****

**《数据库原理》课程设计说明书**

图书馆后台管理系统

学 院： 计算机与信息科学学院

专 业： 软件工程

班 级： 软件1601

姓名学号： 魏如意 16020440115

姓名学号： 叶绮斌 16020440118

指导教师： 高金华 职称 讲师

完成时间： 2018年06月

1.需求分析

1.1 数据流图........................................................P2

1.2数据字典.........................................................P2-4

1.3加工说明.........................................................P4

2.数据库设计

2.1概念结构设计

2.1.1画出E-R图.............................................P5

2.2逻辑结构设计

2.2.1模式设计..................................................P6

2.2.2数据库表结构..........................................P7-8

3.应用程序设计

3.1系统设计

3.1.1 功能模块图.............................................P9

3.1.2模块功能说明..........................................P10-12

3.2系统实现

3.2.1界面设计..................................................P12-14

3.2.2程序流程图..............................................P14

3.2.3代码及其注释..........................................P15-36

4.设计小结........................................................P37

**1.需求分析**

**1.1 数据流图**

添加书籍

书籍信息

账户信息

修改密码

**管理员**

借还

**学生**

借阅记录

书籍信息

借还

**1.2数据字典**

**学生信息表(students)**

学生信息包括真实姓名，性别学号，电话号码，专业以及班级

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段描述 | 字段类型 |
| sname | 学生姓名 | varchar(2000) |
| sno | 学号 | varchar(2000) |
| sex | 性别 | varchar(100) |
| telnum | 电话号码 | varchar(50) |
| major | 专业 | varchar(2000) |
| class | 班级 | varchar(1000) |

**账户信息(admin)**

账户信息包括账户名，账户密码，以及账户角色

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段描述 | 字段类型 |
| account | 账户名 | varchar(200) |
| pwd | 密码 | varchar(200) |
| role | 角色 | varchar(200) |
| sno | 学号/工号 | varchar(200) |

**书籍信息(books)**

书籍信息包括书名，书的作者，住的编号，书的价格以及拥有的数量

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段描述 | 字段类型 |
| bname | 书籍名 | varchar(2000) |
| author | 书籍作者 | varchar(2000) |
| bnum | 书籍编号 | varchar(2000) |
| bprices | 书记价格 | varchar(2000) |
| had | 拥有数量 | varchar(2000) |

**借阅记录(logs)**

借阅记录包括借阅书籍的书名，借阅者的学号，借阅时间，借阅时长，借阅状态

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段描述 | 字段类型 |
| bname | 借阅的书籍名 | varchar(200) |
| bsno | 借阅者的学号/工号 | varchar(200) |
| btime | 借阅时间 | date(0) |
| howlong | 借阅时长 | varchar(200) |
| outline | 节借阅状态 | varchar(200) |

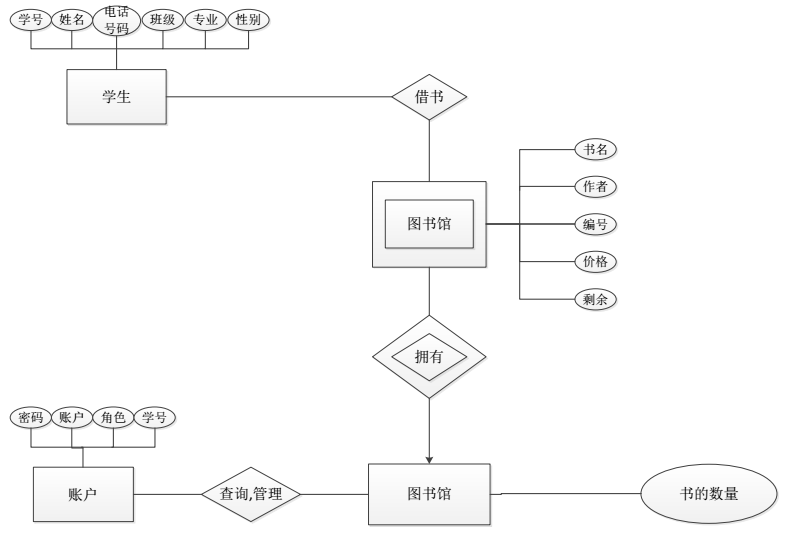
**1.3加工说明**

本数据结构操作系统使用Python3编写，使用时需安装python3环境，以及所需要使用到的库，库的安装使用python的pip工具使用命令行（cmd/bash)切换至程序目录，然后输入pip install -r requirements.txt,本程序在编写是没有写异常，所以当数据库配置不当或者其他问题是，请自行搜索出现的错误，测试运行程序使用的是win10 1803系统，并未对UNIX系统进行适配,在读代码时请参照源代码文件，由于Python讲究缩进，所以本文后的代码只作为功能参考

**2.数据库设计**

**2.1 概念结构设计**

**2.1.1 E-R图**



**2.2 数据库设计**

**2.2.1 模式设计**

数据库设计模式采用的是主从模式，主从模式，是数据库设计模式中最常见，也是日常设计工作中用的最多的一种模式，他描述了两个表之间的主从关系，是典型的一对多关系。

对象的个数较多且不固定；各个对象之间的属性几乎没有差异；对象的属性在数据库设计阶段能够完全确定；各个对象没有独立的业务处理需求，此时用“主从模式”。将各个对象设计为“从表”的记录，与“主表”对象建立一对多的关系。

在本数据库应用系统中，建立了四个主从模式的表，他们分别是，学生信息表，账户信息表，借阅记录表以及图书信息表，四个表之间中，学生信息表，账户信息表以及借阅记录表通过bsno/sno字段进行关联，图书信息表以及借阅记录表通过图书书籍名进行关联其中学生表与账户信息表中学生表为主表，账户信息表为从表，而在图书信息表与借阅记录表中，图书记录表为主表，借阅记录表为从表。操作借阅记录表的同时会影响图书信息表的拥有书籍书字段的值，创建账号的同时也会关联学生信息

**2.2.2 数据库表结构**

**学生信息表(students)**

学生信息包括真实姓名，性别学号，电话号码，专业以及班级

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段描述 | 字段类型 |
| sname | 学生姓名 | varchar(2000) |
| sno | 学号 | varchar(2000) |
| sex | 性别 | varchar(100) |
| telnum | 电话号码 | varchar(50) |
| major | 专业 | varchar(2000) |
| class | 班级 | varchar(1000) |

**账户信息(admin)**

账户信息包括账户名，账户密码，以及账户角色

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段描述 | 字段类型 |
| account | 账户名 | varchar(200) |
| pwd | 密码 | varchar(200) |
| role | 角色 | varchar(200) |
| sno | 学号/工号 | varchar(200) |

**书籍信息(books)**

书籍信息包括书名，书的作者，住的编号，书的价格以及拥有的数量

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段描述 | 字段类型 |
| bname | 书籍名 | varchar(2000) |
| author | 书籍作者 | varchar(2000) |
| bnum | 书籍编号 | varchar(2000) |
| bprices | 书记价格 | varchar(2000) |
| had | 拥有数量 | varchar(2000) |

**借阅记录(logs)**

借阅记录包括借阅书籍的书名，借阅者的学号，借阅时间，借阅时长，借阅状态

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段描述 | 字段类型 |
| bname | 借阅的书籍名 | varchar(200) |
| bsno | 借阅者的学号/工号 | varchar(200) |
| btime | 借阅时间 | date(0) |
| howlong | 借阅时长 | varchar(200) |
| outline | 节借阅状态 | varchar(200) |

**3.应用程序设计**

**3.1系统设计**

**3.1.1 功能模块图**

本程序包含有登陆模块，主界面显示模块，书记搜索模块，书记添加模块，还书借书模块这几个大模块，而这几个大模块还会掉用一些小模块来提高运行效率以及提高代码的可读性，其次还调用了一些系统模块来处理登陆密码隐写以及数据库连接等功能，具体情况如下图(箭头表示调用顺序，采用未加粗字体的为小模块或模块简介)

学生显示的学生页面:1.修改密码 2.图书查询 3. 图书借阅/归还 4.退出

显示的学生页面:1.修改密码 2.图书查询 3. 图书借阅/归还 4.退出

界面显示

**菜单显示模块**

**登陆模块**

角色获取模块

显示的管理员页面:1.修改管理员密码 2.图书查询 3.图书添加/删除4. 图书借阅/归还 5.退出

学生搜索模块

显示所有图书模块

管理员

需要调用

**借书模块**

**密码修改模块**

**还书模块**

**图书查询模块**

通过输入的选项执行相应的模块

**加书模块**

**书籍添加/删除模块**

**删书模块**

**还书模块**

**图书借阅/归还模块**

**借书模块**

**3.1.2模块功能说明**

**1.登陆模块(login)：**输入账号和密码，调用系统自带的隐写模块(getpass)，隐写密码，输入次数超过三次退出系统，三次以内重新调用该模块，这些情况下返回数值0告诉main方法登陆状况，登陆成功则返回登陆用户名

**2.菜单显示模块(menu):**此模块会调用角色获取模块，并通过调用系统的内置时间方法(time)获取当地时间并显示，并通过登陆模块(login)返回的用户名参数以及角色获取模块(get\_role)返回的角色信息显示到程序主界面，读取输入的选择并返回给main函数的choice变量

**3.角色获取模块(get\_role):**登陆成功后会调用此模块，通过登录模块(login)中输入的用户名搜索数据库得到他的角色，并返回角色名给菜单模块(menu),通过菜单模块(menu)打印在程序主界面

**4.密码修改模块(change\_pwd):**只能修改当前用户的密码，先会要求输入原密码登陆才能进行修改，若错误超过三次，自动退出程序，若密码输入真确，则可修改密码，新密码要求输入两次已确认输入正确，最后将新密码存入数据库，当密码修改成功后则会要求重新登陆系统(登陆模块)

**5.图书查询模块(search\_book):**通过传入书的作者，名称，编号等参数，在数据库中查询相关书籍，并返回查询结果(元组),若元组长度为零，则无此书，反之则遍历元组内的元素并打印。

**6.借书还书模块(broww\_back\_book):**该模块属于总结性模块且要求用户选择是借书还是还书，然后分别调用借书模块(brrow\_book)和还书模块(back\_book)

**7.借书模块(brow\_book):**输入学生学号或者姓名通过调用学生搜索方法(search\_all\_student)确认是否有这个学生,若存在这个学生，则调用显示所有书籍的方法(show\_all\_book)，然后让学生/管理员选择并输入图书的书名和编号后，通过数据库查询书名，书号，书的剩余本书，学号，通过内置的time方法获取借阅时间，然后将借阅记录存入数据库

**8.还书模块(reback\_book):**输入学生学号或者姓名通过调用学生搜索方法(search\_all\_student)确认是否有这个学生,若存在这个学生，则调用显示所有书籍的方法(reback\_book)，通过读取数据库中的log表获取这个学生所借阅的书籍并将编号存入book字典中，让用户选择归还那本书，用户归还后log表中的outline字段会显示已归还，books表中该书的had字段中书的数目加1

**9.书籍添加/删除模块(add\_del\_book):**该模块属于总结性模块且要求用户选择是加书还是删书，然后分别调用添加书籍模块(add\_book)和删除书籍模块(del\_book)，此模块只有在登录用户角色为管理员时才会在菜单方法(menu)中显示

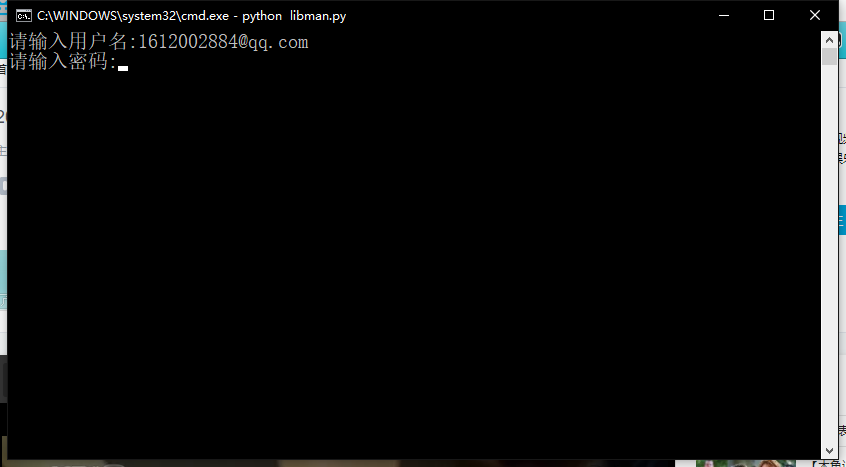
**10.添加书籍模块(add\_book):**用户输入图书的相关信息(书名，编号，作者，价格，拥有数量）后，打印在屏幕上，当用户确认无误后输入y确认，图书信息将会存入数据库中的books表中。

**11.添加书籍模块(add\_book):**通过调用书记搜索(search\_book)方法获取用户输入的要删除图书的相关信息并显示，若搜索没有所输入的书籍则返回没有相关书籍的提示，并返回上一级菜单，若搜索到相关书籍，通过输入书记的具体名字以及编号删除该书籍在数据库books表中的记录

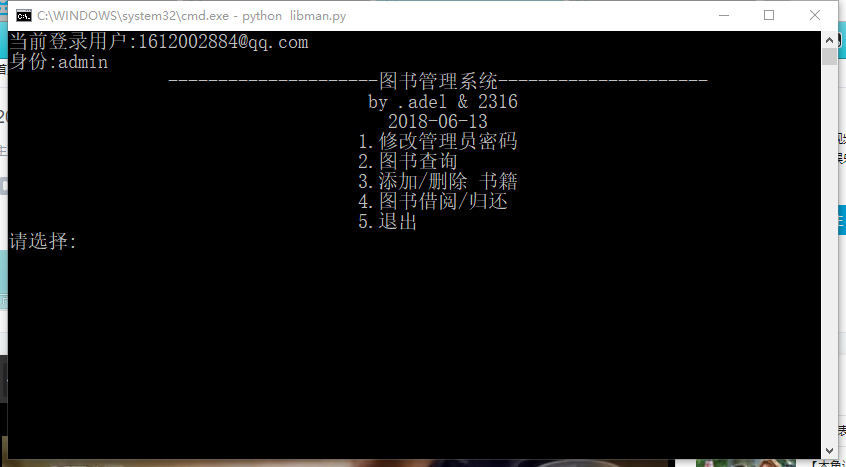
**3.2 系统实现**

**3.2.1界面设计**

登陆界面



主界面



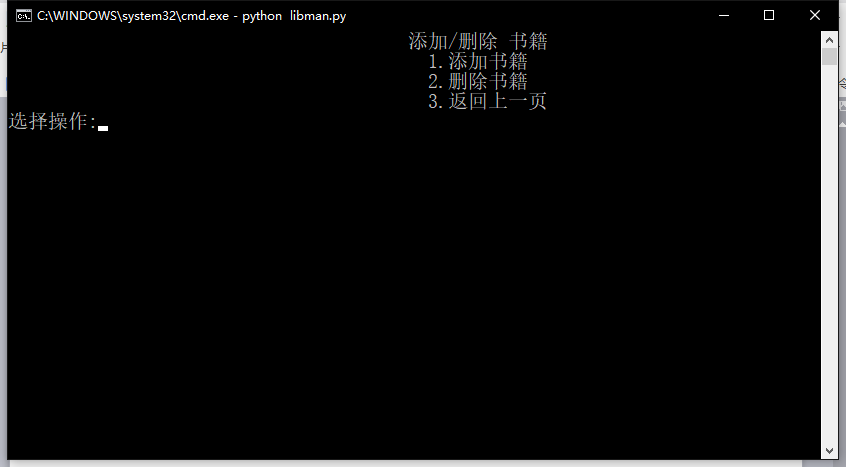
图书查询界面



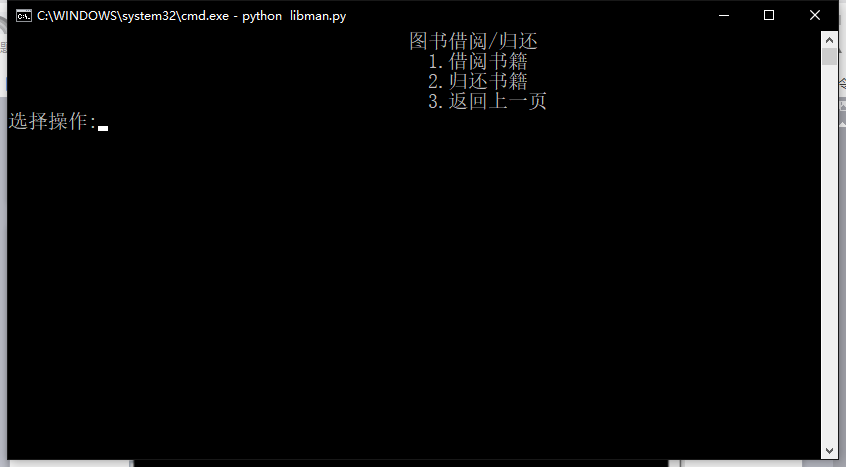
密码修改界面



书记添加和删除界面



图书借阅归还界面



**3.2.2程序流程图**

**3.2.3代码及其注释**

#coding:utf-8  **//确定统一编码**

import pymysql **//mysql操作库**

import string

import getpass **//密码隐写方法导入**

import os

import time

from datetime import date

class book(object): **//图书类，用来封装图书信息**

name = None

author = None

id = None

price = 0.0

had = 0

class student(object): **//学生类，用来封装学生信息**

name = None

sno = None

sex = None

tel = None

major = None

classes = None

def login(): **//登陆模块**

os.system('cls')

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset = 'utf8'

)

cu = db.cursor() **//数据库连接以及操作游标的生成**

uname = str(input("请输入用户名:"))

pwd = str(getpass.getpass("请输入密码:"))

r = cu.execute('select account from admin where account = %s',(uname))

if r == 1:

re = cu.execute('select pwd from admin where account = %s',(uname))

upwd = cu.fetchmany()[0][0]

if pwd == upwd:

os.system('cls')

print('登陆成功^\_^') **//从数据库读取账号和密码进行登陆对比**

db.close()

time.sleep(1)

return uname

else:

os.system('cls')

print('用户名或密码错误。。。。')

time.sleep(1)

return 0

else:

os.system('cls')

print("用户名或密码错误。。。。")

time.sleep(1)

return 0 **//登陆模块当成功登陆时返回登陆用户名，失败时返回0**

def add\_admin(): **//账户添加模块**

account = input('输入新用户名:')

pwd = input('输入新用户名的密码:')

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset = 'utf8'

)

cu = db.cursor()

r = cu.execute('select account from admin where account = %s', (account))

if r == 0: **//判断用户名是否已经存在**

cu.execute("insert into admin(account,pwd) values(%s,%s)", (account,pwd))

print('添加用户' + account + '成功')

db.commit()

cu.close()

db.close()

time.sleep(1)

else:

print("用户名已存在。。")

db.close()

time.sleep(1)

def search\_book(bname): **//图书搜索模块**

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset='utf8'

)

cu = db.cursor()

cu.execute("select \* from books where bname LIKE '%{}%' or author LIKE '%{}%' or bnum like '{}'".format(bname,bname,bname))

return cu.fetchall() **//返回搜索结果构成的元组**

def search\_student(sname): **//学生搜索模块，为后期的借书调用**

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset = 'utf8'

)

cu = db.cursor()

cu.execute("select \* from students where sname = '{}' or sno = '{}'".format(sname,sname))

return cu.fetchall() **//返回学生搜索结果**

def add\_del\_book():

print('\t\t\t\t\t 1.添加书籍')

print('\t\t\t\t\t 2.删除书籍')

print('\t\t\t\t\t 3.返回上一页')

choice = str(input("选择操作:"))

if choice == '1':

add\_book()

elif choice == '2':

del\_book()

elif choice == '3':

return 0

else:

os.system('cls')

print('输入的选项有误，请重新输入')

time.sleep(1)

os.system('cls')

add\_del\_book()

def add\_book():

os.system('cls')

books = book()

books.name = str(input('书籍名称:'))

books.author = str(input('书籍作者:'))

books.id = str(input('书籍编号:'))

books.price = str(input('书籍价格:'))

books.had = int(input('书籍本书:'))

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset = 'utf8'

)

cu = db.cursor()

if len(search\_book(books.name)) == 0:

cu.execute('insert into books(bname,author,bnum,bprices,had) values(%s,%s,%s,%s,%s)',('《' + books.name + '》',books.author,books.id,books.price + '元',str(books.had) + '本'))

print("书名:" + books.name + "\n作者:" + books.author + "\n编号:" + books.id + "\n价格:" + books.price + "\n剩余本书:" + str(books.had) + "\n")

ans = input('\n确认无误(y/n):')

if ans == 'y' or ans == 'Y':

db.commit()

cu.close()

db.close()

print('\n录入成功!!!')

os.system('pause')

else:

os.system('cls')

add\_del\_book()

else:

print("\n书籍已存在....")

time.sleep(1)

os.system('cls')

add\_del\_book()

def show\_all\_book():

os.system('cls')

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset='utf8'

)

cu = db.cursor()

cu.execute('select \* from books')

for i in cu.fetchall():

print("书名:" + i[0] + "\n作者:" + i[1] + "\n编号:" + i[2] + "\n价格:" + str(i[3]) + "\n剩余本书:" + str(i[4]) + "\n")

def del\_book():

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset = 'utf8'

)

cu = db.cursor()

show\_all\_book()

books = str(input('输入你想删除的书的书名:'))

re = search\_book(books)

if len(re) != 0:

os.system('cls')

for i in re:

print("书名:" + i[0] + "\n作者:" + i[1] + "\n编号:" + i[2] + "\n价格:" + str(i[3]) + "\n剩余本书:" + str(i[4]) + "\n")

ans = input('\n确认删除(y/n):')

if ans == 'y' or ans == 'Y':

cu.execute("delete from books where bname like '《{}》'".format(books))

db.commit()

cu.close()

db.close()

print('\n删除成功!!!')

os.system('pause')

else:

print('\n没有找到所输入的书籍。。。\n')

time.sleep(1)

os.system('pause')

def broww\_back\_book():

print('\t\t\t\t\t 1.借阅书籍')

print('\t\t\t\t\t 2.归还书籍')

print('\t\t\t\t\t 3.返回上一页')

choice = str(input("选择操作:"))

if choice == '1':

brow\_book()

elif choice == '2':

reback\_book()

elif choice == '3':

return 0

else:

os.system('cls')

print('输入的选项有误，请重新输入')

time.sleep(1)

os.system('cls')

broww\_back\_book()

def brow\_book():

student = str(input("输入学生的学号或姓名:"))

sre = search\_student(student)

if len(sre) != 0:

show\_all\_book()

books = input('输入你想借阅的书籍的书名或编号:')

bre = search\_book(books)

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset = 'utf8'

)

cu = db.cursor()

if len(bre) != 0:

os.system('cls')

if int(bre[0][4][:-1]) != 0:

for i in bre:

print("书名:" + i[0] + "\n作者:" + i[1] + "\n编号:" + i[2] + "\n价格:" + str(i[3]) + "\n剩余本书:" + str(i[4]) + "\n")

ans = input('\n确认借阅(y/n):')

if ans == 'y' or ans == 'Y':

while True:

days = input("输入借阅天数(小于30天):")

if days.isdigit() or (days <= 30 and days > 0):

break

else:

print("输入天数不符合要求...")

had = int(bre[0][4][:-1])

bname = str(bre[0][0])

bsno = str(sre[0][1])

sname = str(sre[0][0])

times = date.today()

btime = times

outline = 0

cu.execute('insert into logs(bname,bsno,btime,howlong,outline) values (%s,%s,%s,%s,%s)',(bname,bsno,str(times),str(days) + '天',str(outline) + '天'))

cu.execute("update books set had = '{}' where bname = '{}'".format(str(had - 1)+'本',bname))

db.commit()

cu.close()

db.close()

print('\n借阅成功!!!')

os.system('pause')

else:

os.system('cls')

print('\t\t\t\t\t图书借阅/归还')

broww\_back\_book()

else:

print("所选书籍已没有存库。。。抱歉。")

os.system('cls')

print('\t\t\t\t\t图书借阅/归还')

broww\_back\_book()

else:

print('\n没有找到所输入的书籍。。。\n')

time.sleep(1)

os.system('pause')

os.system('cls')

print('\t\t\t\t\t图书借阅/归还')

broww\_back\_book()

else:

print('\n没有找到输入的学生。。。\n')

time.sleep(1)

os.system('pause')

os.system('cls')

print('\t\t\t\t\t图书借阅/归还')

broww\_back\_book()

def show\_stu\_logs(sno):

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset = 'utf8'

)

cu = db.cursor()

cu.execute("select \* from logs where bsno = '{}'".format(sno))

return cu.fetchall()

def reback\_book():

os.system('cls')

student = str(input("输入学生的学号或姓名:"))

sre = search\_student(student)

books = []

if len(sre) != 0:

bname = sre[0][0]

bsno = sre[0][1]

logs = show\_stu\_logs(bsno)

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset = 'utf8'

)

cu = db.cursor()

os.system('cls')

for log in logs:

cu.execute("select bnum from books where bname = '{}'".format(log[0]))

print(cu.fetchall())

print("借阅者:{}\n学号:{}\n借阅书籍:{}\n借阅时间:{}\n借阅时长:{}\n借阅状态:{}\n".format(bname,bsno,log[0],log[2],log[3],log[4]))

books.append(log[0])

os.system('pause')

book = input('输入要归还的书的书名或编号:')

cu.execute("select bname from books where bname = '{}' or bnum = '{}' or bname like '{}'".format(book,book,book))

print(cu.fetchall())

os.system('pause')

if book in books:

cu.execute("update logs set outline = '已归还' where bname")

else:

pass

else:

os.system('cls')

print('输入的学生信息有误。。。')

os.system('pause')

os.system('cls')

broww\_back\_book()

def change\_pwd(a):

i = 0

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman'

)

cu = db.cursor()

cu.execute("select pwd from admin where account = '{}'".format(a))

oldpwd = cu.fetchall()

while i<3:

pwd = str(getpass.getpass("请输入用户{}的密码:".format(a)))

if pwd == oldpwd[0][0]:

os.system('cls')

while True:

pwd = str(getpass.getpass("请输入用户{}的新密码:".format(a)))

pwd1 = str(getpass.getpass("请再次输入用户{}的新密码:".format(a)))

if pwd == pwd1:

cu.execute("update admin set pwd = '{}' where account = '{}'".format(pwd,a))

db.commit()

cu.close()

db.close()

os.system('cls')

print('管理员{}密码修改成功。。请重新登陆。'.format(a))

os.system('pause')

os.system('cls')

login()

break

else:

os.system('cls')

print("两次输入的密码不一致，请重新输入。。。")

os.system('pause')

os.system('cls')

break

else:

os.system('cls')

print("输入管理员{}的密码错误。。".format(a))

os.system('pause')

os.system('cls')

i += 1

if i == 3:

os.system('cls')

print("多次错误。。。退出。。")

os.system('pause')

os.\_exit(0)

def menu(role):

print("\t\t---------------------图书管理系统---------------------")

print("\t\t\t\t by .adel & 2316")

print('\t\t\t\t '+ str(date.today()))

if role != 'student':

print('\t\t\t\t 1.修改管理员密码') #增

print('\t\t\t\t 2.图书查询') #查

print('\t\t\t\t 3.添加/删除 书籍') #增,删

print('\t\t\t\t 4.图书借阅/归还') #增,改

print('\t\t\t\t 5.退出')

choice = input('请选择:')

return choice

else:

print('\t\t\t\t 1.修改密码')

print('\t\t\t\t 2.图书查询')

print('\t\t\t\t 3.图书借阅/归还')

print('\t\t\t\t 4.退出')

choice = input('请选择:')

if choice == '4':

choice = '5'

if choice == '3':

choice = '4'

return choice

def get\_role(name):

db = pymysql.connect(host='localhost',

port=3306,

user='root',

passwd='weiruyi123',

db='libman',

charset='utf8'

)

cu = db.cursor()

cu.execute("select role from admin where account = '{}'".format(name))

return cu.fetchall()[0][0]

i = 0

while i < 3:

a = login()

if type(a) == str:

role = get\_role(a)

break

else:

i += 1

if i == 3:

os.system('cls')

print('不知道密码就别猜啦。。。-\_-')

os.\_exit(0)

os.system('cls')

while True:

print('当前登录用户:{}'.format(a))

print('身份:{}'.format(role))

choice = menu(role)

if choice=='1':

os.system('cls')

print('\t\t\t\t\t管理员密码修改')

change\_pwd(a)

os.system('cls')

elif choice =='2':

os.system('cls')

print('\t\t\t\t\t图书查询')

bname = input('输入书籍名称或者作者:')

result = search\_book(bname)

if len(result) != 0:

os.system('cls')

for i in result:

print("书名:" + i[0] + "\n作者:" + i[1] + "\n编号:" + i[2] + "\n价格:" + str(i[3]) + "\n剩余本书:" + str(i[4]) + "\n")

os.system('pause')

else:

os.system('cls')

print('没有相关的书籍。。。')

time.sleep(1)

os.system('cls')

elif choice =='3':

if role == 'student':

print('输入的选项无效,请重新输入。。。')

time.sleep(1)

os.system('cls')

continue

os.system('cls')

print('\t\t\t\t\t添加/删除 书籍')

add\_del\_book()

os.system('cls')

elif choice =='4':

os.system('cls')

print('\t\t\t\t\t图书借阅/归还')

broww\_back\_book()

os.system('cls')

elif choice =='5':

os.system('cls')

print("正在退出。。。。")

time.sleep(1)

os.system('cls')

break

else:

print('输入的选项无效,请重新输入。。。')

time.sleep(1)

os.system('cls')

os.system('exit')

**4.程序小结**

通过此次的课程设计，对于数据库与程序之间的结合与操作有了更深层次的了解和应用，自己也在此过程中重新复习了Python语言，对于一些Python的模块运用也更加的熟悉，但美中不足的是由于后期代码已经完成了很多，当时没有想到加图形界面，导致后期的话很麻烦，所以放弃了做GUI的想法，这是最遗憾的，还有就是代码没有进行必要的精简，没有进行审计，而且在数据库读取写入以及文件的操作时也没有进行抛异常操作，若真的是做投入市场的产品，这写对于用户是十分的不友好的，在今后我会改进