

学生书店 图书管理系统

Java EE 期末项目设计

2018 年 6 月 26 日

目录

目录.....	2
1. 引言.....	4
1.1 编写目的.....	4
1.2 项目背景.....	4
1.3 相关定义.....	4
1.4 团队成员.....	5
2. 需求概述.....	6
2.1 目标.....	6
2.2 用户类和特征.....	6
2.3 运行环境.....	6
3. 需求分析.....	7
3.1 数据需求.....	7
3.2 事物需求.....	8
3.3 性能需求.....	8
3.4 安全性需求.....	8
3.5 确定用例及编写用例文档.....	9
4. 开发平台和技术简介.....	10
4.1 开发及运行环境.....	10
4.2 开发平台.....	10
4.3 Tomcat 服务器.....	10
4.4 MySQL	11
4.5 Navicat for MySQL	11
4.6 主要技术简介.....	11
4.7 设计模式.....	13
5. 概要设计.....	15

5.1 系统的业务流程图.....	15
5.2 数据库设计.....	16
6. 详细设计.....	19
6.1 数据流程图.....	19
6.2 用户功能模块.....	20
6.3 管理员功能模块.....	23
6.4 数据库连接.....	25
7. 故障处理和总结.....	27
7.1 故障处理.....	27
7.2 总结.....	27
附录 测试用例	29

1. 引言

1.1 编写目的

软件需求分析,其目的是用于说明软件产品或软件项目需要满足的条件和限制。在软件工程项目中,首先要获取用户的需求,通过对软件需要的提取、分析、文档化和验证,为进一步的设计和实现提供数据。软件需求与其他活动也密切相关,如软件测试、配置管理和质量管理等。

本需求的编写是为了研究利用数据库做基础的图书管理系统软件的开发途径和应用方法。同时它也是进行项目策划、概要设计和详细设计的基础,是维护人员进行内部维护,信息更新,验收和测试的依据。

1.2 项目背景

传统人工的管理方式存在着许多缺点,如: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倍感头痛,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,耗费劳动量大,难于避免错误的产生,如借书证挂失后,仍有少数学生冒用已挂失的借书证借出图书,造成图书外流。

本产品是针对电脑管理图书的需求设计的,可以完成读者登记、购入新书、读者借还书、图书注销等。能够有效的弥补传统管理方式的缺陷。

1.3 相关定义

数据流图(Data Flow Diagram):简称 DFD,它从数据传递和加工角度,以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程,是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

数据字典:它是数据的信息的集合,即对数据流图中包含的所有元素的定义的集合。其作用是为设计人员提供关于数据描述的信息。数据字典应该由对以

下四类元素的定义组成：数据流、数据流分量（即数据元素）、数据存储、处理。

1.4 团队成员

项目成员	负责模块
彭佳丽	后台开发，数据库设计
邹宇	文档编写

2. 需求概述

2.1 目标

“图书管理系统”主要提供图书信息和读者基本信息的维护以及借阅等功能。本系统是提高图书管理工作的效率，减少相关人员的工作量，使学校的图书管理工作真正做到科学、合理的规划，系统、高效的实施。

为了实现文档集中存储，分布管理，便于资料的便捷查询和安全控制，提高设备使用效率，减少维护成本。该系统建立了统一的在线实时更新的文档库，可对多种形式的信息内容进行存贮，从声音、图形、图像等多媒体信息到高度结构化的文档，从 e-mail 到讨论数据库；提供层次化的文档存储模式，实现对文档的生命管理，对文档可进行签入/签出检查、版本控制、跨所有管理内容的全文检索、审核追踪、文档描述等。

2.2 用户类和特征

最高用户是图书管理员和读者，图书管理员需要进行用户的创建、修改和删除等工作，要求具备计算机知识，如权限管理等。读者是普通用户，具备一定的计算机操作即可。

2.3 运行环境

由于开发语言——Java 支持跨平台，因此本项目——图书管理系统支持 Windows、iOS、Linux 等主流操作系统。

3. 需求分析

3.1 数据需求

图书馆管理信息系统需要完成功能主要有：

- 1) 读者基本信息的输入，包括借书证编号、读者姓名、读者性别等。
- 2) 读者基本信息的查询、修改，包括读者借书证编号、读者姓名、读者性别等。
- 3) 书籍类别标准的制定、类别信息的输入，包括类别编号、类别名称。
- 4) 书籍类别信息的查询、修改，包括类别编号、类别名称。
- 5) 书籍库存信息的输入，包括书籍编号、书籍名称、书籍类别、作者姓名、出版社名称、出版日期、登记日期。
- 6) 书籍库存信息的查询，修改，包括书籍编号、书籍名称、书籍类别、作者姓名、出版社名称、出版日期登记日期等。
- 7) 借书信息的输入，包括读者借书证编号、书籍编号、借书日期。
- 8) 借书信息的查询、修改，包括借书证编号、借书证编号、读者姓名、书籍编号、书籍名称、借书日期等。
- 9) 还书信息的输入，包括借书证编号、书籍编号、还书日期。
- 10) 还书信息的查询和修改，包括还书读者借书证编号、读者姓名、书籍编号、书籍名称、借书日期、还书日期等。
- 11) 超期还书罚款输入，还书超出期限包括超出期限还书的读者借书证号，书籍编号，罚款金额。
- 12) 超期还书罚款查询，删除，包括读者借书证编号、读者姓名、书籍编号、书籍名称，罚款金额等
- 13) 管理员管理：包括创建读者用户信息、删除读者用户信息、添加图书信息、删除图书信息。
- 14) 超级管理员管理：包括创建管理员用户信息、删除管理员用户信息、创建读者用户信息、删除读者用户信息、添加图书信息、删除图书信息。

3.2 事物需求

- 1) 在读者信息管理部分, 要求:
 - a. 可以查询读者信息。
 - b. 可以对读者信息进行添加及删除的操作。
- 2) 在书籍信息管理部分, 要求:
 - a. 可以浏览书籍信息, 要求:
 - b. 可以对书籍信息进行维护, 包括添加及删除的操作。
- 3) 在借阅信息管理部分, 要求:。
 - a. 可以浏览借阅信息。
 - b. 可以对借阅信息进行维护操作。
- 4) 在归还信息管理部分, 要求:
 - a. 可以浏览归还信息
 - b. 对归还信息可修改维护操作
- 5) 在管理者信息管理部分, 要求:
 - a. 显示当前数据库中管理者情况。
 - b. 对管理者信息维护操作。
- 6) 在罚款信息管理部分要求:
 - a. 可以浏览罚款信息
 - b. 对罚款信息可以更新

3.3 性能需求

学校对系统性能无特殊的要求, 只要查询图书的时候没有明显的延迟就可以了, 查询的时间不要超过 3 秒钟。

3.4 安全性需求

对于图书管理系统的图书量会非常大, 所以在对这些图书导入和查询时要保证速度。在图书借阅过程中又要保证事务的完整性。对于整个系统, 需要完整的权限控制, 防止某些人恶意攻击系统, 修改原始记录, 同时对于数据库中的数据

需要定时备份，防止系统数据丢失。此外，系统要求用户在登陆时需要身份验证。

3.5 确定用例及编写用例文档

用例描述了一个完整的系统事件流程，其重点在于执行者与系统之间的交互而不是内在的系统活动，并对执行者产生有价值的可观测结果。

在这个“图书管理系统”中的用例图如下所示。

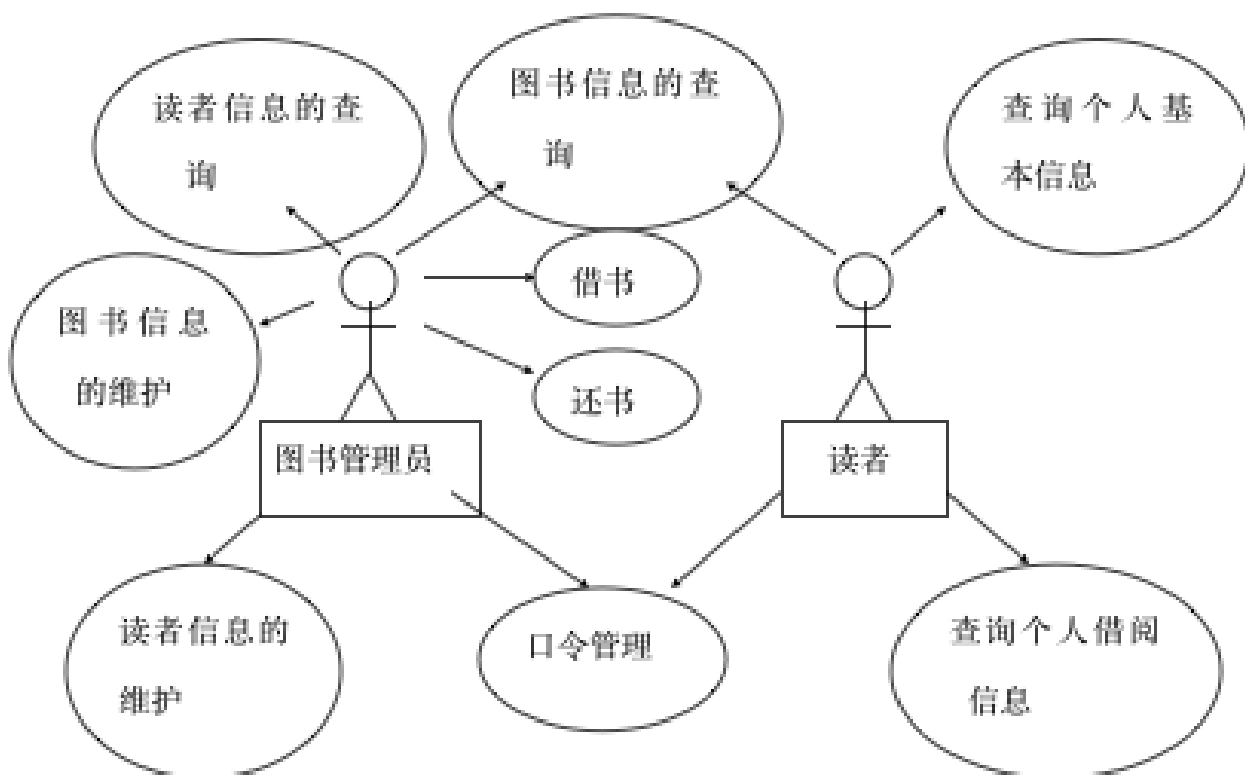


图 3-1 “图书管理系统”用例图

详细用例图见附录。

4. 开发平台和技术简介

4.1 开发及运行环境

硬件平台：

CPU：I5-8250U

内存：4G

软件平台：

操作系统：Windows10

数据库：MySQL5.5、Navicat for MySQL

开发工具：Eclipse

Web 服务器：Tomcat8 JDK1.8

4.2 开发平台

Eclipse IDE: IDE 是 integrated development environment 集成开发环境, SDK 是 service development pack 服务开发包。eclipse 就是 Java 业界最著名的集成开发环境。并且 Eclipse 开源, 操作简单, 而且也有很多插件, 十分的方便。

4.3 Tomcat 服务器

Tomcat 系统其实就是应用服务器系统中的一类, 属于轻量级的应用服务器。主要是在软件的开发过程中使用的。当然, 使用 Tomcat 系统的服务器, 我们就叫它 Tomcat 服务器。这是一个 Web 应用服务器, 并且是免费的进行开放源代码。在 Tomcat 服务器运行的过程中, 它占用的系统资源小, 并且扩展性能好。与此同时, Tomcat 服务器还支持负载平衡与邮件服务等开发应用系统常用的功能。Tomcat 服务器使用 Tomcat 系统, 所具有的优点也很鲜明, 首先它具有免费的开源代码, 其次是 Tomcat 系统技术先进并且性能也是相当的稳定, 最后 Tomcat 服务器具有良好的扩展性和安全性, 很适合一些中小企业使用。

4.4 MySQL

MySQL 是一个关系型数据库管理系统，开发者为瑞典 MySQL AB 公司。目前 MySQL 被广泛地应用在互联网行业。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，许多互联网公司选择了 MySQL 作为后端数据库。特别它还具有以下的优点：多语言支持：Mysql 为 C、C++、Python、Java、Perl、PHP、Ruby 等多种编程语言提供了 API，访问和使用方便；可以移植性好：MySQL 是跨平台的；高效：MySQL 的核心程序采用完全的多线程编程；支持大量数据查询和存储：Mysql 可以承受大量的并发访问。

4.5 Navicat for MySQL

Navicat for MySQL 是一套专为 MySQL 设计的强大数据库管理及开发工具。它可以用于任何 3.21 或以上的 MySQL 数据库服务器，并支持大部份 MySQL 最新版本的功能，包括触发器、存储过程、函数、事件、检索、权限管理等等。它为专业开发者提供了一套强大的足够尖端的工具，但对于新用户仍然易于学习。它提供类似于 MySQL 的用管理界面工具。此解决方案的出现，解放了 PHP、J2EE 等程序员以及数据库设计者、管理者的大脑，降低开发成本，为用户带来更高的开发效率。此外 Navicat for MySQL 使用了极好的图形用户界面(GUI)，可以用一种安全和更为容易的方式快速和容易地创建、组织、存取和共享信息。

4.6 主要技术简介

1) HTML

超文本标记语言，标准通用标记语言下的一个应用。“超文本”就是指页面内可以包含图片、链接，甚至音乐、程序等非文字元素。超文本标记语言的结构包括“头”部分(Head)、和“主体”部分(Body)，其中“头”部提供关于网页的信息，“主体”部分提供网页的具体内容。HTML 是一种规范，一种标准，它通过标记符号来标记要显示的网页中的各个部分。网页文件本身是一种文

本文件，通过在文本文件中添加标记符，可以告诉浏览器如何显示其中的内容（如：文字如何处理，画面如何安排，图片如何显示等）。

2) CSS

层叠样式表(英文全称:Cascading Style Sheets)是一种用来表现 [HTML](#)(标准通用标记语言的一个应用)或 [XML](#)(标准通用标记语言的一个子集)等文件样式的计算机语言。CSS 不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。具有丰富的样式定义、易于使用和修改、多页面应用、层叠和页面压缩的特点。

3) JavaScript

JavaScript 是一种属于网络的脚本语言,已经被广泛用于 Web 应用开发,常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。通常来说 JavaScript 脚本是通过嵌入在 HTML 中来实现自身的功能的。通俗来讲，就是增加网页的互动性。

4) Servlet

就是 Java 的一个接口，是 Java 给的一种规范。servlet 接口定义的是一套处理网络请求的规范，所有实现 servlet 的类，都需要实现它那五个方法，其中最主要的是两个生命周期方法 `init()`和 `destroy()`，还有一个处理请求的 `service()`。

5) JSP

JSP 全称 Java Server Pages，是一种动态网页开发技术。它使用 JSP 标签在 HTML 网页中插入 Java 代码。标签通常以`<%`开头以`%>`结束。JSP 是一种 Java servlet，主要用于实现 Java web 应用程序的用户界面部分。网页开发者们通过结合 HTML 代码、XHTML 代码、XML 元素以及嵌入 JSP 操作和命令来编写 JSP。JSP 通过网页表单获取用户输入数据、访问数据库及其他数据源，然后动态地创建网页。JSP 标签有多种功能，比如访问数据库、记录用户选择信息、访问 JavaBeans 组件等，还可以在不同的网页中传递控制信息和共享信息。

6) JavaBean

JavaBean 在 J2EE 的开发中，通常用于封装数据。在 JSP 技术中，它提供了

三个关于 JavaBean 组件的动作元素，即 JSP 标签，它们分别为：

<jsp:useBean>标签：用于在 JSP 页面中查找或实例化一个 JavaBean 的组件；

<jsp:setproperty>标签：用于在 JSP 页面中设置一个 JavaBean 组件的属性；

<jsp:getproperty>标签：用于在 JSP 页面中获取一个 JavaBean 组件的属性；

4.7 设计模式

本次的图书管理项目是属于轻量级的开发。所采用的是 MVC 框架，即 Model View Controller，是模型—视图—控制器的缩写，它同时提供了对 HTML、CSS 和 JavaScript 的完全控制。

模型：模型持有所有的数据、状态和程序逻辑。模型独立于视图和控制器。

视图：用来呈现模型。视图通常直接从模型中取得它需要显示的状态与数据。对于相同的信息可以有多个不同的显示形式或视图。

控制器：位于视图和模型中间，负责接受用户的输入，将输入进行解析并反馈给模型，通常一个视图具有一个控制器。MVC 模式将它们分离以提高系统的灵活性和复用性，不使用 MVC 模式，用户界面设计往往将这些对象混在一起。

它还具有以下的优点：

一个模型提供不同的多个视图表现形式，也能够为一个模型创建新的视图而无须重写模型。一旦模型的数据发生变化，模型将通知有关的视图，每个视图相应地刷新自己。

模型可复用。因为模型是独立于视图的，所以可以把一个模型独立地移植到新的平台工作。

提高开发效率。在开发界面显示部分时，你仅仅需要考虑的是如何布局一个好的用户界面；开发模型时，你仅仅要考虑的是业务逻辑和数据维护，这样能使开发者专注于某一方面的开发，提高开发效率。

总的来说，MVC 设计模式可以方便开发人员分工协作，提高开发效率，增强程序的可维护性和拓展性而且还利用 Controller 将 Model 与 View 分离，降低它们之间的耦合度。

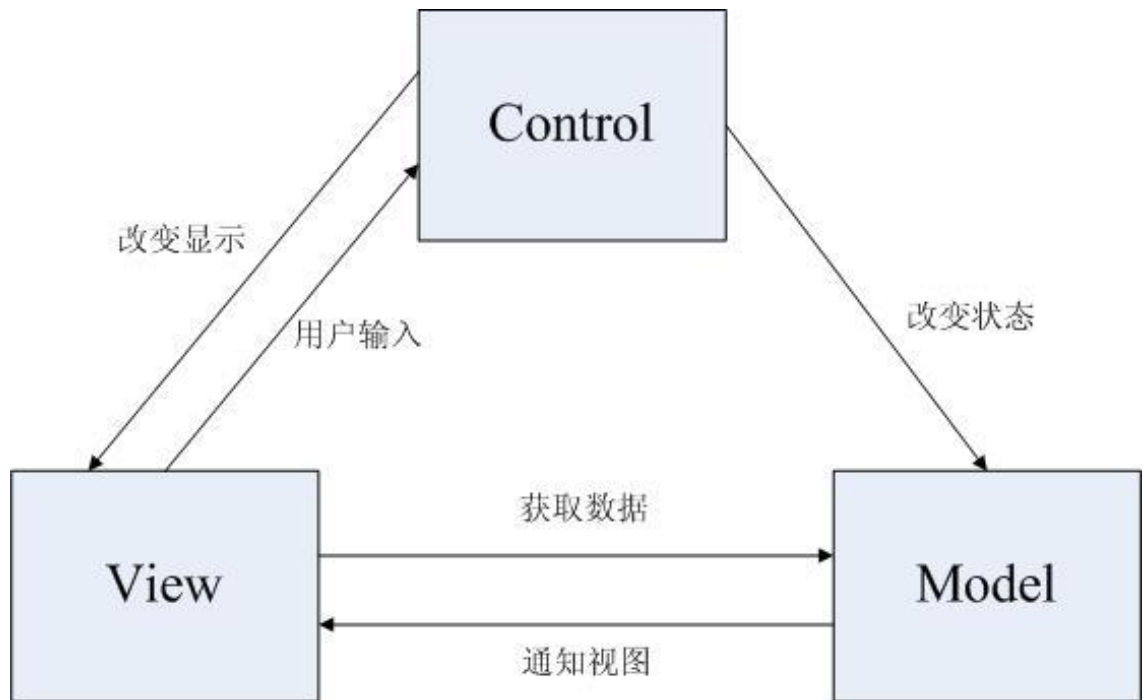


图 4-1 MVC 架构图

5. 概要设计

5.1 系统的业务流程图

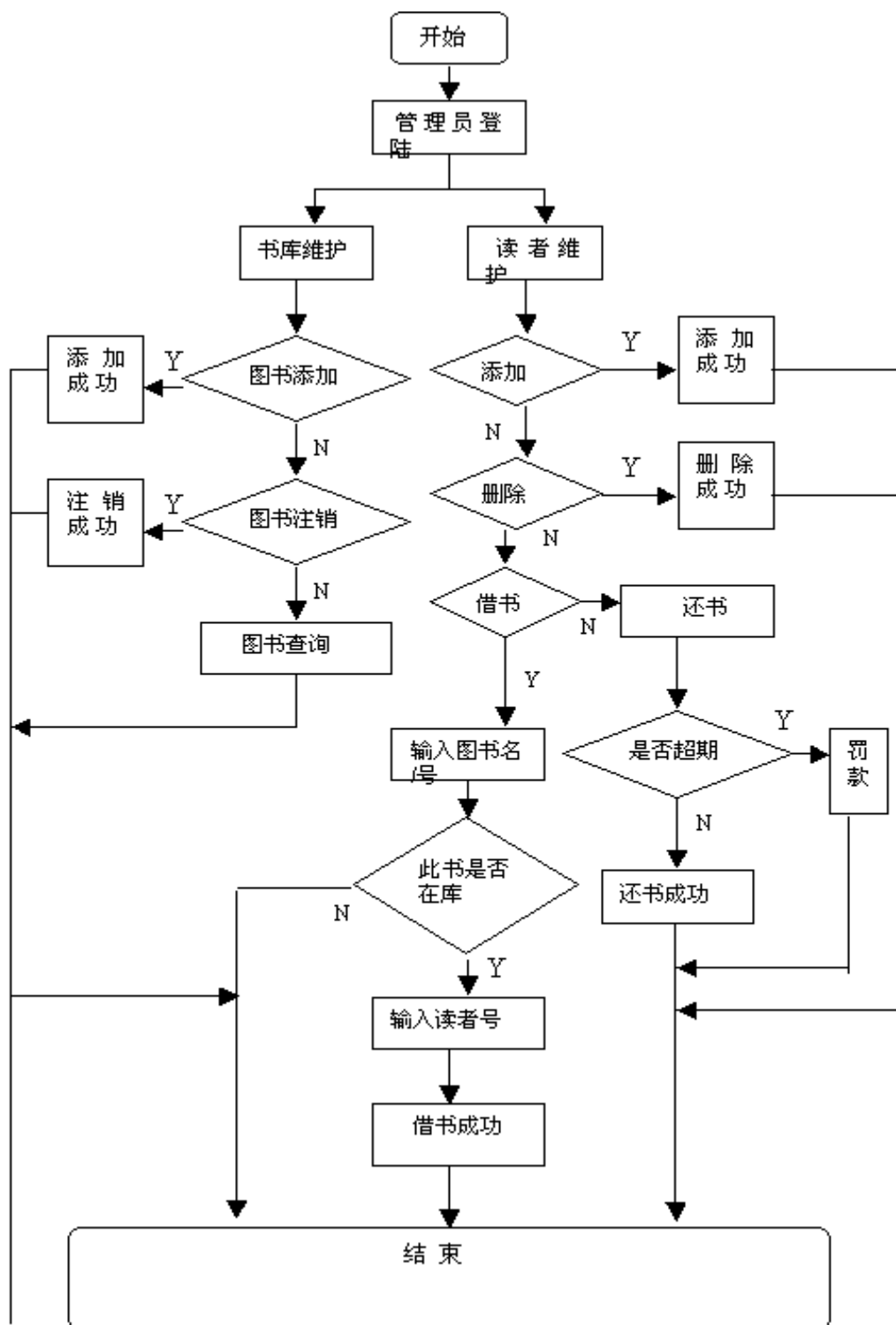


图 5-1 业务流程图

管理员登录系统后有两种主要的功能：书库维护和读者维护。书库维护功能主要是对图书的操作，添加图书、注销图书和图书的查询，当完成相应的操作就可以结束操作。读者维护功能主要是对读者的操作，添加读者、删除读者和读者借书的操作。读者无法操作这个系统，只能简单的查询书籍信息和结束情况。真正的借书操作是由管理员操作的，当读者借书成功后，还书的时候管理员需要判断是否超过了借书期限。如果超过了借书期限，则需要读者缴纳对应的罚款才能完成还书功能。具体见上图 5-1 业务流程图。

5.2 数据库设计

5.2.1 E-R 图

这个图书管理系统包括读者、书籍、管理员三个实体，具体 E-R 图如下图 5-2。

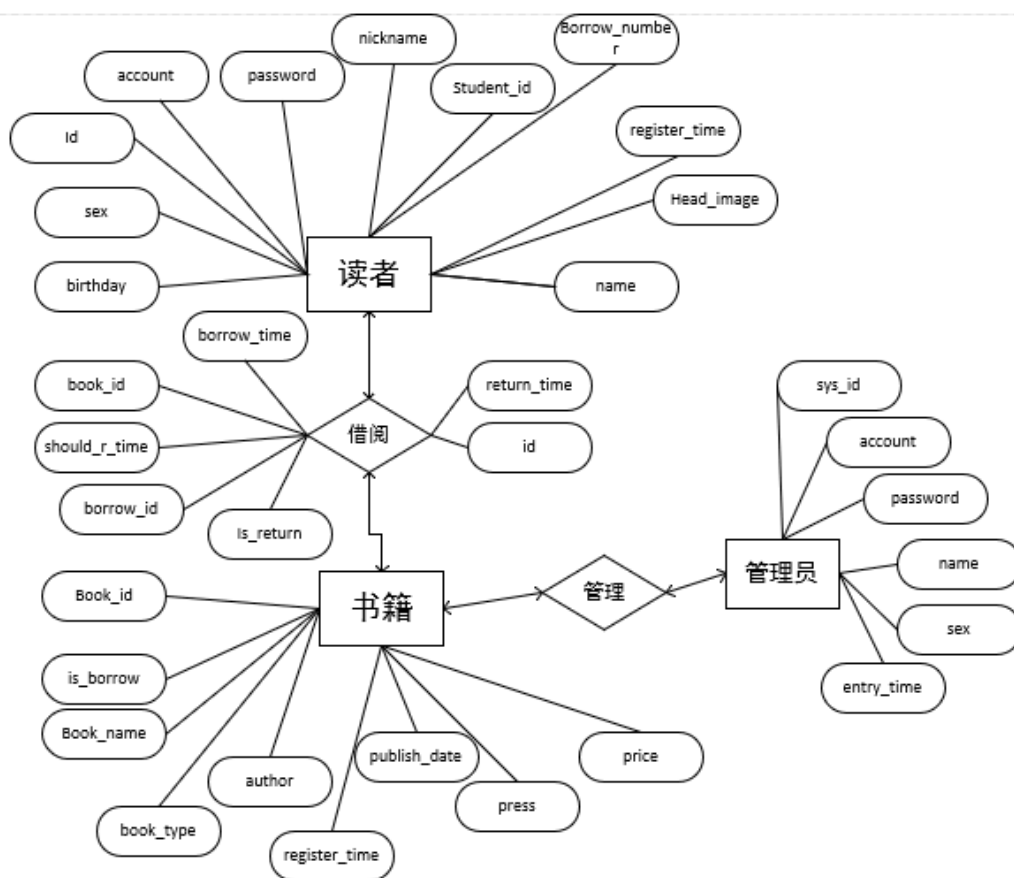


图 5-2 E-R 图

5.2.2 数据结构

1) j2ee_amerce 表，这个是惩罚表，包括惩罚 ID、借书的 ID、被惩罚的读者名字、借书的书名、书籍的 ID、借书的时间、应该还书的时间、实际还书的时间和罚金的数量。

名	类型	长度	小数点	不是 null	
amerce_id	int	11	0	<input checked="" type="checkbox"/>	 1
borrow_id	int	11	0	<input checked="" type="checkbox"/>	
name	varchar	15	0	<input checked="" type="checkbox"/>	
book_name	varchar	32	0	<input checked="" type="checkbox"/>	
book_id	char	7	0	<input checked="" type="checkbox"/>	
borrow_time	datetime	0	0	<input checked="" type="checkbox"/>	
should_r_time	datetime	0	0	<input checked="" type="checkbox"/>	
return_time	datetime	0	0	<input checked="" type="checkbox"/>	
finer	double	0	0	<input checked="" type="checkbox"/>	

图 5-3 惩罚表图

2) j2ee_book 表，这是书籍的信息包，包括书籍 ID、书籍名字、书记类型、作者、出版社、出版日期、书籍的价格、注册的时间和书籍目前的状态是否被借走。

名	类型	长度	小数点	不是 null	
book_id	char	7	0	<input checked="" type="checkbox"/>	 1
book_name	varchar	32	0	<input checked="" type="checkbox"/>	
book_type	tinyint	1	0	<input checked="" type="checkbox"/>	
author	varchar	32	0	<input checked="" type="checkbox"/>	
press	varchar	50	0	<input checked="" type="checkbox"/>	
publish_date	date	0	0	<input checked="" type="checkbox"/>	
price	double	0	0	<input checked="" type="checkbox"/>	
register_time	timestamp	0	0	<input checked="" type="checkbox"/>	
is_borrow	tinyint	1	0	<input checked="" type="checkbox"/>	

图 5-4 书籍信息表图

3) j2ee_borrow 表，这是借书的信息表，包括借书 ID、书籍的 ID、借书的时间、应该归还书籍的时间、是否归还图书、归还图书的时间和借书人的 ID。




名	类型	长度	小数点	不是 null	
borrow_id	int	11	0	<input checked="" type="checkbox"/>	 2
book_id	char	7	0	<input checked="" type="checkbox"/>	 3
borrow_time	timestamp	0	0	<input checked="" type="checkbox"/>	
should_r_time	timestamp	0	0	<input checked="" type="checkbox"/>	
is_return	tinyint	1	0	<input checked="" type="checkbox"/>	
return_time	timestamp	0	0	<input type="checkbox"/>	
id	int	11	0	<input checked="" type="checkbox"/>	 1

图 5-5 借书信息表图

4) j2ee_master 表，这是超级管理员信息表，包括管理员 ID、管理员登录账户、管理员登录密码、管理员的名字、管理员性别和管理员账号的登记时间。


名	类型	长度	小数点	不是 null	
sys_id	char	6	0	<input checked="" type="checkbox"/>	 1
account	varchar	15	0	<input checked="" type="checkbox"/>	
password	varchar	18	0	<input checked="" type="checkbox"/>	
name	varchar	15	0	<input checked="" type="checkbox"/>	
sex	char	1	0	<input checked="" type="checkbox"/>	
entry_time	timestamp	0	0	<input checked="" type="checkbox"/>	

图 5-6 超级管理员信息表图

5) j2ee_reader 表，这是读者信息表，包括读者 ID、读者登录账号、读者登录密码、读者的头像、昵称、读者的性别、读者的生日、学生号、借书的数量、读者账号注册时间和读者的名字。


名	类型	长度	小数点	不是 null	
id	int	11	0	<input checked="" type="checkbox"/>	 1
account	varchar	15	0	<input checked="" type="checkbox"/>	
password	varchar	18	0	<input checked="" type="checkbox"/>	
head_image	varchar	150	0	<input type="checkbox"/>	
nickname	varchar	15	0	<input type="checkbox"/>	
sex	tinyint	1	0	<input type="checkbox"/>	
birthday	date	0	0	<input type="checkbox"/>	
student_id	varchar	15	0	<input type="checkbox"/>	
borrow_number	tinyint	1	0	<input checked="" type="checkbox"/>	
register_time	timestamp	0	0	<input type="checkbox"/>	
name	varchar	15	0	<input checked="" type="checkbox"/>	

图 5-7 读者信息表图

6. 详细设计

6.1 数据流程图

通过对系统需求的调查分析，细化软件功能，把系统划分成几个子系统，明确每个子系统所要完成的主要逻辑功能，采用以图形的方式描绘数据在系统中流动和处理。数据流图是用来表示系统的逻辑模型。

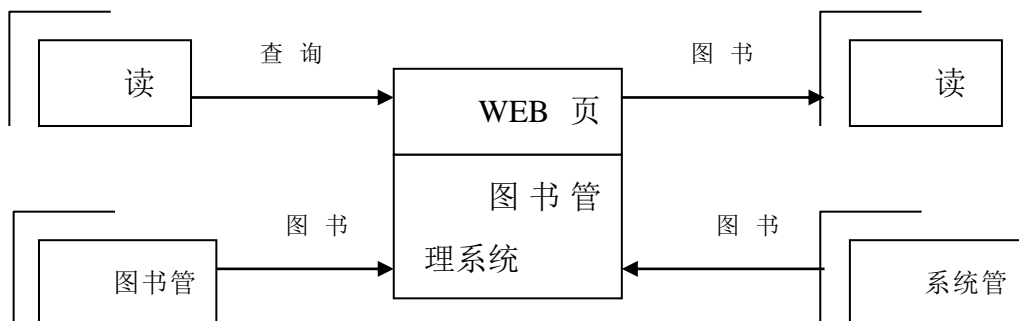


图 6-1 顶层数据流图

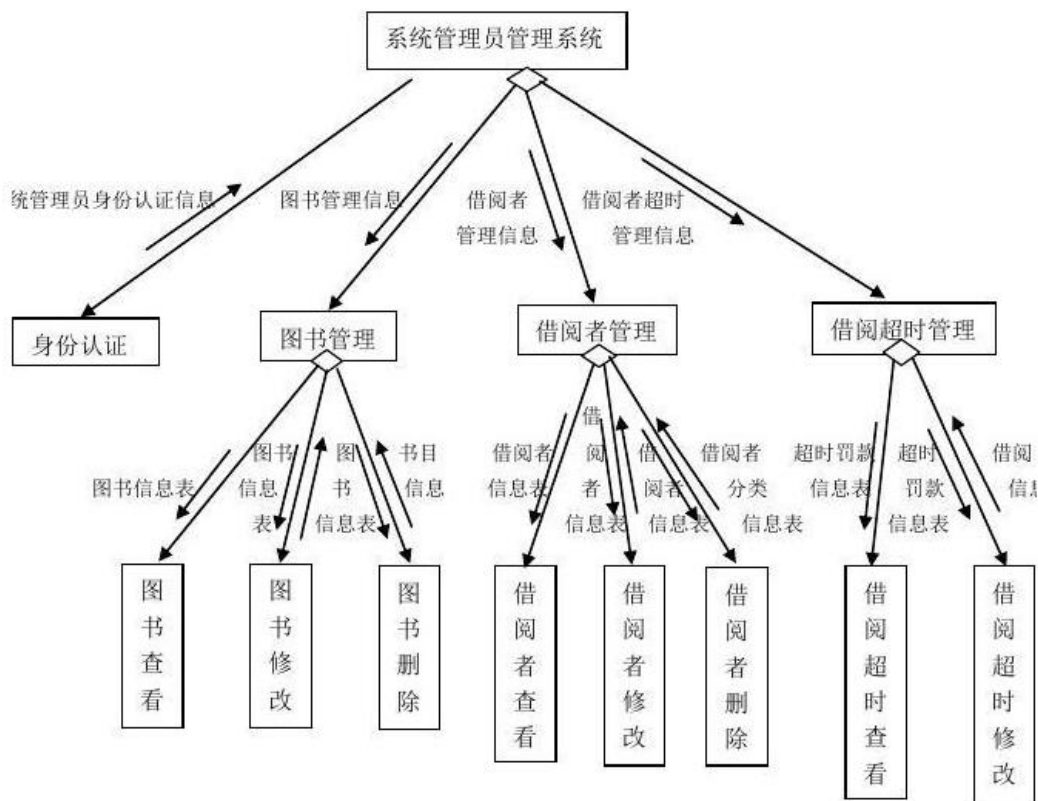


图 6-2 系统管理员流程图

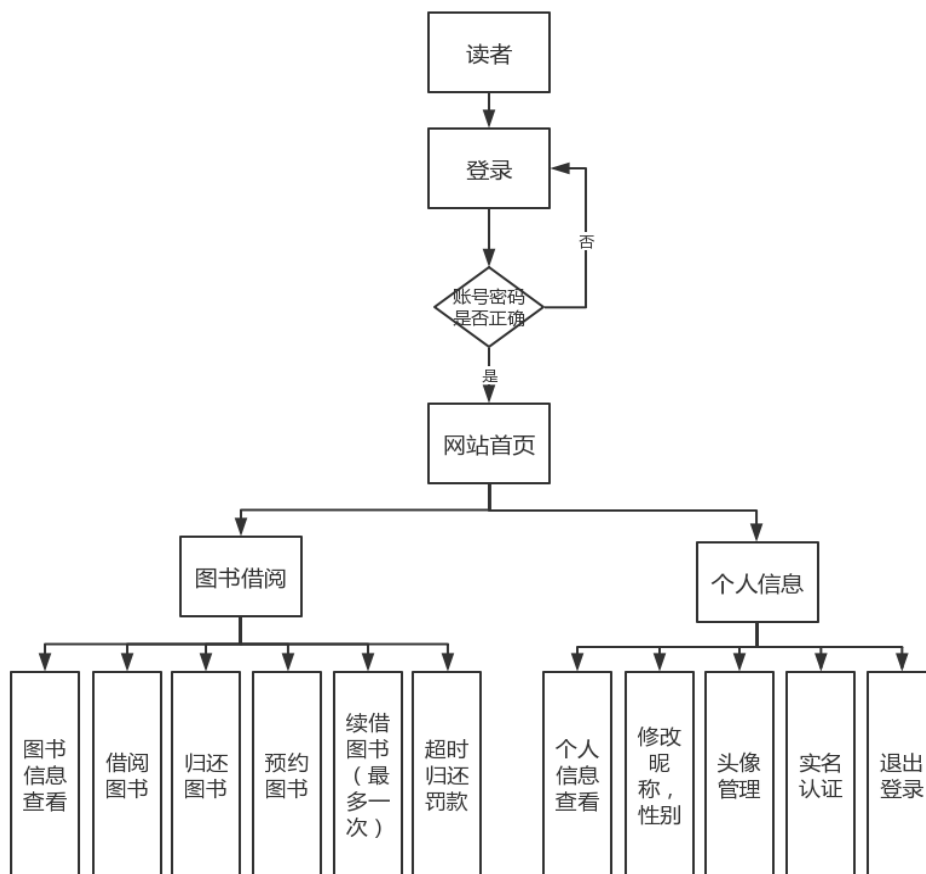


图 6-3 读者流程图

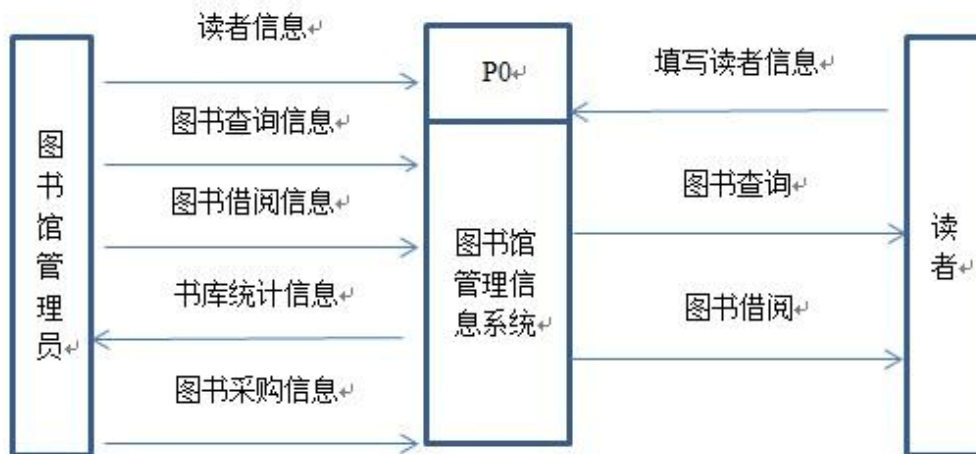


图 6-4 数据流图

6.2 用户功能模块

1) 用户登录界面



The login interface is titled "图书管理系统" (Library Management System). It contains two input fields for "登录名:" (Username) and "密码:" (Password). Below these fields are two radio buttons for user roles: "读者" (Reader) and "管理员" (Administrator). A blue "登录" (Login) button is centered below the radio buttons. At the bottom, there are two links: "注册" (Register) and "跳过登录" (Skip Login).

图 6-5 用户登录界面

2) 用户注册界面



The registration interface is titled "用户注册" (User Registration). It contains several input fields: "用户名:" (Username), "学号:" (Student ID), "真实名字:" (Real Name), "密码:" (Password), and "确认密码:" (Confirm Password). Below these fields is a "性别:" (Gender) section with three radio buttons: "男" (Male), "女" (Female), and "保密" (Privacy). A "验证码:" (Verification Code) field is followed by a blue box displaying "F4K3". A blue "注册" (Register) button is centered below the verification code field. At the bottom left, there is a link "有账号, 登录?" (Have an account, login?).

图 6-6 用户注册界面

3) 用户主界面



图 6-7 用户主界面

4) 用户登录功能

主要就是 form 标签里面的 action 属性，表示将表单中的内容提交给后台的 login 这个 sevelet 进行处理，其中 name 属性使用 login.account 标记输入的值，在 login 中使用 getAccount 获取标签中输入的值，其中密码，登录身份和账号的方法相同，就不一一赘述了。如下图所示：

```
<s:form action="login" method="post" theme="simple" id="login_form">
  <span id="nav_name">图书管理系统</span>
  <ul>
    <li>
      <span id="text_name1">登录名:</span>
      <s:textfield name="login.account" size="20" id="text_in1"/>
    </li>
    <li>
      <span id="text_name2">密码:</span>
      <s:password name="login.password" size="21" id="text_in2" />
    </li>
    <li>
      <s:div id="choice_box">
        <s:radio name="login.identity" list="#{'reader':'读者','master':'管理员'}" value="'reader'" id="choice"></s:radio>
      </s:div>
    </li>
    <s:submit value="登录" id="login_bu"/>
    <s:div id="link_a">
      <s:a href="register.jsp" id="link1">注册</s:a>
      <s:a href="index.jsp" id="link2">跳过登录</s:a>
    </s:div>
  </ul>
</s:form>
```

图 6-8 登录功能实现代码

5) 用户查看个人信息

```
public String showPersonInfo() {
    borrow_id = new IsLogin().isLogin();
    if (borrow_id > 0) {
        ReaderDao db1 = new ReaderDao();
        personInfo = db1.getPersonInfo(borrow_id);
    }
}
```

```

        session.put("personInfo", personInfo);
        return SUCCESS;
    } else {
        // 用户没有登录, 跳转到登录页面
        return "no_login";
    }
}
}

```

6.3 管理员功能模块

1) 管理员登录

通过 CSS 设计的管理员登录界面提示管理员输入自己的信息, 然后通过 String 接收用户输入的账户和密码。Form 表单里的 action 的自动注入, 将表单里的内容提交给后台进行处理, 若相对应就成功登录, 若不对应, 则登录失败。如下图所示:

```

public boolean adminLogin(String account, String password)
{
    try {
        String sql="select * from j2ee_master where account='"+account+"' and password='"+password+"'";
        rs=stm.executeQuery(sql);
        if(rs.next())
        {
            return true;
        }
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("管理员登录失败");
    }
    return false;
}

```

图 6-9 管理员登录

处理登录的 Servlet: 处理从 JSP 发送来的数据, 进行后台的信息比较, 通过 Dao 层提供的接口和该接口的实现类, 然后再直接调用这个公共类 Dao 类, 进行修改、添加、和删除数据库的操作。此外, 使 Try Catch 捕捉 try 部分的异常, 把错误信息存储在 Exception 里, 保证程序正常运行。

```

public String login() throws Exception {
    String identity = login.getIdentity();
    if (identity.equals("master")) {
        // 管理员登录
        MasterDao masterDB = new MasterDao();
        if (masterDB.adminLogin(login.getAccount(), login.getPassword())) {
            System.out.println("管理员登录成功");
            // 跳转到管理员首页
            return "master_index";
        } else {
            System.out.println("管理员登录失败");
            // 登录失败，重新登录
            return ERROR;
        }
    }
}

```

图 6-10 处理登录

2) 插入图书

添加图书，将图书的信息输入，获取 jbook 中的 get 方法获取到的图书信息的数据，然后调用 set 方法，设置图书的信息，如果在数据库中插入图书信息成功，就返回“添加书籍成功”，否则返回“添加书籍失败”。

```

public String addBook() {
    Add_book info = new Add_book();
    info.setAuthor(jbook.getAuthor());
    info.setBook_id(jbook.getBook_id());
    info.setBook_name(jbook.getBook_name());
    info.setBook_type(jbook.getBook_type());
    info.setPress(jbook.getPress());
    info.setPrice(jbook.getPrice());
    info.setStr_date(jbook.getStr_date());

    BooksDao db = new BooksDao();
    if (db.addBooks(info)) {
        System.out.println("添加书籍成功");
        return SUCCESS;
    } else {
        System.out.println("添加书籍失败");
        return ERROR;
    }
}

```

3) 删除图书

先使用 if 判断，如果 action.equal 是 delete 删除，就删除图书，和添加书籍相同，成功则返回“删除图书成功”，否则返回“删除图书失败”。

```

if (action.equals("delete")) {
    // 删除图书
    if (db.deleteBookFromBookid(bookId)) {

```



```

        System.out.println("删除图书成功");
        return SUCCESS;
    } else {
        System.out.println("删除图书失败");
        return ERROR;
    }
}

```

4) 用户还书

读者归还书籍的步骤: 先判断还书时间是否不大于应还时间, 是则直接还书, 否则先计算超时金额, 并插入罚款表中, 然后再还书。首先获取还书时间, `Timestamp returnTime = new Timestamp(System.currentTimeMillis());` 然后获取超时金额, 如果等于 0, 表示没有超时或者超时小于 1 小时 (不包括 1 小时), 那就不用缴纳超时金额, 直接进入还书步骤。管理员操作归还书籍, 还书成功后还要更新书籍信息。

```

BorrowDao db = new BorrowDao();
if (db.returnBooks(id, time)) {
    System.out.println("更新还书信息成功, 等待将库存加1");
    // 还书成功, 将库存加1
    BooksDao db1 = new BooksDao();
    ReaderDao db2 = new ReaderDao();
    if (db1.updateBookNum(bookId, "increase") &&
        db2.updateReaderBorrowNumber(borrow_id, "reduce")) {
        System.out.println("修改对应书籍库存成功———还书成功");

        return SUCCESS;
    } else {
        System.out.println("修改对应书籍库存失败———还书失败");
        return ERROR;
    }
} else {
    System.out.println("更新还书信息失败");
    return ERROR;
}
}

```

6.4 数据库连接

这次的期末项目采用的 Mysql, 连接云服务器。先导入 JDBC 的 jar 包, 然后申明驱动程序的名字, 使用 `Url` 指向要访问的数据库名 `j2ee`, `bingqie` 声明

MySQL 数据库配置是的用户名和密码。加载驱动后，使用 `DriverManager.getConnection` 方法获取数据库连接，如果连接失败，则抛出异常：获取数据库连接失败。

```
public Database() {
    className="com.mysql.jdbc.Driver";
    url="jdbc:mysql://localhost:3306/j2ee";
    username="root";
    password="123456";
}

//获取数据库连接
public void getConnection(){
    try {
        Class.forName(className);    //加载驱动
        conn=DriverManager.getConnection(url, username, password);
    } catch (Exception e) {
        System.out.println("获取数据库连接失败");
    }
}
```

7. 故障处理和总结

7.1 故障处理

在正常情况下，应不出错。一旦发生意外，比如掉电、网络不通等，也应保证系统数据不会丢失。

故障出现后可能采取的变通措施，包括：

备份：使用附加存储设备备份数据。备份频率为每日一次。需手动备份。

恢复及再启动：如果数据造成丢失，可使用备份数据还原。

7.2 总结

我们不但成功的连接了数据库，也成功实现了很多功能：在管理员方面实现了管理员登陆、管理日志、管理员添加、管理员切换四个功能，在读者管理方面实现了读者添加、读者修改、读者查询、读者删除四个功能，在书籍管理方面实现了新书入库、书籍查询两个功能，在书籍借阅与归还方面实现了借阅书籍、归还书籍两个功能，在借阅超时方面也成功实现了此功能。在众多功能中刚开始时借阅书籍方面出了问题，但最后在全组人认真思考、仔细研究下终于改正了错误，使此功能终于可以正常使用了。

这次的课程设计可以看作是一次理论与实践相结合的桥梁，通过这次的课程设计，我们学习到了许多的知识，也认识到了自己目前的不足，那就是缺乏相应的知识与经验，所以在运用和操作方面都不是那么的得心应手。但是，经过这段时间对相关书籍的阅读和分析，我们组顺利的完成了设计，我还明白了在编写程序的时候，应该尽量使界面简洁大方，布局统一。变量类型的定义，一定要够用就好，这样程序就可以尽可能的减少对系统资源的占用。在设计时也免不了存在着一些不足，所以在今后的学习中我们会努力取得更大的进步，对于我们不足的地方希望老师能够及时给予批评，以便我们在今后的学习或工作中能够及时的改正。总之，这次课程设计为我提供了与众不同的学习方法，在书本中面对现实，为我们将来在社会上立足提供了良好的前提。

课程设计虽然结束了，但我感觉要想把课程设计做好，那必须把基础知识学透彻，当然更离不开团体的力量，因为团体的力量是无穷的。人生有数，学无止境，学习生活是无穷尽的，只有掌握更多的、更新的知识才能使自己始终立于不败之地。如果只为现在所取得的成绩而沉浸于喜悦、骄傲之中的话，那么，我想我们不会前进，只能停留在原来的地方，甚至还会倒退。所以我们应该从两周的课程设计中所暴露出的问题入手，不断完善自己的欠缺和不足。相信此次课程设计为我们以后继续从事计算机工作打下了扎实的基础，也明确了以后的学习方向。

附录 测试用例

1. 图书信息的维护用例

用例名：图书信息的维护。

参与执行者：图书管理员。

入口条件：图书管理员已经登陆到该系统中。

事件流：当有新入库时，图书管理员在录入页面输入书的信息，单击“提交”按钮，系统将书的信息保存到数据库中；当某一本图书的信息需要修改时，图书管理员通过输入查询条件，搜索出该书时，单击“修改”按钮，系统在可编辑状态显示图书的当前信息，图书管理员修改具体信息，单击“保存”按钮，系统将更新数据库中该书的信息；当需要删除一本或多本图书时，图书管理员查找到需要删除的图书记录，单击“删除”按钮，系统提示“确定要删除？”对话框，当管理员选择“是”时，系统将删除数据库中相应图书的信息，反之，则不做任何操作。

出口条件：系统将数据库中的信息进行相应的操作：添加图书信息时，将新的图书信息保存在数据库中；修改图书信息时，将数据库中该图书的信息做相应的更新操作；删除图书信息时，则删除数据库中的相应图书记录。

异常事件：在图书进行修改和删除时，先查出需要进行处理的图书记录，如果数据库中不存在符合条件的记录，查询无结果时，则无法进行修改和删除操作。

2. 读者信息的维护用例

用例名：读者信息的维护。

参与执行者：图书管理员。

入口条件：图书管理员已经登陆到该系统中。

事件流：当有新的读者时，图书管理员在录入页面输入书的信息，单击“提交”按钮，系统将读者的信息保存到数据库中；当某一读者的信息需要修改时，图书管理员通过输入查询条件，搜索出该读者信息时，单击“修改”按钮，系统在可编辑状态显示读者的当前信息，图书管理员修改具体信息，单击“保存”按钮，系统将更新数据库中该读者的信息；当需要删除一个或多个读者时，图书管理员查找到需要删除的读者记录，单击“删除”按钮，系统提示“确定要删除？”

对话框，当管理员选择“是”时，系统将删除数据库中该读者的信息，反之，则不做任何操作。

出口条件：系统将数据库中的读者信息进行相应的操作。添加读者信息时，将新的读者信息保存在数据库中；修改读者信息时，将数据库中该读者的信息做相应的更新操作；删除读者信息时，则删除数据库中的相应读者记录。

异常事件：在进行修改和删除读者信息时，先查出需要进行处理的读者记录，如果数据库中不存在符合条件的记录，查询无结果时，则无法进行修改和删除操作。

3. 图书信息的查询用例

用例名：图书信息查询。

参与执行者：图书管理员、读者。

入口条件：无。

事件流：通过交互界面输入查询条件搜索图书记录。

出口条件：若有符合条件的课程信息，则系统显示这些图书信息。否则系统提示用户重新输入查询条件。

4. 读者信息的查询用例

用例名：读者信息的查询。

参与执行者：图书管理员。

入口条件：用户已经登陆到该系统中。

事件流：通过查询界面输入查询条件搜索该读者记录。

出口条件：若有符合条件的读者信息，则系统显示读者信息。否则系统提示用户重新输入查询条件。

5. 查询个人基本信息用例

用例名：查询个人基本信息。

参与执行者：读者。

入口条件：用户已经登陆到该系统中。

事件流：单击“查询个人基本信息”按钮。

出口条件：系统显示读者本人信息。

6. 查询个人借阅信息用例

用例名：查询个人借阅信息。

参与执行者： 读者。

入口条件：用户已经登陆到该系统中。

事件流： 单击“查询个人基本信息”按钮。

出口条件：系统显示读者的借阅信息。

7. 借书用例

用例名：借书。

参与执行者： 管理员、读者。

入口条件：图书管理员已经登陆到该系统中。

事件流： 图书管理员在借书界面，输入图书编号好读者证号，单击“保存”按钮。

出口条件： 系统将这条借书记录保存到数据库中。

异常事件：如果图书为入库，则数据库中不存在该图书编号，提示“该书未入库”；如果数据库中不存在该读者证号，也相应给出提示。

8. 还书用例

用例名：还书。

参与执行者： 图书管理员、读者。

入口条件：图书管理员已经登陆到该系统中。

事件流：图书管理员在还书页面，输入图书编号，单击“还书”按钮。

出口条件： 系统将删除数据库中的该条借书记录。

异常事件：如果数据库中不存在这本书的借阅记录，提示“非本管借出的图书”，如果数据库中不存在该读者证号，也相应给出提示。

9. 口令管理用例

用例名：口令管理

参与执行者： 图书管理员、读者。

入口条件：用户已经登陆到该系统中。

事件流：用户单击“修改密码”按钮，在口令修改页面输入新的密码，单击“保存”按钮。

出口条件： 数据库中的密码被修改成最新的密码。