# 二、需求分析

## 1.1 功能描述

### **1.1.1“读者”功能**

#### 1）图书的查询：图书的查询可以通过搜索图书id、书名、作者名、出版社来实现,显示结果中需要包括书籍信息以及是否被借阅的情况；

#### 2）图书的借阅：借阅图书时，需要判断该读者现已借阅的图书数量，若超过3本，则暂时不能借书；若尚有未缴纳罚金，暂时不能借书；否则点击“确认借阅”，即完成借阅。

#### 3）图书的归还：归还图书时，若尚有罚金未缴纳，提示学生去管理中心线下缴纳罚金再进行线下还书；否则界面显示所有待还书籍的信息，点击“确认还书”或输入图书id再点击“确认”即完成还书。

#### 4）查看、修改个人信息：个人信息包括用户id、姓名、密码、姓名以及电话；修改个人信息：包括修改姓名、密码、性别、电话等，其中用户id不可修改。

#### 5）查询借阅历史：通过输入书籍的id、书名、作者或出版社来查询借阅记录；记录内容包括读者id、书籍信息和借阅的开始日期以及结束日期；若书籍尚未归还，“结束日期”后会显示“尚未归还”。

**6）查看处罚记录：**处罚记录的内容包括违规的借阅记录的信息；如若相关书籍仍未进行线下归还且读者也未缴纳罚金，则提示尚未缴纳罚金；否则显示该条借阅记录的超期天数。

#### 7）注册

### **1.1.2“管理员”功能**

#### 1）添加图书：输入图书的Id、书名、作者、出版社、其他信息，系统将信息导入数据库；

#### 2）删除图书：如果书籍未被借出，则可通过鼠标点击“删除图书”来完成删除任务；否则不能进行删除。

#### 3）修改图书信息：通过点击“修改信息”，打开修改图书信息页面，修改相应信息后(书名、作者、出版社、其他信息)，点击保存；

#### 4）借阅处罚管理：输入用户id，可以查看用户的借阅违规处罚情况，用户缴纳罚款后，可点击“缴纳罚款”，撤销对该生的处罚，恢复其正常借阅和归还权利；

#### 5）权限管理：输入最长借阅天数可修改读者最长借阅期限；点击“开启”、“关闭”按钮可以开启或者关闭“黑名单”功能。

#### 6）用户管理：可以输入读者id或者姓名来查找读者，点击“删除用户”可以删除该用户；若“黑名单”功能被启用，可以点击“加入黑名单”、“移除黑名单”来控制用户的登陆权限。

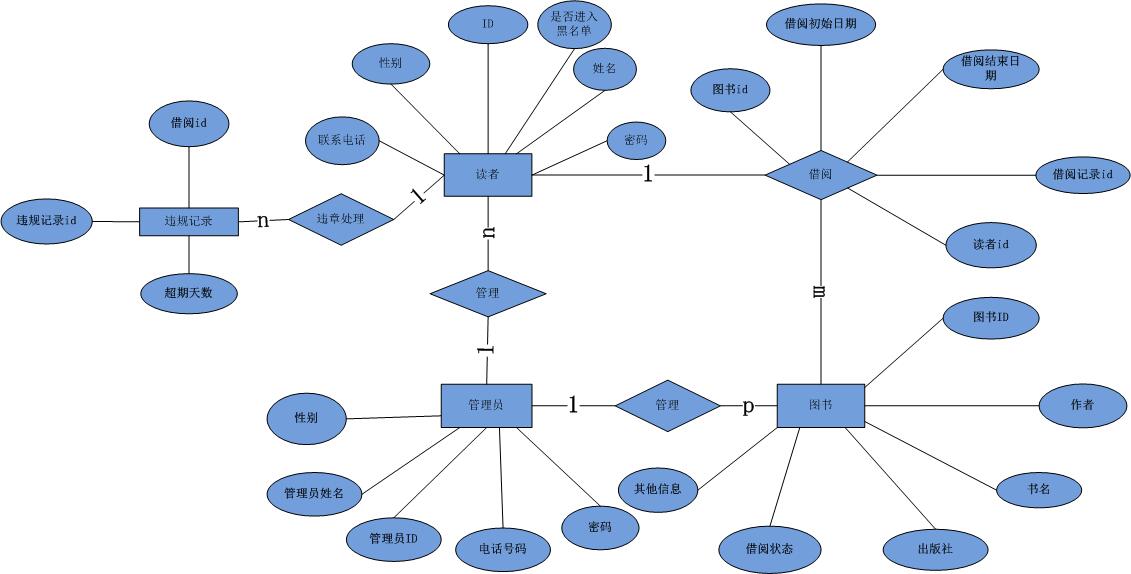
#### 7）管理员个人信息修改

## 1.2 系统用例图



# 三、数据库概要设计

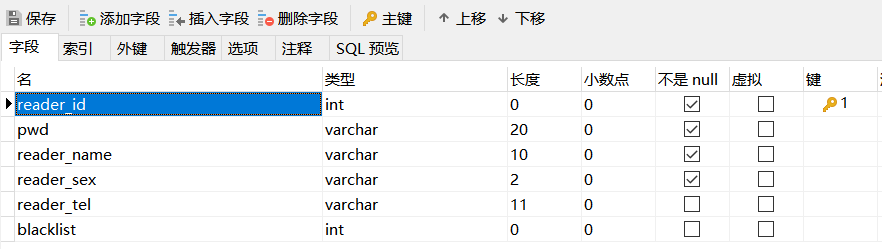
## 2.1 系统E-R图



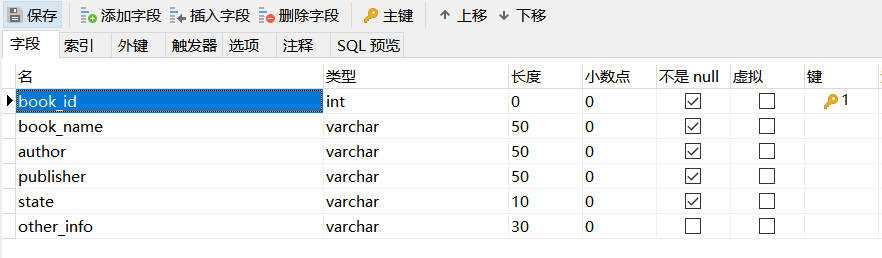
# 四、数据库逻辑结构设计

## 3.1 系统关系模型

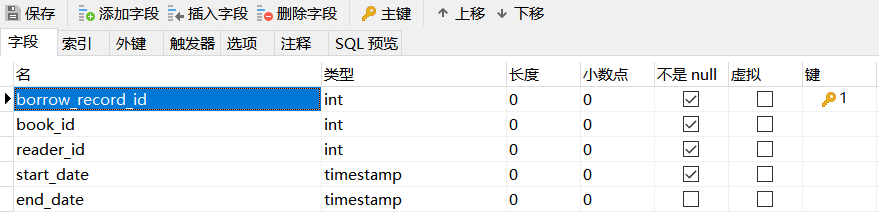
**读者(读者id,密码,姓名，性别，联系电话）**



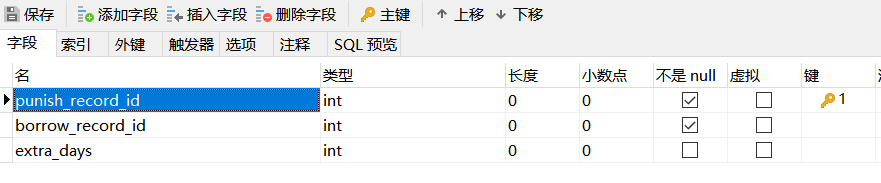
**图书(图书id,书名,作者，出版社，借阅状态，其他信息）**



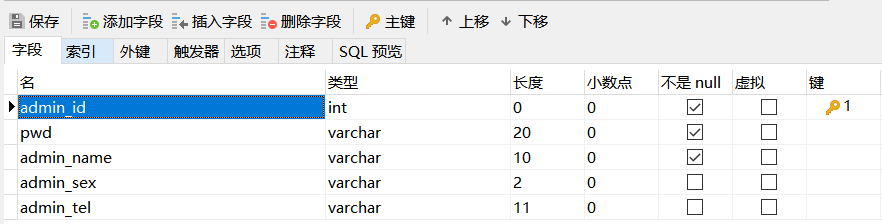
**借阅(借阅记录id，图书id,读者id,借阅初始日期，借阅结束日期）**

****

**违规记录(违规记录id ，借阅记录id，超期天数)**

****

**管理员（管理员id,密码,姓名,性别,联系电话）**

****

## 3.2 数据库结构SQL代码

### **3.2.1基本表**

**读者(读者id,密码,姓名，性别，联系电话）**

CREATE TABLE reader (

reader\_id INT ( 10 ) PRIMARY KEY CHECK(reader\_id LIKE '\_\_\_\_%'),/\*读者号必须大于4位\*/

pwd VARCHAR ( 20 ) CHECK(pwd LIKE '\_\_\_\_%')NOT NULL,/\*密码必须大于4位\*/

reader\_name VARCHAR ( 10 ) NOT NULL,

reader\_sex VARCHAR ( 2 ) NOT NULL CHECK (

reader\_sex IN ( "男", "女" )),/\*性别男或女\*/

reader\_tel VARCHAR ( 11 ) CHECK(reader\_tel LIKE '\_\_\_\_\_\_\_\_\_\_\_')/\*电话必须11位\*/

);

**图书(图书id,书名,作者，出版社，借阅状态，其他信息）**

CREATE TABLE book (

book\_id INT ( 10 ) PRIMARY KEY CHECK(book\_id LIKE '\_\_\_\_%'),/\*书号必须大于4位\*/

book\_name VARCHAR ( 50 ) NOT NULL,

author VARCHAR ( 50 ) NOT NULL,

publisher VARCHAR ( 50 ) NOT NULL,

state VARCHAR ( 10 ) CHECK (

state IN ( '在库', '不在库' )) NOT NULL,/\*书籍状态只有在库与不在库\*/

other\_info VARCHAR ( 30 )

);

**借阅(借阅记录id，图书id,读者id,借阅初始日期，借阅结束日期）**

CREATE TABLE borrow\_record (/\*借阅记录\*/

borrow\_record\_id INT ( 10 ) PRIMARY KEY,

book\_id INT ( 10 ) NOT NULL,

reader\_id INT ( 10 ) NOT NULL,

start\_date TIMESTAMP NOT NULL,

end\_date TIMESTAMP,/\*end\_date可以空值，空值代表未归还\*/

FOREIGN KEY ( book\_id ) REFERENCES book ( book\_id ) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY ( reader\_id ) REFERENCES reader ( reader\_id ) ON DELETE CASCADE ON UPDATE CASCADE );

**违规记录(违规记录id ，借阅记录id，超期天数)**

CREATE TABLE punish\_record (/\*处罚记录\*/

punish\_record\_id INT ( 10 ) PRIMARY KEY,

borrow\_record\_id INT ( 10 ) UNIQUE NOT NULL,

extra\_days INT ( 10 ),/\*可以空值，空值代表尚未缴纳罚款\*/

FOREIGN KEY ( borrow\_record\_id ) REFERENCES borrow\_record ( borrow\_record\_id ) ON DELETE CASCADE ON UPDATE CASCADE

);

**管理员（管理员id,密码,姓名,性别,联系电话）**

CREATE TABLE admin (

admin\_id INT ( 10 ) PRIMARY KEY CHECK(admin\_id LIKE '\_\_\_\_%'),/\*完整性约束同读者\*/

pwd VARCHAR ( 20 ) CHECK(pwd LIKE '\_\_\_\_%') NOT NULL,

admin\_name VARCHAR ( 10 ) NOT NULL,

admin\_sex VARCHAR ( 2 ) CHECK (

admin\_sex IN ( '男', '女' )),

admin\_tel VARCHAR ( 11 ) CHECK(admin\_tel LIKE '\_\_\_\_\_\_\_\_\_\_\_')

);

### **3.2.1存储过程**

------------------------------------------------------------**借书过程**------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `borrow`(readerid INT,bookid INT)/\*借书过程\*/

BEGIN

DECLARE \_state VARCHAR(10);

DECLARE num INT;

SELECT COUNT(\*) INTO num FROM borrow\_record;

SET num=num+1;

SELECT state INTO \_state FROM book WHERE book\_id=bookid;

IF(\_state="在库")

THEN

UPDATE book SET state="不在库" WHERE book\_id=bookid;

INSERT INTO borrow\_record(borrow\_record\_id,reader\_id,book\_id,start\_date) VALUES(num,readerid,bookid,CURRENT\_TIMESTAMP());

END IF;

END

------------------------------------------------------------**还书过程**------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `return\_book`(readerid INT, bookid INT)/\*还书过程\*/

BEGIN

/\*输入id和书的id还书\*/

DECLARE

\_state VARCHAR ( 10 );

DECLARE

num INT;

SELECT

COUNT(\*) INTO num

FROM

borrow\_record;

SET num = num + 1;

SELECT

state INTO \_state

FROM

book

WHERE

book\_id = bookid;

IF

( \_state = "不在库" ) THEN

UPDATE book

SET state = "在库"

WHERE

book\_id = bookid;

UPDATE borrow\_record

SET end\_date = CURRENT\_TIMESTAMP()

WHERE

readerid = reader\_id

AND bookid = book\_id

AND end\_date IS NULL;

END IF;

END

----------------------------------------------------------**更新处罚记录过程**---------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `update\_punish`(max\_keeping\_days INT)/\*更新处罚记录过程\*/

BEGIN

/\*输入最大借阅天数，更新\*/

DECLARE

\_borrow\_record\_id INT;

DECLARE

startdate DATE;

DECLARE

enddate DATE;

DECLARE

num INT;

DECLARE

done INT DEFAULT 0;

DECLARE

num1 INT;

DECLARE

cur CURSOR FOR SELECT

borrow\_record\_id,

start\_date,

end\_date

FROM

borrow\_record;

DECLARE

CONTINUE HANDLER FOR NOT found

SET done = 1;

/\*指定游标循环结束时的返回值 \*/

OPEN cur;

label :/\*遍历借阅记录\*/

LOOP

FETCH cur INTO \_borrow\_record\_id,

startdate,

enddate;

SELECT

COUNT(\*) INTO num

FROM

punish\_record;

SELECT

COUNT(\*) INTO num1

FROM

punish\_record

WHERE

borrow\_record\_id = \_borrow\_record\_id;/\*判断该条记录是否已经加入违规记录\*/

SET num = num + 1;

IF

( num1=0&&enddate IS NULL && DATEDIFF( CURRENT\_DATE (), startdate )> max\_keeping\_days ) THEN/\*如果该条记录超过最大借阅期且未被收录\*/

INSERT INTO punish\_record ( punish\_record\_id, borrow\_record\_id )

VALUES

( num, \_borrow\_record\_id );

END IF;

IF

done = 1 THEN

LEAVE label;

END IF;

END LOOP label;

CLOSE cur;

END

--------------------------------------------------------**-恢复借阅权利过程**---------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `recover\_rights`( id INT )/\*恢复借阅权利过程\*/

BEGIN

DECLARE

bookid INT;

DECLARE

readerid INT;

DECLARE

startdate DATE;

SELECT

book\_id INTO bookid

FROM

punish\_record,

borrow\_record

WHERE

punish\_record.borrow\_record\_id = borrow\_record.borrow\_record\_id

AND punish\_record.punish\_record\_id = id;

SELECT

start\_date INTO startdate

FROM

punish\_record,

borrow\_record

WHERE

punish\_record.borrow\_record\_id = borrow\_record.borrow\_record\_id

AND punish\_record.punish\_record\_id = id;

SELECT

reader\_id INTO readerid

FROM

punish\_record,

borrow\_record

WHERE

punish\_record.borrow\_record\_id = borrow\_record.borrow\_record\_id

AND punish\_record.punish\_record\_id = id;

UPDATE punish\_record

SET extra\_days = DATEDIFF( CURRENT\_DATE (), startdate ) /\*计算超期天数\*/

WHERE

punish\_record\_id =id;

call return\_book( readerid, bookid );/\*调用还书过程\*/

END

### **3.2.3定时事件**

/\*定时事件\*/

CREATE EVENT punish

ON SCHEDULE EVERY 1 MINUTE/\*正常情况应是每天定时更新一次违规记录，为了方便演示一分钟更新一次\*/

DO call update\_punish(10);

# 五、系统设计

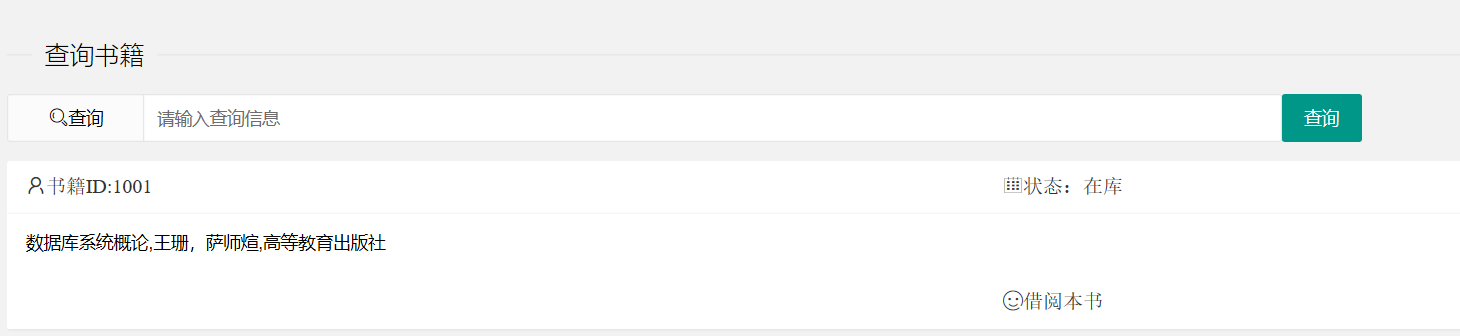
## 4.1 系统流程图

## 0P[5XE)9R1A8RNV0R0GD~AV

## 4.2 模块设计

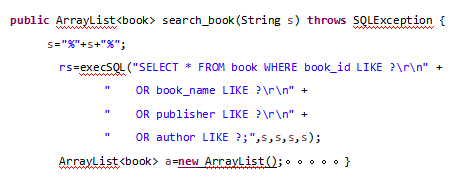
**1）图书的查询、借阅**

前端设计：



后端设计：

主要采用字符串匹配技术进行模糊查询

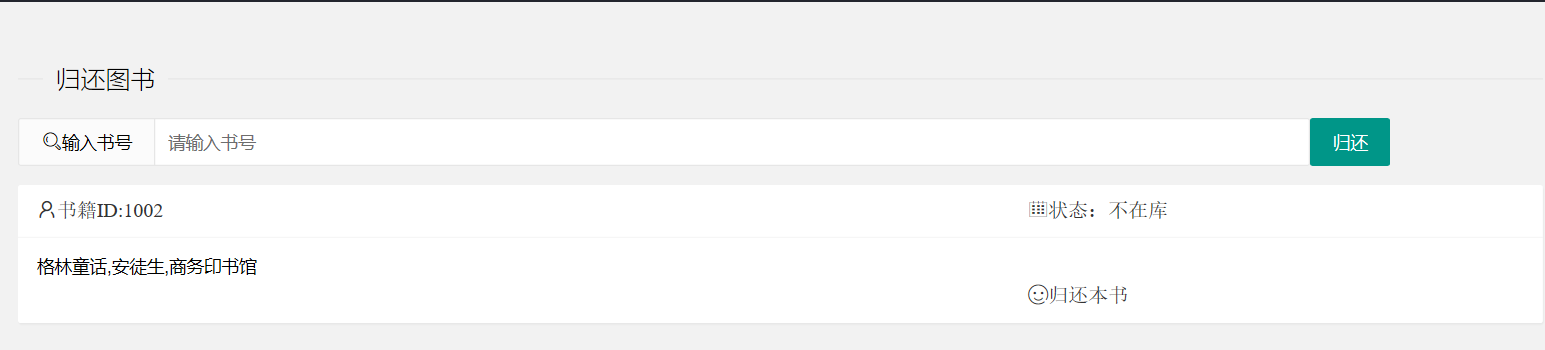


直接调用存储过程进行借书



1. **图书的归还**

前端设计：

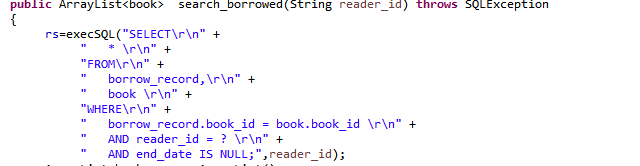


后端设计：

直接调用存储过程进行还书

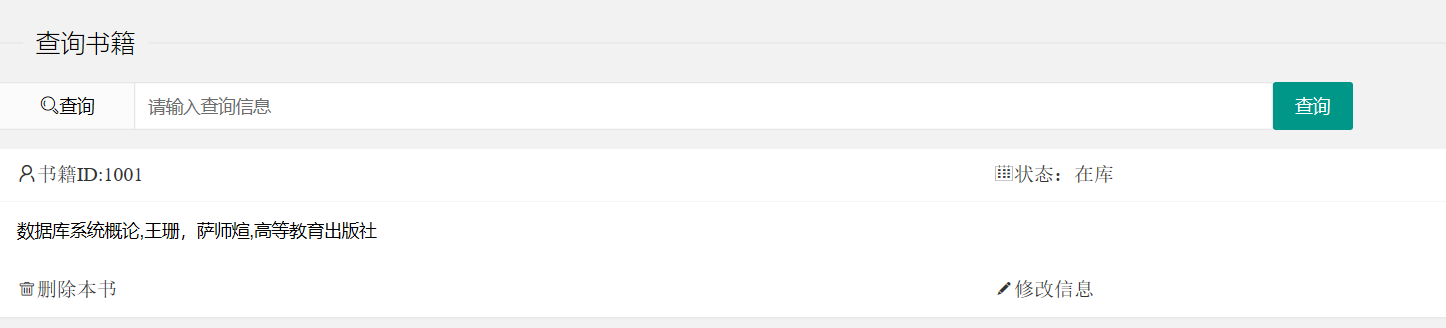


搜索读者已经借阅的图书（当借阅记录的结束日期为空时说明未归还）

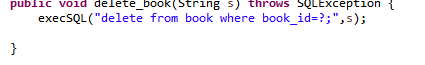


1. **图书的删除：**

前端设计：



后端设计：



1. **图书的添加**

前端设计：



后端设计：



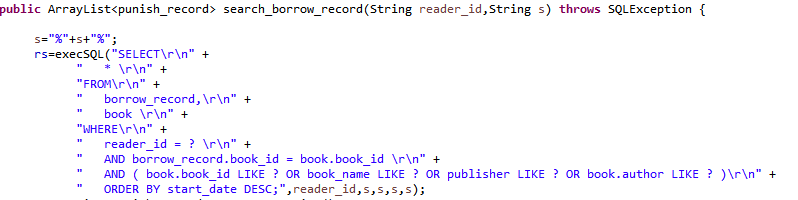
1. **图书的修改：与添加类似，不再赘述**
2. **借阅历史的查询**

前端设计：



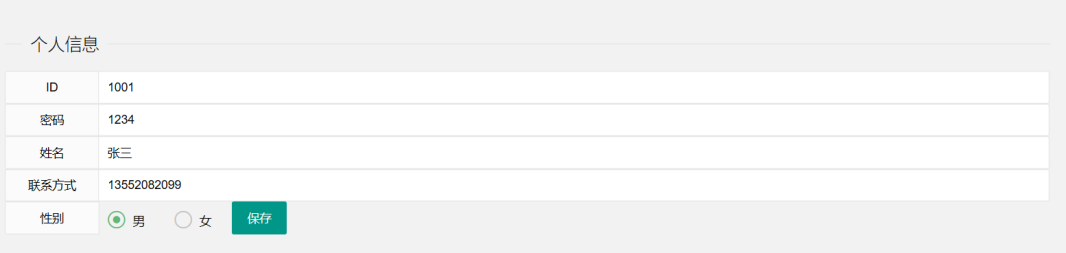
后端设计：

查询借阅记录，主要采用字符串匹配技术进行模糊查询

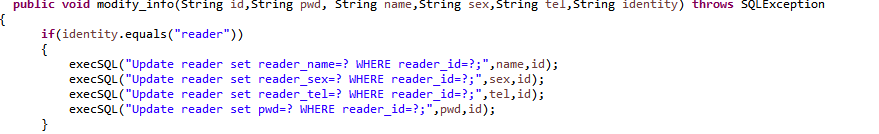


1. **个人信息的查看、修改**

前端设计：



后端设计：



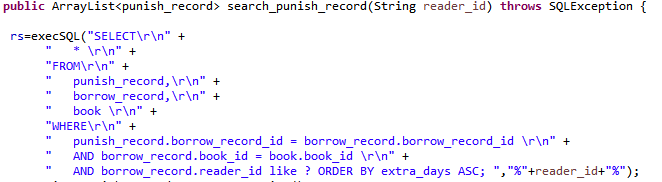
1. **违规记录的查看、处理**

前端设计：



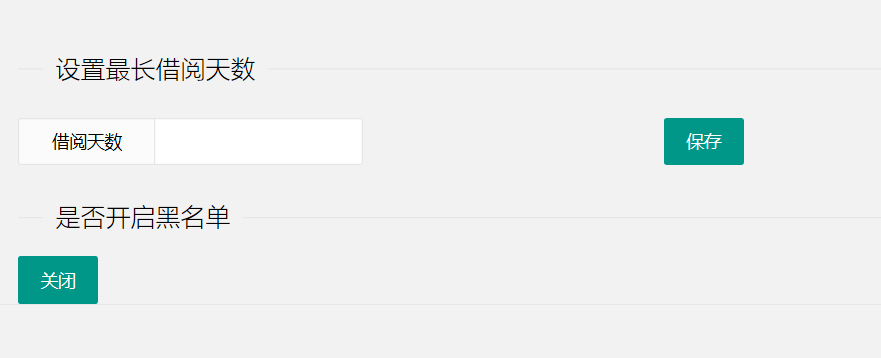
后端设计：

采用字符串匹配技术进行模糊查询



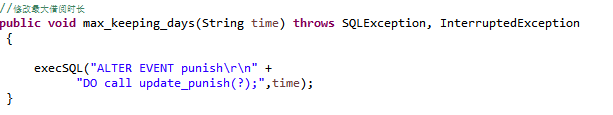
1. **权限管理**

前端设计：

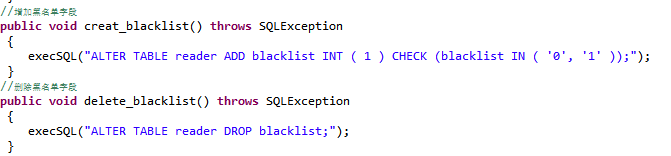


后端设计：

修改最长借阅期，即修改定时事件的工作，修改存储过程的传入参数；



开启或关闭“黑名单”功能：对reader表进行字段的删除与添加



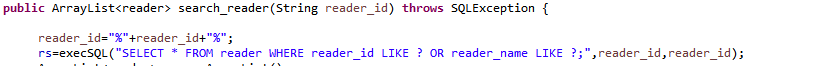
1. **读者管理**

前端设计：

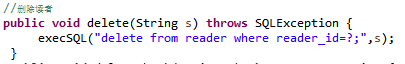


后端设计:

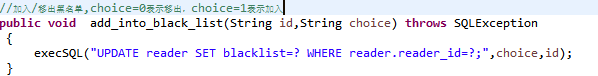
搜索用户，利用读者id或者姓名进行模糊查询



删除用户



加入/移出黑名单



1. **注册**

前端设计：



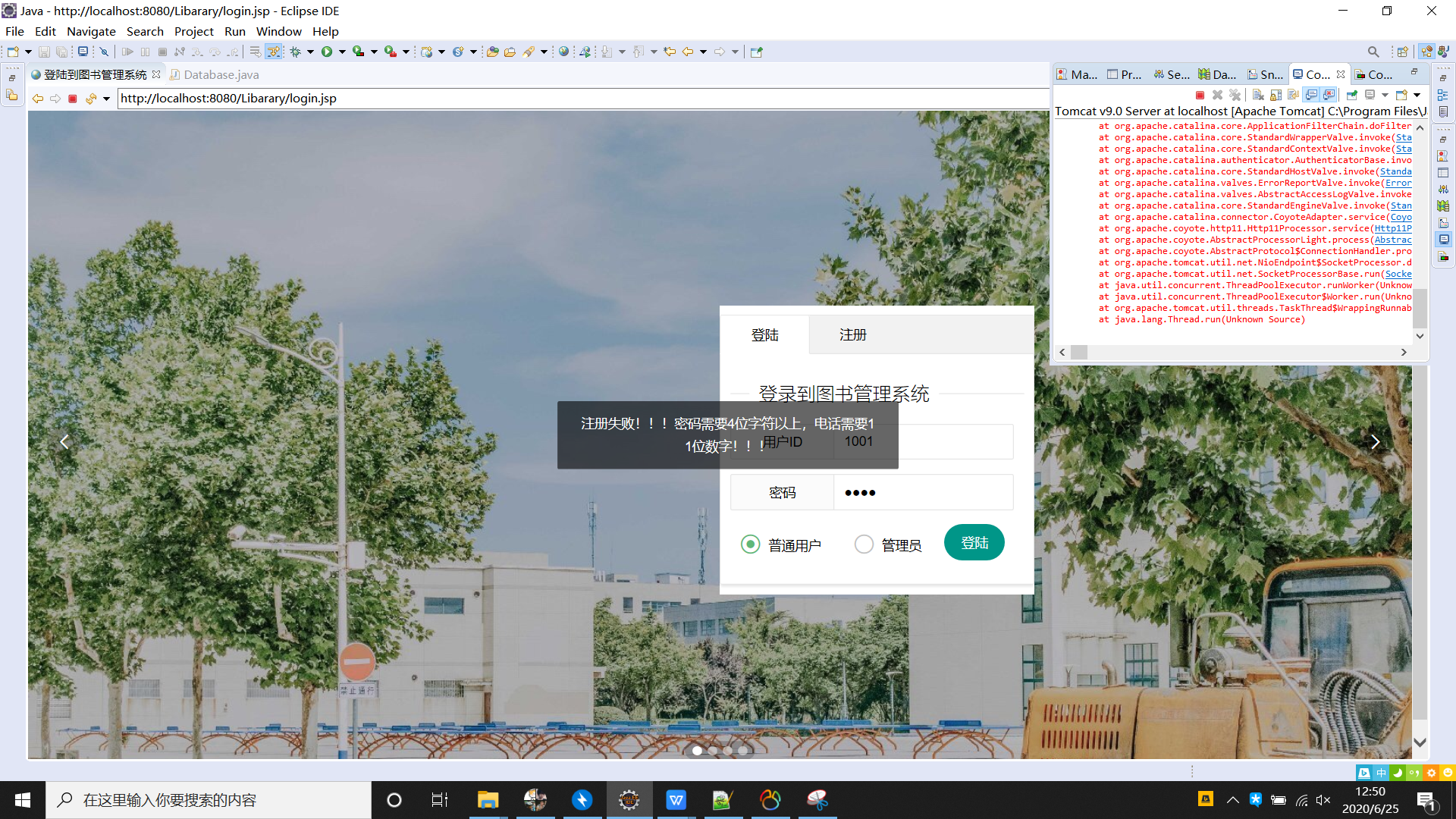
# 后端设计：

# 

1. **异常管理：**

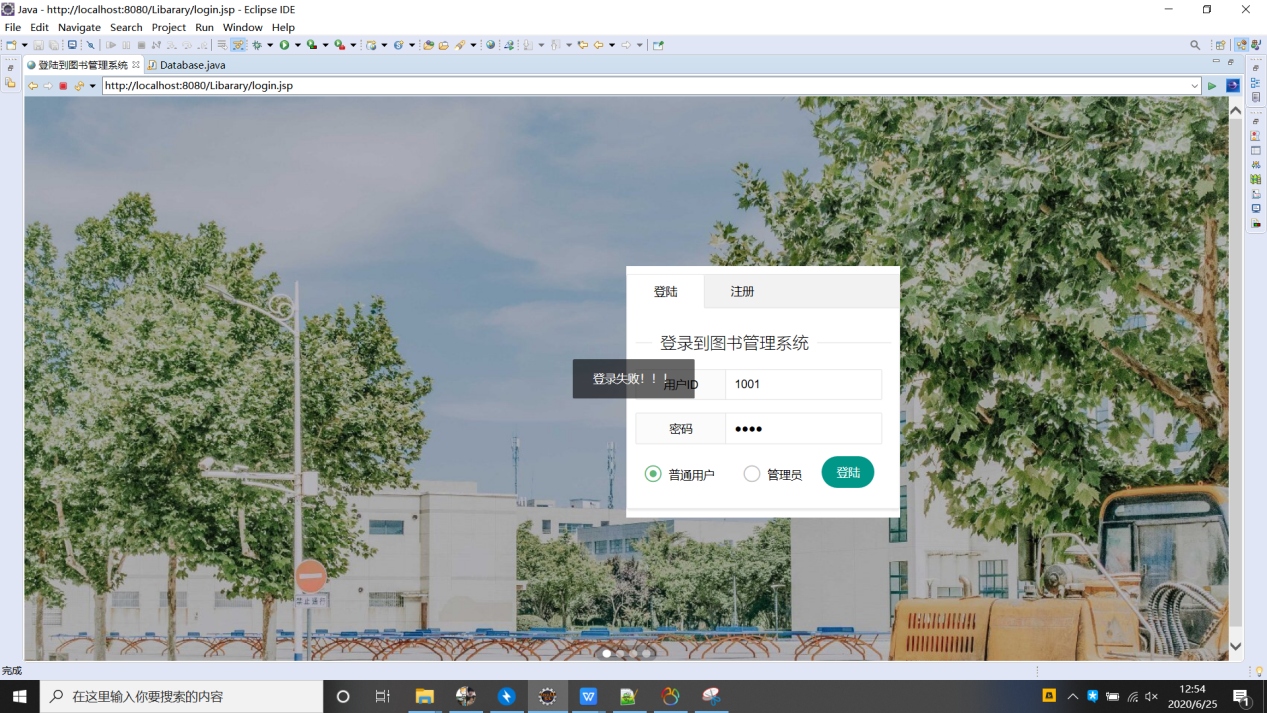
前端设计：

注册密码位数少于4位、电话位数为满足11位时提示“注册失败”；

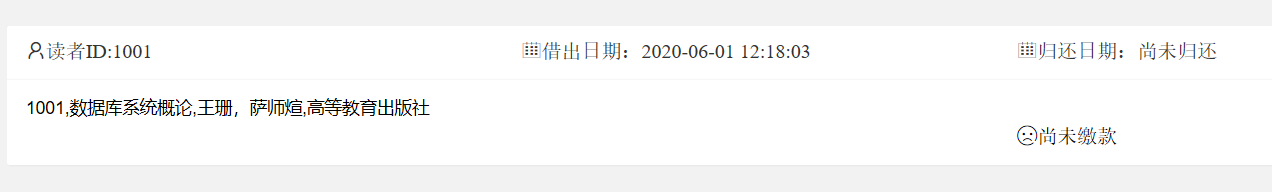


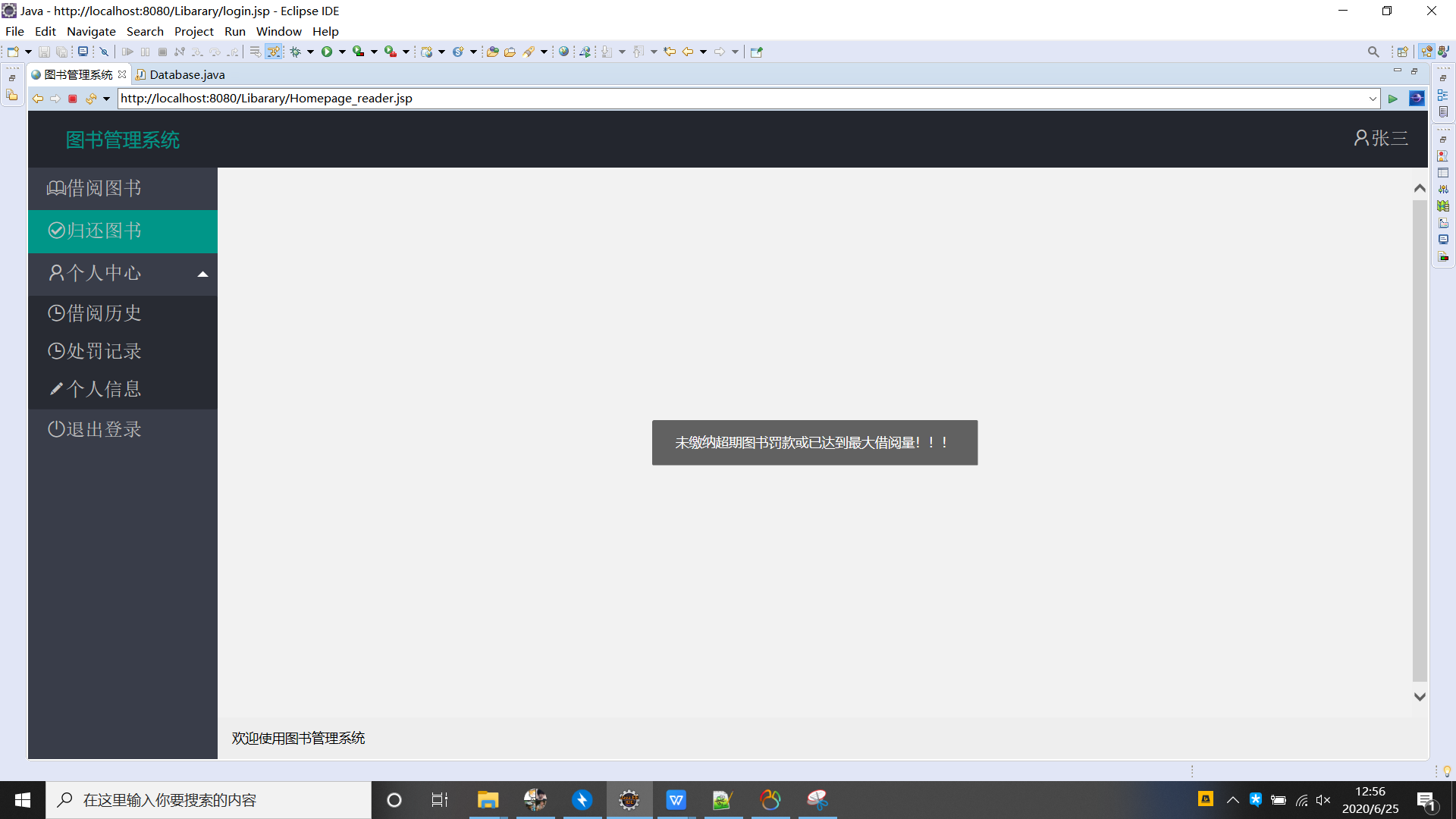
黑名单用户的“登录失败”提示；





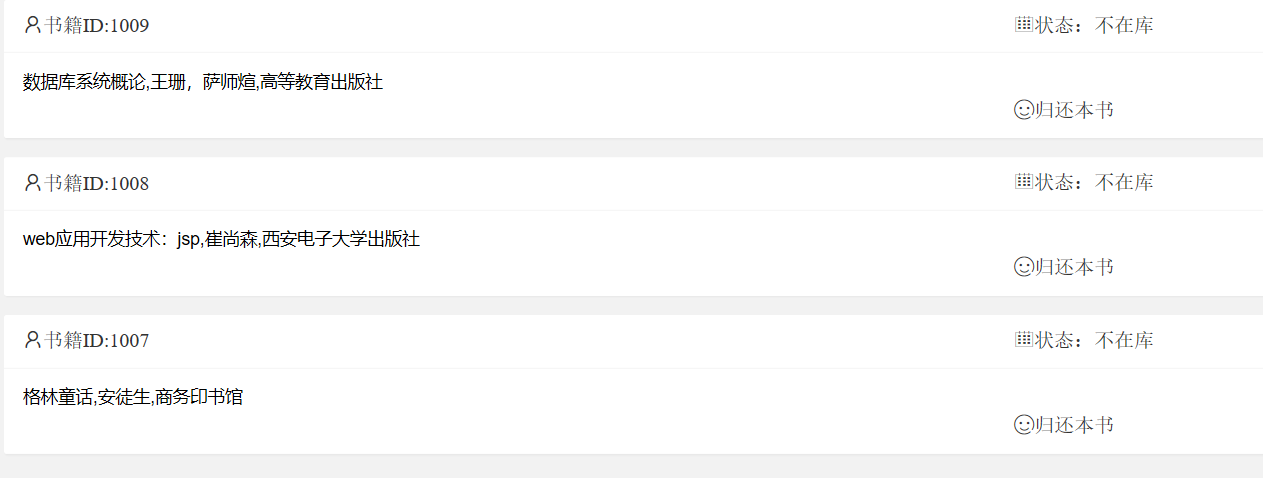
未缴罚金，还书失败时的提示；



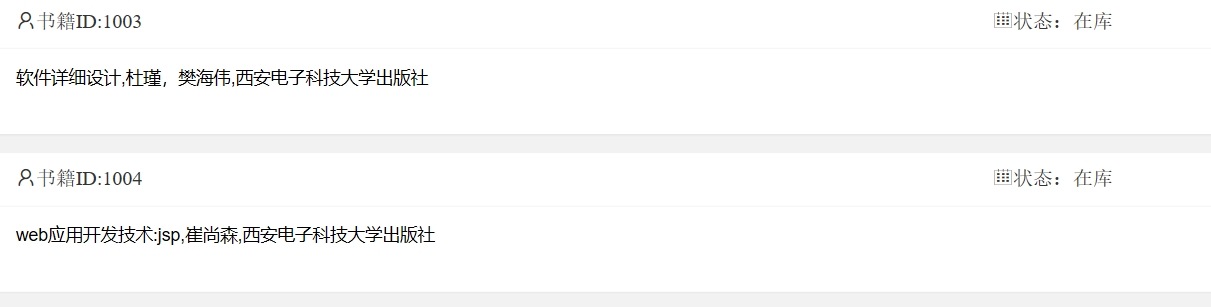


达到最大借阅量，无法继续借阅；

已经借阅三本；

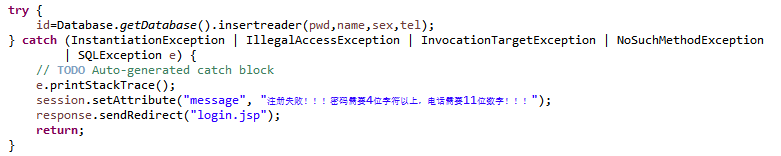


无法继续进行借阅；



后端设计：

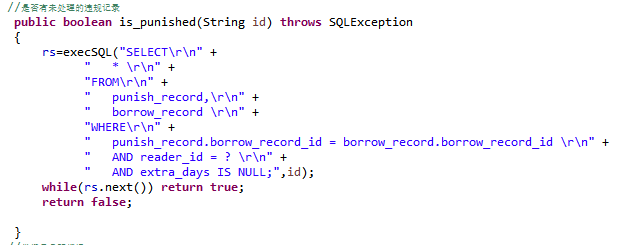
提示“注册失败”，获取抛出的异常，将错误信息传送到前端；



检查当前用户是否是黑名单用户；



检查当前用户是否有未缴纳的罚金；



检查当前用户是否已经达到最大借阅量；

