



上海海事大学
Shanghai Maritime University

面向对象课程设计 图书管理系统

院(系)名称: 信息工程学院

专业名称: 计算机科学与技术

学生姓名: 孙思进

指导教师: 金世双

二〇二一年十一月

摘 要

利用 VC6.0 开发工具开发一个图书管理系统系统。系统实现的功能模块有：用户登录、用户注册、数据库读取、数据库查询、借书、还书、查询书和升级权限的功能。该系统的特色是轻量型和易开发，同时又有高可移植性的特点。系统操作实现的功能比较完善，人机界面实现简单，达到操作过程中的直观、方便、实用、安全等要求。系统具有一定的实用性。实现了图书管理信息系统的图书检索、借阅管理、统计报表管理、图书个性化推荐等功能,能够有效提升图书馆的智能化信息管理和个性化信息服务水平。

关键词: 图书管理系统; 推荐系统; 信息管理

ABSTRACT

Using VC6.0 development tool to develop a library management system. The functional modules implemented by the system include: user login, user registration, database reading, database query, borrowing, returning, querying and upgrading permissions.

The system is characterized by light weight, easy development and high portability. The functions of the system operation are relatively perfect, and the realization of man-machine interface is simple, so as to meet the requirements of intuition, convenience, practicality and safety in the operation process. The system has certain practicability.

It realizes the functions of book retrieval, borrowing management, statistical report management and Book personalized recommendation of the library management information system, and can effectively improve the intelligent information management and personalized information service level of the library.

Key words: Book Manage System;Recommend System;Information Management

目 录

摘要	I
ABSTRACT	II
1 系统概述	1
1.1 系统设计背景	1
1.2 系统设计目标	2
1.3 系统设计环境	2
1.3.1 主机	2
1.3.2 软件	2
1.3.3 C++	2
1.3.4 面向对象	2
1.3.5 QT	3
2 需求分析	4
2.1 系统需求分析	4
2.2 数据流图	4
2.3 数据字典	5
2.3.1 数据项	5
2.3.2 数据字典	5
2.3.3 数据存储	5
2.4 概念结构设计	6
3 总体设计	8
3.1 数据库	8
3.2 系统功能设计	8
4 主要功能模块	9
4.1 类	9
4.1.1 账户类	9
4.1.2 书本类	9

4.2	登录 & 注册	9
4.2.1	类	9
4.2.2	界面	9
4.2.3	实现	9
4.2.4	提示	12
4.3	主界面	14
4.3.1	类	14
4.3.2	界面	14
4.3.3	实现	14
4.4	借书	20
4.4.1	类	20
4.4.2	界面	20
4.4.3	实现	21
4.4.4	提示	24
4.5	还书	25
4.5.1	类	25
4.5.2	界面	25
4.5.3	实现	25
4.5.4	提示	27
4.6	查询	27
4.6.1	类	27
4.6.2	界面	28
4.7	管理员	28
4.7.1	类	28
4.7.2	界面	28
4.7.3	实现	29
4.8	升 & 降级	32
4.8.1	类	32
4.8.2	界面	32
4.8.3	实现	32
5	数据结构说明	36
5.1	界面类	36
5.1.1	登录	36
5.1.2	账户	36
5.1.3	书本	38

6 总结	40
参考文献	41
致谢	42

1 系统概述

1.1 系统设计背景

随着时代的发展,知识改变了世界,人们越来越重视阅读,图书馆的人流量激增,图书馆的规模也随之扩大,馆藏书籍也逐渐增加,图书相关的信量的成倍增加,庞大的信息量使得传统的人工图书管理出现了诸多问题。

图书馆作为一种信息资源的集散地,图书和用户借阅资料繁多,包含很多的信息数据的管理,现今,有很多的图书馆都是初步开始使用,甚至尚未使用计算机进行信息管理。

根据调查得知,他们以前对信息管理的主要方式是基于文本、表格等纸介质的手工处理,对于图书借阅情况(如借书天数、超过限定借书时间的天数)的统计和核实等往往采用对借书卡的人工检查进行,对借阅者的借阅权限、以及借阅天数等用人工计算、手抄进行。

数据信息处理工作量大,容易出错;由于数据繁多,容易丢失,且不易查找。总的来说,缺乏系统,规范的信息管理手段。有必要建立一个图书管理系统,使图书管理工作规范化,系统化,程序化,避免图书管理的随意性,提高信息处理的速度和准确性,能够及时、准确、有效的查询和修改图书情况。

- 效率低。

图书馆的藏书种类繁多,书籍数量也众多,若实用人工的方法将藏书准确地分门别类、快速检索效率是非常低的,往往到最后是终于查询到准确的图书信息馆中图书已经被他人借走。这个问题也会随着藏书的增加而越来越突出。

- 人工任务繁重且容易出错。

大量的借还书任务,使得人工工作量大,随着图书馆越来越收到民众的欢迎,借书和还书的登记会超出预期,导致超负荷工作出现诸多问题。而且人工操作难免会有瑕疵出现。

- 图书管理不易。

图书馆的藏书是会随着时代的推进而更新的,由于藏书的数量和种类的变化,加上自然损耗和人为破坏这些都需要图书管理来统计,如此庞大的工作人工是很难做到良好的效果的。

1.2 系统设计目标

本系统设计一个高效、便捷、实用的图书管理系统，将海量的图书信息用计算机科学技术进行系统地整理编排储存显得尤为重要。系统开发的总的设计目标是实现将图书管理变的系统化、自动化、规范化，对各种书籍和资料进行集中统一管理。主要用户涉及读者（借阅、归还、检索书籍、逾期缴费）和管理者（读者管理，书籍管理）。

1.3 系统设计环境

1.3.1 主机

测试系统和应用系统都安装了 Windows 10 普通版，版本为 1709。

1.3.2 软件

软件环境均列出在表1.1中。

表 1.1 软件环境

序号	软件	版本	备注
1	Visual Studio	2019	编译环境
2	Qt	4.10.0	UI 界面

1.3.3 C++

C++ 是 C 语言的继承，它既可以进行 C 语言的过程化程序设计，又可以进行以抽象数据类型为特点的基于对象的程序设计，还可以进行以继承和多态为特点的面向对象的程序设计。C++ 擅长面向对象程序设计的同时，还可以进行基于过程的程序设计，因而 C++ 就适应的问题规模而论，大小由之。C++ 不仅拥有计算机高效运行的实用性特征，同时还致力于提高大规模程序的编程质量与程序设计语言的问题描述能

1.3.4 面向对象

面向对象 (Object Oriented) 是软件开发方法，一种编程范式。面向对象的概念和应用已超越了程序设计和软件开发，扩展到如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术、人工智能等领域。面

面向对象是一种对现实世界理解和抽象的方法，是计算机编程技术发展到现在一定阶段后的产物。面向对象是相对于面向过程来讲的，面向对象方法，把相关的数据和方法组织为一个整体来看待，从更高的层次来进行系统建模，更贴近事物的自然运行模式。

1.3.5 QT

Qt 是一个 1991 年由 QtCompany 开发的跨平台 C++ 图形用户界面应用程序开发框架。它既可以开发 GUI 程序，也可用于开发非 GUI 程序，比如控制台工具和服务器。Qt 是面向对象的框架，使用特殊的代码生成扩展（称为元对象编译器 (Meta Object Compiler, moc)）以及一些宏，Qt 很容易扩展，并且允许真正地组件编程。2008 年，Qt Company 科技被诺基亚公司收购，Qt 也因此成为诺基亚旗下的编程语言工具。2012 年，Qt 被 Digia 收购。2014 年 4 月，跨平台集成开发环境 Qt Creator 3.1.0 正式发布，实现了对于 iOS 的完全支持，新增 WinRT、Beautififier 等插件，废弃了无 Python 接口的 GDB 调试支持，集成了基于 Clang 的 C/C++ 代码模块，并对 Android 支持做出了调整，至此实现了全面支持 iOS、Android、WP, 它提供给应用程序开发者建立艺术级的图形用户界面所需的所有功能。基本上，Qt 同 X Window 上的 Motif, Openwin, GTK 等图形界面库和 Windows 平台上的 MFC, OWL, VCL, ATL 是同类型的东西。

2 需求分析

2.1 系统需求分析

为了进一步的学习面向对象课程的知识，把理论运用到实际的设计中，我们将具体完成以下任务。

- 结合实际情况写出需求分析。
- 根据需求分析设计满足需求的界面。
- 为界面和控件设计具体的功能实现函数。
- 运行调试程序，对程序的缺陷进行修改。
- 调试无误，运行并测试具体的功能是否能满足实际需求。

结合实际情况，主要设计目标有：

- 添加账户
- 修改账户
- 删除账户
- 查询书本（包括借还功能）
- 升级权限

2.2 数据流图

我们通过图2.1的结构表达了对数据流处理过程：

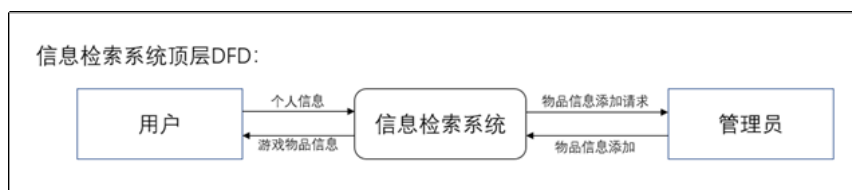


图 2.1 信息检索 DFD

2.3 数据字典

2.3.1 数据项

数据项中我们使用了 Acocount 类作为父类，以此派生出 User、Guest、Admin 三个子类，他们拥有不同的权限。

同样，书本类使用 Book 作为主类，以此派生出 Story、Education、Philosophy 等不同书本类型的分类。

表 2.1 继承和派生关系

bookname	author	label	classname
Mom's Family Wall Calendar 2016	Sandra Boynton	3	Calendars
(Harry Potter, Book 1)	J.K. Rowling	4	Children's Books
Cracking the Coding Interview	Gayle Laakmann McDowell	6	Computers & Technology
The Wright Brothers	David McCullough	10	Engineering & Transportation
Turn Your Weight Loss	Phil McGraw	11	Health, Fitness & Dieting

2.3.2 数据字典

本软件主要是用栈来存储数据，并再此的基础上进行查改删。

对于时间复杂度来说，栈的查找数量级在 $O(n)$ 、修改的复杂度仅为 $O(1)$ 。

2.3.3 数据存储

对于一个账号类，他的数据类型拥有账号、密码、权限；书本类中拥有书名、种类，标签、作者。

图书管理系统其本质就是对图书信息的管理，所以首先我们就必须弄清楚到底有那些信息，其次我们应该如何对这些信息进行系统地分类。结合实际情况进行合理分析，我们得出如下数据需求：

- 借阅者
借阅者编号，借阅者电话。
- 书籍
书籍编号，书名，作者，出版日期，在库数，所在书库，入库日期，出库日期。
- 管理者
管理者编号，负责书库。
- 借阅表
借阅者编号，书籍编号，借出日期，到期日期，拖欠日期，罚款金额

- 销书清单
书籍编号、管理者编号、书名、销书日期、销书数量

2.4 概念结构设计

通过将现代信息技术应用到图书馆管理工作中，我们能够充分发挥图书馆馆藏资源的价值，让所有数据能够被系统化地呈现在读者面前，以供读者进行分析和借阅。

另外，图片信息扫描系统让我们能够将一些珍贵的纸质图书转化为电子版图书，并且上传到计算机网络阅读平台，以供读者随时随地的通过网络平台进行阅读，让学习过程更加方便，不必受到时间和空间的约束。

与此同时，网络平台能够实现读者与读者之间的交流，共享阅读体会，这对于构建积极向上的阅读氛围有重要帮助。

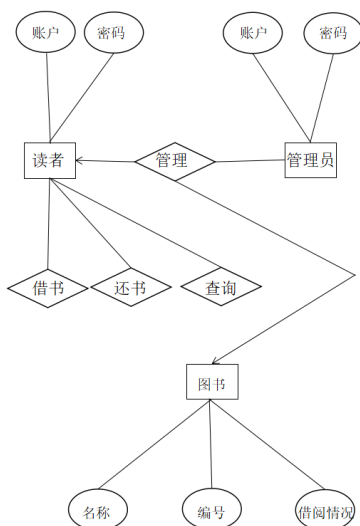


图 2.2 系统 E-R 图

为了更好地将现代信息技术融入图书馆管理工作之中，我们就必须要图书管理任务量化，做到有的放矢，围绕提升服务质量和效率来开展工作，立足于人民群众的实际阅读需求开展图书馆建设。计算机管理系统的应用够提升工作的透明度和规范度，并且减少了工作岗位人员的使用数量，极大地减少了人为失误状况的发生，让各项工作环节变得更加程序化，环环相扣，从而有效改变了传统图书管理中存在的不足。

所以，基于计算机的现代图书管理系统能够更好地收集用户信息，图书管理工作不仅仅是简单的借阅归还，而是朝着互联网方向和平台维护与升级的方向发展，这要求相关的计算机图书管理者不断优化平台建设，实现电子图书管理、

网上平台借阅、用户行为习惯分析的内容，从而让整个服务更加多元化，促进图书馆的现代化建设。

通过上述分析，实现如上的功能，我们将系统流程图写在了图2.3。

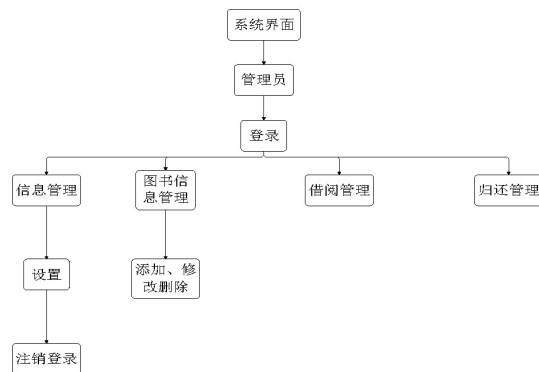


图 2.3 系统结构

3 总体设计

3.1 数据库

由于轻量化的使用，我们将数据格式保存为 csv，通过 csv 的表格式读取，可以增加可读性，而 csv 文件后续也可以转化为数据库，为以后的升级创造了环境。

3.2 系统功能设计

高校图书馆在情报分析、资源组织揭示、知识服务等方面具备专业性优势，承担着高校智库的角色。高校图书馆在情报分析、资源组织揭示、知识服务等方面具备专业性优势，承担着高校智库的角色。高校图书馆在情报分析、资源组织揭示、知识服务等方面具备专业性优势，承担着高校智库的角色。

通过图3.1的架构图，我们设想一种除了图书管理系统以外的多媒体图书馆，为学生提供更多的功能。



图 3.1 系统组成

4 主要功能模块

4.1 类

4.1.1 账户类

账户类涉及到一个 *Account* 作为父类, *User*、*Guest* 和 *Admin* 三个子类。

4.1.2 书本类

书本类涉及到一个 *Book* 作为父类, *Calendars*、*Children's Books*、*Computers*、*Engineering* 和 *Dieting*。

4.2 登录 & 注册

本软件的登录和注册功能集成在一个界面, 当用户输入完账号密码以后, 若他想要注册, 那么点击注册即可。

4.2.1 类

登录界面使用了 *login* 类。

4.2.2 界面

其界面如图4.1所示。

4.2.3 实现

成员函数主要涉及到登录、注册、查询数据、写入数据的功能。

```
1 #include "login.h"
2 #include "ui_login.h"
3 #include "iostream"
4 #include "bookui.h"
5
6 Login::Login(QWidget *parent)
```



图 4.1 登录和注册界面

```
7 : QMainWindow( parent )
8 , ui( new Ui:: Login )
9 {
10     ui->setupUi( this );
11 }
12
13 Login::~Login()
14 {
15     delete ui;
16 }
17
18 void Login::on_login_clicked()
19 {
20     QString acc = this->ui->account->toPlainText();
21     QString pas = this->ui->passowrd->toPlainText();
22     string a = acc.toStdString();
23     string p = pas.toStdString();
24     if ( this->Acc->compare(a,p) )
25     {
26         bookui *p = nullptr;
27         p = new bookui;
28         p->setPath( this->Acc->getPath() );
29         p->setAccount( this->Acc );
```



```
30     p->setLogin(a, this->Acc->getAuthByAcc(a));
31     p->ShowName(a);
32     p->show();
33 }
34 else if(!this->Acc->compareAcc(a))
35 {
36     error *err = new error;
37     err->show();
38     err->ShowText("账号不存在");
39 }
40 else
41 {
42     error *err = new error;
43     err->show();
44     err->ShowText("密码错误");
45 }
46 }
47
48 void Login::on_regist_clicked()
49 {
50     QString acc = this->ui->account->toPlainText();
51     QString pas = this->ui->passowrd->toPlainText();
52     string a = acc.toStdString();
53     string p = pas.toStdString();
54     if(!this->Acc->compareAcc(a))
55     {
56         this->Acc->addUser(a,p,0);
57         error *err = new error;
58         err->show();
59         err->ShowText("注册成功");
60         qDebug("success");
61     }
62     else if(a.length()<=5 or p.length()<=5)
63     {
64         error *err = new error;
65         err->show();
66         err->ShowText("账号或密码太短，请重新输入");
```

```

67     this->ui->account->clear();
68     this->ui->passowrd->clear();
69     qDebug("unsucccess");
70 }
71 else
72 {
73     error *err = new error;
74     this->ui->account->clear();
75     this->ui->passowrd->clear();
76     err->show();
77     err->ShowText("注册失败，账号重复");
78     qDebug("unsucccess");
79 }
80
81 }
82
83 void Login::AddAccount(Account *a)
84 {
85     this->Acc = a;
86 }

```

本软件通过登录和注册两个按钮，即 QT 中的槽和信号的模块实现进入下一个窗口的功能。

登录需要根据读者或管理员提供的登录账号和密码进行，系统会自动进行后台数据库的验证，并根据错误信息返回登录，在登录模块进行权限验证，用于区分读者身份和管理员身份。登录模块效果如图 3 所示。读者模块主要包含关于读者仅限的操作。用户登录后会直接跳转到个人信息页面也可以跳转到其他用户的操作页面主要包括个人信息、图书查阅和密码修改三个小模块。个人信息模块主要显示个人基础信息以及当前书籍借阅和历史书籍借阅情况。

图书查询模块会根据读者输入的信息按类别进行检索查询，将检索结果进行分页显示，便于用户查阅。

4.2.4 提示

当用户输入了错误的账号或者密码，软件的提示如图4.2所示。

当用户想要注册账号时，输入的账号已经存在，那么会提示账号已存在，如图4.3所示。



图 4.2 密码错误



图 4.3 账号已存在

当用户注册账号成功时，保留原始输入的账号、密码和界面，软件读取输入的字符串，存储到 csv 文件中，如图4.4所示，提示账号注册成功。



图 4.4 账号注册成功

当用户输入的账号和密码与 csv 文件中储存的数据一致时，弹出登陆成功的提示，跳转到主界面，如下文图4.5所示。

4.3 主界面

本软件的主界面设计了四个主要的功能：借书、查书、还书和管理员功能。同时，软件在主界面显示了 6 本新书速递，展示了入库时间最晚的 6 本书。

4.3.1 类

主界面使用了 Bookui 类。

4.3.2 界面

主界面如图4.5所示，对于不同的登录用户，如图4.6，会显示不同的管理员权限。

4.3.3 实现

通过四个按钮，跳转到四个不同的功能界面。

由于图书涉及到多个不同的种类,我们将其分为 *Calendars*、*Children's Books*、*Computers*、*Engineering* 和 *Dieting*。

```
1 #include "bookui.h"
2 #include "ui_bookui.h"
3
```

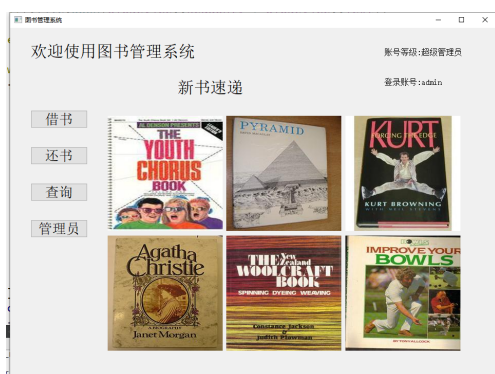


图 4.5 主界面-管理员

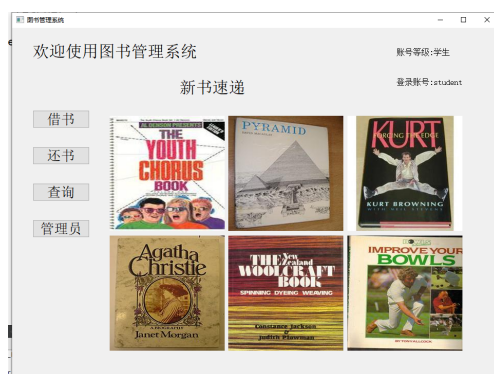


图 4.6 主界面-学生

```

4 bookui::bookui(QWidget *parent) :
5   QWidget(parent),
6   ui(new Ui::bookui)
7   {
8       ui->setupUi(this);
9   }
10
11 bookui::~~bookui()
12 {
13     delete ui;
14 }
15
16 void bookui::setAccount(Account *a)
17 {
18     this->Acc = a;
19 }
20
21 string bookui::getLoginAcc()
22 {
23     return this->loginAcc;
24 }
25
26 void bookui::setLogin(string a, int b)
27 {
28     this->loginAcc = a;
29     this->loginAuth = b;

```

```
30 }
31
32 int bookui::getLoginAuth()
33 {
34     return this->loginAuth;
35 }
36
37 void bookui::ShowName(string s)
38 {
39     this->ui->AccountLabel->setText(QString::fromStdString("
    登录账号:" + s));
40     string a;
41     this->loginAuth = this->getLoginAuth();
42     a = this->Acc->getAuthString(this->Acc->getAuthByAcc(s));
43     this->ui->AuthLabel->setText(QString::fromStdString("账号
    等级:" + a));
44     qDebug("是否运行");
45     Book *d = new Book;
46     d->add();
47     d->setPath(this->Acc->getPath());
48     this->setBook(d);
49 }
50
51 void bookui::setBook(Book *bo)
52 {
53     this->book = bo;
54 }
55
56 void bookui::on_BorrowButton_clicked()
57 {
58     if (this->getLoginAuth() >= 2)
59     {
60         borrow *c = new borrow;
61         c->setBook(this->book);
62         c->setLoginAccount(this->loginAcc);
63         c->show();
64     }
```

```
65     else
66     {
67         error *e = new error;
68         e->ShowText("非管理员权限只能使用查询和还书功能。");
69         e->show();
70     }
71 }
72
73 void bookui::on_LendButton_clicked()
74 {
75     if (this->getLoginAuth() >= 2)
76     {
77         lend *l = new lend;
78         l->setBook(this->book);
79         l->show();
80     }
81     else
82     {
83         error *e = new error;
84         e->ShowText("访客权限只能使用查询功能。");
85         e->show();
86     }
87 }
88
89 void bookui::on_AdminButton_clicked()
90 {
91     if (this->getLoginAuth() >= 3)
92     {
93         auto *a = new admin;
94         a->setAuthr(getLoginAuth());
95         a->setAcc(this->Acc);
96         a->show();
97     }
98     else
99     {
100         error *e = new error;
101         e->ShowText("非超级管理员权限无法使用管理员功能。");
```

```

102     e->show();
103 }
104 }
105
106 void bookui::on_SearchButton_clicked()
107 {
108
109     auto *s = new findd;
110     s->setBook(this->book);
111     s->show();
112 }
113
114 void bookui::setPath(string s)
115 {
116     this->path = s;
117 }
118
119 string bookui::getPath()
120 {
121     return this->path;
122 }
123 #include "lend.h"
124 #include "ui_lend.h"
125
126 lend::lend(QWidget *parent) :
127     QWidget(parent),
128     ui(new Ui::lend)
129 {
130     ui->setupUi(this);
131 }
132
133 lend::~lend()
134 {
135     delete ui;
136 }
137
138 void lend::setBook(Book *b)

```



```
139 {
140     this ->book = b;
141 }
142
143 void lend::on_SerachButton_clicked()
144 {
145     qDebug("%d",1);
146     QString SearchText = this ->ui->SearchText->toPlainText();
147     string s = SearchText.toStdString();
148     qDebug("%d",2);
149     int z = -1;
150     for(int i = 0;i<this ->book->getL();i++)
151     {
152         if(this ->book->getName(i) == s || this ->book->getISBN(i)
153             ) == s)
154         {
155             z = i;
156             break;
157         }
158     }
159
160     if (z == -1)
161     {
162         error *p = new error;
163         p->show();
164         p->ShowText("找不到此信息，请重新输入");
165         this ->ui->SearchText->clear();
166     }
167     else
168     {
169         this ->ui->BookLabel->setPixmap(QString::fromStdString("
170             :/img/img/book/" + this ->book->getISBN(z) + ".jpg"));
171         if (this ->book->getStatus(z))
172         {
173             this ->book->setStatus(z,0);
174             this ->book->writeBook();
175             error *p = new error;
```

```

174     p->show();
175     p->ShowText("还书成功");
176 }
177 else
178 {
179     error *p = new error;
180     p->show();
181     p->ShowText("书本未处于出借状态，请重新确认");
182 }
183 }
184 }

```

4.4 借书

借书界面主要实现的是借书和查询书的功能，当用户权限不是管理员时，无法使用该功能。

4.4.1 类

借书使用了 borrow 类。

4.4.2 界面

界面主要涉及到查询和借阅两个按钮，图书查询模块会根据读者输入的信息按类别进行检索查询，将检索结果进行分页显示，便于用户查阅。当用户按下查询时，界面的信息将会更新，如图4.7所示。

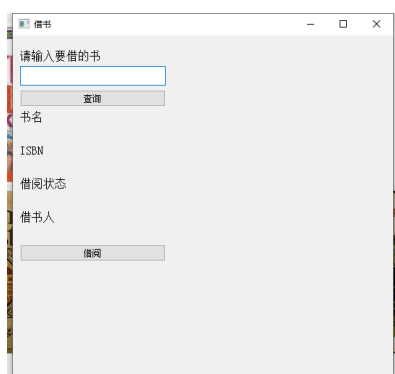


图 4.7 借书界面

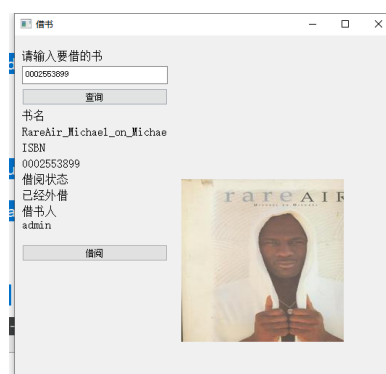


图 4.8 主界面-学生

当按下查询以后，会显示该本书的信息和书本封面，如图4.8所示。

4.4.3 实现

读者模块主要包含关于读者仅限的操作。用户登录后会直接跳转到个人信息页面也可以跳转到其他用户的操作页面主要包括个人信息、图书查阅和密码修改三个小模块。个人信息模块主要显示个人基础信息以及当前书籍借阅和历史书籍借阅情况。

该功能的实现主要依靠数据库的查询和修改，其成员函数如下：

```
1 #include "bookui.h"
2 #include "ui_bookui.h"
3
4 bookui::bookui(QWidget *parent) :
5     QWidget(parent),
6     ui(new Ui::bookui)
7 {
8     ui->setupUi(this);
9 }
10
11 bookui::~bookui()
12 {
13     delete ui;
14 }
15
16 void bookui::setAccount(Account *a)
17 {
18     this->Acc = a;
19 }
20
21 string bookui::getLoginAcc()
22 {
23     return this->loginAcc;
24 }
25
26 void bookui::setLogin(string a,int b)
27 {
28     this->loginAcc= a;
29     this->loginAuth = b;
30 }
```

```
31
32 int bookui::getLoginAuth()
33 {
34     return this ->loginAuth;
35 }
36
37 void bookui::ShowName(string s)
38 {
39     this ->ui->AccountLabel->setText(QString::fromStdString("
    登录账号:" + s));
40     string a;
41     this ->loginAuth = this ->getLoginAuth();
42     a = this ->Acc->getAuthString(this ->Acc->getAuthByAcc(s));
43     this ->ui->AuthLabel->setText(QString::fromStdString("账号
    等级:" + a));
44     qDebug("是否运行");
45     Book *d = new Book;
46     d->add();
47     d->setPath(this ->Acc->getPath());
48     this ->setBook(d);
49 }
50
51 void bookui::setBook(Book *bo)
52 {
53     this ->book = bo;
54 }
55
56 void bookui::on_BorrowButton_clicked()
57 {
58     if (this ->getLoginAuth() >= 2)
59     {
60         borrow *c = new borrow;
61         c->setBook(this ->book);
62         c->setLoginAccount(this ->loginAcc);
63         c->show();
64     }
65     else
```

```
66 {
67     error *e = new error;
68     e->ShowText("非管理员权限只能使用查询和还书功能。");
69     e->show();
70 }
71 }
72
73 void bookui::on_LendButton_clicked()
74 {
75     if (this->getLoginAuth() >= 2)
76     {
77         lend *l = new lend;
78         l->setBook(this->book);
79         l->show();
80     }
81     else
82     {
83         error *e = new error;
84         e->ShowText("访客权限只能使用查询功能。");
85         e->show();
86     }
87 }
88
89 void bookui::on_AdminButton_clicked()
90 {
91     if (this->getLoginAuth() >= 3)
92     {
93         auto *a = new admin;
94         a->setAuthr(getLoginAuth());
95         a->setAcc(this->Acc);
96         a->show();
97     }
98     else
99     {
100         error *e = new error;
101         e->ShowText("非超级管理员权限无法使用管理员功能。");
102         e->show();
```

```
103     }
104 }
105
106 void bookui::on_SearchButton_clicked()
107 {
108
109     auto *s = new findd;
110     s->setBook(this->book);
111     s->show();
112 }
113
114 void bookui::setPath(string s)
115 {
116     this->path = s;
117 }
118
119 string bookui::getPath()
120 {
121     return this->path;
122 }
```

4.4.4 提示

若用户的权限不够，将无法打开结束界面，其提示如图所示。



图 4.9 账号权限不够

4.5 还书

该界面主要为了做到还书的功能，通过查询数据库，先判断其是否属于借出状态，在进行还书，写入数据。

4.5.1 类

还书界面使用了 `Lend` 类。

4.5.2 界面

还书的界面如图4.10所示。

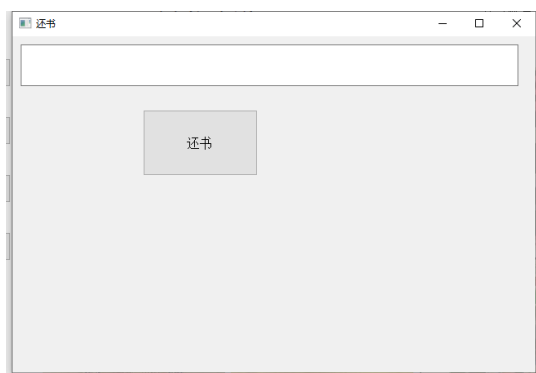


图 4.10 还书界面

4.5.3 实现

先调取用户输入的账号，查询数据库进行匹配，成员函数如下所示。

```
1 #include "lend.h"
2 #include "ui_lend.h"
3
4 lend::lend(QWidget *parent) :
5     QWidget(parent),
6     ui(new Ui::lend)
7     {
8         ui->setupUi(this);
9     }
10
11 lend::~lend()
12 {
```

```
13     delete ui;
14 }
15
16 void lend::setBook(Book *b)
17 {
18     this->book = b;
19 }
20
21 void lend::on_SerachButton_clicked()
22 {
23     qDebug("%d",1);
24     QString SearchText = this->ui->SearchText->toPlainText();
25     string s = SearchText.toStdString();
26     qDebug("%d",2);
27     int z = -1;
28     for(int i = 0; i < this->book->getL(); i++)
29     {
30         if(this->book->getName(i) == s || this->book->getISBN(i) == s)
31         {
32             z = i;
33             break;
34         }
35     }
36
37     if (z == -1)
38     {
39         error *p = new error;
40         p->show();
41         p->ShowText("找不到此信息，请重新输入");
42         this->ui->SearchText->clear();
43     }
44     else
45     {
46         this->ui->BookLabel->setPixmap(QString::fromStdString("/img/img/book/" + this->book->getISBN(z) + ".jpg"));
47         if (this->book->getStatus(z))
```



```

48     {
49         this->book->setStatus(z,0);
50         this->book->writeBook();
51         error *p = new error;
52         p->show();
53         p->ShowText("还书成功");
54     }
55     else
56     {
57         error *p = new error;
58         p->show();
59         p->ShowText("书本未处于出借状态，请重新确认");
60     }
61 }
62 }

```

4.5.4 提示

还书的界面会出现三个提示,还书成功(图4.11)、书本未处于借阅状态(图4.12)和书本不存在(图4.13)。

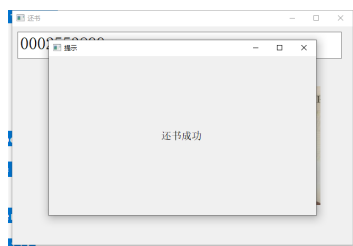


图 4.11 还书成功

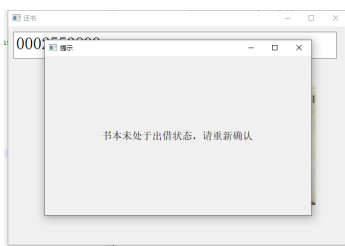


图 4.12 书本未借阅

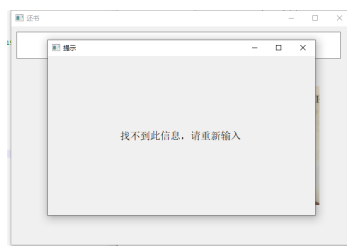


图 4.13 书本不存在

4.6 查询

查询功能是借书的继承功能，其功能大致与借书类似，由于查询功能可供访客使用，消去了显示借阅者的功能，如图4.14所示。

4.6.1 类

查询使用了 search 类。

4.6.2 界面

图书查询模块会根据读者输入的信息按类别进行检索查询, 将检索结果进行分页显示, 便于用户查阅。

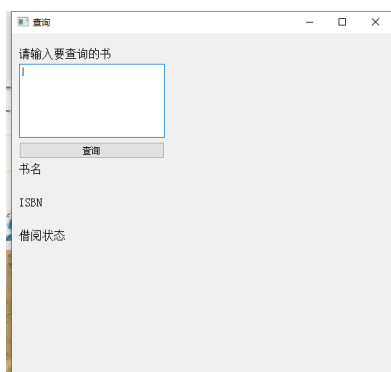


图 4.14 查询界面

4.7 管理员

超级管理员是管理员中最大的权限, 他可以控制除了他以外的用户进行升降级权限。

4.7.1 类

管理员使用了 Admin 类。

4.7.2 界面

管理员界面主要有查询账号 (图4.15)、修改密码 (图4.16)、修改权限的功能。先查询账号, 再显示其信息, 如图4.15所示。



图 4.15 管理员界面



图 4.16 查询账号和权限

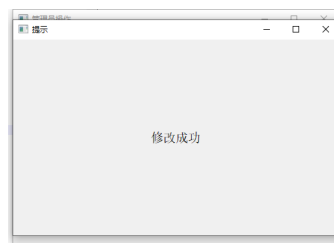


图 4.17 修改密码

4.7.3 实现

对于上述三个功能，实现了查询账号、修改密码、修改权限的功能。

```
1 #include "admin.h"
2 #include "ui_admin.h"
3
4 admin::admin(QWidget *parent) :
5     QWidget(parent),
6     ui(new Ui::Admin)
7 {
8     ui->setupUi(this);
9 }
10
11 admin::~admin()
12 {
13     delete ui;
14 }
15
16 void admin::on_AccountButton_clicked()
17 {
18     QString acc = this->ui->AccountText->toPlainText();
19     string ac = acc.toStdString();
20     int z = -1;
21     qDebug("%d", this->Acc->getL());
22     for(int i = 0; i < this->Acc->getL(); i++)
23     {
24         qDebug(qUtf8Printable(QString::fromStdString(this->Acc
25             ->getAcc(i))));
26         if (ac == this->Acc->getAcc(i))
27         {
28             z = i;
29             break;
30         }
31     }
32     qDebug("%d", z);
33     if(z+1)
```

```
34     this->ui->PasswordLabel->setText(QString::fromStdString  
    (this->Acc->getPas(z)));  
35     this->ui->AuthLabel->setText(QString::fromStdString(  
        this->Acc->getAuthString(this->Acc->getAuth(z))));;  
36 }  
37 else  
38 {  
39     error *e = new error;  
40     e->ShowText("未找到此账号");  
41     e->show();  
42 }  
43 }  
44  
45 void admin::setAcc(Account *a)  
46 {  
47     this->Acc = a;  
48 }  
49  
50 void admin::on_PasswordButton_clicked()  
51 {  
52     QString s = this->ui->AccountText->toPlainText();  
53     string ss = s.toStdString();  
54     string pas = this->ui->PasswordSet->toPlainText().  
        toStdString();  
55     int z = -1;  
56     qDebug("%d", this->Acc->getL());  
57     for(int i = 0; i < this->Acc->getL(); i++)  
58         if (ss == this->Acc->getAcc(i))  
59         {  
60             z = i;  
61             break;  
62         }  
63  
64     qDebug("%d", z);  
65     if (pas.length() < 5)  
66     {  
67         error *e = new error;
```

```
68     e->ShowText("密码长度过短，请重新输入");
69     e->show();
70     this->ui->PasswordLabel->clear();
71     this->ui->PasswordSet->clear();
72 }
73 else if(z+1)
74 {
75     error *e = new error;
76     e->ShowText("修改成功");
77     e->show();
78     this->Acc->setPas(z, pas);
79     this->ui->PasswordLabel->setText(QString::fromStdString
      (pas));
80     this->Acc->save();
81 }
82 else
83 {
84     error *e = new error;
85     e->ShowText("未找到此账号");
86     e->show();
87 }
88 }
89
90 void admin::on_AuthButton_clicked()
91 {
92     if(this->getAuthr() >=3)
93     {
94         auth *a = new auth;
95         a->setAcc(this->Acc);
96         a->show();
97     }
98     else
99     {
100         error *e = new error;
101         e->ShowText("普通管理员无法修改权限");
102         e->show();
103     }
```

```
104 }  
105  
106 void admin::setAuthr(int a)  
107 {  
108     this->authr = a;  
109 }  
110  
111 int admin::getAuthr()  
112 {  
113     return this->authr;  
114 }
```

4.8 升 & 降级

通过点击修改权限的按钮，进入升降级功能。

4.8.1 类

升降级使用了 auth 类。

4.8.2 界面

其界面如图4.18所示。



图 4.18 升降级

4.8.3 实现

对于一个用户，无法升级到超级管理员，也无法降级到访客以下。

```

1  #include "auth.h"
2  #include "ui_auth.h"
3
4  auth::auth(QWidget *parent) :
5  QWidget(parent),
6  ui(new Ui::auth)
7  {
8      ui->setupUi(this);
9  }
10
11 auth::~auth()
12 {
13     delete ui;
14 }
15
16 void auth::setAcc(Account *a)
17 {
18     this->Acc = a;
19 }
20 void auth::on_up_clicked()
21 {
22     QString s = this->ui->textEdit->toPlainText();
23     string ss = s.toStdString();
24     int z = -1;
25     qDebug("%d", this->Acc->getL());
26     for(int i = 0; i < this->Acc->getL(); i++)
27         if (ss == this->Acc->getAcc(i))
28         {
29             z = i;
30             break;
31         }
32     if(z + 1)
33     {
34         if(this->Acc->getAuth(z) < 2)
35         {
36             this->Acc->setAuth(z, this->Acc->getAuth(z) + 1);

```

```

37     this -> Acc -> save();
38     error *e = new error;
39     e->ShowText("权限升级成功");
40     e->show();
41     this -> ui -> now -> setText(QString::fromStdString(this ->
        Acc->getAuthString(this -> Acc->getAuth(z))));
42     this -> ui -> present -> setText(QString::fromStdString(
        this -> Acc->getAuthString(this -> Acc->getAuth(z) -
        1)));
43 }
44 else
45 {
46     error *e = new error;
47     e->ShowText("该账户已经是普通管理员权限，无法升级");
48     e->show();
49     this -> ui -> now -> setText(QString::fromStdString(this ->
        Acc->getAuthString(this -> Acc->getAuth(z))));
50     this -> ui -> present -> clear();
51 }
52 }
53 }
54
55 void auth::on_down_clicked()
56 {
57     QString s = this -> ui -> textEdit -> toPlainText();
58     string ss = s.toStdString();
59     int z = -1;
60     qDebug("%d", this -> Acc->getL());
61     for(int i = 0; i < this -> Acc->getL(); i++)
62     if (ss == this -> Acc->getAcc(i))
63     {
64         z = i;
65         break;
66     }
67     if(z + 1)
68     {
69         if(this -> Acc->getAuth(z) < 2 && this -> Acc->getAuth(z) >

```



```
0)
{
70     qDebug("%d 等级", this->Acc->getAuth(z));
71     this->Acc->setAuth(z, this->Acc->getAuth(z) - 1);
72     this->Acc->save();
73     error *e = new error;
74     e->ShowText("权限降级成功");
75     e->show();
76     this->ui->now->setText(QString::fromStdString(this->
77         Acc->getAuthString(this->Acc->getAuth(z))));
78     this->ui->present->setText(QString::fromStdString(
79         this->Acc->getAuthString(this->Acc->getAuth(z + 1)
80         )));
81 }
82
83 else if(this->Acc->getAuth(z) == 3)
84 {
85     error *e = new error;
86     e->ShowText("超级管理员无法降级");
87     e->show();
88     this->ui->now->setText(QString::fromStdString(this->
89         Acc->getAuthString(this->Acc->getAuth(z))));
90     this->ui->present->clear();
91 }
92
93 else
94 {
95     error *e = new error;
96     e->ShowText("该账户已经是访客权限，无法降级");
97     e->show();
98     this->ui->now->setText(QString::fromStdString(this->
99         Acc->getAuthString(this->Acc->getAuth(z))));
100    this->ui->present->clear();
101 }
102 }
```

5 数据结构说明

5.1 界面类

5.1.1 登录

```
1 #include <QMainWindow>
2 #include "account.h"
3 #include "error.h"
4 QT_BEGIN_NAMESPACE
5 namespace Ui { class Login; }
6 using namespace std;
7 QT_END_NAMESPACE
8
9 class Login : public QMainWindow
10 {
11     Q_OBJECT
12
13     public:
14     Login(QWidget *parent = nullptr);
15     ~Login();
16     void AddAccount(Account *a);
17
18     public slots:
19     void on_login_clicked();
20     void on_regist_clicked();
21     private:
22     Ui::Login *ui;
23     Account *Acc;
24 };
```

5.1.2 账户

```
1 #include "fstream"
2 #include "QFile"
3 #include "string"
4 #include <QFileDialog>
5 #include <QMainWindow>
6 #include <QDebug>
7
8 #define MAXACCOUNT 100
9 using namespace std;
10
11 class Account
12 {
13     private:
14         string acc[MAXACCOUNT];
15         string pas[MAXACCOUNT];
16         int auth[MAXACCOUNT]; // 0 访客 1 学生 2 老师 3 普通管理员 4 总
            管理员
17         int l;
18         string path;
19     public:
20         Account();
21         string getAcc(int i);
22         string getPas(int i);
23         int getAuth(int i);
24         void addUserLoad(string a, string p, int u);
25         void addUser(string a, string p, int u);
26         void save();
27         int getL();
28         bool compareAcc(string s);
29         bool comparePas(string s);
30         bool compare(string s1, string s2);
31         void load();
32         int getAuthByAcc(string a);
33         void setPas(int i, string s);
34         void setPath(string s);
35         string getPath();
```

```
36 void setAuth(int i, int j);
37 string getAuthString(int i);
38 };
```

5.1.3 书本

```
1 #include "string"
2 #include "QDebug"
3 #include "QFile"
4 #include "QString"
5 #include "error.h"
6 using namespace std;
7
8 class Book
9 {
10     private:
11         string ISBN[100];
12         string name[100];
13         bool status[100];
14         string borrower[100];
15         int no[100];
16         int l;
17     public:
18         void addBook(string i, string n, bool s, string b, int no);
19         void add();
20         int getL();
21         string getISBN(int i);
22         string getName(int i);
23         bool getStatus(int i);
24         string getBorrower(int i);
25         int getNo(int i);
26         void writeBook();
27         void setBorrower(int i, string s);
28         void setStatus(int i, int s);
29         string path;
30         string getPath();
```

```
31 void setPath(string s);  
32 };
```

6 总结

在图书管理系统软件开发的过程中,从系统的规划到分析,再到系统设计、实施、运行和测评等一系列开发步骤,都严格按照软件开发流程进行,同时,在开发的过程中),对原有的初步方案不断修改经过反复调试,系统可实现登录、注册、图书管理、读者管理、借阅管理以及新书订购等功能、运行结果表明,该系统可满足小型图书馆的书籍管理和借阅工作。下一步的工作是进一步扩展属软件功能。扩大应用范围,并进一步提高软件的安全性等指标,使图书管理系统的功能更加完善。

参考文献

- [1] 王玉庆. 基于 Java 的图书查询系统设计与实现 [J]. 信息与电脑 (理论版),2021,33(02):138-140.
- [2] 李萍, 李芳. 基于 B/S 结构的医院图书管理系统的设计与实现 [J]. 泰山学院学报,2013,35(03):88-93.
- [3] 曹定舟, 陈云忠, 缪毅. 医院图书借阅管理系统的设计与应用 [J]. 中国医疗设备,2009,24(04):90-91.
- [4] 刘帅. 图书管理信息系统的设计与实现 [D]. 吉林大学,2011.
- [5] 王思楠. 图书馆流通过程中遇到的部分问题以及解决办法——以天津市少年儿童图书馆为例 [J]. 内蒙古科技与经济,2020(12):122-123.

致 谢

通过本课设任课老师金世双老师的指导，我们小组完成了本项课程设计。

感谢你, 感谢他和她, 感谢大家.

为期近一学期的论文写作和软件设计即将画上一个圆满的句号，在论文写作的过程中，从论文的选题到确定思路，从资料的搜集、提纲的拟定到内容的写作与修改，继而诸多观点的梳理。

论文的点评中总是闪烁着智慧的火花，与他的每次交谈我都能从中获益。他渊博的学识，敏锐的学术洞察力，严谨的治学态度，一丝不苟的负责精神，以及对学生孜孜不倦的教诲都给予了我极其深刻的印象，让我受益匪浅。在此，谨向金老师表示我最衷心地感谢和最诚挚的敬意。

同时，也向从课程设计来所有教授过我和帮助过我的教授老师表示感谢，感谢您们对我的谆谆教诲、耐心指导和无私的帮助。

感谢我的同学和朋友们，感谢你们在我论文写作过程中给予我的鼓励、关心和无私的帮助。

最后，衷心地感谢我的家人，感谢你们一直以来给予我的支持和鼓励。