**图书管理系统**

**软件工程A班**

**组员：莫剑涛（组长）17053103**

**林显泽**

**黄万涛**

**邝力通**

**分工表：**

|  |  |  |
| --- | --- | --- |
| **莫剑涛** | **后端+前端开发** | **25%** |
| **林显泽** | **文档+后端开发** | **25%** |
| **黄万涛** | **前端开发+整理** | **25%** |
| **邝力通** | **文档+数据库** | **25%** |

目 录

[摘 要 III](#_Toc231917716)

[1.图书借阅管理需求分析 3](#_Toc231917723)

[1.1 可行性分析 3](#_Toc231917724)

[1.1.1.技术可行性 3](#_Toc231917725)

[1.1.2.经济可行性 3](#_Toc231917726)

[1.2 图书借阅管理系统需求概述 3](#_Toc231917727)

[1.2.1系统目标 3](#_Toc231917728)

[1.2.2用户类和用户特性 4](#_Toc231917729)

[1.3 图书借阅管理系统需求模型 4](#_Toc231917730)

[1.3.1功能描述 4](#_Toc231917731)

[1.3.2图书管理员详细功能描述 5](#_Toc231917732)

[1.3.3读者详细功能描述 5](#_Toc231917733)

[1.3.4主要用例的用例描述 6](#_Toc231917734)

[2.总体设计 9](#_Toc231917735)

[2.1 数据库设计 9](#_Toc231917736)

[2.1.1数据库设计概述 9](#_Toc231917737)

[2.1.2图书信息表结构设计 10](#_Toc231917738)

[2.1.3图书类型信息表结构设计 11](#_Toc231917739)

[2.1.4 读者信息表结构设计 11](#_Toc231917740)

[2.1.5读者类型信息表结构设计 12](#_Toc231917741)

[2.1.6图书借阅信息表结构设计 12](#_Toc231917742)

[2.1.7图书归还信息表结构设计 13](#_Toc231917743)

[2.1.8用户信息表结构设计 13](#_Toc231917744)

[2.1.9图书馆信息表结构设计 14](#_Toc231917745)

[2.1.10办证参数信息表结构设计 14](#_Toc231917746)

[2.2 系统总体结构设计 15](#_Toc231917747)

[2.2.1图书管理系统总体结构图 15](#_Toc231917748)

[2.2.2 系统管理员模块功能 15](#_Toc231917749)

[2.2.3 读者管理模块功能 16](#_Toc231917750)

[2.2.4 图书管理模块功能 16](#_Toc231917751)

[2.2.5 图书借还模块功能 17](#_Toc231917752)

[2.2.6 系统查询模块功能 18](#_Toc231917753)

[3.程序设计与编码 20](#_Toc231917754)

[3.1开发平台与工具 20](#_Toc231917755)

[3.1.1 J2EE平台 20](#_Toc231917756)

3[.1.2 WEB服务器和数据库 20](#_Toc231917757)

3.[2程序设计 21](#_Toc231917758)

[3.2.1程序设计概述 21](#_Toc231917759)

[3.2.2数据库与Web服务器的连接 21](#_Toc231917760)

3[.2.3登录模块程序设计 23](#_Toc231917761)

[3.2.4系统管理员功能模块的实现 25](#_Toc231917762)

[3.2.5读者管理功能模块的实现 26](#_Toc231917763)

[3.2.6查询功能模块的实现 27](#_Toc231917764)

[3.2.7图书管理功能模块的实现 29](#_Toc231917765)

[3.2.8图书借还功能模块的实现 30](#_Toc231917766)

[4.软件测试 33](#_Toc231917767)

[4.1 软件测试的方法与步骤 33](#_Toc231917768)

[4.2 测试用例设计与测试用例的运行过程及测试结果分析 34](#_Toc231917769)

[4.2.1模块测试 34](#_Toc231917770)

[4.2.2集成测试 35](#_Toc231917771)

[4.2.3 验收测试 36](#_Toc231917772)

# 1.图书借阅管理需求分析

## 1.1 可行性分析

采用现代化统一的计算机信息网站系统，能够有效优化图书馆管理系统，使其高效的发挥最大作用，能够迅捷的为读者提供相应的服务。开发本系统的可行性研究如下：

### 1.1.1 技术可行性

技术上的可行性分析主要分析技术条件能否顺利完成开发工作，软、硬件能否满足需要。本系统采用JSP开发出友好美观的人机界面，便于用户理解、操作。数据库管理系统采用MySQL，它能够处理大量数据，同时保持数据的完整性、安全性。因此本系统的开发平台已成熟可行。硬件方面，在科技飞速发展的今天，硬件更新速度越来越快，容量越来越大，可靠性越来越高，价格越来越便宜，因此硬件平台也能够满足本系统所需[2]。

### 1.1.2.经济可行性

鉴于计算机技术发展异常迅猛,在硬件软件配置以及开发技术均以可行的情况下开发这样一个管理系统成本不会很高，但其可以大大提高图书馆的工作效率,也是图书馆管理发展的必然趋势,其必将有比较宽阔的市场,因此改性统在经济可行性上时可行的[2]。

## 1.2 图书借阅管理系统需求概述

### 1.2.1系统目标

该系统主要建立一个基于B/S模式的图书馆借阅管理系统,面对当起很多小型图书管理仍是人工管理带来的检索速度慢,效率低,借阅归还图书量大,图书统计工作量大,藏书不能完成及时更新的问题,该系统可以对跟系统的三个用户类型的使用实现:

1.对于读者在本系统的应用下可实现按照各种方式(如:书名,编号,作者)查询图书馆的藏书请客,方便的借阅图书,续借图书,归还图书,能够查询自己的借阅图书情况。

2.对于图书馆工作人员能够实现方便的对图书进行查询,方便的进行读者借阅情况查询,方便的进行借书还书处理等,便捷的对图书信息进行添加、修改、删除,分类管理等操作,对读者信息进行相关添加,修改,分类管理等操作。

3.对于系统管理员可以对图书馆信息进行修改更新操作,对系统用户进行添加、修改、删除、权限设置等操作,对图书馆的办证参数进行修改维护等操作功能。

### 1.2.2用户类和用户特性

图书借阅管理系统是一个基于B/S模式的对图书馆进行高效率管理的应用系统,它的用户主要是读者和图书管理员,学生通过该系统进行图书查询进而对自己需要的图书进行借阅及自己的借阅情况进行查询,图书管理员则通过本系统实现对图书及读者的高效管理,除此之外,还需要一个系统管理员对不同的用户进行权限的设置等操作[2]。

三类用户的具体描述如下表所示:

表2.1用户具体描述

|  |  |
| --- | --- |
| 用户类 | 描述 |
| 读者 | 读者是该系统的重要的使用角色,他们通过该系统查询自己需要的图书信息,并像图书管理员提出借阅图书的申请进而借阅自己所需的图书,还可以通过对自己借阅情况进行查询。 |
| 图书管理员 | 图书管理员是该系统的另一个重要使用者,图书管理员通过该系统进行图书的增加,修改,删除,分类管理等操作,实现对读者借阅归还续接图书的方便操作,实现对系统中图书,读者,读者借阅情况的查询,信息更改维护等操作,管理读者类型,对不同类型读者可借阅的图书数量进行设置等图书馆的基本操作。 |
| 系统员 | 系统管理员主要是图书管理系统中用户的管理,实现用户添加修改删除以及用户权限设置等操作，实现对图书馆基本信息的修改维护等操作,还包括对图书馆书架的设置操作,以及校外读者办证所需费用、证件有效期等参数的设置。 |

## 1.3 图书借阅管理系统需求模型

### 1.3.1功能描述

图书借阅管理系统的主要任务是实现读者迅速检索查询,方便借阅归还图书,图书管理员高效的完成系统的各项基本操作,系统管理员是管理用户设置权限等操作[3],从图2.1可以看出图书借阅管理系统要完成一下功能:

1. 登录。

读者、图书管理员,系统管理员进入该系统必须登录，身份验证正确了才可以进入该系统，以不同身份进入该系统所对应的系统使用权限是不同的。

2. 系统设置功能：

系统管理员可以设置图书馆相关的参数信息。

3.用户管理功能：

对系统用户进行添加,修改,删除,权限设置等操作。

4.查询功能：

对图书馆的馆藏图书,借阅历史,读者用户等信息进行查询。

5.其他功能

系统管理员可以修改自己的密码,并且拥有其他用户所拥有的所有功能。

下面的系统用例图描述了整个系统用户之间的动作联系,及功能模块的概述[4]。

 图2.1系统用例图

### 1.3.2图书管理员详细功能描述

1. 读者管理功能：

对读者的类型和读者档案进行管理,包括添加,修改,删除读者类型和读者用户的相关信息,管理不同类型读者借阅图书的数量。

1. 图书管理功能：

包括对图书类型和具体图书信息的管理,可以增加,修改,删除图书,丰富具体图书的信息,对不同图书进行分类操作。

1. 图书借阅功能：

可以完成对读者借阅,续接和还书的操作。

1. 系统查询功能：

查询图书相关资料,借阅历史和借阅到期题型.

1. 修改密码功能：

可以修改自己的登录密码。

### 1.3.3读者详细功能描述

1) 修改登陆密码：

修改自己的登录密码。

2) 查询功能：

对图书馆图书信息进行查询,对自己当前借阅书籍进行查询,对图书规划到期题型进行查看。

### 1.3.4主要用例的用例描述

图书借阅借阅管理系统涉及到的用例包括:图书借阅,图书归还,读书查询,读者信息管理,图书信息管理,用户管理等,现就系统的主要用例图书借阅,图书归还,图书查询进行详细分析。

1)用例“图书借阅”

|  |  |
| --- | --- |
| 用例名称 | 图书借阅 |
| 标识符 | UC-1 |
| 参与者 | 读者，图书管理员 |
| 描述 | 读书可以通过查询等方式获得自己想借阅的图书的名称,编号,等其他可唯一识别的信息,向图书管理员提出借阅请求,管理员在系统中记录相应信息,将图书交给读者,借阅成功。 |
| 前置条件 | 1. 登录；2. 进入图书借阅的页面 |
| 后置条件 | 1． 更新图书借阅列表； |
| 主干过程 | 1. 借阅图书 2. 读者请求借阅新地图书,并提供自己的编号 3. 系统显示读者借阅情况的表单 4. 读者提供想借阅的图书的标号 5. 系统存储读者和借阅的图书，并将之存储到数据库中 6. 系统更新借阅列表 |
| 分支过程 | 1.1 选择重置（第3步后分支出来）  1. 读者选择重置  2. 系统刷新该页面 |
| 异常 | 1.0.E.1 读者借阅已满（第2步）  1．读者借阅的数目已经到达自己借阅的上线  2. 系统将错误信息显示在借阅页面  3. 系统重新启动该用例 |
| 被包含用例 | 无 |
| 被扩展用例 | 无 |
| 优先级 | 高 |

2)用例“图书归还”

|  |  |
| --- | --- |
| 用例名称 | 图书归还 |
| 标识符 | UC-1 |
| 参与者 | 读者，图书管理员 |
| 描述 | 读者将自己借阅的图书归还图书馆。 |
| 前置条件 | 1. 登录；2. 进入图书归还页面 |
| 后置条件 | 1． 更新图书归还列表； |
| 主干过程 | 1. 图书归还 2. 读者请求归还借阅的图书并提供自己的编号 3. 系统显示该读者的借阅信息表 4. 使用者填将要归还的图书交给管理员 5. 管理员输入图书编号,系统存储归还信息，并将之存储到数据库中 6. 系统更图书归还列表 |
| 分支过程 | 1.1 选择重置（第3步后分支出来）  1. 读者选择重置  2. 系统刷新该页面 |
| 异常 | 1.0.E.1 读者借阅超时（第4步）  1．该书超出了应该归还的时间范围  2. 系统将罚款信息显示在归还页面  3. 需要缴纳罚金 |
| 被包含用例 | 无 |
| 被扩展用例 | 无 |
| 优先级 | 高 |

3)用例“图书查询”

|  |  |
| --- | --- |
| 用例名称 | 图书查询 |
| 标识符 | UC-3 |
| 参与者 | 读者,图书管理员 |
| 描述 | 读者通过图书的标号,名称等信息对相应的图书进行查询。 |
| 前置条件 | 1. 登录； 2. 转到图书查询页面 |
| 后置条件 | 1. 查询页面显示相应的图书的信息 |
| 主干过程 | * 1. 图书查询  1. 读者输入想要查询的图书信息 2. 系统显示相应的图书信息 3. 读者点击读书名称,跳转到图书详细信息链接页面 |
| 分支过程 | * 1. 输入信息时(第2步后)  1. 系统显示:请选择查询依据 2. 用户进行相应选择    1. 重新查询（第2步后） 3. 系统已经显示了相应图书信息 4. 读者想查询其他图书 |
| 异常 | 5.0.E.1 查询的图书不存在（第2步后）  1. 读者输入的图书信息不能在数据库中查询到  2. 系统显示暂时无该图书信息  3. 重新启动该用例 |
| 被包含用例 | 无 |
| 被扩展用例 | 无 |
| 优先级 | 高 |

# 2.总体设计

## 2.1 数据库设计

### 2.1.1数据库设计概述

数据库是整个系统的基石,数据库的设计优劣直接影响到整个系统的设计成败,本节对数据库的设计进行专门阐述[5]。

数据库是数据管理的最新技术。十多年来数据库管理系统已从专用的应用程序发展成为通用的系统软件。由于数据库具有数据结构化，最低冗余度，较高的程序与数据独立性，易于扩充，易于编制应用程序等优点，较大的信息系统都是建立在数据库设计之上的。因此不仅大型计算机及中小型计算机，甚至微型机都配有数据库管理系统[6]。

数据库系统的出现使信息系统从以加工数据的程序为中心转向围绕共享的数据库为中心的新阶段。这样既便于数据的集中管理，又有利于应用程序的研制和维护，提高了数据的利用性和相容性，提高了决策的可靠性。目前，数据库已经成为现代信息系统不可分割的重要组成部分。数据库技术也是计算机领域中发展最快的技术之一。

数据库设计是把现实世界的实体模型与需求转换成数据库的模型的过程，它是建立数据库应用系统的核心问题。数据库及其应用的性能都建立在良好的数据库设计的基础之上，数据库的数据是一切操作的基础，如果数据库设计不好，那么其它一切用于提高数据库性能的方法收效都是有限的。数据库设计的关键是如何使设计的数据库能合理地存储用户的数据，方便用户进行数据处理[6]。

设计数据库必须遵循一定的规则，在关系型数据库中，这种规则就是范式，范式是符合某一种级别的关系模式的集合。一般人们设计数据库遵循第三范式。即：数据库表中不包含已在其他表中包含的非主关键字信息。采用范式减少了数据冗余，节约了存储空间，同时加快了增、删、改的速度[6]。

整个系统所包括的信息有图书信息、读者信息、留言信息、图书借阅信息、图书归还信息、系统用户信息、读者类型信息。可将这些信息抽象为下列系统所需要的数据项和数据结构：

1)图书信息（编号，图书名称，图书类型，作者，译者，ISBN号，价格，出版社，所在书架，入库时间，操作员）

2)图书类型(编号,名称,可借阅天数)

3)读者信息（编号，姓名，性别，条形码，读者类型，出生年月，有效证件，证件号码,登记日期,电话,邮箱,操作员）

4)读者类型(名称,可借阅图书本数)

5)图书借阅信息（图书编号，读者ID，借出时间，应还时间，是否归还,操作员）

6)图书归还信息（图书编号，读者ID，归还时间，操作员）

7)用户(编号,用户名称,密码)

8)图书馆信息(编号,名称,馆长,电话,地址,邮箱,创建日期,简介)

图书馆参数信息(编号,办证费用,有效期限)

注：带下划线表示主键

在这里使用E-R图描述了图书借阅管理系统的数据模型。图3.1图书借阅管理系统E-R图描述了该系统所涉及到的实体以及他们之间的关系。具体结构如下图所示:

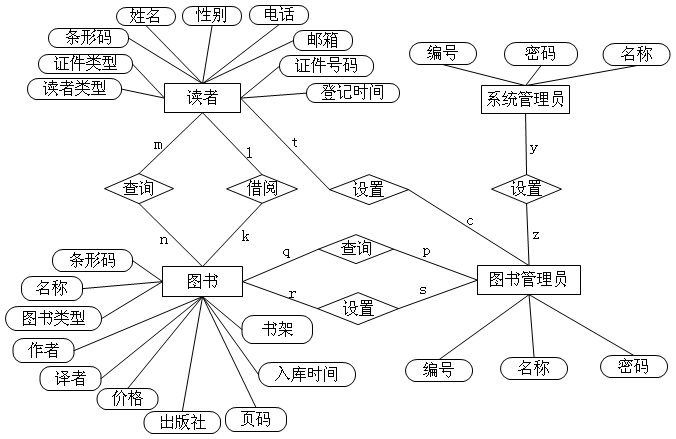


图3.1 系统E-R 图

### 2.1.2图书信息表结构设计

图书信息表主要用于存储图书馆中所藏图书的相关信息,其中的相关信息是在图书入库时由操作员进行添加完善,此表主要用于读者和图书管理员对馆中图书的查询,系统用户根据图书的某个属性进行查询,便可得知图书的其他相关信息,其中图书所在书架属性是便于读者借阅时对图书的寻找,图书价格是在读者不慎将图书遗失时对遗失图书进行赔偿的依据。表的具体结果如下:

表3.2图书信息表tb\_bookinfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| barcode | varchar | 30 | No | Key |
| bookname | varchar | 70 | No |  |
| author | varchar | 30 | Yes | 作者 |
| translator | varchar | 30 | Yes | 译者 |
| ISBN | varchar | 20 | No |  |
| price | float | 8,2 | No | 图书价格 |
| bookcase | varchar | 10 | No | 图书所在书架 |
| press | varchar | 70 | No | 出版社 |
| intime | date |  | No | 图书入库时间 |
| operator | varchar | 30 | No | 操作员 |
| page | int | 10 | Yes | 图书页码 |

### 2.1.3图书类型信息表结构设计

该表的设计主要是方便对图书的分类,和对图书的查询,在实际应用中图书管理员就是根据图书类型的不同将之分列在不同的书架,以方便读者的借阅寻找,其中可借阅天数项设置了不同类型图书的借阅期限,可根据图书的具体情况进行不同的维护管理,表的具体结构设计如下:

表3.3图书类型信息表tb\_booktype

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| ID | int | 10 | No | Key |
| bookname | varchar | 30 | No |  |
| days | int | 10 | No | 可借阅天数 |

### 2.1.4 读者信息表结构设计

读者信息表的设计是为了图书馆管理员对读者进行管理,其中读者ID,不同类型证件的号码都是唯一的,是读者在借阅图书时需要输入对读者身份进行识别的信息,读者邮件电话等信息室为了与读者进行联系,读者类型信息决定了读者一次性可借阅的图书的数量,注册时间可用于查询计算读者身份有效的期限,操作员是为了便于对信息才操作的查询。表的具体结构设计如下:

表3.4读者信息表tb\_reader

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| Id | int | 10 | No | Key |
| name | varchar | 20 | No |  |
| sex | varchar | 4 | No |  |
| barcode | varchar | 30 | No |  |
| readertype | varchar | 11 | No | 读者类型 |
| tel | varchar | 20 | Yes | 电话 |
| e-mail | varchar | 100 | Yes |  |
| paperType | varchar | 10 | No | 证件类型 |
| PaperNO. | Varchar | 20 | No | 证件号码 |
| birthday | date |  | Yes |  |
| intime | date |  | No | 登记时间 |
| operator | varchar | 30 | No | 操作员 |

### 2.1.5读者类型信息表结构设计

此表的设计是为了对不同身份的读者进行分类方便读者的管理,其中可借阅图书数量的属性设定,是根据读者需求的不同对起权限进行的设置,用于规定不同类型读者一次可借阅的图书数量,表的具体结构设计如下:

表3.5 读者类型信息表tb\_readertype

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| ID | int | 10 | No | Key |
| readername | varchar | 50 | No |  |
| number | int | 4 | No | 可借图书本数 |

### 2.1.6图书借阅信息表结构设计

该表的设计是用于对读者借阅图书进行管理,表中图书ID属性是对借阅图书的唯一性识别标识,读者ID号记录借阅的相应读者,结出时间记录了相应的归还时间,以及归还时是否超时,操作员是对借阅进行操作的人员的记录方便日后的查询,是否归还标识可查询当起读书是否被归还,表的具体结构设计如下:

表 3.6 图书借阅信息表 tb\_borrow

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| ID | int | 10 | No | Key |
| readerID | varchar | 10 | No |  |
| bookID | int | 10 | No |  |
| borrowTime | date |  | No | 结出时间 |
| backTime | date |  | No | 应归还时间 |
| operator | varchar | 30 | No | 操作员 |
| ifback | tinyint | 1 | No | 是否归还 |

### 2.1.7图书归还信息表结构设计

与图书借阅信息表形成对照的是图书归还信息表,该表的设计除了像上表一样把借阅的图书与相应的借阅者进行对应的联系以外还记录了读者应归还图书的时间,以此判断读者的借阅是否超时,表的具体结构设计如下:

表 3.7图书归还信息表tb\_giveback

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| ID | int | 10 | No | Key |
| readerID | varchar | 11 | No |  |
| bookID | int | 11 | No |  |
| backTime | date |  | No | 归还时间 |
| operator | varchar | 30 | No | 操作员 |

### 2.1.8用户信息表结构设计

该表的设计用于记录图书管理系统用户的信息,方便对用户的管理,表中包括用户ID 和用户的登录密码,表的具体结构设计如下:

表 3.8系统用户信息表tb\_user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| ID | int | 10 | No | Key |
| name | varchar | 30 | No |  |
| password | varchar | 30 | No |  |

### 2.1.9图书馆信息表结构设计

该表包含了图书馆的名称,地址,联系方式,建馆时间,简介等信息,是对图书馆基本属性信息的描述,方便外界读者对图书馆的了解,表的具体结构设计如下:

表3.9图书馆信息表 tb\_library

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| ID | int | 10 | No | Key |
| libraryname | varchar | 50 | No |  |
| curator | varchar | 10 | No |  |
| tel | varchar | 20 | Yes | 结出时间 |
| address | varchar | 100 | No | 应归还时间 |
| e-mail | varchar | 100 | No |  |
| url | varchar | 100 | No | 图书馆网站 |
| createDate | varchar |  | No | 创馆时间 |
| introduce | text |  | Yes | 图书馆简介 |

### 2.1.10办证参数信息表结构设计

该表是为校外人员设计的,对于需要到图书馆进行借阅的读者可办理临时的借阅证用于对图书的借阅凭证,表中包含了办证所需费用以及证件的有效期限,表的具体结果设计如下:

表3.10图书证参数信息表tb\_parameter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 说明 |
| ID | int | 10 | No | Key |
| Cost | int | 10 | No | 办证费用 |
| validity | int | 10 | No | 有效时间 |

## 2.2 系统总体结构设计

该系统在Windows98/2000/XP环境下，主要采用JSP开发工具，MySQL数据库来设计，开发过程与成果应符合GB/T 11457-1995软件工程术语，GB/T 8567-1988计算机软件产品开发文件编制指南等[7]。

### 2.2.1图书管理系统总体结构图

绘制系统结构图的过程，实际上就是对系统功能模块进行分解设计的过程，即合理地将数据流程图转变为所需要的系统结构图[8]。

系统结构图将会使读者和用户能直观的了解系统的结构模式，理解系统的各个功能的结构，能很好地方便用户使用和理解整个系统。

本系统的结构图如下：

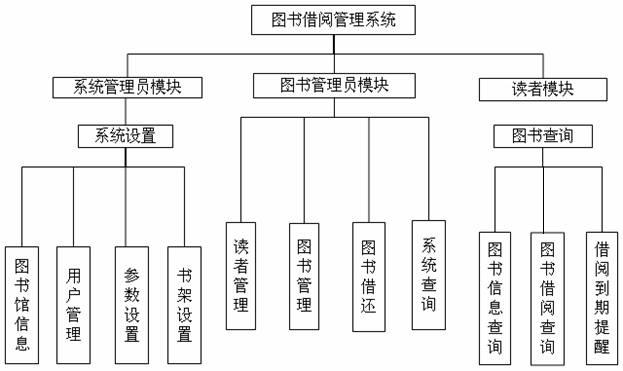


图3.3系统总体结构图

根据需求分析的结果，按照“低耦合、高内聚”的原则，本系统将划分为以下主要功能模块：系统管理员功能模块，读者管理功能模块，图书管理功能模块，图书借还功能模块；系统查询功能模块[8]。

### 2.2.2 系统管理员模块功能

该模块主要包括图书馆信息设置,用户管理,参数设置,书架设置。

1.图书馆信息设置

该功能选项用于系统管理员对图书馆名称,地址,联系方式,简介等信息的管理更新,以便于读者和外界人士对图书馆的了解。该功能是对td\_library表进行维护修改等操作,修改后的信息将被保存在该表中。

2.用户设置

该功能子模块主要是系统管理员对系统用户的管理,通过此子模块的功能实现可以对用户进行添加、修改、删除、权限设置等操作,该子模块能将图书馆的不同工作细化到不同的相关人员,极大地提高了图书馆的工作效率。该操作是对tb\_users表进行操作,在对用户进行了相关的操作后把操作后的最新信息存放在该表中。

3.图书馆参数设置

通过该子模块设置在图书馆办理临时读者证的费用及证件有效期限。该操作是对于tb\_parameter表进行,并把操作后的最新数据存放在该表中。

### 2.2.3 读者管理模块功能

该模块主要包含读者类型管理和读者信息管理两个子模块:

1. 读者类型管理

该子模块是对图书馆系统用户读者的类型进行维护,修改等操作,在此模块中主要设置不同类型读者一次性可借阅的图书的数量,该操作是对于tb\_resderType表进行,并将操作结果保存在该表中。

1. 读者信息管理

该子模块是对读者的基本信息进行管理,可以对读者的基本信息进行添加,修改,删除操作,这下操作均是对tb\_resder表进行的,并将操作后的结果保存在该表中。

### 2.2.4 图书管理模块功能

图书管理功能模块可分为图书类型管理和图书信息管理两个子模块,其各自的实现分别如下面表中所示:

表3.11 图书类型管理描述

|  |  |
| --- | --- |
|  | 图书类型管理 |
| 功能描述 | 对图书进行类型设置分类,并对不同类型图书可被借阅的天数进行设置 |
| 访问的数据库表 | 图书类型表:tb\_bookType |
| 进行的操作 | 添加、修改、删除图书类型,对不同类型图书可被借阅的天数进行设置 |
| 产生的结果 | 对图书类型进行管理,对不同类型图书参数进行设置 |
| 结果存储位置或输出 | 结果存储在图书类型表(tb\_bookType)中,结果在图书类型查询页面输出 |

表3.12 图书信息管理描述

|  |  |
| --- | --- |
|  | 图书信息管理 |
| 功能描述 | 对图书进行基本操作和信息管理 |
| 访问的数据库表 | 图书类型表:tb\_bookType |
| 进行的操作 | 添加、修改、删除图书,对图书的编号、所在书架、价格、出版社等基本信息进行管理 |
| 产生的结果 | 对图书基本操作管理,对不同图书参数进行各自信息的设置管理 |
| 结果存储位置或输出 | 结果存储在图书类型表(tb\_book)中,结果在图书查询页面输出 |

### 2.2.5 图书借还模块功能

该功能模块主要实现对读者借阅、续接、归还图书的操作,其中子模块各自的描述如下各表所列:

表3.13 图书借阅描述

|  |  |
| --- | --- |
|  | 图书借阅管理 |
| 功能描述 | 对读者借阅图书进行基本操作和信息管理 |
| 访问的数据库表 | 图书信息表: tb\_bookinfo  读者信息表:tb\_reader  读者类型信息表:tb\_resderType |
| 进行的操作 | 对读者借阅图书进行管理 |
| 产生的结果 | 读者借阅成功,系统对借阅信息进行记录 |
| 结果存储位置或输出 | 结果存储在图书借阅表(tb\_borrow)中,结果在图书借阅查询页面输出 |

表3.14 图书续借描述

|  |  |
| --- | --- |
|  | 图书续接管理 |
| 功能描述 | 对读者借阅图书进行提续接操作 |
| 访问的数据库表 | 图书借阅表: tb\_borrow |
| 进行的操作 | 对借阅的图书进行续接 |
| 产生的结果 | 读者归还日期延后一个月 |
| 结果存储位置或输出 | 结果存储在图书借阅表(tb\_borrow)中 |

表3.15图书归还描述

|  |  |
| --- | --- |
|  | 图书归还管理 |
| 功能描述 | 对读者归还图书进行基本操作和信息管理 |
| 访问的数据库表 | 图书借阅信息表: tb\_borrow  读者类型信息表:tb\_resderType |
| 进行的操作 | 对读者借阅图书进行管理 |
| 产生的结果 | 读者借阅成功,系统对借阅信息进行记录 |
| 结果存储位置或输出 | 结果存储在图书归还表(tb\_giveback)中 |

### 2.2.6 系统查询模块功能

该模块包括对图书馆藏书进行查询,对读者借阅情况进行查询,以及对借阅到期和超期的读者进行提醒的信息,其中三个子模块的各自实习如下所示:

表3.16 图书查询描述

|  |  |
| --- | --- |
|  | 图书查询 |
| 功能描述 | 系统用户对馆藏图书信息进行查询操作 |
| 访问的数据库表 | 图书信息表: tb\_book |
| 进行的操作 | 用户通过图书的编号,作者,出版社等信息对图书进行相关查询 |
| 产生的结果 | 读者查询到相应的图书或系统提醒查询的图书不存在 |
| 结果存储位置或输出 | 结果在图书查询页面输出 |

表3.17 图书借阅查询描述

|  |  |
| --- | --- |
|  | 图书借阅查询 |
| 功能描述 | 系统用户对读者借阅图书信息进行查询操作 |
| 访问的数据库表 | 图书借阅表: tb\_borrow |
| 进行的操作 | 用户通过图书的编号,读者编号等信息对摸个读者或某本图书的借阅情况进行相关查询 |
| 产生的结果 | 查询到相应的读者或图书得借阅情况 |
| 结果存储位置或输出 | 结果在图书查询页面输出 |

表3.18 图书借阅到期提醒描述

|  |  |
| --- | --- |
|  | 图书借阅到期提醒管理 |
| 功能描述 | 对读者借阅的到期图书进行提醒 |
| 访问的数据库表 | 图书借阅表: tb\_borrow  图书归还表:tb\_giveback  读者信息表:tb\_reader  读者类型信息表:tb\_resderType |
| 进行的操作 | 对借阅到期和超期的读者进行提醒 |
| 产生的结果 | 向借阅到期和借阅超期的读者发送邮件等提醒信息 |
| 结果存储位置或输出 | 结果存储在图书借阅到期提醒表 |

# 3.程序设计与编码

## 3.1开发平台与工具

### 3.1.1 J2EE平台

J2EE ，即是Java2平台企业版（Java 2 Platform Enterprise Edition），是原Sun公司（现已被甲骨文公司收购）为企业级应用推出的标准平台。它简化了企业解决方案的开发、部署和管理相关复杂问题的体系结构，J2EE技术的基础就是核心Java平台或Java 2平台的标准版，J2EE不仅巩固了标准版中的许多优点，例如"编写一次、随处运行"的特性、方便存取数据库的JDBC API、CORBA技术以及能够在Internet应用中保护数据的安全模式等等，同时还提供了对 EJB（Enterprise JavaBeans）、Java Servlets API、JSP（Java Server Pages）以及XML技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。J2EE体系结构提供中间层集成框架用来满足无需太多费用而又需要高可用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发平台，J2EE降低了开发多层应用的费用和复杂性，同时提供对现有应用程序集成强有力支持，完全支持Enterprise JavaBeans，有良好的向导支持打包和部署应用，添加目录支持，增强了安全机制，提高了性能[9]。

在开发图书馆借阅管理系统的过程中，应用Myeclipse6.0.1，它可以在数据库和J2EE的开发、发布，以及应用程序服务器的整合方面极大的提高工作效率。Myeclipse是功能丰富的J2EE集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持HTML, Struts, JSF, CSS, Javascript, SQL[10]。

### 3.1.2 WEB服务器和数据库

在系统的开发过程中使用的Web应用服务器是Tomcat，是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由Apache、SUN和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现。Tomcat是一个小型的轻量级应用服务器，它运行时占用的系统资源小、扩展性好、支持负载平衡和邮件服务等开发应用系统常用的功能，因此在中小型系统和并发访问用户不是很多的时候，经常被使用[11]。

使用MySQL作为数据库开发工具。MySQL是一个小型关系型数据库管理系统，开发者为瑞典MySQL AB公司。在2008年1月16号被Sun公司收购。目前MySQL被广泛地应用在Internet上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库[11]。

为了开发的便捷快速，使用struts第三方插件。Struts是一个全新的MVC框架，实在WebWork基础上发展起来的[12]。

开发系统用的系统工具如表4.1所示:

表4.1 系统开发平台和工具

|  |  |
| --- | --- |
| 工具名称 | 用途 |
| JDK 1.6.0\_11  MyEclipse  MySQL 5.0  SQLYog 6.1  Tomcat 6.0  Struts 1.0 | Java 开发工具包  J2EE集成开发环境  小型关系数据库管理系统  MySQL图形化数据库管理工具  Web应用服务器  第三方插件，可扩展的Java EE Web框架 |

## 3.2程序设计

### 3.2.1程序设计概述

在设计的Web层应用了著名的MVC模式,V有JSP来实现,为了业务逻辑和表示的分离.它是基于Web应用系统,它的客户端使用Broswer,然后是Web层的应用,业务逻辑层(有EJB实现),资源管理层。客户请求浏览页面,一般Web层的View有JSP组成,并且使用了大量Taglib。把每个请求映射到某个HTMLAction类来响应它。HTML Action类是一个标准的类,执行选择的HTML Action。使用MVC模式减少了代码的复制，即减少了代码的维护，由于模型返回的格式不带任何显示格式，因而模型可以直接应用于接口的使用，还因为MVC模型把不同的模型和不同的视图组合在一起完成不同的请求，因此，控制层可以说包含了用户请求权限的概念[13]。

在设计中还因应用了Struts框架,Struts跟Tomcat、Turbine等诸多Apache项目一样，是开源软件，这是它的一大优点。使开发者能更深入的了解其内部实现机制[11]。

除此之外，Struts的优点主要集中体现在Taglib和页面导航。Taglib是Struts的标记库，灵活动用，能大大提高开发效率。页面导航使系统的脉络更加清晰。通过一个配置文件，即可把握整个系统各部分之间的联系，这对于后期的维护有着莫大的好处[11]。

### 3.2.2数据库与Web服务器的连接

数据库连接时采用连接池技术链接MySQL,具体代码实现如下:

**public** ConnDB(){

**try** {

InputStream in=getClass().getResourceAsStream(*propFileName*);

*prop*.load(in); //通过输入流对象加载Properties文件

*dbClassName* = *prop*.getProperty("DB\_CLASS\_NAME"); //获取数据库驱动

*dbUrl* = *prop*.getProperty("DB\_URL",

"jdbc:mysql://127.0.0.1:3306/db\_librarySys?user=root&password=123&useUnicode=true");

}

**catch** (Exception e) {

e.printStackTrace(); //输出异常信息

}

}

**public** **static** Connection getConnection() {

Connection conn = **null**;

**try** {

Class.*forName*(*dbClassName*).newInstance();

conn = DriverManager.*getConnection*(*dbUrl*);

}

**catch** (Exception ee) {

ee.printStackTrace();

}

**if** (conn == **null**) {

System.*err*.println(

"警告: DbConnectionManager.getConnection() 获得数据库链接失败.\r\n\r\n链接类型:" +

*dbClassName* + "\r\n链接位置:" + *dbUrl*);

}

**return** conn;

}

/\*

\* 功能：执行查询语句

\*/

**public** ResultSet executeQuery(String sql) {

**try** {

conn = *getConnection*();

stmt = conn.createStatement(ResultSet.*TYPE\_SCROLL\_INSENSITIVE*,

ResultSet.*CONCUR\_READ\_ONLY*);

rs = stmt.executeQuery(sql);

} **catch** (SQLException ex) {

System.*err*.println(ex.getMessage());

}

**return** rs;

}

/\*

\* 功能:执行更新操作

\*/

**public** **int** executeUpdate(String sql) {

**int** result = 0;

**try** {

conn = *getConnection*(); //调用getConnection()方法构造Connection对象的一个实例conn

stmt = conn.createStatement(ResultSet.*TYPE\_SCROLL\_INSENSITIVE*,

ResultSet.*CONCUR\_READ\_ONLY*);

result = stmt.executeUpdate(sql); //执行更新操作

} **catch** (SQLException ex) {

result = 0;

}

**return** result;

}

/\*

\* 功能:关闭数据库的连接

\*/

**public** **void** close() {

**try** {

**if** (rs != **null**) {

rs.close();

}

**if** (stmt != **null**) {

stmt.close();

}

**if** (conn != **null**) {

conn.close();

}

} **catch** (Exception e) {

e.printStackTrace(System.*err*);

}

}

}

### 3.2.3登录模块程序设计

本模块主要是用户通过图书管理系统的首页登录进入该系统。用户输入正确的用户名和密码，系统会根据用户的身份进行相应权限划分；如果登录信息有错误，则系统提示登入错误的信息，并且禁止系统用户进行任何操作。图书借阅管理系统的登录主页面如图4.1所示。



图4.1 图书管理系统登录首页

用户在登录页面写好用户名和密码，选择登录，登录成功则跳转到系统的首页，否则提示错误信息[14]。在服务器端进行用户身份验证的程序流程图如图4.2程序流程图所示

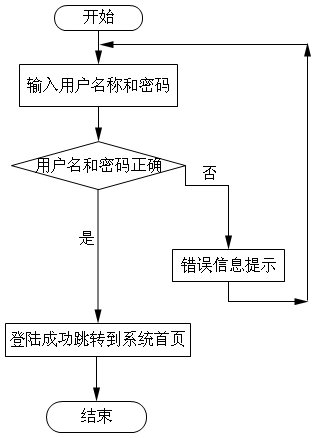


图4.2用户登录流程图

### 4.2.4系统管理员功能模块的实现

本模块中最主要的是管理系统用户,设置图书馆信息等操作,其中管理系统用户包括对他们进行添加,修改,删除及权限划分操作。

其中系统管理员读系统用户进行管理的操作页面如下图所示：



图4.3 系统管理员管理用户操作界面

其中对系统用户权限的设置是该操作的重要部分也是系统提高效率的关键所在，起具体实现代码为：

**private** ActionForward managerModify(ActionMapping mapping, ActionForm form,

HttpServletRequest request, HttpServletResponse response) {

ManagerForm managerForm = (ManagerForm) form;

managerForm.setId(managerForm.getId()); //获取并设置管理员ID号

managerForm.setName(managerForm.getName()); //获取并设置管理员名称

managerForm.setPwd(managerForm.getPwd()); //获取并设置管理员密码

managerForm.setSysset(managerForm.getSysset()); //获取并设置系统设置权限

managerForm.setReaderset(managerForm.getReaderset()); //获取并设置读者管理权限

managerForm.setBookset(managerForm.getBookset()); //获取并设置图书管理权限

managerForm.setBorrowback(managerForm.getBorrowback()); //获取并设置图书借还权限

managerForm.setSysquery(managerForm.getSysquery()); //获取并设置系统查询权限

**int** ret = managerDAO.update(managerForm); //调用设置管理员权限的方法

**if** (ret == 0) {

request.setAttribute("error", "设置管理员权限失败！");//保存错误提示信息到error参数中

**return** mapping.findForward("error"); //转到错误提示页面

} **else** {

**return** mapping.findForward("managerModify"); //转到权限设置成功页面

}

}

### 3.2.5读者管理功能模块的实现

读者管理主要实现对读者的类型管理和信息管理两个功能子模块，其中读者类型管理中需要设置不同类型读者一次性可借阅的图书数量，读者信息管理则主要管理维护读者的基本信息，其各自操作界面如下所示：

1) 读者类型管理功能界面：



图4.4 读者类型管理操作界面

2)读者信息管理功能界面：



图4.5 读者信息管理界面

该管理功能在设置读者信息时需要输入读者姓名，性别，条形码，选择读者类别，有效证件，证件号码，电话，e-mail等信息。其中的姓名，性别，条形码，证件号码是必填选项。

### 3.2.6查询功能模块的实现

查询功能模块包括图书查询，读者借阅查询和借阅到期提醒三个子功能模块。

1) 图书查询可根据图书的条形码，书名，作者，出版社等不同信息进行查询，其操作界面图下图所示：



图4.6 图书查询操作界面

实现该功能中根据读者不同需求的按条件查询实现语句如下：

**public** Collection query(String strif){

BookForm bookForm=**null**;

Collection bookColl=**new** ArrayList();

String sql="";

**if**(strif!="all" && strif!=**null** && strif!=""){

sql="select \* from (select b.\*,c.name as bookcaseName,p.pubname as publishing,t.typename from tb\_bookinfo b left join tb\_bookcase c on b.bookcase=c.id join tb\_publishing p on b.ISBN=p.ISBN join tb\_booktype t on b.typeid=t.id where b.del=0) as book where book."+strif+"'";

}**else**{

sql="select b.\*,c.name as bookcaseName,p.pubname as publishing,t.typename from tb\_bookinfo b left join tb\_bookcase c on b.bookcase=c.id join tb\_publishing p on b.ISBN=p.ISBN join tb\_booktype t on b.typeid=t.id where b.del=0";

}

System.*out*.println("图书查询时的SQL："+sql);

ResultSet rs=conn.executeQuery(sql);

**try** {

**while** (rs.next()) {

bookForm=**new** BookForm();

bookForm.setBarcode(rs.getString(1));

bookForm.setBookName(rs.getString(2));

bookForm.setTypeId(rs.getInt(3));

bookForm.setAuthor(rs.getString(4));

bookForm.setTranslator(rs.getString(5));

bookForm.setIsbn(rs.getString(6));

bookForm.setPrice(Float.*valueOf*(rs.getString(7))); //此处必须进行类型转换

bookForm.setPage(rs.getInt(8));

bookForm.setBookcaseid(rs.getInt(9));

bookForm.setInTime(rs.getString(10));

bookForm.setOperator(rs.getString(11));

bookForm.setDel(rs.getInt(12));

bookForm.setId(Integer.*valueOf*(rs.getString(13)));

bookForm.setBookcaseName(rs.getString(14));

bookForm.setPublishing(rs.getString(15));

bookForm.setTypeName(rs.getString(16));

bookColl.add(bookForm);

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

conn.close();

**return** bookColl;

}

2)借阅查询子模块是对图书当前状态和读者当前的借阅情况进行查询,具有此权限的用户登录并进入该操作界面后选中相应的查询条件,并输入相应的查询信息,系统即可在页面显示被查询的图书的状态,或相应读者的当起借阅情况,此外,还可以输入相应时间段内的所有借阅清单,或选择某个读者查询他在某个时间段内的借阅情况,查询某个时间段内的此子功能操作界面为:



图4.7借阅查询操作界面

系统中同时选中日期和限制条件进行查询时,程序是在条件查询的基础上选中符合时间条件限制的内容,其实现代码为:

**if** (flag.length == 2) {

**if** (request.getParameter("f") != **null**) {

str = request.getParameter("f") + " like '%" +

request.getParameter("key") + "%'";

}

System.*out*.println("日期和条件");

String sdate = request.getParameter("sdate");

String edate = request.getParameter("edate");

String str1 = **null**;

**if** (sdate != **null** && edate != **null**) {

str1 = "borrowTime between '" + sdate + "' and '" + edate +

"'";

}

str = str + " and borr." + str1;

System.*out*.println("条件和日期：" + str);

}

3) 借阅到期提醒子模块是将系统当前时间与应归还的时间进行比较,如果系统当前时间以超过图书应归还时间,相应的读者借阅信息便会被显示在借阅到期提醒界面,其界面为:



图4.8 借阅到期提醒

### 3.2.7图书管理功能模块的实现

与读者管理模块类似，图书管理模块也分为图书类型管理和图书档案管理两个子模块，其中图书类型管理功能处理将图书分类提高管理效率外还设置了不同类型图书可借阅的天数，对不同类型图书进行区别管理，图书档案管理是管理维护图书馆藏书的基本信息。这两个子模块的操作界面为：

1) 图书类型管理界面：



图4.9 图书类型管理界面

2) 图书档案管理界面：



图4.10 图书档案管理界面

图书档案中包含图书的条形码，图书名称，图书类型，作者，译者，价格，出版社，所在书架等图书基本信息，其中作者，译者，页码是备选信息，可以不填。

### 3.2.8图书借还功能模块的实现

该功能模块可分为图书借阅,图书归还,图书续借三个子模块.

1)图书借阅。该子模块实现系统的借阅功能,读者向图书管理员提供自己的编号会显示出读者当起的借阅情况,再提供要借阅的图书标号,图书管理员将信息记录在系统中,借阅成功,其操作界面如下图:



图4.11 图书借阅操作界面

如果在借阅时所要借阅的图书不存在或者读者借阅已达借阅图书的上线,则读者不能接续借阅图书,系统将给出相应提示,此功能的实现代码为:

**private** ActionForward bookborrow(ActionMapping mapping, ActionForm form,

HttpServletRequest request,

HttpServletResponse response){

//查询读者信息

//此处一定不能使用该语句进行转换

readerForm.setBarcode(request.getParameter("barcode"));

ReaderForm reader = (ReaderForm) readerDAO.queryM(readerForm);

request.setAttribute("readerinfo", reader);

//查询读者的借阅信息

request.setAttribute("borrowinfo",borrowDAO.borrowinfo(request.getParameter("barcode")));

//完成借阅

String f = request.getParameter("f");

String key = request.getParameter("inputkey");

**if** (key != **null** && !key.equals("")) {

String operator = request.getParameter("operator");

BookForm bookForm=bookDAO.queryB(f, key);

**if** (bookForm!=**null**){

**int** ret = borrowDAO.insertBorrow(reader, bookDAO.queryB(f, key),

operator);

**if** (ret == 1) {

request.setAttribute("bar", request.getParameter("barcode"));

**return** mapping.findForward("bookborrowok");

} **else** {

request.setAttribute("error", "添加借阅信息失败!");

**return** mapping.findForward("error");

}

}**else**{

request.setAttribute("error", "没有该图书!");

**return** mapping.findForward("error");

}

}

**return** mapping.findForward("bookborrow");

}

2)图书归还。该子模块实现系统的图书归还功能,读者向图书管理员提供自己的编号会显示出读者当起的借阅情况,再提供要归还的图书标号,图书管理员将信息记录在系统中,归还成功,其操作界面如下图：



图4.12 图书归还操作界面

# 4.软件测试

## 4.1 软件测试的方法与步骤

该系统在本地服务器上进行运行和调试，鉴于对系统的内部结构和处理算法的完全了解以及对系统功能的全面掌握对系统进行白盒测试和黑盒测试。

在开发软件系统的过程中，需要面对错综复杂的问题，因此，在软件生存周期的每个阶段都不可避免地会产生错误。测试目的在于：发现错误而执行一个程序的过程,测试重要发现一个发现其中尚未发现的错误。

本系统的测试阶段信息流程下图所示[15]。

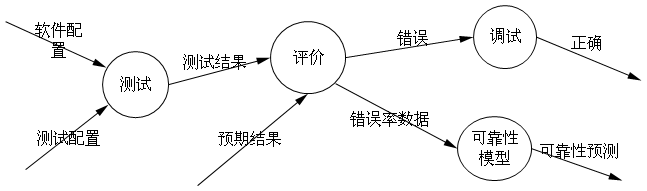


图5.1 测试阶段信息流程图

为了设计出有效地测试方案按照下面准则进行测试:所有测试都应追溯到用户需求；在完成了需求模型就要着手制定测试计划，在编码之前最所有测试工作进行计划和设计；运用Pareto原理着重对占出现错误80%的容易出错的20%的模块进行测试,从小规模开始逐步进行大规模测试,通常先重点测试单个程序模块再转向集成的模块簇;精心设计测试方案,尽可能充分覆盖程序逻辑使之达到要求的可靠性[15]。

按照软件工程的观点，软件测试（主要是指多模块程序的测试）共包括4个层次。

1)单元测试。单元测试的用例从单元详细设计中导出。在单元测试中可以采用功能性测试和结构性测试两种。

2)集成测试和确认测试。这一阶段的任务，是通过了单元测试的模块逐步组装起来，通过测试与纠错，最终得到一个满足需求的目标软件。

3)验证测试。在这个测试步骤中所发现的往往是需求规格说明的错误。一般来说，系统测试是功能性测试，不是结构性测试[15]。

在整个测试中，采用了白盒测试和黑盒测试相结合的方法。

## 4.2 测试用例设计与测试用例的运行过程及测试结果分析

### 5.2.1模块测试

1)登录模块测试

表5.1登录模块用例测试

|  |  |  |
| --- | --- | --- |
| 用例 | 测试数据 | 预期结果 |
| UC1:输入正确的用户名和密码 | 用户名:admin密码:admin | 系统跳转到登陆成功页面 |
| UC2:输入空用户名 | 用户名: | 请输入用户名 |
| UC3:输入用户名和空密码 | 用户名:admin 密码: | 请输入密码 |
| UC4:输入用户名和错误的密码 | 用户名:admin 密码:321ff | 您输入的用户名或密码错误 |

当用户输入的用户名或密码错误是系统会跳转到出错页面并提示“您输入的用户名或密码错误”,用户点击确定,系统会重新返回登陆页面,其中错误信息提示页面如图5.1所示:

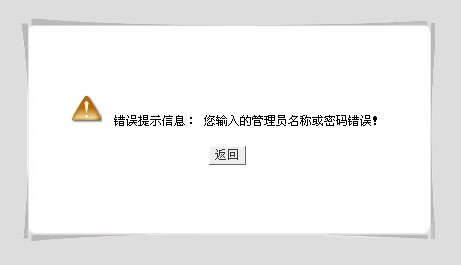


图5.2 系统出错提示页面

2)更改密码模块测试

用户登录系统后可对自己的登录密码进行更改,更改过程要求用户提供原始密码,输入新密码,重新输入新密码进行确认,如果用户输入的原始密码错误,或两次输入的新密码不一致将无法执行密码更新操作,系统会给出相应的提示,起显示页面为:



图5.3 两次密码不一致提示页面

### 4.2.2集成测试

把经过单元测试的模块放在一起形成一个子系统并对它进行测试用于检验模块相互间的协调和通信,这里主要对图书管理员功能系统进行了测试,经过检验其功能均得以顺利实现,下图为管理员操作的读者借阅模块成功运行界面：



图5.5 图书管理员子系统功能测试截图

把经过测试的子系统装配成一个完整的系统进行测试，经过黑盒测试于白盒测试相结合的方式，对整个系统的各个功能模块进行了测试，并调试改正其中的设计和编码错误，经过这个环节的操作整个系统的功能基本实现成功运行。

### 4.2.3 验收测试

在进行了以上的测试工作后，将整个软件系统作为单一的实体进行测试，测试内容与系统测试基本类似，但它是在用户积极参与下进行的，此测试过程主要使用实际数据,进行测试,验证测试的目的是验证系统确实能够满足用户的需求,经过这个环节的实际数据测试，系统的各个功能实现都达到了系统需求设计的要求。

注明：此项目文档与项目源代码仅作为作业使用。