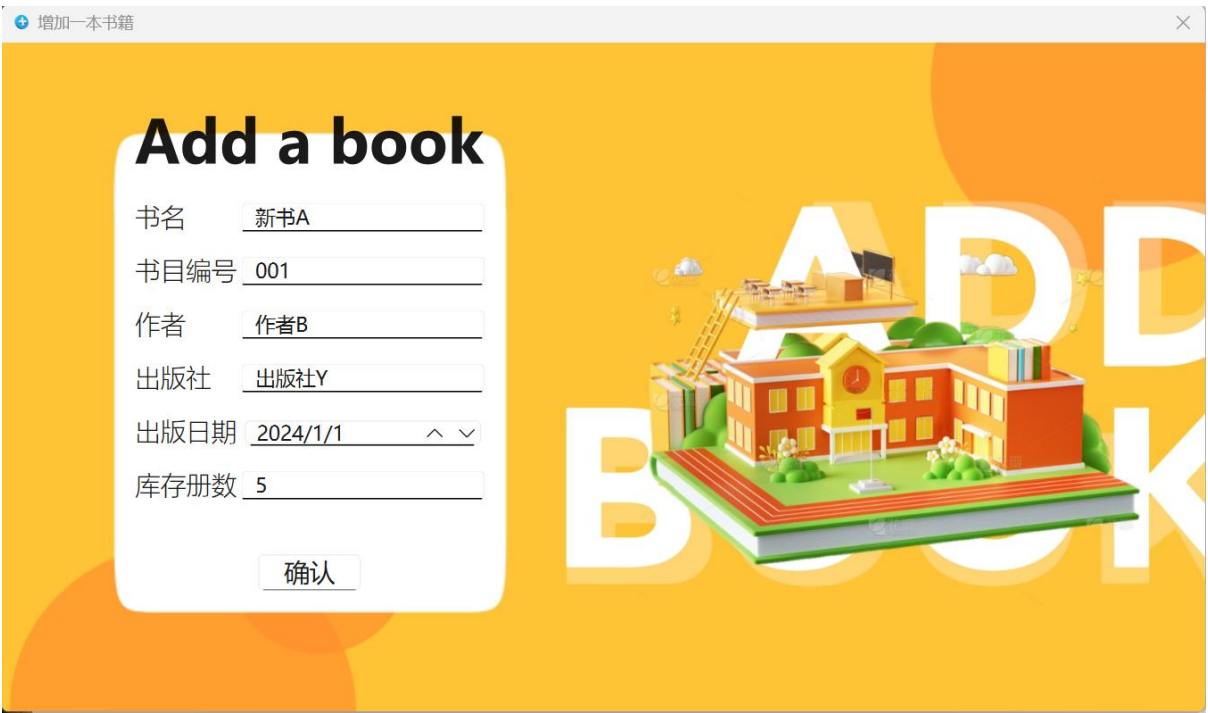
- 1.打开书籍管理界面。
- 2.点击“添加书籍”按钮。
- 3.输入书籍信息：书名、书目编号、作者名、出版日期、出版社、库存册数。
- 4.点击“确认”按钮。

测试数据：书名：“新书 A”，书目编号：001，作者名：“作者 B”，出版日期：“20240101”，出版社：“出版社 Y”，库存册数：5。

预期结果：书籍添加成功，书籍信息出现在书籍列表中。

测试结果：点击【增加书籍】按钮，填入书籍信息，点击“确认“

测试
报告



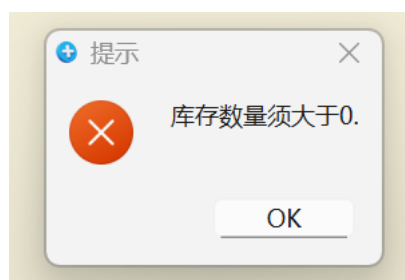


其它测试情况:

输入的书名或书目编号为空:



输入的库存册数为 0:



1.2 删除书籍

标题：成功删除书籍

前提条件：系统已启动，至少有一本书籍在系统中。

测试步骤：

- 1.打开书籍管理界面。
- 2.输入书籍书名或书目编号。
- 3.点击“OK”按钮。

测试数据：选择系统中已存在的一本书。

预期结果：书籍删除成功，书籍信息不再出现在书籍列表中。

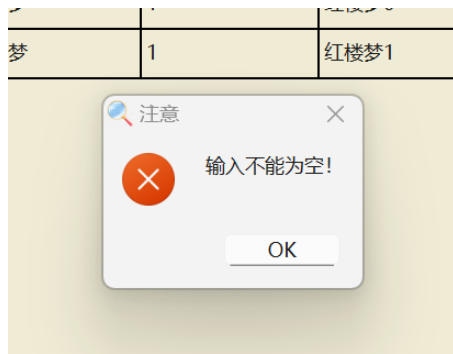
测试结果：



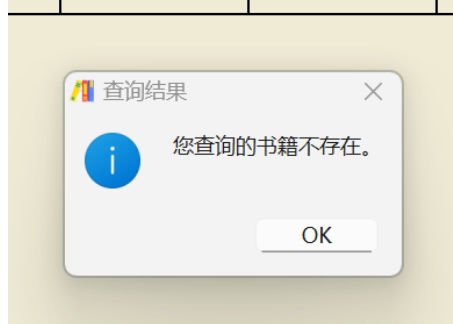
点击“OK”后：



其它测试情况：
输入的信息为空：



输入的书籍不在书籍列表中：



2.创建和管理借阅登记信息

2.1 借书操作

标题：成功借书

前提条件：系统已启动，书籍库存中有可借的书籍。

测试步骤：

- 1.打开借书界面。
- 2.输入书名或书目编号。
- 3.点击“OK”。
- 4.输入借书证号和借阅日期。
- 5.点击“OK”。

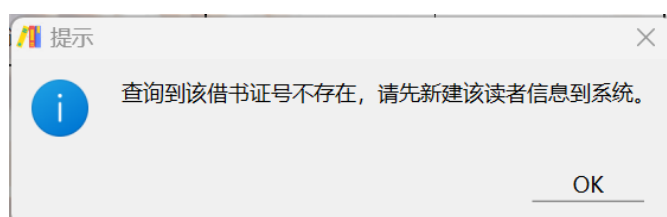
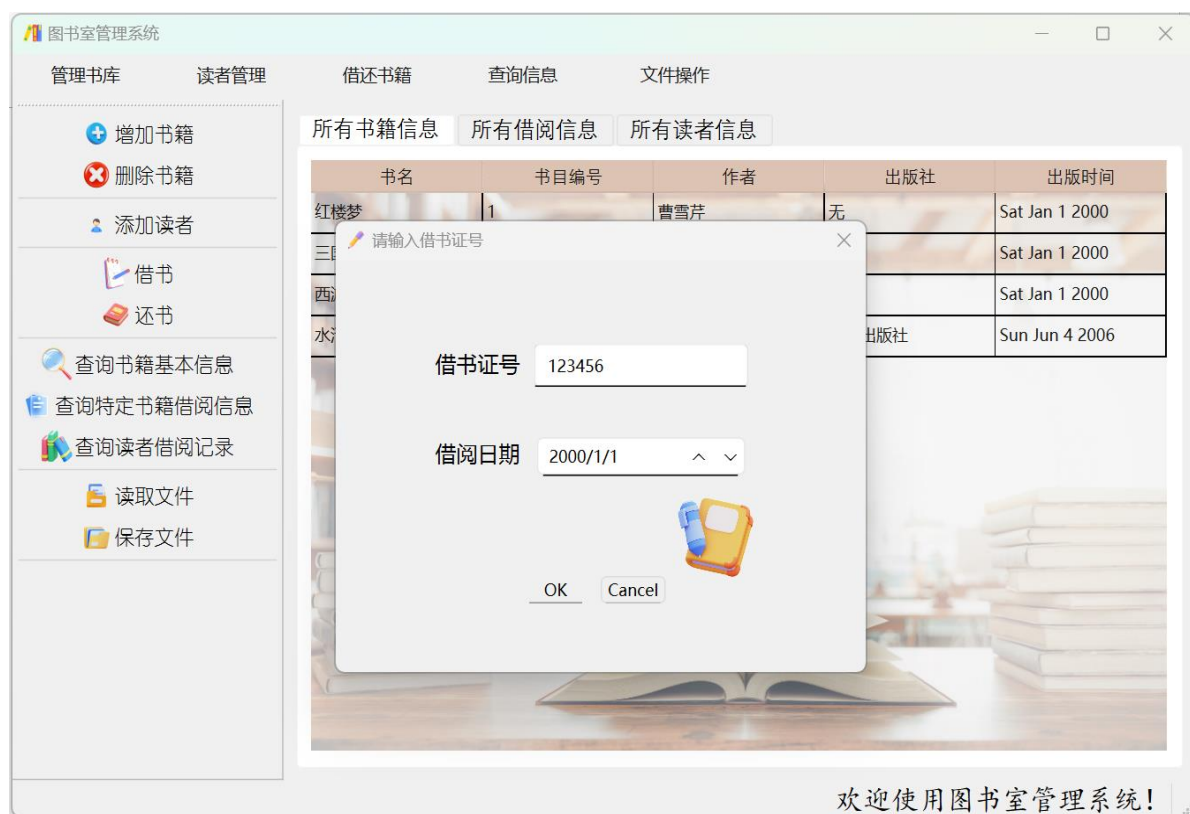
测试数据：书名：“新书 B”，书目编号：002。

预期结果：系统显示借书成功的消息，书籍登记信息更新为“已借出”，并记录借书证号和借出日期。

测试结果：



点击“OK”后：

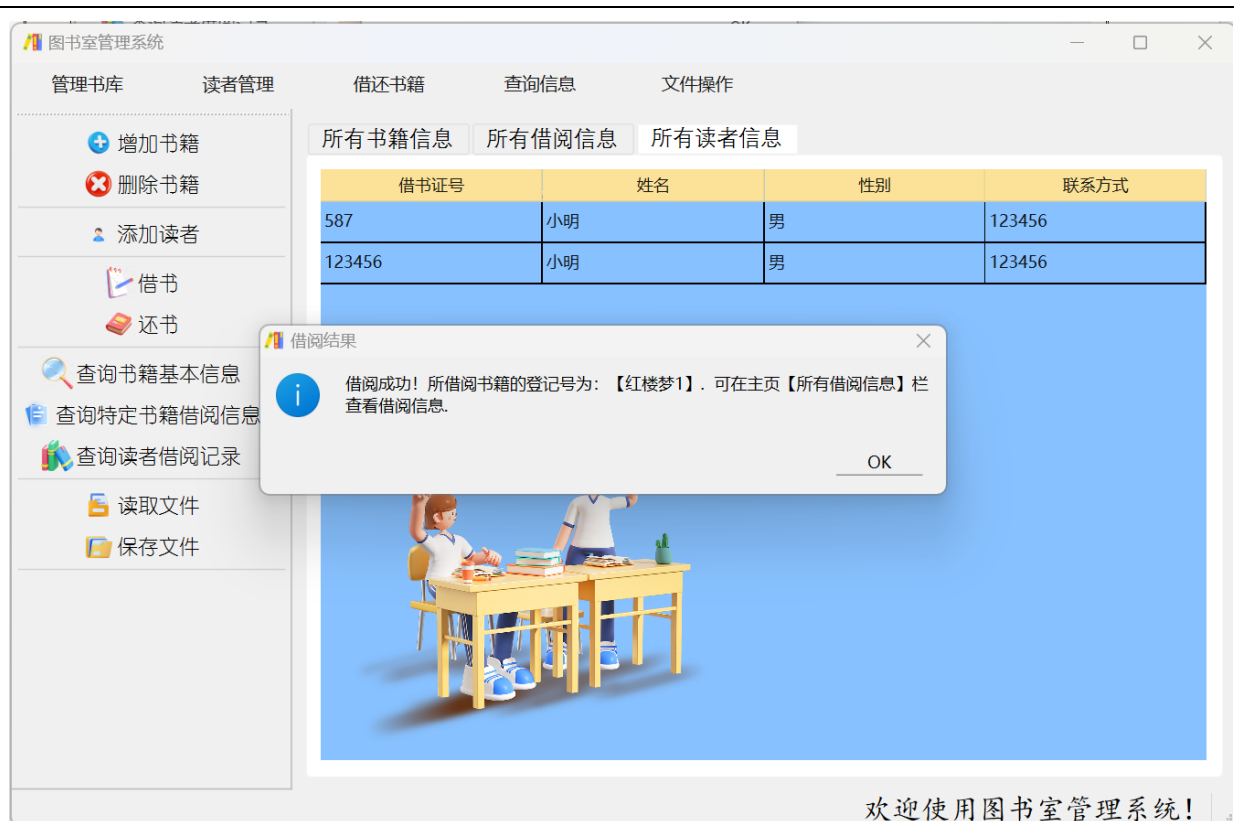




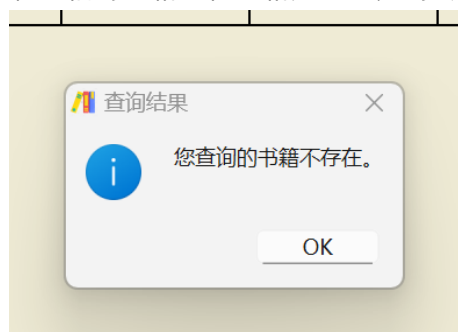
继续点击“OK”：



点击消息对话框的“OK”：



说明：若读者的借书证号在读者列表中，则不会弹出提示新建读者信息的提示对话框。
若要借的书籍不在书籍列表中，则会弹出提示对话框：



2.2 还书操作

标题：成功还书

前提条件：系统已启动，有已借出的书籍。

测试步骤：

1. 打开还书界面。

2.输入书籍登记号。

3.点击“OK”。

测试数据：书名：“红楼梦”，书目编号：1。

预期结果：系统显示还书成功的消息，书籍登记信息更新为“可借阅”，并删除借书证和借阅日期信息。

测试结果：

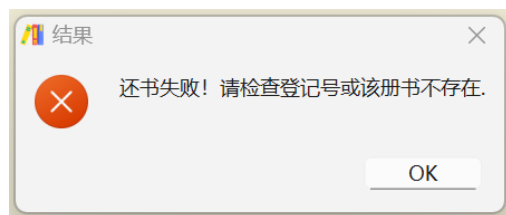


点击“OK”后：



其它测试情况：

若输入的书籍登记号不在被借阅书册中：



2.3 添加读者

标题：成功添加读者

前提条件：系统已启动

测试步骤：

1. 点击【添加读者】按钮

2. 填入读者信息

3. 点击 OK

测试数据：

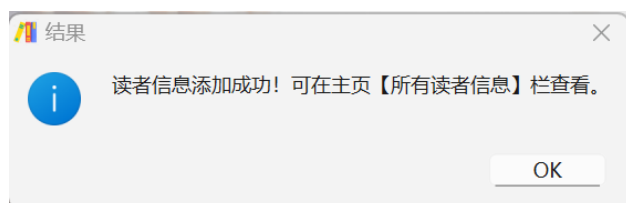
借书证号：123，姓名：小明，性别：男，[联系方式：12345678@qq.com](mailto:12345678@qq.com)

预期结果：读者成功添加

测试结果：

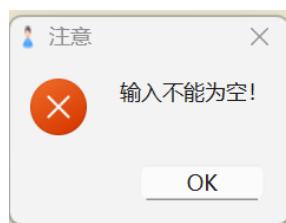


点击“OK”：





若输入的借书证号或姓名为空：



3. 查询功能

3.1 基本查询

标题：通过书名或书目编号查询书籍

前提条件：系统已启动，至少有一本书籍在系统中。

测试步骤：

1. 打开查询界面。
2. 输入书名或者书目编号。
3. 点击“查询”按钮。

测试数据：书目编号：1

预期结果：系统显示与书名匹配的书籍信息。

测试结果：



点击“OK”后：



3.2 查询特定书籍借阅信息

标题：通过书籍登记号查询书籍借阅信息

前提条件：系统已启动，至少有一本书籍在系统中。

测试步骤：

1. 打开查询界面。
2. 输入书籍登记号。
3. 点击“OK”按钮。

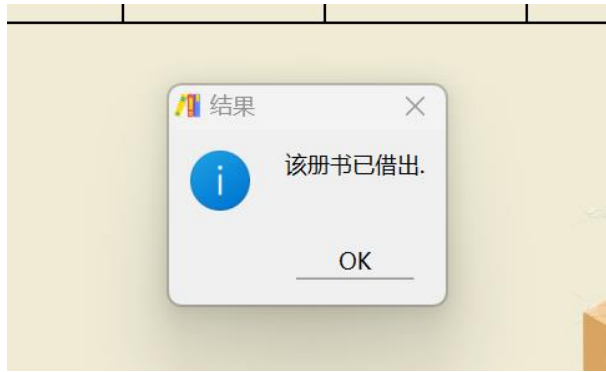
测试数据：登记号：红楼梦 1

预期结果：系统显示相应书册的借阅状态。

测试结果：



点击“OK”:



3.3 查询读者借阅记录

标题：系统能够展示特定读者的借阅记录

前提条件：系统已启动，至少有一本书籍在系统中，读者已在读者列表中且成功借阅过至少一本书籍。

测试步骤：

1. 点击【查询读者借阅记录】按钮。

2. 输入借书证号。

测试数据：读者姓名：韩少婷；借书证号：151；借阅书籍登记号：红楼梦 0，三国演义 0

预期结果：程序弹出窗口显示读者借阅的书籍。

测试结果：



点击“OK”:

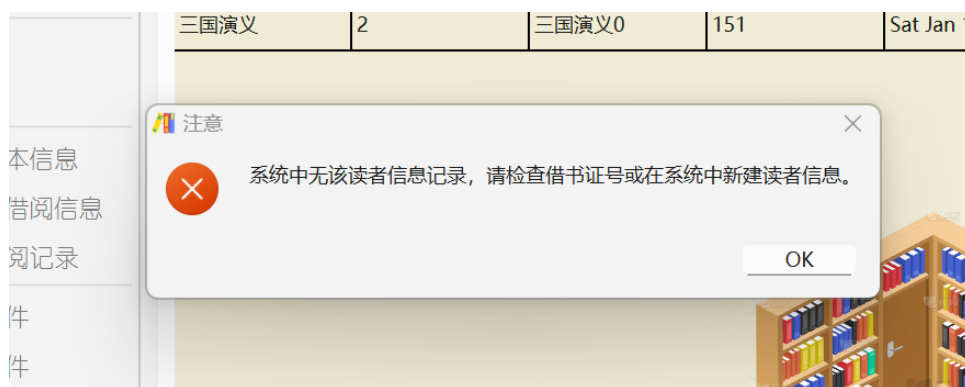


点击“OK”：



其它测试情况：

输入的借书证号对应的读者不在读者列表中：



4.数据文件读写

4.1 保存书籍信息

标题：系统能够正确写出书籍信息

前提条件：系统已启动，至少有一本书籍在系统中。

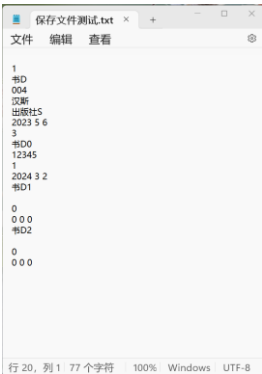
测试步骤：

- 1.打开保存文件界面。
- 2.选择保存路径，设置文件名。
- 3.打开文件检查书籍信息。

测试数据：书名：“书 D”，书目编号：004，作者：汉斯，出版社：出版社 S，出版日期：2023.5.6，库存册数：3.其中一本已借出，登记号：书 D0，借书证号 12345，借阅日期：2024.3.2.

预期结果：打开保存的.txt 文件能够查看之前添加或修改的书籍信息。

测试结果：



4.1 读取书籍信息

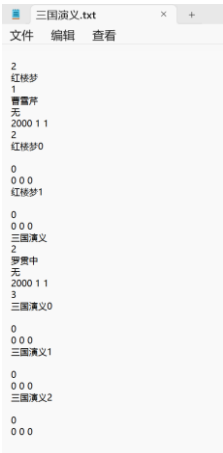
标题：系统能够正确读取文件中的书籍信息

前提条件：系统已启动，有一份符合程序读写格式的.txt 文件。

测试步骤：

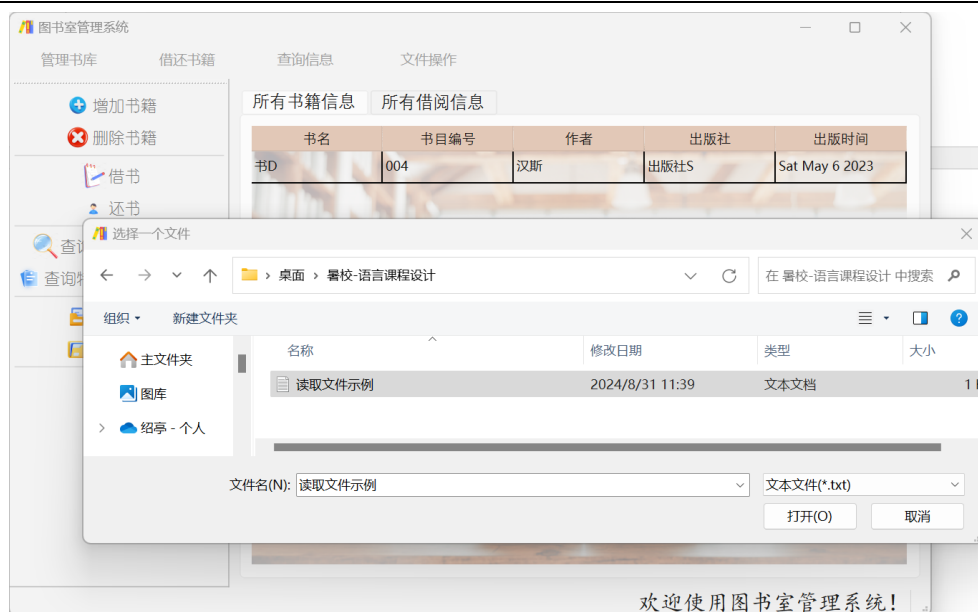
- 1.打开读取文件界面。
- 2.选择保存路径，选择待读取文件。
- 3.返回程序主窗口页面检查书籍信息。

测试数据：已有待读取测试.txt 文件如图：



预期结果：文件中的书籍信息被成功读取，并添加显示在原有的书籍列表中。

测试结果：



点击“打开”：



信息读取成功，如期添加到列表中。

总结

写 Qt 大作业是一次宝贵的历练，从前我写的都是作业和考试中的小题目，实现比较简单，代码量也不多。第一次自己编写一个程序，设计程序的框架结构，代码编写中经常是“牵一发而动全身”，编写方式不能再随意，都要规范且认真。

实现中也遇到了诸多问题，让我积攒了经验。例如起初我并不能很好地梳理各个类之间的关系和结构层次，导致我在调用数据时经常直接调用数据，甚至无意间调用他类的私人数据成员，以至于 bug 丛生。而在逐渐熟悉程序编写的过程中，我逐渐理清了思维，在每个类创建相应的成员函数来访问核心数据，调用时层层深入，保证了封装性也维持了程序的有序结构。

实现中我也遇到了多次代码编译没有报错，而在运行测试中程序自动崩溃的情况。在这些过程中我也有了一些经验，例如我了解到数组越界是导致程序崩溃的一种常见原因。这个时候要合理运用指针对象，使得对数据的改动是有效的、永久的。我也遇到过将一个临时变量的地址反复传给列表的不同地址导致程序崩溃不能运行，最后将临时变量的声明移入循环内部，使程序每次创建不同地址的临时变量，从而解决了问题。

写程序是一个大活，经常一坐几个小时还找不出 bug 的修改方式，我常常要在上机课结束后再找一个空教室，继续写上一两个小时。感谢为我提供建议的老师 and 同学，帮助我解决了一些能力之外的问题，增长了我的认知。我自认完成的大作业与程序员开发的成熟的程序相比只是九牛一毛，然而也因为它感受到一些成就感，自己搭建全部功能、自己设计 ui 界面...正所谓“进一寸有一寸的欢喜”，终究是进了一步，也近了一步。

报告填写说明：

- 1) 设计任务：填写课程设计的任务及要求；
- 2) 任务分析：填写对设计任务的理解和规划；
- 3) 设计思路 and 流程：填写软件设计的基本思路 and 重要流程图；
- 4) 功能介绍：填写系统的主要功能；
- 5) 文件与对象描述：对系统的组成文件及其类对象进行介绍；
- 6) 测试报告：对所使用的测试数据及测试结果进行介绍，请在文中插入重要的软件的运行界面及数据显示结果图片（提示，图片截屏请使用快捷键 **Alt+PrintScreen** 或 **PrintScreen**）；
- 7) 总结：本课程设计的小结、心得及建议。