

摘 要

随着信息技术在管理上越来越深入而广泛的应用，管理信息系统的实施在技术上已逐步成熟。管理信息系统是一个不断发展的新型学科，企业要生存要发展，要高效率地把企业活动有机地组织起来，就必须加强企业管理，即加强对企业内部的各种资源（人、财、物等）的有效管理，建立与自身特点相适应的管理信息系统。

本文介绍了在 PowerBuider 环境下采用“自上而下地总体规划，自下而上地应用开发”的策略开发一个管理信息系统的过程。通过分析传统的人工管理图书馆的不足，创建了一套行之有效的计算机管理图书馆的方案。文章详细介绍了图书管理信息系统的系统分析部分，包括可行性分析、组织机构分析、管理职能分析、业务流程分析、数据流程分析、数据字典、处理描述等等；系统设计部分主要介绍了系统功能设计和数据库设计；系统实现部分列出了几个主要的程序框图，并附带了一些主要的窗口和程序。

本系统界面友好，操作简单，比较实用。

【关键字】 管理信息系统、图书管理、PowerBuider 应用。

Abstract

With more and more widespread and profound application of information technology in management, the implement of management information system has become mature in technology step by step. Managing information system is a new subject. Enterprise needs existence and development, so enterprise activities should be organized efficiently and organically, which means tightening up the enterprise management and strengthening effective management of any resource (staff, finance, property, etc.) internal the enterprise, and also establishing a management information system fitting in with its own characteristics.

This article introduces the detailed process of exploring a management information system under the environment of PowerBuilder, utilizing “Top-Bottom” overall plan and a strategy according to “Bottom-Top” application and exploitation. That is to establish a set of effective scheme for library management by computer, through analyzing disadvantages of traditional library management by human resources. This article emphasizes on three sections. The system analysis section of library management information includes feasible analysis, organizing organs analysis, management function analysis, technological process of business analysis, technological process of data analysis, data dictionary, disposal description and the like. The system design section mainly focuses on system function design and data base design. And the system realization section has provided several major program charts, together with the main windows and programs.

This economical and pragmatic system has explicit interface, with simple operation.

[Keywords] MIS、library management、PowerBuilder application

目 录

一 前 言.....	1
1 MIS 系统开发方法.....	1
2 MIS 系统开发过程.....	2
二 系统需求分析.....	4
1 现行业务描述.....	4
2 现行系统存在问题的分析.....	5
3 解决方案.....	6
1 系统目标设计.....	7
2 系统功能分析.....	8
3 系统各功能模块设计.....	8
4 数据流图.....	9
1 数据库需求分析.....	13
2 数据库逻辑结构设计.....	14
1 POWERBUILDER 开发工具简介.....	20
2 POWERBUILDER 8 应用程序开发的基本步骤.....	20
3 编码规范.....	21
4 创建祖先窗口和全局函数.....	21
5 应用程序对象 APP_LIBRARAIN.....	24
7 补充说明.....	48
七 结束语.....	49

一 前言

随着人类社会的发展，人类对知识的需求也不断地增长。在这种形势下，书籍就渐渐地成为人们获取并增长知识的主要途径，而图书馆就自然而然地在人们的生活中占据了一定的位置，如何科学地管理图书馆不但关系到读者求知的方便程度，也关系到图书馆的发展，因此，开发一套完善的图书馆管理系统就必不可少。

管理信息系统(简称 MIS)是介于信息论，经济管理理论，统计学与运筹学及计算机科学之间的一门边缘性，综合性，系统性的交叉科学，它是随着管理科学，信息技术，计算机技术等的发展而产生和发展起来的。

图书馆管理系统是典型的信息管理系统，其开发主要包括后台数据库的建立和维护以及前端的应用程序的开发两个方面。对于前者要求建立数据的一致性和完整性，对于后者则要求应用程序功能的完备，易用等特点。基于上述考虑本系统主要利用 PowerBuilder 8 作前端的应用开发工具，利用 Adaptive Server Anywhere 7 作为后台的数据库，利用 WINDOWS 作为系统平台开发的图书管理系统。另外本图书馆管理系统利用软件工程化思想和方法，总体上是采用结构化生命法进行系统分析和设计的，而系统实现等步骤则采用了原型法和面对对象的方法。

1 MIS 系统开发方法

管理信息系统的开发方法有生命周期法、原型法和面向对象的开发方法等：

生命周期法 (Life Cycle Method) 是 20 世纪 60 年代发展起来的一种应用广泛且比较成熟的管理系统开发方法，它的基本思想是将系统的开发工作从开始到结束划分为若干个阶段，每个阶段都有明确的任务，而系统开发出来后，并不意味着生

命周期的结束，而意味着根据组织的需要对系统的修改和重建的开始。

原型法(Prototyping Method)是 20 世纪 80 年代发展起来的，旨在改变生命周期法的缺点的一种系统开发方法，该法的开发思路是首先根据用户的要求，由用户和开发者共同确定系统的基本要求和主要功能，利用系统快速生成工具，建立一个系统模型，再在此基础上与用户交流，将模型不断补充、修改、完善，如此反复，最终直至用户和开发者都比较满意为止，从而形成一个相对稳定、较为理想的管理信息系统。

面向对象(Object Oriented)的开发方法于 20 世纪 80 年代开始兴起的，是一种基于问题对象的自底向上的一种系统开发方法，这种方法的特点是以对象为基础，对象是分析问题和解决问题的核心。

2 MIS 系统开发过程

一个 MIS 系统的开发过程一般包括如下几个步骤：

(1) 需求分析：需求分析主要是了解用户的需求。需求了解得越详细，程序的后期开发与维护费用就会越少。一般的开发团队中，需求分析都是由资历较深的系统分析员或项目经理担当，可见它的重要性。需求分析制订好后，需要反复修改。将最后的结果交给用户审定，确认无误后，由系统分析员完成需求分析文档，再开始下一步工作。

(2) 概要设计：概要设计紧跟在需求分析之后。用户需求明确后，将得到的数据分析后，开始构建数据库的逻辑结构。此时，数据库中的表格还未成形，通过各种分析工具（如 PowerDesigner 等）画出数据流图，最后就可抽象出数据库的具体表结构。这时由系统分析人员反复审核。确认所有的需求都考虑在内，没有遗漏后，就可以开始制订概要设计文档。概要设计文档形成后，整个程序的逻辑框架也就形成了。

(3) 详细设计：概要设计完成后，根据设计中制订的业务模块。就可以进行详细分析设计了。详细设计就是将各个业务模块的窗口全部建好，各

个窗口控件的处理代码全部用语言表达出。所以详细设计是整个系统中最繁琐的环节。详细设计完成后，整个程序就确定了，再由编程人员根据详细设计文档将代码完成。整个开发工作就宣告结束。

1) 程序编码：程序编码相对于其他环节来说比较简单，程序员只需要根据详细分析文档写程序编码，保证代码没有错误即可。程序编码需要注意的是整个程序书写中命名的规范化与编程风格的规范化，这需要较长时间的培养来形成。需要在不断的实践中形成自己独特的风格。总的来说，不要过分地追求复杂的算法，因为那可能会导致后期维护人员无法读懂你的代码而造成维护的困难。

(4) 测试：程序编码完成后，就需要测试。测试有几种类型，主要是测试代码有无逻辑错误以及在加载数据环境下程序的稳定性问题。测试工作中发现的错误应及时改正，然后将它记录到测试文档中。

(5) 打包：测试完成，确认无误后。程序就可以打包发行了。打包一般使用工具如 PWISE 等。

以上是一个完整的 MIS 系统开发过程，其实不仅 MIS 系统，其他类型的程序开发也基本相同。

二 系统需求分析

整个软件生命周期中，开发所占的费用和时间都很小。后期维护工作一般要占整个软件生命周期的 80% 以上。所以系统分析很重要，一个好的系统分析可以减少很多后期维护工作。

1 现行业务描述

下面以一所学校的图书馆为例子进行分析，画出图书馆的组织结构图如下：

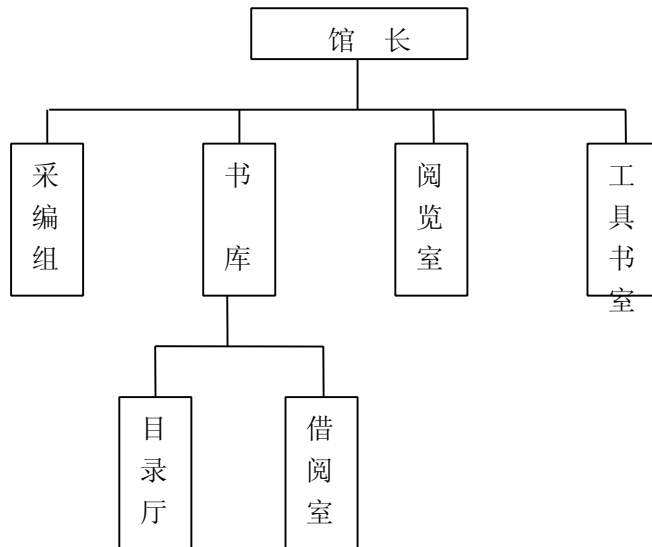


图 2.1

该图书馆各个部门负责的主要业务如下：

(1) 采编组主要负责图书采编工作，包括购置新书、打印编目、增加数量。

(2) 目录厅主要负责读者查询工作，包括可借图书（按图书类别、图书名称或图书作者查询）；新书目录（按图书类别、日期查询）。

(3) 借阅组主要负责图书流通、查询统计、借阅查询等工作。

(4) 阅览室、工具书室主要负责读者阅览工作，包括：阅览各种杂志、

报纸、阅览各种工具书。

下面绘制出图书馆流通业务中借书的流程图：

- (1) 读者在目录厅查阅索引卡；
 - (2) 读者写出所借图书的分类号、种次号、交给图书管理员，并出示本人的借书证；
 - (3) 图书管理员根据图书的分类号、种次号到书库找书；
 - (4) 将图书交给读者，并由读者填写所借图书的借书卡。
 - (5) 图书管理员把借书卡保存到写有该读者借书证号的口袋里。
- 得出该图书馆业务流程图如下所示：

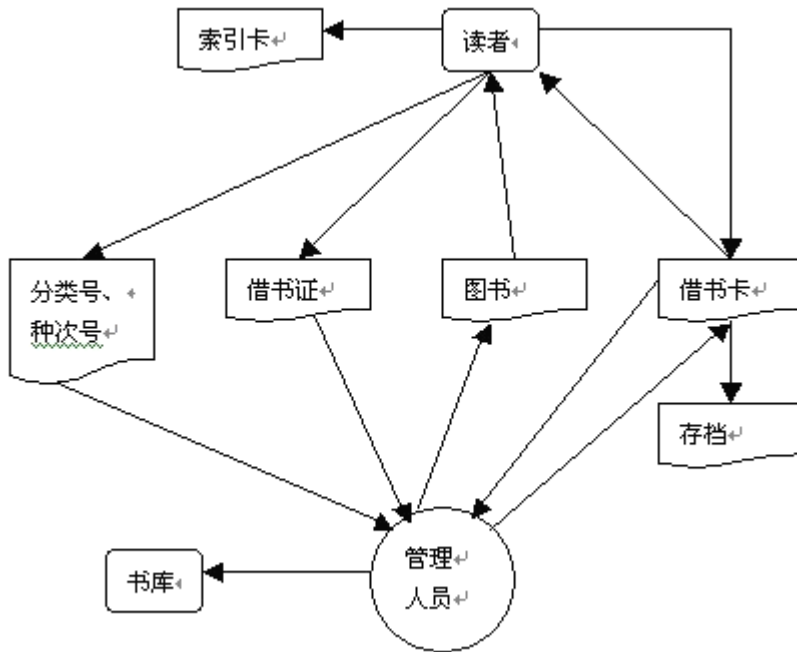


图 2.2

2 现行系统存在问题的分析

该图书馆工作大部分还是进行着手工管理，工作起来效率很低，由于不可避免的人为因素，经常造成数据的遗漏、误报。计算机信息化管理有着储存信息量大、速度快等许多优点，提供给我们的处理信息及时快捷，

同时也提高了我们工作人员的自身素质。

3 解决方案

基于以上的对问题的分析，利用计算机给我们提供信息，及时地调整学校图书馆图书结构，并且对学生们的借阅过程形成一整套动态的管理就十分必要了。

图书馆信息系统的建立，需要进行用户的需求调查与分析，以确定系统目标，提出解决问题的详细方案，这是系统建设的重要环节。

学校图书馆规模相对于商业图书馆小，但要有着完善的图书信息，对每一册书要有详细的记录信息，功能上对系统的需求是全面的。因此对用户的需求可分为四个方面：一方面是图书馆工作人员通过计算机来管理各类图书，分类编号，调整图书结构，增加图书库存，适应学生的需求；二方面是学生们能通过计算机来查询浏览图书馆中的图书，来确定自己需要的书籍，为借阅做好准备；三方面是工作人员对学生们的借书证发放；四方面也是图书馆的核心工作，对图书的借阅管理。

这四个方面形成了学校图书馆一个工作整体，采用计算机自动化信息处理，可方便快捷地共享信息、交流信息，高效地协同工作。

在软、硬件方面对系统的需求，软件要求易学，界面友好，容易掌握，可以很简单方便的管理各种图书信息。硬件的配置要求不能太高，这样可以很好的适应当前的学校图书馆。

在开发方式上对系统的需求，由于学校图书馆是一种无商业利益的，所以系统设计开发的周期要短，在短时间内完成，减少开发成本，提高开发效率，方便、简单、实用作为系统开发的指导思想。

三 系统总体结构设计

如果要完成一个完整的图书管理系统，需要的相关信息很多。由于种种条件的限制，同时考虑到开发成本，所以本系统仅实现基本的管理功能。并不能完全满足上面例子所描述的图书馆，使用前应进行二次开发。

1 系统目标设计

系统开发的总目标是实现内部图书借阅管理的系统化、规范化和自动化。

能够对图书进行注册登记，也就是将图书的基本信息（如：书名、作者、价格等）预先存入数据库中，供以后检索。

能够对借阅人进行注册登记，包括记录借阅人的姓名、地址、电话等信息。

提供方便的查询方法。如：以书名、作者、出版社、出版时间（确切的时间、时间段、某一时间之前、某一时间之后）等信息进行图书检索，并能反映出图书的借阅情况；以借阅人编号对借阅人信息进行检索；以出版社名称查询出版社联系方式信息。

提供统计分析功能。可以展现出图书类型比例、库存与借出比例。

提供旧书销毁功能，对于淘汰、损坏、丢失的书目可及时对数据库进行修改。

能够对使用该管理系统的用户进行管理，按照不同的工作职能提供不同的功能授权。

提供较为完善的差错控制与友好的用户界面，尽量避免误操作。

2 系统功能分析

根据以上功能，将图书管理系统的功能分为：

数据输入部分：主要包括图书基本信息的录入、借阅人基本信息的录入、用户基本信息的录入。

数据输出部分：主要是各种统计查询，包括：根据图书信息（如书名、作者、出版社等）查阅图书及其借阅信息、统计输出图书类型比例等。

数据处理部分：主要涉及借阅和归还的处理，如一本书借出后，必须在数据库中将该书标记为已借出，以防出现数据库中有书但图书馆无书的情况；一本书归还后，同样必须在数据库中将其标记为已经归还，以便再次借出。

3 系统各功能模块设计

在系统功能分析的基础上，考虑 PowerBuilder 程序编制的特点，得到如图所示的系统功能模块图：

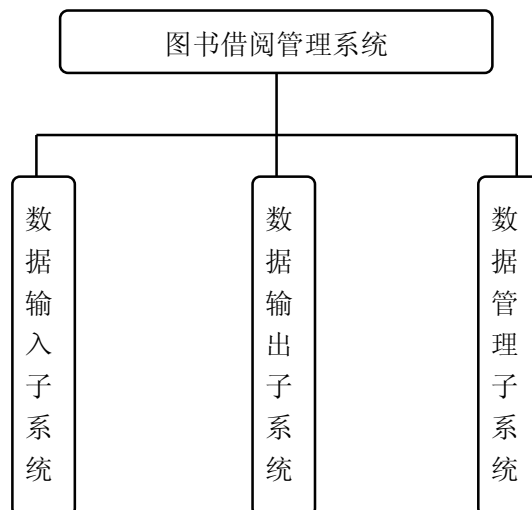


图 3.1

4 数据流图

(1) 图书基本信息登记/修改

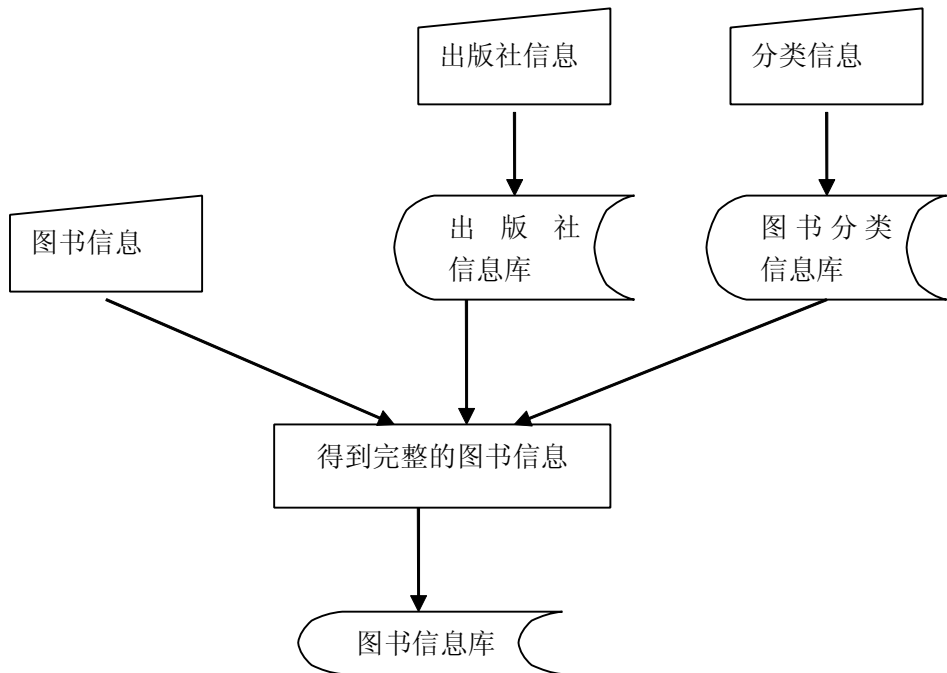


图 3.2

说明：出版社信息人工录入后，存放在出版社信息库中，便于今后书籍信息的录入以及出版社信息的查询。分类信息由图书馆进行分类定制，存放在图书分类信息库中，

用于图书的分类管理。登记新书时，当涉及到出版社和分类的填写时，只需做相应的选择即可，并存放在图书信息库中。

(2) 借阅人基本信息登记/修改

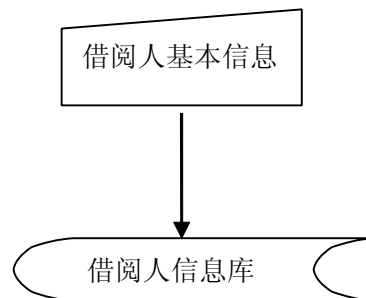


图 3.3

说明：录入借阅人基本信息后存放在借阅人信息库中。

(3) 图书查询

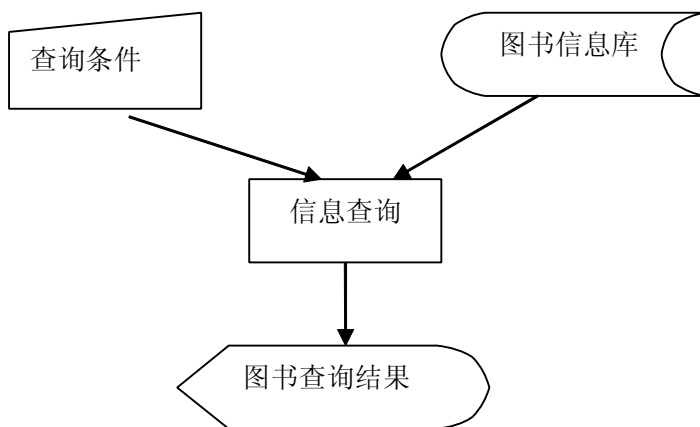


图 3.4

说明：输入查询条件后，检索图书信息库，并反应出查询结果。

(4) 图书统计

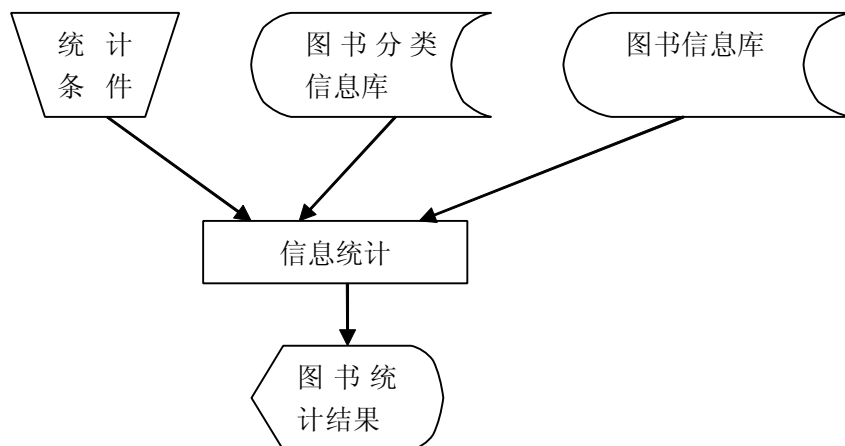


图 3.5

说明：根据选择的统计条件，检索图书分类信息库或图书信息库，并反应出统计结果。

(5) 图书借阅

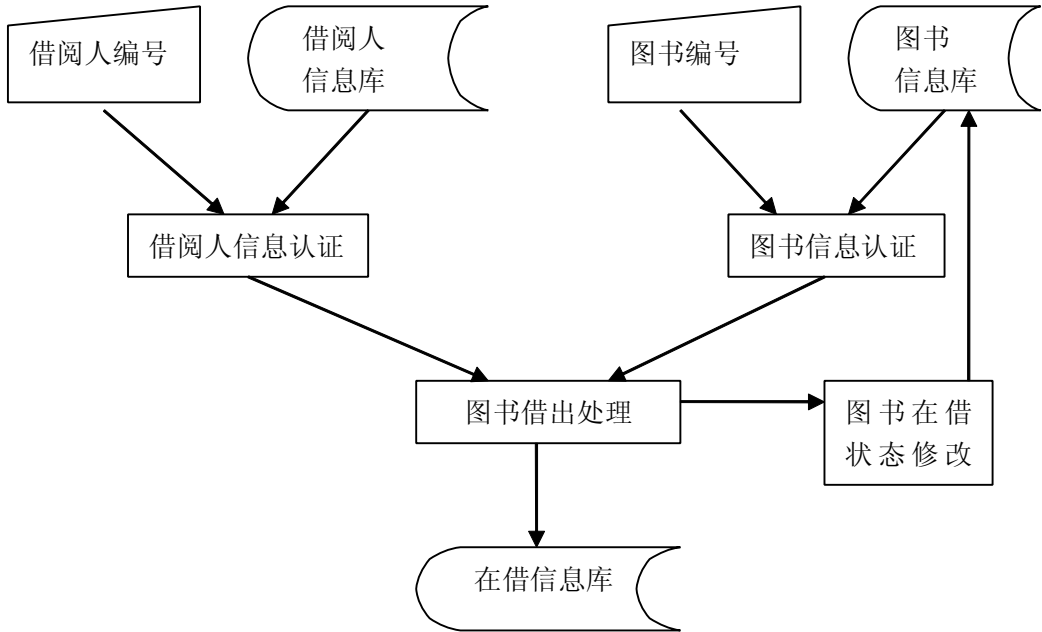


图 3.6

说明：输入借阅人编号，通过检索借阅人信息库，核实借阅人身份。输入图书编号，通过

检索图书信息库，核实图书信息。进行图书借出处理后，在图书借阅库中进行登记，

并将借出图书的在借标志设为“是”。

(6) 图书归还

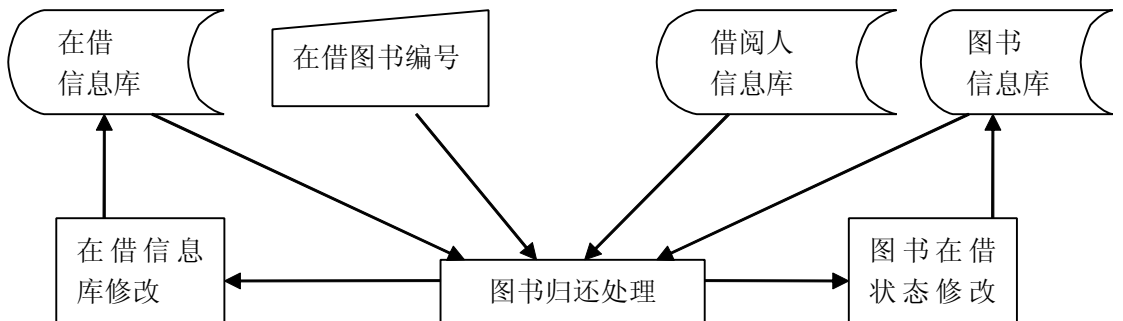


图 3.7

说明：输入图书编号，检索在借信息库，反应出图输借阅情况。进行归还操作后，修改在

借信息库中相关条目，同时在图书信息库中修改相关图书在借标志为“否

四 数据库设计

数据库的设计是 MIS 系统的核心部分，设计数据库，首先要进行需求分析，然后进行数据库的概念结构设计、逻辑结构设计、数据库结构实现等步骤。

1 数据库需求分析

根据以上的需求分析和数据组织，开始设计数据结构，即根据需求勾画出实体/关系图（E/R）。在概念上，E/R 图代表的是系统需要的数据及其这些数据之间的关系。如图所示的实体/关系图：

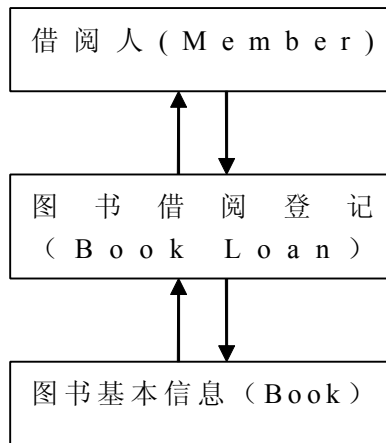


图 4.1

从图中可以看出，在这个系统中实际存在的实体：图书和借阅人，其中借阅人和图书是多对多关系，针对本系统，通过对图书借阅管理的内容和数据流程分析，设计数据项和数据结构如下：

- (1) 图书基本信息，其数据项有图书编号、图书名称、作者、出版社等。
- (2) 借阅人基本信息，其数据项有借阅人编号、借阅人姓名、电话等。
- (3) 图书借阅登记，其数据项有借阅序号、借阅图书编号、借阅人

编号等。

为了实现图书信息录入的方便性与规范性以及相关的统计功能，还应增加出版社信息

与图书分类信息：

(4) 出版社信息，其数据项有出版社编号、出版社名称、地址、电话、传真等。

(5) 图书分类信息，其数据项有分类编号、分类名称、同一类型图书数目。

同时针对于本系统的多用户使用特点，增加用户信息表：

(6) 用户信息表，其数据项有用户名、密码、用户权限。

为了实现图书借阅超期罚款制度，还应增设罚金规则表：

(7) 罚金规则表，其数据项包括免费使用天数、罚金费率、规则启用开关。

2 数据库逻辑结构设计

数据库设计有几个范式，一般我们要做到的是第三范式，即数据表中没有冗余字段以及同一个表中的字段没有函数依赖关系，冗余字段即在一个表中已经保存过的信息，在另一个表中就不应该存在，如果需要的话，可以通过表间的关联来得到，函数依赖性就是一个表中的字段间不应该有计算关系，如一个表中有单价字段、数量字段，就不应该有一个总金额字段。如果程序运行过程中需要总金额，可以实时计算。不过在一些较常用的表中，我们可以适当地保留冗余字段，这样，在程序运行过程中可以减少由于表间互相关联而使用速度降低等问题。这就是所谓的第四范式。数据表设计时，最好不要使用用户输入的信息作为主键，每一个数据表自己定义一个主键，添加信息是由程序自动添加，这样就可以减少数据更新时产生的错误。表与表相关联的外键最好是由程序自动生成的主键，这样数

数据库就比较规范了。

另外，数据表设计时一般都应该有一些标志字段，标志字段可以定义成 CHAR(1) 或 BIT

型。建议实际应用中定义成 CHAR(1) 字段可以存储多种可能的状态，在最初设计时，可能我们没有考虑到的一些情况，在程序后来的开发中，可以通过设计标志字段为不同的 值来解决，这样就避免了修改数据库结构。

数据库初期设计时一定要谨慎，把所有可能的情况都考虑进去，即使当时没有用到，也要将它留在数据库中作为备用字段以便将来扩充。

程序一旦开始编码，就应该尽量避免再修改数据库。因为如果数据库结构一旦改变，

所有与修改的数据表相关的业务都有可能受到影响，而某些影响还很难看到，这样就容易形成一个恶性循环。错误越改越多，越改越乱，最终导致程序的失败。PB 的数据窗口与其他语言的数据控件不一样，它的很多东西是预编译的。即使你一个模块已经调试无误，但只要数据库结构改动。相应的模块就一定要重新修改，否则一定会出问题。

图书借阅管理系统数据库中各个表的设计结果如下面几个表格所示。每个表格表示为数据库中的一个表。

借阅人基本信息表 memberinfo

说明：记录借阅人的基本情况，memberid 设为主键、索引。

表 4.1

列名	数据类型	长度	可否为空	注释
Memberid	Char	10	Not Null	借阅人编号
Membername	Char	10	Null	借阅人姓名
Id	varchar	20	Null	身份证号
Add	Char	50	Null	联系地址
Phone	Char	20	Null	联系电话

Mobile	Char	20	Null	移动电话
--------	------	----	------	------

用户基本信息表 people

说明：记录系统用户的用户名、密码与权限，name 设为主键。

表 4.2

列名	数据类型	长度	可否为空	注释
name	Char	10	Not Null	用户名
password	Char	10	Not Null	密码
level	char	1	Not Null	权限

图书基本信息表 bookinfo

说明：记录图书基本信息，bookid 设为主键、索引。

表 4.3

列名	数据类型	长度	可否为空	注释
Bookid	Char	10	Not null	图书编号
Isbn	Char	30	Null	图书 ISBN
Classname	Char	20	Null	图书类型
Title	Char	40	Null	图书名称
publishdate	date	-	Null	出版日期
author	Char	20	Null	作者
words	int	-	Null	字数（千字）
pub	Char	30	Null	出版社
price	numeric	19.4	Null	价格
disk	Char	1	Null	附带光盘
keyword	Char	30	Null	关键词
number	int	-	Null	存放代码
Abstract	varchar	100	Null	摘要
bookonloan	Char	1	Null	图书在借标志

				(Y-在借 N-未借)
--	--	--	--	-------------

出版社基本信息表 publishinfo

说明：记录出版社信息，pubid 设为主键。

表 4.4

列名	数据类型	长度	可否为空	注释
Pubid	char	4	Not Null	出版社编号
Pub	char	30	Null	出版社名称
Add	char	50	Null	联系地址
Zip	char	6	Null	邮政编码
Tel	char	16	Null	联系电话
Fax	char	16	Null	传真
http	char	16	Null	网址

分类信息表 bookclass

说明：记录定制的分类情况，classid 设为主键。

表 4.5

列名	数据类型	长度	可否为空	注释
Classid	Char	2	Not Null	分类编号
Classname	Char	20	Null	分类名称
Number	int	-	Null	数目

借阅情况信息表 loan

说明：进行借阅的登记，loanid 设为主键、自动增长。

表 4.6

列名	数据类型	长度	可否为空	注释
Loanid	Int	-	Not Null	借阅序号
Bookid	Char	10	Not Null	图书编号

memberid	Char	10	Not Null	借阅人编号
Loandate	date	-	Null	借阅日期

规则信息表 regular

说明：记录设置的借阅规则，days 设为主键。

表 4.7

列名	数据类型	长度	可否为空	注释
Days	Int	4	Not Null	免费借阅天数
Fine	Dec	5.2	Null	罚金标准
Flag	Char	1	Null	规则开关

(1) 规划有效的索引

- a. 在组合表的列中创建索引，包括主关键字和外部关键字所在的列。
- b. 在列或类组合中创建唯一的索引能增强唯一性。
- c. 浏览索引并卸载不使用的索引。索引需要一定硬盘空间和时间来维护。具有较高数据插入操作频率的数据库最好不要索引。有较高读操作频率的数据库应该有更多的索引。
- d. 避免在簇索引中包括不必要的列。在可能的情况下，使用较小的数据类型，例如用 varchar 替代 char。
- e. 考虑使用簇索引来支持排序和范围化查询。在为数据检索优化表时，簇索引必须支持数据的分组索引。为簇关键字选择列或列组，簇关键字以经常需要的顺序排序数据或以必须被一起访问的记录而分组记录。
- f. 创建支持一般查询的查找参数索引。具有高选择性的列是索引的最好候选列。具有高密度的列是索引糟糕的候选列。

(2) 使用约束实现数据的完整性

PRIMARY KEY 约束在表中定义了主关键字，它是行唯一的标识符，它可以强制实体完整性。在使用 PRIMARY KEY 约束时考虑以下

事实：

- a. 每个表只能有一个 PRIMARY KEY 约束。
- b. 键入的值必须是唯一的。
- c. 不允许有空值。
- d. PRIMARY KEY 约束在指定的列创建唯一的索引，可以指定簇索引和非簇索引（如果

非簇索引先前并不存在，簇索引是默认的）。

UNIQUE 约束指定，在一列中的两行不能有相同的值。该约束使用唯一的索引来强制实体的完整性。在已有一个主关键字时 UNIQUE 约束很有用，例如雇员号，但是必须保证其他标识符（例如，雇员驾驶证号）也是唯一的。在使用 UNIQUE 约束时，考虑以下事实：

- a. 允许有空值。
- b. 在一个表中可以设置多个 UNIQUE 约束。
- c. 可以将 UNIQUE 约束运用于具有唯一值的单列或多列，但不能用于表的主关键字。
- d. 通过在指定的列或列组中创建唯一的索引，可以使 UNIQUE 索引得到强制

五 图书管理系统的具体实现

1 PowerBuilder 开发工具简介

PowerBuilder 是目前开发 Client/Server 结构的应用系统的主要工具之一。PowerBuilder8.0 在继承与发展以前版本优秀功能的基础上,为适应网络应用程序的开发要求集成了 Enterprise Application Studio 中 PowerBuilder 以及 Power Site 的功能,在传统数据库开发的基础上,将 Web 应用更好地集成到 PowerBuilder 中。PowerBuilder 全面支持面向对象编程。是功能强大并易于使用的第 4 代编程语言(4GL),它内置了包括数据窗口在内的多种对象类,可以方便地访问数据库。相对于其他应用工具,PowerBuilder 具有工作效率高、成本低、质量高、功能强的特点。

2 Powerbuilder 8 应用程序开发的基本步骤

我们要开发应用程序时,首先要对它进行分析。无论哪种、哪方面的应用程序,都要先建立一个应用对象。下面我们介绍以下 PowerBuilder 8 应用程序开发的基本步骤:

- (1) 首先要建立应用对象。
- (2) 创建窗口。在窗口里放置各种控件和编写事件响应的脚本。
- (3) 创建菜单。窗口里的菜单可包括菜单条,下拉式菜单,级联菜单和弹出式菜单为菜单编写事件响应的脚本。
- (4) 创建用户对象。如果想要重复使用某个控件的功能,可以把窗口上经常放置的控件定义为用户对象。
- (5) 创建数据窗口。数据窗口可以检索数据库中的数据,可以建立

各种报或统计表，可以修改数据库。

(6) 创建函数、结构、事件。为了能够更好地支持脚本，编写自定义的函数，定义结构类型变量，也可以为对象和控件定义自己的事件。

(7) 运行与调试。可以在开发环境中随时运行应用程序，发现错误后，可以用调试工具进行调试。

(8) 当应用程序开发完毕后，可以把它编译成可执行的文件，让用户比较容易地建立应用系统的运行环境。

3 编码规范

为了减少在软件开发过程中的错误，在软件开发过程中应该遵守一定的标准。

给对象命名要有一定的规范，部件名称可以达到 40 个字符，窗口的命名：W_功能代码_功能描述。数据窗口的命名：DW_功能代码_功能描述。菜单命名：M_功能代码_功能描述。

标识符命名时，应该使标识符有一定的字面含义，有助于程序的调试和脚本的可读性的提高。本系统中使用的命名规范为：变量作用域+变量类型+”_”+具有一定字面含义的名称。例如：li_selectrow 反映出的含义：”l”代表是本地变量，是 local 的缩写，”i”代表是 integer 类型的变量，selectrow 表示该变量是用来记录一个行号的计数器。

4 创建祖先窗口和全局函数

为充分利用 PB 的面向对象的特性。程序开发时一般创建几个模板窗口。将功能窗口上的某些常用功能封装在模板窗口中。然后将这些模板窗口作为祖先窗口。所有的子孙窗口都可以通过继承的方法来生成。这样就减少了代码的书写量。使得整个程序界面保持整齐。当修改祖先窗口时，所有的子孙窗口都会自动修改。所以，祖先窗口的确定要十分谨慎。

全局函数与局部函数的作用类似。唯一不同的是全局函数的作用域是整个程序周期。不论你在任何一个模块的代码中都可以调用它。所以我们可以把某些常用功能写成全局函数。在程序的其他地方反复调用。

4.1 函数 level(character lev)

功能介绍：通过传入的 lev 值，将某些菜单项设为“非使能”，以限制某些功能的使用。Lev 值即不同职责的权限（系统管理 1、图书管理 2、借阅人管理 3、借阅操作员 4），从 people 表中的 level 字段中读取。

代码分析：

```
choose case lev
  case "1"
  case "2"
    m_main.m_借阅人管理.enabled=false
    m_main.m_图书租赁.enabled=false
    m_main.m_系统菜单.m_用户管理.enabled=false
    m_main.m_系统菜单.m_借阅规则设置.enabled=false
    m_main.m_系统菜单.m_图书信息设置.enabled=false
  case "3"
    m_main.m_图书管理.enabled=false
    m_main.m_图书租赁.enabled=false
    m_main.m_图书查询.enabled=false
    m_main.m_统计资料.enabled=false
    m_main.m_系统菜单.m_用户管理.enabled=false
    m_main.m_系统菜单.m_借阅规则设置.enabled=false
```

```

        m_main.m_系统菜单.m_图书信息设置.enabled=false
    case "4"
        m_main.m_图书管理.enabled=false
        m_main.m_借阅人管理.enabled=false
        m_main.m_系统菜单.m_用户管理.enabled=false
        m_main.m_系统菜单.m_借阅规则设置.enabled=false
        m_main.m_系统菜单.m_图书信息设置.enabled=false
    end choose
end choose

```

4.2 函数 countday

countday (date date1,date date2)

功能介绍：通过传入的 date1（起始日期）值与 date2（结束日期）值，计算出中间相差的天数，返回值为 integer 型。

代码分析：

```

int day
day=(integer(year(date2)) - integer(year(date1)))*365 +&
    (integer(month(date2)) - integer(month(date1)))*30 +&
    (integer(day(date2)) - integer(day(date1)))*1
return day

```

4.3 函数 countfine

countfine (integer days, decimal fine, integer countday)

功能介绍：根据图书免费借阅天数 days、罚金费率 fine, 以及已借阅的天数（由 countday 函数算出），计算出费用，返回值为 decimal

型。

代码分析：

```
dec countfine
if countday>=0 and countday<=days then//免费使用期内罚金
为 0
    countfine=0
else
    countfine=(countday - days)*fine//超根据期多少计算罚金
end if
return countfine
```

5 应用程序对象 App_librarian

功能介绍：PB 程序由一个应用程序开始，即每个 PB 程序在开始运行时，先执行应用程序对象的 Open 事件。在 Open 事件中连接数据库，并打开登陆窗口 w_login。

代码分析：

```
// Profile librarian
SQLCA.DBMS = "ODBC"
SQLCA.AutoCommit = False
SQLCA.DBParm
=
"ConnectString='DSN=librarian;UID=;PWD=' "
Connect using SQLCA;
//判断数据库连接是否成功
if sqlca.sqlcode = 0 then
    open(w_login)
else
```

```
messagebox('系统提示','连接数据库失败!')  
end if
```

6 具体窗口的实现

(1) 登录窗口 w_login

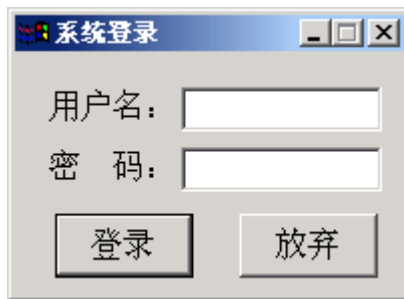


图 5.1

功能介绍：本窗口主要是检查操作员输入的用户名及密码是否正确，如果正确，允许登录。如果错误，显示出错误提示。

操作方法：填写“用户名”与“密码”后，点击“登录”按钮进行验证，点击“放弃”退出。

代码分析：

Open 事件：`sle_1.setfocus()`// “用户名”输入框获得焦点

```
this.width=923
```

```
this.height=588
```

“登录”按钮：`string name,pwd`

```
char lev
```

```
int test
```

```
name=sle_1.text
```

```
pwd=sle_2.text
```

```

//检测用户名与密码是否匹配
SELECT level
INTO :lev
FROM people
WHERE name=:name and password=:pwd;
if lev<>"" then
open(w_main)
level(lev)//根据权限设定菜单状态
else
sle_1.setfocus()
test=messagebox("提示：", "用户名或者密码错误！
", Exclamation!,
OKCancel!, 2)
end if//检测是否重新登录
if test=1 then
return 1
else
close(parent)
end if

```

(2) 主窗口 w_main

功能介绍：本窗口作为菜单及其他子窗口的容器。

窗口设置：本窗口为容器窗口，故它的 WindowType 为 mdihelp!, 表示本窗口为多文档界面，可以有菜单、工具栏与状态栏。WindowState 为 maximized!, 表示窗口在运行时是最大化的风格。MenuName 属性设为 m_main, 即是与本窗口连接的菜单名称。

(3) 菜单 m_main

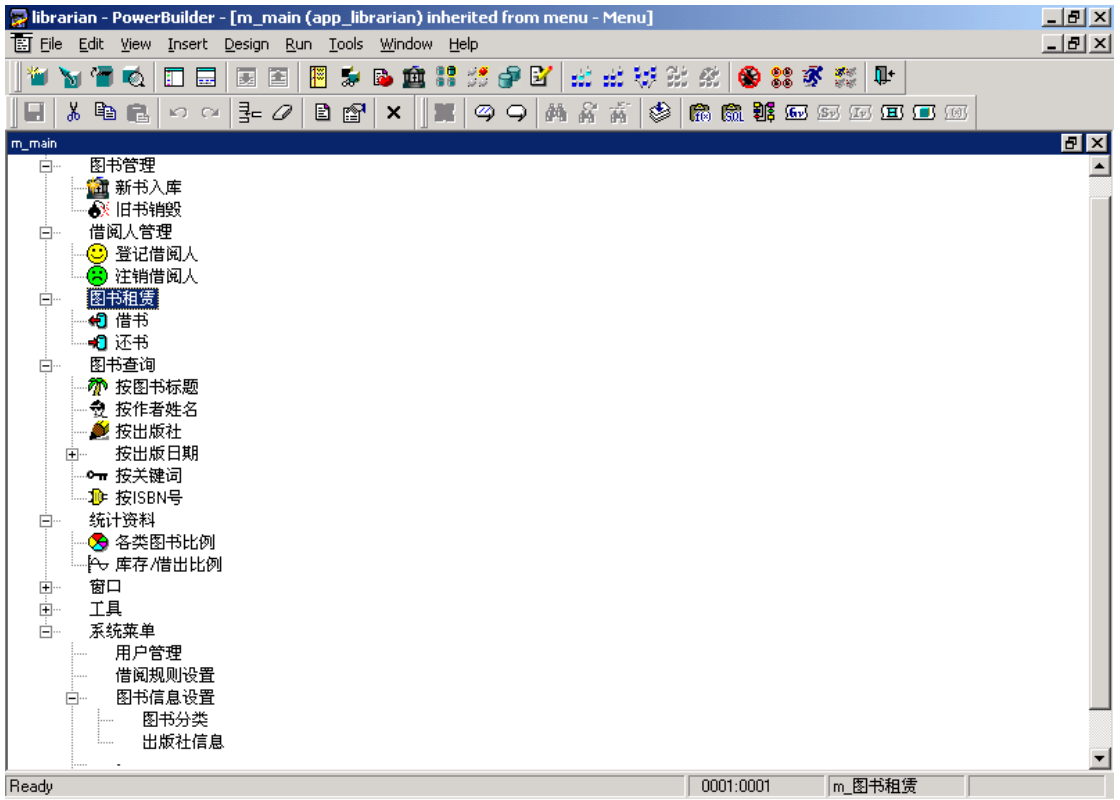


图 5.2

功能介绍：打开各功能窗口

操作方法：点击菜单项或工具栏上相应的按钮，状态栏可显示帮助信息。

代码分析：每个相应的菜单项的 clicked 事件都使用 opensheet 方法来打开相应的功能窗口，如“新书入库”的菜单项下的 clicked 事件的处理代码为：

```
opensheet(w_newbook, "w_newbook", parentwindow, 0, cascaded!
```

```
)
```

(4) 新书入库窗口 w_newbook

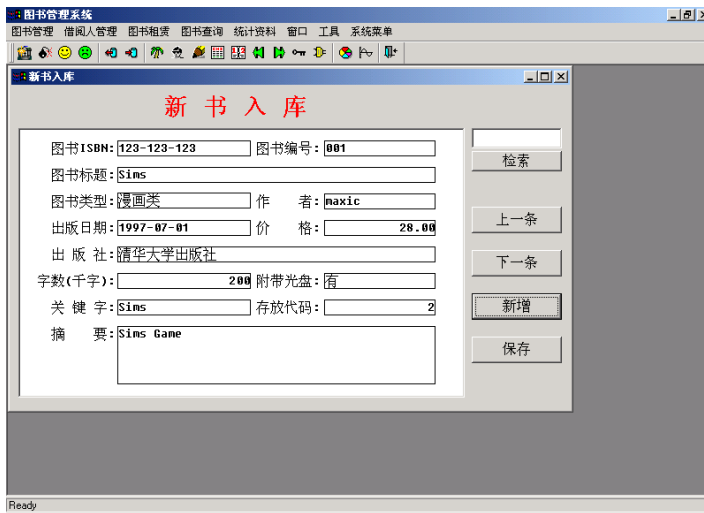


图 5.3

功能介绍：增加、修改图书信息。

操作方法：a. 增加图书信息：点击“新增”按钮，添加图书信息，完成后点击“保存”按钮

b. 修改图书信息：通过“图书编号”检索出图书信息，修改后点击“保存”按钮

代码分析：

Open 事件：`dw_1.settransobject(sqlca) //为数据窗口指定事务对象`

`dw_1.retrieve()//为数据窗口从数据库中获得数据`

`this.width=2926 //设置窗口的高度和宽度`

`this.height=1572`

“检索”按钮：`string ls_bookid,ls_filter`

`ls_bookid=sle_bookid.text`

`dw_1.setfilter("bookid='"+ls_bookid+"'")//设置过`

滤条件

`dw_1.filter()//过滤数据`

“上一条”按钮：`dw_1.scrollpriorrow()//滚动到上一行`

“下一条”按钮: `dw_1.scrollnextrow()` //滚动到下一行

“新增”按钮: `long ll_row`

```
ll_row=dw_1.insertrow(0) //在数据窗口中增加一行
dw_1.scrolltorow(ll_row) //数据窗口滚动到当前行
dw_1.setcolumn(2) //使第一列为当前列, 并得到焦点
dw_1.setfocus()
```

“保存”按钮: `long ll_row`

```
dw_1.accepttext() //将数据写入主缓冲区
if dw_1.modifiedcount()>0 then //检测数据是否被修
```

改

```
if dw_1.update()>0 then //如果被修改则更新数据库
    commit; //提交事务
```

```
    cb_new.setfocus() // “新增”按钮获得焦点
```

```
else
```

```
    rollback; //回滚事务
```

messagebox(“系统提示”,“保存失败! 请检验数据的
正确性”)

```
dw_1.setcolumn(1) //数据窗口第一列获得焦点
```

```
dw_1.setfocus()
```

```
end if
```

```
else
```

```
cb_new.setfocus() // “新增”按钮获得焦点
```

```
end if
```


(5) 旧书销毁窗口 w_delbook

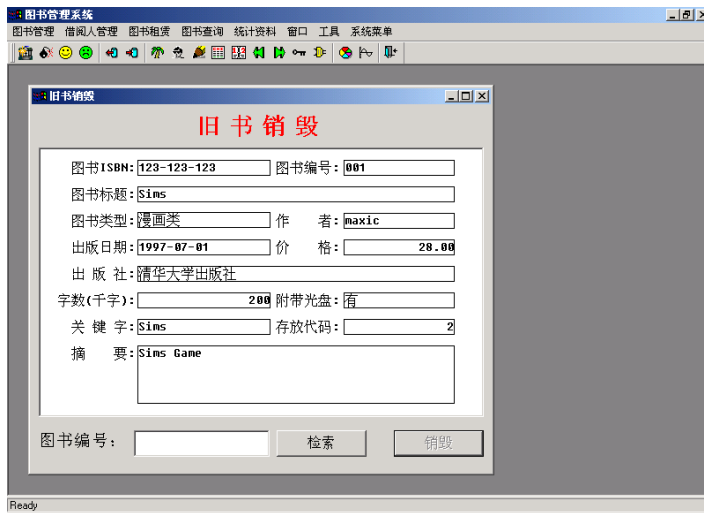


图 5.4

功能介绍：删除图书信息。

操作方法：首先通过图书编号检索出图书信息，然后点击“销毁”按钮注销图书。

代码分析：

Open 事件：类同“新书入库窗口 w_newbook”

“检索”按钮：string ls_bookid, ls_filter

```
ls_bookid=sle_bookid.text//设置过滤条件
```

```
dw_1.setfilter("bookid='"+ls_bookid+"'")//过滤数据
```

据

```
dw_1.filter()//使“销毁”按钮可用
```

```
cb_del.enabled=true
```

```
sle_bookid.setfocus()
```

“销毁”按钮：long li_row

```
li_row=dw_1.getrow()//得到当前行号
```

```
if li_row>0 then//判断是否有可删除的行
```

```
if messagebox('删除询问','真的要注销此书吗?')
```

```

        ',question!,Yesno!,2)=1 then
dw_1.deleterow(li_row) //删除行
if dw_1.update()>0 then//更新数据库
    commit;
else
    rollback;
    messagebox("错误提示","无法注销!")
end if
li_row=dw_1.getrow()//得到当前行
if li_row<dw_1.rowcount() then//判断是否至少有两行
    dw_1.selectrow(0,false) //高亮显示当前行
    dw_1.selectrow(li_row,true)
end if
end if
else
    messagebox("系统提示","没有要注销的数据")
end if
cb_del.enabled=false//使“销毁”按钮恢复不可用状态

```

存在

(6) 登记借阅人窗口 w_newmember

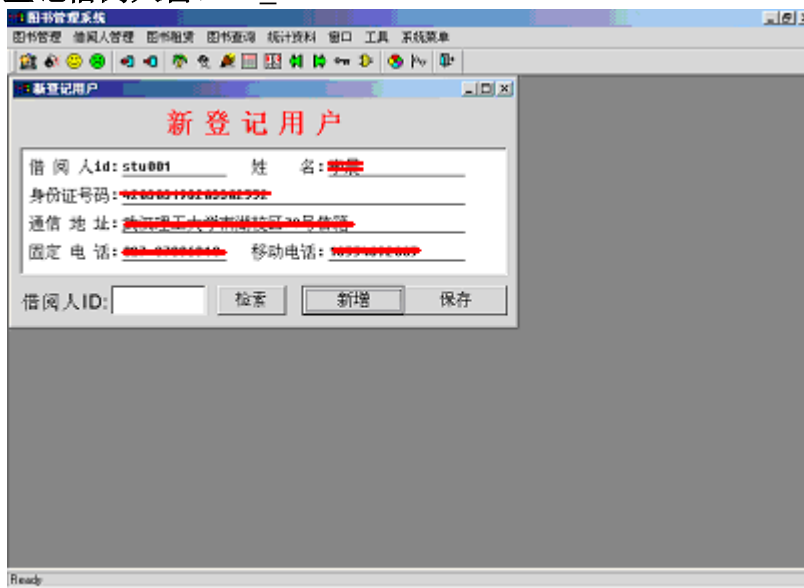


图 5.5

功能介绍：增加、修改借阅人信息

操作方法：类同“新书入库窗口 w_newbook”

代码分析：

Open 事件：类同“新书入库窗口 w_newbook”

“检索”按钮：类同“新书入库窗口 w_newbook”

“新增”按钮：类同“新书入库窗口 w_newbook”

“保存”按钮：类同“新书入库窗口 w_newbook”

(7) 注销借阅人窗口 w_delmember

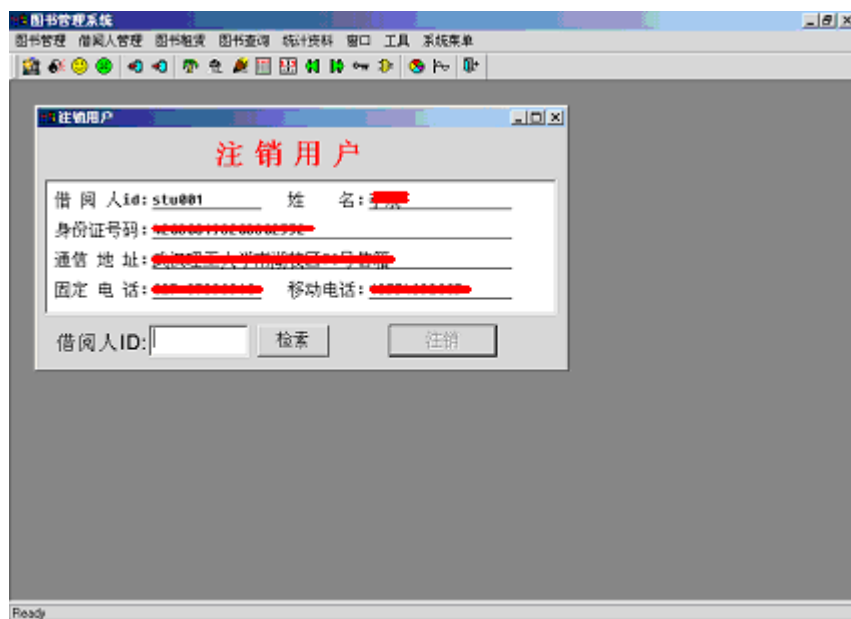


图 5.6

功能介绍：删除借阅人信息

操作方法：类同“旧书销毁窗口 w_delbook”

代码分析：

Open 事件：类同“旧书销毁窗口 w_delbook”

“检索”按钮：类同“旧书销毁窗口 w_delbook”

“销毁”按钮：类同“旧书销毁窗口 w_delbook”

(8) 借书窗口 w_loan

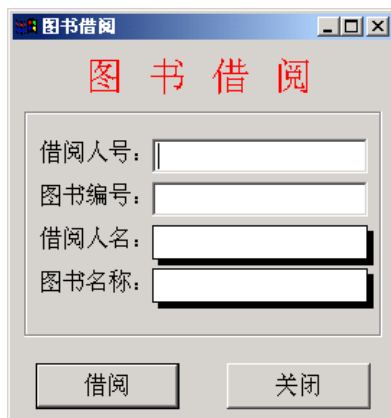


图 5.7

功能介绍：借阅图书，

操作方法：输入“借阅人编号”与“图书编号”后，借阅人姓名与图书编号将显示出来，然后点击“借阅”按钮借阅图书。

代码分析：

Open 事件：this.width=1243

```
this.height=1160
```

“借阅人号”输入框 modified 事件：

```
string ls_memberid,ls_membername
ls_memberid=trim(this.text)
select membername//根据“借阅人编号”查询出“借阅人姓名”
into :ls_membername
from memberinfo
where memberid=:ls_memberid;
if sqlca.sqlcode<>0 then//判断是否有该“借阅人”
    messagebox(“系统提示”,“没有此借阅人信息!”)
    sle_memberid.setfocus()//“借阅人编号”输入框获得焦点
return
else
    sle_bookid.setfocus()//“图书编号”输入框获得焦点
    sle_membername.text=ls_membername//显示“”借阅人姓名
end if
```

“图书编号”输入框 modified 事件：

```
string ls_bookid,ls_title
ls_bookid=trim(this.text)
```

```

select title
into :ls_title
from bookinfo
where bookid=:ls_bookid;
if sqlca.sqlcode<>0 then
messagebox("系统提示", "没有此图书信息!")
sle_bookid.setfocus()
return
else
sle_title.text=ls_title
end if

```

“借阅”按钮: string ls_bookid, ls_memberid, ls_bookonloan

```
date ldt_nowdate
```

```
ldt_nowdate=today()//获得“当前日期”
```

```
ls_bookid=trim(sle_bookid.text)//获得“借阅人号”
```

```
ls_memberid=trim(sle_memberid.text)//获得“图书编
```

号”

```
//强制输入“借阅人号”与“图书编号”
```

```
if ls_memberid="" or isnull(ls_memberid) then
```

```
messagebox("系统提示", "请输入借阅人编号!")
```

```
sle_memberid.setfocus()
```

```
return
```

```
end if
```

```
if ls_bookid="" or isnull(ls_bookid) then
```

```
messagebox("系统提示", "请输入图书编号!")
```

```
sle_bookid.setfocus()
```

```

return
end if
//根据“图书编号”查询图书是否借出
SELECT bookonloan INTO:ls_bookonloan
FROM bookinfo WHERE bookid=:ls_bookid;
if ls_bookonloan="y" then
messagebox("系统提示","本书已经外借!")
sle_bookid.setfocus()
return
else
//添加图书借阅记录
INSERT INTO loan
        (bookid,memberid,loandate)
VALUES (:ls_bookid,:ls_memberid,:ldt_nowdate);
commit;
//设置该图书为“借出”状态
UPDATE bookinfo set bookonloan='y'
WHERE bookid=:ls_bookid;
commit;
end if
//“借阅人号”得到焦点，所有输入框清空
sle_memberid.setfocus()
sle_bookid.text=""
sle_title.text=""
sle_memberid.text=""
sle_membername.text=""

```

“关闭”按钮: close(parent)

(9) 还书窗口 w_return

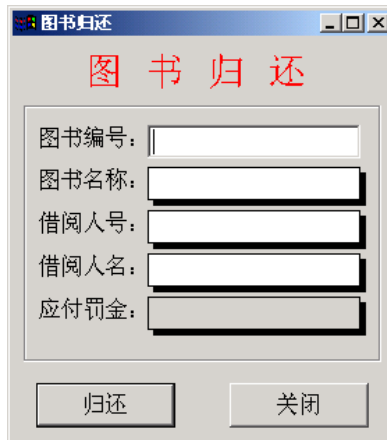


图 5.8

功能介绍: 归还图书

操作方法: 输入“图书编号”, 判断后显示出图书名称、借阅人编号、借阅人姓名。并根据是否启用了“罚金制度”(在“罚金规则窗口”设置), 决定是否计算罚金。

代码分析:

Open 事件:

```
char lc_flag
//查询“规则”是否启用, 并表现出来
select flag
into :lc_flag
from regular;
if lc_flag='y' then sle_fine.enabled=true
this.width=1285
this.height=1256
```

“图书编号”输入框 modified 事件:

```
string ls_bookid,ls_title,ls_memberid,ls_membername
date ld_datel,ld_date2
int li_days
dec ld_fine,ld_countfine
ld_date2=today()//获得当前日期
ls_bookid=trim(this.text)//获得“图书编号”
//根据“图书编号”查询出“借阅日期”
SELECT loandate
INTO :ld_datel
FROM loan
```



```

WHERE bookid=:ls_bookid;
//查询“免费天数”、“罚金规定”
SELECT days,fine
INTO :li_days,:ld_fine
FROM regular;
//计算罚金
ld_countfine=countfine(li_days,ld_fine,countday(ld_date1,ld_d
ate2))

//查询“图书名称”
SELECT title
INTO :ls_title
FROM bookinfo
WHERE bookid=:ls_bookid;
//查询“借阅人号”、“借阅人姓名”
SELECT memberid,membername
INTO :ls_memberid,:ls_membername
FROM memberinfo
WHERE memberid in
        (SELECT memberid
        FROM loan
        WHERE bookid=:ls_bookid);
if sqlca.sqlcode<>0 then
    messagebox("系统提示","没有借阅信息!")
    sle_bookid.setfocus()
    return
else
    sle_title.text=ls_title
    sle_memberid.text=ls_memberid
    sle_membername.text=ls_membername
    if sle_fine.enabled=true then
sle_fine.text=string(ld_countfine)
    end if
“归还”按钮: string ls_bookid,ls_bookonloan
    ls_bookid=sle_bookid.text
    if ls_bookid="" or isnull(ls_bookid) then
        messagebox("系统提示","请输入图书编号!")
        sle_bookid.setfocus()
        return
    else

```

```

select bookonloan into :ls_bookonloan
from bookinfo where bookid=:ls_bookid;
if ls_bookonloan="n" then
    messagebox("系统提示","本书已归还!")
else
    DELETE loan where bookid=:ls_bookid;
    update    bookinfo    set    bookonloan='n'    where
bookid=:ls_bookid;
    commit;
end if
end if
sle_bookid.setfocus()
sle_bookid.text=""
sle_title.text=""
sle_memberid.text=""
sle_membername.text=""
sle_fine.text=""
“关闭”按钮: close(parent)

```

(10) 图书查询

“按图书标题查询”窗口 w_search_title
“按作者姓名查询”窗口 w_search_author
“按出版社查询”窗口 w_search_pub
“按关键词查询”窗口 w_search_keyword
“按 ISBN 号查询”窗口 w_search_isbn
“按出版日期-某个阶段查询”窗口 w_search_date_between
“按出版日期-某个日期查询”窗口 w_search_date_date
“按出版日期-某日期之前查询”窗口 w_search_date_before
“按出版日期-某日期之后查询”窗口 w_search_date_after
说明：由于“按出版日期查询”使用的是 date 型数据，所以分开阐述。

其中，查询条件为 char 型：

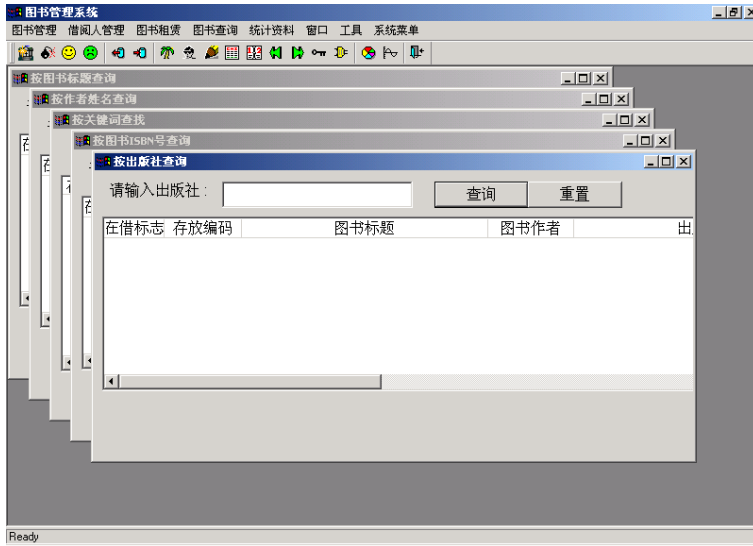


图 5.9

说明：属于此类的查询包括：“按图书标题查询”、“按作者姓名查询”、“按出版社查询”

“按关键词查询”、“按 ISBN 号查询”。由于“按图书标题查询”窗口 `w_search_title` 为模板窗口，其他窗口都是继承得来，所以仅分析这个窗口。

功能介绍：根据查询条件，检索出符合条件的数据

操作方法：输入检索条件，点击“查询”按钮进行查询，点击“重置”按钮重新输入查询条件。

代码分析：

Open 事件：`dw_1.settransobject(sqlca)`

`this.width=3616`

`this.height=1096`

“查询”按钮：`string ls_title`

`ls_title=trim(sle_find.text)`

`if sle_find.text="" then`

`messagebox("提示信息","请输入"+mid(st_1.text,4,4))`

`else`

`ls_title=""+ls_title+" "//设置检索条件`

`dw_1.retrieve(ls_title)`

`end if`

`sle_find.setfocus()`

“重置”按钮：`sle_find.text=""`

`sle_find.setfocus()`

数据窗口的数据源 SQL 语句：

`SELECT "bookinfo"."bookonloan",`

```

"bookinfo"."number",
"bookinfo"."title",
"bookinfo"."author",
"bookinfo"."pub",
"bookinfo"."publishdate",
"bookinfo"."isbn",
"bookinfo"."keywords",
"bookinfo"."abstract"
FROM "bookinfo"
WHERE "bookinfo"."title" like :query//:query 为查询变量

```

查询条件为 date 型:

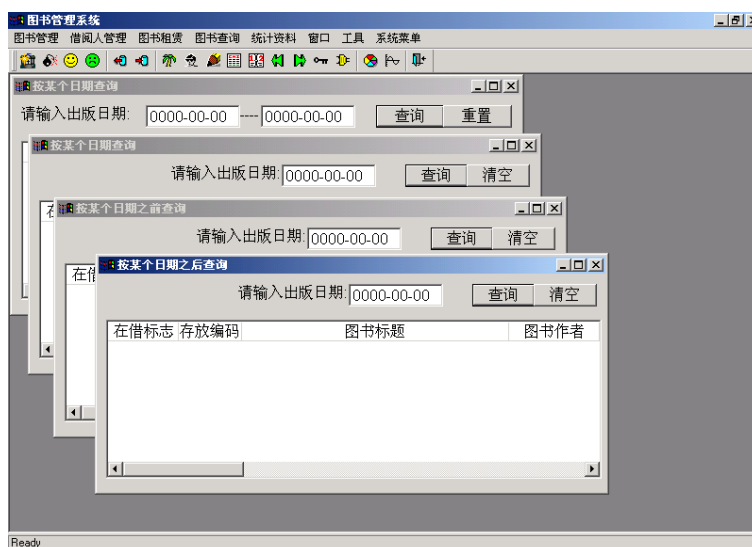


图 5.10

说明：属于此类查询的查询包括：“按出版日期-某个阶段查询”、“按出版日期-某个日期查询”、“按出版日期-某日期之前查询”、“按出版日期-某日期之后查询”。我们将对“按出版日期-某个阶段查询”窗口 w_search_date_between 进行分析。

功能介绍：根据查询条件，检索出符合条件的数据

操作方法：输入检索条件，点击“查询”按钮进行查询，点击“重置”按钮重新输入查询条件。

代码分析：

Open 事件：类同“按图书标题查询”窗口

“查询”按钮：date id_start, id_end

```
if isdate(em_date.text) then
    if isdate(em_date2.text) then
        id_start=date(em_date.text)
        id_end=date(em_date2.text)
    else
        messagebox("数据错误", "请重新输入结束日期.")
        em_date2.setfocus()
        return
    end if
else
    messagebox("数据错误", "请重新输入开始日期.")
    em_date.setfocus()
    return
end if
if id_start>id_end then
    messagebox("数据错误", "开始日期不能大于结束日期.")
else
    dw_1.retrieve(id_start, id_end)
end if
em_date.setfocus()
```

“重置”按钮：em_date.text=""

em_date2.text=""

em_date.setfocus()

数据窗口的数据源 SQL 语句:

```
SELECT "bookinfo"."bookonloan",  
       "bookinfo"."number",  
       "bookinfo"."title",  
       "bookinfo"."author",  
       "bookinfo"."pub",  
       "bookinfo"."publishdate",  
       "bookinfo"."isbn",  
       "bookinfo"."keywords",  
       "bookinfo"."abstract"  
FROM "bookinfo"  
WHERE "bookinfo"."publishdate" between :date1
```

and :date2

//:date1 与:date2 为查询变量

(11) 统计窗口

各类图书比例窗口 w_count_booktype、库存/借出比例窗口 w_count_loan

说明: 由于这两个窗口大同小异, 所以仅对 w_count_booktype 进行说明。

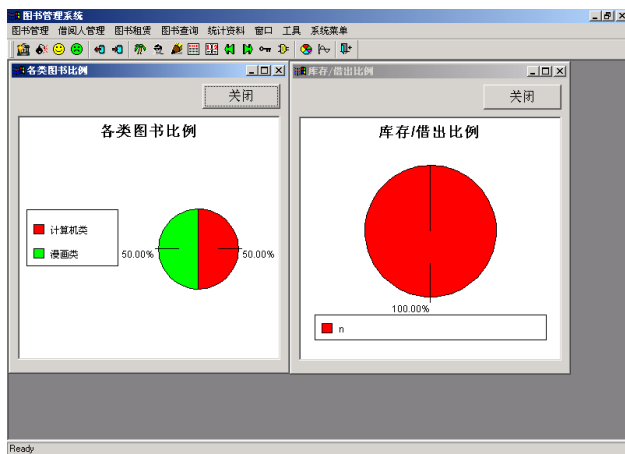


图 5.11

功能介绍：统计现有各类图书比例，并以饼型图显示出来。

数据窗口的数据源 SQL 语句：

```
SELECT "bookinfo"."bookonloan",  
       count(*)  
FROM "bookinfo"  
GROUP BY "bookinfo"."bookonloan"
```

(12) 用户管理窗口

w_people

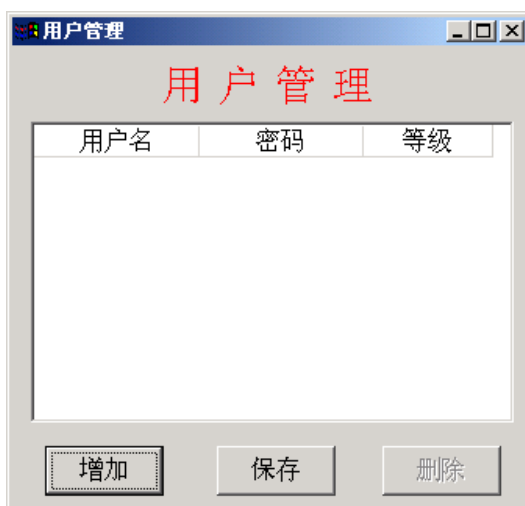


图 5.12

功能介绍：添加、删除、修改用户信息。

操作方法：a. 添加用户：点击“添加”按钮，输入用户名、密码、选择等级，点击“保存”

b. 删除用户：点击欲删除的用户，点击“删除”按钮

c. 修改用户：修改用户信息，点击“保存”按钮

代码分析：

Open 事件：类同“新书入库窗口 w_newbook”

数据窗口 rowfocuschanged 事件：`this.selectrow(0, false)` //高亮显示当前选中的行

```
        this.selectrow(currentrow, true)
```

数据窗口 clicked 事件：`cb_del.enabled=true` //点击数据窗口后使“删除”按钮可用

“增加”按钮：类同“新书入库窗口 w_newbook”

“保存”按钮：类同“新书入库窗口 w_newbook”

“删除”按钮：类同“旧书销毁窗口 w_delbook”

(13) 图书信息设置

图书分类设置窗口 w_bookclass、出版社信息窗口 w_pubinfo

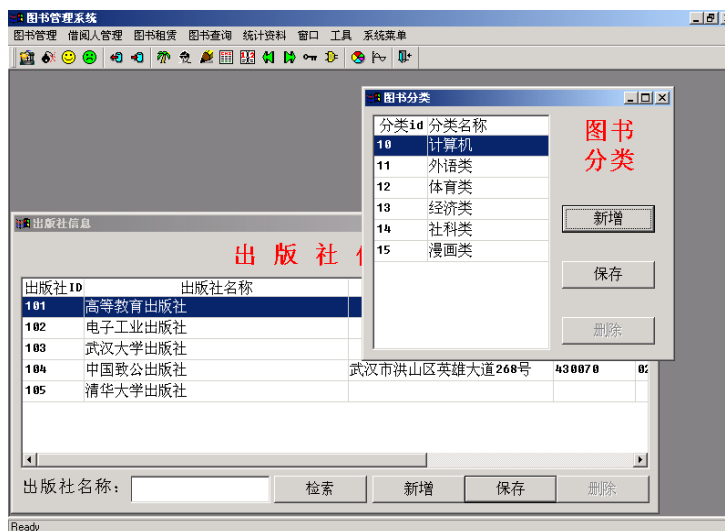


图 5.13

功能介绍：设置图书分类信息与出版社信息，同时可以对出版社信息进行查询

说明：设置“图书分类信息”是为了今后统计现有图书类型比例，设置“出版社信息”是为了今后查询出版社信息，同时在“新书入库”时方便相应项目的输入（下拉 Datwindow）

代码分析：略

(14) 借阅规则设置窗口

w_regular

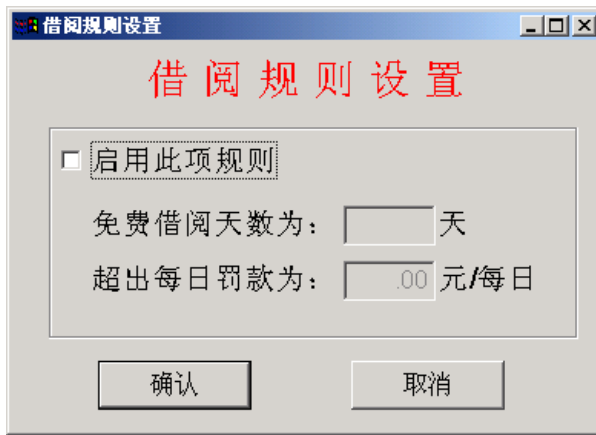


图 5.14

功能介绍：设置罚金规则

操作方法：填写“免费使用天数”、“罚金费率”，点击“启用规则”开启或关闭此项规则，点击“确定”实施规则。

代码分析：

Open 事件：int li_days

```
dec ld_fine
char lc_flag
//查询“免费天数”、“罚金”、“启用开关”
select Days, fine, flag
into :li_days, :ld_fine, :lc_flag
from regular;
sle_days.text=string(li_days)
em_fine.text=string(ld_fine)
//判断是否启用设置，并表现出来
if lc_flag='y' then
    cbx_regular.checked=true
```

```
sle_days.enabled=true
em_fine.enabled=true
end if
this.width=1787
this.height=1120
```

“启用”复选框: if this.checked then

```
    sle_days.enabled=true
    em_fine.enabled=true
else
    sle_days.enabled=false
    em_fine.enabled=false
end if
```

“确认”按钮: char lc_flag

```
int li_days
dec ld_fine
li_days=integer(sle_days.text)
ld_fine=dec(em_fine.text)
//判断是否启用规则
if cbx_regular.checked then
if sle_days.text="" or isnull(sle_days.text) then
    messagebox("系统提示","请输入免费借阅天数!")
    sle_days.setfocus()
return
end if
if em_fine.text=".00" or isnull(em_fine.text)
```

then

```

        messagebox("系统提示","请输入罚金!")
        em_fine.setfocus()

    return

    end if

//启用规则
    lc_flag='y'
else
//关闭规则
    lc_flag='n'
end if

//更新规则

UPDATE                regular                SET
Days=:li_days, fine=:ld_fine, Flag=:lc_flag;

        close(parent)

“取消”按钮: close(parent)

```

7 补充说明

(1) 窗口中的 Tab 顺序应符合实际操作习惯，并将合适的按钮设为 default

(2) 将“删除”按钮初始状态设为“非使能”，并在适当的时候通过程序使其“使能”，是为了防止误删。

(3) 图书借阅与归还窗口中部分输入框设为 DisplayOnly，仅做结果显示之用。“归还”

窗口中“应付罚金”输入框的使能与否由该项规则是否“启用”来定。

七 结束语

在对管理信息系统的设计过程有了基本了解后, 我使用 Visual Basic 6.0 设计了这个“图书管理系统”。系统基本实现了图书借阅管理的系统化、规范化和自动化。例如对图书进行登记, 对信息进行查询; 添加用户以及注销用户; 图书的借阅、归还; 可以按作者、出版社、出版时间等信息对图书进行查询, 并能反映出图书的借阅情况。实现了简单的统计分析功能, 可以展示出库存图书与借出图书的数量、馆藏图书种类的情况。并能对借阅规则进行设置。还能够对使用该管理系统的用户进行管理, 按照不同的工作职能提供不同的功能授权。

通过对该系统的设计、开发和调试, 一方面让我了解了系统的开发过程与方法, 另一方面, 让我初步掌握了 Visual Basic 开发工具的使用方法, 熟悉了窗口、菜单以及一些可视化控件的使用方法。通过使用还了解到窗口对象、数据窗口对象、各类控件的属性与方法。同时, 在制作过程中遇到了不少问题和困难, 通过查阅书籍及互联网得到了不小的帮助。

因为条件的限制, 该系统还存在不少的缺点和漏洞, 只能尽量改正与完善。不足之处, 还请多多指教。

参 考 文 献

- [1]. 罗超理, 李万红. 管理信息系统原理与应用. 清华大学出版社, 2002
- [2]. 艾德才. 计算机信息管理基础. 中国水利水电出版社, 2001
- [3]. Jeff Robin. Management of Information System. 机械工业出版社, 2002
- [4]. Tim F Lee. Implement MIS. 北京出版社, 2001
- [5]. 萨师煊, 王珊. 数据库系统概论. 高等教育出版社, 2000
- [6]. Efram G.Mallach. Decision Support and Data Warehouse Systems. 电子工业出版社, 2001
- [7]. Michael J.Hernandez. Database Design for Mere Mortals. 中国电力出版社, 2003
- [8]. Abraham Silberchatz. Database System Concepts. 机械工业出版社, 2003
- [9]. Graeme Malcolm. SQL Server 2000 with XML. 清华大学出版社, 2003
- [10]. 李晓喆, 张晓辉. SQL Server 2000 管理及应用系统开发. 人民邮电出版社, 2002
- [11]. 徐松林, 路斌, 王冬春. PowerBuilder 数据库应用开发教程. 清华大学出版社, 2003
- [12]. 东方人华, 仝丽娟. PowerBuilder 8.0 入门与提高. 清华大学出版社, 2003
- [13]. 郭兴成. PowerBuilder 8 教程. 北京希望电子出版社, 2002
- [14]. 王志勇, 阮坚, 王鹏. 用 PowerBuilder 开发小型数据库. 人民邮电出版社, 2002
- [15]. 周刚, 赵永军, 方小伟. PowerBuilder 项目开发实践. 中国铁道出版社, 2003
- [16]. <http://www.kejx.com/>

开题报告

经过大学四年理论课程的学习，以及校内校外的实践，极大的丰富了自身的理论基础，并且也具备了一些处理简单问题的能力，但即将踏上社会的我深知这还远远不够，因此我利用了毕业设计的机会，在老师的指导下去完成一个具有挑战性的，其具体应用及社会服务相结合的项目，深知通过对它的开发，将对我的能力有更高层次的突破。

开发软件的一个目的是针对目前普遍的图书管理系统存在的功能不全，操作复杂，系统要求高等一系列问题，而设想一个具有个性化的图书管理系统。该管理系统与 MIS 系统相联系，在图书馆内部建成可靠，方便，并且功能齐全的 MIS 系统。从而在图书馆对新书的反应；对书籍借阅的管理能力；对读者和图书馆工作人员的管理能力；对图书馆管理人员软件操作的适应时间和操作感觉这些方面都将大大的提高。当然对该软件的态度是渴望获得显著的社会效益。

开发软件的另一个目的是使其具有强大的实用价值，即它可以满足中小型图书馆的借阅与管理的需要。在一般的学校与科研机构，其下属的很多分支的研究中心，试验中心，各个学院，它们往往都有着自己的规模较大的图书资料室。而这些图书资料室由于本身规模不大因此其对书籍或资料的管理模式一般都比较旧，而且在没有能力也没有必要引进大型的图书馆管理软硬件的基础上，寻求一个针对中小型图书资料室的管理软件是必要的。

这种管理软件对硬件的要求很低，一般有一个比较简单的服务器与 PC 机组成的网络即可，再加上使用比较廉价、性能不错的软件，这样就可以以较低的成本来实现一个足够使用的功能，而这种模式也正满足了那种中小型资料室，图书馆的要求。但是目前对这种管理软件的开发还处于一种比较原始的阶段。开发者往往都是出于资金和时间的顾虑使用比较陈旧的技术，并且各为己见，并且很少涉及网络。可是当今是一个网络化的社会，像资料室，图书馆这样的信息机构不能与网络联系起来岂不可惜，再加上如今单机版的图书管理软件多如牛毛，去开发一个类似的软件无疑是一种在时间上、精神和资源上的极大浪费。所以出于多方面的考虑觉得有必要为这种有需要的中小型图书馆，资料室开发一种基于网络的图书资料管理软件。

项目的具体目的：

1. 实现图书馆对外借书,还书的简易操作，提高图书馆对最平凡工作的效率。
2. 实现图书馆对所藏图书的按类别，书名等多方面的查询，最大的方便读者和图书馆工作人员对所需图书的查询。
3. 建立图书馆外借读者数据库，方便工作人员对读者进行有效管理。
4. 建立图书馆工作人员数据库，限定每个工作人员对软件操作的权限，最大限度的保护数据库。
5. 实现图书馆对新书入库，旧书注销的简单处理，并且建立书籍档案，方便进货。

实现方法：

后台数据库支持:采用 PowerBuild 8.0,主要是由于 PowerBuild 8.0 是美国 Sybase 公司 PowerSoft 的新一代数据库应用开发工具,它能够设计传统的高性能、基于客户/服务器体系结构的应用系统,也能够用于开发基于 Internet 的应用系统。它具有容易使用,便于维护的优点,使使用部门可以节省培训费用,加快磨合周期,同时有利于操作人员的培训,是对数据的安全性要求不是特别高,同时又有比较大的数据量的单位的较好选择。作为 PowerBuild 7.0 的后续版本,PowerBuild 8.0 扩展了 PowerBuild 7.0 版的性能、可靠性、质量和易用性。PowerBuild 8.0 增加了几种新的功能,由此成为数据仓库和电子商务应用程序的优秀数据库平台。这些功能如:

- 新的开发调试界面,使程序员对于工作环境有了直观的了解,最大的方便了程序员的工作,新增的剪贴板能够让你保存最常用的代码段,避免了重复输入。
- 新的 Web 特性,这是 PB8.0 相对于 7.0 最大的改进,它提供了因特网浏览器中使用数据窗口控件的瘦客户端的解决方式。普通数据窗口和数据存储中可以使用的方法和属性现在大部分可以使用于 Web 数据窗口了。
- 新的数据库连接,在改进已有接口的前提下,使用新的数据库连接接口 jdbc/jbd 数据接口。它封装了 jdbc 函数功能于一个 PB 数据库接口,不仅支持使用 SUN 公司的 jdk/jre1.1,还支持使用微软的 java vm。因此它具有对现在 windows 操作平台更加好的兼容性。

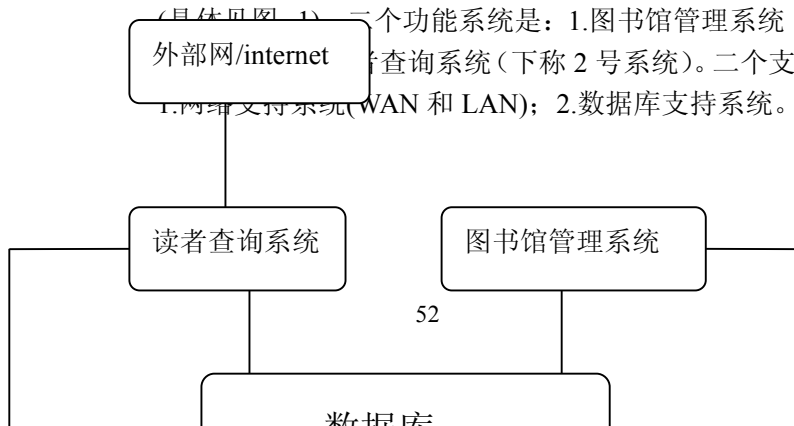
power builder 具有的独特的报表系统使制作表格变的极其方便,同时根据用户的需要可以随意调节表格的形式,而不象其他的语言,当用户的需要改变时,需要很大的工作量来更改程序适应用户的需要,同时,由于该语言具有良好的数据库接口,可以比较方便的进行数据库操作。虽然 power builder 的语法较为简单,对于算法和其他的一些复杂的表示方式缺乏有效的方法,但对于数据库运用,这些缺点并不是经常出现。因此,同时也用 power builder 作为前台客户端界面的实现语言。即整个系统均使用 power builder,以提高可靠性和安全性。

数据的安全性:

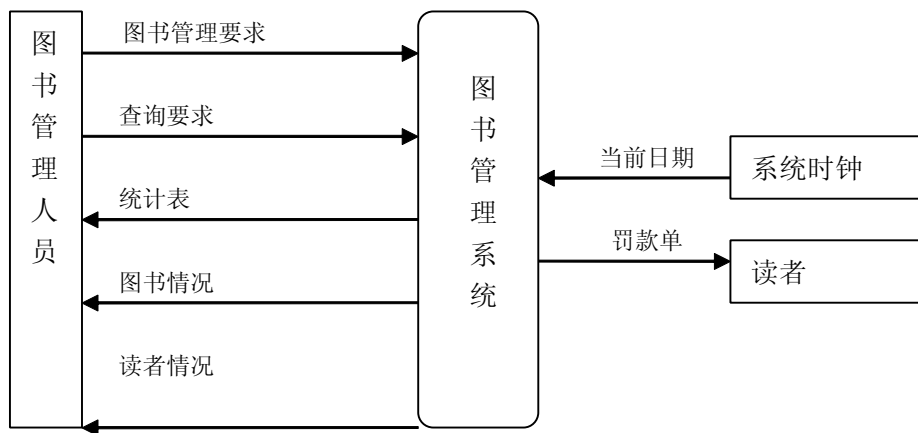
由于该系统统计管理单位的各种书目,人员数据及出入库数据,因此需要保证数据的可靠性和安全性。对于软件方面,采用手动或定时对数据库采取备份,使由于数据破坏所产生的损失降到最低,同时便于恢复。

总体功能结构:

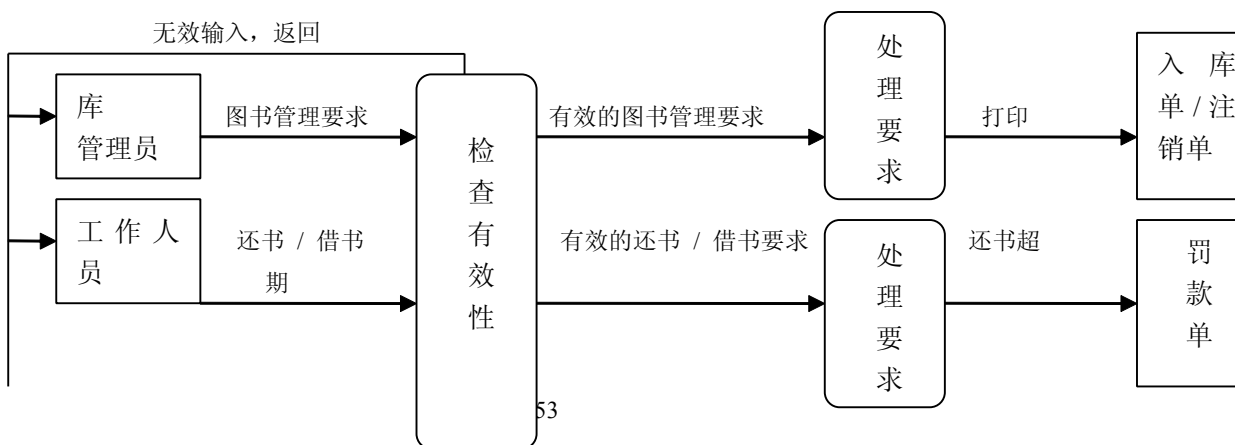
根据图书馆的基本要求和功能目标,总体框架是建立二个功能系统和二个支持系统(具体见图 1)。二个功能系统是:1.图书馆管理系统(下称 1 号系统)和读者查询系统(下称 2 号系统)。二个支持系统是:1.网络支持系统(WAN 和 LAN); 2.数据库支持系统。

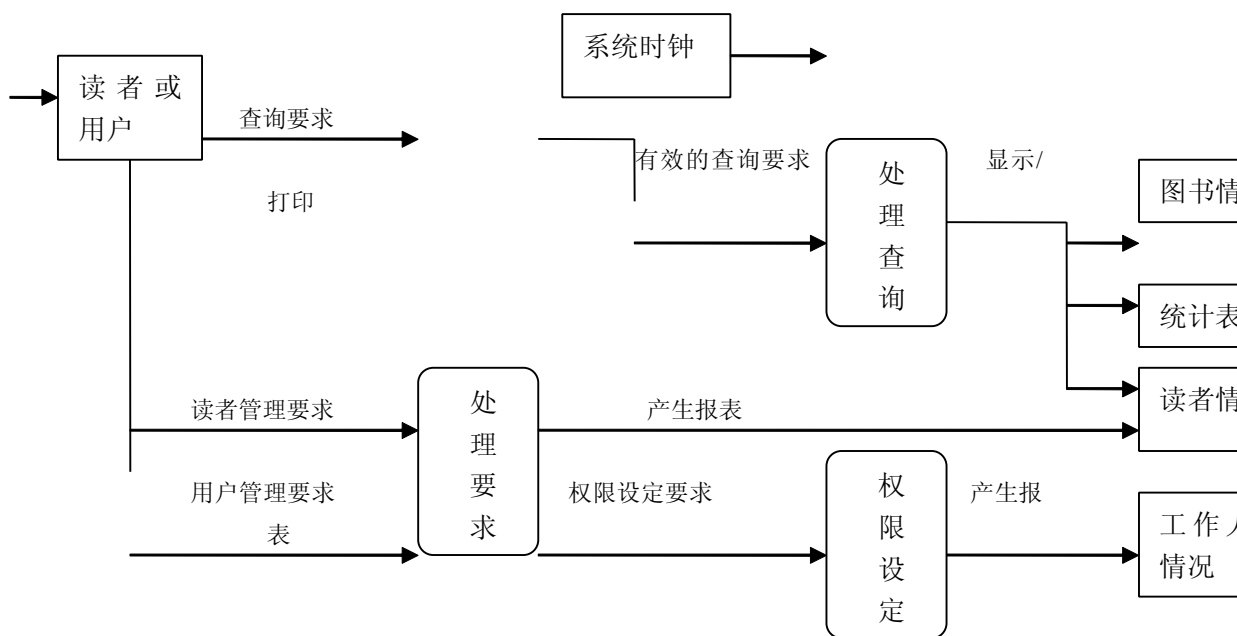


(图-1)



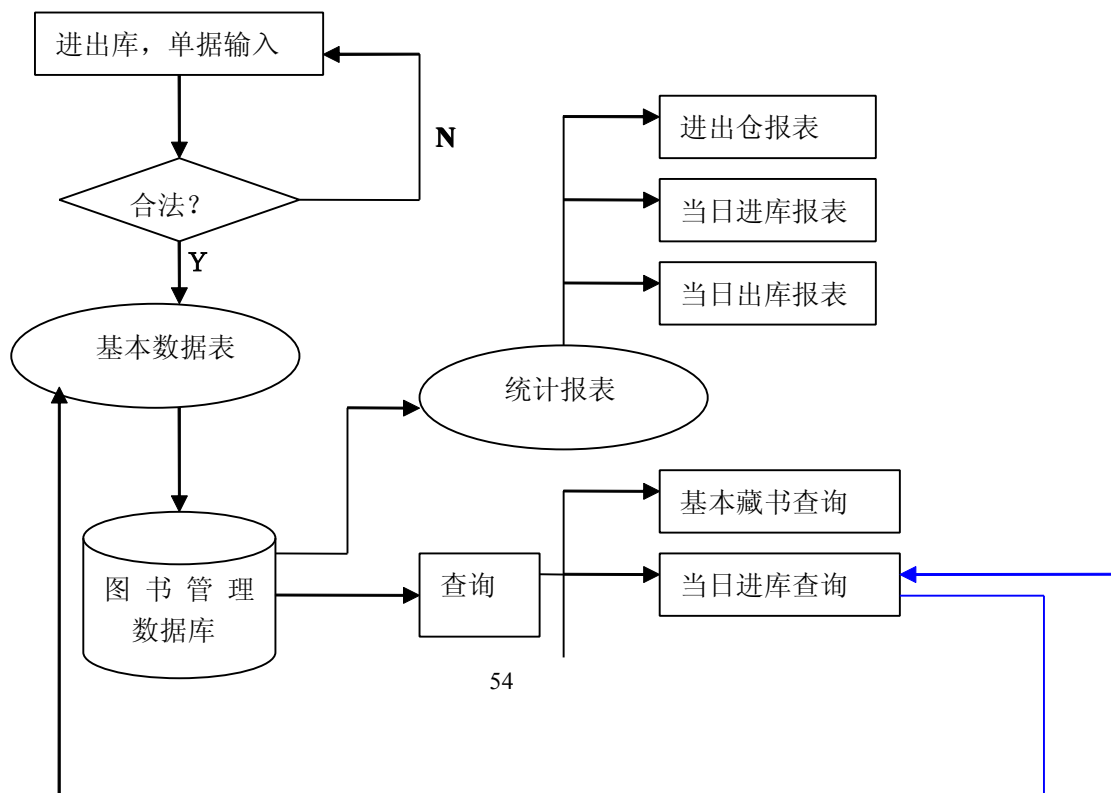
顶层数据流图

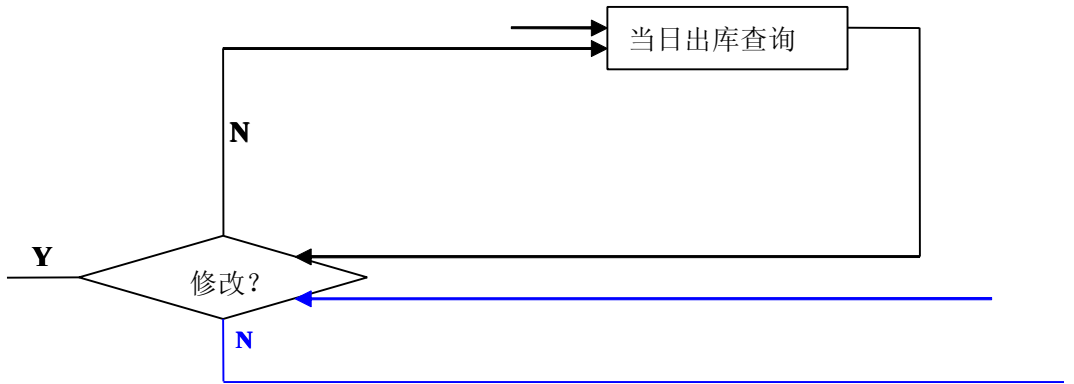




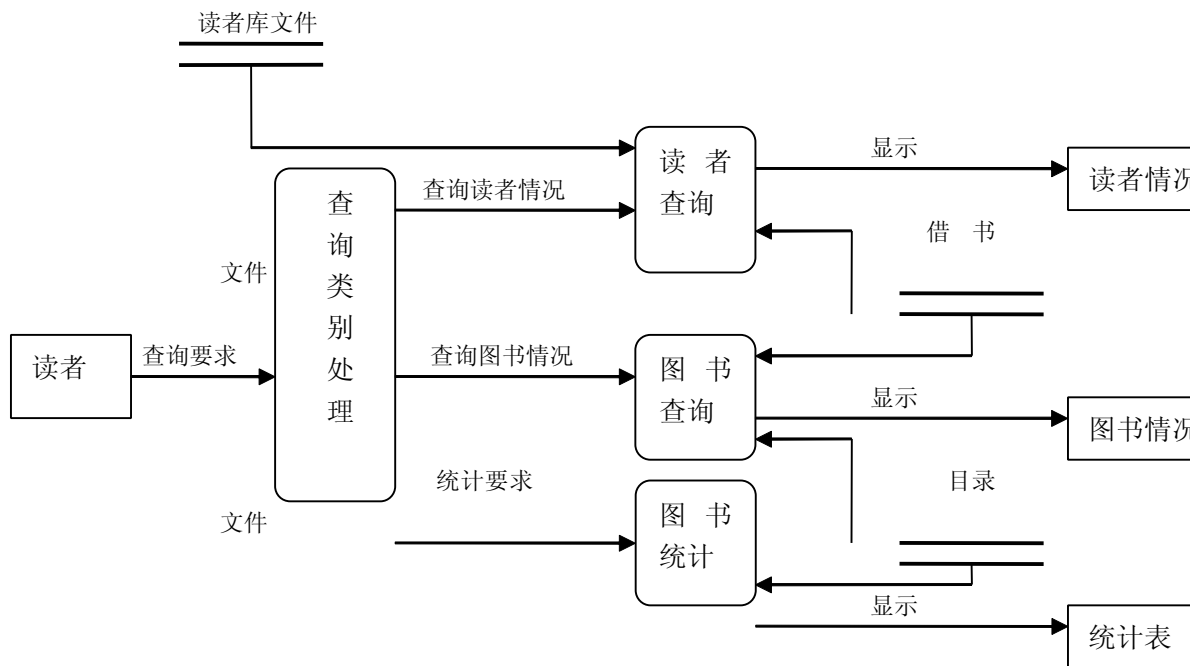
0层数据流图

(图—2)





(图-3)



(图-4)

需求分析:



图书馆管理系统即1号系统是该系统软件的重要内容之一,该子系统为图书馆的库存书籍的动态调整提供了决策依据,有利于及时对有大量需求的新书进行订货入库。它的主要功能包括:新书入库,旧书注销,借书,还书,馆内图书详细信息,图书查询,读者管理,馆内图书统计,读者信息管理,图书馆工作人员权限管理,名称设计,数据库备份等功能。

下面是2号系统的主要功能的简单描述（图—2简单数据流程图）：

1. 图书管理（图—3系统流程图）：

这是图书馆最重要的模块之一，它将对图书馆中的书籍进行管理操作，所有的原始数据输入都是在这里实现的，有了这里的数据输入，才能开始以后的诸如借书、还书等操作。

。新书入库管理：

该功能允许用户输入图书单，在输入过程中，程序自动生成图书单编号以加以区分两次不同的入库书单，同时程序也检查输入的合法性，并自动记录当日入库的图书。

。当日入库图书查询

该功能允许用户对当日的入库书单针对不同条件进行查询，并且提供二次修改，避免出错。

。旧书注销：

该功能允许用户输入注销单，在输入过程中，程序自动生成注销单编号以加以区分两次不同的注销书单，并且自动记录当日注销的图书。

。当日注销图书查询：

该功能允许用户对当日的注销单针对不同条件进行查询，并且提供还原，避免出错。

2. 业务管理：

这是图书馆使用频率最高的模块，因此他在功能上应该直观明确，在操作上应该简单易用。

。图书借阅：

该功能要求用户输入图书编号或者图书条码，并且检查输入合法性，然后程序动态生成该书的详细信息；再输入借书证号，并且检查输入合法性，然后程序动态生成该读者的详细信息；系统记录借书的经手人，最后由借书按钮实现操作。

。图书归还：

与图书借阅类似，追加图书超期罚款功能，自动根据预设值计算出罚款金额。

3. 人员档案管理：

这是针对读者和图书馆工作人员而设计的一个人士管理模块，它将完整的记录读者的信息，以及图书馆工作人员的信息，并且他还具有限制图书馆工作人员操作权限的功能，最大限度的保证了数据的安全。

。读者管理：

记录读者的详细信息，并提供增加、撤除、查询等操作。

。图书馆工作人员管理：

在读者管理子模块的基础上增加权限管理功能，保证数据安全。

4. 数据库备份：

这是一个涉及到安全问题的重要模块，它可以保证数据库的安全，防止因

外界的突发事件所造成的损失，避免因数据流失而重复输入的困扰。该功能分为自动备份，和手动立即备份两部分，满足用户需要。

5. 名称设计：

这是一个很具个性化的功能，实现很简单，但非常吸引人，你可以任意为你的管理系统命名，比如“杭州 X 中学图书管理系统”。

B:

外部读者查询系统，即 2 号系统，这是一个面向读者的综合查询系统，采用与管理系统相同的数据库，具有唯一的功能：查询。他可以按照读者的要求，用不同方法对所藏书籍进行查询，并能进行书名的模糊查询，还可以查询读者所借书的到期日，和观看读者本人的详细资料，力争最大的满足读者的需求。（图-4 简单数据流图）

性能要求：

对新书入库单，注销单的处理及查询；图书查询；借书还书速度应较快
登录仓库管理系统时应输入密码，以保证系统的安全性。

系统要有较强的兼容性，能在各种 PC 机上正常运行。在系统运行异常及数据文件损坏时能给出出错提示，保证系统的可靠性。

运行环境：

1. 推荐配置：

cpu: Intel Pentium 以上

内存: 64M

硬盘: 100M 空闲空间

2. 支持软件：

操作系统: Windows98 , Windows2000 , WindowsMe.

本人开发进度计划：

2 月—3 月底: 学习 PowerBuilder8.0, SQL Sever2000 技术, 看文档和有关书籍。

4 月—5 月初: 进行开发工具的选择与配置, 准备开发环境, 具体代码的编写。