

浙江大学实验报告

课程名称： 数据库系统原理 实验类型： 实验 5

实验项目名称： 设计并实现一个精简的图书管理系统

学生姓名： 沈丹琼 专业： 数字媒体技术 学号： 3170104112

同组学生姓名： 无 指导老师： 陈纬奇

实验地点： 紫金港 实验日期： 2019 年 4 月 27 日

目录

1	实验目的和要求	3
1.1	实验目的	3
1.2	系统需求	3
1.2.1	基本数据对象	3
1.2.2	基本功能模块	4
2	主要仪器设备	4
2.1	数据库平台	4
2.2	开发工具	5
3	系统功能介绍	5
3.1	图书查询功能	5
3.1.1	输入查询属性	5
3.1.2	输入相应属性信息	5
3.1.3	输入排序方式	6
3.1.4	查询结果显示	7
3.1.5	更改排序方式	7
3.1.6	翻页	8
3.1.7	返回上一级	8
3.2	图书管理员操作综述	9
3.2.1	登陆	9
3.2.2	图书管理功能	9
3.3	图书入库功能	10
3.3.1	单本导入	10
3.3.2	文件导入	11
3.4	管理借书卡	13
3.4.1	增加借书卡	13
3.4.2	删除借书证	14
3.5	借书	15
3.5.1	输入借书卡号	15
3.5.2	点击“click to borrow”，输入相关信息	17
3.5.3	借书结果显示	18
3.5.4	返回上一级	21
3.6	还书功能	21
3.6.1	输入学号，显示该用户已借的全部书籍	21
3.6.2	成功还书流程	22
3.6.3	后台校验数据	24
3.7	查看书单功能	24
4	系统构架描述	25
4.1	系统功能构架	25
4.2	系统实现构架	27
4.2.1	函数清单及简介	27
4.2.2	函数调用关系	27
5	各模块流程图与实现技术	28
5.1	查询模块	28
5.1.1	流程图	28
5.1.2	实现技术	28
5.2	管理员登陆	31
5.2.1	流程图	31

5.2.2 实现技术	32
5.3 图书入库	33
5.3.1 流程图	33
5.3.2 实现细节	33
5.4 借书卡管理	34
5.4.1 流程图	34
5.4.2 实现细节	34
5.5 借书	35
5.5.1 流程图	35
5.5.2 实现技术	35
5.6 还书	37
5.6.1 流程图	37
5.6.2 实现技术	37
5.7 查看书库	39
5.7.1 流程图	39
5.7.2 实现技术	39
6 数据库表结构设计	40
7 心得讨论与改进空间	41
7.1 改进空间	41
7.1.1 图书入库功能	41
7.1.2 增加查看借书记录的按钮	41
7.1.3 程序架构还有改进空间	41
7.2 心得体会	41
7.2.1 对 ODBC 接口的使用	41
7.2.2 宏定义的使用	41

1 实验目的和要求

1.1 实验目的

设计并实现一个精简的图书管理系统，要求具有图书入库、查询、借书、还书、借书证管理等功能。
通过该图书馆系统的设计与实现，提高学生的系统编程能力，加深对数据库系统原理及应用的理解。

1.2 系统需求

1.2.1 基本数据对象

对象名称	包含属性
------	------

书	书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存
借书证	卡号, 姓名, 单位, 类别 (教师 学生等)
管理员	管理员 ID, 密码, 姓名, 联系方式
借书记录	卡号, 借书证号, 借期, 还期, 经手人 (管理员 ID)

1.2.2 基本功能模块

模块名称	功能描述
管理员登陆	输入管理员 ID, 密码; 登入系统 或 返回 ID/密码 错误.
图书入库	<p>1. 单本入库</p> <p>2. 批量入库 (方便最后测试)</p> <p>图书信息存放在文件中, 每条图书信息为一行. 一行中的内容如下 (书号, 类别, 书名, 出版社, 年份, 作者, 价格, 数量)</p> <p>Note: 其中 年份、数量是整数类型; 价格是两位小数类型; 其余为字符串类型</p> <p>Sample: (book_no_1, Computer Science, Computer Architecture, xxx, 2004, xxx, 90.00, 2)</p>
图书查询	<p>要求可以对书的 类别, 书名, 出版社, 年份(年份区间), 作者, 价格(区间) 进行查询. 每条图书信息包括以下内容: (书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存)</p> <p>可选要求: 可以按用户指定属性对图书信息进行排序. (默认是书名)</p>
借书	<p>1.输入借书证卡号 显示该借书证所有已借书籍 (返回, 格式同查询模块)</p> <p>2.输入书号 如果该书还有库存, 则借书成功, 同时库存数减一。 否则输出该书无库存, 且输出最近归还的时间。</p>
还书	<p>1.输入借书证卡号 显示该借书证所有已借书籍 (返回, 格式同查询模块)</p> <p>2.输入书号 如果该书在已借书籍列表内, 则还书成功, 同时库存加一。 否则输出出错信息。</p>
借书证管理	增加或删除一个借书证.

除图书查询功能外, 其余功能模块都应该由**图书管理员**操作。

2 主要仪器设备

2.1 数据库平台

SQL Server 或 MySQL

其中 MySQL 详细信息请参见 <http://www.mysql.com>

MySQL APIs:

1. MySQL ODBC 3.51
2. MySQL JDBC 5.0
3. MySQL PHP APIs

有关 MySQL 的安装, 请参看有关参考书。

2.2 开发工具

Visual studio

3 系统功能介绍

3.1 图书查询功能

3.1.1 输入查询属性

点击首页的搜索框，输入查询的属性类型（1 category;2 title;3 press;4 year;5 author;6 price），可查询的属性类型包括类型种类，书名，出版社，出版年份，作者，价格。（图 3-1）

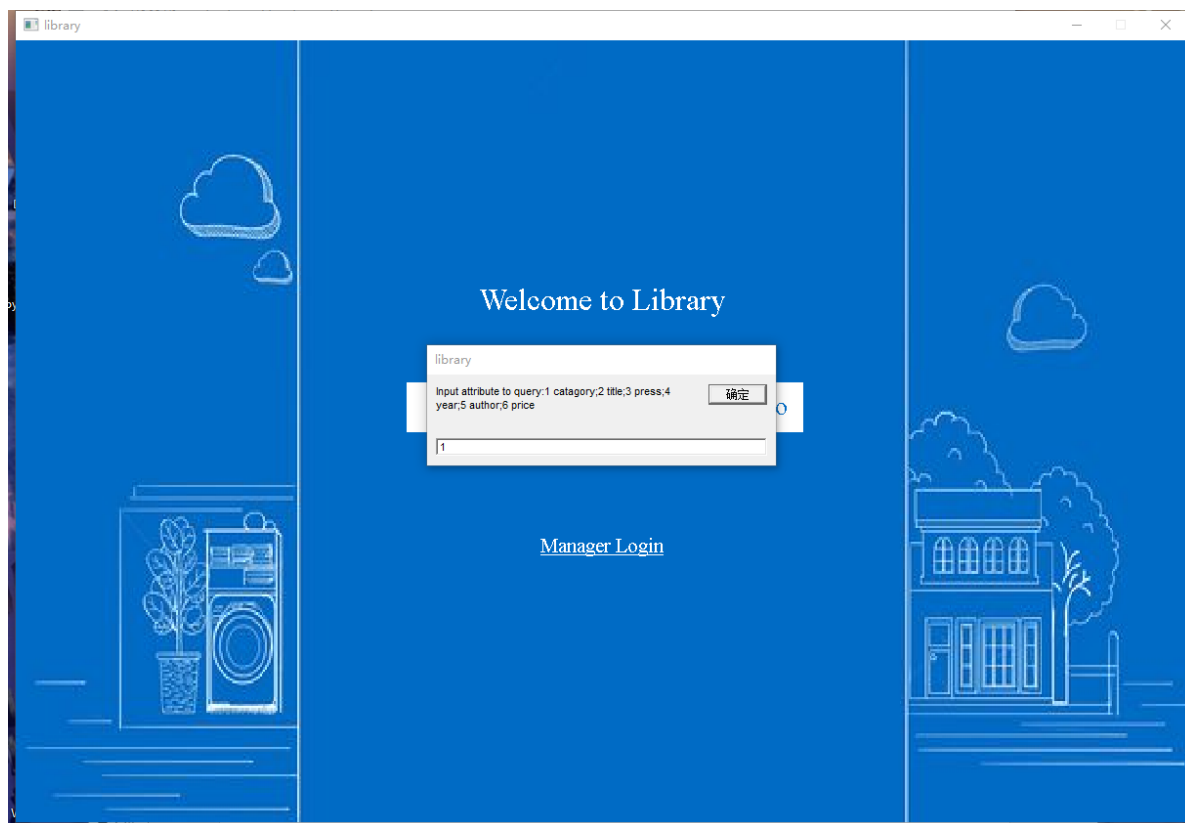


图 3-1

3.1.2 输入相应属性信息

随后输入相应的属性信息。（图 3-2）

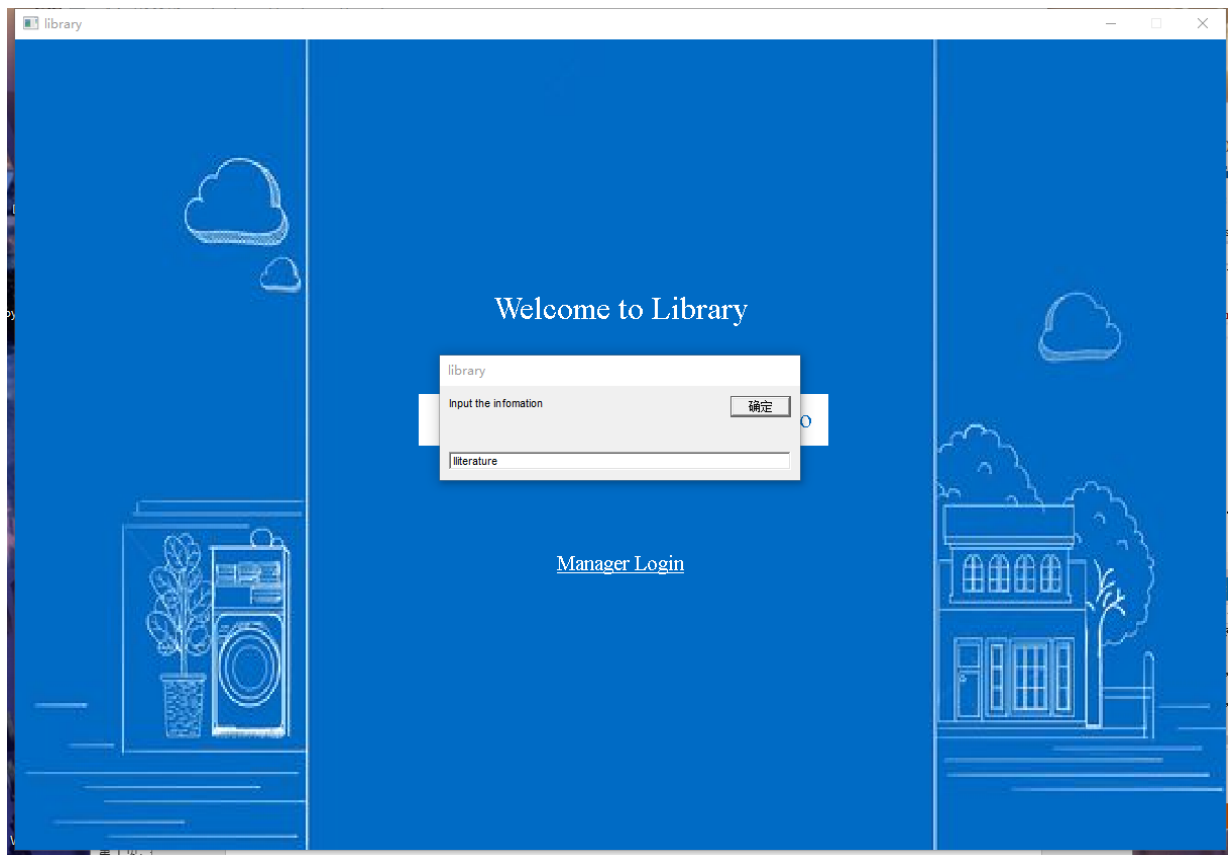


图 3-2

3.1.3 输入排序方式

输入字符选择相应的排序方式（1 category 升序;2 title 升序;3 press 升序;4 year 升序;5 author 升序; 6 price 升序;a category 降序;b title 降序;c press 降序;d year 降序;e author 降序;f price 降序）（图 3-3）

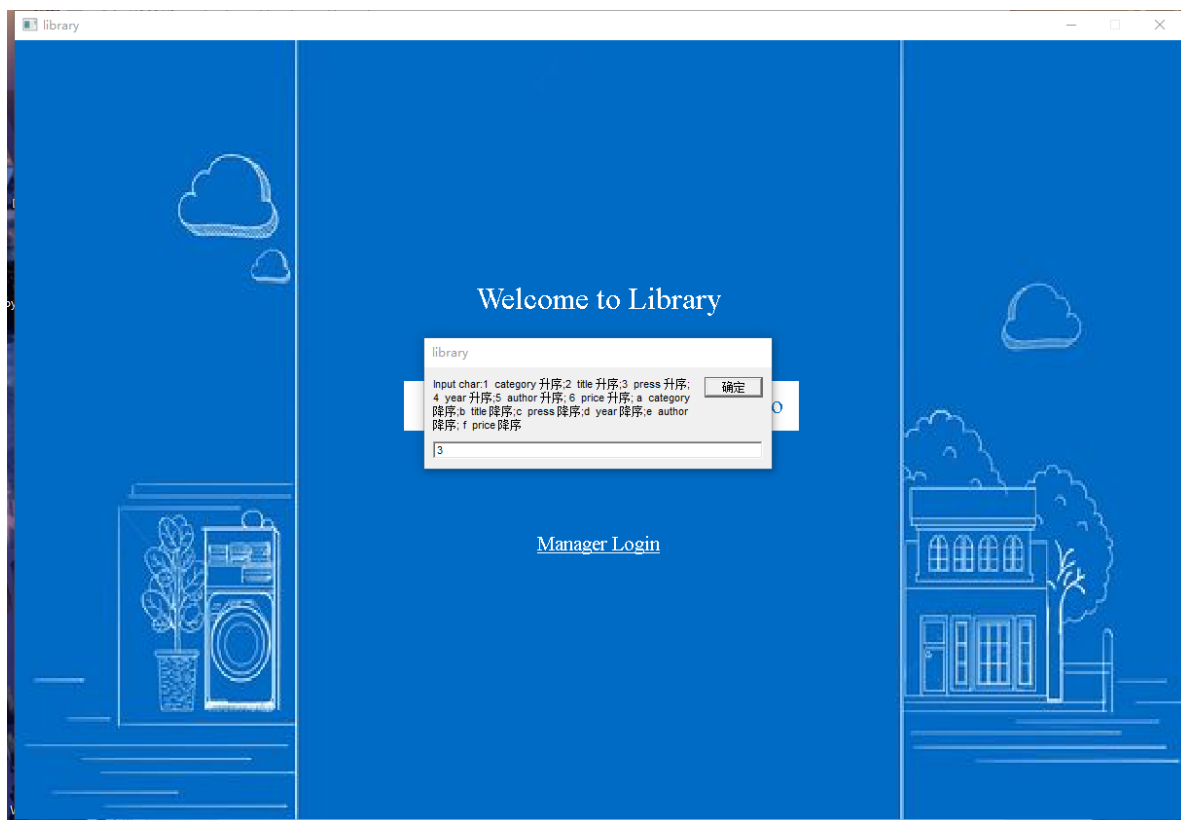


图 3-3

3.1.4 查询结果显示

输入查询属性，相关信息及排序方式之后，跳转到查询结果界面。（图 3-4 查询结果界面）

library

↶

↷

Category

Title

Press

Year

Author

Price

ISBN	CATAGORY	TITLE	PRESS	DATE	AUTHOR	PRICE	TOTAL	STOCK
9781405368919	Iliterature	DK_Eyewitness_Travel_Guide_Norway	Dorling_Kindersley_Ltd	2012	"Evensberg,I_Snoore"	244.64	50	40
9781405394314	Iliterature	The_Gluten-Free_Cookbook	Dorling_Kindersley_Ltd	2012	No_Contributor	277.28	50	40
9781405394406	Iliterature	The_Motorbike_Book	Dorling_Kindersley_Ltd	2012	No_Contributor	408	50	40
9780751362114	Iliterature	The_Secret_Life_of_Trees	Dorling_Kindersley_Ltd	2012	"Chevallier,Chana"	48.8	50	40
9781405360777	Iliterature	DK_Eyewitness_Travel_Guide_India	Dorling_Kindersley_Ltd	2011	No_Contributor	326.24	50	40
9781409386575	Iliterature	DK_Pocket_Eyewitness_Sharks	Dorling_Kindersley_Ltd	2012	No_Contributor	67.89	50	40
9781405368933	Iliterature	DK_Eyewitness_Travel_Guide_Cosmca	Dorling_Kindersley_Ltd	2012	No_Contributor	212	50	40
9781905490578	Iliterature	Pub_Walks_in_Underhill_Country	Penguin_Adult	2011	"Seguit,Nat"	212.56	50	40
9780140279108	Iliterature	New_Boy	Penguin_Adult	1998	"Sutcliffe,William"	114.08	50	40
9780140367225	Iliterature	The_Robinson_Crusoe	Penguin_Adult	2004	"Defoe,Daniel"	114.08	50	40
9780140424744	Iliterature	A_Shropshire_Lad_and_Other_Poems	Penguin_Adult	2010	"Housman,A.E."	122.56	50	40
9780140238899	Iliterature	NIGHT_OF_THE_KRAIT	Penguin_Books_India	2000	"WARRIER,SHASHI"	108.69	50	40
9780140060898	Iliterature	The_Mosquito_Coast	Penguin_Books_Ltd	2011	"Theroux,Paul"	135.78	50	40
9780140071818	Iliterature	The_Kingdom_by_the_Sea	Penguin_Books_Ltd	2011	"Theroux,Paul"	149.49	50	40

⏪

⏩

图 3-4 查询结果界面

查询结果每页显示 16 行数据，按照用户指定的方式进行排序。点击“<”“>”按钮可进行翻页。
点击上方属性按钮上下的“^”“v”按钮可以更改查询结果排序方式，“^”表示按照其下方显示的属性升序排列，“v”表示按照其上方显示的属性降序排列。
点击页面左上方查询按钮，返回上一级菜单。

3.1.5 更改排序方式

按照不同属性排序。以下检测按照价格升序排序，点击 PRICE 上方“^”按钮。（图 3-5 按照价格升序排列）

library

↶

↷

Category

Title

Press

Year

Author

Price

ISBN	CATAGORY	TITLE	PRESS	DATE	AUTHOR	PRICE	TOTAL	STOCK
9780751362114	Iliterature	The_Secret_Life_of_Trees	Dorling_Kindersley_Ltd	2012	"Chevallier,Chana"	48.8	50	40
9780140367201	Iliterature	Tales_of_Mystery_and_Terror	Penguin_Publishing	1995	"Poe,Edgar_Allan"	56.78	50	40
9780140260656	Iliterature	The_Little_Book_of_Calm	Penguin_Books_Ltd	1996	"Wilson,Paul"	67.89	50	40
9781409386575	Iliterature	DK_Pocket_Eyewitness_Sharks	Dorling_Kindersley_Ltd	2012	No_Contributor	67.89	50	40
9780140374667	Iliterature	The_Wesppuppy_on_Holiday	Penguin_Publishing	1995	"Wilson,Jacqueline"	69.55	50	40
9780140424591	Iliterature	Selected_Works	Penguin_Publishing	1974	"Rochester,Earl_of"	69.55	50	40
9780140362886	Iliterature	Mr_Majuka_and_the_School_Inspector	Penguin_Publishing	2004	"Carpenter,Humphrey"	71.02	50	40
9780140363821	Iliterature	The_Last_Polar_Bears	Penguin_Publishing	2003	"Hosse,Harry"	71.02	50	40
9780140364194	Iliterature	The_Bully	Penguin_Publishing	1995	"Needle,Jan"	71.02	50	40
9780140366419	Iliterature	Mr_Majuka_and_the_Ghost_Train	Penguin_Publishing	2004	"Carpenter,Humphrey"	71.02	50	40
9780140367065	Iliterature	The_Story_of_the_Treasure_Seekers	Penguin_Publishing	1995	"Nesbit,E."	71.02	50	40
9780140367362	Iliterature	The_Water_Babies	Penguin_Publishing	1995	"Kingsley,Charles"	71.02	50	40
9780140367485	Iliterature	The_Lost_World	Penguin_Publishing	2001	thur_Cowan_Doyle,Newsham,J	71.02	50	40
9780140367492	Iliterature	The_Prince_and_the_Pauper	Penguin_Publishing	2004	"Twain,Mark"	71.02	50	40

⏪

⏩

图 3-5 按照价格升序排列

3.1.6 翻页

点击上下翻页按钮。（图 3-6 翻页）



图 3-6 翻页

3.1.7 返回上一级

返回按钮，回到上一级。（图 3-7 返回）

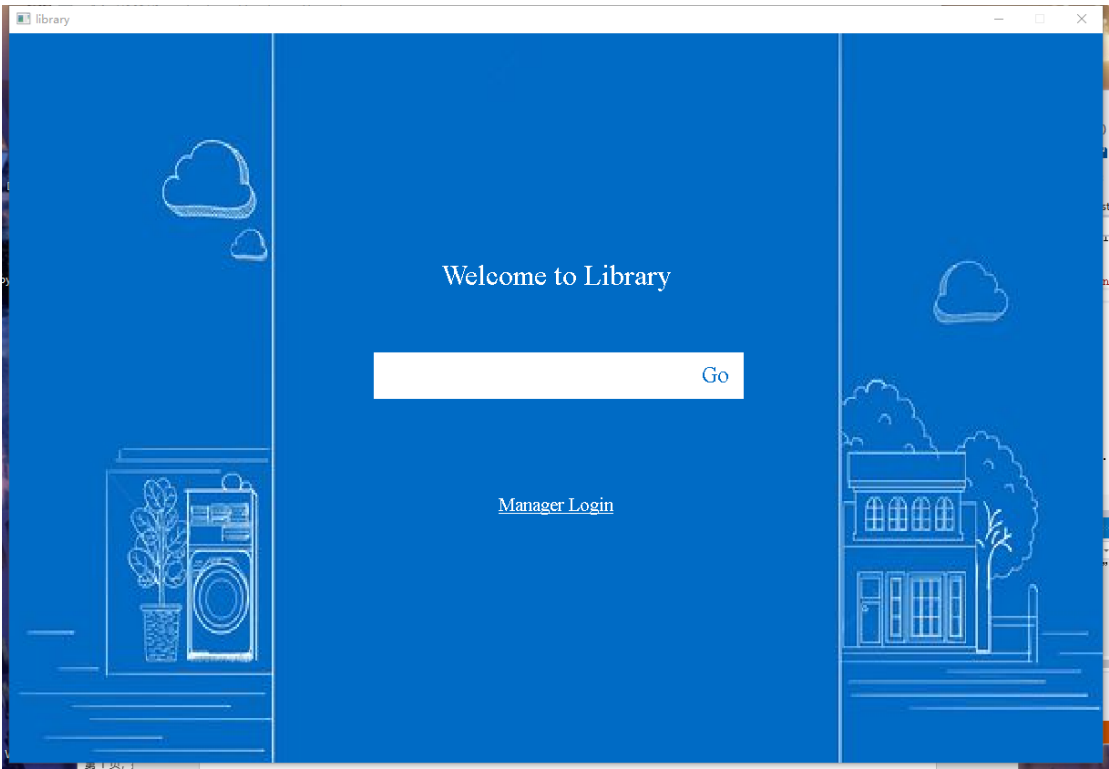


图 3-7 返回

3.2 图书管理员操作综述

3.2.1 登陆

图书管理员首先需要登陆，才能进行后续操作。点击首页上的 manager login 按钮，输入账号和密码。

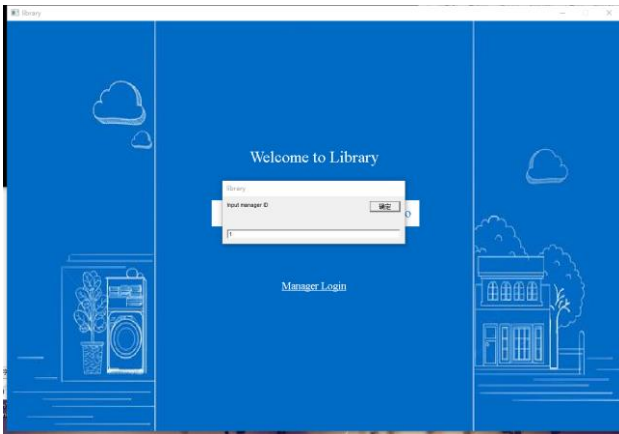


图 3-8 输入账号

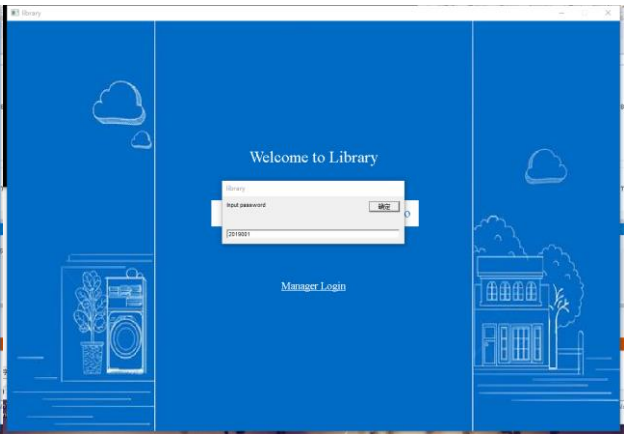


图 3-9 输入密码

密码错误，无法进入管理系统。

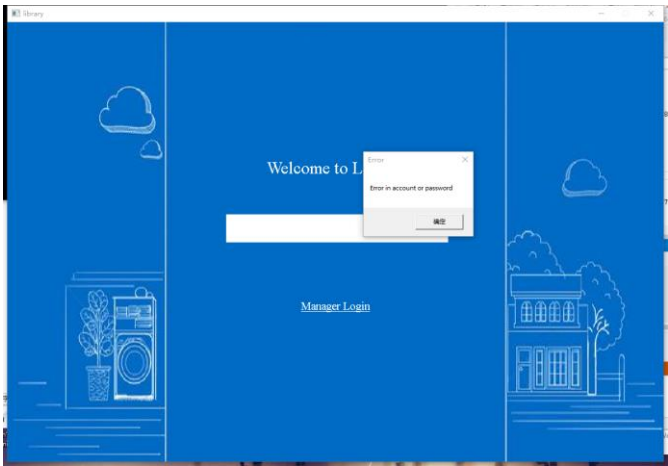


图 3-10 密码错误提示

密码正确，跳转到图书管理界面。

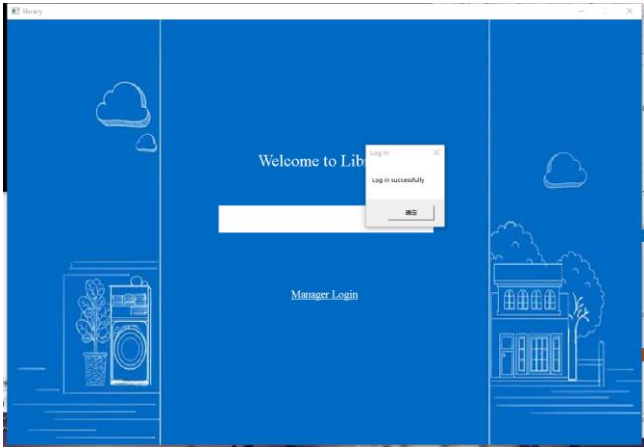


图 3-11 成功登录提示

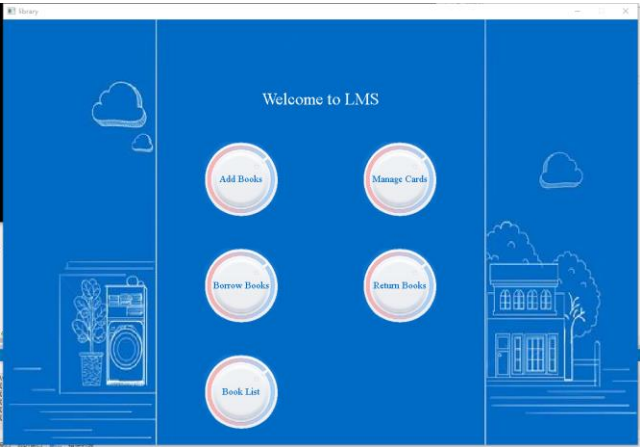


图 3-12 进入图书管理界面

3.2.2 图书管理功能

图书管理界面中有五个功能，包括图书入库（add books）、借书卡管理（manage cards）、借书（borrow books）、还

书（return books）、查看书单（Book list）。以下章节将具体介绍。

3.3 图书入库功能

导入书库有两种模式，1 是单本导入，2 是导入 txt 文件。
首先进行模式选择。

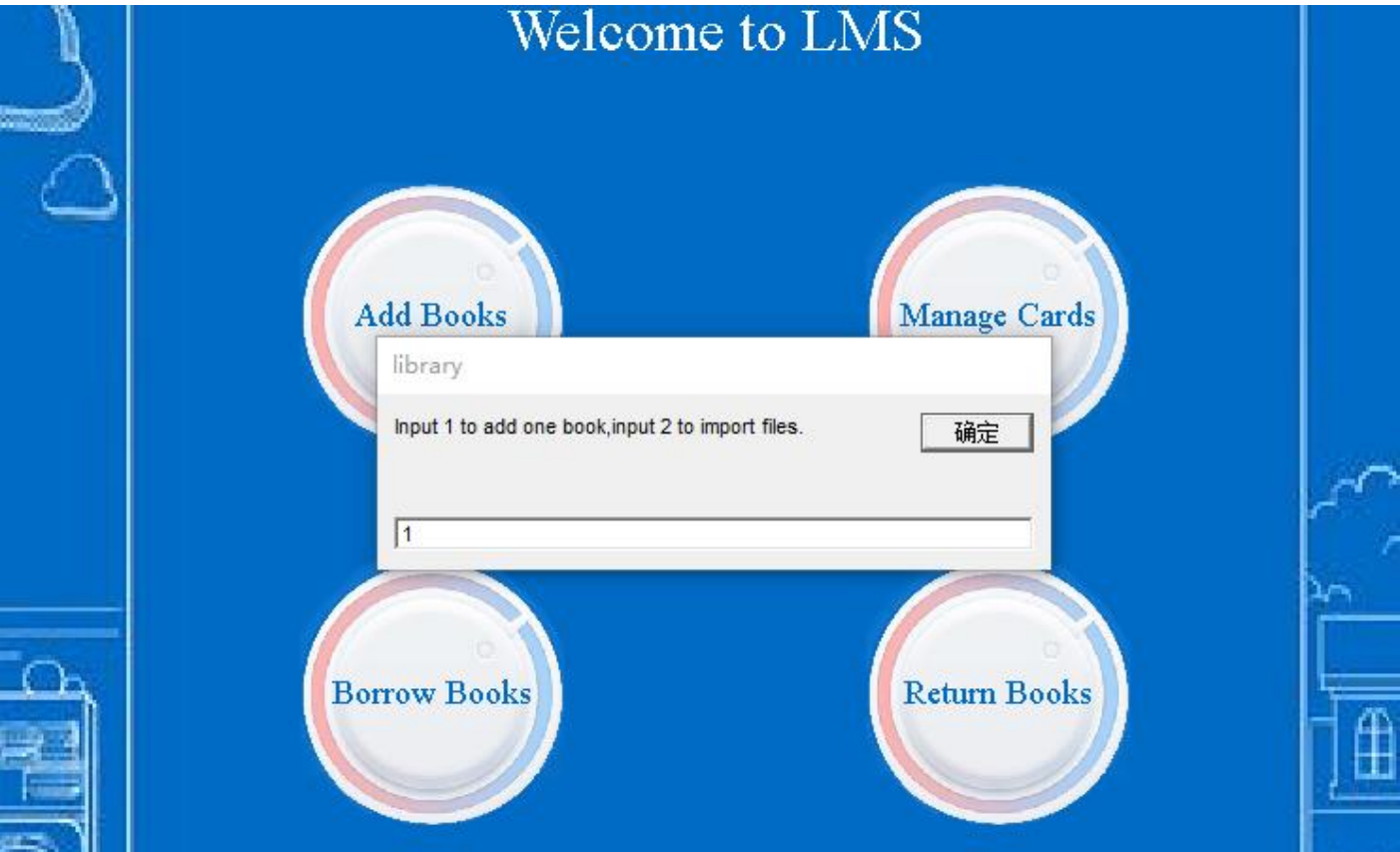
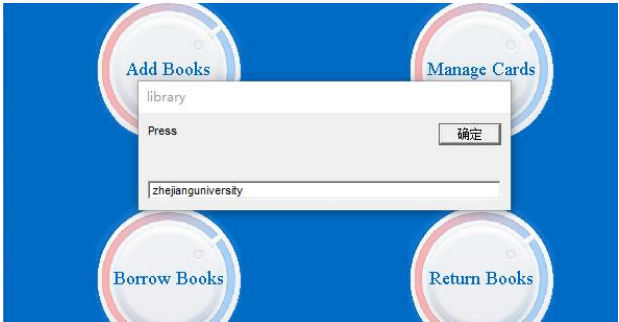
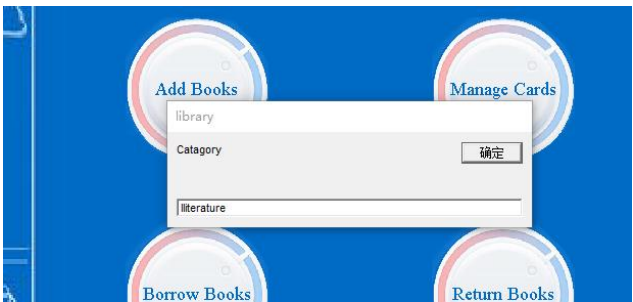
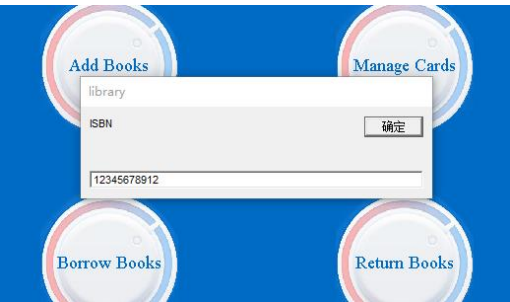


图 3-13 选择单本导入模式或文件导入模式

3.3.1 单本导入

1、输入基本信息，包括 ISBN,category 等。



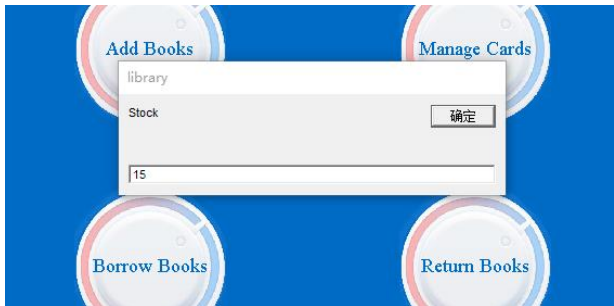
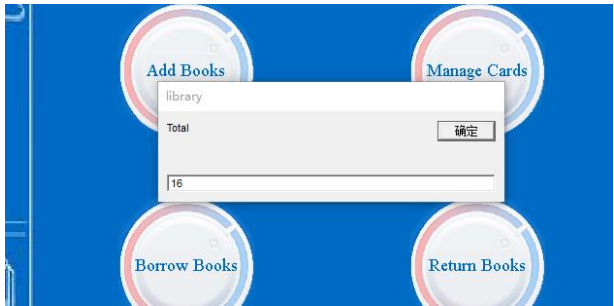
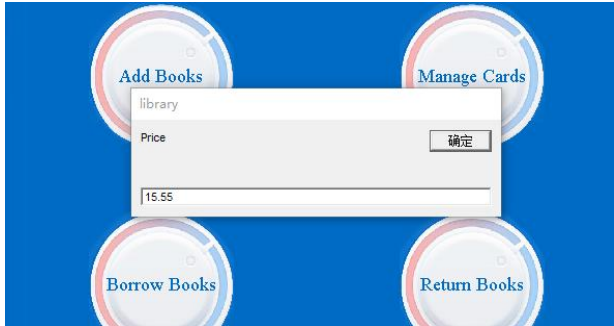
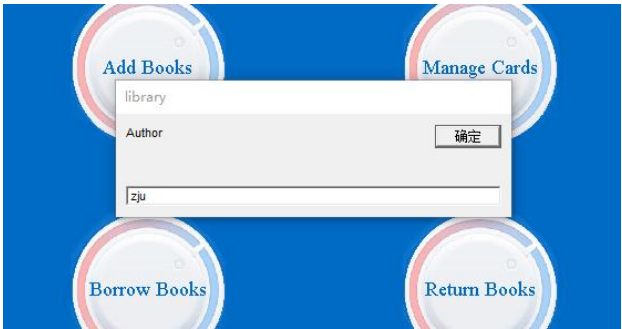


图 3-14 输入图书相关信息

2、输入过程中，如有错误，可以输入 0 取消；若输入信息均正确，输入 1 确认。

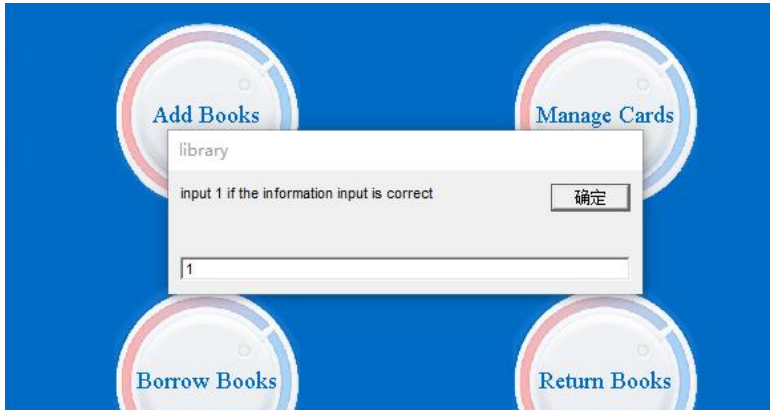


图 3-15 确认输入情况

3、在 mysql 中查看导入结果，插入成功。

	ISBN	catagory	title	press	year	author	price	total	stock
►	12345678912	Iliterature	database	zhejianguniversity	2019	zju	15.55	16	15

图 3-16 在 mysql 中查看结果

3.3.2 文件导入

1、输入当前目录下的文件名称或者文件的绝对路径



图 3-17 输入路径

2、查看控制台中打印结果

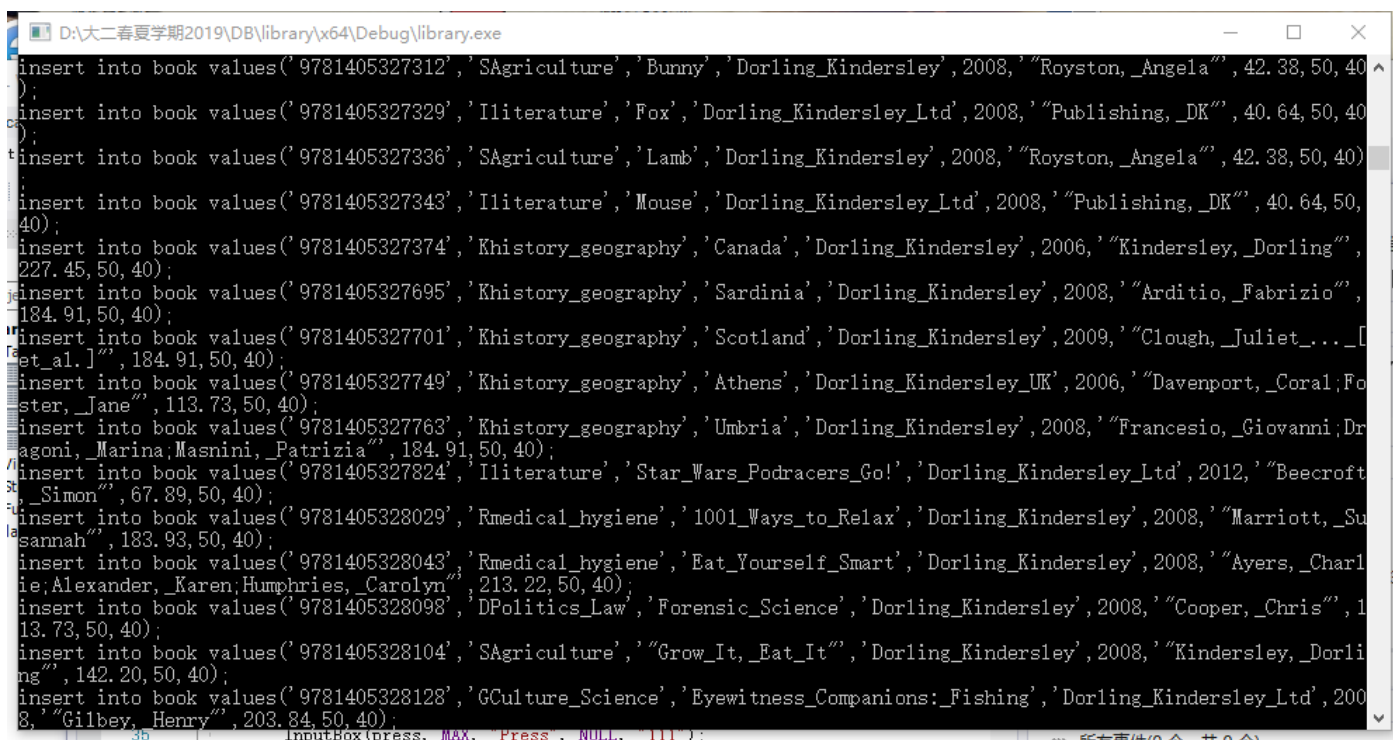


图 3-18 后台数据校验

3、查看数据库

ISBN	catagory	title	press	year	author	price	
9780140071818	Iliterature	The_Kingdom_by_the_Sea	Penguin_Books_Ltd	2011	"Theroux, Paul"	149.49	5
9780140079326	Iliterature	Treasures_of_Time	Penguin_Publishing	2000	"Lively, Penelope"	127.96	5
9780140081770	Iliterature	Captain_Hornblower_R.N.	Penguin_Publishing	1987	"Forester, C.S."	184.91	5
9780140081794	Khistry_geography	How_to_be_a_Brit	Penguin_Publishing	1986	"Mikes, George"	127.96	5
9780140081954	Iliterature	The_Mortgaged_Heart	Penguin_Books_Ltd	2012	"McCullers, Carson"	163.04	5
9780140083583	Iliterature	Clock_Without_Hands	Penguin_Books_Ltd	2012	"McCullers, Carson"	195.68	5
9780140089585	Iliterature	The_Second_Rumpole_Omni...	Penguin_Publishing	1988	"Mortimer, John"	212.56	5
9780140099379	HLanguage	Portuguese_Phrase_Book	Penguin_Publishing	1997	"Figueiredo, Antonio..."	98.02	5
9780140100242	Rmedical_hygiene	The_Complete_F-Plan_Diet	Penguin_Publishing	1987	"Eyton, Audrey"	156.44	5
9780140102390	Iliterature	Pack_of_Cards	Penguin_Publishing	2000	"Lively, Penelope"	125.18	5
9780140109795	Iliterature	"Farewell, My Lovely"	Penquin Books Ltd	2005	"Chandler, Raymond"	146.72	5

图 3-19 数据库中校验

3.4 管理借书卡

3.4.1 增加借书卡

1、输入相关信息

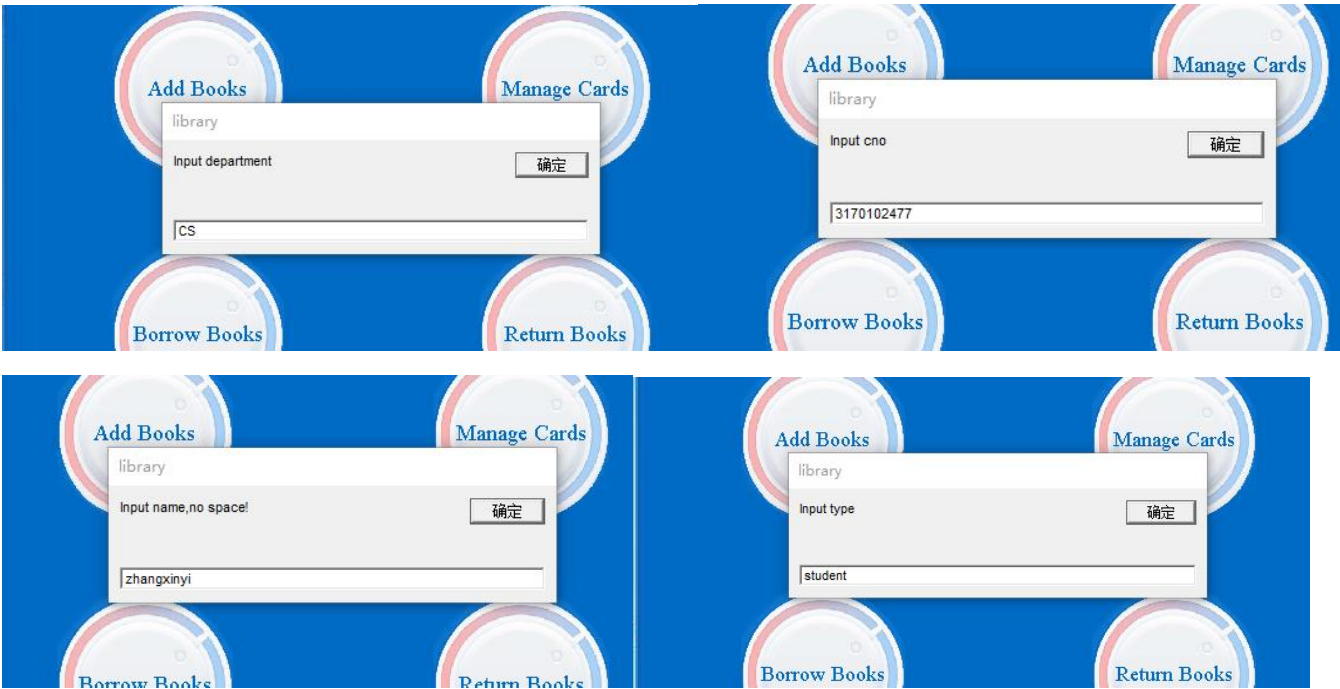


图 3-20 输入卡号、姓名、部门和类型的信息

2、确认信息

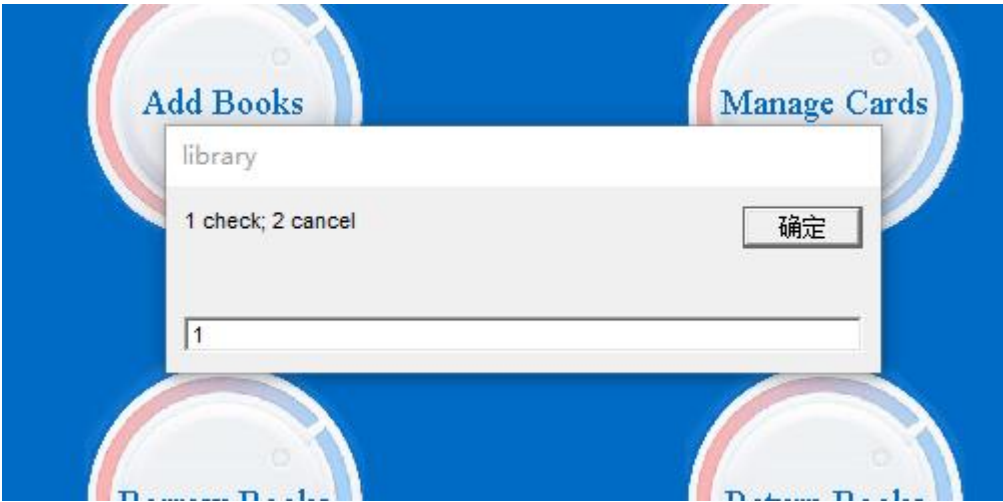


图 3-21 确认信息

3、显示增加结果

(1) 如果原本的 card 表中没有该学号，则可加入

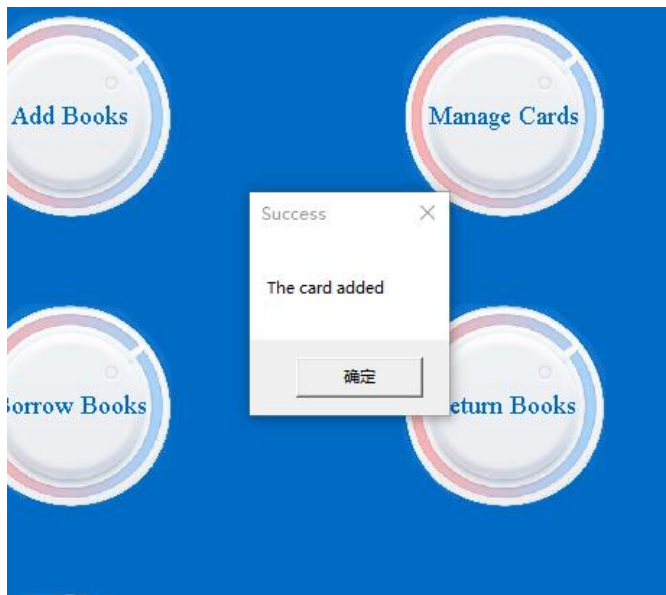


图 3-22 成功

(2) 如果原本的 card 表中有该学号，则不可加入

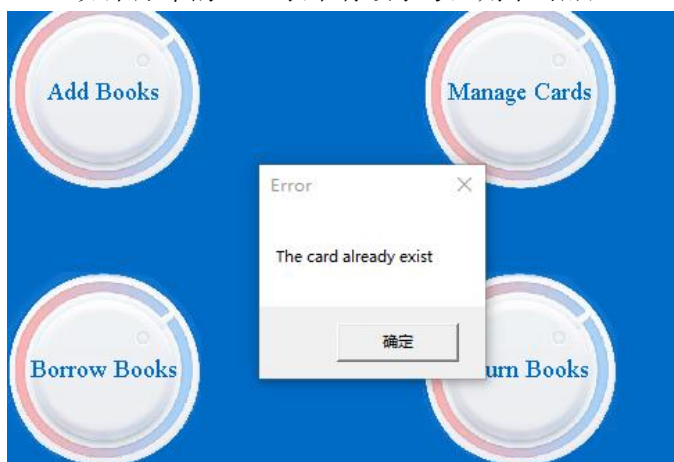


图 3-23 失败

3.4.2 删除借书证

1、输入相关信息

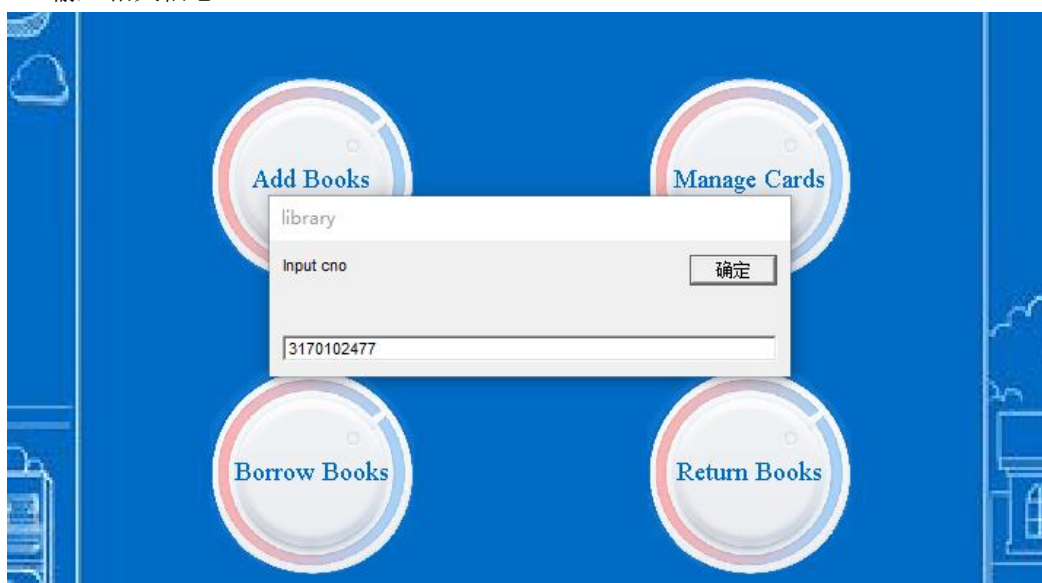


图 3-24 可被删除的

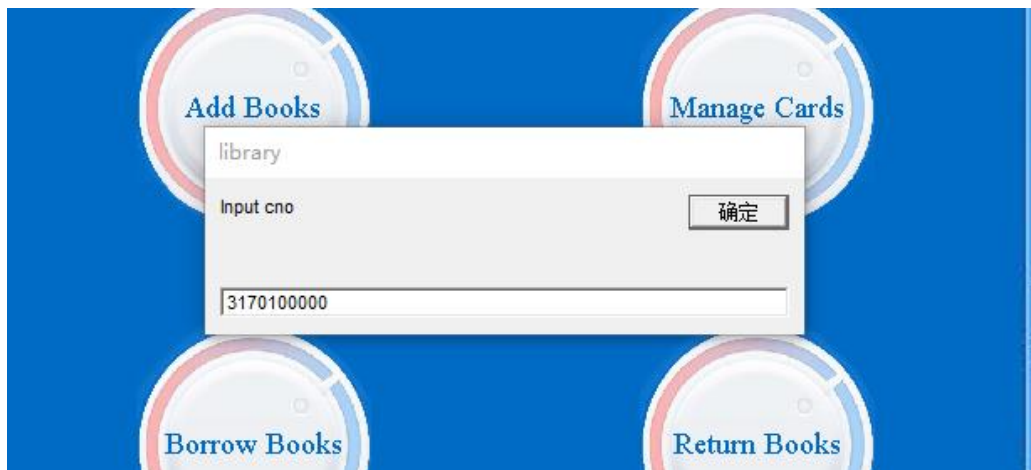


图 3-25 不能被删除的

2、确认

3、删除结果提示

(1) 若该卡还有书为归还，则不可删除

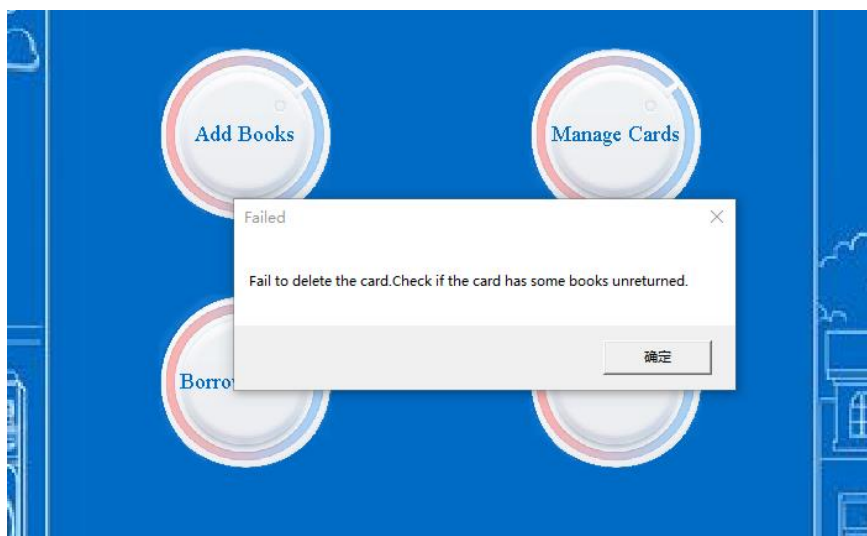


图 3-26 删除失败

(2) 该卡已还清所有书籍，可被删除

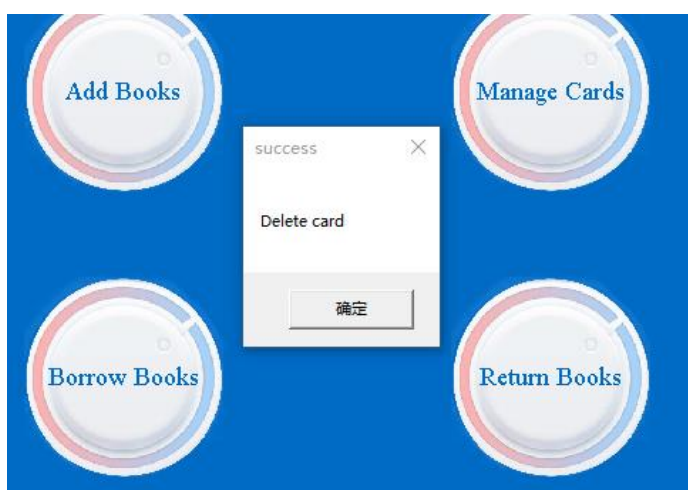


图 3-27 删除成功

3.5 借书

3.5.1 输入借书卡号

输入借书卡号，显示该账户已借的全部书籍，若书籍超过 16 本，可单击翻页查看。

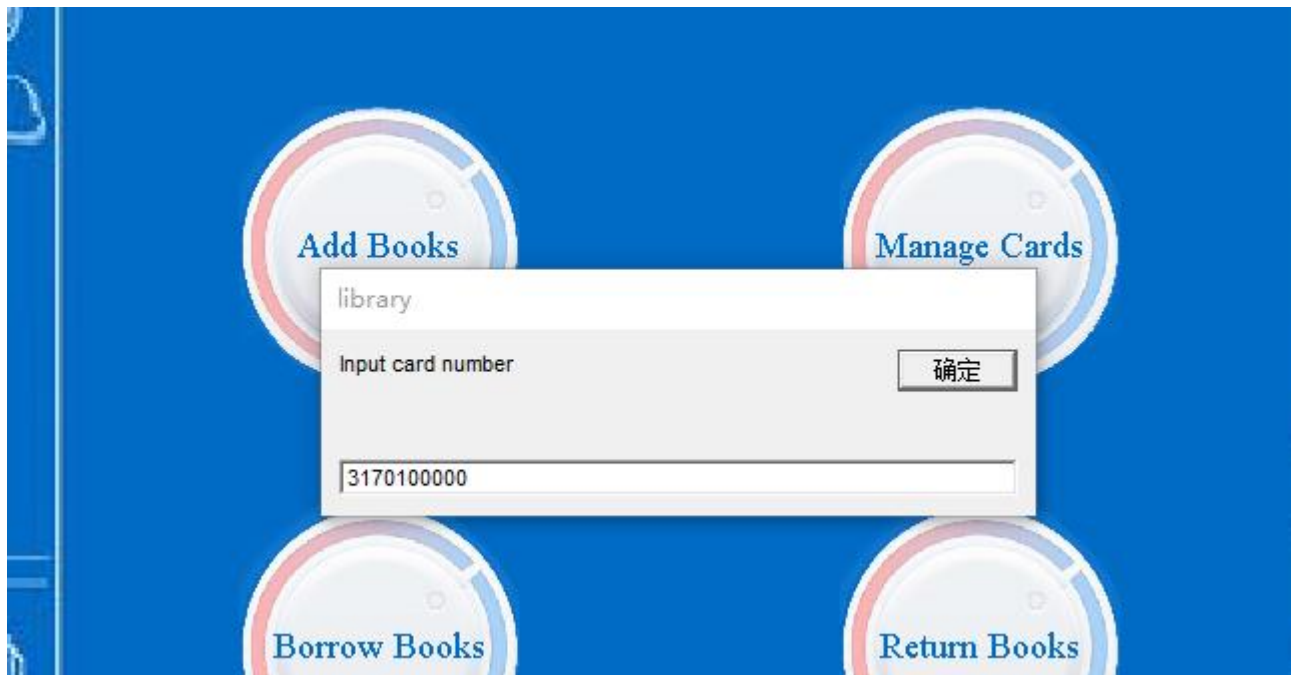


图 3-28 输入借书卡号

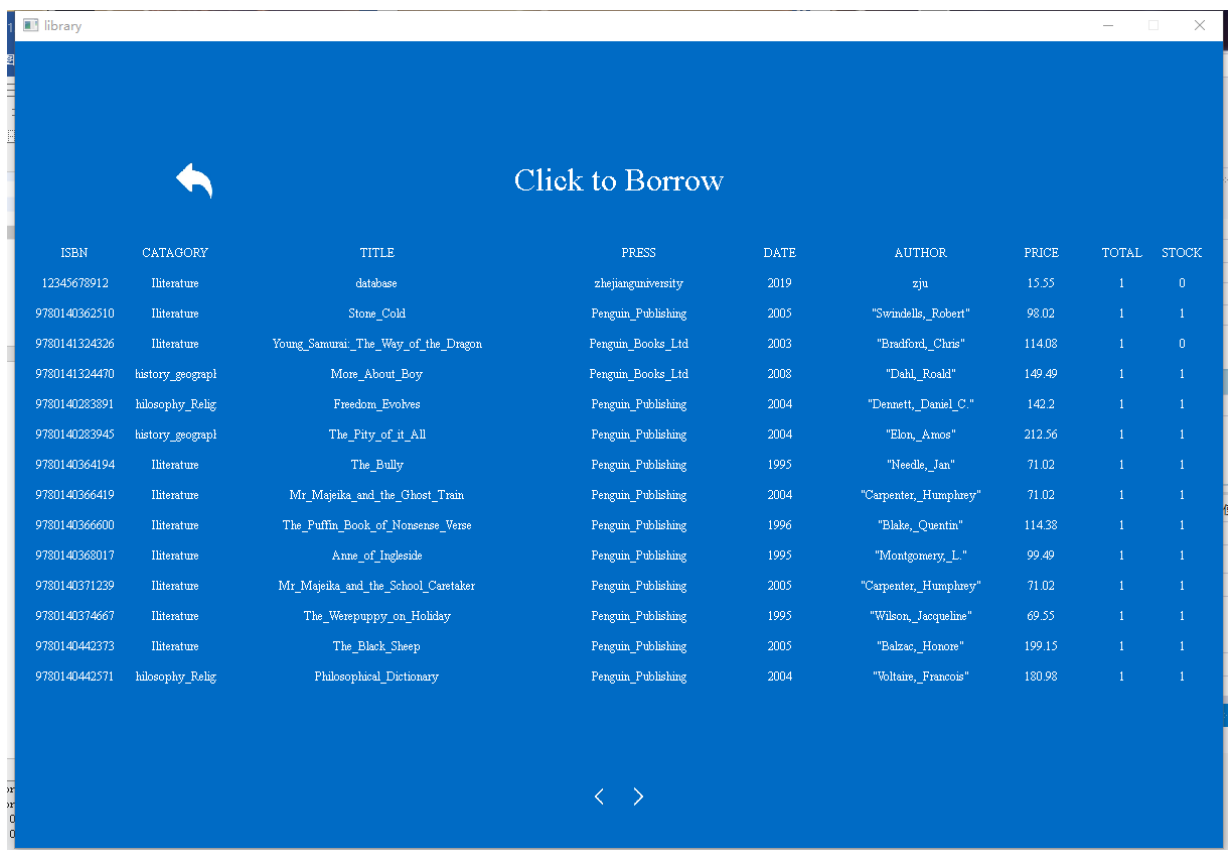


图 3-29 显示该用户已借书籍

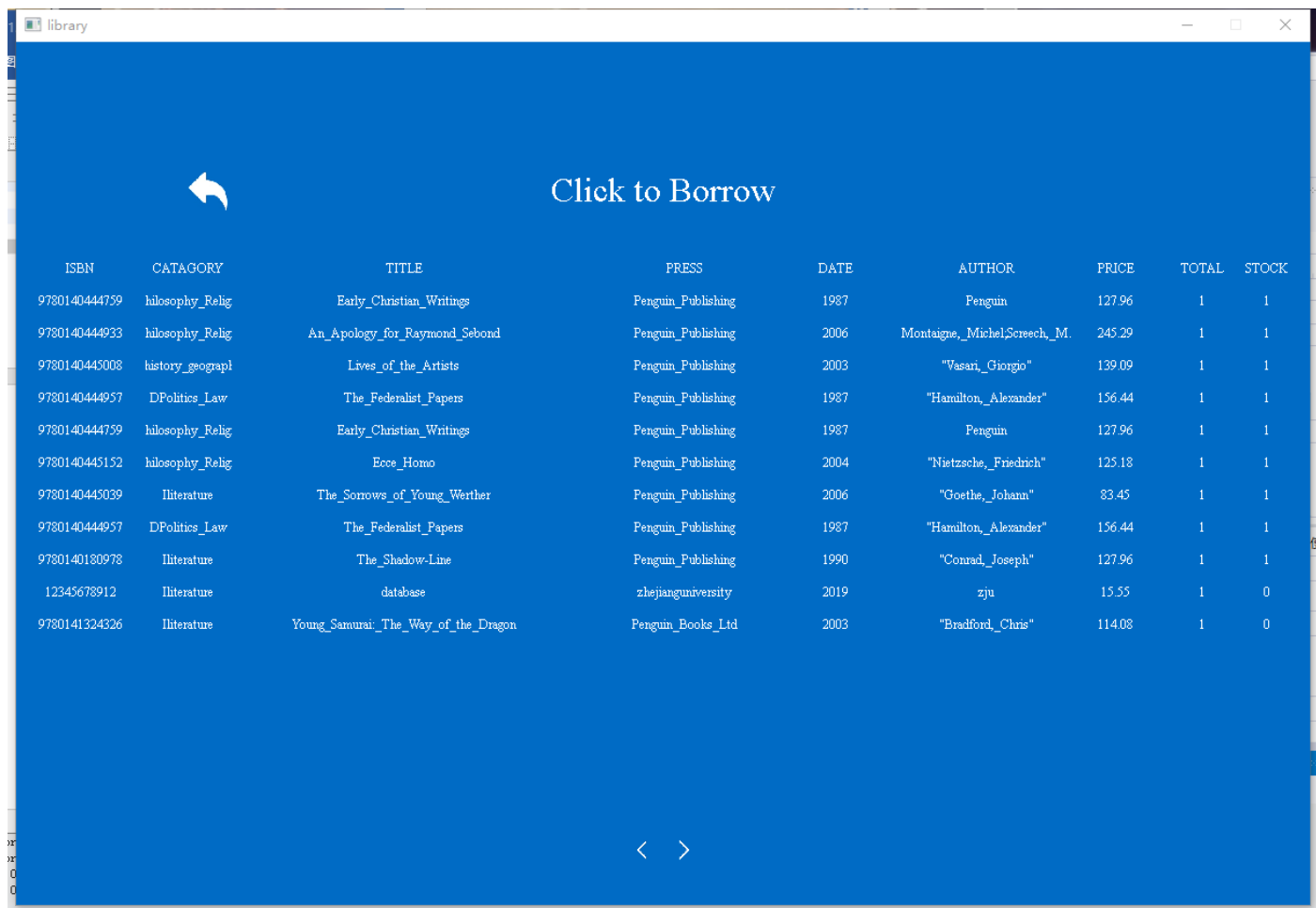


图 3-30 用户已借书籍翻页

3.5.2 点击“click to borrow”，输入相关信息

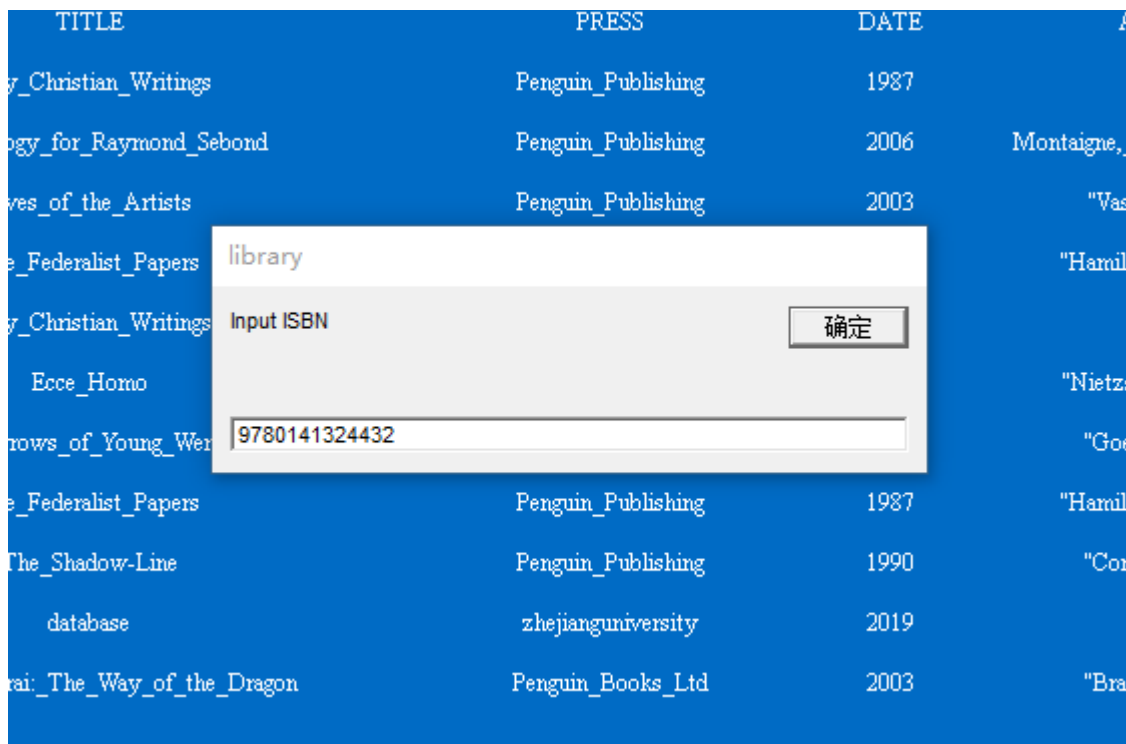


图 3-31 输入 ISBN 进行借书

TITLE	PRESS	DATE	AUTH
Early_Christian_Writings	Penguin_Publishing	1987	Pengu
Apology_for_Raymond_Sebond	Penguin_Publishing	2006	Montaigne_Mich
Lives_of_the_Artists	Penguin_Publishing	2003	"Vasari_G
The_Federalist_Papers			"Hamilton_A
Early_Christian_Writings			Pengu
Ecce_Homo			"Nietzsche,_
he_Sorrows_of_Young_Wer			"Goethe,_J
The_Federalist_Papers	Penguin_Publishing	1987	"Hamilton_A
The_Shadow-Line	Penguin_Publishing	1990	"Conrad,_J
database	zhejianguniversity	2019	zju
Samurai_The_Way_of_the_Dragon	Penguin_Books_Ltd	2003	"Bradford,

library

Input date

确定

2019-4-26

图 3-32 输入借书当天日期

借书信息核对，若输入正确，则继续确认进行借书，否则取消借书。

TITLE	PRESS	DATE	AUTHOR
Early_Christian_Writings	Penguin_Publishing	1987	Penguin
An_Apology_for_Raymond_Sebond	Penguin_Publishing	2006	Montaigne_Michel;Scre
Lives_of_the_Artists	Penguin_Publishing	2003	"Vasari_Giorgio"
The_Federalist_Papers			"Hamilton_Alexand
Early_Christian_Writings			Penguin
Ecce_Homo			"Nietzsche_Friedric
The_Sorrows_of_Young_Wer			"Goethe_Johann"
The_Federalist_Papers	Penguin_Publishing	1987	"Hamilton_Alexand
The_Shadow-Line	Penguin_Publishing	1990	"Conrad_Joseph"
database	zhejianguniversity	2019	zju
Young_Samurai_The_Way_of_the_Dragon	Penguin_Books_Ltd	2003	"Bradford_Chris"

library

1 check,2 cancel

确定

1

图 3-33 核对

3.5.3 借书结果显示

1、该书不在书库中，进行提示

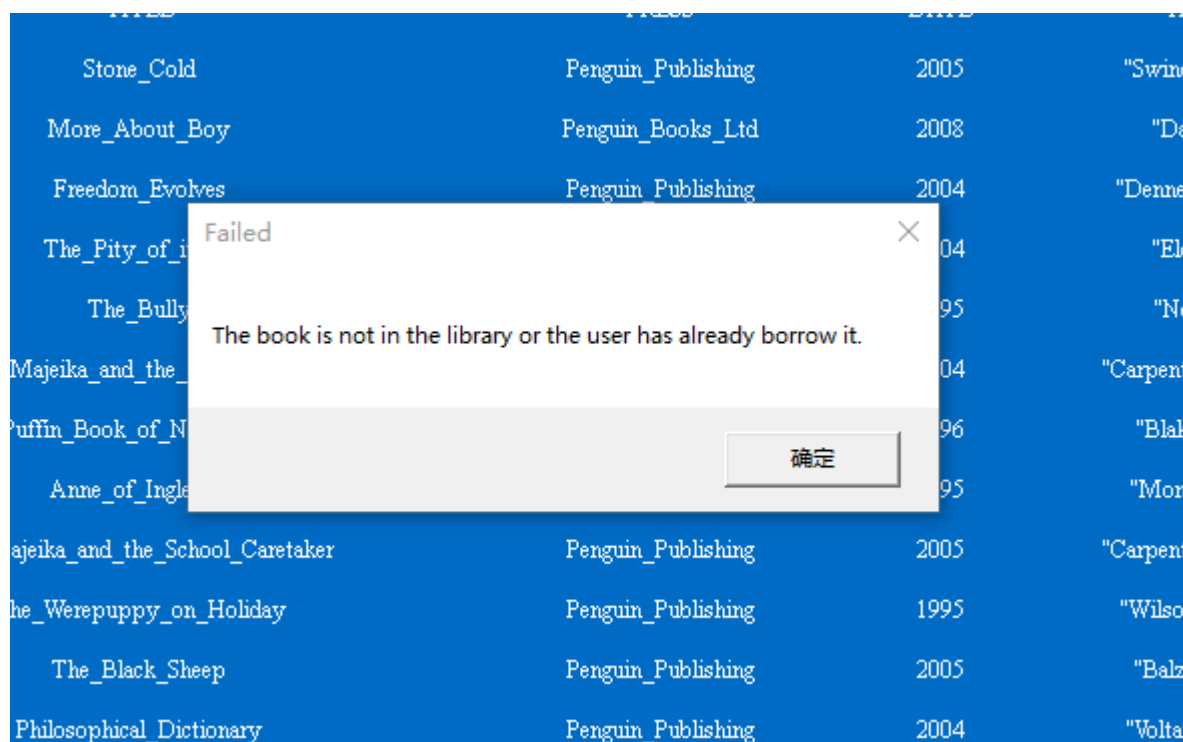


图 3-34 查无此书或该用户已借过该书

2、该书暂无库存，全部借出

若该书无库存，显示该书最久远的一次为归还的时间，以使读者知道离现在最近的一次归还时间。借书时限为 30 天。

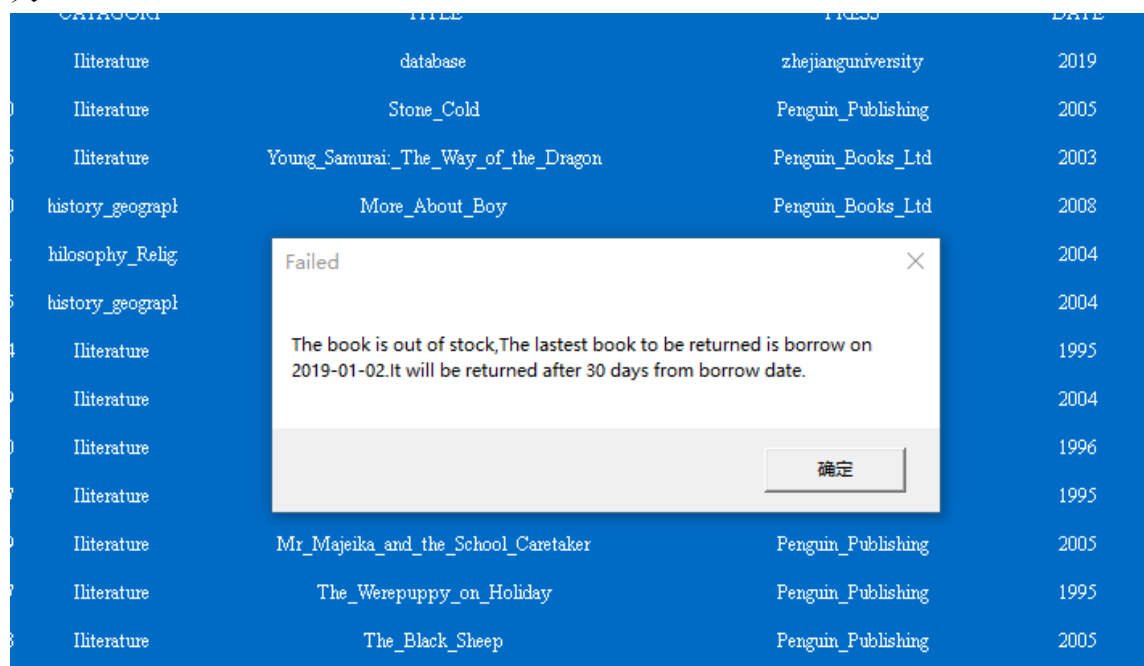


图 3-35 显示最近的归还时间

3、借书成功——该书在图书馆中且有库存

提示借书成功，然后刷新该用户已借书籍列表。

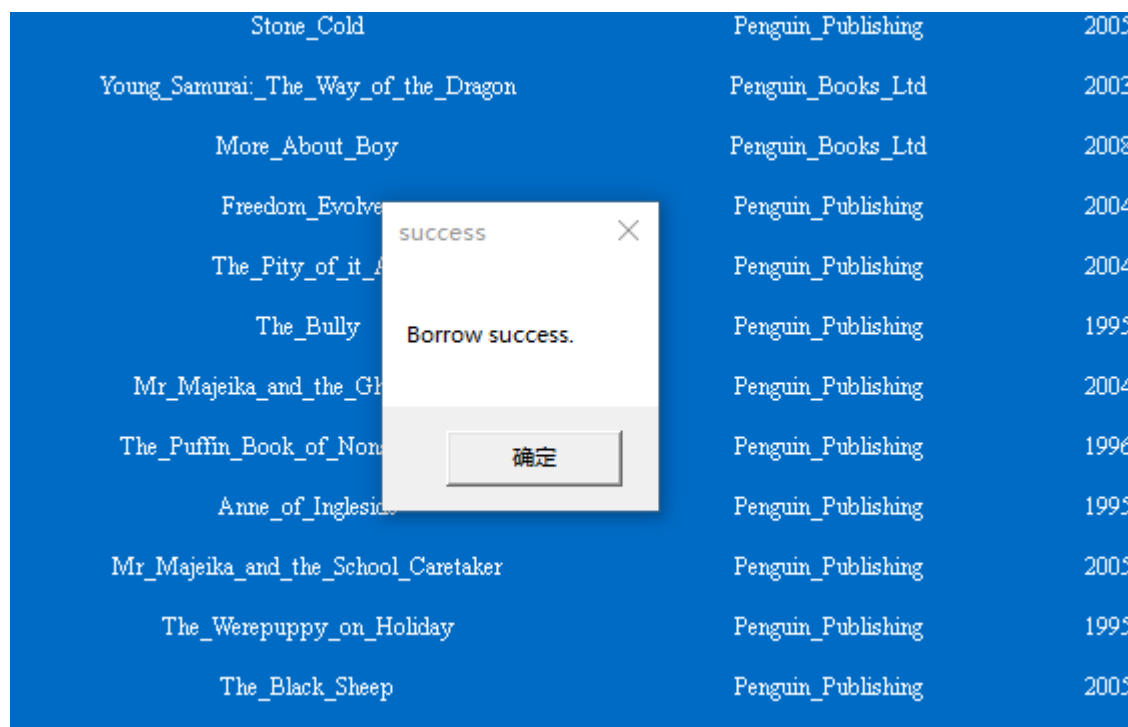


图 3-36 借书成功

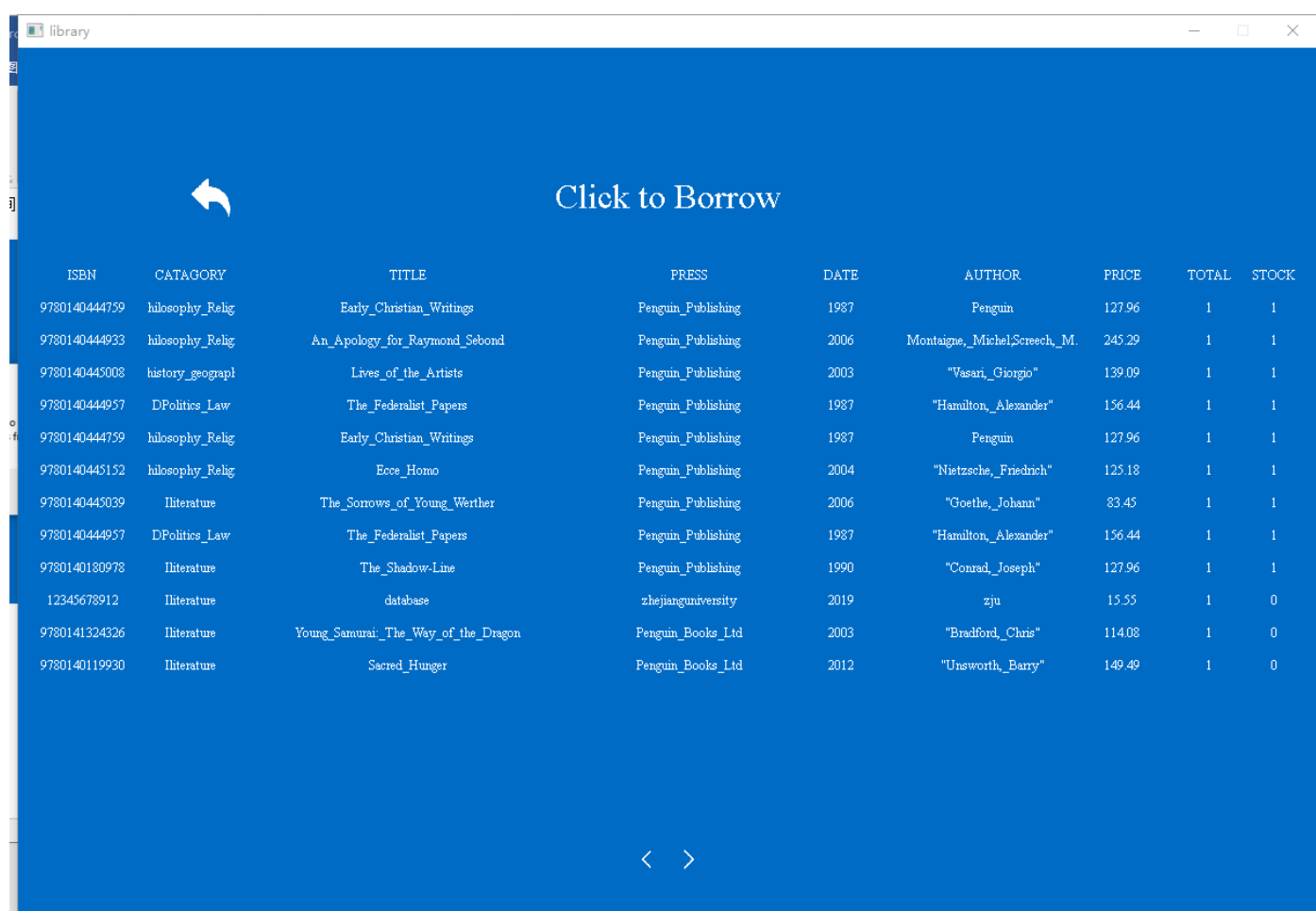


图 3-37 刷新已借书籍列表，新借的书籍被添加在列表最后

*此处不允许同一用户借同样 ISBN 码的书籍

3.5.4 返回上一级

单击左上角返回按钮，回到图书管理界面。

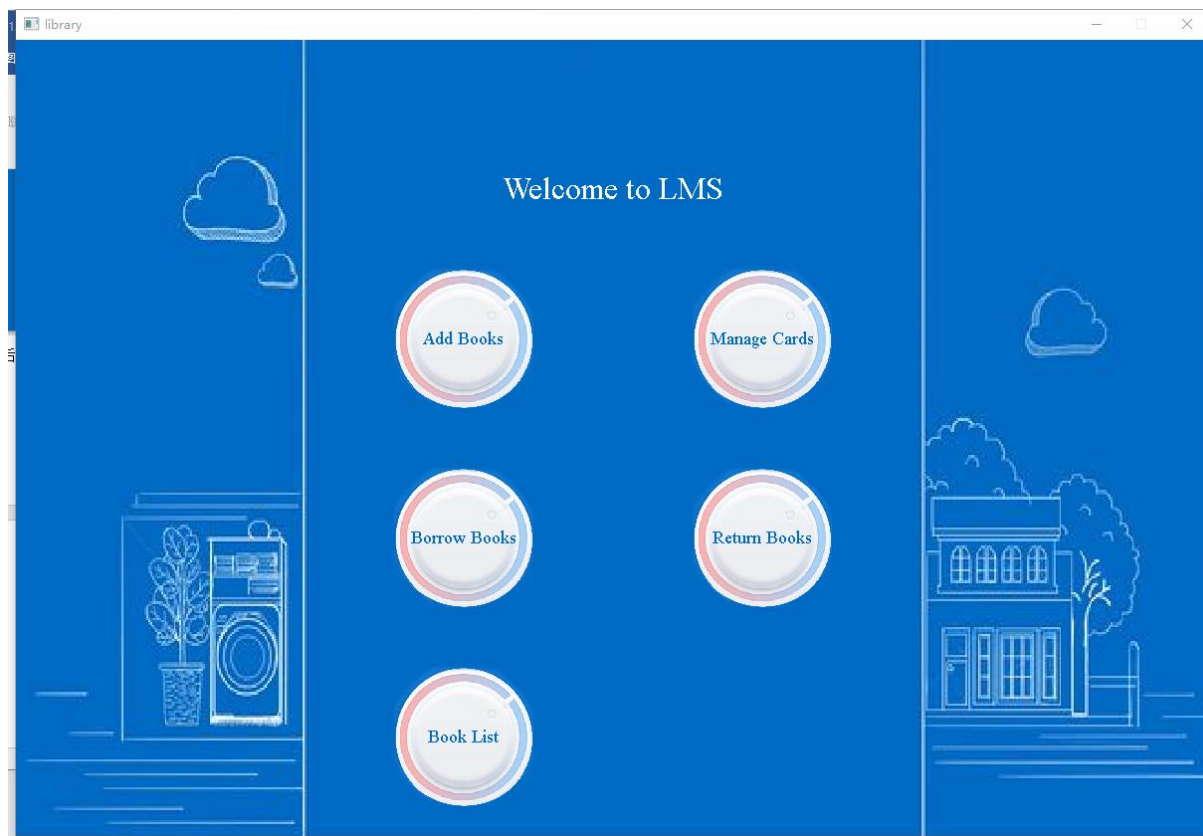


图 3-38 返回

3.6 还书功能

3.6.1 输入学号，显示该用户已借的全部书籍

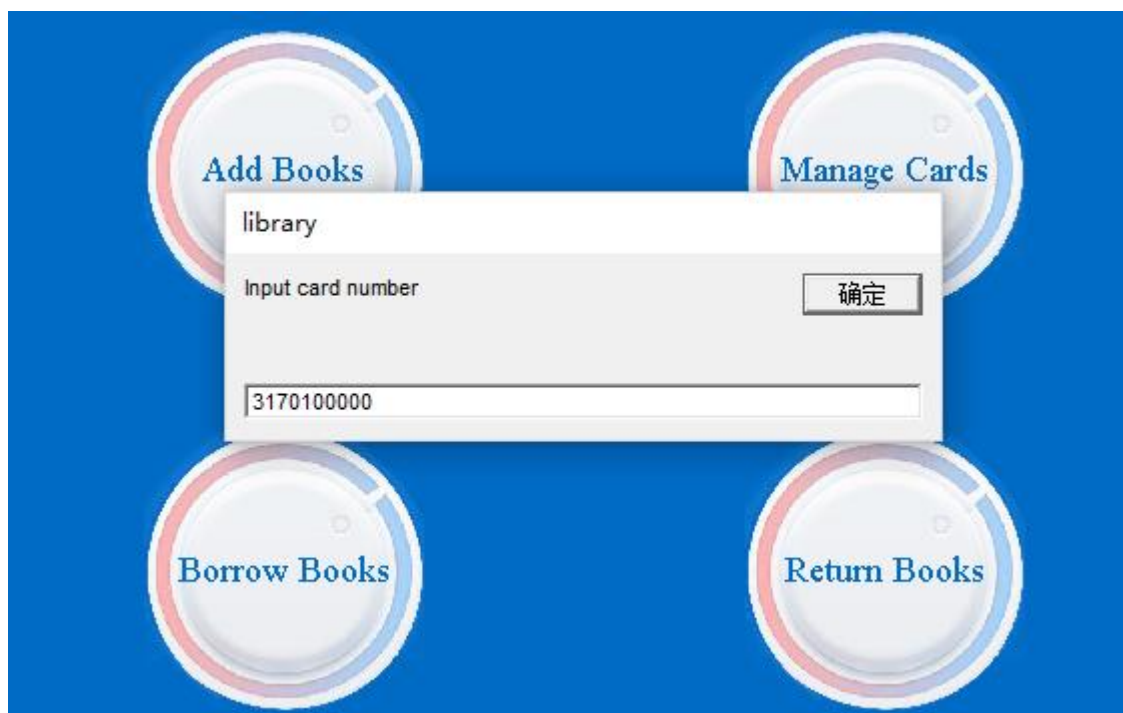


图 3-39 显示已借书籍

3.6.2 输入 ISBN 码

若图书馆中无该书，或者用户未还记录中无此书，显示提示。若用户借了此书尚未归还，则可进行后续操作。

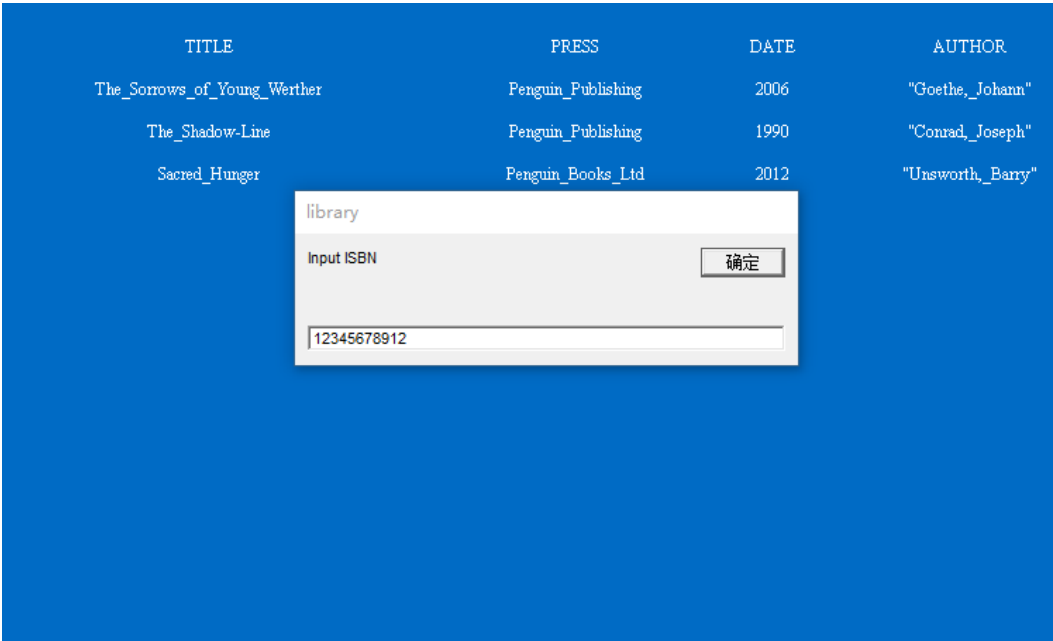


图 3-40 输入用户未借用的书籍

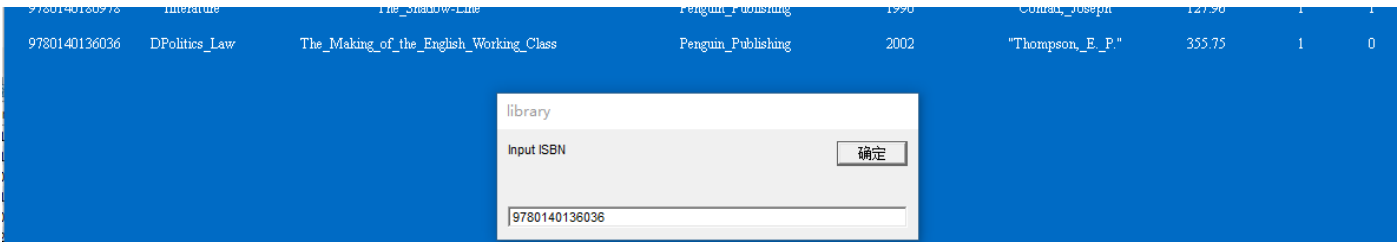


图 3-41 输入列表中最后一个 ISBN 码，确认是用户借阅中的书籍

3.6.2 成功还书流程

确认用户对此书的状态为借阅中后，首先将借阅记录 record 转移到 oldrecord 表中，并增加上还书日期的信息。执行增加记录后进行成功提示。随后将 record 中旧的记录删除，成功后进行提示。最后将 book 表中该书的库存加一，若成功，则整个还书过程成功。最后再刷新一遍已借书籍，以便管理员核对。

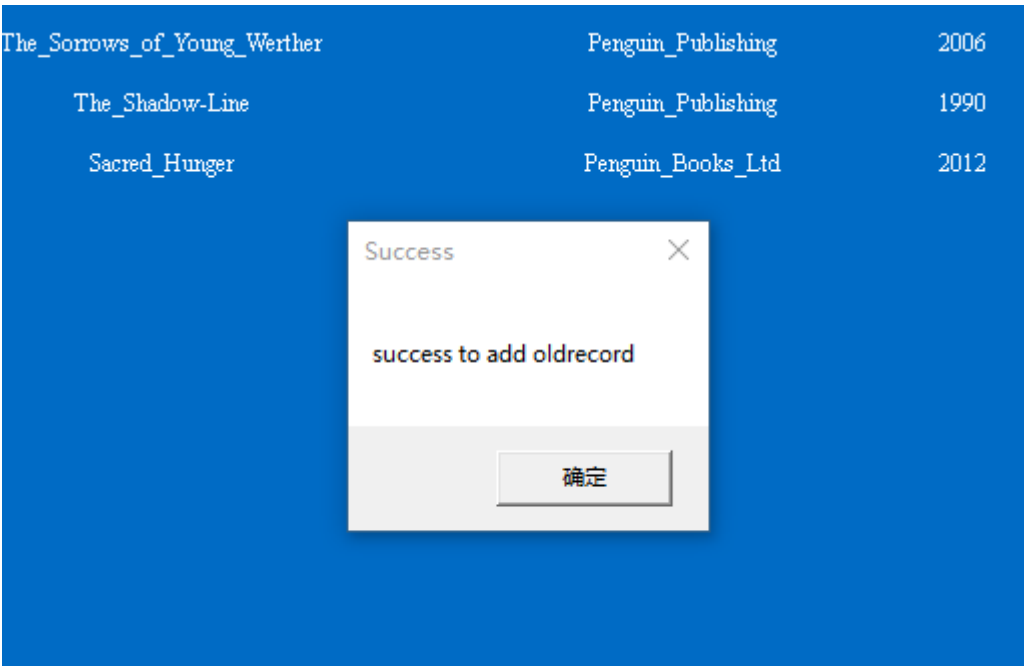


图 3-42 在记录表中加入借书记录

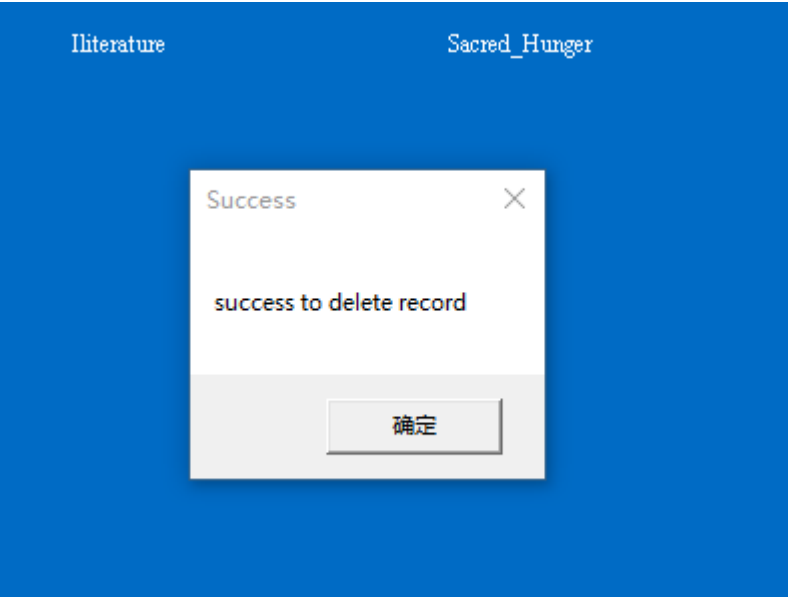


图 3-43 在 record 表中删除记录

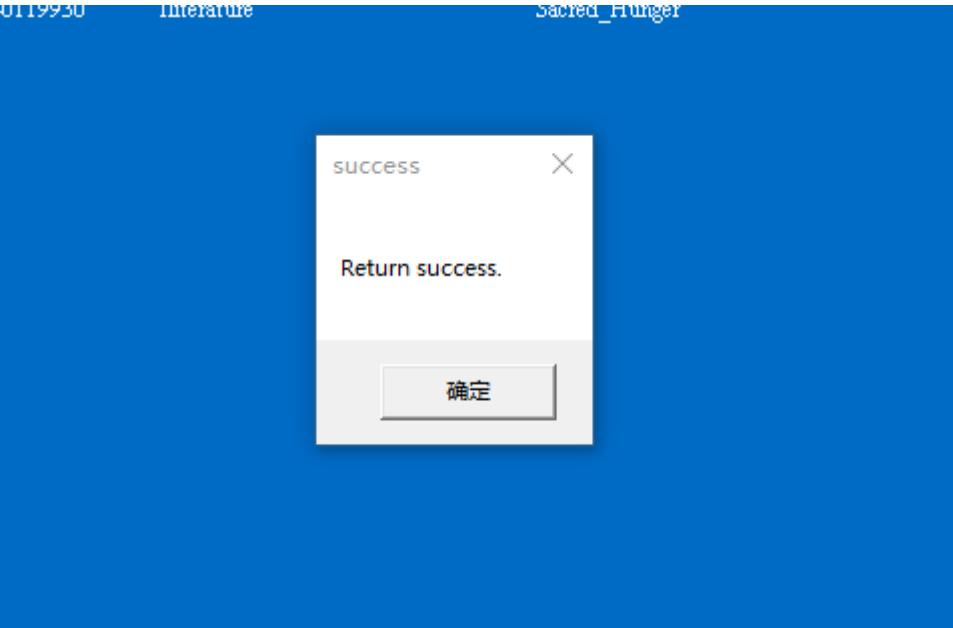


图 3-44 还书成功提示

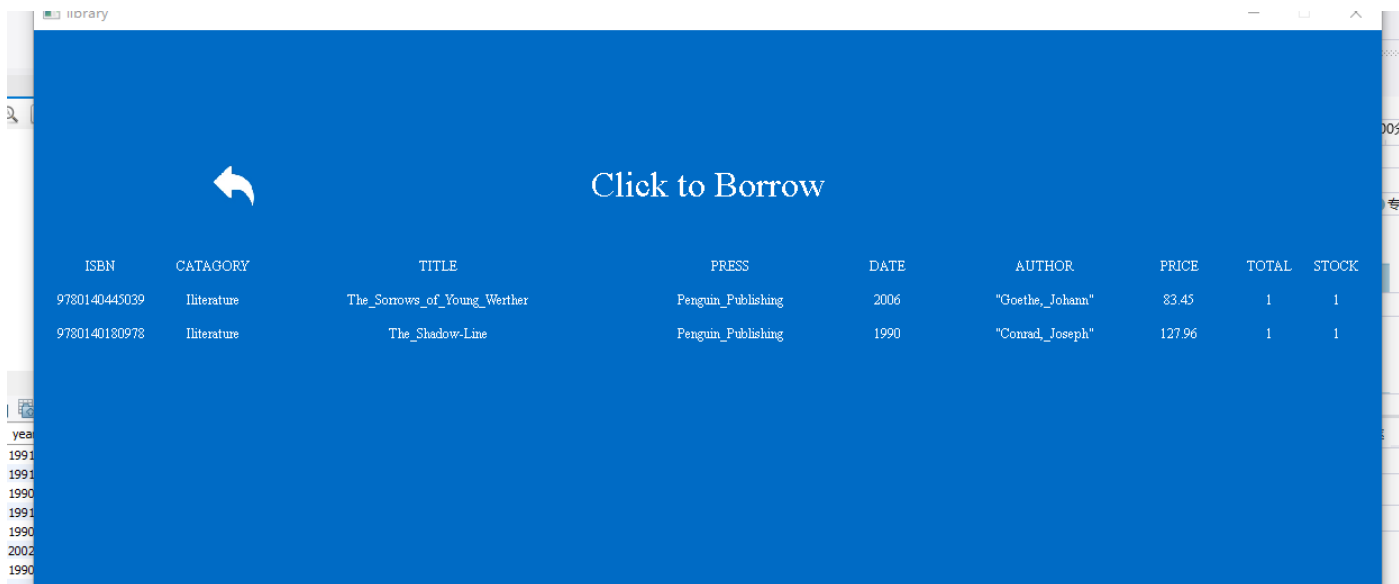


图 3-45 刷新书库，记录比原先减少一行

3.6.3 后台校验数据

在 mysql 中检查数据。还书之前，ISBN 码为 978014119930，库存为 0，还书后库存增加为 1。

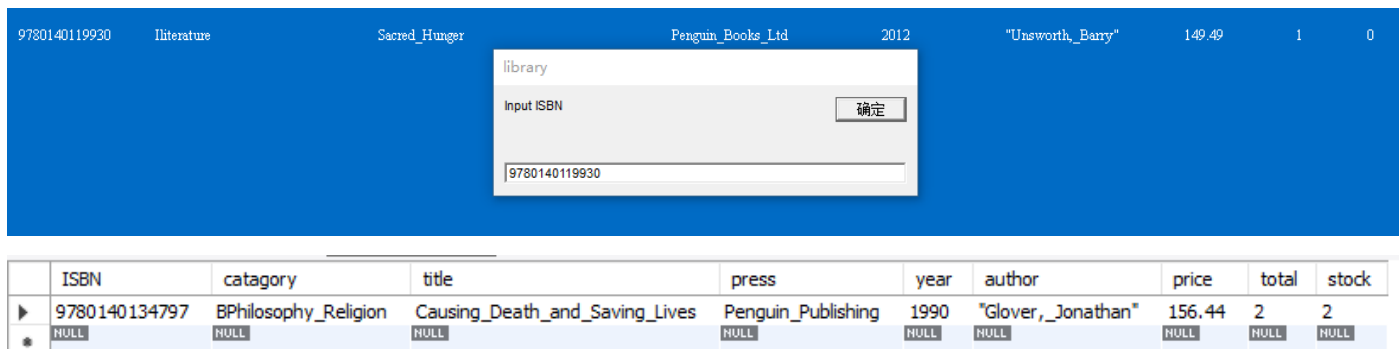


图 3-46 数据库中数据校验

3.7 查看书单功能

点击“book list”，查看目前书库中的所有书籍，可指定排序方式，可在查询结果过界面切换排序方式。

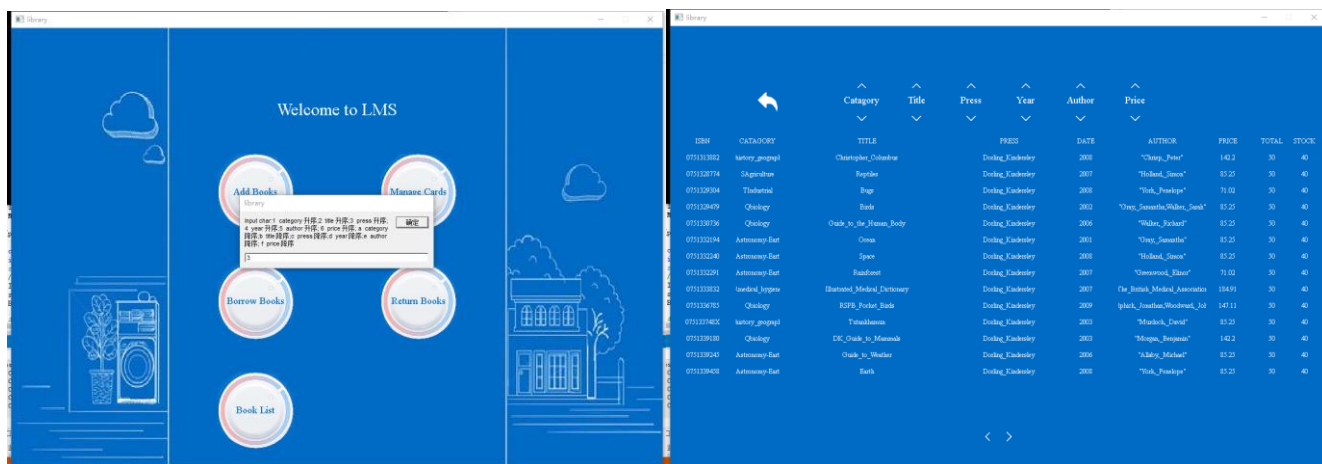


图 3-47 输入排序方式

图 3-48 查询结果界面

查询结果界面与输入属性进行查询的界面一致，包括翻页，按制定属性重新排序以及返回功能。

4 系统构架描述

4.1 系统功能构架

首页

图书查询

- 输入查询属性
- 输入相应信息
- 选择排序方式
- 显示查询结果
- 查看翻页结果
- 点击属性重新排序
- 返回上一级

管理员登陆

- 输入账号
- 输入密码
- 核对账号密码
 - 登录成功
 - 登录失败

图书导入

- 单本导入
 - 输入相应信息
 - 核对
 - 导入book表中
- 文件导入
 - 输入文件路径
 - 导入book表中

图书管理

管理借书卡

- 增加借书卡
 - 已在库中, 不可加入
 - 不在库中, 可加入
- 删除借书卡
 - 有书未还, 不可删
 - 还清书籍, 可删

借书

- 输入学号, 显示已借书籍
- 输入书号, 判断库存和已借状态
- 输出借阅状态, 刷新借书列表

还书

- 输入学号, 显示已借书籍
- 输入书号, 判断还书状态, 刷新库存
- 输出还书状态, 刷新借书列表

查看书库列表

- 查看书库中全部书目

图 4-1 功能架构

4.2 系统实现构架

4.2.1 函数清单及简介

```
1. void ManagerLogin(MYSQL * conn_ptr, char* ID, char* password); //管理员登陆，判断输入的账号与密码是否对应
2. //界面显示相关函数
3. void homepage(); //绘制图书馆里系统首页
4. void managerpage(); //绘制图书管理界面
5. void SortingAttributes(); //绘制根据属性排序的图书列表界面
6. void borrowpage(); //绘制显示读者已借书籍列表界面，有借书按钮
7. void returnpage(); //绘制显示读者已借书籍列表界面，有还书按钮
8. //书库管理相关函数
9. void BookEntry(int option, MYSQL *conn_ptr); //图书入库函数
10. void Query(MYSQL* conn_ptr, char* choice, char* info, char* order, int current_page); //向数据库传输特定查询语句的函数
11. void queryborrow(MYSQL* conn_ptr, char * cno, int current_page); // 查询当前用户借阅的书籍，并全部列出
12. void BookBorrow(MYSQL *conn_ptr); //借书函数
13. //按照格式输出查询结果
14. void output(string query, MYSQL* conn_ptr, int current_page); //在屏幕上显示查询结果
15. //与借书卡管理相关的函数
16. void addcard(MYSQL* conn_ptr, char* cno, char* name, char* depart, char* type); //增加借书卡的函数
17. void deletecard(MYSQL* conn_ptr, char* cno); //删除借书卡函数
18. //借书还书相关函数
19. void BookBorrow(MYSQL* conn_ptr, char* ISBN, char* cno, char* borrow_date, char* ID); //还书函数
20. void BookReturn(MYSQL *conn_ptr, char* cno, char* ISBN); //借书函数
```

4.2.2 函数调用关系

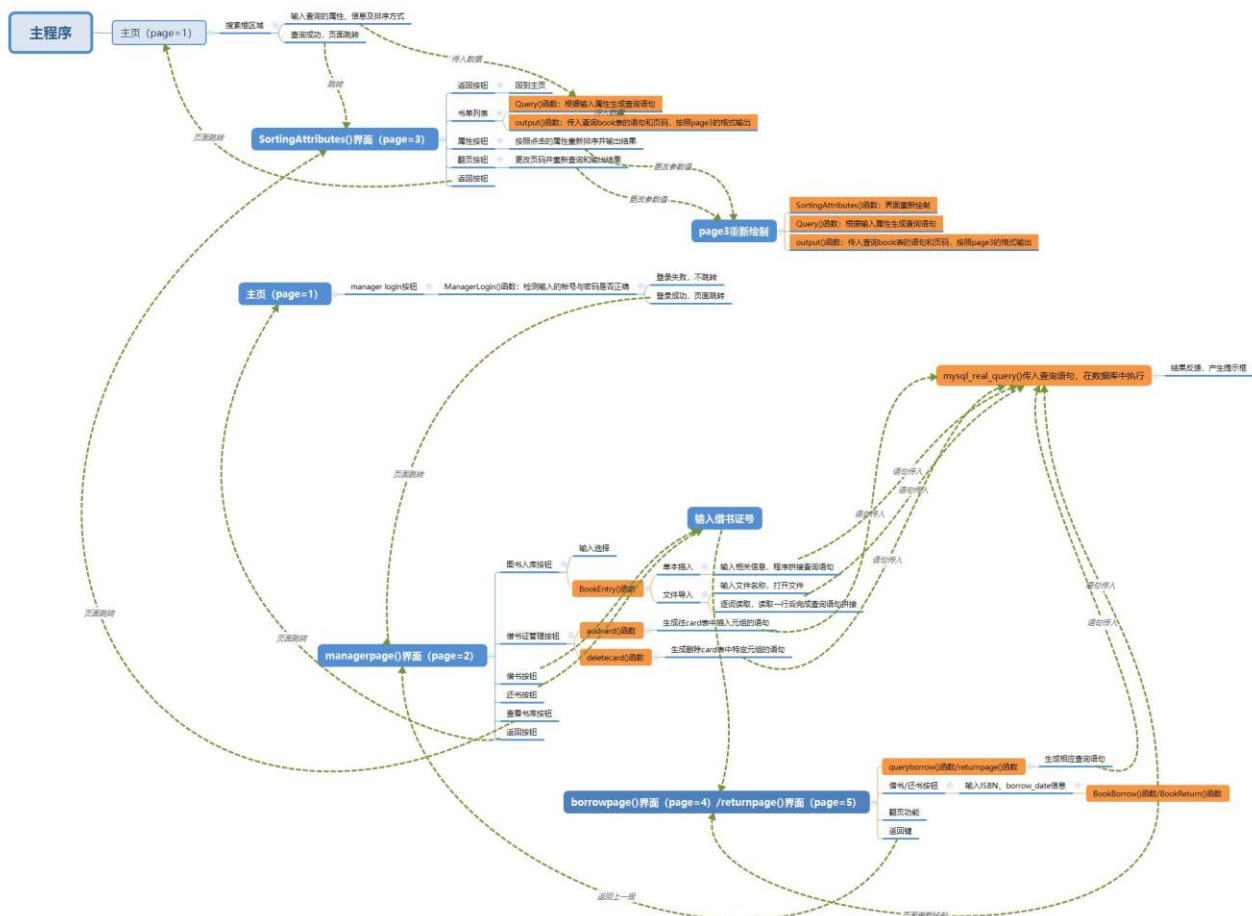
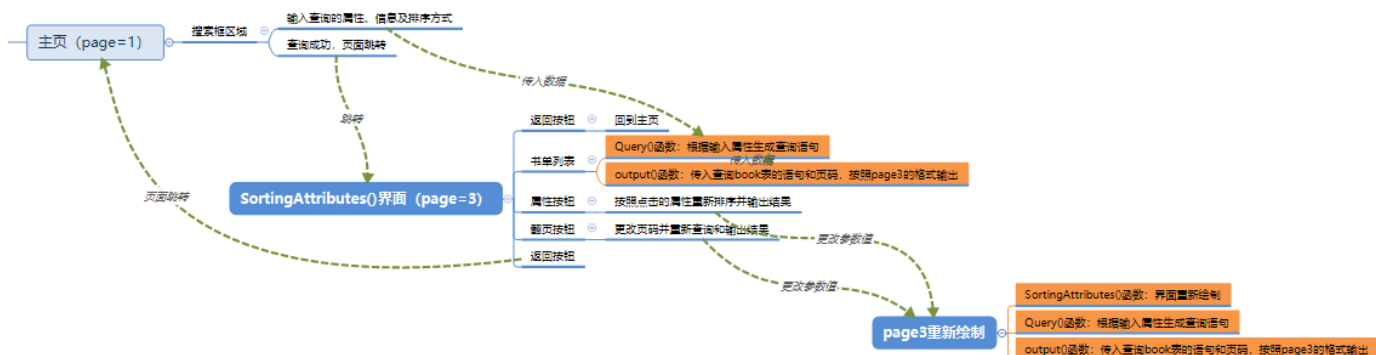


图 4-2 程序内部调用关系（点击查看图片）

5 各模块流程图与实现技术

5.1 查询模块

5.1.1 流程图



5.1.2 实现技术

1、主页的按钮检测，点击查询功能

```

1. if (page==1&& m.x > 400 && m.x < 800 && m.y>350 && m.y < 450)//点击查询按钮
2.     {
3.         InputBox(choice, 2, "Input attribute to query:1 catagory;2 title;3 press;4 year;5
author;6 price",NULL,"1");
4.         if (choice[0] == '1' || choice[0] == '2' || choice[0] == '3' || choice[0] == '4' |
| choice[0] == '5' || choice[0] == '6')
5.             {
6.                 respage = 1;
7.                 //是否需要防止 info 和 order 未初始化?
8.                 InputBox(info, MAX, "Input the infomation",NULL,"literature");
9.                 InputBox(order, MAX, "Input char:1 category 升序;2 title 升序;3 press 升
序;4 year 升序;5 author 升序; 6 price 升序; a category 降序;b title 降序;c press 降
序;d year 降序;e author 降序; f price 降序", NULL, "3");
10.                SortingAttributes();
11.                Query(conn_ptr,choice, info,order,respage);
12.            }
13.        else
14.            MessageBox(hw, "Error input", "Error", MB_OK);
15.    }

```

2、页面显示函数 SortingAttributes()

(代码省略)

3、Query(conn_ptr,choice, info,order,respage)

根据输入的属性生成查询语句，使用 `mysql_real_query()` 查询，根据查询结果进行不同的处理。查询成功时，在相应区域输出各个属性信息。

```

1. void Query(MYSQL* conn_ptr, char* choice, char* info, char * order, int current_page)/*查询图书的函
数*/
2. {
3.     Prodeuce diffrenet query words->query;
4.     output(query, conn_ptr, current_page);
5. }

```

按照指定方式在窗口中打印查询到的元组

```

1. void output(string query,MYSQL* conn_ptr,int current_page)
2. {
3.     char query_c[MAX];
4.     s2c(query, query_c);
5.     cout << query_c;
6.     if (!QueryS(query_c))//如果查询成功，进行后续操作
7.     {
8.         MYSQL_RES *res;
9.         MYSQL_ROW row;
10.        unsigned int i, num_fields;
11.        res = mysql_store_result(conn_ptr);
12.        num_fields = mysql_num_fields(res);
13.        cout << endl << num_fields << endl;

```

```

14.     int count = 0;
15.     my_ulonglong num_rows;
16.     num_rows = mysql_num_rows(res);
17.     cout << "num_rows" << num_rows << endl;
18.
19.     int current_rows = 1;
20.     total_page = (int)ceil((double)num_rows / 16.0);
21.     cout << total_page;
22.     while (row = mysql_fetch_row(res))//把每一行查询结果取出
23.     {
24.         current_rows++;
25.         if (current_rows > (current_page - 1) * 16 && current_rows < current_page * 16)
26.         {
27.             RECT r;
28.             settextrcolor(WHITE);
29.
30.             count++;
31.             for (i = 0;i < num_fields;i++)
32.             {
33.                 switch (i)
34.                 {
35.                     case 0:
36.                         r = { 20,200 + ROWSPACE * count,100,220 + ROWSPACE * count };
37.                         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
38.                         break;
39.                     case 1:
40.                         r = { 120,200 + ROWSPACE * count ,200,220 + ROWSPACE * count };
41.                         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
42.                         break;
43.                     case 2:
44.                         r = { 220,200 + ROWSPACE * count,500,220 + ROWSPACE * count };
45.                         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
46.                         break;
47.                     case 3:
48.                         r = { 540,200 + ROWSPACE * count,700,220 + ROWSPACE * count };
49.                         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
50.                         break;
51.                     case 4:
52.                         r = { 720,200 + ROWSPACE * count,800,220 + ROWSPACE * count };
53.                         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
54.                         break;
55.                     case 5:
56.                         r = { 820,200 + ROWSPACE * count,980,220 + ROWSPACE * count };

```

```

57.         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
58.         break;
59.     case 6:
60.         r = { 980,200 + ROWSPACE * count,1060,220 + ROWSPACE * count };
61.         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
62.         break;
63.     case 7:
64.         r = { 1080,200 + ROWSPACE * count,1120,220 + ROWSPACE * count };
65.         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
66.         break;
67.     case 8:
68.         r = { 1140,200 + ROWSPACE * count,1180,220 + ROWSPACE * count };
69.         drawtext(row[i] ? row[i] : "null", &r, DT_CENTER | DT_VCENTER | DT_SINGLEL
INE);
70.         break;
71.     }
72. }
73. }
74. }
75. }
76. else
77. {
78.     MessageBox(hw, "Query Error", "Error", MB_OK);
79. }
80. }

```

5.2 管理员登陆

5.2.1 流程图

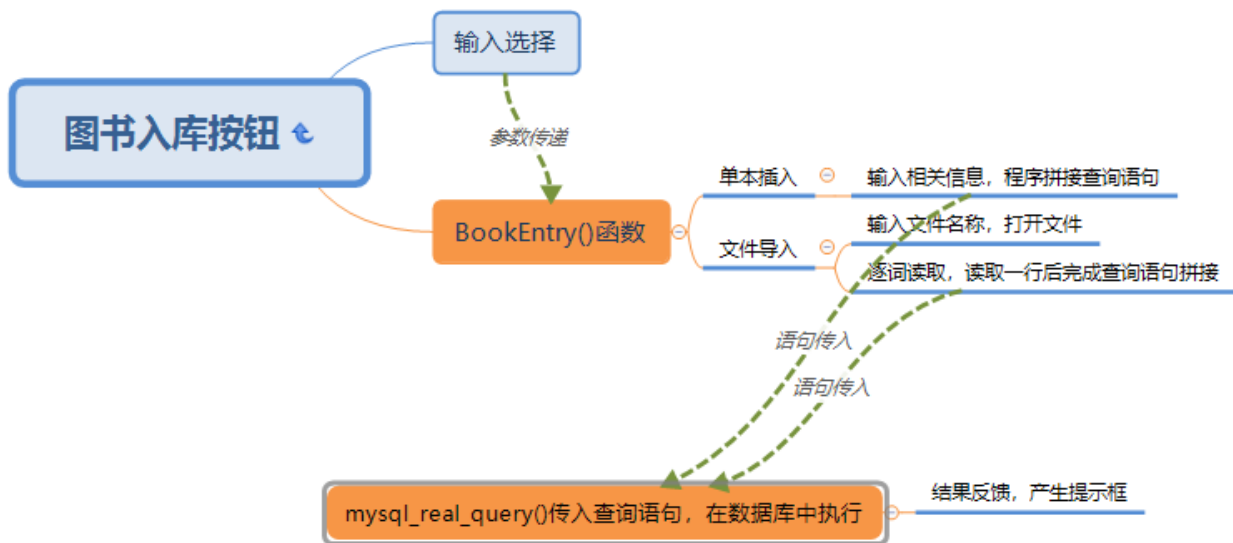


5.2.2 实现技术

```
1. void ManagerLogin(MYSQL * conn_ptr, char* ID, char* password)//管理员登陆，判断输入的账号与密码是否对
   应
2. {
3.     input->ID,password;
4.     res= mysql_store_result(conn_ptr);
5.     mysql_free_result(res);
6.
7.     if (mysql_query(conn_ptr, (const char*)MI))
8.     {
9.         cout << "error input";
10.    }
11.    else //success
12.    {
13.        res = mysql_store_result(conn_ptr);
14.        count = mysql_num_rows(res);
15.        cout << "count:" << count << endl;;
16.        if (count)
17.        {
18.            //log in successfully
19.            MessageBox(hw, "Log in successfully", "Log in", MB_OK);
20.            managerpage();
21.        }
22.        else
23.        {
24.            MessageBox(hw, "Error in account or password", "Error", MB_OK);
25.        }
26.    }
27. }
```


5.3 图书入库

5.3.1 流程图

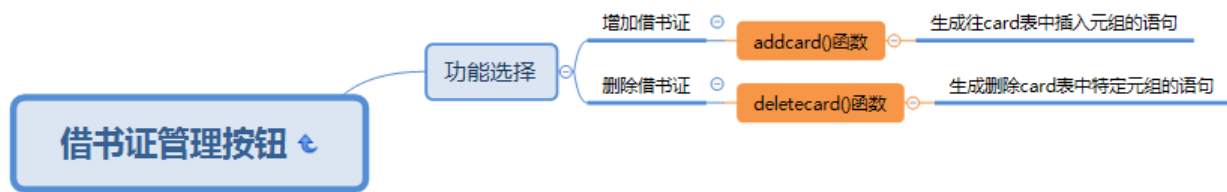


5.3.2 实现细节

```
1. void BookEntry(int option, MYSQL *conn_ptr)//导入图书
2. {
3.     if(option==1)//选择单本导入
4.     {
5.         input the information needed;
6.         check the data input;
7.         if (correct)
8.         {
9.             produce the query sentence sm;
10.            QueryS(sm);
11.        }
12.    }
13.    if (option == 2)//选择从文件中导入
14.    {
15.        read the file with the path input
16.        //open file success
17.        while (!fin.eof())//not reach the end of file
18.        {
19.            read the information;
20.            produce the query sentence sm;
21.            QueryS(sm);
22.        }
23.    }
24. }
```

5.4 借书卡管理

5.4.1 流程图

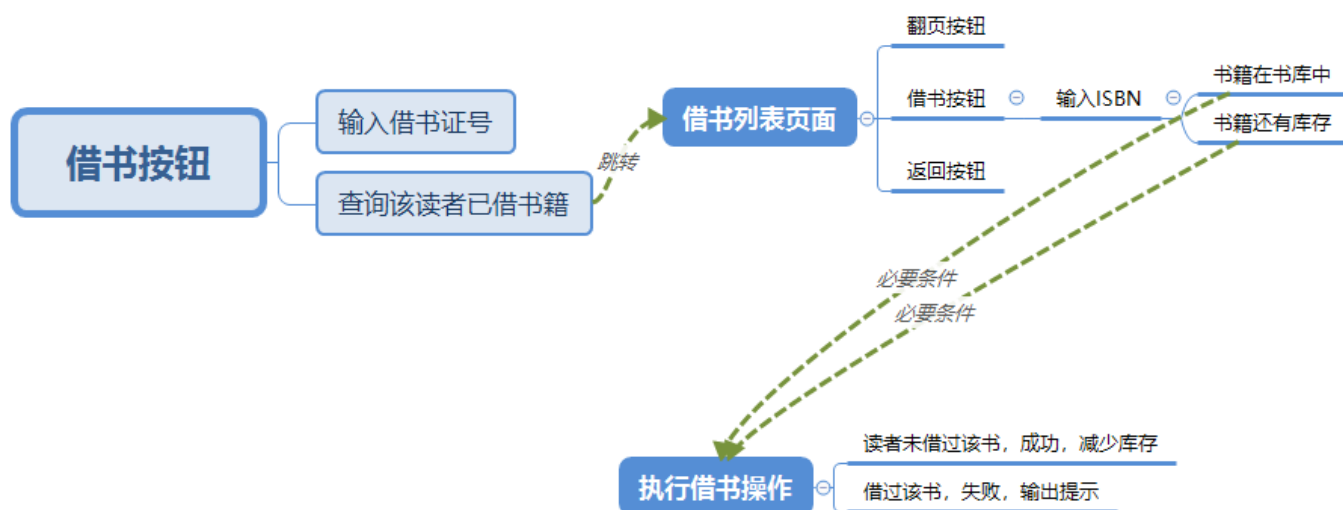


5.4.2 实现细节

```
1. void addcard(MYSQL* conn_ptr, char* cno, char* name, char* depart, char* type)
2. {
3.     //检测卡是否已经存在, 不允许相同的卡号存在
4.
5.     produce the insert sentence Add with the data given;
6.     if (!QueryS(Add))
7.     {
8.         MessageBox(hw, "The card added", "Success", MB_OK);
9.     }
10.    else//默认将出错当成卡片已存在
11.    {
12.        MessageBox(hw, "The card already exist", "Error", MB_OK);
13.    }
14. }
15.
16. void deletecard(MYSQL* conn_ptr, char* cno)
17. {
18.     produce the delete sentence Del with the data given;
19.     if (!QueryS(Del))
20.     {
21.         MessageBox(hw, "Delete card", "success", MB_OK);
22.     }
23.     else//这张卡仍有书未归还
24.     {
25.         MessageBox(hw, "Fail to delete the card.Check if the card has some books unreturned.", "Failed", MB_OK);
26.     }
27. }
```

5.5 借书

5.5.1 流程图



5.5.2 实现技术

点击借书按钮，输入学号，页面跳转，显示读者借阅的全部书籍。

```
1.  if (page == 2 && m.x > 380 && m.x < 520 && m.y>430 && m.y < 570)//借书功能
2.      {
3.          //输入借书证号，显示该用户已经借的书籍
4.          cout << "all borrow!" << endl;
5.          InputBox(cno, MAX, "Input card number", NULL, "3170100000");
6.          borpage = 1;
7.          borrowpage();
8.          queryborrow(conn_ptr, cno, borpage);
9.      }
```

在借书页面中实现借书功能。

```
1.  if (page == 4 && m.x > 480 && m.x < 720 && m.y>120 && m.y < 156)//点击 click to borrow
2.      {
3.          char ISBN[MAX];
4.          char borrow_date[MAX];
5.          char check[2];
6.          InputBox(ISBN, MAX, "Input ISBN", NULL, "9780141324432");
7.          InputBox(borrow_date, MAX, "Input date", NULL, "2019-4-26");
8.          InputBox(check, 2, "1 check,2 cancel");
9.          if (check[0] == '1')//确认输入的借书信息是正确的
10.         {
11.             BookBorrow(conn_ptr, ISBN, cno, borrow_date, ID);
12.             borpage = 1;
13.             borrowpage();
14.             queryborrow(conn_ptr, cno, borpage);
15.         }
16.     }
```

借书功能具体实现。

```
1. void BookBorrow(MYSQL* conn_ptr, char* ISBN, char* cno, char* borrow_date, char* ID)
2. {
3.     produce the query sentence left the check the book's stock;
4.     if (!QueryS(Left))//查询语句正确执行
5.     {
6.         //把结果取出来看看 s 是否大于 1
7.         //计算出整型 Stock 的数值
8.         /*使用复杂的检测以避免出错*/
9.         row = mysql_fetch_row(res);
10.        if (row)//如果查询得到元组, 说明该书是存在的
11.        {
12.            stock = row[0];
13.            (string)stock->(int) Stock;
14.            if(Stock>=1)//如果还有库存, 则允许借出
15.                INSTOCK = 1;
16.            else//该书没有库存
17.            {
18.                //查询一下最久远的借书日期, 并把日期输出
19.                produce the query sentence Date to select record of the book;
20.                if (!QueryS(Date))
21.                {
22.                    MYSQL_RES* res;
23.                    MYSQL_ROW row;
24.                    res = mysql_store_result(conn_ptr);
25.                    /*如果该书没有库存, 输出该书最久远的一次借书时间, 以显示其将会被归还的最近的时间*/
26.                    if (row = mysql_fetch_row(res))//查到了 record 里面关于该书的借出记录
27.                    {
28.                        string neardate="The book is out of stock,The lastest book to be returned
is borrow on ";
29.                        neardate += row[0];
30.                        neardate += ".It will be returned after 30 days from borrow date.";
31.
32.                        MessageBox(hw, neardate.c_str(), "Failed", MB_OK);
33.                    }
34.                }
35.            else
36.                MessageBox(hw, "Query Error", "Error", MB_OK);
37.        }
38.    }
39.    else//否则说明该书不存在
40.        MessageBox(hw, "The book is not in the library or the user has already borrow it.", "Failed", MB_OK);
41.    }
42.    else
43.    {
44.        //提示输入的图书编码有误, 返回上一级
45.        MessageBox(hw, "Duplicated borrow.", "Error", MB_OK);
```

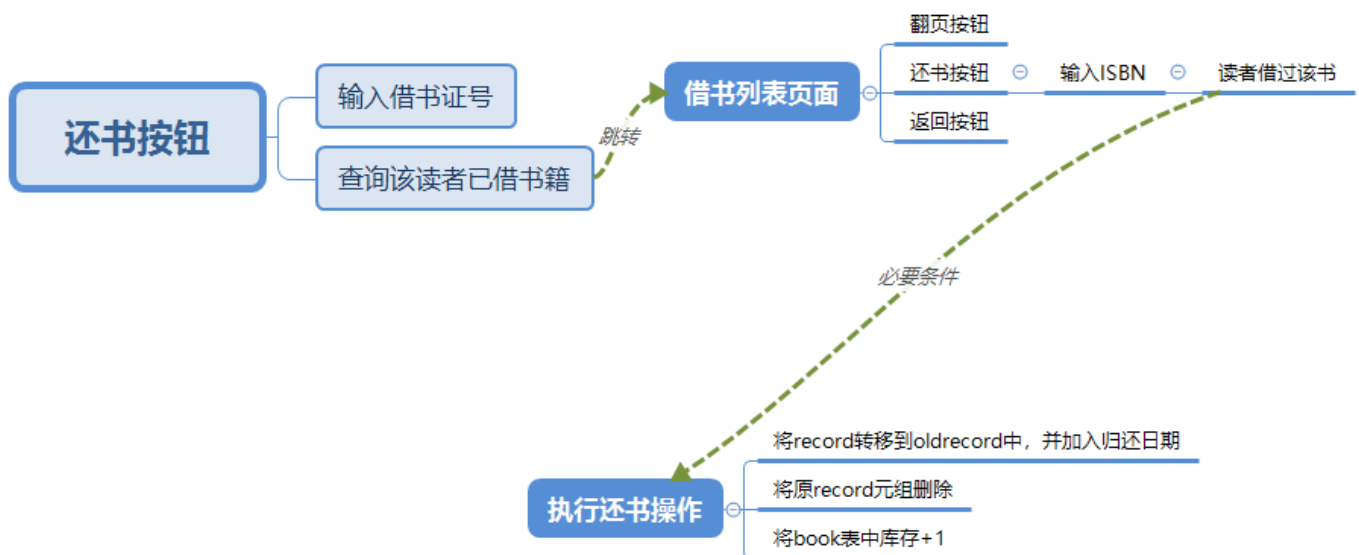
```

46.     return;
47. }
48.
49. if (INSTOCK)//有库存才执行下列操作
50. {
51.     //添加到借书记录中
52.     produce the query sentence Q to insert data to table record;
53.     produce the query sentence Reduce to insert data to table book to update stock;
54.     if (!QueryS(Q))
55.     {
56.         if(!QueryS(Reduce))
57.             MessageBox(hw, "Borrow success.", "success", MB_OK);
58.         else
59.             MessageBox(hw, "Failed to update stock.", "failed", MB_OK);
60.     }
61.     else
62.     {
63.         cout << "insert 语句执行失败! " << endl;
64.     }
65. }
66. }

```

5.6 还书

5.6.1 流程图



5.6.2 实现技术

还书的具体实现。

```

1. void BookReturn(MYSQL *conn_ptr, char* cno, char* ISBN)

```

```

2. {
3.     //确定输入的信息是否有误,是否存在这样的元组
4.     produce the query sentence Ret to check if there's record in the table record;
5.     if (!QueryS(Ret))//查询成功, null 也是成功
6.     {
7.         if (row = mysql_fetch_row(res))//row 非空说明有借书记录
8.         {
9.             InputBox(return_date, MAX, "Input the return date", NULL, "2019/4/27");
10.            for (i = 0;i < num_fields;i++)
11.            {
12.                switch (i)
13.                {
14.                    case 2:
15.                        borrow_date = row[i] ? row[i] : "null";
16.                        break;
17.                    case 4:
18.                        manager_ID = row[i] ? row[i] : "null";
19.                }
20.            }
21.            //开始拼接查询语句
22.            produce the query sentence Addold the insert data into table oldrecord;
23.            if (!QueryS(Addold))//成功插入了记录, 可以开始执行原纪录的删除
24.            {
25.                produce the query sentence Del to delete data from table record;
26.                if (!QueryS(Del))//成功删除,修改图书数据
27.                {
28.                    produce the query sentence Left to see the stock in table book;
29.                    if (!QueryS(Left))//查询语句正确执行
30.                    {
31.                        query the current stock of the book;
32.                        stock--;
33.                        produce the query sentence Increase to update the stock in table book;
34.                        if (!QueryS(Increase))
35.                        {
36.                            MessageBox(hw, "Return success.", "success", MB_OK);
37.                            //还书成功之后,刷新已借列表
38.                        }
39.                        else
40.                            MessageBox(hw, "Failed to update stock.", "failed", MB_OK);
41.                    }
42.                }
43.            }
44.            else
45.            {
46.                MessageBox(hw, "Unknown mistake ,failed to delete record", "Error", MB_OK);
47.            }
48.        }
49.        else
50.        {
51.            MessageBox(hw, "Unknown mistake ,failed to add to old record", "Error", MB_OK);

```

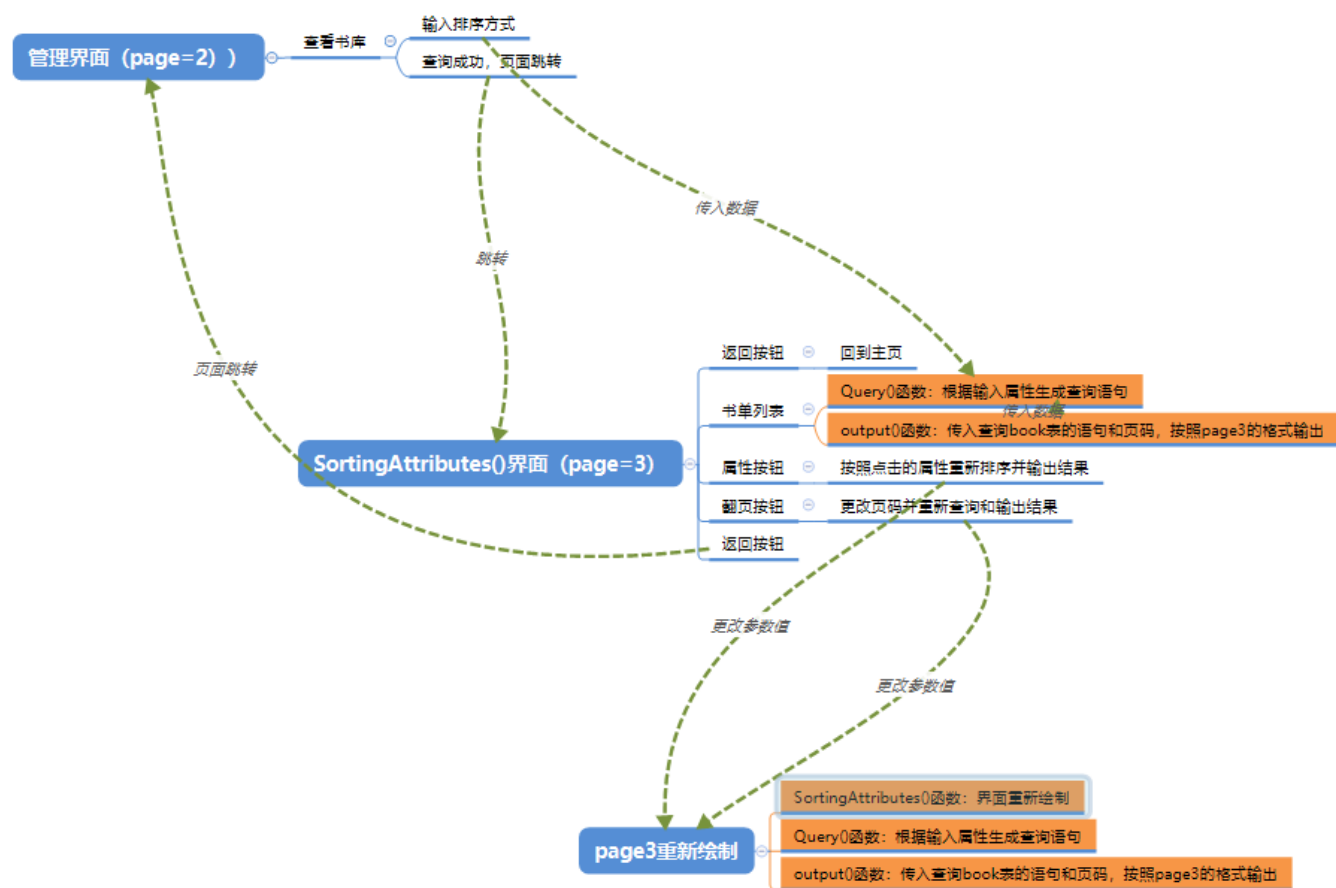
```

51.         }
52.
53.     }
54.     else//说明取回的元组是空的，查不到该借书记录
55.     {
56.         MessageBox(hw, "The user did not borrow the book", "Error", MB_OK);
57.     }
58. }
59. else
60. {
61.     MessageBox(hw, "Unknown error", "Error", MB_OK);
62. }

```

5.7 查看书库

5.7.1 流程图



5.7.2 实现技术

同查询功能，语句稍作变动，使用 SelectAll 作为识别判断。

```

1. if (page == 2 && m.x > 380 && m.x < 520 && m.y>630 && m.y < 770)//查看书库按钮
2. {
3.     cout << "all book!" << endl;

```

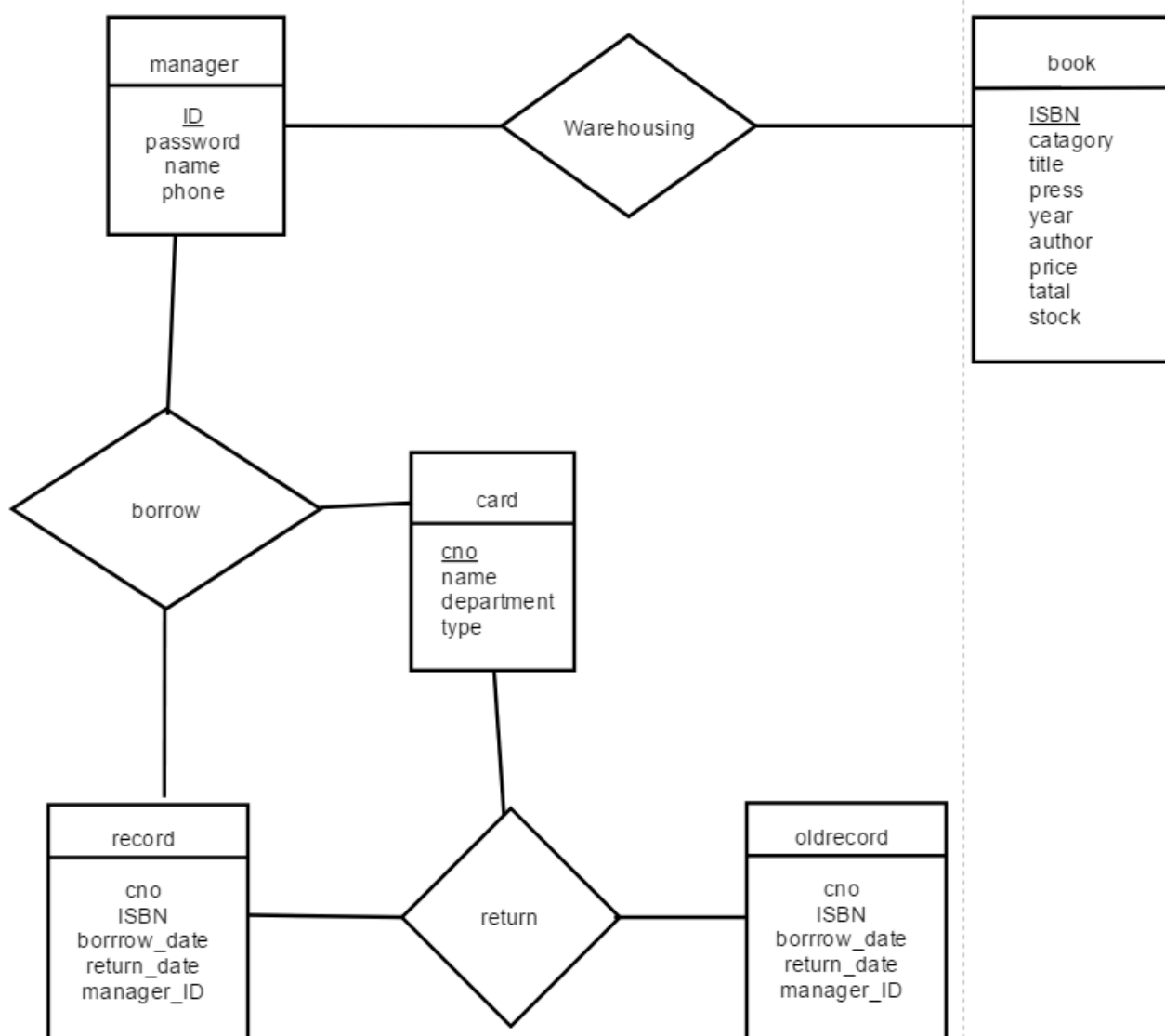
```

4.         SelectAll = 1;
5.         respage = 1;
6.         InputBox(order, MAX, "Input char:1 category 升序;2 title 升序;3 press 升
序;4 year 升序;5 author 升序; 6 price 升序; a category 降序;b title 降序;c press 降
序;d year 降序;e author 降序; f price 降序", NULL, "3");
7.         SortingAttributes();
8.         Query(conn_ptr, choice, info, order, respage);
9.     }

```

6 数据库表结构设计

数据库的设计在 mysql 中进行，使用 create table 等语句进行建表，主键约束和外键约束等。



7 心得讨论与改进空间

7.1 改进空间

7.1.1 图书入库功能

文件导入之后可以输出成功导入的元组数，以便管理员核对导入过程是否出错。关于核对导入情况的问题，为了实现的便捷性，我给程序加上了查看书库中全部书目的功能。但还是没有直接输出元组数来得方便，在时间条件允许的情况下我会进一步改进。

7.1.2 增加查看借书记录的按钮

为方便管理员对图书和借书卡进行管理，可以增加查看借书记录的按钮，包括未还的书籍列表和已还书籍列表，即将 `record` 和 `oldrecord` 表的内容均输出到屏幕上。

7.1.3 程序架构还有改进空间

程序中有部分函数赘余，可以进行适当的合并与删减。在有相关的项目经验之后，对程序架构有更深入的理解，可以做出更加规范合适的函数规划，使程序更为简明清晰。

7.2 心得体会

7.2.1 对 ODBC 接口的使用

在完成程序的过程中，我阅读了《Mysql 5.1 参考手册_C_API》一书，书中介绍了 ODBC 接口的函数定义与用法，其中的示例程序为我理解这些接口提供了很大的帮助，我使用的接口主要包括数据库连接的 `mysql_connect()`，查询语句传输的 `mysql_query()` 和 `mysql_real_query()`，以及查询结果取回的 `mysql_store_result()`，以及各个元组读取的 `mysql_fetch_row()` 等。

7.2.2 宏定义的使用

```
1. #define s2c(s,c) strncpy_s(c, s.c_str(), s.length() + 1)
2. #define QueryS(s) mysql_real_query(conn_ptr, s, (unsigned int)strlen(s))
```

在程序中我使用较多的宏定义为查询语句和 `string` 转换为 `char` 数组函数，使用宏定义可以使程序编写的时候更为便捷，也易于理解，这与将大段代码原样写入的执行效果是一致的，但是程序表达更为简洁易懂。