
目 录

摘 要	I
Abstract	II
第 1 章 绪论	1
1.1 项目意义	1
1.2 国内外图书漂流的发展现状	1
1.3 本文主要研究内容	2
1.4 文章组织结构	2
第二章 系统分析	3
2.1 问题定义	3
2.2 可行性分析	3
2.3 需求分析	4
2.4 非功能需求	16
第三章 系统设计	18
3.1 概要设计	18
3.2 功能设计	18
3.3 数据库设计	20
3.4 类的设计	23
3.5 原型设计	31
第四章 系统实现与测试	37
4.1 系统运行环境	37
4.2 系统层次结构	37
4.3 具体功能实现	38
4.4 测试	41
总结	48
致谢	49
参考文献	50

摘 要

本系统是图书漂流系统，用户可以将闲置书籍上传，并分享给其他想阅读这些闲置书籍的用户。过去用户想要阅读一本图书，需要将其购买下来，但很有可能看过一遍之后就不再看了，而有阅读这本书籍需求的人又要再去购买这本图书，导致了资源的浪费。

然而，线下的图书漂流活动面临着举办时间短并且举办时间不稳定、地点时常变动、漂流的书籍数量和种类不多不全面等问题，导致了用户并没有很好的关注到这类活动。现在，本图书漂流系统建立在线上，解决了线下活动面临的问题，通过用户自主上传无用书籍，渐渐积攒书籍数量，并且线上的方式可以让用户足不出户就得到图书。

本系统前端将采用 Vue3.0 框架、利用 TS 技术和 Element-plus 组件库，后端采用 Spring Boot 框架及其他衍生框架、MySQL5.7 和 Redis 非关系型数据库的技术实现，保证了用户数据和网站数据的持久化和安全性。本系统前端的页面布局符合大众操作习惯，各功能部件摆放合理。后端业务逻辑清晰合理，数据安全性也有保障。本文详细说明了本系统从最初的需求分析到详细设计，以及最后的测试和部署。

关键词：图书漂流；数据持久化；前后端分离；Spring Boot；

Abstract

This system is a Book drifting system. Users can upload idle books and share them with other users who want to read these idle books. In the past, when users wanted to read a book, they needed to buy it, but they were likely to stop reading it after reading it once, and those who needed to read the book had to buy the book again, resulting in a waste of resources. Therefore, book drifting system is an activity of reading relay, wisdom collision and resource sharing. It is a way of recycling information resources. More importantly, book drifting can not only make a large number of temporarily idle books flow effectively, but also make people get the opportunity to read nearby. At the same time, it also makes many readers get a pleasant experience in reading sharing and information exchange.

However, offline Book rafting activities are faced with problems such as unsustainable holding, frequent changes in location, few and incomplete number and types of books, resulting in users' not paying good attention to such activities. Now, the book drifting system is established online, which solves the problems faced by offline activities. Through users' independent uploading of useless books, the number of books is gradually accumulated, and the online way allows users to get books without leaving home.

Vue framework, TS and element plus, and the back end adopts the technologies of spring boot framework, mybatis, MySQL and redis to ensure the persistence and security of user data and website data. The page layout of the front end of the system conforms to the public's operating habits, and the functional components are placed reasonably. The back-end business logic is clear and the data security is high.

Keywords: Bookcrossing; Data persistence; front end and back end; SpringBoot

第1章 绪论

1.1 项目意义

图书漂流这种阅读方式不仅很好的诠释了自由接力，以及每个阅读者的思想碰撞。而且正是这种共享资源的方式，在信息快速更迭的时代，人们已经成功做到了循环利用。

这种模式最初起源于西欧，对于 20 世纪中期喜爱阅读书籍的人们来说，已经不局限于图书馆内获得阅读机会。图书漂流提供了一种更为便捷提效的方式，人们甚至可以将家中尘封很久未翻阅的图书重新利用起来^[1]。

数据在社会面流转起来，让书籍遇到属于他的读者，毕竟每一本书的生命不该落灰。漂流模式的兴起真正做到了信息资源的互通和不浪费。

图书漂流是一段文明的传播，书香的四溢。传递至今，已经不仅限于投放户外，越来越多的图书漂流客正在发挥他们烂漫的想象力，在投放说明添加了自己的规则，打破了时间和空间的界限，真正实现了图书^[2]。

漂流的目的：心灵因交流而美丽。

1.2 国内外图书漂流的发展现状

1.2.1 国外图书漂流的发展现状

漂流书籍可以追溯到 20 世纪 60 年代的欧洲。这是一本书，由书商所有，但不再在提供给它标签上阅读。它将被放置在公共场所，并免费提供给阅读的人^[3]。无论谁找到了这本书，现在都可以拥有它。读完这本书后，你必须按照标签上的说明把它放在公共环境中。这代表了上个世纪的传统驾驶活动。

进入 21 世纪以来，现代信息技术和互联网的飞速发展，推动了漂流网站的快速发展和漂流活动的普及。2001 年 4 月，美国的罗恩·霍恩贝克（Ron Hornbeck）发布了世界上第一个专门研究书的漂移的网站：<http://bookcrossing.com> 网站设计由强调网站设计的用户手册确定^[4]。此后，虚拟网络领域的漂流图书业务迅速发展，并在欧美国家广泛传播。到 2020 年，全世界将有 200 万会员，登记的漂移量将超过 1000 万。

1.2.2 国内图书漂流的发展现状

在全球一体化和经济全球化的形势下，漂流图书作为一种新的阅读方式和社会活动，迅速扩展到中国内地，越来越多的读者受到它的欢迎。特别是从 2003 年开始，中国开展了

漂流活动。一位名叫吴刚的中国人在全国媒体上开设了一个受西方社会欢迎的网站“书漂”，引起了国内读者的关注^[5]。中国流网的灵感来源于创始人劳夫和简在瑞士之行中收集的第一批流网书籍。2006 年回国后，他们建立了第一个中国的图书漂流网。近年来，中国漂网图书网站发展迅速。

1.3 本文主要研究内容

本文的开发的系统主要是基于 web 的图书漂流系统的设计与实现，该系统主要采用 Vue、Spring Boot、MySQL、Redis、Element、Mybatis、Axios 等技术实现页面数据展示和交互、前后端数据传输、数据持久化和可视化以及后台对各种数据的管理，使得系统对图书漂流的管理更加高效并降低成本^[6]。

1.4 文章组织结构

该文章共分四个章节，如下：

第一章：绪论

第二章：系统分析

第三章：系统设计

第四章：系统实现与测试

第二章 系统分析

2.1 问题定义

现代已经是互联网的时代了，所有基于数据的管理都应该使用计算机进行管理。现在利用计算机和管理系统进行数据的存储和管理已经是非常普遍的方式。这种方式不仅工作效率高，安全性高，而且出错率非常的低。

再进行大数据量的处理和管理中，传统的人工管理方式成本太高，而且十分容易出错。这时，计算机就能充分发挥他的优势。图书漂流涉及到的图书量和用户量都十分庞大，如果采用人工管理的方式，随着漂流记录的增多，就会变得十分混乱。而且查询很久之前的记录也十分不方便，有时还会查找不到造成数据丢失的问题。同时资料的堆放管理也是一个问题^[7]。

所以基于以上种种问题，搭建一个图书漂流系统就变得刻不容缓。通过计算机对图书漂流的管理，实现图书漂流的可视化、系统化、规划化。并对图书和漂流者统一的管理。如果出现问题可以更好的追根溯源，并实时的对用户权限进行限制。

图书漂流者可以直接登录系统中查看各个图书的漂流情况，如果漂流者之前注册过账号，可以登录系统并开始漂流任意还未被漂流的书籍。用户可以在列表中搜索自己所喜欢的名著。

系统后台的安全性和保密性要求就相对更高，因为管理员的权限很大，可以对漂流信息和图书信息做出修改，还可以对用户权限做出限制并对那些对系统造成恶劣影响的用户进行直接封禁账号的处理。

2.2 可行性分析

2.2.1 市场可行性

国内暂时还没有较大的线上图书漂流系统，现在年轻人普遍具有浏览互联网的习惯，也有许多年轻人喜欢阅读电子书。预计用户量较大，阅读需求充足有保障。建立图书漂流系统具备市场可行性。

2.2.2 技术可行性

目前国内网站搭建技术也比较成熟，所以通过 Spring Boot 跟 Vue 搭建网站没有技术难度。网站管理操作也十分简单，网站管理人员经过短时间培训即可管理网站，并且该网

站许多功能都是已用户为主导的，管理员只在其中发挥次要作用，不需要对漂流业务有过多的干预^[8]。建立图书漂流网站具备技术可行性。

2.2.3 政策可行性

阅读活动向来都是国家大力支持的有益身心发展的活动。全国各地都会举办一些线下的图书漂流活动。近些年，国家也出台了多项关于图书漂流活动的扶持政策^[9]。鼓励青少年以及其他年龄段人参与到阅读中来。建立图书漂流网站具备政策可行性。

2.2.4 模式可行性

现在的生活模式非常快，人们很难有大块的时间去阅读，因此很难会有一大块的时间专门拿出来花在阅读上，所以人们购买新书的欲望也会降低^[10]。这时就需要一个图书漂流的方式，让人们获取新书的途径更加低成本。建立图书漂流系统具备模式可行性。

2.3 需求分析

2.3.1 功能需求

本系统主要有两大角色，一个是普通用户，能够进入首页查看公告和新闻还有浏览图书信息页面，也能查看图书信息；能够进入用户的个人中心进行更新个人信息，维护之前上传的图书信息以及处理其他用户的漂流请求等等。

另一个角色是系统管理员角色。管理员主要是对系统中的所有信息进行维护和增删改查，还可以添加和更新系统的新闻和公告，还可以对用户进行管理。

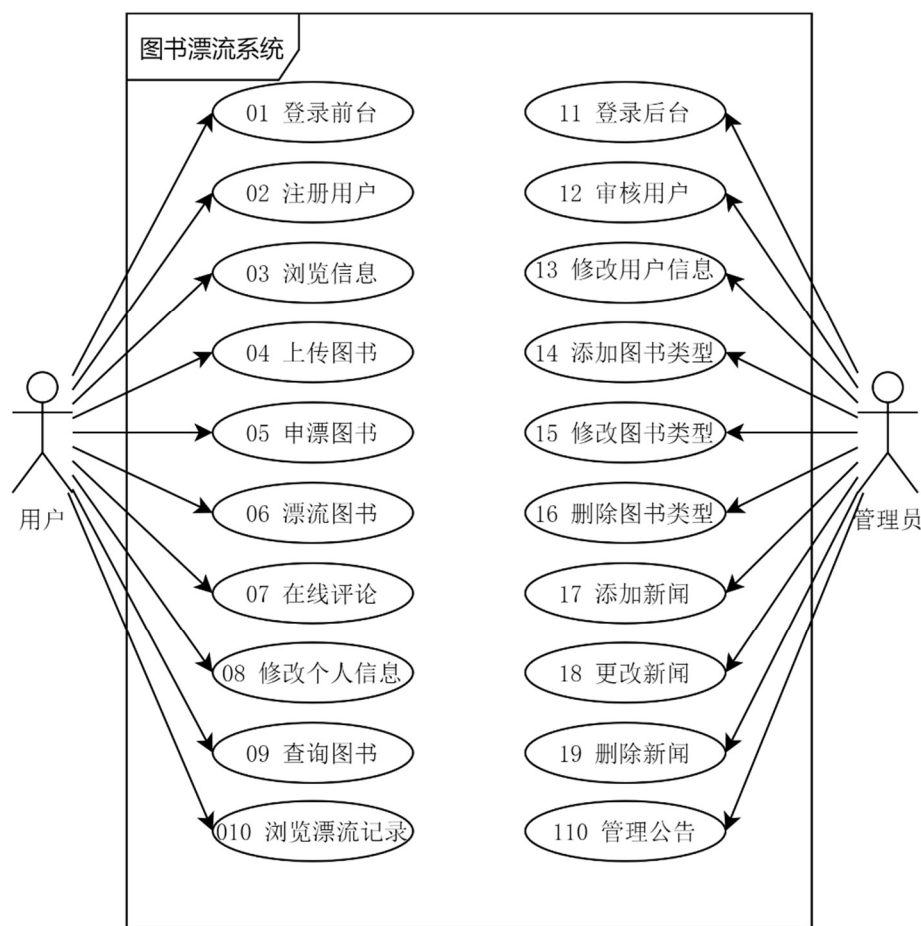


图 2.1 系统用例图

本系统主要功能都集中在用户上，因为图书漂流是用户自发的一种行为，管理员只管理用户、网站信息和图书类别信息。

对于用户角色：该角色主要包含了浏览信息、上传图书、申请漂流图书、修改个人资料、在线评论、查询图书、查询图书漂流记录等功能。为用户提供了一个可以对图书进行漂流和评价以及对自己信息进行管理的一个平台，界面简洁，有良好的交互能力，用户可以按照自己的需求对图书进行查找，同时，如果该图书在可漂流状态，可以对该图书进行漂流。还可以对图书进行评价，方便给其他用户对该图书有一个直观的感受，并帮助其他用户决定是否漂流该图书。

对于管理员角色：该角色主要包含了对网站信息的管理，如新闻管理、公告管理。还可以对用户上传的图书类型进行增加、删除、修改。对系统用户进行审核，用户个人信息的查看和修改。

2.3.2 用户功能

用户功能架构图如下：

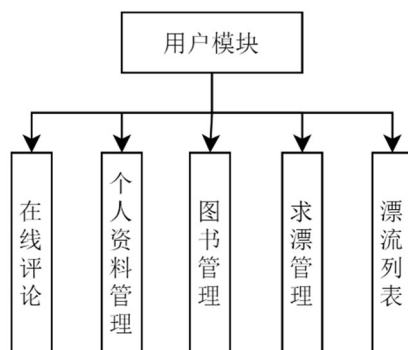


图 2.2 用户功能架构图

前台登录是用户进入用户系统的一个入口，用户可以访问本系统的登录页面，并且输入账号密码，再点击登录按钮进行登录。

表 2.1 前台登录

用例名称	登录前台	标识号	01
参与者	用户		
用例描述	用户可在前台进行登录		
用例需求	邮箱名唯一		
前置条件	用户打开浏览器进入登录界面		
后置条件	无		
基本操作流程	1. 用户点击登录进入登录界面； 2. 用户输入账号； 3. 用户输入密码； 4. 用户点击登录；		
异常处理流程	2.1 输入用户名不存在，要求重新输入； 3.1 输入密码不正确，要求重新输入；		

注册用户是游客成为系统用户的一个途径，让游客通过审核后，成为系统用户，进而对本系统的用户功能进行操作。

表 2.2 注册用户

用例名称	注册用户	标识号	02
参与者	用户		

用例描述	用户可在注册页面进行注册
用例需求	1. 用户名和邮箱名唯一； 2. 登录使用邮箱； 3. 密码非明文显示；
前置条件	用户打开浏览器进入欢迎界面
后置条件	无
基本操作流程	1. 用户点击登录进入登录界面； 2. 用户输入用户名； 3. 用户输入邮箱号 4. 用户输入密码； 5. 用户点击注册；
异常处理流程	2.1 输入用户名重复，要求重新输入； 3.1 输入用户名重复，要求重新输入；

用户登录后，可以浏览网站信息。主要是浏览新闻信息、公告信息和用户上传的图书信息。用户登录或者游客进入后首先到达首页页面，首页包含网站的一些简介和站内链接。

表 2.3 浏览信息

用例名称	浏览信息	标识号	03
参与者	用户		
用例描述	用户可在前台浏览现有图书和网站信息		
用例需求			
前置条件	用户打开浏览器进入浏览信息界面		
后置条件	无		
基本操作流程	1. 用户点击浏览界面； 2. 界面显示图书信息和网站信息；		
异常处理流程	无		

用户登录后，进入个人中心就可以选择上传图书，可以将手头上闲置的图书进行上传，供其他有需求的用户申请漂流。图书漂流系统的大部分图书都是基于用户的上传得到的。

表 2.4 上传图书用例

用例名称	上传图书	标识号	04
------	------	-----	----

参与者	用户
用例描述	用户可在个人页面上上传图书
用例需求	每种图书编号唯一
前置条件	用户进入系统完成登录并进入个人页面
后置条件	无
基本操作流程	1. 用户点击上传图书; 2. 用户输入图书编号; 3. 用户输入图书名; 4. 用户输入图书作者; 5. 用户选择图书类型; 6. 用户点击上传;
异常处理流程	无

用户登录后,可以浏览图书信息,只要图书在可漂流的状态,就可以对自己想要漂流的图书进行申请漂流。申请漂流后需要拥有图书的用户对该漂流请求进行同意。同意过后即可获得该图书,拒绝后也可以再次进行申请。

表 2.5 申漂图书用例

用例名称	申漂图书	标识号	05
参与者	用户		
用例描述	用户可在图书信息页面申请漂流该图书		
用例需求	用户进入系统完成登录并有申漂权限		
前置条件	用户进入图书信息页面并未申请漂流该图书		
后置条件	拥有该图书的用户收到漂流申请		
基本操作流程	1. 用户点击申漂图书; 2. 系统生成漂流信息;		
异常处理流程	2.1 图书已经被其他用户漂流;		

当用户拥有图书或者上传图书后,就会有对这些书心仪的用户,这些用户会发起对这些图书的申请漂流请求,用户可以在个人中心对这些请求进行同意或者拒绝。如果同意,那么就会更改图书的拥有者包括自己上传的图书;如果拒绝,图书的拥有者就不会更改,就可以继续持有该书。

表 2.6 漂流图书用例

用例名称	漂流图书	标识号	06
参与者	用户		

用例描述	用户可在个人页面的求漂申请进行同意，即漂流该图书
用例需求	用户进入系统完成登录并有漂流请求
前置条件	用户进入个人页面并且有可同意的求漂申请
后置条件	改变图书的拥有者
基本操作流程	1. 用户选择一个求漂信息； 2. 用户对求漂申请进行同意； 3. 系统修改漂流信息状态； 4. 系统修改图书信息；
异常处理流程	无

用户在登录后，可以对图书发表评论。用户进入图书详情页面，在下方的评论框中输入内容，再点击下面的评论按钮，即可发布评论。

表 2.7 在线评论用例

用例名称	在线评论	标识号	07
参与者	用户		
用例描述	用户可在图书的详情页面下方,对图书的内容和完整度等等,进行一个全方位的评价。		
用例需求	用户进入系统完成登录并有评论权限		
前置条件	用户进入图书详情页面		
后置条件	无		
基本操作流程	1. 用户点击详情页面下方的评论框； 2. 用户在输入框中输入评论信息； 3. 用户点击评论按钮； 4. 系统对评价经行处理；		
异常处理流程	无		

用户登录后，可以进入个人中心修改之前注册时填写的个人信息。

表 2.8 修改个人信息用例

用例名称	修改个人信息	标识号	08
参与者	用户		
用例描述	用户可在个人页面中，对用户的个人信息进行修改		
用例需求	用户进入系统完成登录		
前置条件	用户进入个人页面		
后置条件	无		

基本操作流程	<ol style="list-style-type: none"> 1. 用户点击个人页面中的修改个人信息按钮; 2. 用户输入要修改的邮箱号; 3. 用户输入要修改的用户名; 4. 用户输入要修改的手机号; 5. 用户点击完成按钮;
异常处理流程	<ol style="list-style-type: none"> 2.2 邮箱号为空或已存在, 请重新输入; 2.3 用户名为空或已存在, 请重新输入; 2.4 手机号已存在, 请重新输入;

用户可以在图书列表中查询符合自己需求的图书。

表 2.9 查询图书用例

用例名称	查询图书	标识号	09
参与者	用户		
用例描述	用户可在图书列表中输入想要查找的图书信息, 对图书进查找		
用例需求	用户进入系统完成登录		
前置条件	用户进入浏览图书列表页面		
后置条件	无		
基本操作流程	<ol style="list-style-type: none"> 1. 用户选中图书列表的搜索框; 2. 用户输入要查询的图书信息; 3. 用户点击搜索按钮; 4. 系统显示符合查询条件的图书; 		
异常处理流程	2.1 输入图书编号重复, 要求重新输入;		

用户登录系统后, 进入个人中心, 就可以查看自己的历史漂流记录, 什么时候借的什么书, 从哪里借的, 都可以在记录中浏览。

表 2.10 浏览漂流记录用例

用例名称	浏览漂流记录	标识号	010
参与者	用户		
用例描述	用户可在个人页面中查看自己的漂流记录。		
用例需求	用户进入系统完成登录		
前置条件	用户进入个人页面		
后置条件	无		
基本操作流程	<ol style="list-style-type: none"> 1. 用户进入个人页面; 2. 用户查看浏览漂流记录; 		

	3. 系统显示漂流记录;
异常处理流程	2.3 用户没有漂流记录就显示空;

2.3.3 管理员功能

管理员功能架构图如下:

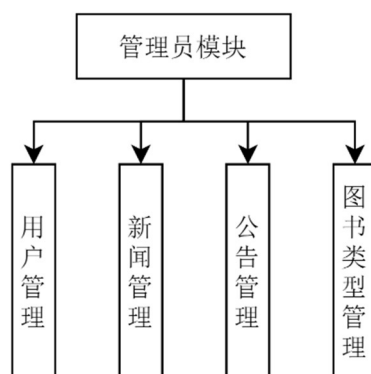


图 2.3 管理员功能架构图

登录后台是管理员进入后台管理系统的基本途径也是唯一途径。管理员通过登录进入到管理系统中，从而对整个网站的基本信息进行修改，包括新闻和公告的发布等。

表 2.11 登录后台用例

用例名称	登录后台	标识号	11
参与者	管理员		
用例描述	管理员可以通过输入账号密码进入后台管理系统		
用例需求	无		
前置条件	管理员进入登录界面		
后置条件	无		
基本操作流程	1. 管理员输入账号; 2. 管理员输入密码; 3. 管理员点击登录按钮; 4. 系统校验账号密码; 5. 系统跳转至后台管理系统;		
异常处理流程	2.1 账号尚未注册或该账号为前台用户账号, 请重新输入; 2.2 密码不正确, 请重新输入;		

管理员可以对注册的用户进行审核, 默认为待审核状态, 可以正常的登录进系统, 进行各种操作, 审核通过后, 将不再显示在带审核列表中, 如果审核不通过则无法正常登录

进系统，用户登录将显示账号异常。

表 2.12 审核用户用例

用例名称	审核用户	标识号	12
参与者	管理员		
用例描述	管理员可以对用户的注册经行审核，审核不通过的用户不允许登录。		
用例需求	管理员已经登录		
前置条件	管理员进入用户管理界面		
后置条件	无		
基本操作流程	1. 管理员进入用户审核页面； 2. 管理员选择需要审核的用户； 3. 系统显示用户信息； 4. 管理员选择审核结果；		
异常处理流程	无		

管理员可以对系统中的用户的个人信息进行修改，如果发现非法用户名，还可以修改用户的审核状态，让用户的审核状态变为审核不通过状态，从而禁止用户进入系统。

表 2.13 修改用户信息用例

用例名称	修改用户信息	标识号	13
参与者	管理员		
用例描述	管理员可以修改用户的个人信息		
用例需求	管理员已经登录		
前置条件	管理员进入用户管理界面		
后置条件	无		
基本操作流程	1. 管理员选择需要修改的用户 2. 管理员点击用户右边的编辑按钮 3. 管理员输入要修改的用户的邮箱号； 4. 管理员输入要修改的用户的用户名； 5. 管理员输入要修改的用户的手机号； 6. 管理员点击完成；		
异常处理流程	2.3 邮箱号为空或已存在，请重新输入； 2.4 用户名为空或已存在，请重新输入； 2.5 手机号已存在，请重新输入；		

管理员可以添加图书类型，供用户上传图书时，对上传的图书进行分类。默认有一个

基础类型。防止没有图书类型，用户上传图书的过程没有图书类别进行选择，导致上传时图书的类型为空。

表 2.14 添加图书类型用例

用例名称	添加图书类型	标识号	14
参与者	管理员		
用例描述	管理员可以添加图书类型		
用例需求	管理员已经登录		
前置条件	管理员进入图书类型管理界面		
后置条件	无		
基本操作流程	1. 管理员点击添加图书类型； 2. 管理员输入图书类型名称； 3. 管理员点击添加按钮；		
异常处理流程	2.2 图书名称已存在，请重新输入；		

管理员可以对添加的图书类别进行修改。防止管理员在第一次创建时，输入了错误的图书类别名称而无法修改的错误，或者需要临时修改图书类别名称时，有可以更改名称的能力。

表 2.15 修改图书类型用例

用例名称	修改图书类型	标识号	15
参与者	管理员		
用例描述	管理员可以修改图书类型		
用例需求	管理员已经登录		
前置条件	管理员进入图书类型管理界面		
后置条件	无		
基本操作流程	1. 管理员选择要修改的图书类型； 2. 管理员点击编辑按钮； 3. 管理员输入新的图书类型名称； 4. 管理员点击确定；		
异常处理流程	2.3 图书名称已存在，请重新输入；		

管理员可以对添加的图书类别进行删除。管理员可以对过时的不需要的类型进行删除。删除时会再进行一次确认操作，防止误删。删除后，本系统无法对删除了的图书类型进行找回操作。

表 2.16 删除图书类型用例

用例名称	删除图书类型	标识号	16
参与者	管理员		
用例描述	管理员可以删除图书类型		
用例需求	管理员已经登录		
前置条件	管理员进入图书类型管理界面		
后置条件	从属于删除类型下的图书的类型变为基本类型		
基本操作流程	1. 管理员选择要删除的图书类型； 2. 管理员点击删除按钮； 3. 系统显示要删除的图书类型名称； 4. 管理员点击确定；		
异常处理流程	无		

管理员可以添加本系统内的新闻。

表 2.17 添加新闻用例

用例名称	添加图书类型	标识号	17
参与者	管理员		
用例描述	管理员可以添加站内新闻		
用例需求	管理员已经登录		
前置条件	管理员进入新闻管理界面		
后置条件	无		
基本操作流程	1. 管理员点击添加新闻； 2. 管理员输入新闻标题； 3. 管理员输入新闻内容； 4. 管理员点击提交按钮；		
异常处理流程	2.2 新闻标题为空，请重新输入；		

管理员可以修改本系统内的新闻。

表 2.18 修改新闻用例

用例名称	修改新闻	标识号	18
参与者	管理员		
用例描述	管理员可以修改新闻		
用例需求	管理员已经登录		
前置条件	管理员进入新闻管理界面		

后置条件	无
基本操作流程	1. 管理员选择要修改的新闻； 2. 管理员点击编辑按钮； 3. 管理员输入新的新闻标题； 4. 管理员输入新的新闻内容； 5. 管理员点击提交；
异常处理流程	2.3 新闻标题为空，请重新输入；

管理员可以删除本系统内的新闻。

表 2.19 删除新闻用例

用例名称	删除新闻	标识号	19
参与者	管理员		
用例描述	管理员可以删除新闻		
用例需求	管理员已经登录		
前置条件	管理员进入新闻管理界面		
后置条件	无		
基本操作流程	1. 管理员选择要删除的新闻； 2. 管理员点击删除按钮； 3. 系统显示要删除的新闻标题； 4. 管理员点击确定；		
异常处理流程	无		

管理员可以添加本系统内的公告。

表 2.20 添加公告用例

用例名称	添加公告	标识号	20
参与者	管理员		
用例描述	管理员可以添加站内公告		
用例需求	管理员已经登录		
前置条件	管理员进入公告管理界面		
后置条件	无		
基本操作流程	1. 管理员点击添加公告； 2. 管理员输入公告标题； 3. 管理员输入公告内容； 4. 管理员点击提交按钮；		
异常处理流程	2.2 公告标题为空，请重新输入；		

管理员可以修改本系统内的公告。

表 2.21 修改公告用例

用例名称	修改公告	标识号	21
参与者	管理员		
用例描述	管理员可以修改公告		
用例需求	管理员已经登录		
前置条件	管理员进入公告管理界面		
后置条件	无		
基本操作流程	1. 管理员选择要修改的公告； 2. 管理员点击编辑按钮； 3. 管理员输入新的公告标题； 4. 管理员输入新的公告内容； 5. 管理员点击提交；		
异常处理流程	2.3 公告标题为空，请重新输入；		

管理员可以删除本系统内的公告。

表 2.22 删除公告用例

用例名称	删除公告	标识号	22
参与者	管理员		
用例描述	管理员可以删除公告		
用例需求	管理员已经登录		
前置条件	管理员进入公告管理界面		
后置条件	无		
基本操作流程	1. 管理员选择要删除的公告； 2. 管理员点击删除按钮； 3. 系统显示要删除的公告标题； 4. 管理员点击确定；		
异常处理流程	无		

2.4 非功能需求

2.4.1 性能需求

1. 硬件描述：

Web 发布服务器：阿里云轻量级应用服务器，2 核心、2GB 内存、60GB 存储的云服务器，并安装 CentOS7.0 系统。

数据库服务器：阿里云关系型数据库 RDS，通用型 1：4，2 核心、8GB 内存、2000GB 存储的数据库服务器。

2. 性能描述：

网络带宽为 2Mbps。

3. 用户体验要求：

系统必须支持并发用户数大于 20，用户访问图书列表页面响应时间小于 1 秒，用户申请图书漂流响应时间小于 1 秒。

2.4.2 安全性

从软件的安全性分析，登录系统需要及时验证，预防用户输入错误的字符。一旦出现错误输入第一时间提示反馈给用户，确保用户能及时修改。在网页显示中不显示敏感信息，不显示数据库中的安全信息。不保留信息。

2.4.3 易用性

从这款软件的安全和易用性来分析,在使用者不经过任何训练也不必需要任何解释,就可以很容易地使用本系统。为了充分考虑网站页面的简单美观,操作方便,系统的每一个细节都进行了精心设计和处理,每个功能的具体操作和生产流程尽量地符合如今广大消费者的使用习惯。

2.4.4 可拓展性

本系统在开发时期，预留了许多将来可能会使用到的接口。如果将来需要添加新的功能，那么本系统在此基础上进行二次开发也会十分的快捷。

第三章 系统设计

3.1 概要设计

本系统主要分为三个大模块，一个是首页及浏览图书信息的前台模块，一个是用户维护个人信息和维护上传的图书信息并做漂流处理的用户模块，还有一个就是管理员模块，主要是对系统中所有信息的增删改查，还可以更新系统的新闻和公告。通过以上的分析，我们设计出本系统的功能模块图如下。

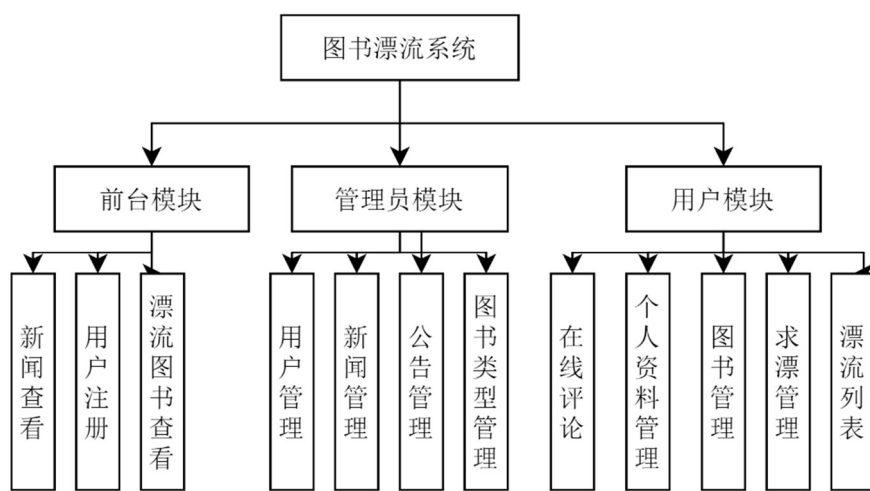


图 3.1 功能模块图

3.2 功能设计

图书漂流模块

用户在个人中心中上传图书后，系统将生成图书记录，图书记录一开始处于可出借状态，供用户来漂流。用户申请漂流可出借状态的图书后，系统生成漂流记录，此时，对申请漂流的用户，漂流的图书状态变为漂流中。拥有此图书的用户可以同意或者拒绝这次的漂流申请。如果用户拒绝漂流申请，这图书的状态将变回可出借状态，用户可再次申请漂流；如果用户同意此次漂流申请，图书的状态将变为已出借，并同时改变图书的拥有者信息。

图书漂流是由用户浏览本系统中的图书，选择用户喜欢的图书，进入图书详情页面进行发起图书漂流，前提必须这本书不属于该用户且该用户没有正在申请这本图书。然后系统发起漂流请求，拥有该图书的用户收到图书漂流请求，可以选择同意或者拒绝。如果同意该漂流请求，系统修改漂流状态为已同意并修改图书的拥有者信息。如果选择拒绝则只

修改漂流信息为已拒绝。用户可以日后重新发起漂流请求。

漂流图书整体流程如下：

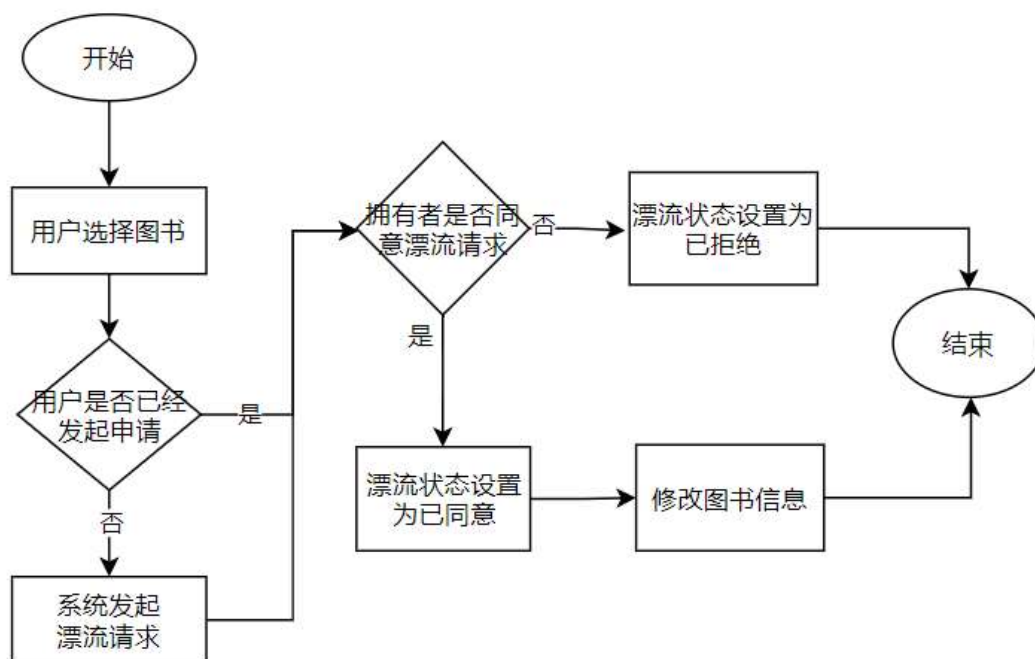


图 3.2 图书漂流流程图

图书状态图如下：

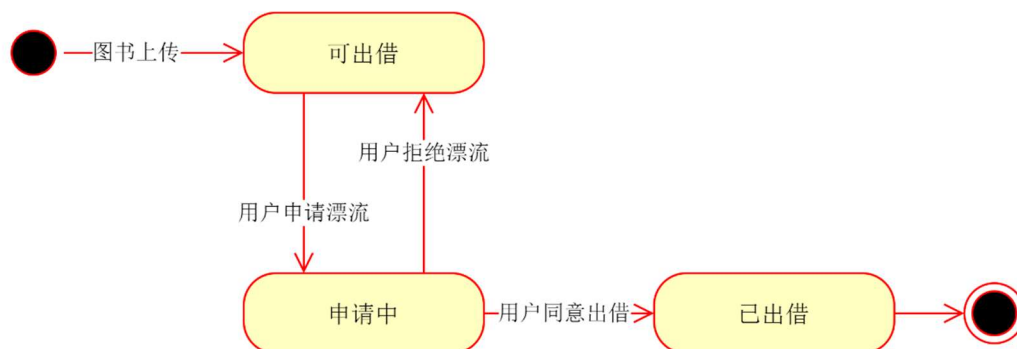


图 3.3 图书状态转换图

当用户申请漂流任一图书之后，系统将生成漂流记录，一开始漂流记录处于申请中状态，拥有该图书的用户可以同意或拒绝该申请。如果用户同意申请，漂流记录状态将转变成已完成，如果用户拒绝申请，漂流记录状态将转变成已拒绝，可以再次发起漂流申请来漂流该图书。

漂流状态图如下：

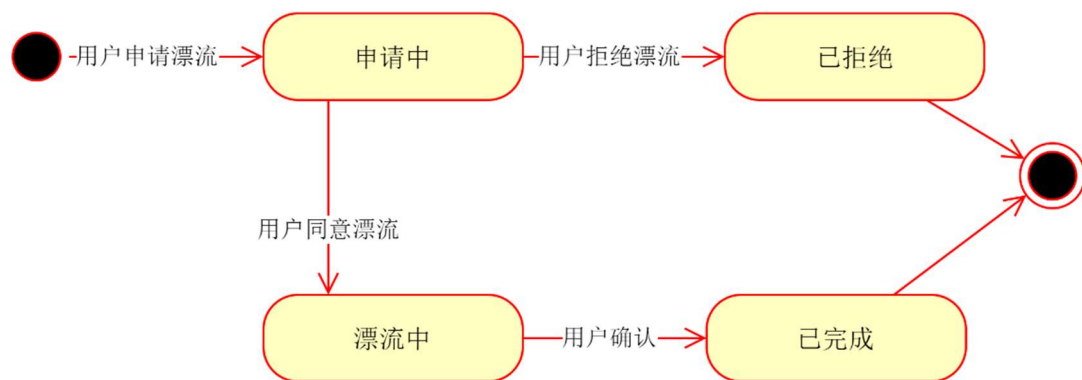


图 3.4 漂流状态图

3.3 数据库设计

数据库设计的具体 E-R 图如下图所示。

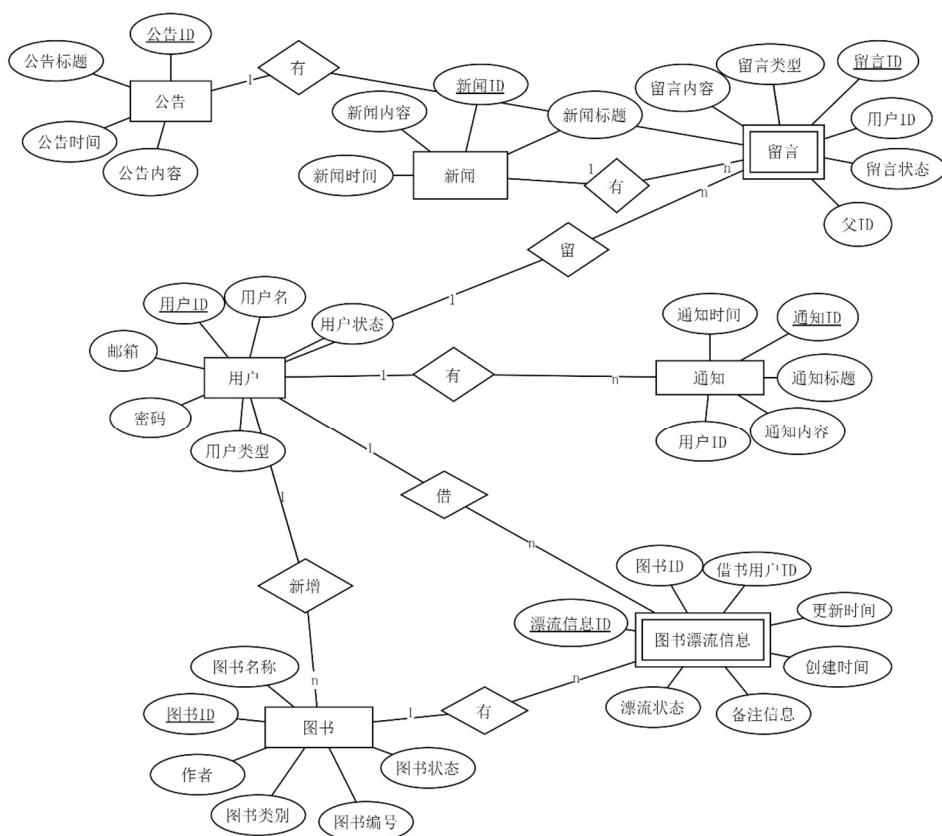


图 3.5 数据库 E-R 图

通知数据表：

存储网站的通知信息。需要存储通知中的标题以及内容还有创建时间。

名	类型	长度	不是 null	虚拟	键	注释
announcement_id	bigint	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	公告ID
title	varchar	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>		公告标题
content	varchar	255	<input type="checkbox"/>	<input type="checkbox"/>		公告内容
create_date	datetime		<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间

图 3.6 announcement 表设计图

新闻数据表：

存储网站的新闻信息。需要存储新闻中的标题以及内容还有创建时间。

名	类型	长度	小数点	不是 null	虚拟	键	注释
news_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	新闻ID
title	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		新闻标题
content	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>		新闻内容
create_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间

图 3.7 news 表设计图

图书数据表：

存储用户上传的图书信息。主要存储图书中的编号、名称、作者、类型、状态、创建日期、拥有者 ID 等。


名	类型	长度	小数点	不是 null	虚拟	键	注释
book_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	图书ID
book_no	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>		图书编号
name	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		图书名称
author	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		图书作者
introduction	text			<input type="checkbox"/>	<input type="checkbox"/>		图书简介
type	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		图书类型
status	varchar	5		<input checked="" type="checkbox"/>	<input type="checkbox"/>		图书状态 (0: 上传中
create_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间
user_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		拥有者用户ID ...

图 3.8 Book 表设计图

图书类型数据表：

存储系统中的图书类型，用于给图书分类，表中 book_type_id 与 book 表中的 type 字段有关联关系。主要存储类型的名称，创建时间等。


名	类型	长度	小数点	不是 null	虚拟	键	注释
book_type_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	图书类型主键
type_code	varchar	6		<input type="checkbox"/>	<input type="checkbox"/>		图书类型代码
type_name	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		图书类型名称
create_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间

图 3.9 book_type 表设计图

评论数据表:

存储网站中用户对图书的评价。主要存储评价中的父评论 ID、评论内容、评论类型、评论状态、创建时间、用户 ID 等。父评论 Id 为回复的评论的 ID。

名	类型	长度	小数点	不是 null	虚拟	键	注释
comment_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	留言ID
parent_id	bigint	20		<input type="checkbox"/>	<input type="checkbox"/>		上级留言ID
content	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		留言内容
type	varchar	5		<input checked="" type="checkbox"/>	<input type="checkbox"/>		留言类型 (1: 公告留言、2: 新闻留言、
status	varchar	5		<input checked="" type="checkbox"/>	<input type="checkbox"/>		留言状态 (0: 根留言、1: 子留言)
create_date	datetime			<input type="checkbox"/>	<input type="checkbox"/>		创建时间
user_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		留言用户ID

图 3.10 comment 表设计图

漂流信息数据表:

漂流信息是本系统的关键信息,由 cross_info 表来存储。表中有漂流的图书主键 book_id、备注信息 remark、漂流类型 type、漂流状态 status、创建时间 create_date、更新时间 update_date、用户主键 user_id

名	类型	长度	小数点	不是 null	虚拟	键	注释
cross_info_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	漂流ID
book_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		图书ID
remark	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>		备注信息
type	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		漂流类型
status	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		漂流状态 (0: 已拒绝、1: 申请中
create_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间
update_date	datetime			<input type="checkbox"/>	<input type="checkbox"/>		更新时间
user_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户ID

图 3.11 cross_info 表设计图

用户数据表:

用户表用来存储系统中的已注册用户。表中有用户名、密码、邮箱、手机号、用户类

型、用户状态等信息。

名	类型	长度	小数点	不是 null	虚拟	键	注释
user_id	bigint	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	用户id
username	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户名称
password	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户密码
email	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户邮箱
tel	varchar	11		<input type="checkbox"/>	<input type="checkbox"/>		用户手机号
type	varchar	5		<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户类型 (0: 管理员、1: 普通用户)
status	varchar	5		<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户状态 (0: 未审核、1: 审核中、2: 审核通过)

图 3.12 user 表设计图

3.4 类的设计

图书模块：

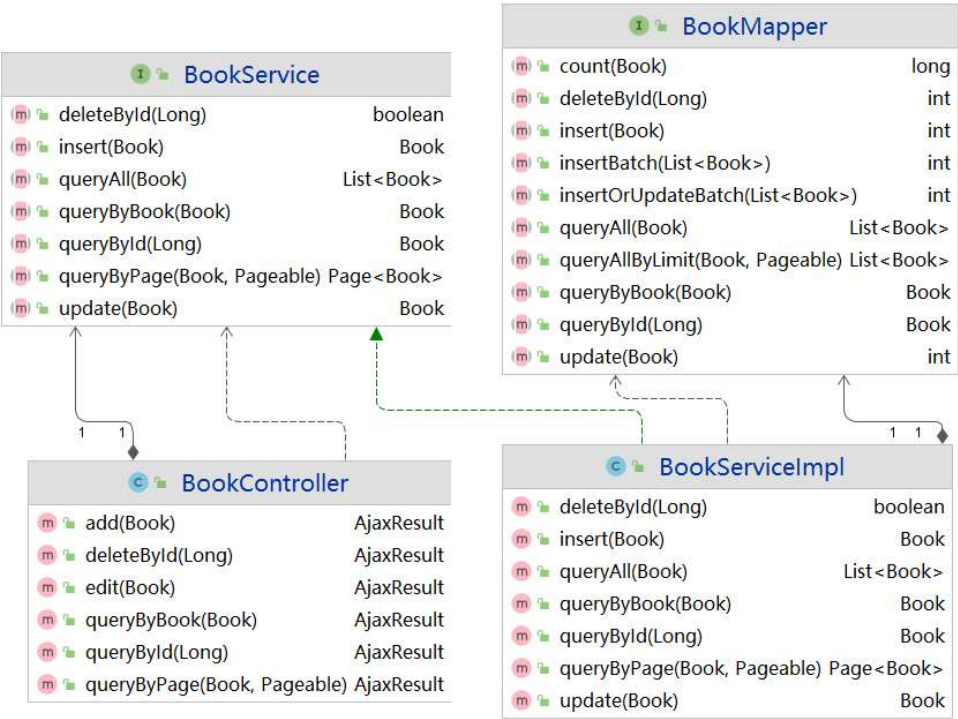


图 3.13 图书模块类图

表 3.1 BookServiceImpl 详细设计

类名	BookServiceImpl
接口	●BookService 接口：图书模块的业务类接口，有关于图书模块的所有功

	能。
属性	<ul style="list-style-type: none"> ●private BookMap bookMap 属性：Mybatis 框架中的 Dao 实现类。连接数据库存储数据。
方法	<ul style="list-style-type: none"> ●public Boolean deleteById(Long)：根据主键删除图书信息。形参传一个主键，用于 Dao 层在数据库中找到对应信息。返回值返回是否删除成功。 ●public void insert(Book)：将形参中的图书类信息，转换成数据库中数据表的信息，并存储到数据库中。返回值返回存储入数据表后的值。 ●public void queryAll(Book)：将形参中的图书类信息作为查找条件，在数据表中查找对应的信息并返回。返回值返回符合条件的图书类数组。 ●public void queryById(Long)：根据主键查找对应的数据。形参传入一个主键，用于 Dao 层在数据库中找到对应信息并返回。返回值返回一个数据对象。 ●public void queryByPage(Book, Pageable)：将形参中的图书类信息作为查找条件，在数据表中查找固定个数的信息并返回。返回值返回符合条件的图书类数组。 ●public void update(Book)：将形参中的图书类信息，更新到数据库中。返回值返回存储入数据表后的值。

通知模块设计：

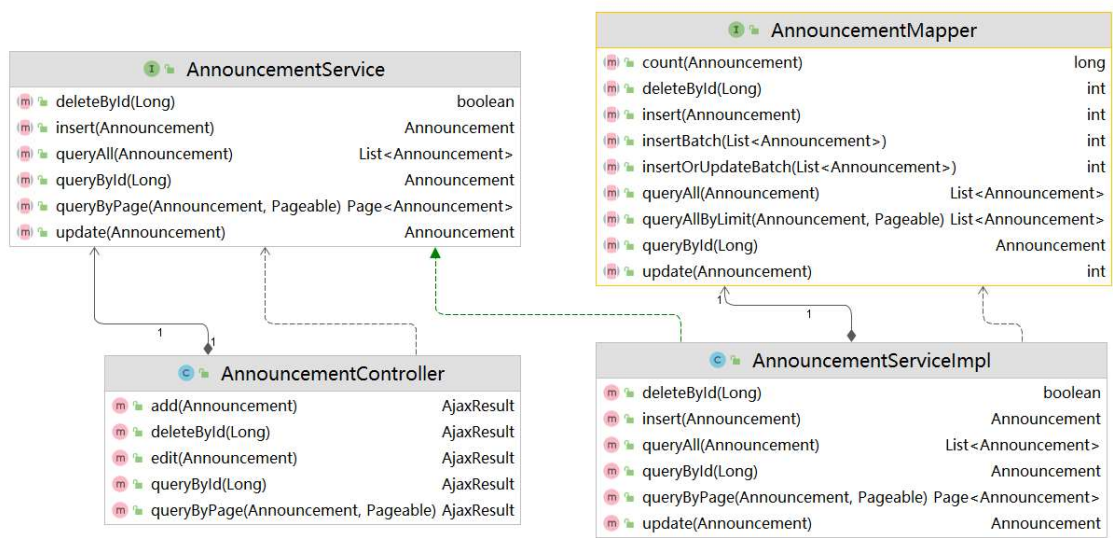


图 3.14 通知模块类图

表 3.2 AnnouncementServiceImpl 详细设计

类名	AnnouncementServiceImpl
接口	●AnnouncementService 接口：通知模块的业务类接口，有关于通知模块的所有功能。
属性	●private AnnounceMap announcementMap 属性：Mybatis 框架中的 Dao 实现类。连接数据库存储数据。
方法	<div>●public Boolean deleteById(Long): 根据主键删除通知信息。形参传一个主键，用于 Dao 层在数据库中找到对应信息。返回值返回是否删除成功。</div> <div>●public void insert(Announcement): 将形参中的通知类信息，转换成数据库中数据表的信息，并存储到数据库中。返回值返回存储入数据表后的值。</div> <div>●public void queryAll(Announcement): 将形参中的通知类信息作为查找条件，在数据表中查找对应的信息并返回。返回值返回符合条件的通知类数组。</div> <div>●public void queryById(Long): 根据主键查找对应的数据。形参传入一个主键，用于 Dao 层在数据库中找到对应信息并返回。返回值返回一个数</div>

	<p>据对象。</p> <p>●public void queryByPage(Announcement, Pageable): 将形参中的通知类信息作为查找条件，在数据表中查找固定个数的信息并返回。返回值返回符合条件的通知类数组。</p> <p>●public void update(Announcement): 将形参中的通知类信息，更新到数据库中。返回值返回存储入数据表后的值。</p>
--	---

新闻模块设计：

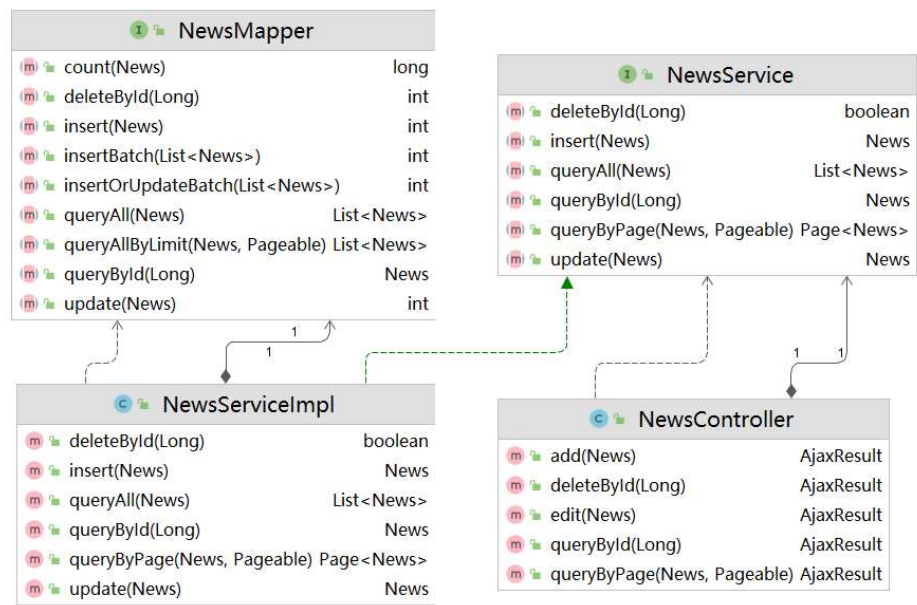


图 3.15 新闻模块类图

表 3.3 NewsServiceImpl 详细设计

类名	NewsServiceImpl
接口	● NewsService 接口：图书模块的业务类接口，有关于图书模块的所有功能。
属性	● private NewsMap newsMap 属性：Mybatis 框架中的 Dao 实现类。连接数据库存储数据。
方法	● public Boolean deleteById(Long): 根据主键删除新闻信息。形参传一个主键，用于 Dao 层在数据库中找到对应信息。返回值返回是否删除成

功。

●**public void insert(News):** 将形参中的新闻类信息，转换成数据库中数据表的信息，并存储到数据库中。返回值返回存储入数据表后的值。

●**public void queryAll(News):** 将形参中的新闻类信息作为查找条件，在数据表中查找对应的信息并返回。返回值返回符合条件的图书类数组。

●**public void queryById(Long):** 根据主键查找对应的数据。形参传入一个主键，用于 Dao 层在数据库中找到对应信息并返回。返回值返回一个数据对象。

●**public void queryByPage(News, Pageable):** 将形参中的新闻类信息作为查找条件，在数据表中查找固定个数的信息并返回。返回值返回符合条件的新闻类数组。

●**public void update(News):** 将形参中的新闻类信息，更新到数据库中。返回值返回存储入数据表后的值。

漂流模块：

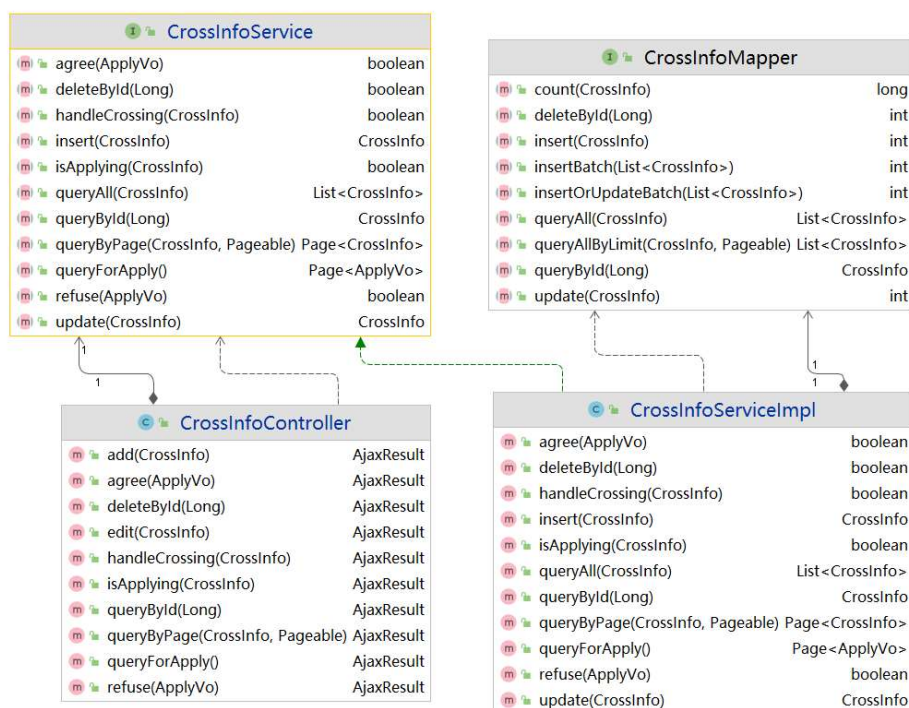


图 3.16 漂流模块类图

表 3.4 CrossInfoServiceImpl 详细设计

类名	CrossInfoServiceImpl
接口	<ul style="list-style-type: none"> ●CrossInfoService 接口：漂流模块的业务类接口，有关于漂流模块的所有功能。
属性	<ul style="list-style-type: none"> ●private CrossInfoMap crossInfoMap 属性：Mybatis 框架中的 Dao 实现类。连接数据库存储数据。
方法	<ul style="list-style-type: none"> ●public Boolean deleteById(Long)：根据主键删除漂流信息。形参传一个主键，用于 Dao 层在数据库中找到对应信息。返回值返回是否删除成功。 ●public void insert(CrossInfo)：将形参中的漂流类信息，转换成数据库中数据表的信息，并存储到数据库中。返回值返回存储入数据表后的值。 ●public void queryAll(CrossInfo)：将形参中的漂流类信息作为查找条件，在数据表中查找对应的信息并返回。返回值返回符合条件的漂流类数组。 ●public void queryById(Long)：根据主键查找对应的数据。形参传入一个主键，用于 Dao 层在数据库中找到对应信息并返回。返回值返回一个数据对象。 ●public void queryByPage(Book, Pageable)：将形参中的漂流类信息作为查找条件，在数据表中查找固定个数的信息并返回。返回值返回符合条件的漂流类数组。 ●public void update(CrossInfo)：将形参中的漂流类信息，更新到数据库中。返回值返回存储入数据表后的值。

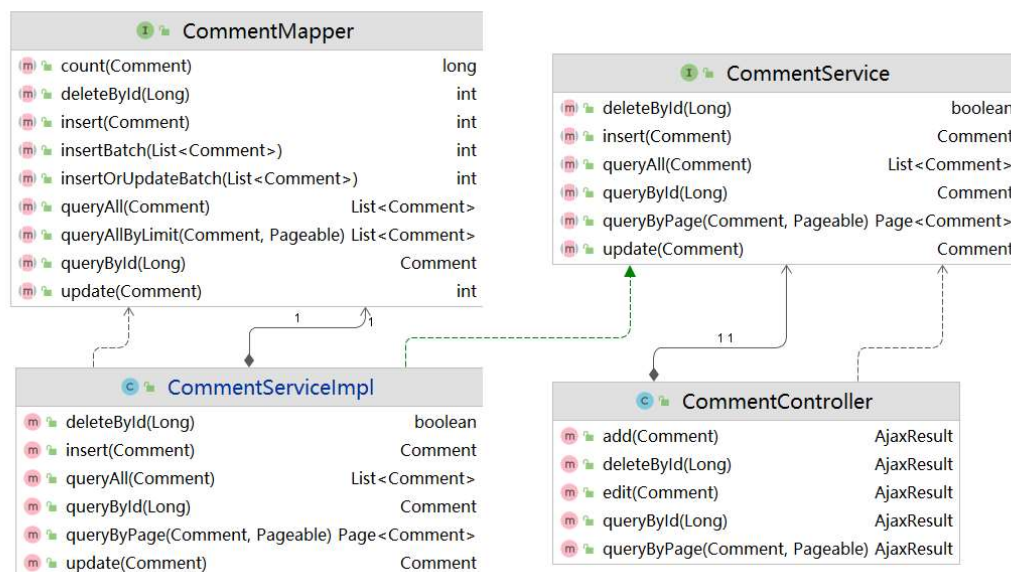


图 3.17 评论模块类图

表 3.5 CommentServiceImpl 详细设计

类名	CommentServiceImpl
接口	●CommentService 接口：评论模块的业务类接口，有关于评论模块的所有功能。
属性	●private CommentMap commentMap 属性：Mybatis 框架中的 Dao 实现类。连接数据库存储数据。
方法	<p>●public Boolean deleteById(Long)：根据主键删除评论信息。形参传一个主键，用于 Dao 层在数据库中找到对应信息。返回值返回是否删除成功。</p> <p>●public void insert(Comment)：将形参中的评论类信息，转换成数据库中数据表的信息，并存储到数据库中。返回值返回存储入数据表后的值。</p> <p>●public void queryAll(Comment)：将形参中的评论类信息作为查找条件，在数据表中查找对应的信息并返回。返回值返回符合条件的评论类数组。</p> <p>●public void queryById(Long)：根据主键查找对应的数据。形参传入一个主键，用于 Dao 层在数据库中找到对应信息并返回。返回值返回一个数据对象。</p>

	<div>●public void queryByPage(Comment, Pageable): 将形参中的评论类信息作为查找条件，在数据表中查找固定个数的信息并返回。返回值返回符合条件的评论类数组。</div> <div>●public void update(Comment): 将形参中的评论类信息，更新到数据库中。返回值返回存储入数据表后的值。</div>
--	---

用户模块：

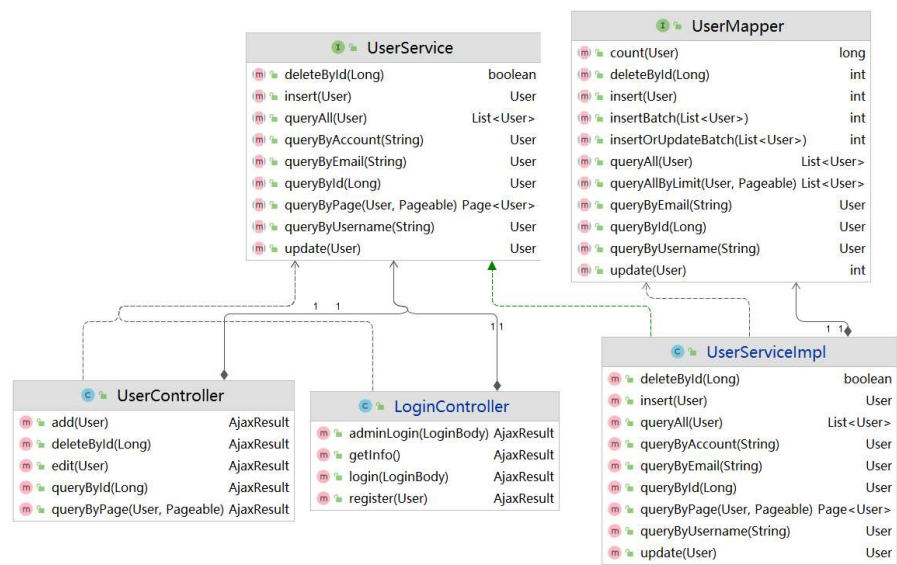


图 3.18 用户模块类图

表 3.6 UserServiceImpl 详细设计

类名	ServiceImpl
接口	● UserService 接口：用户模块的业务类接口，有关于用户模块的所有功能。
属性	● private UserMap userMap 属性：Mybatis 框架中的 Dao 实现类。连接数据库存储数据。
方法	<div>●public Boolean deleteById(Long): 根据主键删除用户信息。形参传一个主键，用于 Dao 层在数据库中找到对应信息。返回值返回是否删除成功。</div> <div>●public void insert(User): 将形参中的用户类信息，转换成数据库中数据</div>

	<p>表的信息，并存储到数据库中。返回值返回存储入数据表后的值。</p> <ul style="list-style-type: none">●public void queryAll(User): 将形参中的用户类信息作为查找条件，在数据表中查找对应的信息并返回。返回值返回符合条件的用户类数组。●public void queryById(Long): 根据主键查找对应的数据。形参传入一个主键，用于 Dao 层在数据库中找到对应信息并返回。返回值返回一个数据对象。●public void queryByPage(User, Pageable): 将形参中的图书类信息作为查找条件，在数据表中查找固定个数的信息并返回。返回值返回符合条件的用户类数组。●public void update(User): 将形参中的用户类信息，更新到数据库中。返回值返回存储入数据表后的值。
--	--

3.5 原型设计

顶部导航栏左侧设计：

点击左侧 logo 可以跳转到首页页面；点击书与人可以跳转至图书列表页面；点击公告可以跳转至公告页面查看站内公告；点击新闻可以跳转至新闻页面查看站内所有新闻。



图 3.19 顶部左侧导航栏

顶部导航栏右侧设计：

未登录时，右侧导航栏有两个按钮：注册和登录。点击注册按钮，可以跳转至注册页面进行注册账号；点击登录按钮，可以跳转至登录页面进行登录账号。

登录账号之后，右侧导航栏将会变成只有一个个人中心按钮，可以进入个人中心，查看相关用户信息并进行相应操作。



图 3.20 顶部右侧导航栏

首页页面设计：

用户进入网站时，最先看到的将是首页页面，因此首页的设计将至关重要。



BookCrossing 傳媒中心

The New York Times The Washington Post San Francisco Chronicle

BBC THE NEW YORK TIMES CNN THE WALL STREET JOURNAL

THE CHRISTIAN SCIENCE MONITOR WIRE D Los Angeles Times CBC

npr USA TODAY THE GLOBE AND MAIL UTNE

Chicago Tribune PC The Philadelphia Inquirer Reader's digest

the guardian The Herald-Sun FamilyCircle DER SPIEGEL

注册

图 3.21 首页页面

注册页面设计：

注册功能是让访问系统的游客变为正式用户的途径。在注册时，需要让用户填写一些必要的信息，如用户名、邮箱、密码等。

用户名称：只能由汉字、字母、数字组成，不能出现其他特殊字符。

邮箱地址：必须符合邮箱格式。作为用户登录时的账号。

账号密码：可以由汉字、字母、数字和一些特殊字符组成，用于登录时的密码校验。

加入書遊記

你即將與來自世界各地超過1,925,263名的愛書同好聚在一起...

這就是書遊記的宗旨...

我們是一個愛書人的社群，喜歡與他人分享我們的書籍。只需要簡單的手續，為你的書貼上一張書遊記標籤，然後將書本“漂流”出去。隨後登入網站，看看它們被誰發現，去了世界哪個角落。

“追隨”書本在世界各地的旅程，看著它們開創自己的生活，既有趣又帶勁。並知道它們會到哪裡去。被誰發現之後，回到這裡輸入你在標籤上寫下的編號...

* 用戶名稱

* 郵箱地址

* 账号密码

立即注册

图 3.22 注册页面

登录页面设计：

用户可以在本系统的登录界面中输入用户的邮箱地址和对应的密码，输入完成即可进行登录。输入完信息后，系统会在后端进行校验，确认无误后即可登录成功。



登入

请输入你的邮箱地址和密码。如果你没有帐户，[注册](#)

* 邮箱地址

* 密码

登录

图 3.23 登录界面

图书列表页面设计：

该页面是本系统的主要页面之一，该页面显示了本系统中所有的图书，用户可以按照自己的需求和喜好进行选择，找到心仪的图书后可以点击图书所在的行进入到具体的图书信息页面中。



图书编号	图书名称	图书作者	图书类型	图书状态	上传日期
SN0001	红楼梦	曹雪芹	2	1	2021-09-23T11:19:48.000+08:00
SN0002	三国演义	罗贯中	2	1	2021-09-23T11:52:20.000+08:00
SN0003	水浒传	施耐庵	2	1	2021-09-23T11:55:40.000+08:00
SN0003	水浒传	施耐庵	2	1	2021-09-23T12:01:04.000+08:00

公告

- 图书漂流系统
- 图书漂流系统

新闻

- 第一条新闻
- 第二条新闻
- 第三条新闻

图 3.24 图书列表页面

图书详情页面设计：

该页面显示了图书的具体信息和漂流状态。可以在该页面中查看图书的具体信息以及图书的漂流状态：可漂流、漂流中、还是漂流完成。也可以在页面下方对图书的内容、完好度进行评价。良好的评价可以对其他用户是否选择漂流该图书起到一个积极的作用。

The image shows a web interface for a book detail page. At the top, there is a navigation bar with a back arrow and the text '详情页面'. Below this is a header section titled '图书详情'. The main content area displays the following information: '图书编号 SN0001', '图书名称 红楼梦', '图书作者 曹雪芹', and '上传时间 2021-09-23T11:19:48.000+08:00'. There is a blue button labeled '申请漂流'. Below this is a section titled '书评区'. It contains a table with two columns: '内容' and '评论时间'. A sample row shows the content '这是一条关于红楼梦的书评!' and the time '2021-09-24T15:43:00.000+08:00'. At the bottom, there is a text input field and a green button labeled '发布评论'.

图 3.25 图书详情页面

个人中心页面设计：

用户的个人中心页面显示了用户的所有信息，如用户当前拥有哪些图书，申请了哪些图书，以及哪些图书在申请过程中等等。用户也可以在个人中心对其他用户的漂流请求进行处理。其他用户如果申请漂流用户手头上的闲置书籍，将会显示在申请漂流的列表中，用户可以进行同意或者拒绝操作。

用户资料

用户名: 张三
邮箱号: xxxxxx@163.com
手机号: 13075808800

编辑资料

退出登录

拥有的图书列表

上传图书

图书编号	图书名称	图书作者	图书类型	图书状态	上传时间	操作
SN0001	红楼梦	曹雪芹	2	1	2021-09-23T11:19:48.000+08:00	<div>编辑</div> <div>下架</div>
SN0002	三国演义	罗贯中	2	1	2021-09-23T11:52:20.000+08:00	<div>编辑</div> <div>下架</div>
SN0003	水浒传	施耐庵	2	1	2021-09-23T11:55:40.000+08:00	<div>编辑</div> <div>下架</div>
SN0003	水浒传	施耐庵	2	1	2021-09-23T12:01:04.000+08:00	<div>编辑</div> <div>下架</div>

申请图书的列表

图书编号	图书名称	图书作者	上传时间	申请人	申请时间	操作
No Data						

漂流信息

图书编号	图书名称	图书作者	上传时间	漂流状态	更新时间
No Data					

图 3.26 个人中心页面

后台登录页面设计:

后台是管理员进行操作的平台,所有安全性也会相对更高。管理员登录时所填写的信息和普通用户相差不多,但在系统检验时将会进行更多的校验工作。

登入

* 邮箱地址

* 账号密码

登录

图 3.27 后台登录页面

后台主页面设计:

在后台主页面中。是管理员对系统的各个部分进行管理的入口。

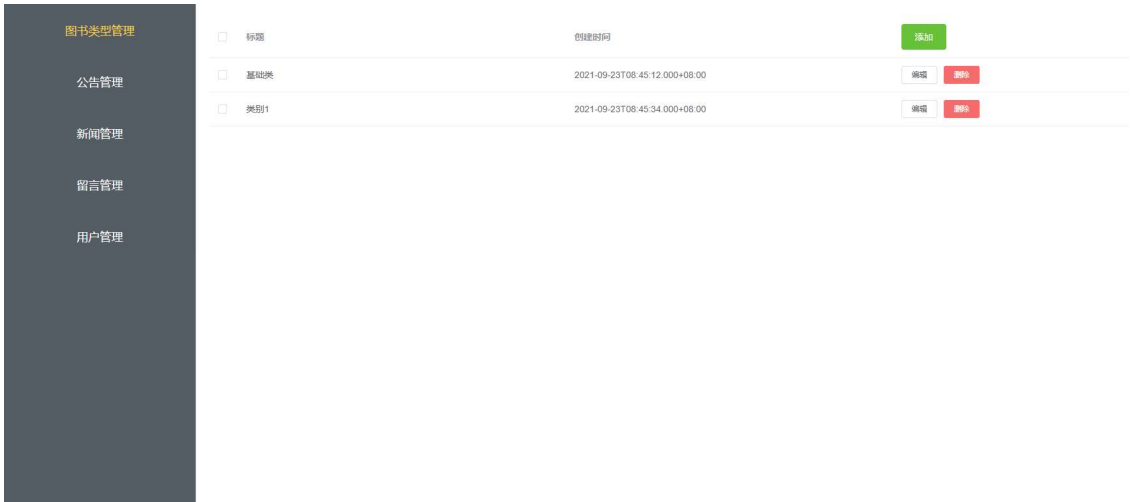


图 3.28 后台管理页面

第四章 系统实现与测试

前面两个章节介绍了本系统的需求分析以及系统设计等部分，本节介绍该系统各个功能的具体实现方式。本节将列举本系统的关键代码以及代码的详细说明，此外还将展示系统各个界面的截图。

4.1 系统运行环境

本系统前端采用 Vue 框架使代码组件化，层次化，降低代码的耦合性，后端代码采用 SpringBoot 框架使代码灵活提高扩展性。

系统环境：

开发环境：Windows 10、MySQL 5.7、Redis 3.2、JDK 1.8、Tomcat 9.0.31

开发语言：Html、Css、TypeScript、Java

开发工具：IDEA 2021、Navicat 15、Postman

MySQL 是一款安全、跨平台、高效的，并与 PHP、Java 等主流编程语言紧密结合的数据库系统，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，使得很多公司都采用 MySQL 数据库以降低成本，因此本系统采用 Mysql5.7 版本。

JDK 目前最新版是 15，但是开发系统需要稳定的版本所以该系统将采用最稳定的 JDK1.8。

4.2 系统层次结构

后端代码部分主要分为三大部分，common 基础类、framework 框架类、project 项目类，common 包里主要是一些常量类，工具类和异常类。Framework 包中主要是一些框架配置类，如 Spring Security、Redis，Swagger、web 等配置类。项目类中才真正是业务实现的类。项目类中主要分为 3 个层：Controller 层、Service 层、Mapper 层。如下是后端代码层次图：

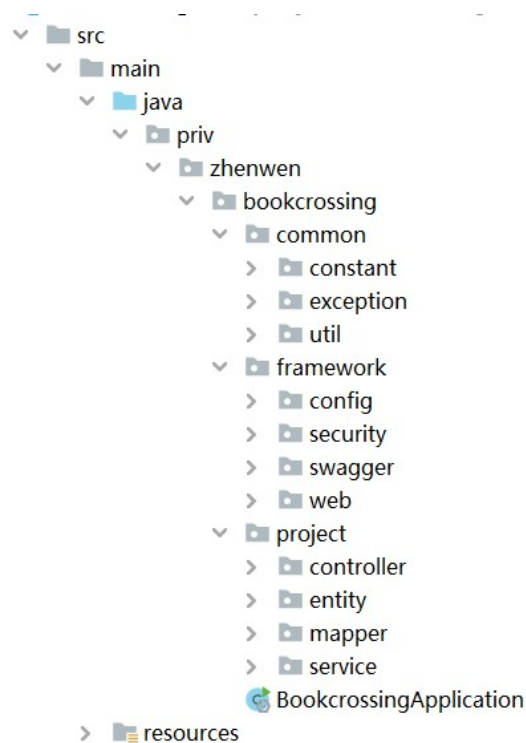


图 4.1 后端代码层次图

前端代码分为 assets、components、request、router、store、views 等。assets 主要负责存放一些静态文件，如 css、js、img 等文件。components 主要负责存放一些公用的组件。request 则负责网络请求等，如对 axios 的封装、对后端数据的预处理等。router 负责实现页面之间的跳转。store 则对组件状态的管理。如下是前端代码层次图：

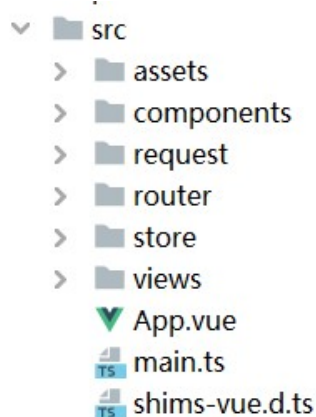


图 4.2 前端代码层次图

4.3 具体功能实现

4.3.1 权限校验和拦截器

在过滤器中设置在前端用户发送请求时可以携带 token 令牌便于后端判断该用户是否登录保证接口的安全性。实现 filter 接口在 doFilterInternal 中校验 token 参数，并在在文件

配置过滤器使其生效，如下图：

```
@Override
protected void doFilterInternal(HttpServletRequest request,
                                HttpServletResponse response,
                                FilterChain chain) throws ServletException, IOException {
    LoginUser loginUser = tokenService.getLoginUser(request);
    if (StringUtils.isNotNull(loginUser) && StringUtils.isNull(SecurityUtils.getAuthentication())) {
        tokenService.verifyToken(loginUser);
        UsernamePasswordAuthenticationToken authenticationToken =
            new UsernamePasswordAuthenticationToken(loginUser, credentials: null, loginUser.getAuthorities());
        authenticationToken.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
        SecurityContextHolder.getContext().setAuthentication(authenticationToken);
    }
    chain.doFilter(request, response);
}
```

图 4.3 token 拦截器代码

在拦截器中判断 token 是否存在并且有效，如果无效就无法访问需要 token 的接口，并且在拦截器中还可以设置拦截路径，让某些路径不需要 token 就可以访问。

如果前端传来的 token 有效，就在 Redis 中刷新 token 的有效时间。

让 Security 的配置类继承 WebSecurityConfigurerAdapter 类并重写 configure 方法，在 configure 方法中添加不需要 token 校验的路径。如下图：

```
@Override
protected void configure(HttpSecurity httpSecurity) throws Exception
{
    httpSecurity
        .csrf().disable()
        .exceptionHandling().authenticationEntryPoint(unauthorizedHandler).and()
        .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS).and()
        .authorizeRequests()
        .antMatchers( ...antPatterns: "/login").anonymous()
        .antMatchers( ...antPatterns: "/admin/login").anonymous()
        .antMatchers( ...antPatterns: "/register").anonymous()
        .antMatchers(
            HttpMethod.GET,
            ...antPatterns: "/*.html",
            "**/*.html",
            "**/*.css",
            "**/*.js",
            "/resources/download"
        ).anonymous()
        // 除上面外的所有请求全部需要鉴权认证
        .anyRequest().authenticated()
        .and()
        .headers().frameOptions().disable();
    httpSecurity.logout().logoutUrl("/logout").logoutSuccessHandler(logoutSuccessHandler);
    // 添加JWT filter
    httpSecurity.addFilterBefore(authenticationTokenFilter, UsernamePasswordAuthenticationFilter.class);
}
```

图 4.4 Security 配置类代码

4.3.2 图书漂流

用户在个人中心中上传图书后，系统将生成图书记录，图书记录一开始处于可出借状态，供用户来漂流。用户申请漂流可出借状态的图书后，系统生成漂流记录，此时，对申请漂流的用户，漂流的图书状态变为漂流中。拥有此图书的用户可以同意或者拒绝这次的

漂流申请。如果用户拒绝漂流申请，这图书的状态将变回可出借状态，用户可再次申请漂流；如果用户同意此次漂流申请，图书的状态将变为已出借，并同时改变图书的拥有者信息。

```
@Override
public Book insert(Book book) {
    if (StringUtils.isNull(book.getUserId())) {
        book.setUserId(loginService.getUserId());
    }
    if (StringUtils.isEmpty(book.getStatus())) {
        book.setStatus("1");
    }
    book.setCreateDate(new Date());

    this.bookMapper.insert(book);
    return book;
}
```

图 4.5 添加图书代码

当用户申请漂流任一图书之后，系统将生成漂流记录，一开始漂流记录处于申请中状态，拥有该图书的用户可以同意或拒绝该申请。如果用户同意申请，漂流记录状态将转变成已完成，如果用户拒绝申请，漂流记录状态将转变成已拒绝，可以再次发起漂流申请来漂流该图书。

```
/**
 * 新增数据
 *
 * @param crossInfo 实例对象
 * @return 实例对象
 */
@Override
public CrossInfo insert(CrossInfo crossInfo) {
    this.crossInfoMapper.insert(crossInfo);
    return crossInfo;
}
```

图 4.6 添加漂流代码

```
/**
 * 处理漂流请求
 * @param crossInfo 漂流请求
 * @return 是否成功
 */
@Override
public boolean handleCrossing(CrossInfo crossInfo) {
    crossInfo.setStatus("1");
    crossInfo.setType("0");
    crossInfo.setCreateDate(new Date());
    crossInfo.setUpdateDate(new Date());

    Long userId = loginService.getUserId();

    crossInfo.setUserId(userId);

    return this.crossInfoMapper.insert(crossInfo) > 0;
}
```

图 4.6 处理漂流请求代码

4.4 测试

测试的目标是为了寻找开发系统中的潜在的各种危险和不足，通过测试工具的手段来实现用最少的人力成本和开发成本对系统进行测试。利用以下举例的测试流程对系统的重要功能进行测试，来确保系统能够正确的进行运行。

4.4.1 测试流程

首先，进行前台登录用例的测试，前台登录用例是用户进入系统并使用系统功能的入口。用户可以通过首页右上角的登录按钮进入登录页面，输入账号密码进行登录。

表 4.1 前台登录测试用例

功能路径	前台/登录
用例编号	01
用例名称	前台登录
用例目的	让用户登录系统进行操作
预置条件	用户未登录

输入数据	账号: xxxxxx@163.com 密码: zzw12345
操作步骤	1. 点击首页右上方登录按钮, 进入登录页面; 2. 输入账号密码; 3. 点击登录;
预期结果	登录成功。
实际结果	登录成功。

注册用户是游客成为系统用户的一个途径, 让游客通过审核后, 成为系统用户, 进而对本系统的用户功能进行操作。前台注册用例也是很重要的一个用例, 它是让系统添加新用户的关键用例。

表 4.2 前台注册测试用例

功能路径	前台/注册
用例编号	02
用例名称	前台注册
用例目的	让新用户注册账号使用系统
预置条件	用户未登录
输入数据	账号: xxxxxx@163.com 密码: zzw12345 确认密码: zzw12345
操作步骤	1. 点击首页右上方注册按钮, 进入注册页面; 2. 输入账号密码和确认密码; 3. 点击注册;
预期结果	注册成功。
实际结果	注册成功。

浏览信息是最基础的功能, 也是不容许出任何差错的一个功能, 如果用户浏览到的信息是错误的话, 更何谈后续的申请图书和漂流图书呢。所以以下的测试用例是对浏览信息功能展开的。

表 4.3 浏览信息测试用例

功能路径	前台/浏览信息
用例编号	03
用例名称	浏览信息
用例目的	让已登录的用户浏览信息
预置条件	用户已登录
输入数据	无
操作步骤	1. 点击首页左上方菜单栏，进入信息页面； 2. 系统显示信息；
预期结果	信息显示成功。
实际结果	信息显示成功。

上传图书功能是本系统添加新图书的重要途径之一。用户可以将自己手头上的图书添加到系统中，别的用户才有新的图书可以进行漂流。下表是对上传图书用例的一个测试用例。

表 4.4 上传图书测试用例

功能路径	前台/上传图书
用例编号	04
用例名称	上传图书
用例目的	用户可在个人页面上上传图书
预置条件	用户已登录
输入数据	图书编号：SN0004 图书名称：西游记 图书作者：吴承恩 图书介绍：四大名著之一：《西游记》 图书类型：文学
操作步骤	1. 点击首页右上方登录按钮，进入登录页面； 2. 输入账号密码； 3. 点击登录；
预期结果	登录成功。

实际结果	登录成功。
------	-------

申请漂流图书功能是本系统的核心功能。只有申漂图书功能是正常运行的情况下，本系统才能让用户之间进行漂流图书。

表 4.5 申漂图书测试用例

功能路径	前台/申漂图书
用例编号	05
用例名称	申漂图书
用例目的	用户可在图书信息页面申请漂流该图书
预置条件	用户进入图书信息页面并未漂流该图书
输入数据	无
操作步骤	1. 点击要漂流的图书，进入图书信息界面； 2. 点击申请漂流按钮； 3. 漂流图书成功；
预期结果	漂流图书成功。
实际结果	漂流图书成功。

漂流图书是用户在图书信息页面申请漂流之后才会出现的用例。该测试用例主要是测试拥有图书的用户是否能正常的进行同意或者拒绝漂流图书。

表 4.6 漂流图书测试用例

功能路径	前台/漂流图书
用例编号	06
用例名称	漂流图书
用例目的	用户可在个人页面的求漂申请进行同意，即漂流该图书。
预置条件	用户进入个人页面并且有可同意的求漂申请
输入数据	无
操作步骤	1. 用户选择一个求漂信息； 2. 用户对求漂申请进行同意；
预期结果	漂流图书成功。
实际结果	漂流图书成功。

评论功能是为了丰富图书的信息，让用户能更加直观的看到整本书的一些评价，以及该书的完好度，内容是否符合自己的喜好等内容。主要测试是否能够正常评论以及评论是否可以正常展示出来。

表 4.7 在线评论测试用例

功能路径	前台/在线评论
用例编号	07
用例名称	在线评论
用例目的	用户可在图书的详情页面下方，对图书的内容等等，进行评价。
预置条件	用户进入个人页面并且有可同意的求漂申请。
输入数据	评论内容：这是一条评论消息。
操作步骤	1. 用户点击详情页面下方的评论框； 2. 用户在输入框中输入评论信息； 3. 用户点击评论按钮；
预期结果	评论成功。
实际结果	评论成功。

用户登录后，可以进入个人中心修改之前注册时填写的个人信息。测试用户是否能够正常修改用户信息。让用户在想更正自己的个人信息时不出差错。

表 4.8 修改个人信息测试用例

功能路径	前台/修改个人信息
用例编号	08
用例名称	修改个人信息
用例目的	用户可在个人页面中，对用户的个人信息进行修改。
预置条件	用户进入个人页面。
输入数据	用户名：张三 邮箱： xxxxxx@163.com 手机号：13075808800
操作步骤	1. 用户点击个人页面中的修改个人信息按钮； 2. 用户输入个人信息数据

	3. 用户点击完成按钮；
预期结果	个人信息修改成功。
实际结果	个人信息修改成功。

该功能主要是让用户在图书列表中按图书的内容来查找特定的图书。测试是否是符合要求的图书被查找出来。符合要求的图书是否全部显示出来。

表 4.9 查询图书测试用例

功能路径	前台/查询图书
用例编号	09
用例名称	查询图书
用例目的	用户可在图书列表中输入想要查找的图书信息，对图书进查找。
预置条件	用户进入浏览图书列表页面。
输入数据	查询图书信息：西游记
操作步骤	1. 用户点击图书列表中的查询图书按钮； 2. 用户输入图书信息 3. 用户点击查询按钮；
预期结果	图书查询成功。
实际结果	图书查询成功。

在用户的个人中心显示漂流记录，测试显示的漂流记录是否正常。让用户能够查看自己具体漂流了哪些书籍，哪些书籍被其他用户漂流。

表 4.10 浏览漂流记录测试用例

功能路径	前台/浏览漂流记录
用例编号	10
用例名称	浏览漂流记录
用例目的	用户可在个人页面中查看自己的漂流记录。
预置条件	用户进入个人信息页面。
输入数据	无
操作步骤	1. 用户进入个人页面； 2. 用户查看浏览漂流记录；

预期结果	漂流记录查询成功。
实际结果	漂流记录查询成功。

总结

完成了历时三个月的毕业设计，从最开始的开题报告再到文献综述最后到系统的设计与开发，这一路走来收获颇多。最开始刚刚接触到这个主题——“图书漂流”时，我还是一头雾水，感觉大学阶段完全没有遇到过这个概念。后来通过查阅资料和咨询老师，我才知道了图书漂流是什么意思，怎么进行的。

在了解了图书漂流的意思和运转模式后，我便开始构思如何将这个理论体系转换为具体的功能代码。我在网上学习了现在互联网上主流的框架。学习如何实现前后端分离，前端具体怎么把请求传递到后端，传递的过程是怎样的，中间有可能会遇到哪些问题，遇到了这些问题，应该如何解决，各个解决方案的优缺点有哪些，应该如何根据实际情况来进行选择。在学习前端页面的布局设计时，也让我受益良多。后端的设计涉及到很多的逻辑问题，也有很多细节上的问题，通过解决这些问题，也让我收获到了很多东西。

通过完成这次的图书漂流系统，我对于图书漂流这个理念有了更深的认识，同时我的技术水平也有了很大的提升。而且使我能综合起来去应用之前所学的计算机网络、数据结构与算法、设计模式、计算机组成原理、操作系统等各个学科的知识，进行更加系统和全面的技术以及能力的练习。我相信，通过这次能力的锻炼，我能在今后的人生道路和职业生涯中大放异彩。

致谢

经过一个月的努力，终于完成了图书漂流系统的设计与实现。在做毕业设计的时候，王老师大概给我讲解了一下本系统的情况，刚开始对系统还一无所知。但由于时间紧任务重，在接到老师给的题目之后，马上就查阅了相关类似的系统。完成了这次毕业设计。

第一，要感谢王华秋老师在这次的毕业设计上给我提供的各种指导、和给予我的支持和帮助，这次能顺利的完成毕业设计离不开王老师的这些帮助，在我遇到一些棘手的技术问题时，王老师总是能点播我，让我攻克难关并把系统做得更加完善。在毕业设计中，王华秋老师展现的学术思维很敏锐，知识储备也十分丰富令我折服。王华秋老师刻画入微的工作态度和一丝不苟的教学作风是我在学术路上的标杆。

感谢四年以来所有传授我知识的各位老师和同学们，没有他们一直以来的辛勤付出和不屑的教导，就没有今天这硕大的成果。专业老师在我刚入校园时，激发了我学习编程的乐趣，也让我对今后的人生和职业生涯有了明确的方向。在我最初学习编程时，遇到了很多不懂和不理解的地方，都是各位老师虚心教导我，让我重新燃起了学习如何处理编程中遇到的各种问题的乐趣。

感谢一起成长的三位室友，在遇到难题时，相互讨论并给出了不同的解决方案供我选择，在我困难时给予了最大的帮助。在我遇到挫折时会激励我，让我重新燃起斗志克服困难。感谢我的家庭给予我经济上的支持以及精神的鼓舞。

最后，我要向对本文进行评阅的各位老师表示感谢。

参考文献

- [1]杨红岗,苏楠.我国图书漂流存在的问题及其发展对策[J].新世纪图书馆,2012(08):29-31.DOI:10.16810/j.cnki.1672-514x.2012.08.008.
- [2]bosco_liu, w_ou, 等.图书漂流[EB/OL].[2021-01-27].<https://baike.baidu.com/item/图书漂流/4423039>.
- [3]刘洪伟,黄羽中,吴泽彬,等.基于 SSH2 的图书漂流系统的设计与实现[J].价值工程,2014,000(011):224-225.
- [4]黄金晶,李文骏.利用 UML 的面向对象分析建模[J].2021(2011-5):13-15.
- [5]杨家炜.基于 Spring Boot 的 web 设计与实现[J].轻工科技,2016,000(007):86-89.
- [6]王松.Spring Boot+Vue 全栈开发实践[M].清华大学出版社,2018:252-256.
- [7]林艺山.基于 O2O 的高校图书馆图书漂流模式构建[J].2022(1).
- [8]杨于卜,刘雁.我国图书漂流网站建设现状分析[J].2022(10).
- [9]郑然.基于互联网的图书漂流研究.
- [10]王静,负琳红.高校图书馆读者服务新举措——以西安电子科技大学"图书漂流"为例[J].2022(10).