

数据库课程设计实验报告

系统名称：图书馆管理系统

学生姓名：马奔腾

班 级：网络 063 班

学 号：0615210184

日 期：2009.04.30

系统需求说明书

1. 需求引言

学生图书管理系统需求说明书，是在采集大量的图书管理资料及模拟数据的基础上进行编写的。

为了更好地适应当前高校图书管的管理需求，缓解手工管理存在的弊端，开发了学校图书管理系统。

2. 需求目的

1. 在软件产品完成目标方面为用户和开发者之间建立共同协议创立一个基础：对要实现的软件功能进行全面描述，帮助用户判断所规定的软件是否符合他们的要求。
2. 提高开发效率：使用户在设计开始之前周密地思考全部需求，从而减少事后重新设计、重新编程和重新测试的返工情况。通过对各种需求进行仔细复查，在开发早期发现若干遗漏、错误的理解和不一致性，以便及时加以纠正。
3. 为了更好地适应当前高校图书管的管理需求，缓解手工管理存在的弊端。

3. 需求背景

随着高校内部网络的广泛建立，在通用信息平台上构筑高效实用的协同工作和学习交流，满足信息高度共享和即时发布的需求，有效实现内部知识管理，已成为广大师生的共同需求。

4. 目标

对图书信息进行图书入库、借阅登记、借阅查询、过期提示、办理借书卡这些日常业务的需要。高效性通过系统的实用，能够改变旧的图书管理模式，使对图书的管理、借阅、查询更加方便有效。

5. 用户的特点

1. 可以存储学校图书馆所有图书的资料以及库存情况，具有安全，高效的特性。
2. 只需一到二名档案录入员即可操作系统，节省大量人力。
3. 可以迅速查到所需信息。
4. 系统对软，硬件方面的需求不高，软件要求易学，容易掌握，可以很简单方便地管理图书信息。
5. 具有严格的系统使用权限管理；
6. 能及时了解图书借阅的情况。

6. 假定和约束

1. 假定：

1. 工作人员通过计算机来管理各类图书，进行图书分类编号，调整图书结构，增加图书库存，适应用户的需求。
2. 工作人员对读者的需求情况能作好全面的掌握，及时得到历史记录。
3. 也是书店的核心工作，对图书的借阅管理。

2. 约束：在图书管理系统相关的表之间，有较强的关联性，不能达到一致性的要求。不适合大型的图书馆。

7. 对功能的规定与设计图

基于系统需求分析，该系统需要实现以下基本功能：

1. 新书入库
2. 新读者注册
3. 资料修改
4. 查询
5. 统计

6. 一个读者可以同时借 5 本书
7. 每本借阅时间为半年
8. 过期按书价的千分之一/天罚款

9. 借书
10. 还书
11. 对借阅数据进行查询和统计

系统概要设计报告

1. 数据库需求分析

根据系统功能设计的要求以及功能模块的划分，对于本系统的数据库，可以列出以下数据项和数据结构：

1. 图书信息表 books

| 字 段 名 称 | 类 型 | 主 键 | 非 空 |
|---------|-----|-----|-----|
| 图书编号 | 文本 | 是 | 是 |
| 图书名称 | 文本 | 否 | 否 |
| 图书作者 | 文本 | 否 | 否 |
| 图书出版社 | 文本 | 否 | 否 |
| 图书价格 | 文本 | 否 | 否 |
| 购买日期 | 文本 | 否 | 否 |
| 图书类别 | 数字 | 否 | 否 |
| 内容介绍 | 文本 | 否 | 否 |
| 图书状态 | 数字 | 否 | 是 |

2. 用户信息表 login

| 字段名称 | 类 型 | 主 键 | 非 空 |
|------|-----|-----|-----|
| 用户名 | 文本 | 否 | 是 |
| 密码 | 文本 | 否 | 否 |
| 权限 | 是/否 | 否 | 是 |

3. 借阅者 borrower

(3)借阅者 borrower

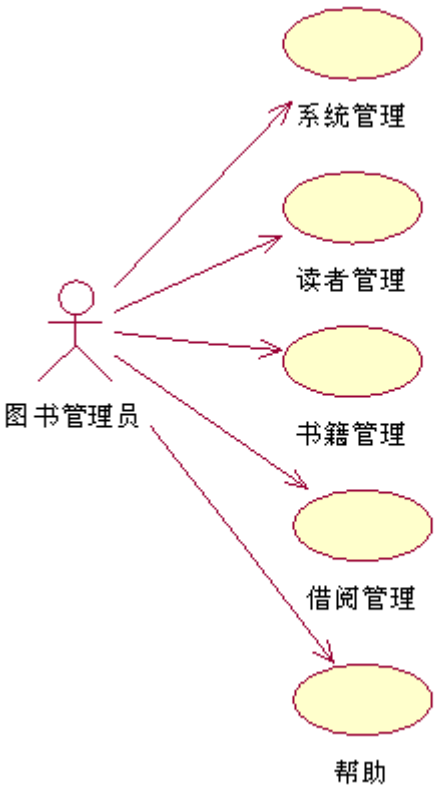
| 字 段 名 称 | 类 型 | 主 键 | 非 空 |
|---------|------|-----|-----|
| 借阅者编号 | 自动增长 | 是 | 是 |
| 借阅者名称 | 文本 | 否 | 是 |
| 借阅者电话 | 文本 | 否 | 是 |
| 借阅者地址 | 数字 | 否 | 是 |
| 备注 | 文本 | 否 | 是 |

4. 借阅图书表 borrowbook

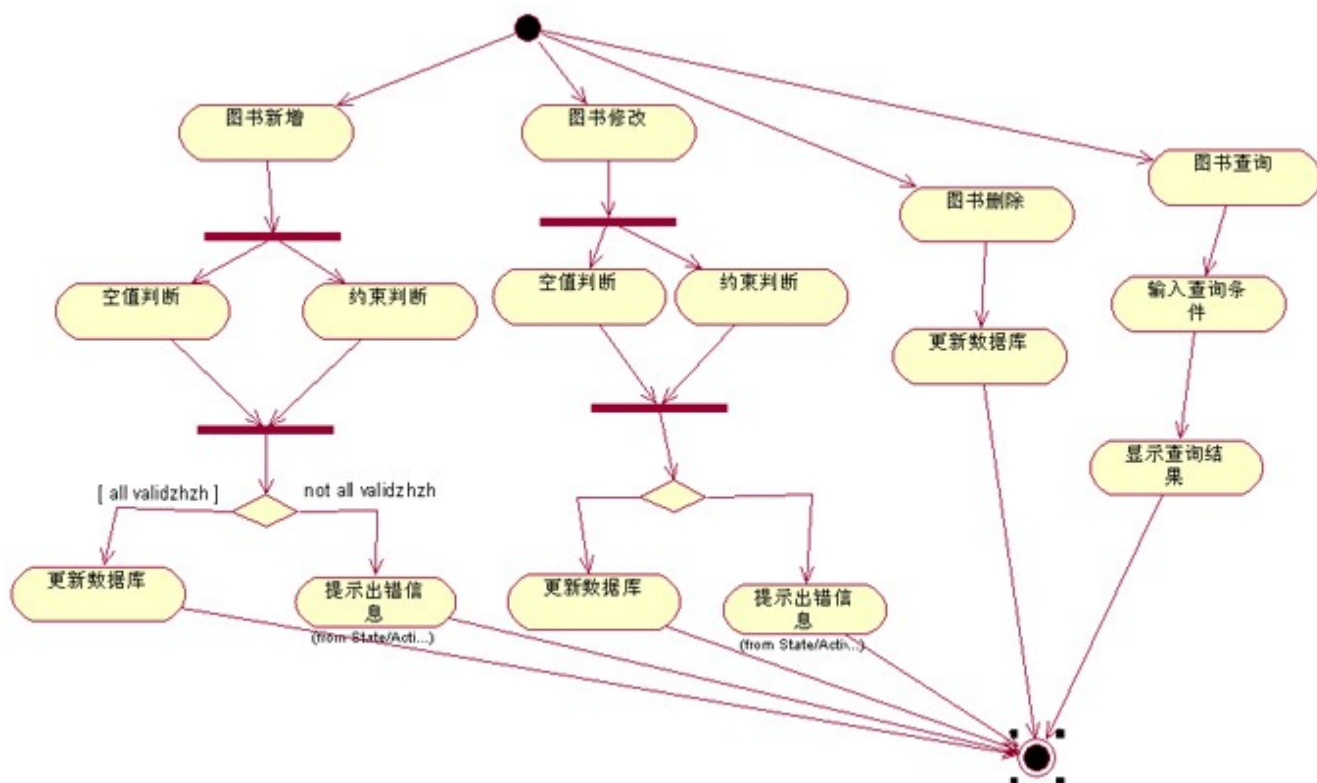
| 字 段 名 称 | 类 型 | 主 键 | 非空 |
|---------|-----|-----|----|
| 借阅者编号 | 时间型 | 否 | 是 |
| 读者姓名 | 自动增 | 否 | 是 |

| | | | |
|--------|------|---|---|
| | 长 | | |
| 借阅者电话 | 自动增长 | 是 | 是 |
| 借阅者住址 | 文本 | 否 | 是 |
| 借阅图书编号 | 文本 | 否 | 是 |
| 借阅日期 | 文本 | 否 | 是 |
| 备注 | 文本 | 否 | 是 |

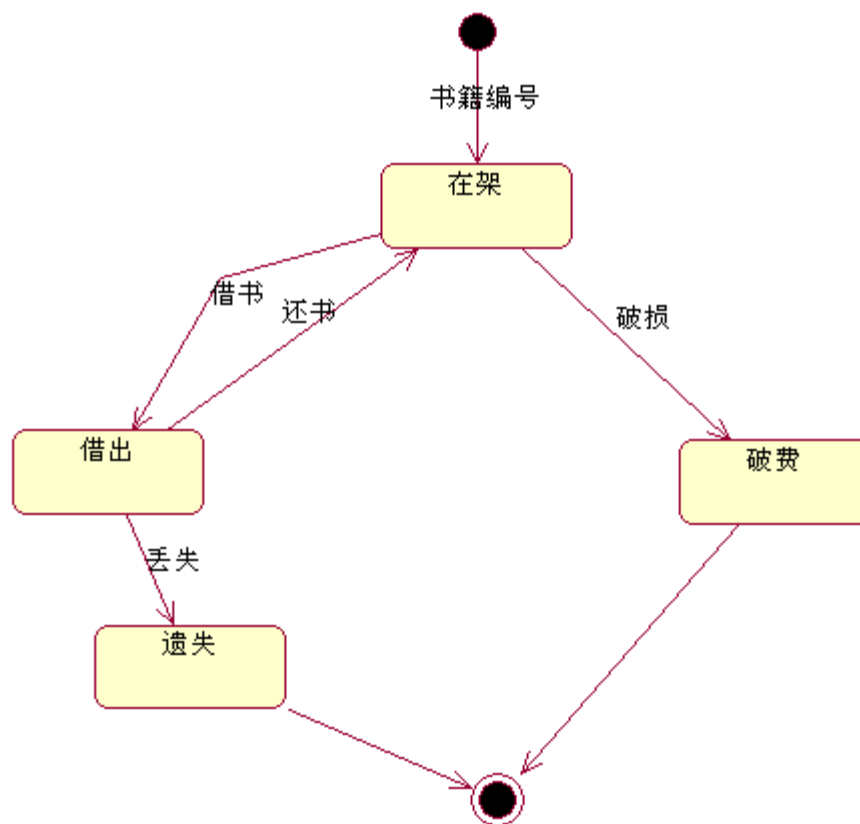
系统流程设计



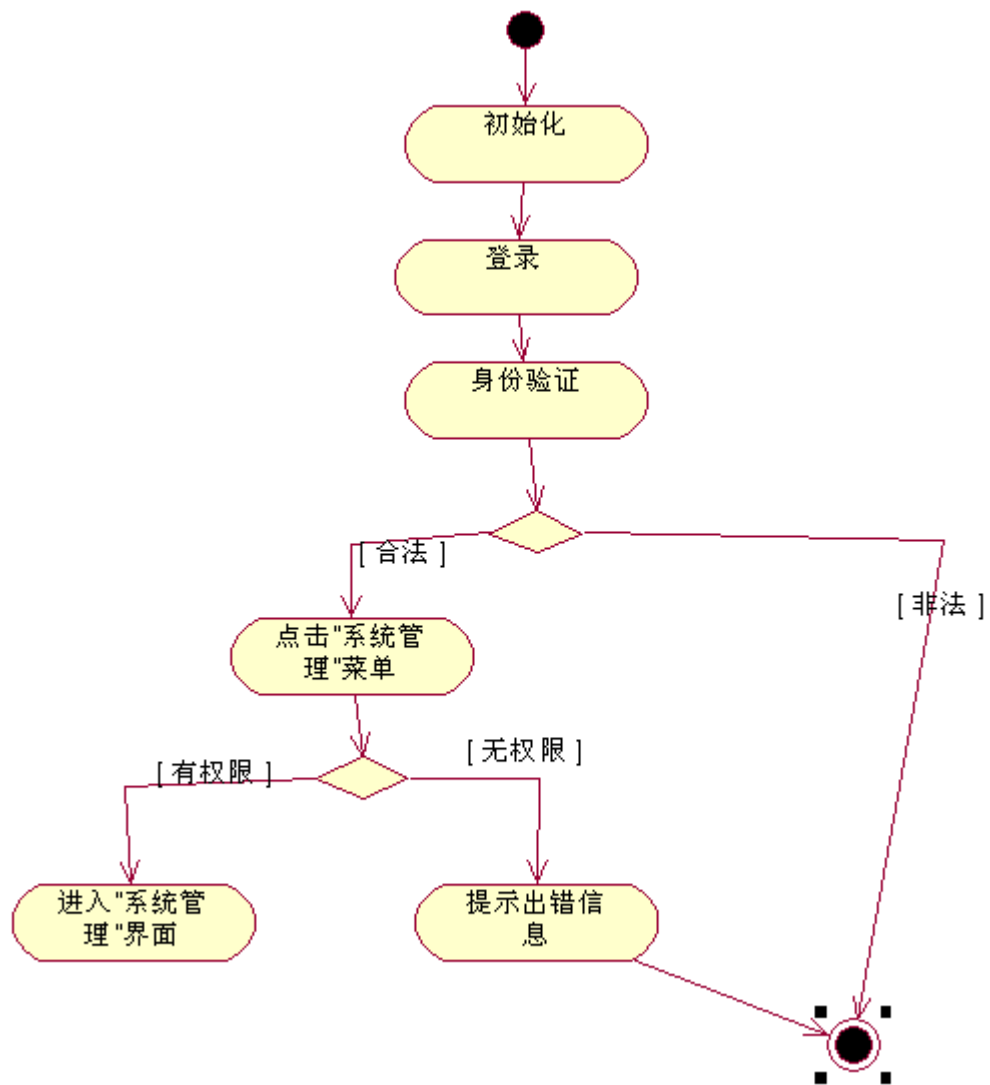
系统功能分析图



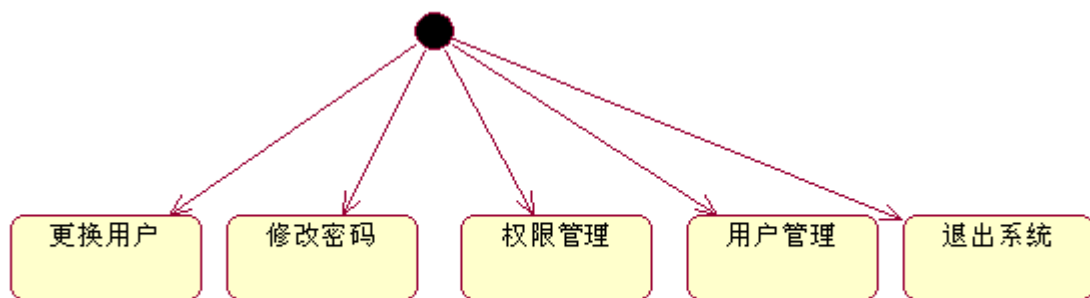
图书管理活动图



图书状态图



系统登录活动图



系统管理状态图

用户使用说明书

系统功能简介

1. 基本信息
2. 系统管理：系统用户管理和用户权限的管理。
3. 读者管理：对读者的基本信息进行管理，包括新增，修改，删除和各种查询。
4. 书籍管理：对各种图书进行管理，包括图书入库、遗失等。
5. 借阅管理：对图书进行借阅登记、借阅查询。
6. 帮助：帮助用户较好地使用该系统,调出用户使用手册。

辅助功能

1. 关于：显示软件授权及版本信息。
2. 退出系统：操作员操作完毕后，关闭系统。
3. 重新登录：重新登录，切换用户。
4. 进度条：美化界面。
5. 帮助文档：帮助用户使用该系统。

系统特点

1. 简单友好的用户界面；
2. 简化实用的业务操作和强大的查询功能；
3. 数据安全性；
4. 系统运行的可靠性、稳定性、准确性；
5. 可维护性、可扩展性、可适应性、开放性和通用性；

系统设置流程

为了使系统正常有效地运行，应当在首次登录系统的前后遵循以下步骤进行设置：

【登录本系统前】

第一步：确定用户名是否已存在

第二步：添加新用户

第三步：用已存在的用户登陆

【登录本系统后】

第三步：登陆系统后，可以看到主菜单页面。

第四步：运行该管理系统各项管理功能。

系统应用前准备

2.1 系统网络结构

用于单机版

2.2 系统硬件配置

工作站应尽量选择性能稳定可靠的品牌。在选型时应考虑硬件配置的兼容性。

2.4 正式登录系统

首先将系统安装好，在保证已设置好系统的基本参数，软硬件能正常运行的情况下，可以正式登录系统开始系统初始设置和业务处理。双击计算机显示器桌面上的特定图标，运行系统程序，出现如下登录窗口，要求输入系统中预先分配的操作员、口令（输入工号后可直接敲击回车键，不必输入姓名），单击“确定”按钮登录系统。

3.1 用户管理

该功能控制了用户的密码和权限，管理者要正确使用自己的密码和权限范围。添加用户时，先点击添加按钮，之后在信息框中输入相应的信息，点击保存即。注意：用户名最长 20 个字符，密码最长 24 个字符。请注意密码输入时的大小写。删除用户时，在列表中选择你需要删除的用户，然后点击删除按钮即可。提示按钮提示用户帮助信息。点击关闭按钮，即可关闭用户管理窗体。

图 3-1：用户管理界面

3.2 重新登陆

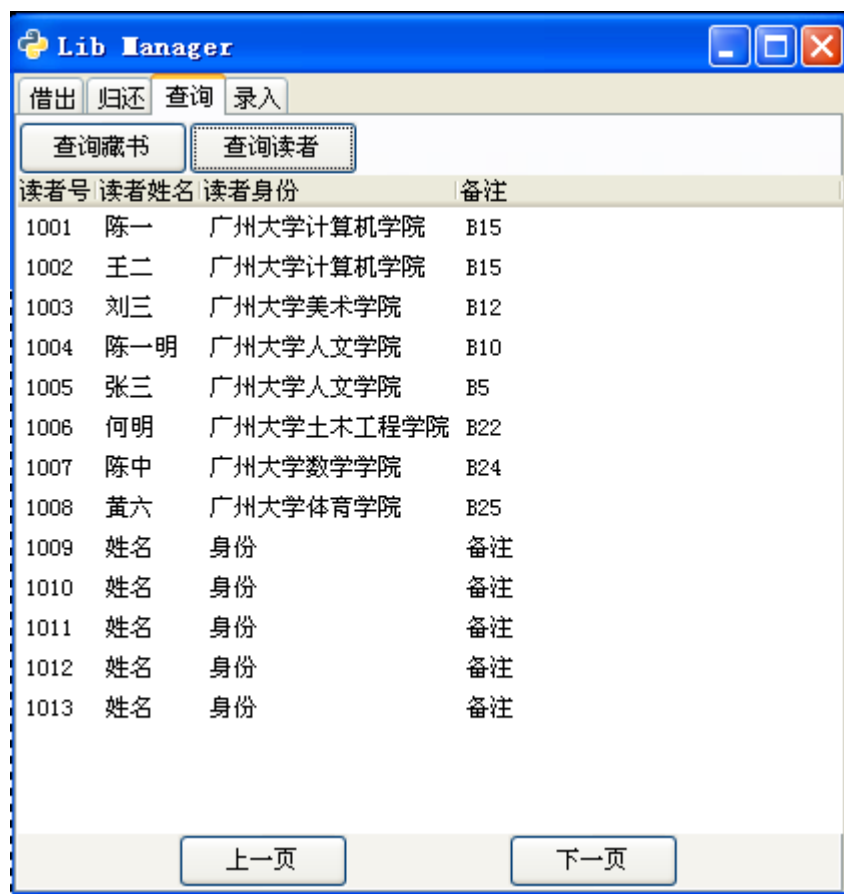
该功能能切换用户。该功能要先回到登陆界面。在登陆界面输入重新登陆用户的用户名及密码，点击确定及切换到重新登陆的用户。点击取消及可退出该窗体。

3.3 关闭系统

点击该项退出系统

第 3 章读者管理

该功能详细记录了读者的详细信息。添加用户时，先点击添加按钮，之后在信息框中输入相应的信息，点击保存即可。



查找借阅者信息界面

点击关闭按钮退出读者信息对话框。

第 4 章书籍管理

5.1 图书浏览

该功能记录了书籍的状态。点击全部图书按钮及显示所有图书信息，



The screenshot shows a window titled "Lib Manager" with a menu bar containing "借出", "归还", "查询", and "录入". Below the menu bar are two buttons: "查询藏书" and "查询读者". The main area displays a table of books with the following columns: 书本号, 书名, 作者, 出版社, 价格, 购入日期, 分类, 简介, and 馆藏数. The table contains 8 rows of data. At the bottom of the window are two buttons: "上一页" and "下一页".

| 书本号 | 书名 | 作者 | 出版社 | 价格 | 购入日期 | 分类 | 简介 | 馆藏数 |
|-----|---------|-----|-------|-----------|------------|-----|----|-----|
| 1 | C语言程序设计 | 谭浩强 | 教育出版社 | 29.900000 | 2009-04-30 | 计算机 | | 5 |
| 2 | 数据结构 | 谭浩强 | 教育出版社 | 29.900000 | 2009-04-30 | 计算机 | | 5 |
| 3 | 数据库原理 | 谭浩强 | 教育出版社 | 29.900000 | 2009-04-30 | 计算机 | | 5 |
| 4 | 汇编语言 | 谭浩强 | 教育出版社 | 29.900000 | 2009-04-30 | 计算机 | | 5 |
| 5 | 大学国文 | 国立 | 教育出版社 | 19.900000 | 2009-04-30 | 文学 | | 5 |
| 6 | 古代诗词 | 国立 | 教育出版社 | 19.900000 | 2009-04-30 | 文学 | | 5 |
| 7 | 广州地图 | 工作 | 广州出版社 | 9.900000 | 2009-04-30 | 工具书 | | 5 |
| 8 | 牛津词典 | 词典 | 广州出版社 | 9.900000 | 2009-04-30 | 工具书 | | 5 |

全部图书信息界面

点击“查询图书”显示尚未借出的图书信息，

5.2 新书添加

该功能可以增加图书，将书入库。添加新书信息时，先点击添加新书按钮，之后在信息框中输入相应的信息，点击保存记录即可。点击关闭按钮及退出新书添加对话框。

Lib Manager

借出 归还 查询 录入

录入书本 录入读者

| 书名 | 作者 | 出版社 | 价格 | 购入日期 | 分类 | 简介 | 馆藏数 |
|----|----|-----|----------|------------|----|----|-----|
| 书名 | 作者 | 出版社 | 0.000000 | 2009-05-11 | 分类 | 简介 | 0 |
| 书名 | 作者 | 出版社 | 0.000000 | 2009-05-11 | 分类 | 简介 | 0 |
| 书名 | 作者 | 出版社 | 0.000000 | 2009-05-11 | 分类 | 简介 | 0 |
| 书名 | 作者 | 出版社 | 0.000000 | 2009-05-11 | 分类 | 简介 | 0 |
| 书名 | 作者 | 出版社 | 0.000000 | 2009-05-11 | 分类 | 简介 | 0 |
| 书名 | 作者 | 出版社 | 0.000000 | 2009-05-11 | 分类 | 简介 | 0 |

新增条目 删除条目 录入提交

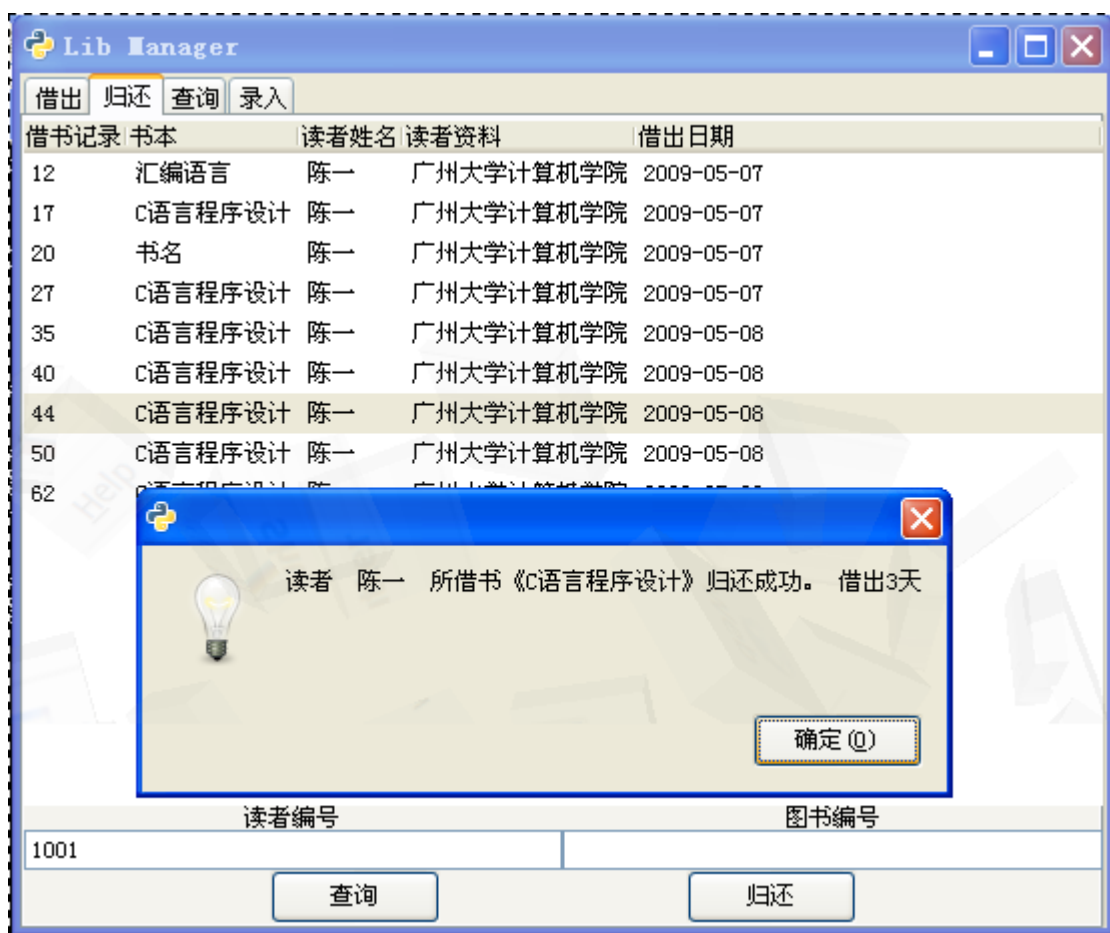
添加新书界面

第 5 章 借阅管理

6.1 图书归还

该功能查看已借图书的归还状态。

查找图书功能：点击“查找图书”按钮在出现的对话框中输入图书编号，按回车键或“确定”按钮，在归还图书信息栏里显示相应的图书信息。



图：图书归还过程（1）

查找读者功能：点击“查找读者”按钮在出现的对话框中输入图书编号，按回车键或“确定”按钮，在归还图人信息栏里显示相应的读者信息。

6.2 图书借阅



该功能查看已借图书的借阅状态。

查找图书功能：点击“查找图书”按钮在出现的对话框中输入图书编号，按回车键或“确定”按钮，在借阅图书信息栏里显示相应的图书信息。

附件：系统主要实现代码

实现语言：Python

图形框架：GTK+ 2.0

数据库框架：SQLite 3.0

代码一：数据库初始化过程

```
#!/usr/bin/python
```

```
#encoding:utf-8
```

```
import sqlite3
```

```
def LibDbInit(con):
```

```
    c = con.cursor()
```

```
    c.execute("""
```

```
CREATE TABLE `books` (
```

```
`BookID` INTEGER PRIMARY KEY,
```

```
`BookName` TEXT ,
```

```
`BookAuthor` TEXT ,
```

```
`BookPublish` TEXT ,
```

```
`BookPrice` FLOAT ,
```

```
`BookDate` TEXT ,
```

```
`BookType` TEXT ,
```

```
`BookContent` TEXT ,
```

```
`BookState` INTEGER
```

```
)
```

```
""")
```

```
c.execute("""
```

```
CREATE TABLE `reader` (
```

```
`ReaderID` INTEGER PRIMARY KEY,
```

```
`ReaderName` TEXT ,
```

```
`ReaderSchool` TEXT ,
```

```
`ReaderInfo` TEXT
```

```
)
```

```
""")
```

```
c.execute("""
```

```
CREATE TABLE `borrow` (
```

```
`BorrowID` INTEGER PRIMARY KEY,
```

```
`ReaderID` INTEGER ,
```

```
`BookID` INTEGER ,
```

```
`BorrowDate` TEXT
```

```
)
```

```
""")
```

```
books = ((1, u"C 语言程序设计", u"谭浩强", u"教育出版社", 29.90, u"2009-04-30",  
u"计算机", u"", 5),
```

```
(2, u"数据结构", u"谭浩强", u"教育出版社", 29.90, u"2009-04-30", u"计算机",  
u"", 5),
```

```
(3, u"数据库原理", u"谭浩强", u"教育出版社", 29.90, u"2009-04-30", u"计算机",  
u"", 5),
```

```
(4, u"汇编语言", u"谭浩强", u"教育出版社", 29.90, u"2009-04-30", u"计算机",  
u"", 5),
```

```
(5, u"大学国文", u"国立", u"教育出版社", 19.90, u"2009-04-30", u"文学", u"",
5),
(6, u"古代诗词", u"国立", u"教育出版社", 19.90, u"2009-04-30", u"文学", u"",
5),
(7, u"广州地图", u"工作", u"广州出版社", 9.90, u"2009-04-30", u"工具书", u"",
5),
(8, u"牛津词典", u"词典", u"广州出版社", 9.90, u"2009-04-30", u"工具书", u"",
5))
```

```
for r in books:
```

```
    c.execute("INSERT INTO books VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)", r)
```

```
readers = ((1001, u"陈一", u"广州大学计算机学院", u"B15"),
(1002, u"王二", u"广州大学计算机学院", u"B15"),
(1003, u"刘三", u"广州大学美术学院", u"B12"),
(1004, u"陈一明", u"广州大学人文学院", u"B10"),
(1005, u"张三", u"广州大学人文学院", u"B5"),
(1006, u"何明", u"广州大学土木工程学院", u"B22"),
(1007, u"陈中", u"广州大学数学学院", u"B24"),
(1008, u"黄六", u"广州大学体育学院", u"B25"))
```

```
for r in readers:
```

```
    c.execute("INSERT INTO reader VALUES (?, ?, ?, ?)", r)
```

```
borrowers = ((1, 1001, 1, u"2009-04-20"),
(2, 1001, 2, u"2009-04-20"),
(3, 1002, 3, u"2009-04-20"),
(4, 1003, 5, u"2009-04-20"),
```

```
(5, 1004, 4, u"2009-04-20"),
(6, 1004, 2, u"2009-04-20"),
(7, 1003, 7, u"2009-04-20"),
(8, 1006, 8, u"2009-04-20"),
(9, 1007, 2, u"2009-04-20"),
(10, 1008, 8, u"2009-04-20"))
```

```
for r in borrows:
    c.execute("INSERT INTO borrow VALUES (?,?,?,?)", r)

# Save (commit) the changes
con.commit()
```

查询视图的代码实现：

```
#QueryView 查询书目
def on_queryBooks_clicked(self, button, data = None):
    self.queryBook()
    self.cursor.execute("SELECT * FROM books LIMIT ? OFFSET ?", [15, 0])
    for row in self.cursor:
        self.queryViewModel.append(row)

#查询读者
def on_queryReaders_clicked(self, button, data = None):
    self.queryReader()
```

```
self.cursor.execute("SELECT * FROM reader LIMIT ? OFFSET ?", [15, 0])
for row in self.cursor:
    self.queryViewModel.append(row)
```

#上一页翻页

```
def on_queryPrev_clicked(self, button, data = None):
```

```
    if self.queryRecord == 0:
        self.errorMessage(u"已到最前页。")
        return
```

```
    self.queryRecord -= 15
```

```
    self.queryViewModel.clear()
```

```
    if self.queryViewModel.get_n_columns() == 9:
```

```
        self.cursor.execute("SELECT * FROM books LIMIT ? OFFSET ?",[15,
self.queryRecord])
```

```
    elif self.queryViewModel.get_n_columns() == 4:
```

```
        self.cursor.execute("SELECT * FROM reader LIMIT ? OFFSET ?",[15,
self.queryRecord])
```

```
    for row in self.cursor:
```

```
        self.queryViewModel.append(row)
```

#下一页翻页

```
def on_queryNext_clicked(self, button, data = None):
```

```
    if self.queryRecord + 15 > self.maxRecords:
```

```
        self.errorMessage(u"已到最后一页。")
```

```

        return

    self.queryRecord += 15
    self.queryViewModel.clear()

    if self.queryViewModel.get_n_columns() == 9:
        self.cursor.execute("SELECT * FROM books LIMIT ? OFFSET ?",[15,
self.queryRecord])

    elif self.queryViewModel.get_n_columns() == 4:
        self.cursor.execute("SELECT * FROM reader LIMIT ? OFFSET ?",[15,
self.queryRecord])

    for row in self.cursor:
        self.queryViewModel.append(row)

```

还书操作实现：

```

def on_returnQuery_clicked(self, button, data = None):
    readerid = self.returnEntryReaderID.get_text()
    bookid = self.returnEntryBookID.get_text()
    #同时输入读者号和书本号的查询
    if len(readerid) and len(bookid):
        self.cursor.execute("SELECT borrow.BorrowID, books.BookName,
reader.ReaderName,reader.ReaderSchool, borrow.BorrowDate FROM borrow,books,reader\
WHERE books.BookID = borrow.BookID AND \
reader.ReaderID = borrow.ReaderID AND\
borrow.ReaderID = ? AND borrow.BookID = ?",

```

```
[readerid, bookid])

    #仅输入读者号的查询
    elif len(readerid):

        self.cursor.execute("SELECT borrow.BorrowID, books.BookName,
reader.ReaderName,reader.ReaderSchool, borrow.BorrowDate FROM borrow,books,reader\
WHERE books.BookID = borrow.BookID AND \
reader.ReaderID = borrow.ReaderID AND\
borrow.ReaderID = ?", [readerid])

    #仅输入书本号的查询
    elif len(bookid):

        self.cursor.execute("SELECT borrow.BorrowID, books.BookName,
reader.ReaderName,reader.ReaderSchool, borrow.BorrowDate FROM borrow,books,reader\
WHERE books.BookID = borrow.BookID AND \
reader.ReaderID = borrow.ReaderID AND\
borrow.BookID = ?", [bookid])

    else:

        self.errorMessage(u"请输入查询条件。")
        return

    self.returnValueModel.clear()

    for row in self.cursor:

        self.returnValueModel.append(row)
```