图书馆管理系统潘聪3110305129 题目：图书馆管理系统

专业：计算机与信息科学班级： 座号：

分工：独立分组 福建工程学院信息科学与工程学院 数据库应用课程设计 1105 3110305129姓名：潘聪

2013年9月9日

图书馆管理系统潘聪3110305129

目录

[1.系统定义 1](#_Toc481440016)

[1.1背景资料： 1](#_Toc481440017)

[1.2系统功能的基本要求： 3](#_Toc481440018)

[1.3目标： 4](#_Toc481440019)

[1.4功能特点： 4](#_Toc481440020)

[2、需求分析 6](#_Toc481440021)

[2.1确定对系统的综合需求，系统功能要求，性能要求，开发环境，运行要求等。 6](#_Toc481440022)

[2.2给出系统的逻辑模型：数据流程图、数据字典 7](#_Toc481440023)

[2.2.1数据流程图 7](#_Toc481440024)

[2.2.2数据字典 9](#_Toc481440025)

[3.系统设计 12](#_Toc481440026)

[3.1概念结构设计 12](#_Toc481440027)

[3.2逻辑结构设计 16](#_Toc481440028)

[3.3系统功能模块图 17](#_Toc481440029)

[3.4系统总流程图 17](#_Toc481440030)

[3.5其它设计图形工具 19](#_Toc481440031)

[4.详细设计 19](#_Toc481440032)

[1.创建数据库 19](#_Toc481440033)

[2.创建管理员表 20](#_Toc481440034)

[3.创建读者表 21](#_Toc481440035)

[4.创建读者类别表 21](#_Toc481440036)

[5.创建图书表 21](#_Toc481440037)

[6.创建图书类别表 22](#_Toc481440038)

[7.创建借阅表 22](#_Toc481440039)

[8.主要存储过程 23](#_Toc481440040)

[5.系统实现与测试 27](#_Toc481440041)

[5.1开发平台和工具选择 27](#_Toc481440042)

[5.2系统测试 28](#_Toc481440043)

[5.2.1测试中常见错误 28](#_Toc481440044)

[5.5.2程序运行截图 30](#_Toc481440045)

[6.课程设计总结 33](#_Toc481440046)

[7.参考文献 33](#_Toc481440047)

# 1.系统定义

## 1.1背景资料：

在高校中，学生的数量较多，图书的数量也比较多，每天图书的借阅量比较大，采用手工的处理方式已经不能满足这种需要了，因此需要开发一个能够处理平时借阅处理的信息系统，这个信息系统负责主要处理图书馆日常的借书、还书和查询业务。 图书管理工作都是采用传统的手工管理，经过对其图书管理工作中的图书信息、读者管理及借阅管理工作做全面调查研究，其现行手工流程具体分析如下：

图书登记：对上级分配、学校购入和社会捐赠的图书进行图书信息登记；

读者登记：图书管理员根据学校档案室提供的教师花名册和学生花名册进行读者登记，然后填发借书证，学生毕业和教师调出时将其借书证注销。

借阅管理：读者特别是学生读者持借书证到图书室借阅图书，图书管理员将借阅日期填写在书籍卡片上，将卡片夹在借书证中间，存放在图书室，学生看完后，送还所借书籍，收回借书证或另借其它书籍。

这种传统人工的管理方式效率低、保密性差, 一方面导致教师、学生对学校图书室藏书情况不了解，图书室究竟有哪些图书也不知道；另一方面图书管理员对图书管理工作力不从心，因为时间一长,将产生大量的文件和数据，这对于查找、更新和维护都带来了不少的困难，耗费劳动力大，难于避免错误的产生，更容易造成图书的外流。

然而现在，计算机的价格已经十分低廉，性能却有了长足的进步。在初级中学都装备有一定数量的电脑，中学教师都掌握了一定的计算机知识，它已经被学校应用于学校工作的许多领域。计算机之所以在中小学发挥如此重要的作用的原因主要有以下几个方面：

首先，计算机可以代替人工进行许多繁杂的劳动；

其次，计算机可以为农村初级中学节省许多资源；

第三，计算机可以大大的提高教职工的工作效率；

第四，计算机可以使敏感文档更加安全，等等。

然而为了更好的应用计算机，本次数据库专周我准备实现一个图书馆管理系统

1． 可以存储学校图书室的图书信息、读者信息和借阅信息，安全、高效；

2． 只需一名图书管理员即可操作系统，节省大量人力； 3． 可以迅速查到所需信息，图书管理系统可以为学校教师和学生提供充足的信息和快捷的查询手段，让教师和学生能够及时查询图书信息，更好地为读者服务。

4． 可以预借想借的书本。

5． 并能做到管籍互借，这样能更好地实现图书的流通，无形中增加了图书的资源。 以适应现行图书管理工作的要求，提高工作效率。

## 1.2系统功能的基本要求：





 基础信息处理：对图书、借阅者等信息的录入、添加、修改和删除。 查询处理：实现图书信息、读者信息和借阅信息的查询。注意采用多种查询方式，如单表查询、多表查询、分类查询、模糊查询、复合条件查询„„。 借阅管理，包括：

⑴借书：借书按借阅者的类别（如：本科生，研究生，教职员工）限制借书数，有超期图书的罚款处理；

⑵还书：进行还书处理。\*能输出快到期图书的还书提醒。

 统计处理：进行相应的统计查询与输出。如分类统计、\*根据图书利用率或周转率进行

每周Hot Books统计等。

## 1.3目标：

1. 支持日常的借书还书运作。

2. 集中管理图书和学生信息，方便日常维护。

3. 有良好的可移植性、可扩展性、可复用性（系统配置灵活，支持用户未来信

息化进程的发展）。

4. C/S系统，人机对话友好，操作方便，简单易学。

5. 系统运行效率高，响应速度合理，安全性较高，复杂业务也能轻松处理网络

运行稳定。

6. 系统建设费较低。

## 1.4功能特点：

1. 操作简单、界面友好。完全控件式的页面布局，使得管理员工作的工作更简

便，用户操作更加方便；借出图书和用户还书都只需一键操作；另外，跟踪

出现的提示信息业让用户随时清楚自己的操作情况。

2. 及时可见。管理员对图书本身信息和借阅信息都是一目了然，对系统内工作

信息的管理，改系统可以达到“及时修改、及时见效”的效果。

3. 功能完善。

管理员登录：

1) 添加（删除）管理员（超级管理员）

2) 借出图书

3) 读者还书

4) 查看图书借阅信息

5) 图书信息录入

6) 用户管理（添加，修改，删除）

7) 每周hot books统计

8) 退出系统

普通用户登录：

1) 新用户注册

2) 个人基本信息的查看和修改

3) 查看个人图书借阅信息

4) 注销账户

5) 查询图书

6) 退出系统

4. 特色功能。用户可以收藏自己喜欢的图书。用户注册信息或者是管理员录入

用户信息的时候同时可以添加用户的照片。根据读者的借阅信息，每周都会有hot books统计。当图书快要到期时提醒用户。

# 2、需求分析

## 2.1确定对系统的综合需求，系统功能要求，性能要求，开发环境，运行要求等。

### 1.功能要求

系统主要实现对图书馆藏书和读者的管理功能。

对于管理员来说，其功能有：添加（删除）管理员（超级管理员），借出图书，读者还书，查看图书借阅信息，图书信息录入，用户管理（添加，修改，删除）每周hot books统计。

对于读者用户来说，主要功能有：新用户注册，个人基本信息的查看和修改，查看个人图书借阅信息，注销账户，查询图书。

### 2.性能要求

图书馆管理系统在运行效率方面要求并不是太高，完成基本功能即可，性能方面并不会影响到整个系统的正常使用。

### 3.开发环境

本套系统所有的开发工具为Microsoft Visual Studio 2012，使用的数据库为Microsoft SQL Server2005，基于.net framework 4.0使用C#语言开发。

### 4.运行要求

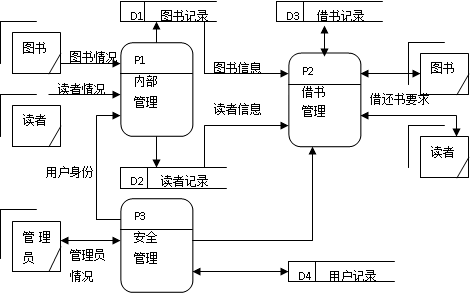
本套系统适用于安装了windows系统的PC机，所以对于使用mac系统的用户只能说声遗憾了。

## 2.2给出系统的逻辑模型：数据流程图、数据字典

### 2.2.1数据流程图

经过详细的调查，弄清了系统现行的业务流程。通过分析和抽象后，用数据流图表示。图1是图书馆管理系统的顶层数据流图。

图1 图书馆管理系统顶层数据流图



图书馆管理系统潘聪3110305129

在图书馆管理系统顶层数据流图中，“ P1 内部管理”和“P2 借书管理”两个处理框所表示的功能都太复杂，对它们进一步细化后得出第二层数据流图。见图2和图3。实际上“P1.1 图书管理”还可以细化，分为新书处理和图书记录维护两个下级处理框，而“P1.2 读者管理”也可以细化为新读者入户和读者记录维护两个下级处理框。

图2 “内部管理”的细化数据流图

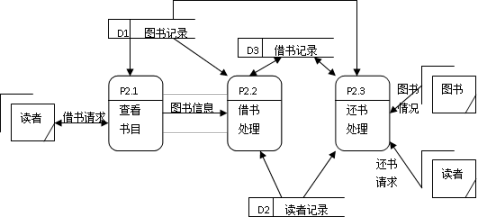
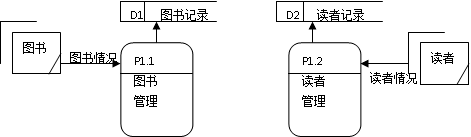


图3 “借书管理”的细化数据流图



经过整理后得出细化后得系统数据流图，如图4所示。

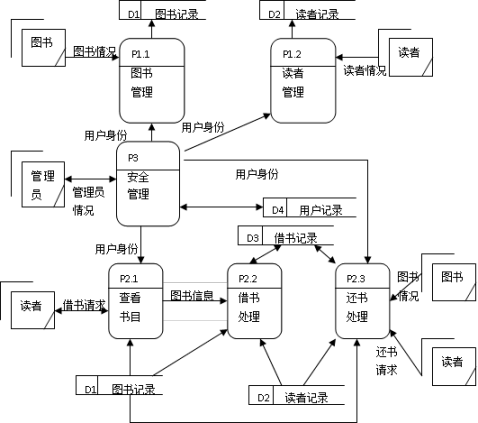


图4 细化后的图书馆管理系统数据流图

### 2.2.2数据字典

有了数据流程图仅仅只能体现系统运行的大致运行流程，要想了解系统内部实现，还必须要有数据字典，下面列出本套系统中使用的数据字典。

#### 2.2.2.1主要的数据流定义

1.数据流名称：图书情况

位置：图书->P1.1，图书

->P2.3

定义：图书情况=类别+出版社+作者+书名+定价+完好否

说明：图书入库时，根据图书情况建立图书记录；读者还书时要核实图书基本信息是否与图书记录相符，要检查图书是否完好。 2．数据流名称：用户身份

位置：P3->{P1.1，P1.2，P2.1，P2.3} 定义：用户身份=[硕士|本科|教职员工] 说明：不同的用户身份进入的处理过程不同。

#### 2.2.2.2主要的存储过程定义

1．存储过程编号：D1

数据结构：借书记录=书号+读者编号+借阅日期

说明：读者借阅了一本书，就生成一条借阅记录。借阅日期为添加记录的当天日期。 2.

#### 2.2.2.3主要的触发器定义

1．触发器编号：T1 说明：管理员删除用户的时候要检查该用户是否有未还的图书，如果有则不可删除

2．触发器编号:T2

说明：用户借书时，同时要先把图书表中的借出总量加1，添加一条借书记

录。

#### 2.2.2.4数据字典



管理员（id，姓名，管理员密码，性别，单位，身份证，管理员类别）

读者类别（id，类别名称，可借本数）



图书（id，书名，ISBN，作者，出版社，价格，图书类别id，借阅次数，图书总数，借出数目）

图书类别（id，类别名称）

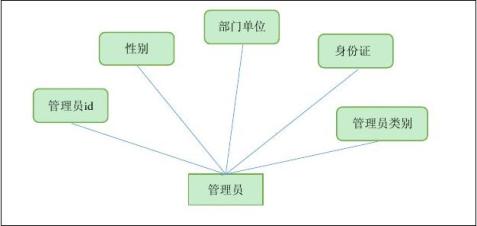
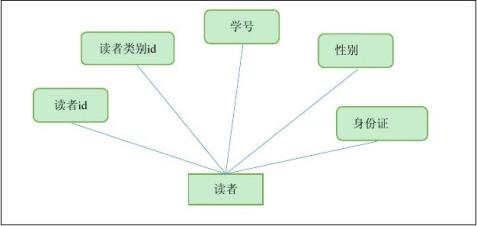


借阅（id，读者id，图书id，借阅日期，还书日期，状态）

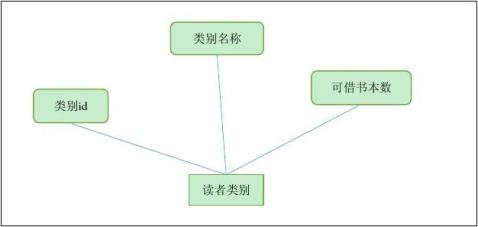
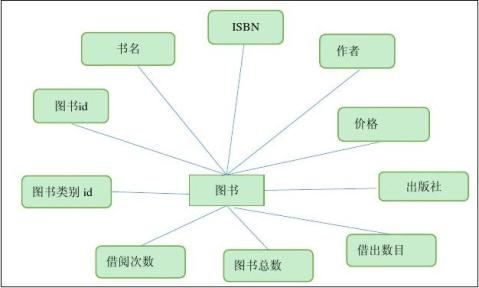
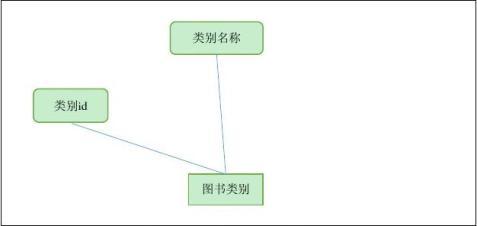
# 3.系统设计

## 3.1概念结构设计

画出系统E-R图。

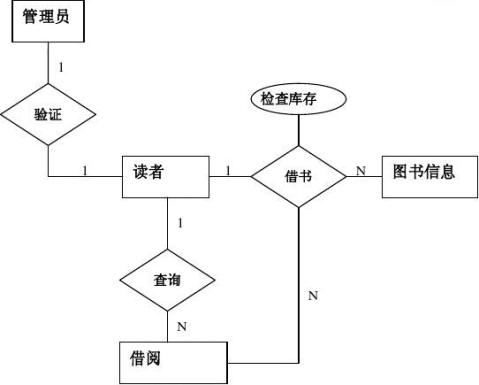
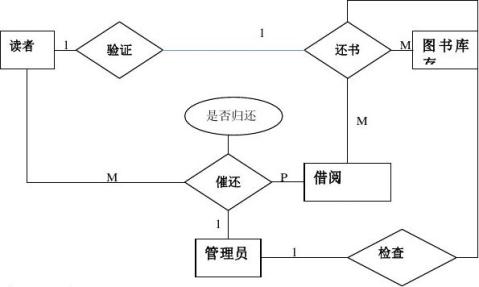


读者还书E-R图：



N

读者借书E-R图



## 3.2逻辑结构设计

将E-R图（或其他概念设计图）转换为具体的DBMS对应的关系模式。

管理员（id，姓名，管理员密码，性别，单位，身份证，管理员类别）

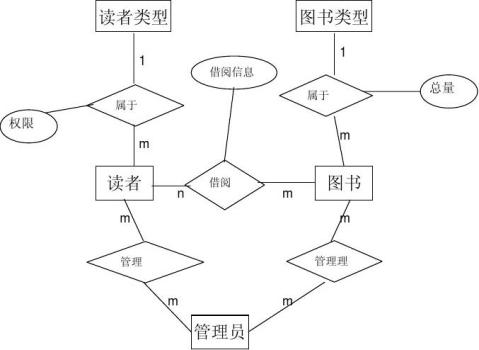
读者（id，姓名，登录密码，学号，性别，读者类别id，身份证）

读者类别（id，类别名称，可借本数）

图书（id，书名，ISBN，作者，出版社，价格，图书类别id，借阅次数，图书总数，借出数目,位置）

图书类别（id，类别名称）

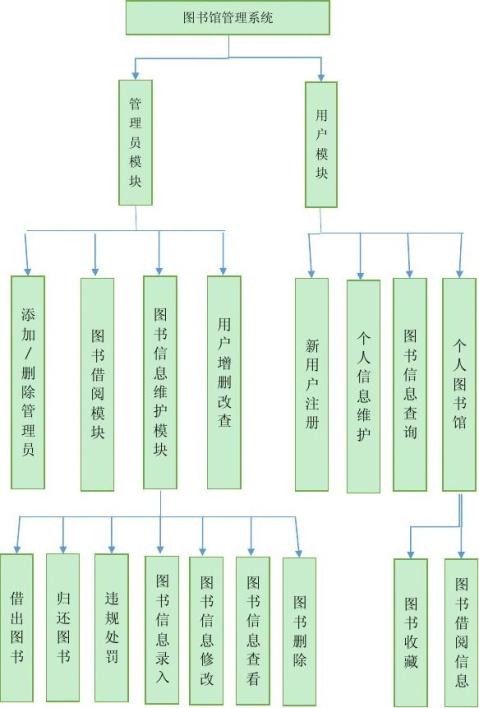
借阅（id，读者id，图书id，借阅日期，还书日期，状态，催还标志位）



## 3.3系统功能模块图

画出系统功能模块图，并对子模块功能进行详细说明。

## 3.4系统总流程图



## 3.5其它设计图形工具

# 4.详细设计

主要的SQL语句，主要模块的程序流程图。

## 1.创建数据库

CREATEDATABASE lib ONPRIMARY

(

NAME='lib',

FILENAME='d:\db\lib.mdf',

SIZE=20MB,

FILEGROWTH=2MB

)

LOGON

(

NAME='libLog',

FILENAME='d:\db\libLog.ldf',

SIZE=1MB,

MAXSIZE=10MB,

FILEGROWTH=10%

);

## 2.创建管理员表

createtable t\_admin

(

admin\_id intidentity(1,1)primarykey,

admin\_name char(20)notnull,

admin\_realname char(20)notnull,

admin\_regtime datetimenotnull,

admin\_password char(35)notnull,

admin\_sex char(2)check(admin\_sex in('男','女'))notnull, admin\_email char(20)notnull,

admin\_tel char(20)notnull,

admin\_dept char(20)notnull,

admin\_idcard char(20),

admin\_category char(20)notnull,

admin\_photo image,

)

insertinto t\_admin

values('admin','潘聪',getDate(),'123','男

','353514599@qq.com','18065140345','图书管理部','320925199310074219','超级管理员','')

## 3.创建读者表

createtable t\_reader

(

) reader\_id intidentity(1,1)primarykey, reader\_name char(20)notnull, reader\_password char(35)notnull, reader\_sex char(2)check(reader\_sex in('男','女'))notnull, reader\_idcard char(20), reader\_photo image, reader\_category int, foreignkey reader\_category references t\_rcategory(rcate\_id)

## 4.创建读者类别表

createtable t\_rcategory

(

rcate\_id intidentity(1,1)primarykey, rcate\_name varchar(20)notnull,

rcate\_days intnotnull,

)

insertinto t\_rcategory values('研究生',10); insertinto t\_rcategory values('本科生',8); insertinto t\_rcategory values('教职工',6);

## 5.创建图书表

createtable t\_book

(

book\_id intidentity(1,1)primarykey, book\_name char(30)notnull, book\_isbn char(20)notnull, book\_writer char(20)notnull, book\_price floatnotnull, book\_press char(20)notnull, book\_press\_time datetime, book\_category int,

book\_count intdefault 0, book\_all int, book\_out intdefault 0, book\_photo image, foreignkey book\_category references t\_bcategory(bcate\_id) )

## 6.创建图书类别表

createtable t\_bcategory

(

bcate\_id intidentity(1,1)primarykey,

bcate\_name char(20)notnull

)

insertinto t\_bcategory values('科技');

insertinto t\_bcategory values('人文');

## 7.创建借阅表

createtable t\_borrow

(

borrow\_id intidentity(1,1)primarykey,

borrow\_reader\_id intnotnull,

borrow\_book\_id int,

borrow\_time datetime,

back\_time datetime,

borrow\_status char(10)check(borrow\_status in('未还','已还'))notnull,

foreignkey borrow\_reader\_id references t\_reader(reader\_id), foreignkey borrow\_book\_id references t\_book(book\_id)

)

## 8.主要存储过程

------------存储过程--------------

--用户借书， Reader\_BorrowBook

--1.查询用户目前已借图书本数，与最大本数比较

--2.图书借出数量加1，图书借出次数加1

create procedure Reader\_BorrowBook @book\_id int,@reader\_id

int,@borrow\_time DateTime,@back\_time DateTime, @outstr char(50) output

as

begin declare @rcate\_days int--用户可借本数 declare @has\_borrow int--用户已借数量 declare @book\_out int --图书借出数量

declare @book\_all int --图书总数 select @rcate\_days=rcate\_days from rc\_view where reader\_id = select @has\_borrow=count(\*) from t\_borrow where @reader\_id

borrow\_reader\_id = @reader\_id and borrow\_status='未还'

select @book\_out = book\_out from t\_book where book\_id = @book\_id

if @has\_borrow >=@rcate\_days set @outstr = '对不起，您目前借书数目已达上限，请归还后再借书！' else if @book\_out >=@book\_all set @outstr = '对不起，该书剩余数量为0！' else begin update t\_book set book\_out=book\_out+1, insert into t\_borrow select @book\_all = book\_all from t\_book where book\_id = @book\_id book\_count=book\_count+1 where book\_id = @book\_id

(borrow\_reader\_id,borrow\_book\_id,borrow\_time,back\_time,borrow\_status)

----------还书存储过程

create procedure Reader\_BackBook @book\_id int,@reader\_id

int,@borrow\_id int

as

begin

update t\_borrow set borrow\_status = '已还' where borrow\_id = @borrow\_id

-------------返回图书借阅时间存储过程，确定用户需要缴纳多少罚款

create procedure borrow\_days @borrow\_id int,@outstr char(50) output

21 / 30

set @outstr='借书成功！' end values(@reader\_id,@book\_id,@borrow\_time,@back\_time,'未还') end update t\_book set book\_out = book\_out-1 where book\_id = @book\_id end

as

begin

declare @days int

declare @nowTime DateTime

declare @backTime DateTime

set @nowTime = getDate()

select @backTime=back\_time from t\_borrow where

borrow\_id=@borrow\_id

SELECT @days = DATEDIFF(day,@backTime,@nowTime)

set @outstr =convert(char(50),@days)

end

--删除类别，

--1.删除类别

--2.删除所有改类别的用户以及借阅的图书

create procedure delete\_rcate @rcate\_id int

as

begin

declare @reader\_id int

delete from t\_rcategory where rcate\_id = @rcate\_id

delete from t\_reader where reader\_category = @rcate\_id select @reader\_id= reader\_id from t\_book where

reader\_category=@rcate\_id

delete from t\_borrow where borrow\_reader\_id = @reader\_id end

--删除图书存储过程

--删除图书的同时要删除借书记录

--删除图书 t\_book

--删除借阅记录 t\_borrow

create procedure delete\_books @book\_id

as

begin

delete from t\_book where book\_id = @book\_id

delete from t\_borrow where book\_id = @book\_id

end

--删除用户的同时要把所有和用户相关的信息全部删除

create procedure delete\_reader @reader\_id int

as

begin

delete from t\_reader where reader\_id = @reader\_id delete from t\_borrow where reader\_id = @reader\_id

end

# 5.系统实现与测试

## 5.1开发平台和工具选择

本套系统所有的开发工具为Microsoft Visual Studio 2012，使用的数据库为Microsoft SQL Server2005，基于.net framework 4.0使用C#语言开发。

WPF（Windows Presentation Foundation）是微软推出的基于Windows Vista的用户界面框架，属于NET Framework 3.0的一部分。它提供了统一的编程模型、语言和框架，真正做到了分离界面设计人员与开发人员的工作；同时它提供了全新的多媒体交互用户图形界面。

## 5.2系统测试

简要说明测试情况。

给出系统代表性运行界面，如主界面，模块界面、设计的查询页和报表等。

## 5.2.1测试中常见错误

本次开发使用C#语言过程中，最容易报错的地方就是ADO.NET部分，即数据访问层，但是大部分问题，还是可以比较容易找出原因的。

在处理用户照片，和图书封面图片时问题比较难以调试

错误一：因为用户照片不是非空字段，图片在数据库中是image字段，在C#中可以用byte数组存，用户可以选择上传或者不上传，所以在上传时要进行如下处理：

new SqlParameter("reader\_photo", rm.reader\_photo == null ? new byte[0] : rm.reader\_photo)

错误二：从数据库中读取图片，先判断image字段的长度是否为0，或者小于某个很小的整数，切忌不能判断image字段是否为空，在与DateGrid结合使用时，也要注意selectItem 是否为空！

如：ReaderModel rm = (ReaderModel)gridReader.SelectedItem;

if (rm != null)

{

if (rm.reader\_photo.Length >20)

{

this.ShowImg(rm.reader\_photo);

}

}

错误三：和图片类似的是普通字段，普通字段上传为空时也要进行处理，分别定义如下两个方法，一个是从数据库中取出空字段，一个是向数据库里面存空字段。

public static object FromDbValue(object value)

{

if (value == DBNull.Value)

{

return null;

}

else

{

return value;

}

}

public static object ToDbValue(object value)

{

if (value == null)

{

return DBNull.Value;

}

else

{

return value;

}

}

如：

存入数据库

new SqlParameter("reader\_bz", SqlHelper.ToDbValue(rm.reader\_bz)

rm.reader\_bz = (string) SqlHelper.FromDbValue( reader.GetValue(10)); 从数据库中读取

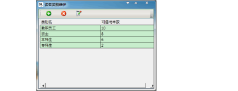
## 5.5.2程序运行截图

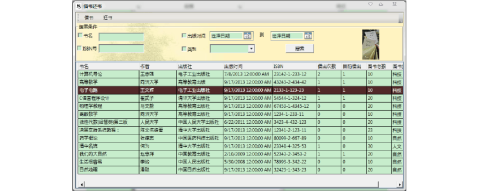
登录：

右下角弹窗显示近三天需要归还的图书：

管理员界面



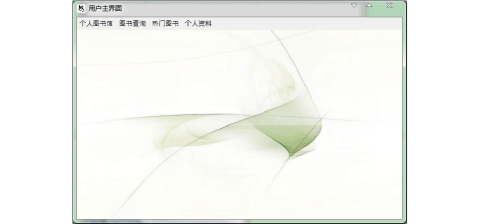


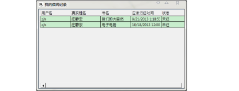


用户界面：



右下角弹窗，提示用户三天之内需要归还的图书





# 6.课程设计总结

本次课程设计中，受益匪浅，由于Delphi学的不是很好，所以在做课程设计的时候果断放弃了Delphi，而选择了C#中的WPF技术，C#的语法和C++，java都非常类似，所以自学起来也非常容易。做完整个课题之后，觉得成功是给有准备的人的，非常感谢老师让我们在第一周周三就开始检查文档。如果没有这项检查，我认为本次课程设计大部分同学会做的不理想。一个项目的分析阶段的重要性远远要大于后期的编码阶段。在开发过程中碰到了很多问题，但是还好现在的网络资源相当丰富，很多问题都能找到解答。这次的课程设计部仅仅锻炼的是我们的编码能力，同时也提高了我们的设计能力，包括UI设计，数据库设计以及整个项目的规划。这是大学以来历时时间最长的一次，同时也是最有意义的一次。

# 7.参考文献

1.《数据库系统概论》，王珊，萨师煊，高等教育出版社