**图书馆管理系统借还模块详细设计**

一、概述

1、借阅

借阅的动作仅发生在查询图书后，借阅选定的图书。借阅不能单独作为一个功能，需要依赖查 询功能共同实现。

2、归还

归还图书动作可以发生在任何时间，当读者存在一本已借阅的图书，可以随时请求归还图书 （无论是否超期、是否为当天、是否欠费和锁定）。归还图书后，系统再进行是否超期等判断。

3、查询图书信息

查询图书信息功能属于信息模块内容，不再设计。

二、算法设计

1、借阅

在查询到一个图书的信息后，点击借阅，系统将判断该读者是否符合借阅图书的条件，如符合 则跳转到借阅工作，如不符合则返回相应的错误信息。借阅时在在库的图书中选取流水号 （BookNo）最小 的图书借阅。借阅后，更新图书信息，将图书的状态更改为“已借阅”，更新图书应 还日期；更新数据库借阅表信息，添加一条借阅记录；更新读者信息，读者在借图书数量+1。

2、归还

在读者请求归还图书后，系统首先生成一条图书归还记录，记录到数据库归还表中。然后，系 统比对应还日期，如不超期，则归还成功，返回成功提示，并跳转到归还页面。如超期，系统计算 罚款，更新该读者的罚款金额，并返回超期归还提示和本次罚款金额、总计罚款金额，并跳转到归 还页面。

三、实现

1、借阅

首先，读者需先登录才能查询图书从而借阅。当读者查询图书后，读者应点击该书的详细浏览 页面才能看到借阅按钮。当读者点击借阅按钮时，系统首先检测读者状态，如为锁定状态，则无法 借阅，否则进入下一步。下一步，系统检测读者再借图书数量，数量如大于等于允许借阅图书的最 大数量，则无法借阅，否则进入下一步。下一步，系统检测读者是否有超期未归还的图书，如有， 则无法借阅，否则进入下一步。下一步，系统检测读者罚款是否超过约定的限度，如超过，则无法 借阅，并将用户状态改为锁定状态，否则，进入下一步。下一步，借阅成功，系统将在查询到的图 书选取第一个书籍编号（理论上为最小的编号）为用户办理借阅，该书状态更新为已借阅，更新该 书的应还日期，更新用户在借图书数量，在数据库借阅表中添加该书借阅的一条记录。

具体算法实现：

/\*\*

Borrow a book

JSP file

BLL

Author:RoMan

\*/

boolean borrow(Reader Reader, Book Book)

{

//check login status

boolean flag = true;

if(Reader.isFreeze())

flag = false;

if(Reader.getBorrowNum() > Reader.getAllowBorrowNum())

flag = false;

if(Reader.hasOverdue())

flag = false;

if(Reader.getPanalty() > LMS.maxPanalty)

{

Reader.setStatus(READER\_FREEZE);

flag = false;

}

if(flag)

{

Book.setStatus(BOOK\_LENT);

Book.setDueTo(/\*current time+MAX\_TIME\_OF\_BORROW\*/);

Reader.setBookInBorrow(Reader.getBookInBorrow()+1);

//Add a record in Table Borrow. Including specific message.

return true;

}

else

{

//Show error message to USL

return false;

}

}

2、归还

首先，读者应先登录才能使用本功能。读者在点击归还按钮后，应该跳转到读者现有在借书籍 列表的页面。读者可以选择任意一本书归还。在第一次迭代中，我们假定每次只允许归还一本书。 在执行归还功能时，首先，检查图书是否超期，如超期，则在用户现有罚款中按照约定的罚款计算 规则进行罚款，然后进行下一步，否则，直接进入下一步。下一步，完成归还动作，读者在借图书 数量-1，图书状态更新为在库，在数据库归还 列表中添加一项记录，为本书的归还信息。

具体算法实现：

USL\_Return:

/\*\*

Return a book

JSP file

USL

Author:RoMan

\*/

<%

//check login status

//get book in borrow from database

//show on screen

%>

BLL\_Return:

/\*\*

Return a book

JSP file

BLL

Author:RoMan

\*/

boolean ReturnAction(Reader Reader,Book Book)

{

//check login status

if((long)Integer.toInteger(CURRENT\_TIME) > (long)Integer.toInteger(Book.getDueTo()))

Reader.setPanalty(Reader.getPanalty() + /\*calculate new panalty to add\*/);

Reader.setBookInBorrow(Reader.getBookInBorrow() - 1);

Book.setStatus(BOOK\_IN\_LIBRARY);

//Add a record to Table Return

}