

**2022 年 秋 季学期**

**课程名称： 软件系统分析与设计**

**报告名称： 软件系统分析与设计“图书馆管理系统”系统汇报**

**汇报小组: 第二小组**

**小组成员: 符梦鑫、王义钧、程旭辉、吴鑫磊、王俊蕊、李泊林**

目录

**一、引言4**

1.编写目的4

2.项目概述4

**二、项目概述4**

1.主要实现的功能的框架图4

2.项目使用到的所有类5

(1)Book ---书籍类5

(2)Manager---管理员类6

(3)User---用户类7

(4)Record---记录类8

(5)FR(FineRecord)---罚款记录类9

(6)OR\_U(OperationRecord\_User)---管理员用户操作记录类10

(7) OR\_B(OperationRecord\_Book)---管理员书籍操作记录类11

(8)Library---图书馆类11

3.当前用到的所有存储文件11

4.基本功能12

(1)登录部分12

(2)图书的增删改查13

(3)图书的借还15

(4)用户的增删改查16

5.附加功能20

(1)罚款功能20

(2)续借功能20

(3)管理员操作记录20

(4)处理用户欠款21

(5)黑名单功能21

(6)书记丢失申报功能21

(7)限制不同身份用户的借书上限21

**三、分工及开发过程22**

1.小组分工22

1.开发过程22

1. **引言**
2. **编写目的**

通过此报告对我们的开发过程进行总结并系统地介绍我们组开发的

项目，汇总出我们组开发的项目实现的所有功能。

1. **项目概述**

本系统采用C++为开发语言，实现了一个较为完善的图书管理系统。本系统实现了用管理员或用户的身份登录，管理员登录后可以对用户、书籍进行增删改查；用户登录后可以进行对书籍的借阅与归还等功能。

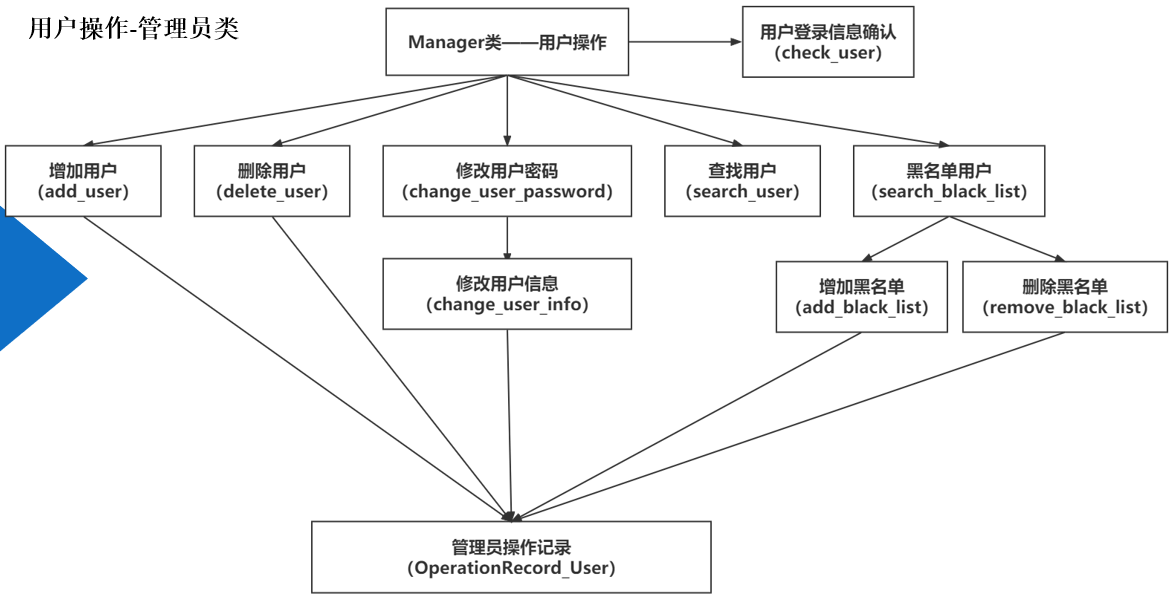
1. **项目概述**
2. **主要实现的功能的框架图**

·用户登录后可以进行的一些操作

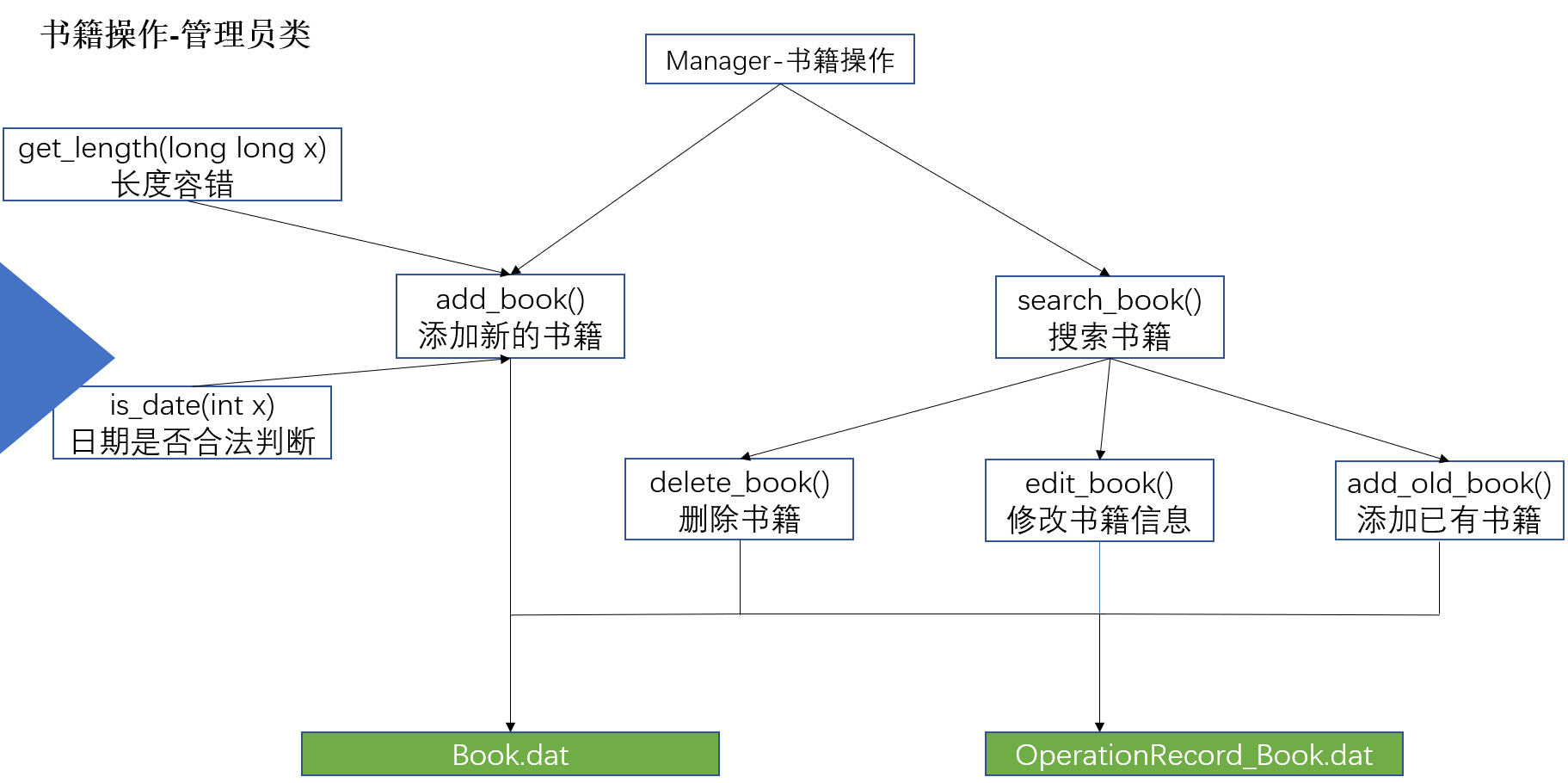
图示

描述已自动生成

·管理员对用户可以进行的操作



·管理员对书籍可以进行的一些操作



1. **项目使用到的所有类：**
2. Book --- 书籍类

class Book

{

private:

int id; // 书籍编号

char name[20]; // 书名，不超过19个字符，一个汉字等于2个字符

char writer[20]; // 作者，不超过19个字符，一个汉字等于2个字符

int state; // 1可借，0不可借被借阅，2丢失，3删除

char press[20]; //出版社

int publication\_date; //出版日期

char category[20]; //类别

long long isbn; // ISBN

double price; // 书籍价格，1000以内

int renew\_time; //续借次数

public:

// 三个构造函数

Book();

Book(int \_id, char\* \_name, char\* \_writer, int \_state, char\* \_press, int \_publication\_date, double \_price, char\* \_category, long long \_isbn, int \_renew\_time);

int get\_id(); // 获取书籍id

char\* get\_name(); // 获取书名

char\* get\_writer(); // 获取作者

int get\_state(); // 获取书籍状态

char\* get\_press();//获取出版社

char\* get\_category();//获取分类

int get\_publication\_date();//获取出版日期

long long get\_isbn();//获取isbn

double get\_price(); // 获取书籍价格

int get\_renew\_time();//获取续借次数

void set\_renew\_time();//设置续借次数(续借加1，天数加30)

void reset\_renew\_time();//重置续借次数为0

void set\_state(int \_state);// 设置状态

void show\_book();//展示书籍信息

void set\_id(int \_id);//设置id

};

1. Manager --- 管理员类

class Manager

{

public:

Manager() {};

Manager(char newName[20], char newPassword[20]);

~Manager() {};

void set\_name(string name); // 设置管理员用户名（Library.cpp)

// 设置管理员密码 （Library.cpp)

void set\_password(string password);

// 获取管理员用户名（OperationRecord\_User.cpp)

char\* get\_name();

void show\_menu(); // 显示菜单栏

void add\_book(); // 添加书籍

void delete\_book(); // 删除书籍

void edit\_book(); // 编辑书籍

void search\_book(); // 查询书籍

User register\_user(); // 注册

void add\_user(); // 添加用户

void delete\_user(); // 删除用户

void search\_user(); // 搜索用户

bool check\_user(User& user); //检查用户是否已注册并返回

void edit\_user\_password(); // 修改用户密码

void show\_record(); // 展示记录

bool save\_password(int id, char password[]); // 保存密码

void exit(); // 注销

void show\_operation\_record\_book();//展示图书操作记录

void show\_operation\_record\_user();//展示用户操作记录

void deal\_fine\_record();

void search\_black\_list(); //查询所有黑名单人员

void add\_black\_list(); //加入黑名单

void remove\_black\_list(); //移除黑名单

void add\_old\_book();//增加已有书籍

private:

char name[20]; // 管理员名称

static char password[20]; //管理员密码

};

1. User --- 用户类

class User {

public:

User() {};

User(int new\_id, char new\_name[], char new\_gender[], int new\_grade, int new\_major, int new\_borrow\_num, int new\_history\_borrow\_num, int new\_state,int new\_status);

~User() {};

void show\_user(); //展示学生信息

void show\_menu(); //展示用户菜单栏（分为正常用户和黑名单用户）

int get\_id(); //获取用户id

char\* get\_name(); //获取用户名

char\* get\_gender(); //获取用户性别

int get\_grade(); //获取用户年级

string get\_major(int \_majorID); //获取用户专业

int get\_majorID(); //获取用户专业ID

int get\_borrow\_num(); //获取借书数量

int get\_history\_borrow\_num(); //获取历史借书数量

char\* get\_password(); //获取用户密码

void set\_id(int \_id); //设置用户id

void set\_name(char \_name[]); //设置用户名

void set\_gender(char gender[]); //设置用户性别

void set\_grade(int \_grade); //设置用户年级

void set\_major(string major); //设置用户专业

void set\_majorID(int major); //设置用户专业ID

void set\_password(char \_passwd[]); //设置用户密码

void set\_borrow\_num(int \_num); //设置用户借书数量

void set\_history\_borrow\_num(int \_num);//设置历史借书数量

int get\_user\_state(); //获取用户状态

void set\_user\_state(int state); //用户状态，0为黑名单，1为存在

void set\_black\_list(); //拉入黑名单

void show\_delay\_book(); //展示逾期未还书籍

void borrow\_book(); //借书

void return\_book(); //还书

void inquire\_history(); //查询历史借书记录

void inquire\_now(); //查询当前借书的信息

bool save\_user\_info(); //保存个人信息

//void change\_password(); //修改密码（已替换为修改信息）

void change\_info(); //修改个人信息

void renew(); //续借书籍

void declare(); //图书丢失申报

int get\_status(); //获取身份

bool get\_fine\_record\_id(int search\_id);//查看该用户是否有欠款信息

void set\_status(int \_status); //设置本科生还是研究生

private:

int id; //学号

char name[11]; //姓名

char gender[10]; //性别

int grade; //年级

int majorID; //专业号

char password[21]; //密码

int borrow\_num; //当前借书数量

int history\_borrow\_num; //历史借书数量

int user\_state; //用户状态，0为被拉黑，1为存在，2为删除

int status; //用户身份0为本科生，1为研究生

};

1. Record --- 记录类

class Record

{

public:

Record(){};

Record(int new\_id\_book, char new\_name\_book[20], int new\_id\_user, char new\_name\_user[20], int new\_state, int record\_state);

~Record() {};

void recording();//保存

int get\_id();//获取书籍id

char\* get\_book\_name();//获取书籍名字

int get\_user\_id();//获取用户id

char\* get\_user\_name();//获取用户名

int get\_book\_state();//获取书籍状态

int get\_record\_state();//获取记录状态

void set\_book\_state(int new\_state);//设置书籍状态

void set\_record\_state(int new\_state);//设置记录状态

void show\_record();//展示记录

void set\_return\_tmp();//设置归还时间

char\* get\_return\_tmp();//获取归还时间

char\* get\_tmp();//获取时间

void set\_tmp(string tmp);//设置借书时间

void set\_user\_name(char \_username[]);//设置用户姓名

private:

int id\_book; // 书籍编号

char name\_book[20]; // 书名，不超过20个字符，一个汉字等于2个字符

int id\_user;//学号

char name\_user[20];//姓名

int book\_state;//书的状态

int record\_state;//记录的状态 0为黑名单记录 1为正常记录

char tmp[256];//时间

char return\_tmp[256];//归还时间

};

1. FR(FineRecord) --- 罚款记录类

class FR

{

public:

FR(){}

FR(int \_id, char\* \_name, int \_category, double \_money );

int get\_id();

char\* get\_name();

void show();

void set\_name(char \_name[20]);

private:

int id;//用户id

char name[20];//用户名

int category;//罚款类型 0-图书丢失 1-超时归还

double money;//罚款金额

};

1. OR\_U(OperationRecord\_User) --- 管理员用户操作记录类

class OR\_U

{

public:

OR\_U(int \_user\_id, char \_user\_name[], int \_grade, int \_major, int \_state, int \_user\_state, char \_manager\_name[]);

OR\_U() {}

int get\_id();//获取用户id

char\* get\_name();//获取用户名

char\* get\_time();//获取时间

int get\_grade();//获取年级

string get\_major();//获取用户专业

char\* get\_ori\_password();//获取修改前密码

void set\_ori\_password(char passwd[20]);//设置用户修改前密码

void set\_now\_password(char passwd[20]);//设置用户修改后密码

char\* get\_now\_password();//获取用户现在密码

char\* get\_manager\_name();//获取管理员名字

int get\_state();//获取操作类型

int get\_user\_state();//获取用户状态

private:

int id;//学号

char name[10];//姓名

int grade;//年级

int majorID;//专业号

char ori\_password[20];//修改前密码

char now\_password[20];//修改后密码

char tmp[256];//时间

char manager\_name[20];//操作人id

int user\_state;//操作人状态

int state;//操作类型

};

1. OR\_B(OperationRecord\_Book) --- 管理员书籍操作记录类

class OR\_B

{

public:

OR\_B(int book\_id1,char \*book\_name1,int state1);

OR\_B(){}

int get\_book\_id();//获取书籍id

char\* get\_book\_name();//获取书籍名

char\* get\_time();////获取书籍时间

char\* get\_manager\_name();//获取管理员名字

int get\_state();//获取操作类型

void set\_book\_id(int \_book\_id);//设置书籍id

private:

int book\_id;//书籍id

char book\_name[20];//书籍名称

char tmp[256];//时间

char manager\_name[20];//操作人id

int state;//操作类型

};

1. Library --- 图书馆类，用于调动整个程序

class Library {

public:

Library();

void begin();//开始

private:

bool sign\_up();//注册

bool sign\_in();//登录

bool manager\_sign\_in();//管理员登录

private:

User \_user;

Manager \_manager;

char \_name[20] = "Manager";//管理员账号

char \_password[20] = "123456";//管理员密码

};

1. **当前用到的所有存储文件：**

·book.dat ------ 存储所有的书籍

·User.dat ------ 存储所有用户的信息

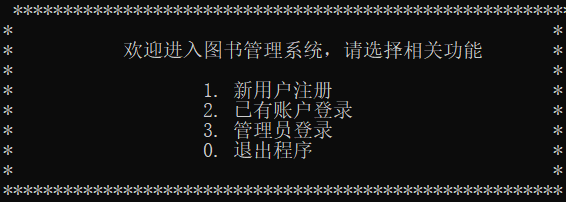
·recording.dat ------ 存储所有的书籍借阅记录

·FineRecord.dat ------ 存储所有的罚款信息

·OperationRecord\_User.dat ------ 存储所有的管理员对用户操作的记录

·OperationRecord\_Book.dat ------存储所有的管理员对书籍操作的记录

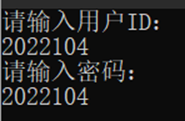
1. **基本功能：**
2. 登录部分

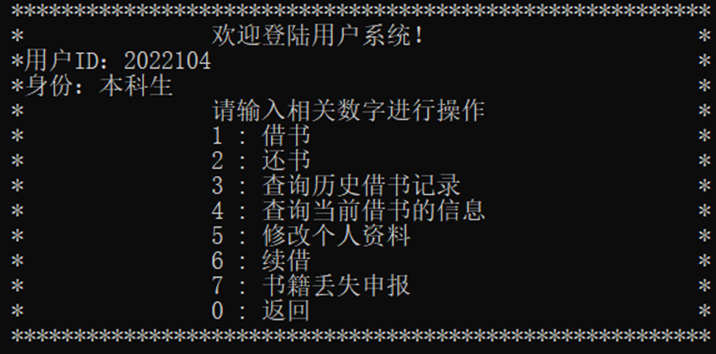


·新用户注册

选择’1’选项进入新用户注册，用户根据提示信息依次输入用户姓名、性别、年级、学院序号、身份选择序号最后确认信息无误后完成用户注册并直接进入用户界面，否则返回重新输入；每一步的输入必须合法

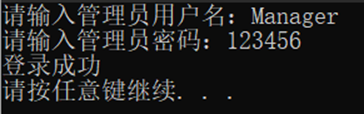
·用户登录

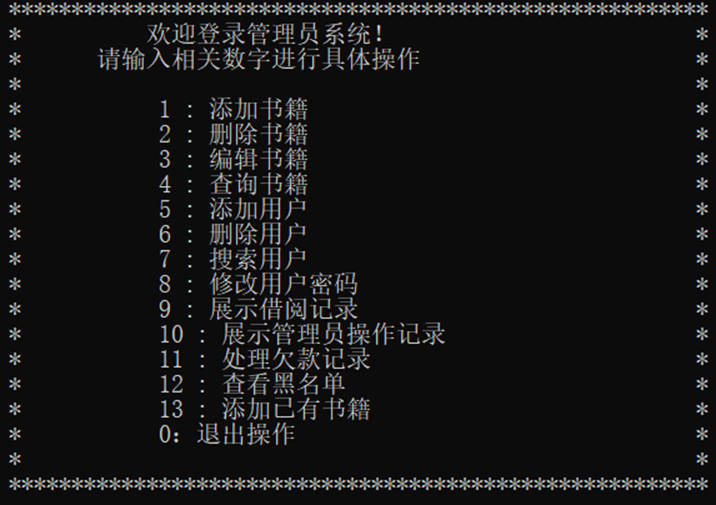




选择’2’选项进入用户登录，根据提示信息输入用户ID和密码，进入学生界面

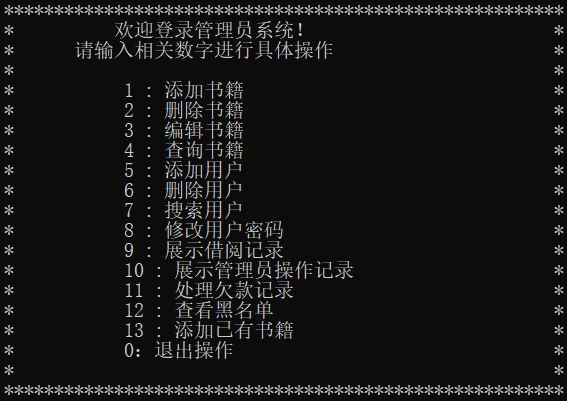
·管理员登录





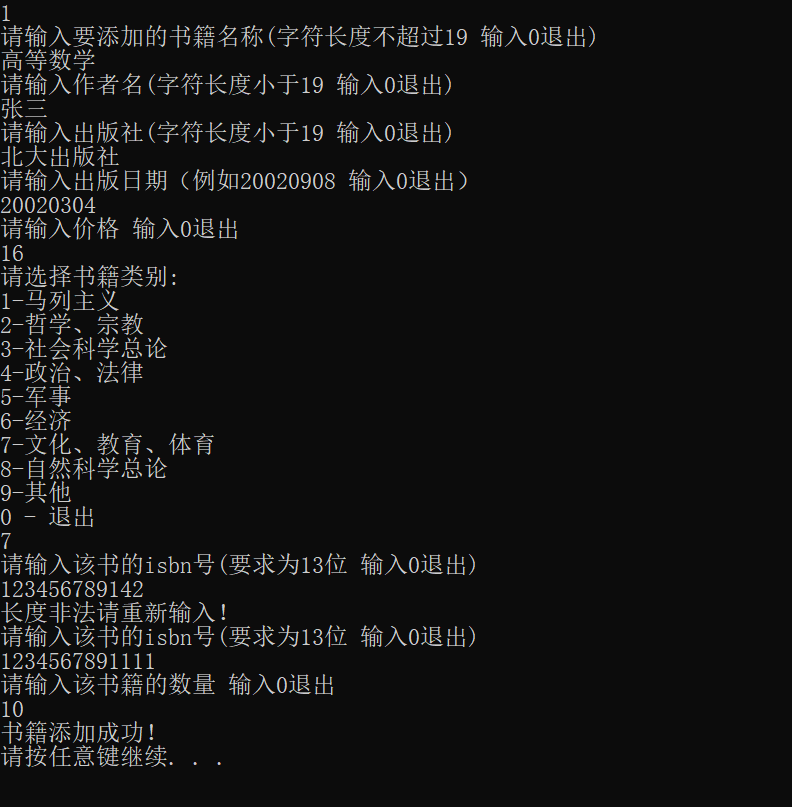
只设置了一个管理员，选择’3’进入管理员登录，根据提示信息依次输入管理员用户名与密码后进入管理员界面

1. 图书的增删改查：
   1. 增加图书



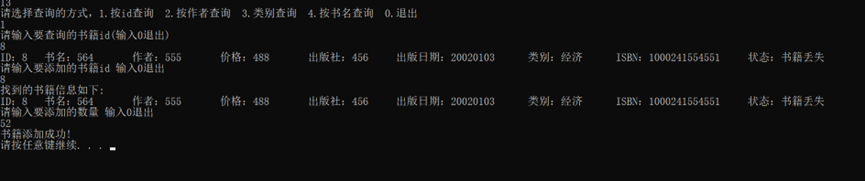
·增加新的书籍

管理员界面选择’1’添加新的书籍，根据提示依次输入或选择新增书籍的信息后，添加书籍成功



·增加已有书籍

管理员界面选择’13’进行已有书籍的添加，根基提示选择出要增加的已有书籍，然后输入要增加的数量，添加成功



* 1. 删除图书

管理员界面选择’2’进入删除书籍，根据提示查询出要删除的书籍，然后输入书籍ID删除书籍



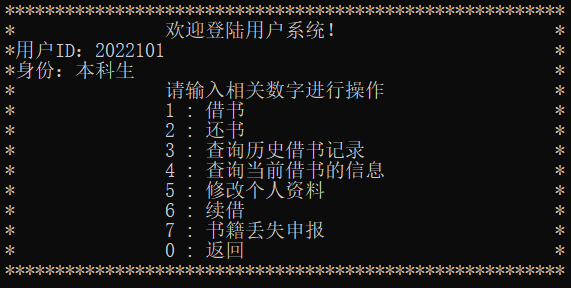
* 1. 修改图书信息

管理员界面选择’3’进行修改书籍信息，按照提示输入修改后的书籍信息，最后确认，修改成功

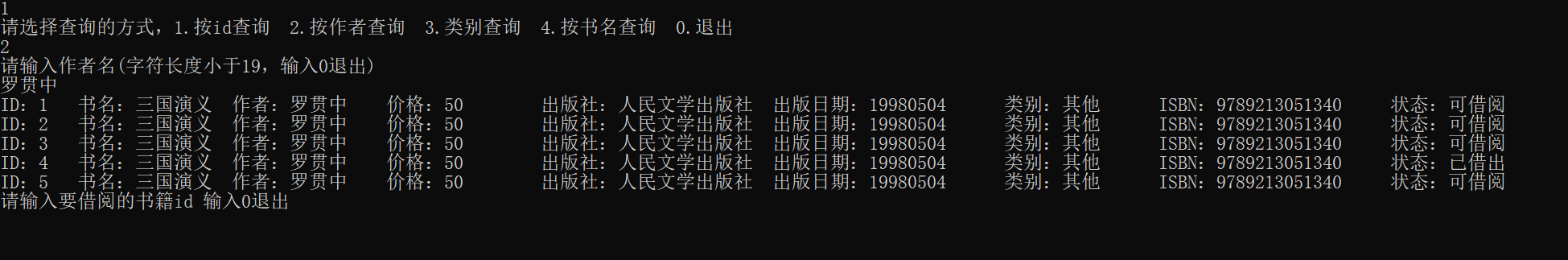


* 1. 查询图书

·用户借书查询书籍



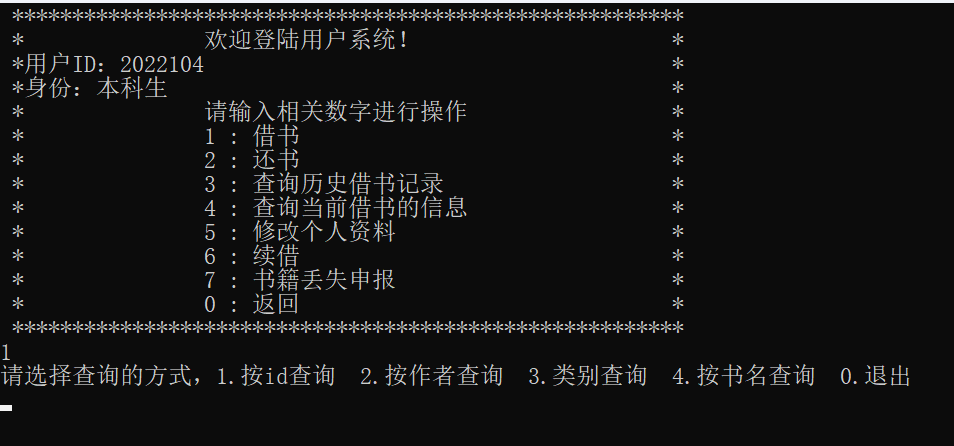
用户界面在选择借书时会先进行书籍的查询然后借书，用户界面选择’1’借书后先需要根据提示查询出要借的书籍，然后进行借书(这里展示按作者名查找)



·管理员查询已有书籍

管理员界面选择’4’进行书籍查询，跟用户进行书籍查询一样也是需要按照提示输入查询方式后进行查询

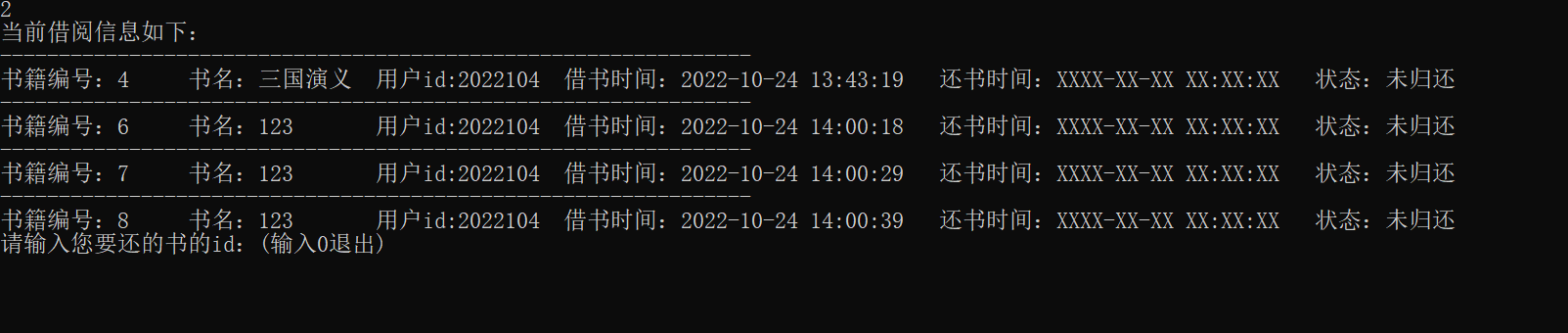
1. 图书的借、还：
   1. 借书



用户界面选择’1’后进行借书，输入1、2、3、4分别可以按照“书籍id”、“作者”、“类别”、“书名”查询书籍，查询出书籍后用户需要输入要借的书的ID 进行借书后借书成功

* 1. 还书

用户界面选择’2’后进行还书，首先会展示出用户当前正在借阅的书籍的信息，用户只能在这些书中进行还书，输入要还的书的ID 后还书成功



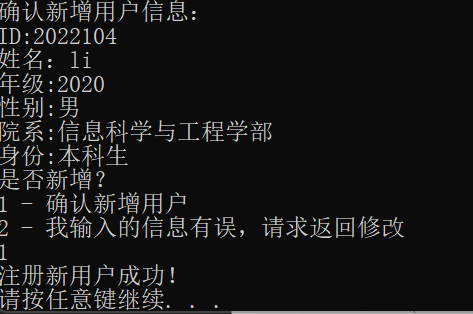
文本

描述已自动生成

1. 用户的增删改查：
   1. 添加新用户

·用户注册

用户可以自己在登录界面进行新用户注册，根据提示输入个人信息并确认无误后注册成功，进入用户界面

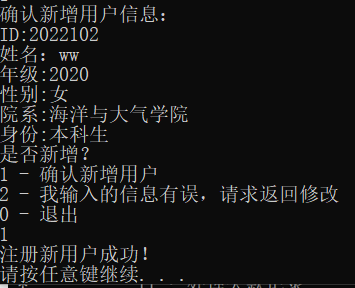
****

·管理员添加用户

管理员界面选择’5’进行添加新用户，根据提示输入用户信息并确认无误后添加用户成功

文本

描述已自动生成



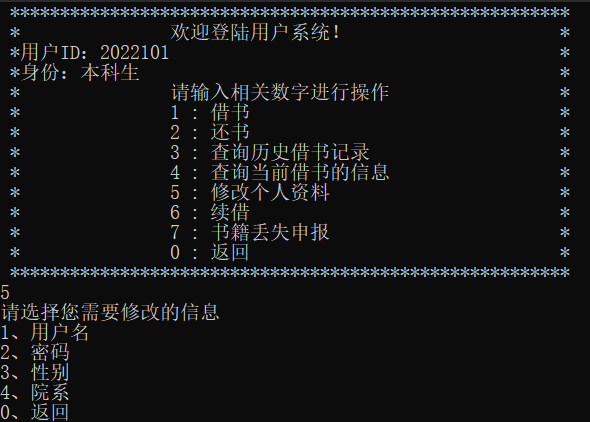
* 1. 删除用户

进入管理员界面后选择’6’进行删除用户操作，需要根据提示输入用户ID，然后删除用户

* 1. 修改用户信息

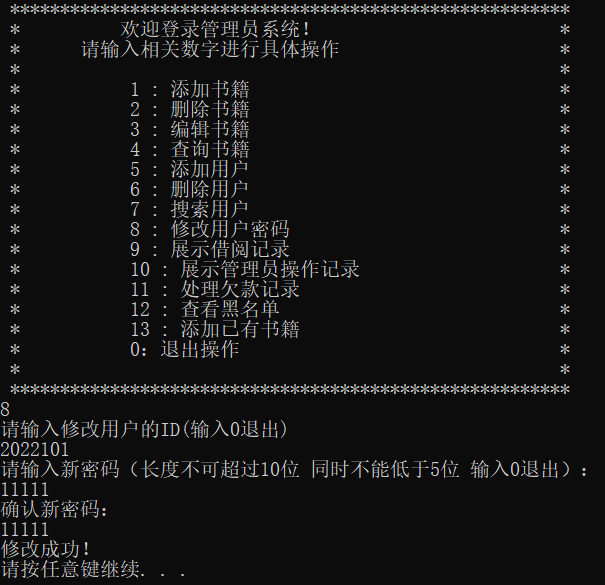
·用户修改自身信息

进入用户界面后选择’5’可以修改个人资料，根据提示选择要修改的内容后进行修改



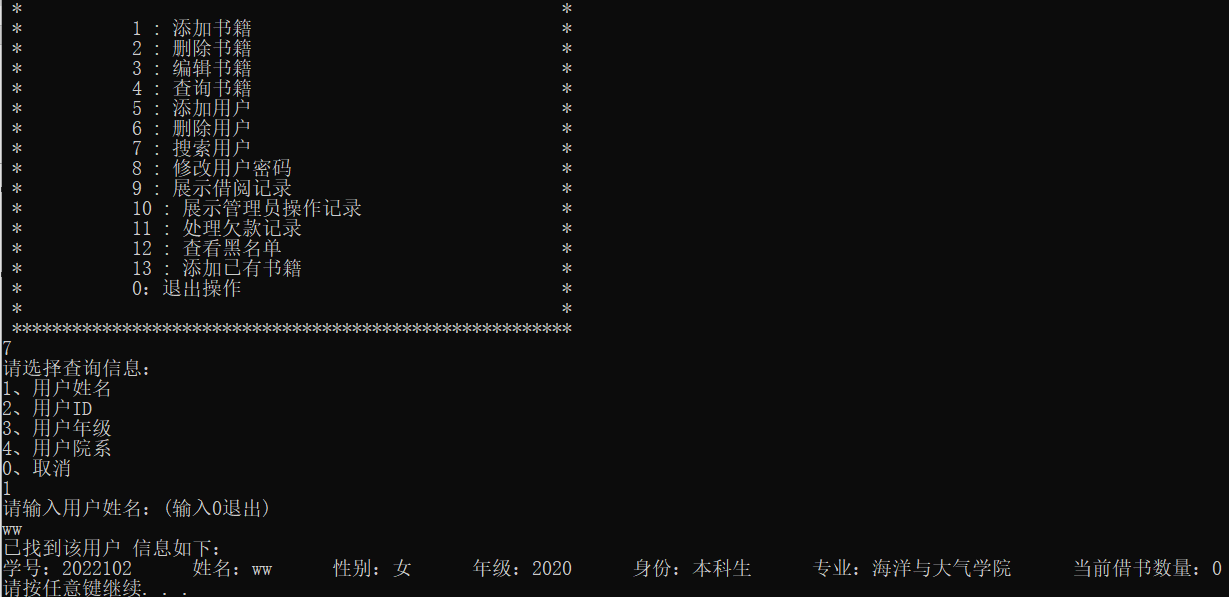
·管理员修改用户密码

进入管理员界面后选择’8’进入修改用户密码界面，根据提示输入要修改的用户的ID，后输入新密码并确认，完成用户密码的修改



* 1. 查询用户

进入管理员界面，选择’7’进行查询用户，根据提示信息选择查找用户的方式，进行用户的查找



* 1. 查询借阅记录

·用户查询当前借书信息

进入用户界面，选择’4’进行当前借阅书籍信息的查询



·用户查询历史借书记录

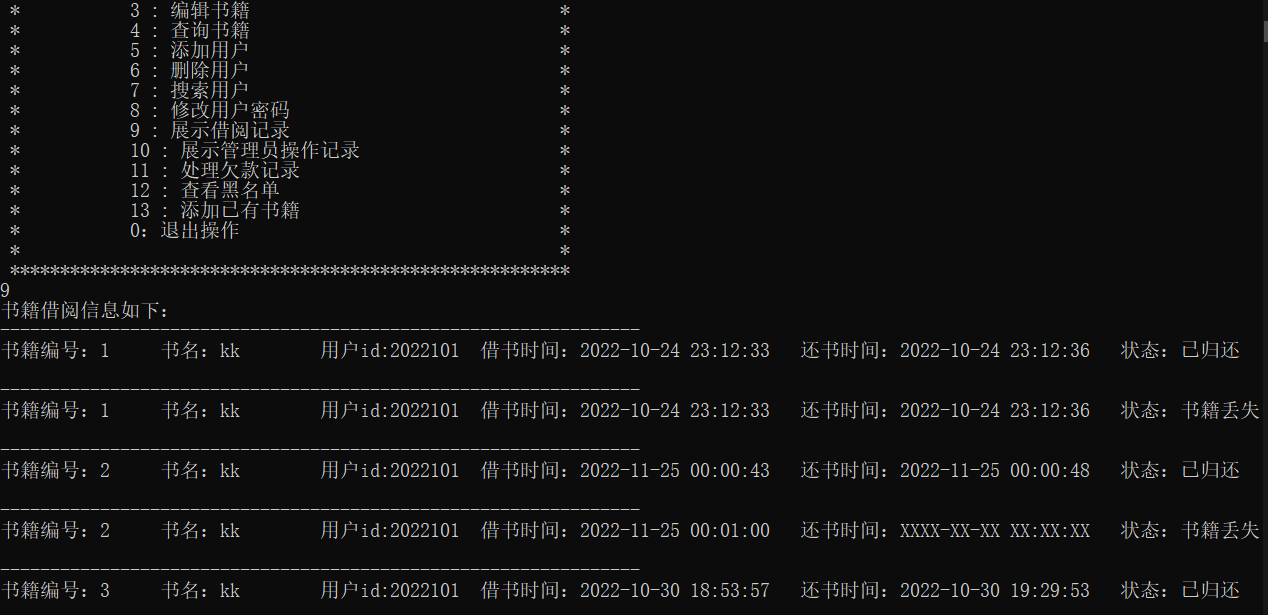
进入用户界面，选择’3’进行历史借阅书籍信息的查询

电脑萤幕画面

描述已自动生成

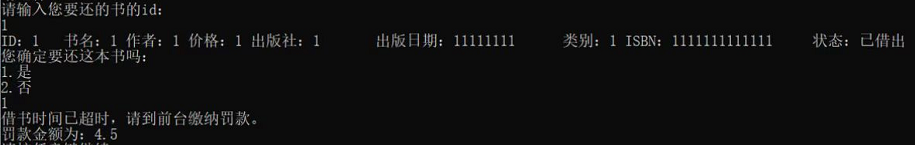
·管理员查询所有借阅记录

进入管理员界面，选择’9’进行所有借书记录的查询



1. **附加功能**：
2. 罚款功能

用户在归还书籍时若超出借书时间限制，则会提示还书超时需缴纳罚款；用户进行图书申报之后也会有罚款记录



文本

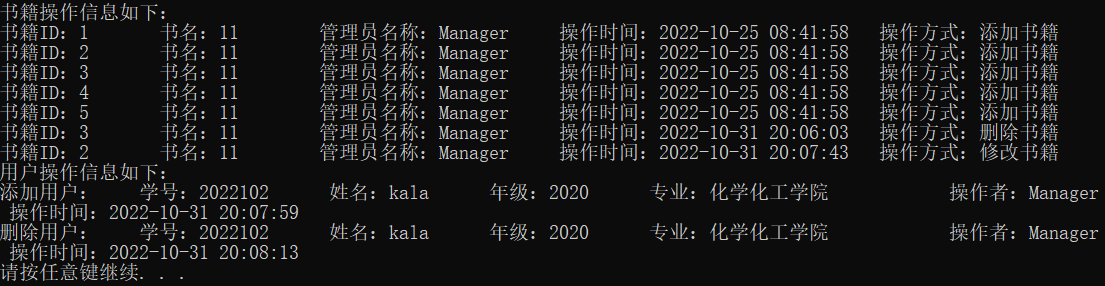
描述已自动生成

1. 续借功能  
   进入用户界面后，选择’6’进行书籍续借，会先显示出该用户当前在正在借阅的所有书籍，用户需要在这些书籍中选择要续借的书籍，输入需要续借的书籍的ID后，续借成功，每本书只能续借1次，续借后借书期限增加30天



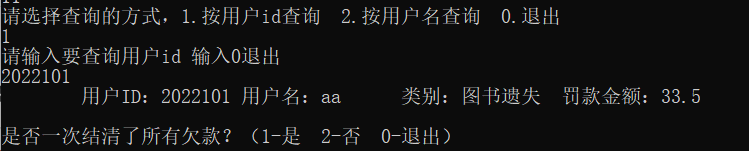
1. 管理员操作记录

进入管理员界面后，输入’10’进入展示管理员操作记录，会展示出管理员所有的操作记录



1. 处理用户欠款、

进入管理员界面后，输入’11’进行用户欠款处理，管理员首先根据提示选择查询罚款信息的方式，查询出之后，进行罚款记录的处理



1. 黑名单功能

黑名单操作逻辑：

·若该用户在黑名单中，且存在逾期未归还书籍，则无法移除黑名单；

·若用户不在黑名单中，但记录中显示该用户存在逾期时间过长未归还书籍记录，则将其加入到黑名单；

·若用户将该长期未还书籍归还，系统逻辑可判断将其移除黑名单，若想立即移除，可联系管理员手动删除。

进入管理员界面后，选择’12’可查看当前黑名单中的人员信息，然后选择相应的操作



1. 书籍丢失申报功能

进入用户界面后，输入’7’进行书籍丢失申报，首先会展示出该用户当前正在借阅的所有书籍，用户根据提示信息输入要进行丢失申报的书籍ID，完成书籍丢失申报，并提示需缴纳罚款

文本

描述已自动生成

1. 限制不同身份用户的借书上限

不同身份用户的借书数量上限是不同的，本科生最多可以同时借5本书，研究生最多可以同时借8本书，当达到最大借书上限之后用户再进行借书会提示当前借书数量已达最大值不可借书

1. **分工及开发过程**
2. 小组分工：

符梦鑫：组长、完成User类的实现、负责完成系统报告

王俊蕊：完成User类的实现、负责完成测试报告

程旭辉：完成Manager类的实现、负责完成技术报告

王义钧：完成Manager类的实现、负责完成技术报告

吴鑫磊：完成Book类的实现、完成主函数、负责完成用户手册

李泊林：完成Book类的实现、完成主函数、负责完成用户手册

1. 开发过程

第1~2周：进行组内讨论，确定开发程序的大概分工以及程序的

大概设计，为后面的开发做准备

第3~4周：完成了User类、Manager类、Book类里的大部分函数接

口

第5周：所有的代码整合到一起，进行debug，并在这周内完成了代

码的初步debug工作，还添加了还书时间记录、借书数量最

大值、借书时间是否超时判断、记录管理员操作记录、图书

续借、书籍丢失申报等功能

第6周：增加了用户身份区分功能、不同用户借书数量上限限制、续

借次数限制、超时还书记录罚款记录、批量增加书籍、图书

丢失申报、黑名单功能

第7周：改进小组汇报之后老师提出的一些问题，并且增加了添加已

有书籍、院系与院系号之间的键值对应，完善了代码的容错

等

第8周：进行代码的测试，并且修改测试中出现的问题，最终通过测

试