



江 苏 科 技 大 学

图 书 管 理 系 统

学 院：_____ 计算机学院

专 业：_____ 计算机科学与技术

小组成员：_____ 周 瑞 192210701143

_____ 朱庆壁 192210701144

指导教师：_____ 张明

2023 年 2 月

摘 要

本系统是一个对图书馆的图书和读者进行管理的操作系统，它能简化传统的图书馆管理模式，提升工作效率，便于管理和维护。该介绍从系统的研究意义、需求分析、功能模块、开发工具及数据库操作、代码设计等几个角度对系统进行分析。

本系统包括读者管理、图书管理、图书借（还）查询、登录和注册四个功能块，包括三个数据库表和六个信息展示界面。本系统利用了 Visual Studio Code、IntelliJ IDEA、MySQL 等开发工具，该系统采用了前后端分离的方式进行开发。运用了 Apache Maven 框架和 Java EE 提供的 HttpServlet 接口、由 Apache Tomcat 提供运行环境。利用 mybatis 框架实现 Java 和 MySQL 的交互。

目录

摘 要	I
目录	II
第一章 绪论	1
第二章 项目相关技术介绍	2
2.1 后端开发技术	2
2.1.1 JAVA 语言	2
2.2 前端开发技术	2
2.2.1 HTML、CSS 与 JavaScript	2
2.3 数据库技术	4
2.3.1 MySql.....	4
第三章 系统分析	5
3.1 可行性分析	5
3.1.1 经济可行性	5
3.1.2 管理可行性	5
3.1.3 技术可行性	5
3.2 系统综合要求	6
3.3 功能性分析	6
3.4 系统开发总体需求	9
第四章 总体设计	11
4.1 系统数据库设计原则	11
4.1.1 概念结构设计	11
第五章 详细设计	15
5.1 方案选择	15
5.2 数据库和 Java 代码连接实现	15
5.3 HttpServlet 接口和 tomcat 框架的连接实现	16
5.3.1 HttpServlet 接口	16
5.3.2 Tomcat 框架	16
5.4 后端总体代码结构和层次	17
5.4.1 项目分层的意义	17
5.4.2 pojo 层	17
5.4.3 mapper 层	17
5.4.4 service 层	18
5.4.5 utils 层	19
5.5 JavaScript 和 HttpServlet 的连接实现	19
5.5.1 HttpServlet 通信	19
5.5.2 JavaScript 数据通信	20
5.6 前端不同页面用户保持登录状态功能实现	22
第六章 系统实现与测试	25
6.1 登录和注册模块	25
6.1.1 登录模块	25

6.1.2	注册模块	25
6.2	主界面	26
6.2.1	登出功能	26
6.2.2	搜索功能	28
6.3	搜索界面	28
6.4	管理员管理书籍界面	29
6.4.1	书籍添加功能	29
6.4.2	书籍修改功能	31
第七章	结语	32
参 考 文 献	33

第一章 绪论

图书馆管理是校园管理中的一项重要业务，图书的阅读和学习是高校广大师生获取知识和信息的重要源泉。随着广大师生阅读需求的不断扩大，各类图书的更新迭代不断加快，传统的图书馆和图书管理模式已经不能满足师生日益增长的借阅需求。近年来，随着高校基础设施的更新换代，现代化的图书馆和图书管理模式进入高校，这同时也对高校图书馆的系统 and 制度建设提出了更高的要求，因此，建立信息化的图书管理系统对于图书借阅者和信息管理者来说，都有着极其重要的价值。

对于图书管理工作，在信息系统建立以前，数十万册图书的数据记录、信息保存更新、借阅登记等工作都需要图书管理员人工操作。大量繁复的工作不仅效率低下，而且出错率高。很多当时记录在册的图书在几经周转后，信息错误、遗漏等情况逐渐增多，对今后的图书管理工作会造成不利影响。在 IT 技术的日新月异下，高校图书馆的图书管理系统在近年来得到了迅速发展。相比于传统的手工管理模式，现代化的管理系统在书籍借阅和图书信息更新等方面有着较为明显的优势，如借阅效率提高，图书信息更新速度加快，管理过程中的成本支出降低，人力资源得到了充分的利用等，另一方面，通过现代化技术手段，也提升了学校图书管理的整体水平，为将来图书管理工作的扩展奠定了坚实基础。

通过对本地高校图书馆的走访调查后发现，当前部分高校图书馆的管理系统响应速度慢、功能不够完善，有的图书馆甚至还在使用传统的手工管理模式，该管理模式存在许多的不足：第一，工作效率低，当图书馆运行一段时间之后，会产生一系列的垃圾数据，在垃圾数据堆积之后，就会使读者在查询过程中难以找到实用信息，增加了查询难度，并且该管理模式极易发生数据丢失，一旦出现数据丢失，想要恢复，工作量及其巨大[1][1]，需要耗费大量的人力物力，并且在这一过程中难以保证数据准确性；第二，安全性不高，传统手工管理的数据信息都是保存在纸上，难以进行加密处理，很容易出现数据泄露的情况，难以保证数据的安全[2]。因此，本文依据当前成熟的计算机技术开发了本系统，目的在于完善这些图书馆管理系统的不足之处，提高图书馆工作人员的管理效率，满足读者的借阅需求。

第二章 项目相关技术介绍

网络软件开发需要诸多技术结合。后端开发技术有 JAVA 等，同时需要了解一定的项目开发流程，使用 Maven 等开发框架、搭建搭建基于 tomcat+HttpServlet 的互联网微服项目架构并实现项目功能。前端开发则要掌握 Html5、CSS、JavaScript 等技术，同时需要 Vue.js 等框架的支持。数据库技术数，则有两大主流数据库 SQL 和 ORACLE，通过数据库技术实现对数据的增加，删除，修改，查询等的管理。此外科学的软件项目管理对于项目的开展尤为重要[3]。接下来对于此次网络软件开发使用到的一些技术进行介绍。

2.1 后端开发技术

2.1.1 JAVA 语言

Java 是一门面向对象的编程语言，不仅吸收了 C++语言的各种优点，还摒弃了 C++里难以理解的多继承、指针等概念，因此 Java 语言具有功能强大和简单易用两个特征。Java 语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。

Java 具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点。Java 可以编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等。

Java 作为当下网络软件开发的主流语言，有着极为强大的实用功能，其跨平台的特点也极大的扩大了 Java 的应用面。

2.2 前端开发技术

2.2.1 HTML、CSS 与 JavaScript

(1) HTML

HTML 的英文全称是 Hyper Text Markup Language，即超文本标记语言。HTML 是由 Web 的发明者 Tim Berners-Lee 和同事 Daniel W Connolly 于 1990 年创立的一种标记语言，它是标准通用化标记语言 SGML 的应用。用 HTML 编写的超文本文档称为 HTML 文档，它能独立于各种操作系统平台(如 UNIX，Windows 等)[4]。使用 HTML，

将所需要表达的信息按某种规则写成 HTML 文件，通过专用的浏览器来识别，并将这些 HTML 文件“翻译”成可以识别的信息，即我们所见到的网页。

HTML5 是互联网的新一代标准，是构建以及呈现互联网内容的一种语言方式，被认为是互联网的核心技术之一。HTML 产生于 1990 年，1997 年 HTML4 成为互联网标准，并广泛应用于互联网应用的开发。

HTML5 是 Web 中核心语言 HTML 的规范，用户使用任何手段进行网页浏览时看到的内容原本都是 HTML 格式的，在浏览器中通过一些技术处理将其转换成为了可识别的信息。HTML5 在从前 HTML4.01 的基础上进行了一定的改进，HTML5 仍处于完善之中。然而，大部分现代浏览器已经具备了 HTML5 的支持。

（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。

从 HTML 被发明开始，样式就以各种形式存在。不同的浏览器结合它们各自的样式语言为用户提供页面效果的控制。最初的 HTML 只包含很少的显示属性，随着 HTML 的成长，为了满足页面设计者的要求，HTML 添加了很多显示功能。但是随着这些功能的增加，HTML 变的越来越杂乱，而且 HTML 页面也越来越臃肿。于是 CSS 便诞生了。

CSS 为 HTML 标记语言提供了一种样式描述，定义了其中元素的显示方式。CSS 在 Web 设计领域是一个突破。利用它可以实现修改一个小的样式更新与之相关的所有页面元素。CSS 具有丰富的样式定义、易于使用和修改、多页面应用、层叠、页面压缩等特点。

（3）JavaScript

JavaScript（简称“JS”）是一种具有函数优先的轻量级，解释型或即时编译型的编程语言。虽然它是作为开发 Web 页面的脚本语言而出名，但是它也被用到了很多非浏览器环境中，JavaScript 基于原型编程、多范式的动态脚本语言，并且支持面向对象、命令式、声明式、函数式编程范式。

JavaScript 最初由 Netscape 的 Brendan Eich 设计，最初将其脚本语言命名为 LiveScript，后来 Netscape 在与 Sun 合作之后将其改名为 JavaScript。JavaScript 最初受 Java 启发而开始设计的，目的之一就是“看上去像 Java”，因此语法上有类似之处，

一些名称和命名规范也借自 Java,但 JavaScript 的主要设计原则源自 Self 和 Scheme。JavaScript 与 Java 名称上的近似,是当时 Netscape 为了营销考虑与 Sun 微系统达成协议的结果。微软同时期也推出了 JScript 来迎战 JavaScript 的脚本语言。Vue.js 框架

Vue.js 是一套构建用户界面的渐进式框架。与其他重量级框架不同的是,Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层,并且非常容易学习,非常容易与其它库或已有项目整合。另一方面,Vue 完全有能力驱动采用单文件组件和 Vue 生态系统支持的库开发的复杂单页应用。

Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。Vue.js 由于只聚焦于视图层,所以自身不是一个全能框架。但是其非常容易学习,非常容易与其它库或已有项目整合。另一方面,在与相关工具和支持库一起使用时,Vue.js 也能驱动复杂的单页应用。

2.3 数据库技术

2.3.1 MySQL

MySQL 是一个关系型数据库管理系统,由瑞典 MySQLAB 公司开发,属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一,在 WEB 应用方面,MySQL 是最好的 RDBMS (Relational Database Management System, 关系数据库管理系统)应用软件之一[5]。

MySQL 是一种关系型数据库管理系统,关系数据库将数据保存在不同的表中,而不是将所有数据放在一个大仓库内,这样就增加了速度并提高了灵活性。

MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策,分为社区版和商业版,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,一般中小型和大型网站的开发都选择 MySQL 作为网站数据库。

第三章 系统分析

可行性，是在当前状况下，是否满足开发系统的条件。在系统开发之前，我们要针对该系统做详细的调查，然后根据调查报告来分析和总结，总结出系统如果开发会出现的问题，最后决定是否有必要开发这个系统，这个过程叫做可行性分析。管理系统实现的条件和可行性是息息相关的。系统的可行性分析与系统的风险分析是成反比的，开发的系统风险越大，那么此管理软件的可行性就会非常小。我通过对学院的调研和分析，对该管理系统做的可行性分析如下。

3.1 可行性分析

3.1.1 经济可行性

目标系统开发所需要求比较低，且系统不是十分复杂，开发的周期较短，人员经济支出有限。当系统开发完实际运行后，将会改变学校原有的图书手工管理，给许多读者带来方便，并且系统的开发将提高读者的时间利用率。

3.1.2 管理可行性

管理可行性主要考虑的内容主要有：①管理员是否支持此系统软件的开发；②学院是否具备管理的条件；③系统的后期管理工作是否能保证正常。伴随着电子信息的飞速发展，计算机已经全民普及，普通人员都具有一定的计算机知识，同时开发设计人员将数据信息标准化，系统界面简单化，所以系统的后期管理是非常方便的。

学校每个职工均有借书证，每个学生的学号也都不相同，同时每个读者在馆里都有自己的个人信息，因此此系统在管理上是可行的。另外，管理员非常支持开发此系统软件，同时在软件正式使用之前也会有相关的培训计划。

由上可知，开发此图书管理系统在经济、技术、管理上面都是可行的。

3.1.3 技术可行性

学校只需要建立一个局域网，并引入适当量的硬件设备就可以实现图书管理系统的应用，目标系统准备使用 Java 技术实现，目前这种技术已经普遍，因此在技术手段上实现本系统成为可能，高校也有计算机师资力量，对一定的软件师生有能力在一定时间内掌握。综上所述，目前实现目标系统的条件已经较为成熟。

3.2 系统综合要求

系统分析是在了解系统思想的基础上把复杂的模块分为多个简单的部分，然后把每个模块的属性了解清楚，每个模块之间的关系也需要知道。系统需求分析就是要详尽的阐述系统的功能和性能，保证开发的系统和用户的需求是非常统一的。需求分析可分为三个阶段来进行，有问题分析、需求阐述和需求审核。在问题分析的最初阶段，设计人员可以对用户进行访谈，也可以以会议的形式来了解用户的需求，设计人员也可以到实地去考察，去图书馆工作一段时间观察和了解管理员的工作。问题分析的关键技术有问题的抽象、问题的分析和需求建模。需求阐述时产生的需求说明的格式必须非常标准。需求规格的主体是由需求模型组成。在需求审核阶段，设计人员在审核需求说明时必须遵循一定的标准：需求一致性、需求正确性、需求可修改性、需求完整性、需求无歧义性、可验证性、可理解力性、可追踪性。

分析用户的需求是软件需求分析中一个重要的环节，也是最困难的，因为不是所有用户对计算机都十分了解，所以开发人员要尽可能的多与用户沟通,明确用户的真实需求。对用户的需求了解的越详细，软件后期维护就越容易。对于图书管理系统来说，读者足不出户就可以查看新书公告、查看和修改自己的个人信息、借阅的图书信息、已过期的图书信息、浏览所需要的图书，可以查看最近一段时间借阅率高的图书。管理员所需要的功能如下：第一，能够对图书进行管理（包括新书的录入和删除以及图书的分类管理）；第二，对读者进行管理（添加、修改和删除读者信息）；第三，发布公告信息；第四，有借书、还书、续借功能。

3.3 功能性分析

功能需求分析是设计人员对管理员和借阅者需要的系统功能进行分析，它是建立系统的基础，需求分析是否到位，决定这个系统在今后的使用效果。经过对我院图书馆的详细调研，在满足用户需求的基础上合理规划系统，经过优化整理确定系统最终要运行的主要功能包括：图书管理、借阅者管理、系统管理、统计排行、图书借还、图书检索等功能。

（1） 系统管理

系统管理实现的功能主要包括本馆介绍功能、管理员的设置功能、操作权限设置功能、口令更改功能、系统参数设置功能、系统提醒功能、数据维护功能等功能，用例图如图 3.1 所示。

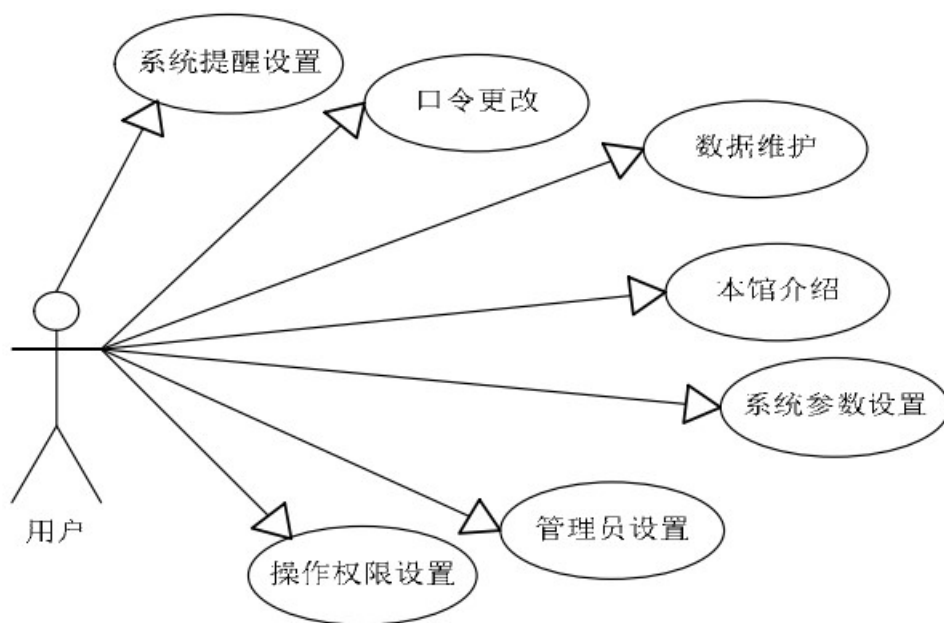


图 3.1 系统管理用例图

(2) 图书管理

图书管理实现的功能包括新书录入功能、图书出库功能、图书修改功能、新书公告功能。用例图如图 3.2 所示。

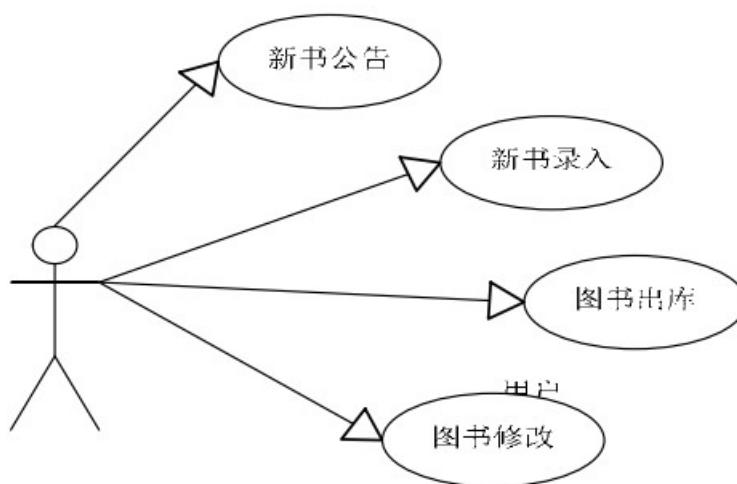


图 3.2 图书管理用例图

(3) 借阅者管理

借阅者管理实现的功能有：查询个人信息功能、修改个人信息功能、借阅证挂失功能、密码修改功能。借阅者管理用例图如图 3.3 所示。

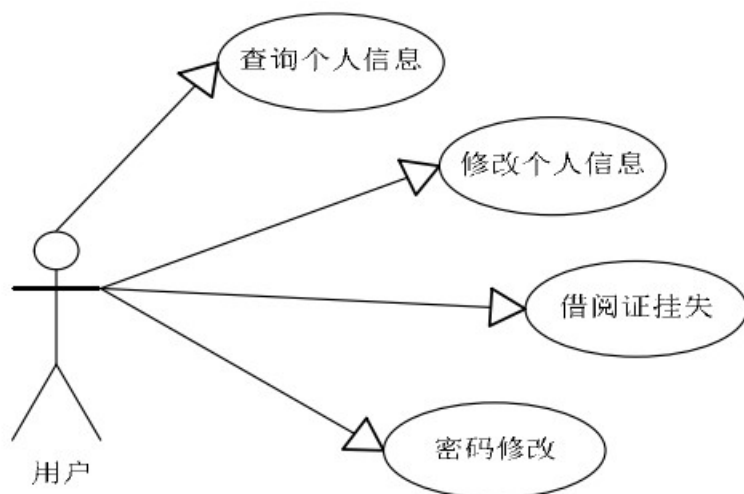


图 3.3 借阅者管理用例图

(4) 图书借还

图书借还实现的功能包括：预约功能、预借功能、借阅功能、续借功能、归还功能。用例图如图 3.4 所示。

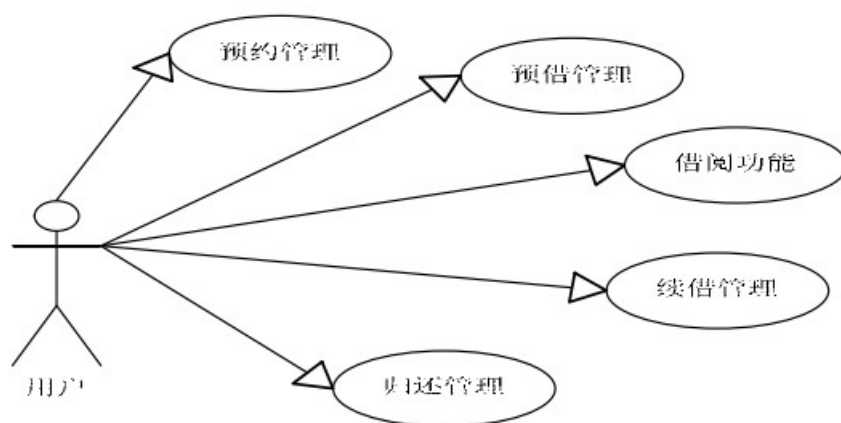


图 3.4 图书借还用例图

(5) 图书检索

图书检索实现的功能包括：按书名检索、类别检索、作者检索、出版社检索、图

书编号检索、关键字检索等功能。用例图如图 3.5 所示。

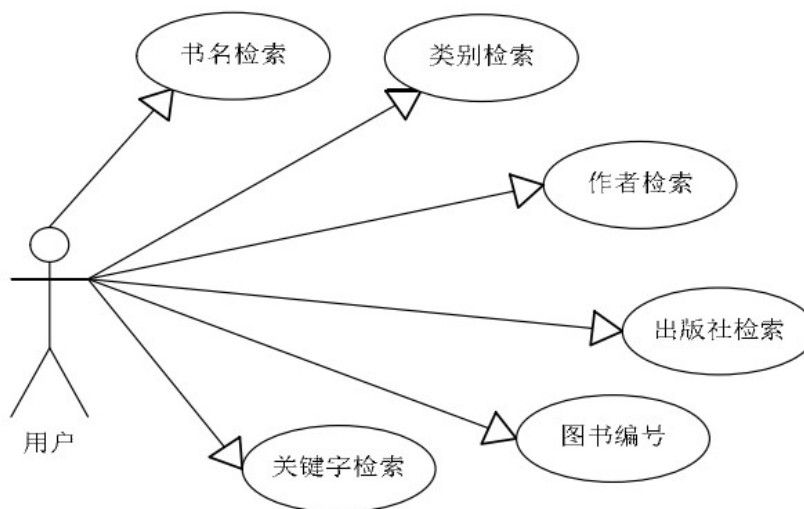


图 3.5 图书检索用例图

(6) 图书推荐

图书推荐实现的功能有图书借阅数量排行、图书借阅分类排行、读者借阅分类排行功能。用例图如图 3.6 所示。

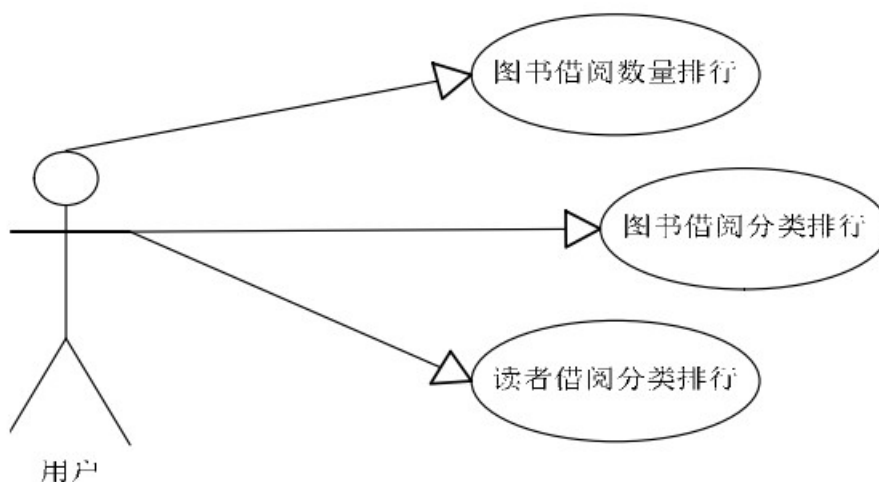


图 3.6 图书推荐用例图

3.4 系统开发总体需求

建设图书管理系统会涉及用户的实际操作和使用体验，因此建设过程要遵循这些原则：

(1) 易用性：尽可能设计简单易操作的图书管理系统来达到用户的标准。用户偏向功能全面、简单上手、通俗易懂的图书管理系统，当然，系统的设置也要符合读者和图书馆工作人员的日常使用习惯，以降低用户掌握系统的困难指数。

（2）安全性：图书管理系统的本质是一套能够对大量数据进行存储管理的系统，所以本系统需要相当高的安全性能和针对数据的严密防护体系，每一个登录系统的用户都需要进行严格监管，以防信息外泄带来不可预估的重大损失。

（3）稳定性：图书馆的用户群体较大，所以系统的稳定性方面也有一定要求，系统应该采取故障恢复和自我检查等方法来保证其稳定性。

（4）经济性：图书管理系统主要是给图书馆工作人员和在校学生使用，因此系统设计的要尽可能的节约成本。

（5）灵活性：图书管理系统应可以对图书馆实际情况的变化及时进行调整，以提高系统灵活的部署能力。

第四章 总体设计

总体设计中首先寻找实现目标系统的各种不同的方案，需求分析阶段得到的数据流图是设想各种可能方案的基础。在总体阶段确定软件的结构，确定系统中每个程序是由哪些模块组成的以及这些模块相互之间的关系。

4.1 系统数据库设计原则

在设计数据库时要以功能需求中的数据为衡量标准，遵循以下的原则：

（1）一致性。要求在系统的全程过程中表示同一事物的数据具有正确性和唯一性，数据结构的一致性，尤其在关系数据库中，更加要求当事务完成时，必须使所有数据都具有一致的状态。

（2）通用性。系统中的数据不仅仅是一个模块对它操作，是所有的模块都能对它进行操作（查询、修改等），这就要求数据具有很好的通用性。

（3）规范性。规范性是大多事物都具有的性质，数据也不例外，只有保证数据的规范性才能确保数据不会冗余、修改正常等问题，才能不影响系统运行。

（4）扩展性。用户的功能需求不会一成不变，也包括前期的需求分析工作再全面也可能会有遗漏的问题，所以必须在设计的时候保证数据的可扩展性、伸缩性和移植性等。

（5）可维护和管理。系统中的数据是可以随时变化的，所以要确保后期数据具有可维护和方便管理的性质。

（6）安全性。要保证系统的正常运行，不仅数据要正确，而且要确保数据的安全，确保数据不能被恶意篡改。

4.1.1 概念结构设计

概念设计就是在数据需求分析后，运用自底向上的方法建立的概念结构。概念模型是站在用户的角度把用户需要的信息和数据转化成看得见的结构，这里概念模型我们用 E-R 图来表示。E-R 图中矩形表示实体集，椭圆则表示属性，菱形表示实体之间的联系。

图书管理系统涉及的实体主要有读者、管理员、图书等，各实体的属性如下所示：

（1）借阅者信息的数据项有：借阅者姓名、借阅者密码、借阅者学号、性别、班级、电话号码、已借书数量等。

(2) 管理员信息的数据项：管理员姓名、工号、密码等。

(3) 图书信息的数据项有：书名、图书编号、作者、出版社、类别、入库日期、图书的总数量、图书当前数量等。

(4) 借书信息的数据项有：借阅者学号、图书编号、借书日期等。

(5) 图书检索的数据项有：书名、类别、关键字、图书编号、作者、出版社等。

(6) 图书推荐的数据项有：图书借阅数量、图书借阅分类、读者借阅分类等。

根据上面得到的实体和实体之间的联系，就能得出系统的 E-R 图。

(1) 借阅者 E-R 图设计

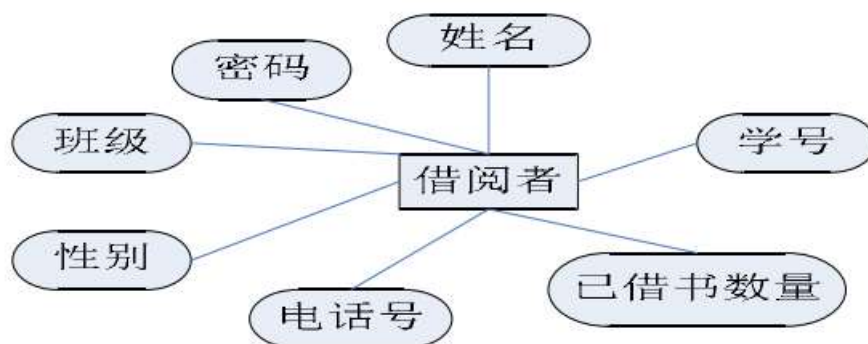


图 4.1 读者 E-R 图

(2) 管理员 E-R 图



图 4.2 管理员 E-R 图

(3) 图书信息 E-R 图

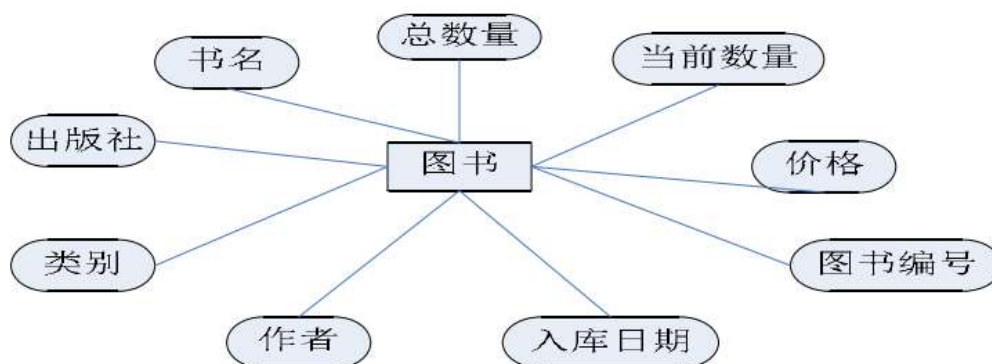


图 4.3 图书信息 E-R 图

(4) 借书 E-R 图

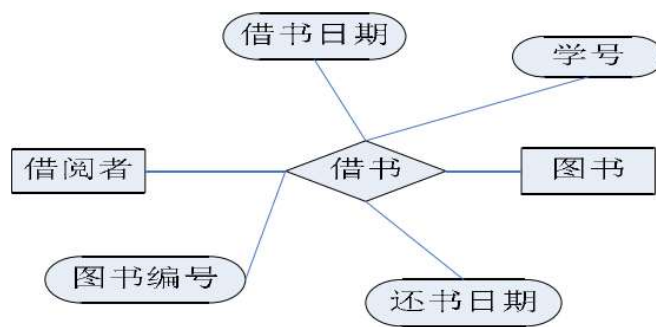


图 4.4 借书 E-R 图

(5) 图书检索 E-R 图

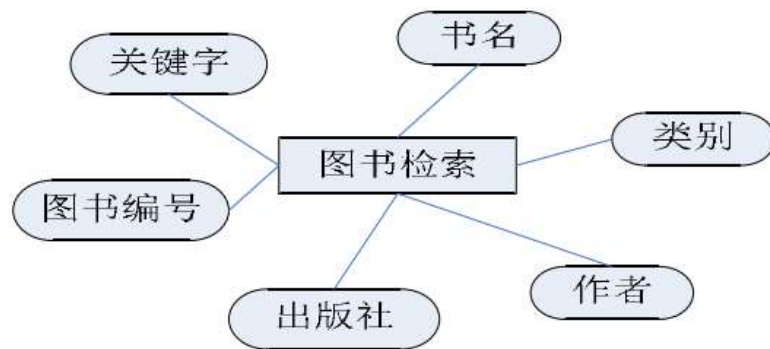


图 4.5 图书检索 E-R 图

(6) 图书推荐 E-R 图

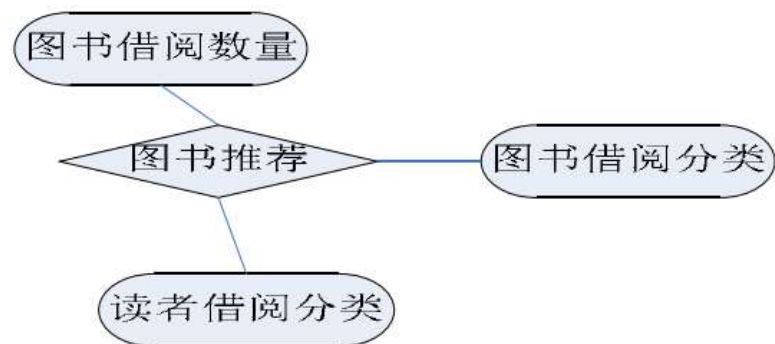


图 4.6 图书推荐 E-R 图

(7) 综合以上模型，得到的整个系统的 E-R 图设计

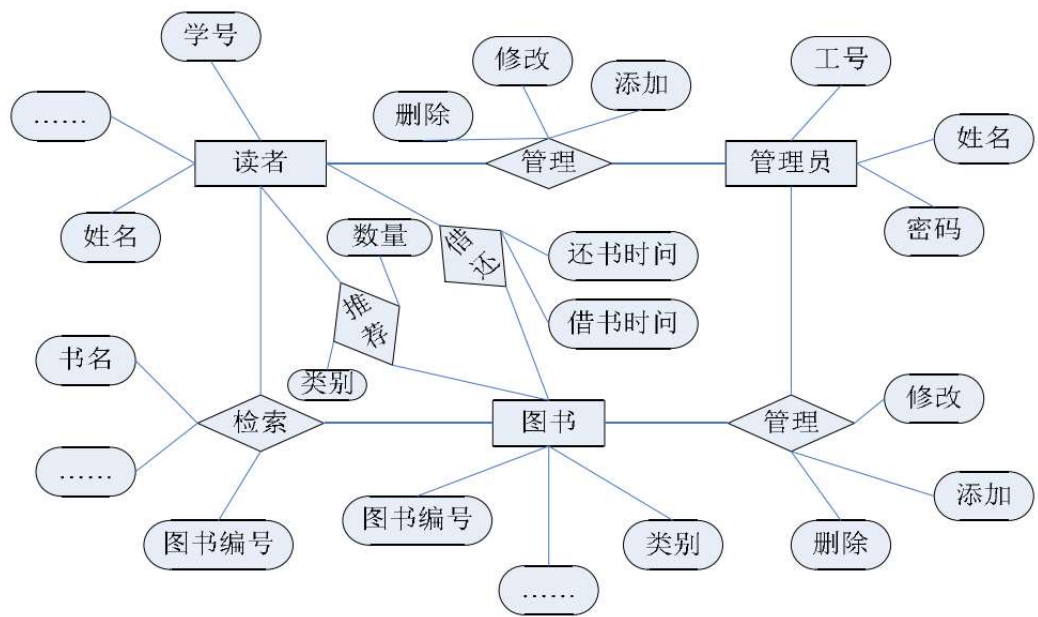


图 4.7 系统 E-R 图

第五章 详细设计

5.1 方案选择

本项目采用前后端分离的设计，前端主要采用 HTML 和 jQuery 框架，通过 JavaScript 实现前端页面和后端服务器间的交互。后端运用了 Apache Maven 框架和 Java EE 提供的 HttpServlet 接口、由 Apache Tomcat 提供运行环境。利用 mybatis 框架实现 Java 和 MySQL 的交互。

5.2 数据库和 Java 代码连接实现

这一部分内容在 Java 代码中主要用到了 mybatis 框架，使用了 MySQL 数据库。由于项目采用的是 Maven 框架，我们可以通过该框架的 pom.xml 文件很方便的导入我们项目所需要的各种依赖，而该小节中的 mybatis 依赖就是通过 pom 文件所添加的。依赖添加代码如图 5.1 所示。

```
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.5.11</version>
</dependency>
```

图 5.1 依赖代码

依赖添加完后，下一步便是配置依赖。mybatis 的配置需要借助 xml 配置文件来实现首先我们需要选定将 Java 代码连向 MySQL 的数据库驱动，而这里我们可以选择 com.mysql.cj.jdbc.Driver，其次我们需要向 mybatis 指明数据库的包含端口连接路径和对数据库进行操作的用户，需要包含用户名和登录密码最终的主要配置代码如图 5.2 所示。

所谓用 Java 代码操作数据库，其实并不是真正用 Java 代码来直接与数据库做交互，而是需要将特定的 Java 代码映射为对应的 MySQL 代码，而这就要引入我们的 mybatis 映射文件了。和配置文件一样，映射文件同样通过 xml 文件进行配置。就像前面提到的一样，映射文件的主要作用就是将 Java 代码映射为开发者想要的 SQL 语句。而这就是简单的将 SQL 语句写进映射文件，写好之后我们还要将该映射文件在

项目中的路径添加到配置文件中。

最终，我们便可以通过自己定义的 mapper 接口来实现通过 Java 代码来对数据库中的数据进行增删改查操作。

```
<dataSource type="POOLED">
  <property name="driver" value="com.mysql.cj.jdbc.Driver"/>
  <property name="url" value="jdbc:mysql://127.0.0.1:3306/homework"/>
  <property name="username" value="root"/>
  <property name="password" value="123456"/>
</dataSource>
```

图 5.2 数据库配置

5.3 HttpServlet 接口和 tomcat 框架的连接实现

5.3.1 HttpServlet 接口

所谓 HttpServlet，就是 JavaEE 提出的 13 个规范之一，其中上节提到的 xml 文件也是 JavaEE 的规范之一。看到 Http 这个词便可以联想到网页。而这个接口这是用来实现 Java 代码和网页之间的交互的。

而说到 Http，就不得不提我们最常遇到的几个数据请求方式了。分别是 GET，POST 和 OPTIONS 请求方式。而 HttpServlet 接口就帮我们对前端以不同的请求方式发送过来的数据分别提供了不同抽象方法。我们对不同的请求方式只需要分别重写 HttpServlet 接口不同的方法便可以接受到对应的数据。比如说一个简单的 OPTIONS 方式的重写如图 5.3 所示。

```
@Override
protected void doOptions(HttpServletRequest req, HttpServletResponse resp){
    resp.setHeader("Access-Control-Allow-Origin", "http://127.0.0.1:4040");
    resp.setHeader("Access-Control-Allow-Headers", "content-type");
    resp.setHeader("Access-Control-Allow-Credentials", "true");
}
```

图 5.3 OPTIONS 方式的重写

重写好不同的方法后还不足以做到 Java 代码和 Web 之间的交互，我们还要给不同的 Service 类分别绑定一个路径，我们可以注解的方式来绑定路径，同样也可以通过 xml 配置文件进行配置。

5.3.2 Tomcat 框架

前面提到的 `HttpServlet` 接口，我们讨论的一直都是接口，而不是实现类。众所周知，接口是无法直接实现功能的，Java 语言中真正实现各种各样的功能是它内部丰富的实现类。所以仅凭 `HttpServlet` 还是无法实现 Java 代码和 Web 间的交互。真正实现这类功能的实现类被继承到了 `tomcat` 中，而这就体现了 `tomcat` 的重要性。

最终我们的 Web 项目要运行起来，就是要将项目部署到 `tomcat` 服务器上。而一般的 Java 项目最终会被封装为 `jar` 文件，而由于 `tomcat` 是一个 Web 服务器，我们若想将项目部署到该服务器上需要将我们的 Java 项目封装为 `war` 文件。而这只需要在 Maven 框架的 `pom` 文件内添加一句项目的封装方式就能实现了。得到了项目的 `war` 文件后，我们需要将该文件放置到 `tomcat` 文件内的 `webapps` 目录下。到这一步，我们的 Java 代码便可以接收到 Web 传来的各类请求数据了。

5.4 后端总体代码结构和层次

5.4.1 项目分层的意义

项目分层是为了符合“高内聚，低耦合”思想，把各个功能模块划分为表示层(UI)、业务逻辑层(BLL)和数据访问层(DAL)三层架构，各层之间采用接口相互访问，并通过对象模型的实体类(Model)作为数据传递的载体，不同的对象模型的实体类一般对应于数据库的不同表，实体类的属性与数据库表的字段名一致。

分层后开发人员分工更明确，将精力更专注于应用系统核心业务逻辑的分析、设计和开发，加快项目的进度，提高了开发效率，有利于项目的更新和维护工作。一个开发中的项目层次结构大致如图 5.4 所示。

5.4.2 pojo 层

该层的内容即为数据库表对应的实体类，与数据库中的属性值基本保持一致。该层也可以称为 `model` 层或者 `domain` 层。而所谓实体类，便是数据库中 Table 在 Java 代码中的具体体现，而由于 Java 中的一切内容都是以类为单位的，所以该层的内容也被封装为一个个不同的类，每个类都与数据库中某个表之间有一个唯一的对那个关系。

5.4.3 mapper 层

又被称作 DAO 层，中文叫数据访问层，全称为 Data Access Object。某个 DAO

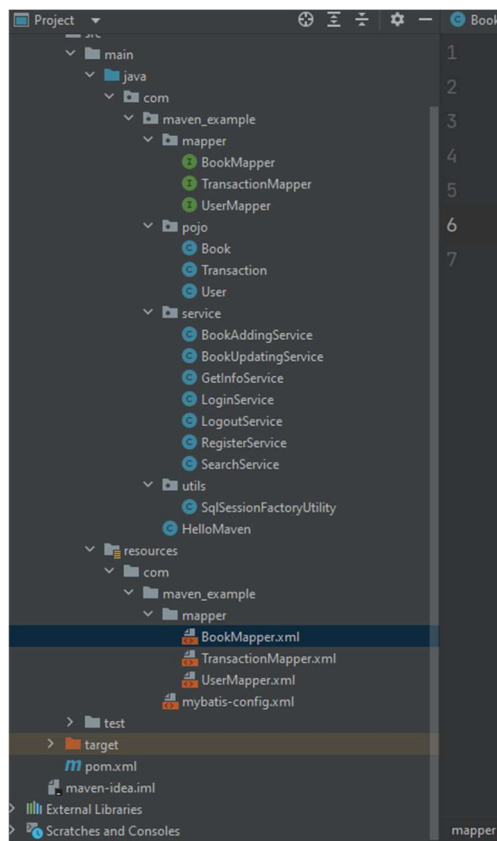


图 5.4 项目文件结构

一定是和数据库的某一张表一一对应的，其中封装了 CRUD（增加 Create、检索 Retrieve、更新 Update 和删除 Delete）基本操作，DAO 只做原子操作。无论多么复杂的查询，DAO 只是封装增删改查。至于增删查改如何去实现一个功能，该层并不负责，该层只需要将每一个细节操作封装为可以被上层的类调用的函数。

在“数据库和 Java 代码连接实现”一节中也提到了 mapper 层，在本项目中，mapper 层主要包含了 mybatis 的映射文件对应的多个接口，接口内实现类对各各数据库表的各种操作的抽象方法，具体实现操作数据库的 MySQL 代码存放在 mybatis 的映射文件中。

5.4.4 service 层

service 层属于业务层，一般项目的业务逻辑就放在该层中。该层的类会调用上面提到的 mapper 层内的接口。在本项目中给层中的一个类就对应着一个业务功能，比如添加图书的功能就对应着该层中的 BookAddingService 类。在类中，一个标准的流程就是先通过 mybatis 的 SqlSessionFactory 类来建立一个该服务类当前与数据库的一个连接，也就是 SqlSession 类，通过这个连接获得 mapper 层接口中对应的类以便通过类间接对数据库进行操作。

为了保证能顺利和 Web 进行沟通，这期间可能还会遇到一些跨域问题或者关于 Session 或者 Cookie 的传递，我们还需要对响应头添加一些设置，向接收响应的 Web 说明该响应具有哪些性质以便 Web 能够正常接收响应。

5.4.5 utils 层

准确来说该层并不属于业务处理的层次，该层的主要任务是提供一些其他层可能会经常用到的工具，方便其他层的类来对该层的类进行调用，这样可以大量的减少项目中的重复代码。比如在本项目中，utils 层中放有一个 SqlSessionFactory 的工具类，因为 service 层中每个需要操作数据库的功能都需要先与数据库建立一个连接，而建立连接需要 SqlSessionFactory 对象，因此该对象会被 service 层中的大多数类所调用，而如果我们分别在每个 service 类中都分辨建立一个 SqlSessionFactory，除了会导致大量的重复代码外，更为严重的是该类会占用系统大量的资源，所以我们应该在 utils 层中建一个工具类，此后每当项目中的某个类需要到对应工具时，就可以直接通过该工具类获得该工具，大量提升了系统运行的效率和项目代码的可阅读性。

5.5 JavaScript 和 HttpServlet 的连接实现

5.5.1 HttpServlet 通信

前后端想要相互通信，就必须事先约定好发送的数据的编码和解码的格式，而在这点上 HttpServlet 似乎做得和其他很多框架或者工具不太一样。因为该项目中几乎所有框架的默认编码方式都是“UTF-8”，而 HttpServlet 的默认编码方式却是“ISO-8859-1”，这就导致了一个很严重的问题。比如说在添加图书的时候，该图书管理系统内几乎所有的书都是中文的，包括书名、作者、描述等信息，同样存在数据库中的这些信息也是中文的。当我们在实现查询功能的时候，我们需要根据用户的输入来向 Web 返回一个包含大量书本信息的列表，也就是一个 List。这时候如果我们采用 HttpServlet 默认的编码方式，Web 端收到的信息就全部以‘?’符号呈现，其中一个结果如图 5.5 所示。

因此，前后端在相互通信前必须事先约定好请求和相应格式，不仅仅是从编码上，还要从传递的数据类型上。至于编码上，我们只需要给 HttpServlet 的返回的响应头和接收到的请求的相应编码类型手动第设置为“UTF-8”的字符集即可。在数据类型上，本项目中前后端约定除了简单的字符串响应数据外，其他所有信息均通过 JSON 来传

递，前端将数据封装成 JSON 字符串，后端便通过解析该 JSON 字符串获得对应的 pojo 层的类，并通过该类向 mapper 层的接口传递参数。

而为了解析由 JavaScript 传递过来的 JSON 字符串，本项目中引入了阿里云的一

书名	作者	描述	数量
???	???	??...	12
????	??	??1...	34
?Q??	??	?Q????????????????1921?12??????????????...	34
??	???	????????????????????????????????????2??????3????...	24
???	???	??1...	36
???	???	??...	12
????	???	??...	32
??	??	????????????????????????????1918??1922????14??...	8

图 5.5 错误书籍信息列表

个依赖，叫做 fastjson，如图 5.6 所示。面对由 JSON 字符串，我们可以调用该依赖的 JSON.parseObject 方法，将收到的 JSON 字符串和想要得到的类对象的字节码文件作为参数传递给该方法就可以顺利的收到前端传来的请求数据了。相应的，在返回数据时，我们可以将当前的数据或者列表作为参数传递给 fastjson 依赖的 JSON.toJSONString 方法，便可以得当前数据对应的 JSON 字符串，随后我们再将该字符串写入响应的打印流当中就可以让前端收到后端响应的数据了。

```
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>2.0.22</version>
</dependency>
```

图 5.6 配置 JSON 依赖

5.5.2 JavaScript 数据通信

这里以用户的登录功能为例。用户输入用户名和密码后，我们需要将用户名和密码传递给后端服务器。首先通过 `Document` 方法取得对应输入框的值，即 `value` 属性，然后使用 `POST` 的数据请求方式将两个字符串封装为 `JSON` 字符串向后端发送数据。但是，仅仅这样是完全不能让后端接受刚刚发送的请求参数的。我们这时候可能会收到浏览器返回的类似图 5.7 的错误信息，出现这个错误信息的原因在于这里涉及到了前后端相互通信的跨域问题以及 `preflight` 的相关知识。

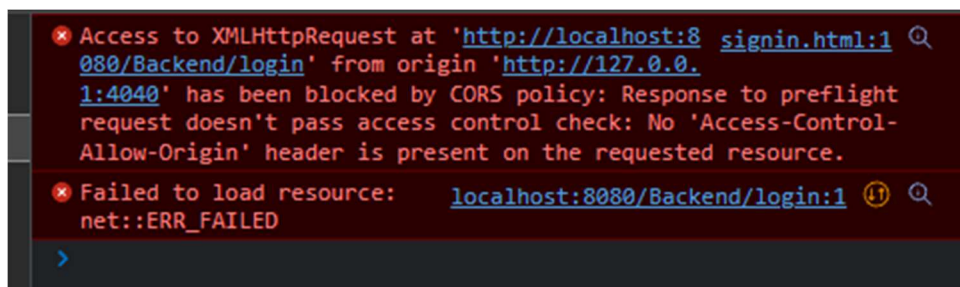


图 5.7 JSON 数据接受错误

首先来介绍一下什么是跨域问题。跨源资源共享（`CORS`，或通俗地译为跨域资源共享）是一种基于 `HTTP` 头的机制，该机制通过允许服务器标示除了它自己以外的其它源（域、协议或端口），使得浏览器允许这些源访问加载自己的资源。跨源资源共享还通过一种机制来检查服务器是否会允许要发送的真实请求，该机制通过浏览器发起一个到服务器托管的跨源资源的“预检”请求。在预检中，浏览器发送的头中标示有 `HTTP` 方法和真实请求中会用到的头。

跨源 `HTTP` 请求的一个例子：运行在 `https://domain-a.com` 的 `JavaScript` 代码使用 `XMLHttpRequest` 来发起一个到 `https://domain-b.com/data.json` 的请求。

出于安全性，浏览器限制脚本内发起的跨源 `HTTP` 请求。例如，`XMLHttpRequest` 和 `Fetch API` 遵循同源策略。这意味着使用这些 `API` 的 `Web` 应用程序只能从加载应用程序的同一个域请求 `HTTP` 资源，除非响应报文包含了正确 `CORS` 响应头。

`CORS` 机制允许 `Web` 应用服务器进行跨源访问控制，从而使跨源数据传输得以安全进行。现代浏览器支持在 `API` 容器中（例如 `XMLHttpRequest` 或 `Fetch`）使用 `CORS`，以降低跨源 `HTTP` 请求所带来的风险。

跨源资源共享标准新增了一组 `HTTP` 标头字段，允许服务器声明哪些源站通过浏览器有权限访问哪些资源。另外，规范要求，对那些可能对服务器数据产生副作用的 `HTTP` 请求方法（特别是 `GET` 以外的 `HTTP` 请求，或者搭配某些 `MIME` 类型的 `POST` 请求），浏览器必须首先使用 `OPTIONS` 方法发起一个预检请求（`preflight request`），从而获知服务端是否允许该跨源请求。服务器确认允许之后，才发起实际

的 HTTP 请求。在预检请求的返回中，服务器端也可以通知客户端，是否需要携带身份凭证（例如 Cookie 和 HTTP 认证相关数据）。

从图 5.7 的信息中我们可以得知该错误产生的原因是预检请求(preflight request)没能通过访问控制检测。而预检请求是使用 OPTIONS 方法发起的，所以我们在接受用户输入的信息时，不能只重写 HttpServlet 的 doPost 方法，还需要重写 doOptions 方法以对预检请求进行处理。就像图 5.7 所提示的那样，我们需要在重写的 doOptions 方法中给响应头设置 Access-Control-Allow-Origin 参数，将该参数与前端的 url 地址进行绑定。另外还应注意到我们的数据是以 JSON 字符串来进行传递的，所以只设置 Access-Control-Allow-Origin 参数还是不够的，这依然不能让预检请求通过访问控制检测。我们还需要给 HttpServlet 的响应头设置 Access-Control-Allow-Headers 参数，以向浏览器表明该服务器可以接受什么样的请求头。同样，在重写的 doPost 方法中我们同样要给响应头设置这两个参数。

至此，我们前端和后端就可以相互发送数据并保证对方能正常接收到了。

5.6 前端不同页面用户保持登录状态功能实现

在上一节通过系统的登录功能描述用户输入的用户名和密码在前后端之间传输大致说明了在跨域情况下前后端之间通信会遇到的几个问题，但我们收到用户输入的数据并向浏览器返回登录成功的信息后，仅仅通过 JavaScript 和 HTML 是没法让用户在除了登录界面的其他页面保持登录状态的。如果我们希望在用户登陆成功后页面自动跳转到主页，同时保持用户的登录状态，比如在页面右上角显示当前用户的用户名和头像之类的功能，像这种功能在目前介绍到的技术下还是没法实现的。概括地说就是我们希望让前端收到数据后能将数据存储下来。而这，就要介绍一下 Cookie 和 Session 的相关知识了。

首先说一下 Cookie。HTTP Cookie（也叫 Web Cookie 或浏览器 Cookie）是服务器发送到用户浏览器并保存在本地的一小块数据。浏览器会存储 cookie 并在下次向同一服务器再发起请求时携带并发送到服务器上。通常，它用于告知服务端两个请求是否来自同一浏览器——如保持用户的登录状态。Cookie 使基于无状态的 HTTP 协议记录稳定的状态信息成为了可能。

Cookie 主要用于以下三个方面：（一）会话状态管理：如用户登录状态、购物车、游戏分数或其它需要记录的信息；（二）个性化设置：如用户自定义设置、主题和其他设置；（三）浏览器行为跟踪：如跟踪分析用户行为等。

从前面的介绍可以看出仅仅使用 Cookie 我们便可以实现保持用户的登录状态这一功能，但是 Cookie 的一个局限性在于：它是存储在客户端的，而存储在客户端就会带来很大的安全隐患。尤其是在对用户名和密码这些敏感数据而言，将这些信息存储在客户端无疑给项目带来了极大的安全隐患。因此我们需要一种更安全的方案来存储用户的敏感数据。而这种更安全的方案便是 Session。

与 Cookie 类似，Session 同样是用来存储各客户端的信息的。但有一个根本性的区别就是：Cookie 存储在客户端，而 Session 是存储在服务器端的，正是因为这一点，同样的数据，使用 Session 存储会比使用 Cookie 存储安全的多。而服务器判断不同的请求是否来自同一个浏览器的功能的实现就通过浏览器向服务器返回的 SessionID 来判断。

当用户第一次登录服务器时，服务器会向浏览器返回一个 SessionID，SessionID 实际上是一个字符串，每一个在一定时限内访问过服务器的浏览器都会有一个唯一 SessionID 来唯一地标识自身的身份。而 SessionID 是通过 Cookie 最开始由服务器像浏览器发送的，所以说 Session 是基于 Cookie 实现的。与直接将数据存储在 Cookie 中不同，通过 Session 我们可以将数据存在服务器，而从客户端所看到的，就是一串 SessionID。而此后每当浏览器像访问服务器时，一般情况下浏览器都会自动携带 SessionID 一项服务器表明自己的身份。

可在跨域的情况下，浏览器在向后端传递数据时默认是不会带上 Cookie 或者 SessionID 等验证信息的，我们需要给 XMLHttpRequest 对象设置 withCredential 属性。将该属性的值设置为 true，即表明我们需要在跨域的情况下向服务器发送携带 Cookie 或 SessionID 等用于验证身份的数据。而这时我们再向服务端发送数据依然没能实现保持用户登录状态的功能。我们很可能收到如图 5.8 所示的错误提示。

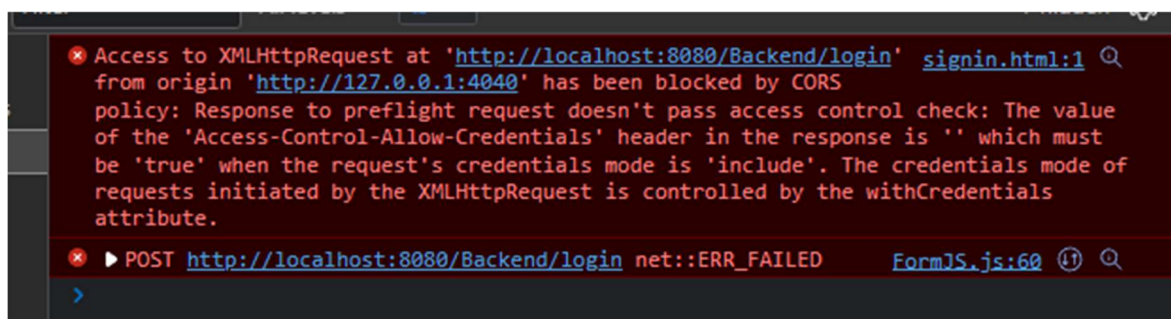


图 5.8 用户登录错误

根据提示我们可以知道，若想在跨域模式中传递身份验证信息，不能只有客户端或者服务器端表示能够发送或接受该类型的数据。我们还需要在服务器的给响应数据设置 Access-Control-Allow-Credentials 参数。

`Access-Control-Allow-Credentials` 响应头用于在请求要求包含 `credentials` (`Request.credentials` 的值为 `include`) 时, 告知浏览器是否可以将对请求的响应暴露给前端 JavaScript 代码。

当请求的 `credentials` 模式 (`Request.credentials`) 为 `include` 时, 浏览器仅在响应标头 `Access-Control-Allow-Credentials` 的值为 `true` 的情况下将响应暴露给前端的 JavaScript 代码。

`Credentials` 可以是 `cookies`、`authorization headers` 或 `TLS client certificates`。

当作为对预检请求的响应的一部分时, 这能表示是否真正的请求可以使用 `credentials`。注意简单的 `GET` 请求没有预检, 所以若一个对资源的请求带了 `credentials`, 如果这个响应头没有随资源返回, 响应就会被浏览器忽视, 不会返回到 web 内容。

`Access-Control-Allow-Credentials` 标头需要与 `XMLHttpRequest.withCredentials` 或 `Fetch API` 的 `Request()` 构造函数中的 `credentials` 选项结合使用。`Credentials` 必须在前后端都被配置 (即 `Access-Control-Allow-Credentials` header 和 `XHR` 或 `Fetch request` 中都要配置) 才能使带 `credentials` 的 `CORS` 请求成功。

至此, 我们从数据库到浏览器的连接功能算是基本完成了。

第六章 系统实现与测试

6.1 登录和注册模块

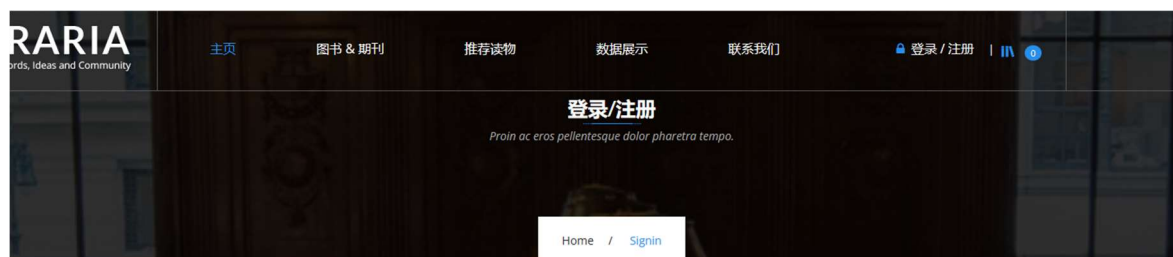


图 6.1 登录与注册

6.1.1 登录模块

从图 6.1 可以看出，在本系统中，我们将登录和注册功能设置在了同一界面。用户在输入框中分别输入用户名和密码。而且用户输入的信息并不会立马交给后台进行检验。用户输入完后，会首先在前端检测用户输入的用户名和密码是否符合规范，比如用户名的长度应大于等于三，密码的长度大于等于六。输入框中还限制了用户的输入长度，用以防止有人蓄意提升服务器运行压力。

当用户输入的用户名不符合规范时，浏览器给出的提示如图 6.2 所示。而当用户输入的用户名和密码不匹配时，浏览器给出的提示如图 6.3 所示。若用户输入了正确的信息，浏览器会自动重定向到网站的主页。

6.1.2 注册模块

与登录模块类似，注册模块也会在将数据交给后台处理前先验证用户的输入是否



图 6.2 用户名提示

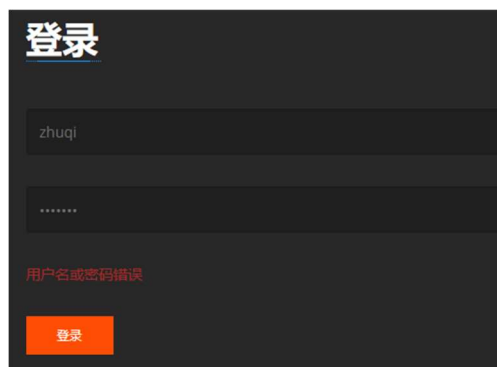


图 6.3 用户信息错误提示

符合规范当用户输入的注册信息长度不符合规范时，浏览器给出的提示如图 6.4 所示。同时，若用户未对不规范的数据进行修改，此时的“注册”按钮点击后是无效的。

注册一个新的账号

如果您之前没有本网站的账号，可以现在注册一个属于自己的账号，

图 6.4 注册账号信息提示

由于用户名是表示用户身份的标识符，所以用户输入的用户名不能与数据库中已有的用户名出现重复。若当前用户输入的用户名已经被其它用户注册过了，浏览器给出的提示如图 6.5 所示。若用户注册成功，浏览器给出的提示如图 6.6 所示。注册成功后，用户便可以用刚注册的账户进行登录并跳转到主页了。

6.2 主界面

6.2.1 登出功能

如果您之前没有本网站的账号，可以现在注册

用户名已存在!

注册

如果您之前没有本网站的账号，可以现在

注册成功!

注册

图 6.5 用户名重复

图 6.6 注册成功

主界面如图 6.7 所示。得益于上一章对 Cookie 和 Session 的分析，观察该页面的右上角，可以发现右上角给出了当前用户的用户名。看到用户名右边向下的箭头可以到该处为一个下拉菜单，当我们把鼠标停留在其上方时，弹出的菜单如图 6.8 所示。

若用户点击了“Logout”按钮，则浏览器右上角显示的信息变为“登录/注册”。如图 6.9 所示。此时该按钮的功能为一个连接，用户点击该链接页面就会跳转到上一节所介绍的登录/注册界面。

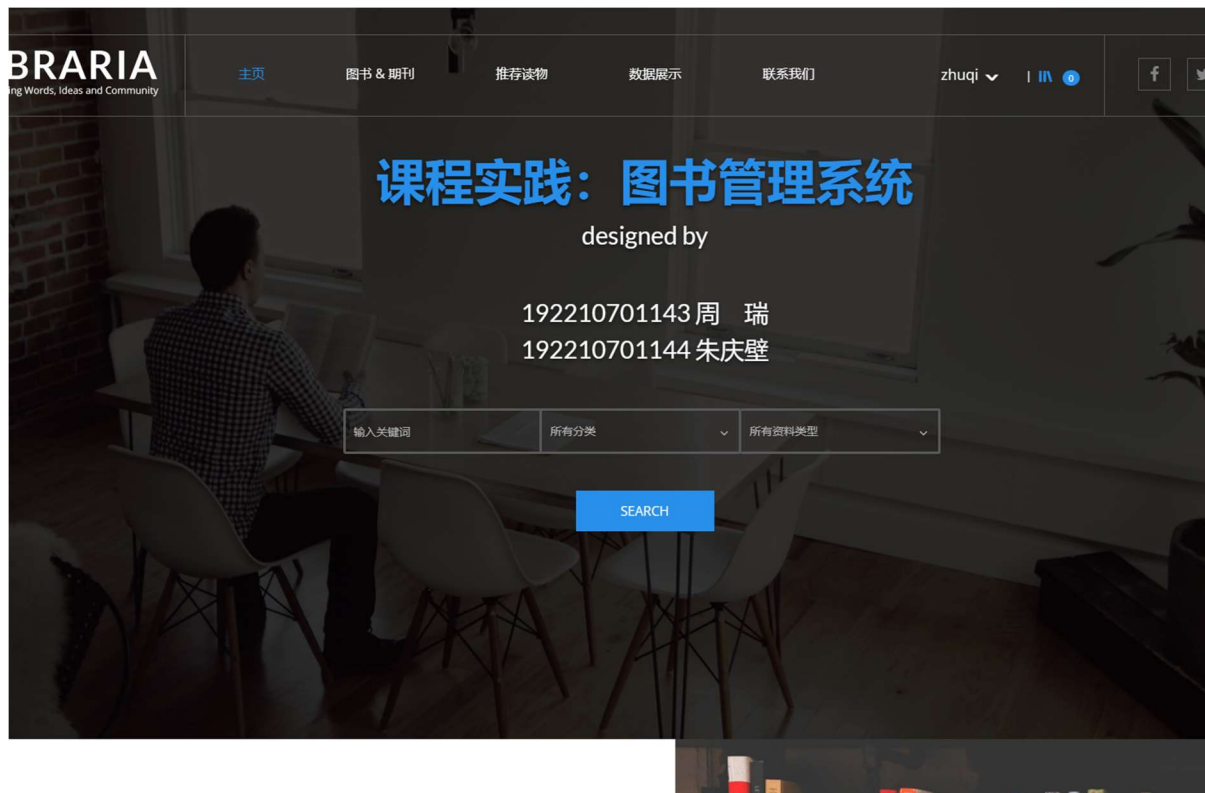


图 6.7 登录跳转页面

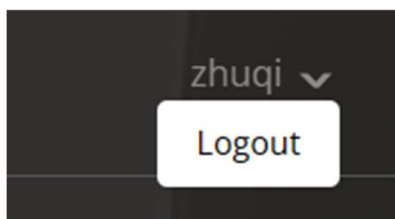


图 6.8 Logout 按钮

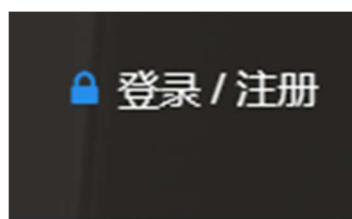


图 6.9 退出后展示信息

6.2.2 搜索功能

主页正中央有三个搜索框和一个提交按钮，该模块用于完成用户的书籍搜索功能，用户可以在左边第一个搜索框内输入关键词，可以是作者的关键词也可以是书名的关键词，同时该搜索功能还支持模糊查询，用户不需要记得作者或书籍的全名也能够搜索出自己想要的书籍。右边两个为选项框，当用户分别点击两个选项框后浏览器给出的提示分别如图 6.10 和图 6.11 所示。



图 6.10 图书分类

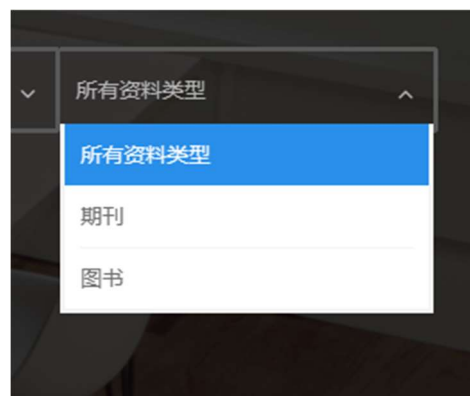


图 6.11 资料类型

用户可以在其中选择自己想要的分类或类型，这样服务器就会对返回的书籍进行筛选。当用户点击“SEARCH”按钮后，浏览器会自动跳转到搜索界面并展示出相应的搜索结果。

6.3 搜索界面

搜索界面如图 6.12 所示。该页面列出了刚刚在主页搜索的返回结果。在该界面可以看到相应书籍的书名、作者、描述和剩余量信息。右边的“借书”和“还书”按钮可以实现用户借书和还书的功能。

同时，列表左上角的搜索模块还支持动态查询，用户不需要对所有选项框进行选择也可以不在输入框中输入关键词也能实现搜索功能。若用户什么都不输入便点击“搜索”按钮，则浏览器返回的是图书馆中所有书籍的列表。

另外关键词输入框还支持模糊查询。样例演示如图 6.13 所示。可以看到在只输入了一个“迅”字的情况下，浏览器便返回了书名或者作者名中有“迅”字的书籍。

主页		你好, zhuqi		
输入关键词		所有分类	所有类型	搜索
书名	作者	描述	数量	操作
西游记	吴承恩	《西游记》是中国神魔小说的经典之作，达...	12	借书 还书
狂人日记	鲁迅	《狂人日记》是鲁迅创作的第一篇短篇白话...	34	借书 还书
阿Q正传	鲁迅	《阿Q正传》是鲁迅创作的中篇小说，创作...	34	借书 还书
三体	刘慈欣	《三体》是刘慈欣创作的长篇科幻小说系列...	24	借书 还书
红楼梦	曹雪芹	《红楼梦》，原名《石头记》，中国古代章...	36	借书 还书
水浒传	施耐庵	《水浒传》是一部长篇英雄传奇，是中国古...	12	借书 还书
三国演义	罗贯中	《三国演义》（全名为《三国志通俗演义》...	32	借书 还书
呐喊	鲁迅	《呐喊》是现代文学家鲁迅的短篇小说集，...	8	借书 还书

图 6.12 搜索页面

迅		所有分类	所有类型	搜索
书名	作者	描述	数量	操作
狂人日记	鲁迅	《狂人日记》是鲁迅创作的第一篇短篇白话...	34	借书 还书
阿Q正传	鲁迅	《阿Q正传》是鲁迅创作的中篇小说，创作...	34	借书 还书
呐喊	鲁迅	《呐喊》是现代文学家鲁迅的短篇小说集，...	8	借书 还书

图 6.13 模糊词汇搜索

6.4 管理员管理书籍界面

6.4.1 书籍添加功能

首先我们先观察一下用户数据表，如图 6.14 所示。可以看到该表除了存储用户的用户名和密码外还存储了用户的身份信息。而管理员管理书籍的界面是只有身份是“管理员”的用户才能进入的。管理员能在该页面内进行书籍的添加和删除操作。该界面与搜索界面类似，不同的地方在与浏览器提供的操作类型。如管理员项进行“添加书籍”的操作则浏览器会展示出一个弹窗如图 6.15 所示。

```
mysql> show columns from user;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
username	varchar(32)	NO		NULL	
password	varchar(32)	NO		NULL	
identity	varchar(32)	NO		NULL	

4 rows in set (0.00 sec)

图 6.14 用户数据表



图 6.15 添加书籍

从图中可以看到弹窗左上角会提示管理员当前正在进行的操作。管理员可以在输入框内输入书籍的相关信息。当管理员输入的信息不符合规范时，浏览器也会给出相应的提示，比如作者名不符合规范的提示如图 6.16 所示。



图 6.16 空缺信息提醒

最后当管理员输入完信息并提交按钮浏览器也会给出对应的提示信息，分别如图 6.17 和图 6.18 所示。



图 6.17 重复信息提醒

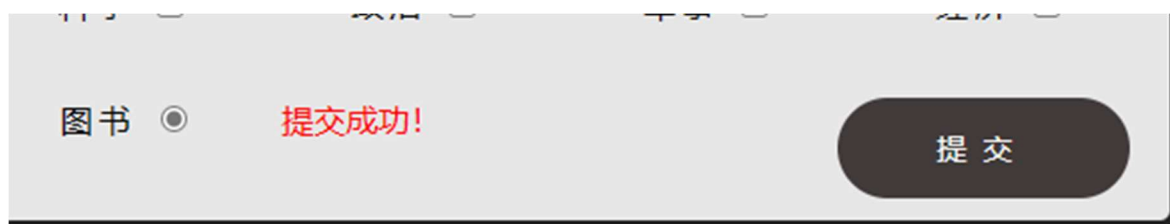


图 6.18 图书信息提交

6.4.2 书籍修改功能

点击对应书籍的修改按钮，浏览器同样会有一个对应书籍信息的弹窗，如图 6.19 所示。可以在左上角看到管理员当前正在进行的操作为修改数据，并且输入框内以自动填充好了对应书籍的相关信息以便管理员进行修改

修改数据

书名

三国演义

作者

罗贯中

描述

《三国演义》（全名为《三国志通俗演义》，又称《三国志演义》）是元末明初小说家罗贯中根据陈寿《三国志》和裴松之注解以及民间三国故事传说经过艺术加工创作而成的长篇章回体历史演义小说，与《西游记》《水浒传》《红楼梦》并称为中国古典四大名著。

数量

32

分类：

哲学 ☐

科学 ☐

政治 ☐

军事 ☐

经济 ☐

类型：

期刊 ☐

图书 ☒

提交

图 6.19 数据信息修改

第七章 结语

本系统是使用 java 技术实现的，对于自身而言很是困难，所以在这个过程中基本处于一边学习一边设计的情况，通过观看各种 java、web 前端开发视频来进行学习并设计。尽管如此还是有许多不足，除了程序本身还是有很多不足，自身的基础知识也不是很牢固。程序自身还有很多繁琐并且重复的地方，由于编写前期受到视频的影响将前端代码写到的后端，采用了输出流的方式，使得程序异常繁琐，导致添加功能时期的进度十分缓慢，对自己造成了很大的困难，走了很多弯路。经过这段时间的努力，对 java 编程、web 前端开发和 mysql 数据库有了更深的理解，对自己以后的学习工作道路大有裨益。

通过开发这个设备管理系统学到了很多 java 全栈知识，例如 springboot 框架、git 分支、数据库、前端等等知识点，使我进步了许多，对后端开发有了一个全新的认识，主要是把基础设施代码和业务代码尽可能的分开，各自不要干扰，而且能把 BEAN 都统一到 spring container 里面去，这样，bean 的生老病死都由 spring 来管理，开发者就只需要关心业务怎么实现就好了，别一会实现功能，中间还要来段事务处理，后面还要加个数据库错误处理啥的。总而言之，spring 解决的问题就是尽可能的业务代码归业务代码，基础设施代码（日志、事务，异常，对外接口.....）归基础设施代码，搞定解耦的问题，希望在以后的学习生涯中可以了解更高效的技术，从浅入深，环环相扣，每一步都会对照着官方文档结合自己的见解进行讲解，同时也会编码实现，理论与实践相结合。

参 考 文 献

- [1] WOLF H. Austrian Library Network and Next Generation Library System: Alma [J]. Bibliothek Forschung und Praxis, 2016, 9(10): 341-346.
- [2] YAHIA Z.M. Library expense control: a system dynamics approach [J]. TheElectronic Library, 2012, 9(10): 492-506.
- [3] JOY K. How to work with Sessions in ASP.Net [J]. Info World.com, 2016, 9(10):123-126.
- [4] DING D., RICH C. Building a SQL Server Inventory Report [J]. SQL Server Magazine, 2010, 5(10): 35-39.
- [5] ADAM F., STEVEN S. Pro ASP.NET MVC 3 Framework [M]. 3rd ed. Apress, Beijing: The People's Posts And Telecommunications Press, 2013.