湖 北 大 学

本 科 毕 业 论 文 （设 计）

题 目 基于微信小程序地共享书城

设计与实现

姓 名 石义钊

学 号 201722110820021

专业年级 软件工程2017级

学 院 计算机与信息工程学院

指导教师 /职称（可填写1-3人）

金红/讲师

二○ 二一 年 五月 九日

目 录

[1 绪论 1](#_Toc2986)

[1.1 研究目的及意义 1](#_Toc31682)

[1.2 国内外研究现状 2](#_Toc20271)

[1.3 论文研究内容 2](#_Toc5436)

[1.4 论文组织结构 3](#_Toc16329)

[2 相关理论与技术 3](#_Toc20723)

[2.1 微信小程序的运行原理 3](#_Toc5682)

[2.1.1 微信小程序的配置 4](#_Toc6830)

[2.1.2 微信小程序的自定义组件 4](#_Toc5437)

[2.2 Spring Boot后台开发框架介绍 5](#_Toc16246)

[2.2.1 Spring全家桶介绍 5](#_Toc7502)

[2.2.2 Spring Boot框架介绍 6](#_Toc12209)

[2.2.3 Spring framework框架介绍 6](#_Toc28788)

[2.2.4 Spring Security框架介绍 7](#_Toc4944)

[2.2.5 Spring MVC框架介绍 7](#_Toc21632)

[2.3 Mybatis-plus持久层开发框架 8](#_Toc18457)

[2.4 Layui前端开发框架 8](#_Toc27267)

[2.5 Thymeleaf模板引擎 9](#_Toc25492)

[2.6 ajax异步传输 9](#_Toc4000)

[2.7 json数据格式统一化 9](#_Toc23856)

[3 系统需求分析与设计 9](#_Toc31000)

[3.1 共享书城的需求分析 9](#_Toc27272)

[3.1.1 个人信息模块 9](#_Toc3569)

[3.1.2 共享社区模块 10](#_Toc9981)

[3.1.3 书籍发表模块 11](#_Toc21168)

[3.1.4 消息管理模块 11](#_Toc10499)

[3.2 共享书城的数据库设计 12](#_Toc8183)

[3.2.1 数据库ER图设计 12](#_Toc31430)

[3.2.2 基于UML的数据库设计 13](#_Toc25354)

[3.2.3 数据库表结构的设计 14](#_Toc22809)

[3.3 共享书城的前端设计 17](#_Toc24456)

[3.3.1 登陆注册页面设计 17](#_Toc29395)

[3.3.2 个人信息页面设计 17](#_Toc25563)

[3.3.3 共享社区模块页面设计 19](#_Toc29690)

[3.3.4 文章发表模块设计模块 20](#_Toc11814)

[3.3.5 好友模块页面设计 21](#_Toc17269)

[3.4 共享书城的后台设计 23](#_Toc28099)

[3.4.1 用户管理模块 23](#_Toc20248)

[3.4.2 文章管理模块 23](#_Toc28044)

[3.4.3 书库管理模块 24](#_Toc24903)

[4 系统实现 24](#_Toc5504)

[4.1 前端实现 25](#_Toc5256)

[4.1.1 前期配置 25](#_Toc9863)

[4.1.2 登陆注册功能实现 26](#_Toc23732)

[4.1.3 个人信息渲染和个性签名功能实现 26](#_Toc26978)

[4.1.4 社区模块的渲染和模糊查询 27](#_Toc6827)

[4.1.5 发表社区和百度地图api 27](#_Toc4615)

[4.2 后端实现 27](#_Toc5817)

[4.2.1 用户管理 27](#_Toc25342)

[4.2.2 文章管理 28](#_Toc24333)

[4.2.3 书库管理 28](#_Toc14454)

[5 系统测试 28](#_Toc13036)

[5.1 服务端运行结果和测试 28](#_Toc8487)

[5.2 小程序端运行结果及测试 29](#_Toc10381)

[6 展望与总结 30](#_Toc6984)

[参考文献 32](#_Toc27326)

[附录一：application.yml配置文件 33](#_Toc32282)

[附录二：pom.xml配置文件 34](#_Toc5311)

基于微信小程序的共享书城的设计与实现

摘 要

本课题是在我国提出文化自信的政治背景下进行设计的，整个系统的设计是前后端分离的，前端是基于微信小程序的客户端和基layui搭建的后台管理页面，后端则是基于Spring Boot和mybatis-plus搭建的，后台业务逻辑是独立实现的，而数据来源则是通过MySQL存储。在微信小程序端也会做一些简单的API调用，如百度地图和百度天气，利用百度提供的API接口实现地图查询和天气查询功能。

还有就是共享，系统设计的共享社区和共享书库是为用户量身打造的，用户都可以免费访问，强大的模糊查询可以快速搜索到自己想要看的书籍，整个过程是基于微信小程序的组件化开发和微信接口发送请求，通过请求回来的数据进行即时渲染就可以搜索到自己喜欢的书籍。后台搭建的mybatis-plus也是可以快速注解开发，整合的spring security也可以做后管理员的权限验证，阻止非法用户访问系统。

总体来讲，系统主要是基于微信小程序实现图书共享功能，用户可以自己编写书籍发表到社区供大家一起讨论，这样来分享广大人民的知识，也可以一起讨论各类书籍，实现文化共享。

【关键字】共享书城，微信小程序，Spring Boot，MySQL，Mybatis-plus

The Design And Implementation of Sharing Bookstore Based on Wechat Applet

Abstract

This topic is designed under the political background of cultural confidence proposed in China. The whole system is designed separately from the front end. The front end is based on the wechat applet client and the management page based on layui. The back end is built based on spring boot and mybatis plus. The background business logic is independently implemented, and the data source is stored through mysql. Some simple API calls will also be made in wechat applet, such as Baidu map and Baidu weather. The API interface provided by Baidu will be used to realize map query and weather query functions.

There is also sharing. The sharing community and library designed by the system are tailor-made for users. Users can access them for free. Powerful fuzzy query can quickly search the books they want to see. The whole process is based on the component development of wechat applet and sending requests through wechat interface, You can search your favorite books by rendering the requested data in real time. Mybatis plus, which is built in the background, can also be used for rapid annotation development, and the integrated spring security can also be used for post administrator authority verification to prevent illegal users from accessing the system.

Generally speaking, the system is mainly based on wechat small program to realize the function of book sharing. Users can write their own books and publish them to the community for discussion. In this way, they can enjoy the knowledge of the broad masses of people and discuss all kinds of books together to realize cultural sharing.

[Keywords] Sharing Bookstore, WeChat Mini Program，Spring Boot，MySQL，Mybatis-plus

# 1 [绪论](#_Toc20937)

## 研究目的及意义

就当下社会现状，大部分的中国人民忙于各种各样的事情，有自己的家庭琐事的，有社会工作的，有国家建设的，大部分的人闲停下来之后会玩手机，而玩手机的大部分群里面玩娱乐游戏者众多，看剧者女性较多，上网看小说者不在少数，我们倾心于各种各样的社会娱乐模式，也沉迷于各种各样的生活固定模式，大部分的人生活方式极其相似，毫无乐趣可言，我研究本课题主要是为了给我们家庭或者社会一种新的生活方式，共享书城，细细研究每一本书，让大家知晓其中的人生哲理，服务我们人民本身，给我们的生活也新增一份新奇的玩意，也让更多文化程度不高的人提高自己的文化水平，相信一日又一日，全社会的共同努力下，全名文化水平均会有很大提高，中国社会的文化素质也会有极大提高，作为一位大学生，能做的真的很少，但是在这个意识层面上，每位中国人民都应该尽自己的一份绵薄之力，鼓动全名的文化学习之风。台湾有一位教授就说：大陆没有文化。复旦大学哲学系教授王德峰十分赞同，我们大陆在各个方面都缺少各种各样的文化洗礼，社会青少年自杀率不断增高，忧郁症患者也愈来愈多，社会迎来了有史以来最大的症状，文化的遗忘，我们每个人都会受到各种各样的影响，争名夺利，攀比之风也愈来愈烈，社会阶层，贫富差距，让人越来越难以置信的文化危机在不知不觉中发生，贩毒犯罪，毒害谋杀之行为不断上演，为了让我们的民族重新拾起伟大的民族文化和精神，让中国人民受到更优秀的传统文化，是我研究本课题的基本出发点。最近一考研学生由于考研失利而弃考，也有人因此结束自己的生命，有的父母甚至逼死自己的亲生女儿，诸此种种，都是我们文化的缺失，教育的缺失。

就文化上而言，本课题有一定的成效，会带来一股新新文化之风，全民都知晓中华上下五千年，知先秦，知两汉，知三国，知隋唐，知五代十国，知魏晋，知元明清，知近代，知如今之中国，以已知推未知，推动社会的进步，丰富国力，为中华民族之崛起之风会再度崛起，从民国的救国救民之风也会转变为富国富民之风，时代之大趋势，民族之大崛起，一辈又一辈的青年将砥砺前行，不再会有而今庸俗的糜烂之风气，把美好的看成无法承受的，高中生，大学生，研究生，社会群众的思维将会有极大提高，不再拘泥于眼前的蝇头小利，也不会相互算计，这一天会很远，但我们应当携手共行，推动社会之进步。通过此课题，会对青少年的基础教育会有一定的促进作用，让更多的少年找回真正的自我，心中怀有民族精神，不会浑浑噩噩，度日如年，甚至有结束自己的性命，导师威胁私下的学生，没有了礼，没有了仁。通过此课题，让更多的人了解到中国文化的渊博以及中国文化的伟大。学习文化成就自我，打造自我。就社会经济上而言，人们学习到文化，提高自己的精神追求，促进市场经济的运转，也提高全民的物质精神水平，让中国变得更加繁荣昌盛，人民更加的幸福，与此同时，促进各方势力兴办教育机构，改造全民对中国文化的学习，而不是如今仅仅是对数理化政史地的基本教育，缺少了真正的文化教育，基础教育，如此一来，全民文化素养会有极大提升，中国身为礼仪之邦，也会更加的以身作则，向全世界宣传咱们的优良传统，与世界人民一同进步。就政治上而言，相应我们的文化自信，让更多的人民学习文化，提高文化素养，而不会整天只为了眼前的娱乐，安稳，愚昧无知而心满意足，改造全名的文化素养，解决人如何安慰自己的精神世界，同时也会让中国人民身处高位者能够自知，不牟利，不谋情，全名精神自由。

在技术上，目前主流的Java框架Spring、SpringMVC、Spring Boot也正在如火如荼的发展，虽然学习了Javaweb、Hibernate、struts框架，相比而言，Spring Boot的简化开发简直是Java开发的不二选择，使用这款框架可以快速构建开发环境，开发过程利用三层架构和maven依赖以及相关的自动配置就可以实现基本的业务逻辑。微信小程序也是目前主流技术之一，目前的小程序也是十分火爆，是前端学习者必学的内容。组件化开发、云开发、api调用、微信小游戏开发以及公众号开发都跟小程序开发有一定的联系。作为软件工程的学习者，研究这些技术十分有必要。

## 国内外研究现状

国内现状：大学生沉迷于游戏、影视剧、小说等等各种网络娱资源，而对于实体书，图书的利用并未充分，对于图书的处理大部分也是废弃，造成巨大浪费；各类社会人士也是疲于谋生更是无从闲暇来阅读，节奏越来越快的社会给了我们新一代越来越大的压力，我们甚至放弃了读书的念头，主张读书无用论；小程序的发展也是近些年来突飞猛进的一个行业，基于小程序的网上共享书城着实太少，不是以商业为目的就是内容太过狭隘，不够广大中国人民阅读，在我们中国，如此深厚的文化也就无法全国性普及，学问高者也就寥寥可数；时代之风引领着我们不断前行，小程序的魅力在于快速传播，简易使用，无需安装，可以在我们生活中留下一道美丽的彩虹，微信已经普及，甚至连有的老人也会用，如果再把小程序传播下去，把共享书城这道光传播下去，将对我们如今文化匮乏有很大的提升；很大大学生也是苦于资源匮乏，校园的资源分配问题有其道理，但大学生的困窘却无人理会，让学生自己去寻找，有的真的无异于大海捞针，如此这般，人才怎能不匮乏，又怎会有建国后那般人才济济之盛况；基于微信小程序的共享书城在某种程度上说，是亟待有志之士开发，而我作为本科大学生，深知自己技术方面不成熟，还需要成长，这个小程序对于我们提高文化素养，了解人类历史应该会有极大帮助；2018年，共享书店出现一年了。从去年7月安徽新华推出首家共享书店迄今，尝试做共享书店者越来越多。乘着共享经济的东风，这种寻求突破传统经营模式困境的新尝试一直备受关注，也备受争议。业内有人认为它只是以“共享”的噱头做营销，也有人说此举是引发传统书店行业的变革。而利用共享书店来做营销完全背弃了共享书店的出发点；安慰合肥三孝口新华书店全24小时营业，在工作日座位都已经占满，成了读书爱好者的读书胜地。

国外现状：目前国外还没有真正意义上的类似于这种的网上共享书城，由于网络盛行，在各个方面，实体书店的经营模式没有网上来的快，便捷，国外的很多网上书店都有很好的名声，诸如eBay，Depository等等网上书店都是供消费者买的，但他们的现状是营销，而我们的出发点在于传播我们的传统文化，所以国外的这种理念目前我是没有发现的，他们在学校里会体验到免费的书籍，免费的图书馆，但是他们的这种教育模式是竞争式教育，并非全然的自觉式教育。

综合来说，我们国内的这种文化自信和文化自觉性的呼声很高，这也要求我们更多的人为之奋斗，为中华文化在中国人所在的每一片土地都留下我们中国人的风采，我们要文化自信。

## 论文研究内容

本课题主要是针对社会的广大群众文化素质进行反思的一次研究，中国广大群众出现弊端的情况屡屡出现，我们应该防患于未然，进行文化普及，特别是传统文化，应从小就普及这种思想，构建更美好的大中国。当然，作为大学生，是以自我为出发点，反观全人类，个人观点难免过于狭隘，所以还需要各路人士共同探讨和解决这类弊端，社会多元化，国家多元化，文化多元化，导致各种各样的人相处在一起，各执其主张，却没能归结于一点，有的宗教冲突，有的派别冲突，有的利益冲突，千奇百怪之事屡屡出现，中国还算太平，有共产主义的信仰支撑着我们中国人民，所以在文化面前需要统一信仰，全国人名，上下一心，众志成城，为中国之未来的繁荣富强而尽一份心出一份力。基于微信小程序的共享书城针对全国人民文化素养的进一步提升，还有提高自身的文化储备，为社会打造知史之才，也为苦于寻找书籍的给界人士提供门道，久而久之，广大人民群众都能知道国家之不易，人民之不易，也能知道儒家文化的深邃，道教的势与术， 诸如此类的历史文化精髓。

在前端方面，通过三大主页的设计，文章社区、共享社区、个人信息三大页面，完成前端所需要基本页面，还有美化工作，运用以前学习的HTML和CSS来进行wxml和wxss的开发工作，还有一些基础的配置文件，丰富页面的整体内容，同时还有全局配置和局部配置相互协调，促使前端工作的顺利进行。在这里，会使用到各种组件来完成页面开发，同时还会发送一些外部请求以达到一些动态效果，包括日历等，也能够转发，实现多功能。Swiper组件实现动态的轮播图效果，给读者有一个很好的体验效果，当然还会有一些小的动态效果增加页面的炫酷。在后端开发中，遇到的问题会更多一些，我决定用Java作为后端开发语言，通过三层架构的思想实现基本的一个请求-->处理-->响应的基本流程，在其中我们会编写各种工具类来处理我们的数据，也会编写各种服务类来处理我们的数据对象，还会有各种操作类，实现数据操作，而在此之前，我会做大量的数据库设计，分析好所有的需求，以备数据库设计的合理性，也可以通过模糊查询搜索一些基本的文章，让用户体验效变得更好。还有服务器的配置，需要申请一个服务器，来接收小程序发出的各种请求，然后再处理和响应。

利用微信小程序的三层架构，视图层、逻辑层和数据层。视图层和逻辑层分离，通过数据驱动，事件交互，不直接操作DOM，视图层负责渲染页面结构，逻辑层负责逻辑处理、数据请求、接口调用等，视图层与逻辑层通过数据和事件进行通信，逻辑层提供数据给视图层，视图层通过绑定/捕获事件发起交互让逻辑层处理。视图使用WebView渲染，JS由JSCore(IOS)/X5(Android)/nmjs(DevTool)渲染解析。JSBridge下架起上层开发与Native（系统层）的桥梁，使得[小程序](http://www.wxapp-union.com/" \t "https://www.cnblogs.com/yangpeng0517/p/_blank)可通过API使用原生的功能，且部分组件为原生组件实现，从而有良好体验。同时进行组件式开发，使整体性更加紧凑。

## 1.4 论文组织结构

本论文最终实现基本的文化书籍阅读功能，整体上系统开发基于微信小程序和Spring Boot开发的共享书城的设计与实现，实现基本功能，各项指标合格，最终写成论文和报告的形式。

在系统开发过程中，遇到了各种问题，环境搭建，服务器配置，前端开发，后端开发等等，遇到问题通过网上查询或询问同学导师都得到了合理解决，利用团队协作，向前辈学习，自学各种视频资源，最终实现论文的整体效果和系统开发的大体功能。整个系统逻辑紧密，本人在开发系统过程中，做了以下工作：

1. 整个系统的需求分析与可行性设计

系统需要实现简单的注册和的登陆、个人用户信息的变更、书库的访问、发表文章等等，这些基本功能与数据库设计和整体上的业务逻辑紧密相关。

1. 整个系统的整体思路把握

首先是搭建好基本的开发矿建，接着进行页面开发，最后做业务处理，最后将前端交互和后端业务逻辑起来，可以通过请求的方式实现。

1. 控制流程图的设计

这一部分这需要借助相关的画图工具进行设计，主要是根据业务逻辑的处理过程进行的。

1. 系统的功能测试

这一部分则是测试基本的开发效果，能否实现这部分的功能。

1. 总结和致谢

主要是对整个项目进行全局性把握，总结自己开发过程中学习到的内容。

# 2 相关理论与技术

## 2.1 微信小程序的运行原理

下载开发工具，了解了微信小程序的基本原理。Wxml格式的文件相当于网页浏览页面的HTML页面，进行基本页面的开发，wxss则是起到美化页面，页面布局等作用，相当也web开发的css文件。json是一种数据格式，并不是编程语言，用来配置静态页面的角色，在小程序里面有全局配置和局部配置，小程序的全局配置文件则是app.json文件用来配置整个页面的全局角色配置，头部颜色，tabbar的设置，头部导航栏设置，pages字段 —— 用于描述当前小程序所有页面路径，这是为了让微信客户端知道当前你的小程序页面定义在哪个目录。window字段用来定义小程序所有页面的顶部背景颜色，文字颜色定义等。Js文件则用于逻辑交互，一个服务仅仅只有界面展示是不够的，还需要和用户做交互：响应用户的点击、获取用户的位置等等。在小程序里边，我们就通过编写 JS 脚本文件来处理用户的操作。这就是基本的四个文件，用来配置和开发每一个页面。

小程序开发的环境分为页面渲染层和逻辑交互层，wxml和wxss扮演渲染层的角色，进行页面布局开发，js扮演逻辑交互层的角色，进行页面数据和各种事件的处理。

小程序的渲染层和逻辑层分别由2个线程管理：渲染层的界面使用了WebView 进行渲染；逻辑层采用JsCore线程运行JS脚本。一个小程序存在多个界面，所以渲染层存在多个WebView线程，这两个线程的通信会经由微信客户端（下文中也会采用Native来代指微信客户端）做中转，逻辑层发送网络请求也经由Native转发，小程序的通信模型下图所示。

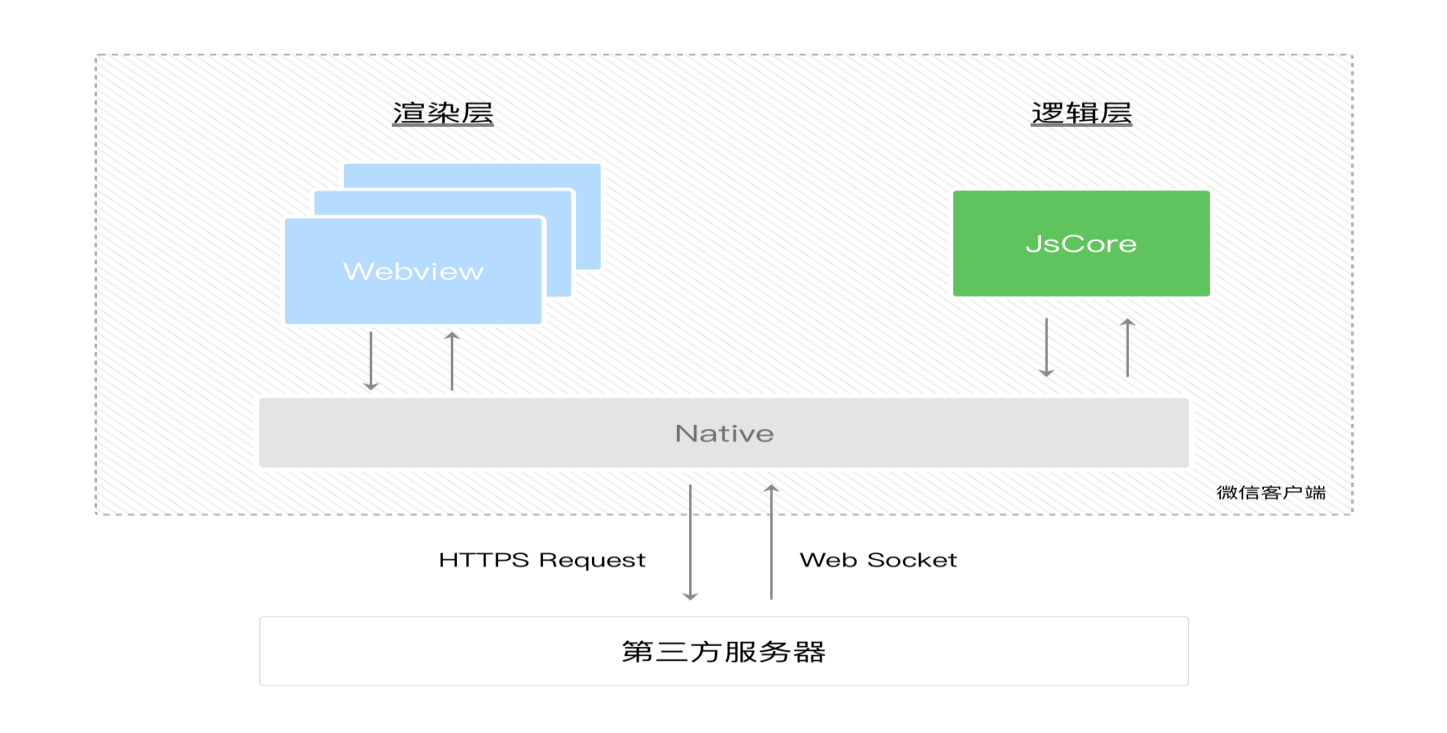


图2.1 微信小程序业务逻辑处理原理

在app.json文件的pages字段可以配置当前小程序的所有页面路径，整个小程序只有一个应用实例，而这个应用实例是所有页面所共享的，而微信小程序很强大的功能在于组件化开发和API的调用上面，通过调用官方API可以实现地图，微信支付等众多功能。

### 2.1.1 微信小程序的配置

微信小程序配置主要分为以下几个部分进行讲解，其一，全部配置，在app.json文件中对微信小程序进行全局配置，pages字段可以配置yye页面文件的路径，在windows字段可以配置头部标题颜色和背景颜色配置，networkTimeout可以设置网络请求超时的相关配置，还可以设置多个tabbar，来做全局配置；其二，页面配置每一个页面都会对应有一个json文件，来做当前页面的配置，页面中配置会覆盖全局配置中的信息，实现自定义页面配置；其三， sitemap配置小程序根目录下的 sitemap.json 文件用来配置小程序及其页面是否允许被微信索引。

### 2.1.2 微信小程序的自定义组件

微信小程序的自定义组件分为以下几个部分讲解，其一，视图容器，Cover-view可覆盖的原生组件包括 [map](https://developers.weixin.qq.com/miniprogram/dev/component/map.html)、[video](https://developers.weixin.qq.com/miniprogram/dev/component/video.html)、[canvas](https://developers.weixin.qq.com/miniprogram/dev/component/canvas.html)、[camera](https://developers.weixin.qq.com/miniprogram/dev/component/camera.html)、[live-player](https://developers.weixin.qq.com/miniprogram/dev/component/live-player.html)、[live-pusher](https://developers.weixin.qq.com/miniprogram/dev/component/live-pusher.html)只支持嵌套 [cover-view](https://developers.weixin.qq.com/miniprogram/dev/component/cover-view.html)、[cover-image](https://developers.weixin.qq.com/miniprogram/dev/component/cover-image.html)，可在 [cover-view](https://developers.weixin.qq.com/miniprogram/dev/component/cover-view.html) 中使用 [button](https://developers.weixin.qq.com/miniprogram/dev/component/button.html)。组件属性的长度单位默认为px，[2.4.0](https://developers.weixin.qq.com/miniprogram/dev/framework/compatibility.html)起支持传入单位(rpx/px)。Movable-view表示可移动的视图组件，page-container表示页面容器。Swiper组件表示滑块试图容器，里面只能放swiper-itemz组件。View组件是最常见的视图容器；其二，基础内容，icon组件表示图标组件，在icon里面的type属性值可以选择以下有效值——success, success\_no\_circle, info, warn, waiting, cancel, download, search, clear，来表示不同的组件图标。Progress表示进度组件，rich-text表示富文本组件，text表示普通文本组件；其三，表单组件，button、checkbox、checkbox-group、eeditor、form、input、keyboard-accessory、label、picker、picker-view、picker-view-column、radio、radio-group、slider、switch、textarea。这些表单组件可以用来做登录功能，各种进度加载问题等等。除此之外，导航组件、媒体组件、地图、画布、开放能力、原生组件等等组件。组件化开发让整个小程序开发变得十分简易，而且页特别容易上手。

## 2.2 Spring Boot后台开发框架介绍

### 2.2.1 Spring全家桶介绍

Spring框架是目前最集成简单地框架，十分容易上手，2002年诞生以来一直备受开发者热爱，它包括SpringMVC、SpringBoot、Spring Cloud、Spring Cloud Dataflow等解决方案。大家都称之为：Spring 全家桶。

Spring framework（Spring框架）对比主流框架：

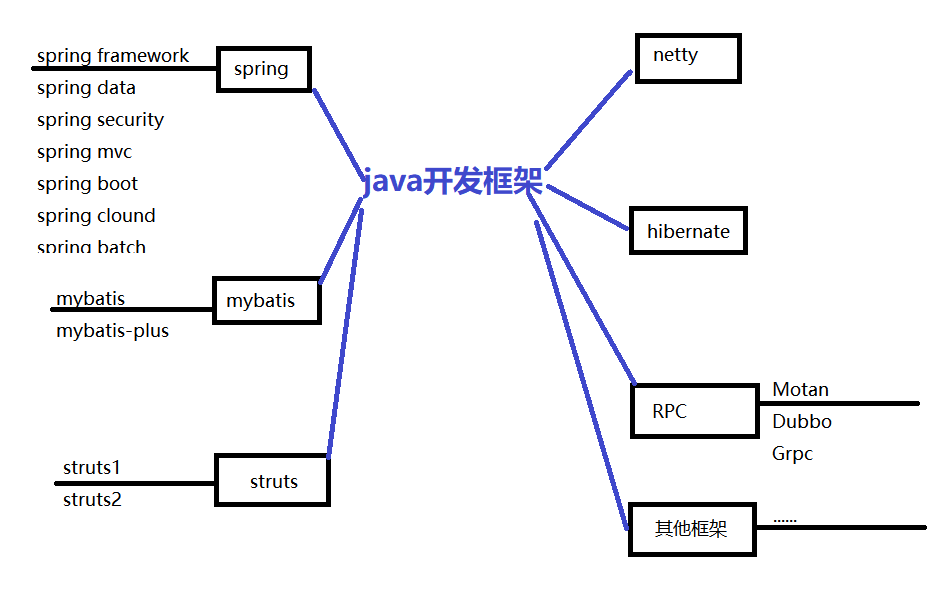


图2.2 Java主流框架介绍

目前主流的框架有很多，spring boot是目前最流行的框架，这款框架在简化配置上有很大的作用，让我们Java相关的开发人员大大提高了开发效率。Spring security在我们做权限验证时起到了很大作用，也能很好的搭建我们后台的安全框架。同时也可以整合一些持久层框架（mybatis/mybatis-plus/hibernate）做我们的项目开发。下图是有关于Spring Boot的一些简单功能，在官方网站我们可以看到很多相关的功能，如微服务、批处理、响应式编程，云开发等等都在我们开发过程中有很大的效果。

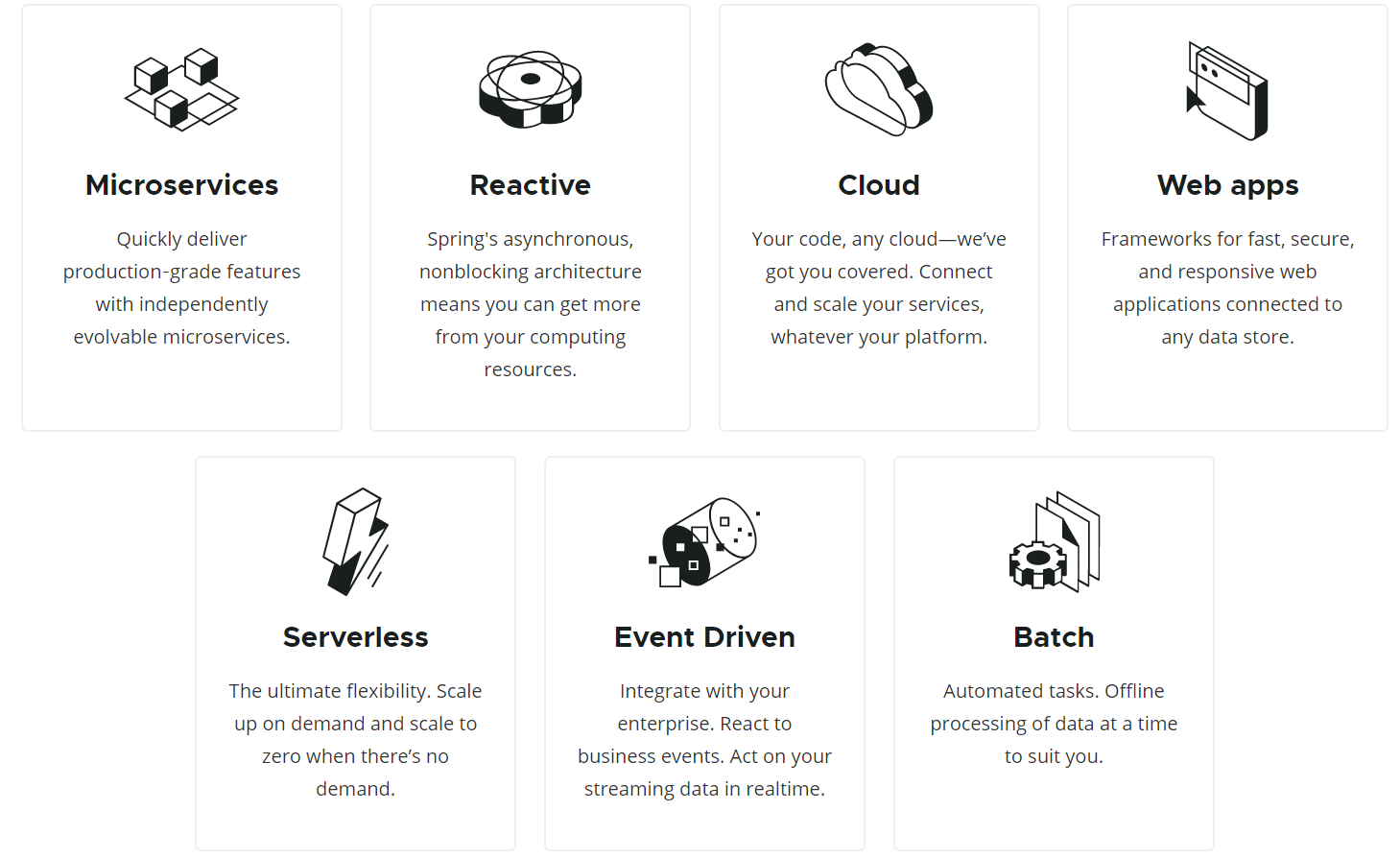


图2.3 spring的功能图[1]（来自spring官方）

Spring可以用来做微服务架构，也可以做响应式编程、云开发、web开发、时间批处理、事件驱动等等众多功能，spring全家桶让Java开发之路更加顺畅和简易，也在我们开发生活中平添一份靓丽的风景线，注解开发，响应式编程开发让我们的更加容易地掌握spring地运行原理。

### 2.2.2 Spring Boot框架介绍

Spring Boot可以创建独立的，基于应用地生产级地spring程序，我们不需要进行过多的配置就可以直接运行，它有以下十分重要的特点，也是我们为什么选择Spring Boot进行开发的重要原因。

其一，Spring Boot能够创建独立的应用；其二，Spring Boot内嵌tomacat、jetty、undertow等web服务器，这样我们就无需配置相关的war文件和相关的web容器配置了，大大简化了我们的开发工作；其三，Spring Boot提供在pom.xml配置相关的启动类和相关依赖能够进行简化配置；其四，Spring Boot的自动配置和第三方库十分丰富；其五，Spring Boot提供生产准备功能，如度量、运行状况检查和外部化配置；其六，Spring Boot没有代码生成，也不需要xml文件配置。Spring Boot面向企业级进行开发，通过IOC容器注入和面向切面编程，大大降低了程序代码之间的耦合度，而且可以简单地整合各种主力框架，如mybatis,hibernate,lombok,elasticsearch等等主流框架。

### 2.2.3 Spring framework框架介绍

spring本质上相当于一个工厂，可以用来创建和实例化对象，他的核心技术Ioc和Aop，Ioc控制反转技术，就是在底层new一个实例对象，它可以降低代码之间的耦合度，提高开发效率，通过工厂模式和xml配置文件按实现基本的对象创建，低沉公园里就是一个对象工厂。Aop面向切面编程，把各个业务分离开发，降低业务逻辑之间的耦合度，可以实时地增加或减少功能，便于开发和维护。

### 2.2.4 Spring Security框架介绍

Spring Security是一个高度安全地用于保护Spring框架开发地安全框架，用于身份验证和身份权限校验，能够对Spring框架起到很好的保护作用。不过Spring Security依赖于很多组件，是一款重量级框架，使用十分简单，只需要引入官方的依赖，然后编写验证的业务逻辑，也就是重写UserDetailsService方法，实现后端安全校验的工作，对比Shiro来说，操作简单，十分容易上手，Shiro通过SecurityManager来管理内部组件实例，通过这个组件来提供安全框架的基本服务。

### 2.2.5 Spring MVC框架介绍

MVC：模型，视图，控制器，springmvc通过前端控制器操作整个流程，从前端到后端数据处理的整个过程都会受到前端控制器的控制。前端通过发送请求给servlet，servlet通过控制service层进行相关的业务处理然后把处理结果返回给servlet，而在service层进行业务处理，同时调用dao层进行相关的数据库操作把数据返回给service,servlet把得到的数据返回给用户。

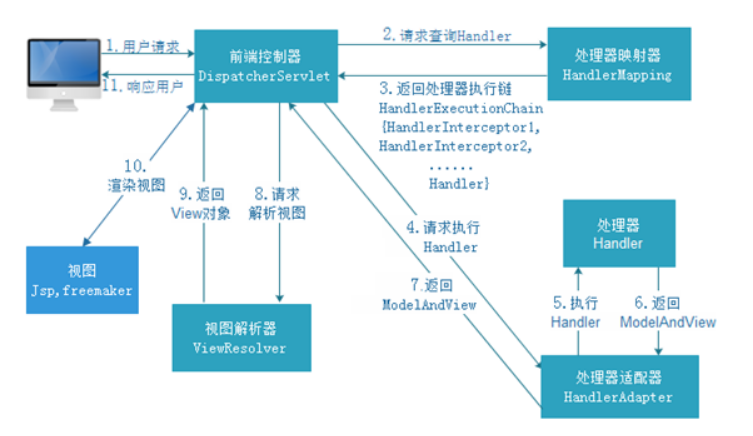


图2.4 Spring MVC工作原理图[2]

Springmvc简单易学，轻量级，而且功能强大，有Restful编程、数据校验、与Spring兼容性好，无缝连接，通过前端控制器DispatchServlet调用HandleMapping和HandleAdapter进行处理前端发送的请求，然后利用controller层进行业务处理，在ModelView中进行相关处理后返回一个view给DispatcherServlet,DispatcherServlet通过调用ViewResolver进行视图解析，最终将得到的页面返回给用户，完成用户的一次请求的处理。

## 2.3 Mybatis-plus持久层开发框架

Mybatis-plus是mybatis的增强版，它是为简化开发而打造的，大大提高了我们的开发效率，支持大部分数据库，无侵入，损耗小，内置分页插件，内置代码生成器，很多功能都为我们的开发提供了很大的便利。在这个框架下，只需要在application.yml配置数据库和服务器配置相关信息，然后编写实体类和Mapper接口，最后进行注解扫面就可以进行使用相关的CRUD操作。

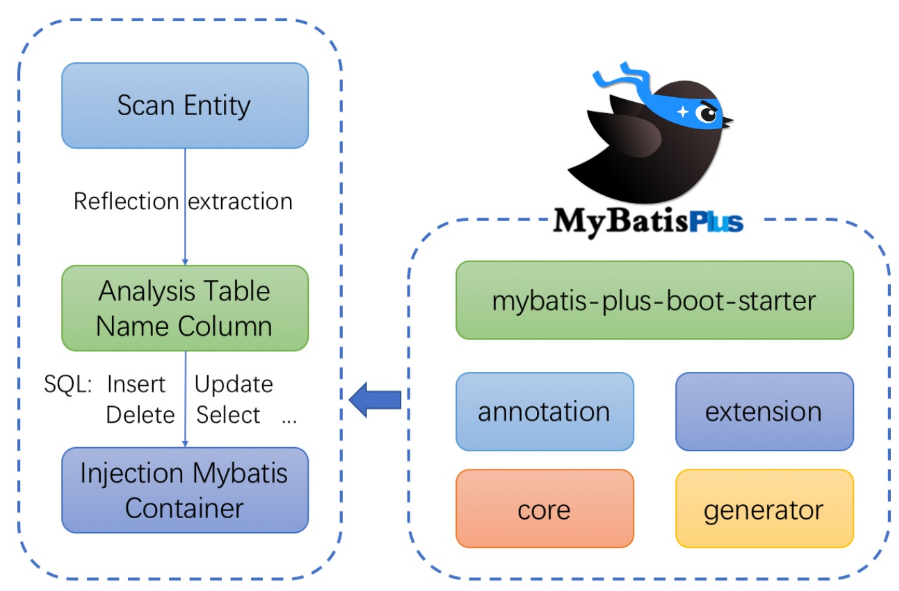


图2.5 MybatisPlus基本介绍图[3]

使用mybatis-plus，不仅可以进行注解开发，而且还内置封装了很多方法，只要我们让mapper接口继承BaseMapper，我们就可以使用里面的所有方法，例如查询表中所有数据或某一条数据、根据某个字段查询、修改、删除等等操作都可以直接调用，让我们的开发效率大大提高，再也不要编写一些简单常用却又繁琐的配置文件进行SQL语句的编写，当然我们也可以整合mybaits做自己复杂的数据库查询工作，这个则多用于模糊查询上。

## 2.4 Layui前端开发框架

Layui框架是一款开源的前端UI框架，用很多开箱即用的Web UI组件。很适合用来搭建后台开发框架的前端模板，有很先进的layer弹出层便于与用户进行上市的交互，内置form表单可以很快捷的进行相应的表单的操作，数据表格用户页面渲染也是十分简易，容易操作，而且layui还有内置的分页操作，通过laypage获取服务端的相应信息即可渲染，操作简便快捷，可以大大简化开发的流程。

就LayUI本身而言，简单，轻量级，十分便于后端开发，在服务端上的渲染效果也很容易操作很部署，不过前端会操作DOM元素，总体来说还是十分适合于后端开发者使用Layui进行后端布局的。

## 2.5 Thymeleaf模板引擎

Thymeleaf是一个流行的模板引擎，该模板引擎采用Java语言开发，模板引擎是一个技术名词，是跨领域跨平台的概念，在Java语言体系下有模板引擎，在C#、PHP语言体系下也有模板引擎。除了thymeleaf之外还有Velocity、FreeMarker等模板引擎，功能类似。

Thymeleaf是一种良好的模板引擎用于浏览器解析页面，特别是现在流行的html5页面，可以用作静态建模。你可以使用它创建经过验证的XML与HTML模板。使用thymeleaf创建的html模板可以在浏览器里面直接打开（展示静态数据），这有利于前后端分离。

## 2.6 ajax异步传输

Ajax异步传输，可以及时的进行响应页面的变化和用户的操作，无需等待的用户体验，给开发者和用户都带来了福利，开发者易于调试，用户体验十分好。Ajax的即时渲染效果很强，让我们在和服务器端交换数据的时候不用重新刷新页面，Asynchronous JavaScript and XML，就是异步的JavaScript和XML，现在使用Ajax则只要要引入jQuery就可以直接调用相关的GET和POST方法进行请求服务器端的数据，Ajax是我们前端开发人员的必备神器，他为我们数据的获取起到了重要作用。

## 2.7 json数据格式统一化

利用json类解析数据，很契合的解决了前端后数据对象的解析问题，Java对象可以转换成json对象，前后端的数据联系也就更加密切了，开发也变得简易了许多，数据交互操作也十分容易了。[JavaScript](https://baike.baidu.com/item/JavaScript" \t "https://baike.baidu.com/item/JSON/_blank) Object Notation是一种轻量级的数据交换格式，易于我们开发人员解析和生成，而且有效地提升了数据交换格式，总的来说，json就是一种数据解析格式，利用一些工具能够把一些对象转化成json字符串，然后通过json转化格式转化成json对象，在这个过程中，数据的传输和封装变得更加安全和简易，也便于开发者很容易的获取数据和渲染数据。

# 3 系统需求分析与设计

## 3.1 共享书城的需求分析

网上共享书城主要是秉持着为广大人民的读书需求所开发，是为了满足读书交流的需求，并在此基础上实现其它的一些基本功能。参考 其他的类似系统开发，需求分析是作为系统开发的第一步，至关重要，基于此，我准备开发的系统有四大模块，分别进行开发，整个系统的四大模块包括个人信息模块、共享社区模块、文章发表模块和共享社区模块。个人信息主要是实现一些基本的个人信息操作，像头像、名称和个性签名之类的操作；共享社区模块属于一个交流社区，大家可以分享书籍交流的经验和感想，并作分享；文章发表模块则是用于用户自己发表自己的文章，交给管理员审核后发表到社区；最后的消息管理模块则是属于待开发的，整个项目由于时间问题，要实现三大模块功能，最后一个功能，待后期完善。整个项目需求分析阶段主要是做功能需求分析，数据库逻辑设计，便于后期开发。

### 3.1.1 个人信息模块

在个人信息模块，用户可以写文章，并对自己写过的文章进行相应的操作，这些文章会随机的发表到社区，供大家交流，同时也可以分享自己读过的文章和书籍，个人用户也可以收藏自己喜欢的书籍加入到自己的收藏区，个人用户在自己的个人中心可以设置自己喜欢的主题样式，同时用户可以在书籍库中访问自己喜欢的书籍并阅读。当然用户也可以对个人的基本信息进行修改操作。

相关操作如图：图3.1

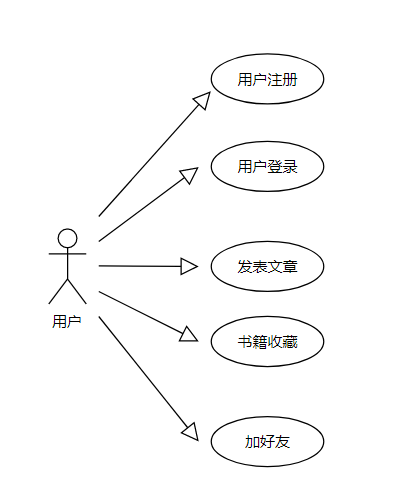


图3.1 用户操作功能图

用户可以注册自己的账号进行登录，登陆后会进入个人主页，在个人主页可以设置个人信息，也可以访问数据，去社区浏览文章

### 3.1.2 共享社区模块

社区模块主要是对用户发表的文章进行展示，并通过渲染技术展现给用户。用户也可以通过模糊查询的检索功能搜索自己喜欢的书籍。实现技术则是根据wx.request发送get请求，然后将返回的数据利用微信js文件的data，然后在wxml中即时渲染即可。

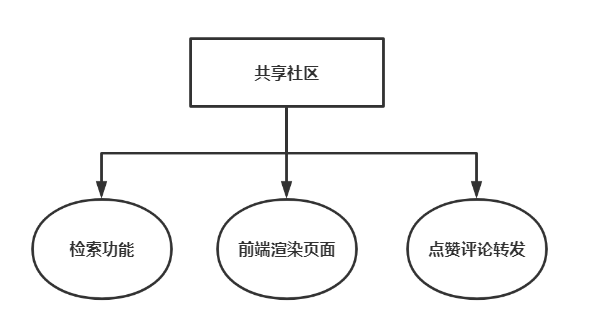


图3.2 社区业务逻辑图

在共享社区，会通过微信渲染技术展示用户新发表的书籍，在这里用户可以浏览相关用户信息，也可以加好友。搜索功能时根据模糊查询进行的，用户通过输入关键字，页面就会通过关键字搜寻相关的数据然后即时的渲染在前端页面上，加好友则需要点击作者到对应用户的主页上发送好友请求，等待对方答复即可。点赞功能则绑定了每一个用户，利用用户id来识别点赞的对象。

### 3.1.3 书籍发表模块

用户可以进行书籍发表、视频发表、图片发表和文章发表。主要是实现书籍发表。通过后台编写地sql语句进行模糊查询发表的书籍。书籍发表和图片发表都是以form表单的形式进行提交的，用户将自己编写的内容发送到服务器，等待服务处理，处理完后发送一个true或者false给前端用户，用户以此来判定自己是否发表成功。

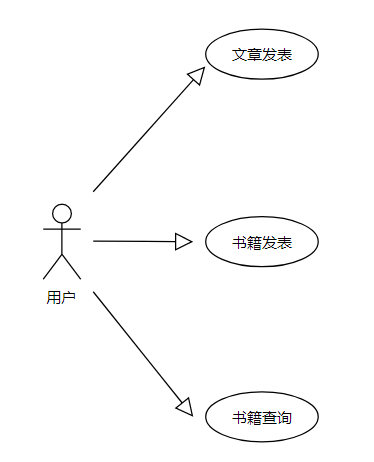


图3.2 用户文章功能图

用户可以在自己登陆后的发表社区进行写文章或者书籍，发表的文章会在自己的发表社区展示，而发表的书籍则会在共享社区中进行展示，同时在共享社区模块，用户也可以查询相关用户发表的书籍进行阅读和访问发表书籍的相应用户。

### 3.1.4 消息管理模块

用于用户之间的交流和沟通，实现了加好友功能。实现地流程是用户通过发送请求给对方用户，对方用户在接收到请求后进行回复，回复同意之后就能加好友，否则就不能加好友。

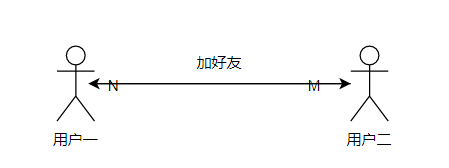


图3.3 好友管理功能图

这一部分则是用户之间相互可以加好友的功能，通过发送请求等待另一方的答复来完成。这个过程的实现是根据一个friends表来实现的，这个表会储存唯一的一对user\_id，他们是好友的话只会有一对这样的值，而发送请求则是通过message来存储，如果用户查询到发送请求则进行处理，否则不做处理。

## 3.2 共享书城的数据库设计

### 3.2.1 数据库ER图设计

ER图的设计主要是根据实体类（如用户类、书籍类等等）和实体之间的关系进行设计的，比如说用户和书库里的书籍有收藏与被收藏的关系，用户与文章有一对多的关系，用户与用户之间有多对多的关系，诸如此类，我们就可以通过主外键进行设计ER图。

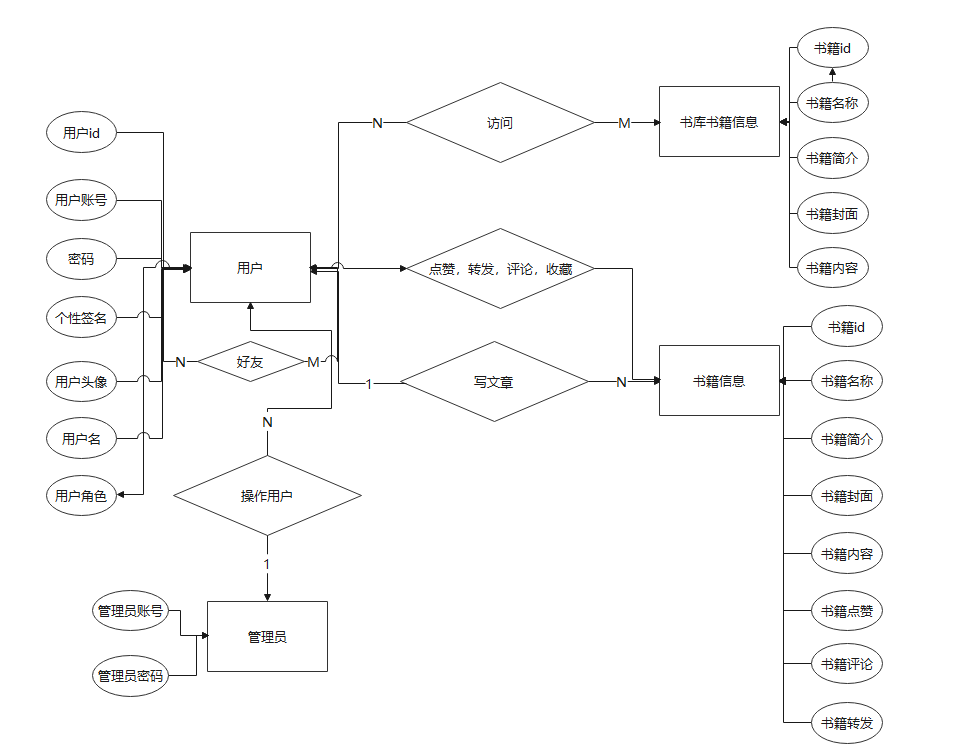


图3.4 ER图设计

在这一部分分析了用户、书籍、书库信息的相关字段以及它们之间的关系，便于后期设计相关的功能，表的设计也是依赖于ER图的设计，每个表的字段对应每个用户的属性，他们之间是一对一的还是一对多的，甚至复杂的有多对多的，都需要分析清楚，这样后期业务逻辑的开发就不会无从下手。

### 3.2.2 基于UML的数据库设计

根据用户表和书库表多对多，收藏功能是他们的关系，需要一张表进行保存relation，也就是stars表。

用户之间可以加好友，每个用户都可以有多个好友，也是多对多，需要一张friends表。

用户可以进行个性签名的发表，每个用户对应多个用户签名，每个个性签名对应一个用户，是一对多的关系，需要一张signatures表。

每个用户可以有多张历史图片，也可以更改自己的图像，需要一张存取图像的一张表，也就是需要一张photos表。

每个用户可以发表文章，需要一张保存文章的表，而且是一对多的，也就是需要一张uarticels表。

每个用户可以发表书籍，需要一张保存书籍的表，而且是一对多的，也就是需要一张ubooks表。

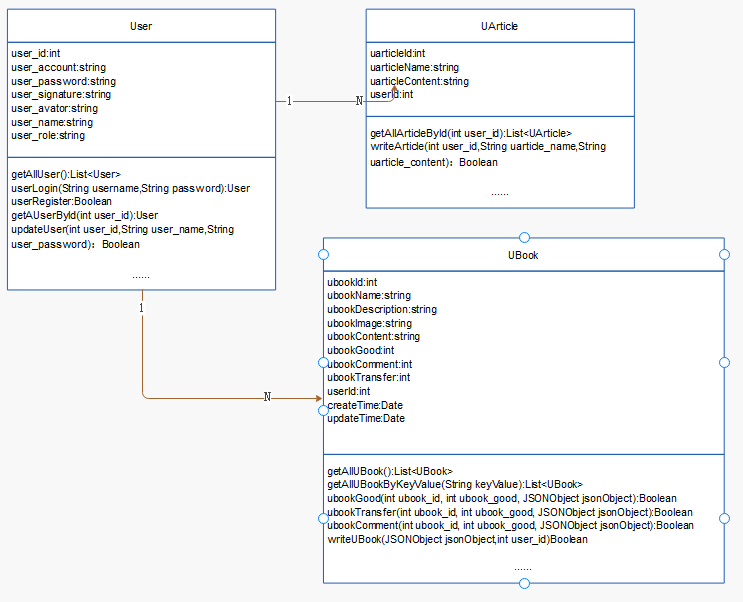


图3.5 UML图设计——用户表和文章书籍表

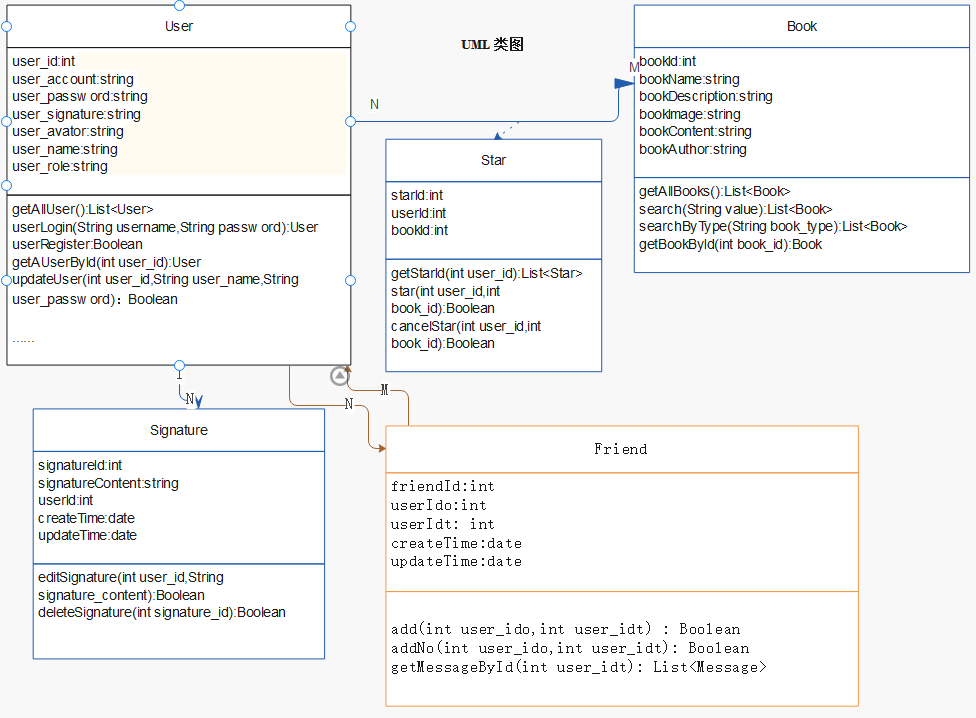


图3.6 UML图设计——用户表和书库表/好友表

类图的设计是为后期后端开发工作做了详细的规划，在后期实现基本功能时编写相对应的方法，能做到后端独立开发，也让后端人员能够专心搞后端。

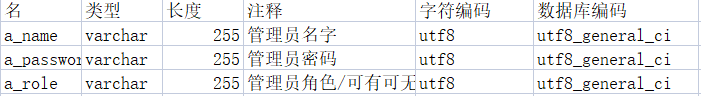
### 3.2.3 数据库表结构的设计与实现

大体上数据库的表有以下这些：管理员表，用户表设计，书籍库表，书籍表，个性签名表，图片表，文章表，好友表。

管理员表：进行项目后台管理者——

在管理员表中主要用于后台管理工作，此表也可以和用户表合在一起，但是由于我做的系统比较小，分开也不会有太大影响，这样也可以在用户表分很多角色，避免发生字段冲突（users表也有admin字段，但它不是后台管理人员）

图3.7 数据库表设计——后台管理员表



在管理员表，包括三个字段，用户密码和角色，整体上是两个字段，用于spring security后台安全框架的校验，具体的实现则是在UserDetailService接口中复写方法进行数据库校验。

用户表：存储进行前台小程序访问的用户——

在用户表中会有很多角色，普通用户，管理员用户（特权用户），微信用户，不同的用户登陆方式不同。

图3.8 数据库表设计——用户信息表

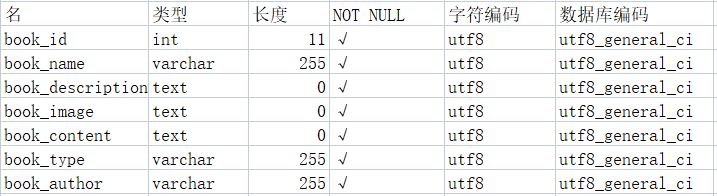


用户表字段有用户账号密码，用于登陆验证，用户签名用于记录当前的用户个性签名，用户头像记录头像地址（服务器地址），用户角色则存储用户的相关权限功能。

库内书籍表：存储共享书库——

库内书籍表是供大家查阅读书的仓库，大家都可以访问，只能进行访问操作。

图3.9 数据库表设计——书库表



书库表是为用户提供的免费书城，用户可以免费访问这里面的资源，这个表存储我们书库的所有数据信息，通过构建这个表的基本信息来展示前端页面所需要的数据，其中book\_type用户存储书籍种类，后期可以根据书籍类别进行查询相关的书籍。

用户书籍表：存储用户发表的书籍——

用户书籍表是一些用户分享书来的书籍，或者是自己写的一写书籍，会展示在社区页面。

图3.10 数据库表设计——用户发表文章表



这个表用于存储用户发表的文章，用户可以在共享社区模块访问这个表的内容，因为这个表会在社区模块进行即时渲染，而且也可以在社区模块通过关键字查询这个表的相关内容。

个性签名表：存储用户发表的个性签名——

个性签名表是用户在个人信息部分可以进行发布自己签名的部分。

图3.11 数据库表设计——用户个性签名表

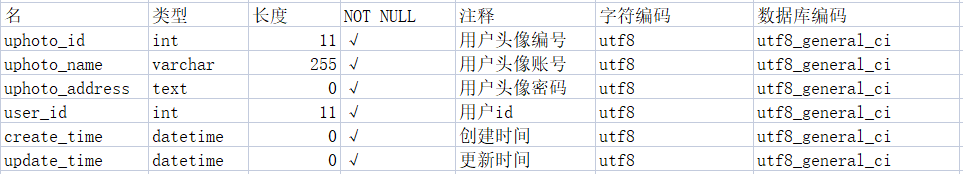


用户在此处存储自己发表的历史签名，后期可以查看，在小程序端也可以进行删除和发表。

图片表：用于存储用户头像——

图片表是存放用户头像地址的表。

图3.12 数据库表设计——用户头像表

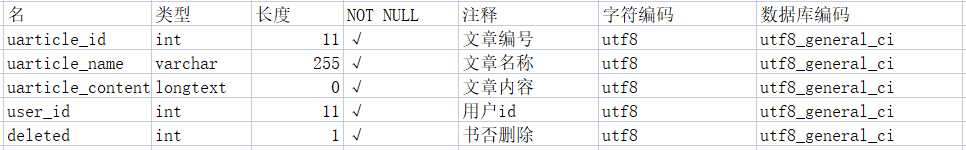


通过user\_id来唯一标识每一张图片，每张图片对应一个user，在此处设置外键，就可以实现用户表到用户头像表的对应关系，后期也可以通过user\_id查询对应用户的历史头像。

文章表：

文章表是用户发表文章的表，会即时的渲染到自己的发表区域。

图3.13 数据库表设计——用户发表书籍表

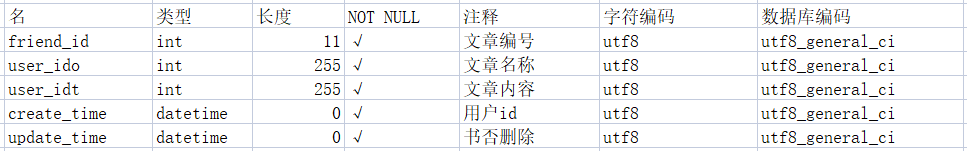


通过user\_id来唯一标识每一篇文章，每篇文章对应一个user，在此处设置外键，就可以实现用户表到用户发表文章表的对应关系，后期也可以通过user\_id查询对应用户发表的文章。

好友表：存储好友——

好友列表，是在对方发送好友请求并同意后记录在册的一张表。

图3.14 数据库表设计——好友表



用户一的id和用户二的id来共同确定一个好友关系，用这个表来记录这种关系，最终达到记录好友列表的功能，后期可以通过根据用户id查询出自己的好友有哪些用户。

根据所画的ER图进行数据库设计，每个实体对应数据库的一张表，实体的属性对应数据库表中的字段，然后根据表与表的关系设计主外键关系。

## 3.3 共享书城的前端设计

前端主要是利用小程序的wxml,wxss,json等四个文件进行开发设计，后台的页面则是根据layui进行搭建。小程序端利用配置文件配置navigationbar和标题以及背景色，wxml开发页面之后，利用wxss进行页面美化，在后台页面搭建过程中，利用框架layui进行搭建后台页面，很容易尽可以搭好基本的前端开发页面。

### 3.3.1 登陆注册页面设计

用户在第一次登陆时，可以去注册，注册完成后自动跳转到登陆页面进行登录。页面背景图是通过设置背景样式从网上获取的一张图片，其中页面的具体实现逻辑是通过wxss进行美化的。



图3.15 注册/登录页面

在这一部分用户可以进行注册然后直接登录到个人主页，注册页面会有用户名和密码验证，符合要求的话就可以注册成功然后跳转到登陆页面进行登录，登陆的时候要正确输入用户名和密码才能登录系统，用户名或者密码错误都会弹出提示框，这部分的实现依赖于button和input以及基本的布局设定。

### 3.3.2 个人信息页面设计

分为两大模块，一部分是个人信息模块，另一部分是功能模块。个人信息模块用于展示个人信息，功能模块进行用户操作，如查看好友列表，搜寻书籍，去修改个人信息页面，查看手机文章等。

在这一部分，用户也可以进行个性签名的设计，也可以查看某地近两天的天气情况。整体上设计是通过盒子样式设计，一层套一层，然后通过margin和padding设计外边距和内边距达到最终效果图如下：

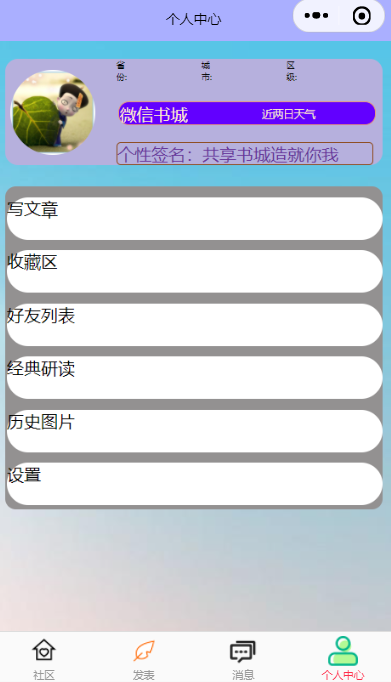


图3.16 个人主页/个人信息修改页面

在这一部分用户可以访问自己主页，进行修改自己的相关信息，也可以去访问收藏的书籍。访问主页主要有写文章、收藏去、经典研读、历史头像以及调用百度天气api查看最近天气。也可以进行个性签名的发表，发表的历史签名也可以去签名列表进行删除操作，最后也可以去修改自己的基本信息。



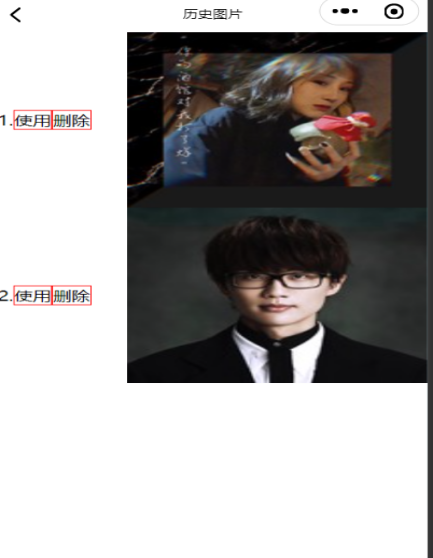


图3.17 历史头像页面和个性签名页面

在这一部分，用户可以查看历史图片和历史个性签名。历史头像这里会记录用户先前用过的头像，在这里用户可以删除自己使用过的图片，也可以重新使用用过的头像，个性签名这边则可以发表一定范围长度的文字，记录自己的个性。

### 3.3.3 共享社区模块页面设计

在这一部分，每一个用户都可以在搜索框进行搜索，页面会即时渲染将搜索出来的结果渲染到页面上，当然用户也可以在自己的发表地方进行发表文章，对应的内容就会及时的渲染到社区页面。用户也可以在自己的个人主要点击功能菜单中的经典阅读就可以搜索相关书籍进行阅读。

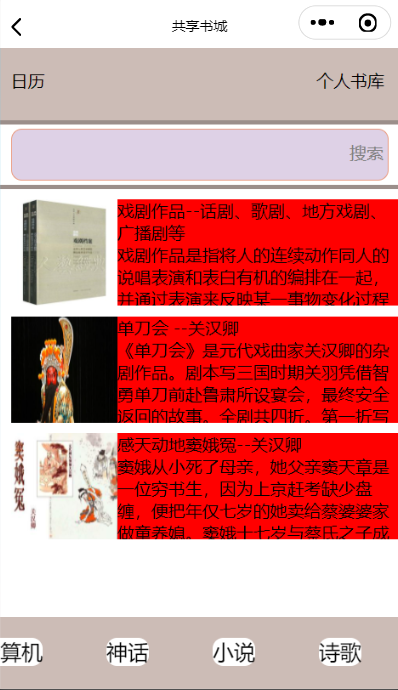
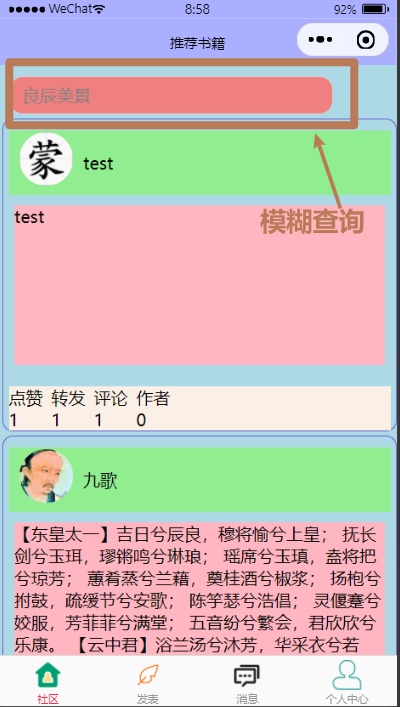


图3.18 社区页面和书库页面

在这一部分，用户可以访问其他用户发表的数据信息，也可以去访问书库资源。书库资源分类有很多，我们可以选择不同的分类查看相应的书籍，也可以通过输入关键字进行查询相关的书籍，然后点击进去阅读自己喜欢的书籍，点击进去之后还有收藏和取消收藏的功能。

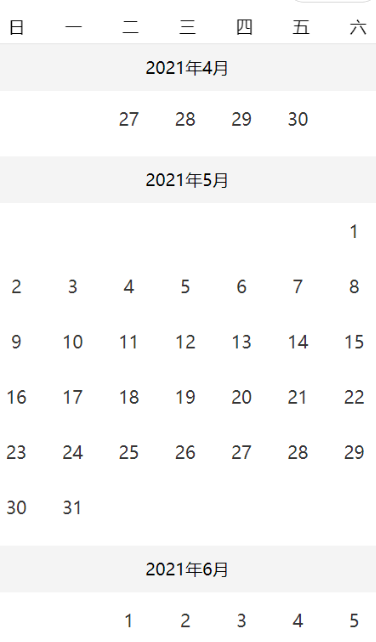


图3.19 日历页面和收藏书籍页面

在这一部分可以查看日历和自己收藏的相关书籍。日历模块是基于页面和js开发的，直接嵌入即可，收藏模块则是通过stars表记录用户id和书籍id，用于映射用户表和书库书籍表多对多的处理，后期用户可以通过自己id查询自己收藏过的书籍，然后进行渲染即可。

### 3.3.4 文章发表模块设计模块

文章发表模块的设计就是通过表单的形式将发表的内容发送到服务器，服务器接收到数据库之后存储到数据库的书籍表中，而发表的文章也会动态的展示在社区模块。

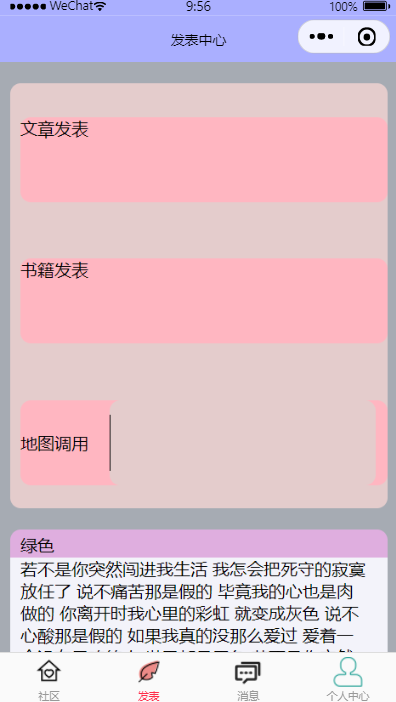


图3.20 文章发表页面

通过输入文章的相关字段，点击发表，服务器端接收到数据并存储到文章表或者用户书籍表，然后返回success给前端，这样用户就可以在自己的文章社区或者共享社区模块查看自己发表的文章或书籍。百度API则是通过调用官方API，把相关字段传给他，然后就会返回一个地图的定位，实现地图的查询功能。



图3.21 百度地图API调用

调用百度地图API。

通过发送请求http://api.map.baidu.com/place/v2/search?query=ATM机&tag=银行&region=北京&output=json&ak=ZGIovf7H68lfID8BWGqkIkbbPGZVYmwO，就会接收到返回的数据，然后根据返回的数据在地图上以图标和文字的形式标明信息。

### 3.3.5 好友模块页面设计

好友模块设计就是在社区浏览的用户可以点击用户进行加好友，如果是本人就直接跳转到自己主页，否则去对方主页，可以发出加好友请求。相应的用户会接收该请求，用户会给出回复，最后加好友功能就完成了。

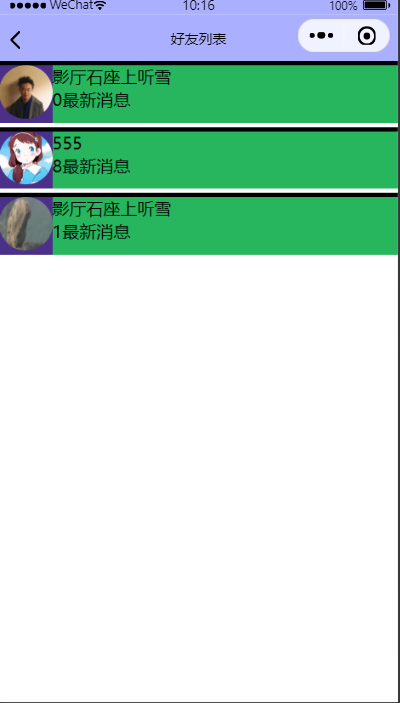
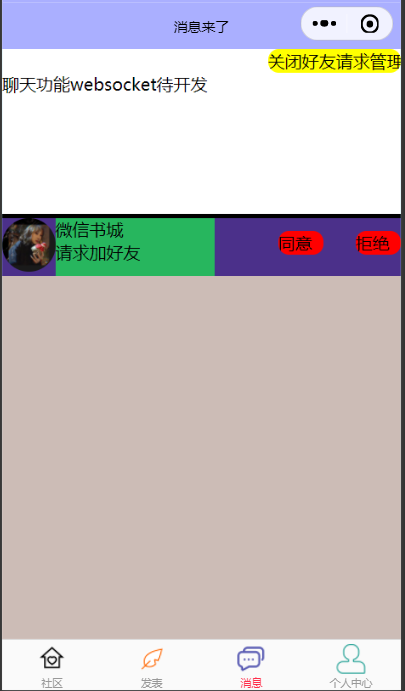


图3.22 加好友相关页面

在这里可以访问其他人的主页，在他的主页可以查看该用户发表过的相关文章，然后给他发出加好友请求。等到对方回复之后，就可以判定是否加好友成功。好友加成功之后就会在自己的好友列表进行渲染。

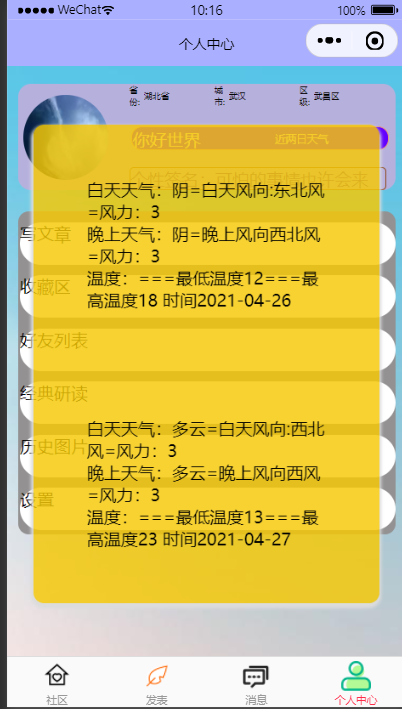


图3.23 调用百度天气的相关API

通过调用百度天气API，就可以获取相关的天气信息，然后在页面上进行渲染就可以了。具体实现逻辑是通过发送两个字段省份和城市给接口，接口会根据对应数据去后台获取对应的json对象，然后返回给前端，接着就可以直接在页面上渲染。

## 3.4 共享书城的后台设计

### 3.4.1 用户管理模块

可以查找相关用户进行相关操作，即可以对用户进行增删改查。输入相关的keyvalue，页面会通过发送Ajax请求将请求回来的数据即时的渲染到页面上，此时用户可以点击相关用户进行对用户的相关操作。



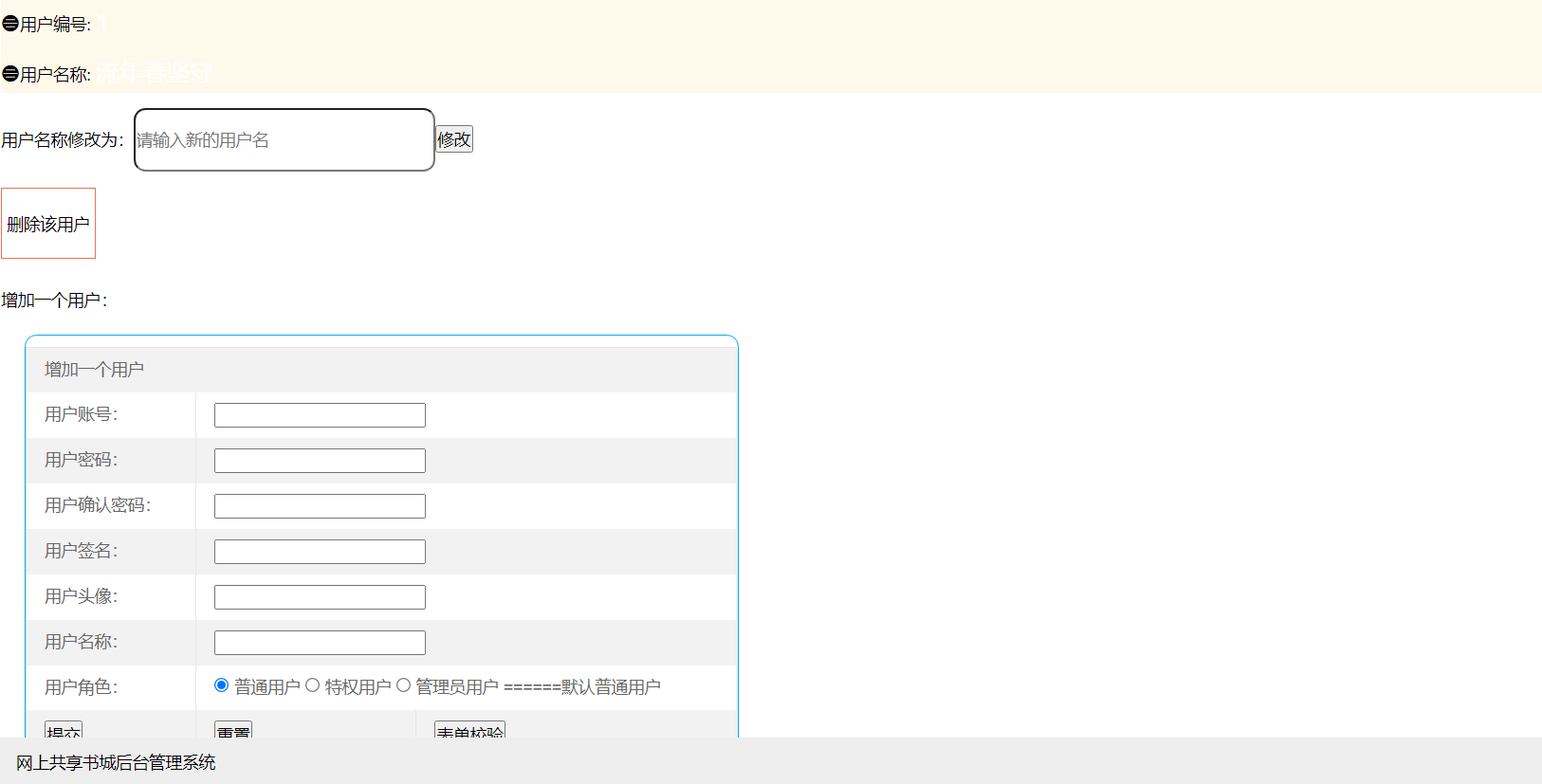
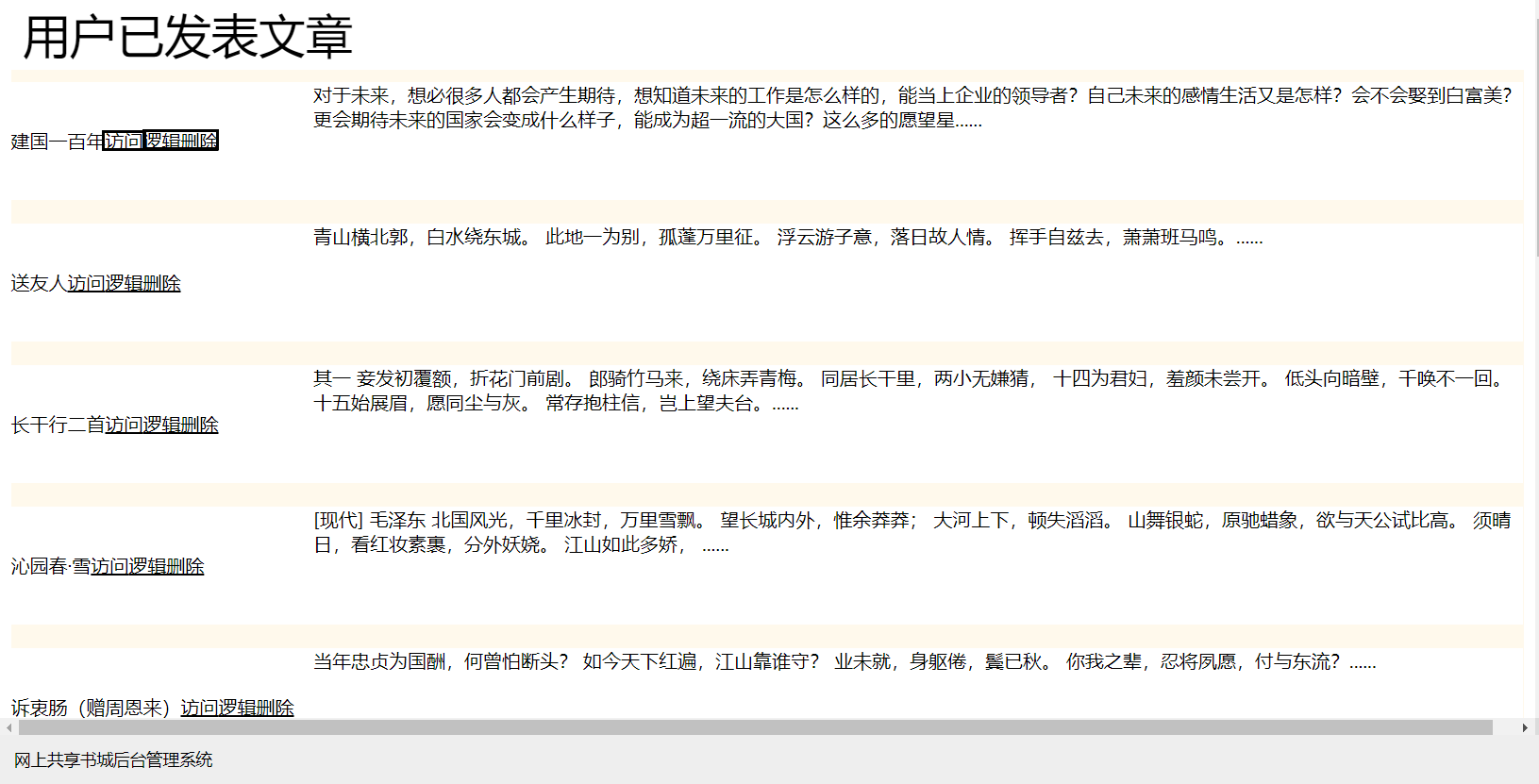


图3.24 用户管理页面

可以对相关用户进行CRUD操作。通过关键字检索用户的信息，然后根据不同用户进行操作，如修改用户名、删除该用户等等。在这里也可以手动添加一个用户，用户模块还有直接将多有用户直接以分页的形式展示在前端页面。

### 3.4.2 文章管理模块

文章管理模块类似于用户操作模块，此处多了一个逻辑删除的功能，也就是说用户点击此处不会真正删除（设置了一个deleted字段），而只有在文章删除那里进行物理删除才会真正删除该文章。



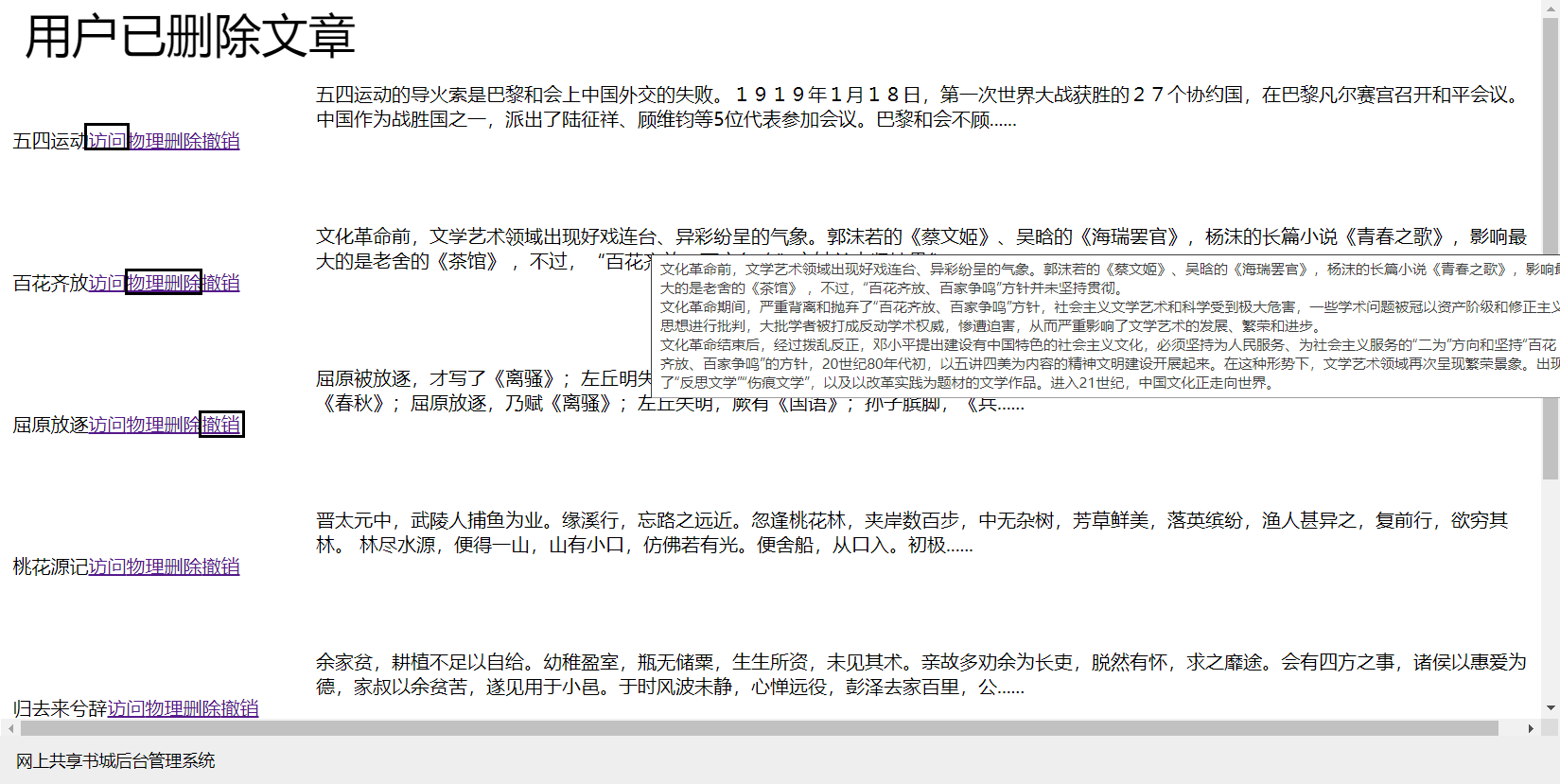


图3.25 发表文章管理页面

对相关发表文章进行删除和访问操作。文章发表这里会展示用户发表过的文章，管理员可以进行访问，也可以进行逻辑删除和物理删除，在文章删除模块可以撤销逻辑删除，也就是说文章删除这里起到了一个回收站的效果。

### 3.4.3 书库管理模块

书库管理只有一个删除功能，页面是通过Ajax即时渲染的。在这部分则是通过Ajax请求数据库，将请求回来的数据通过DOM操作进行循环渲染到前端页面上。

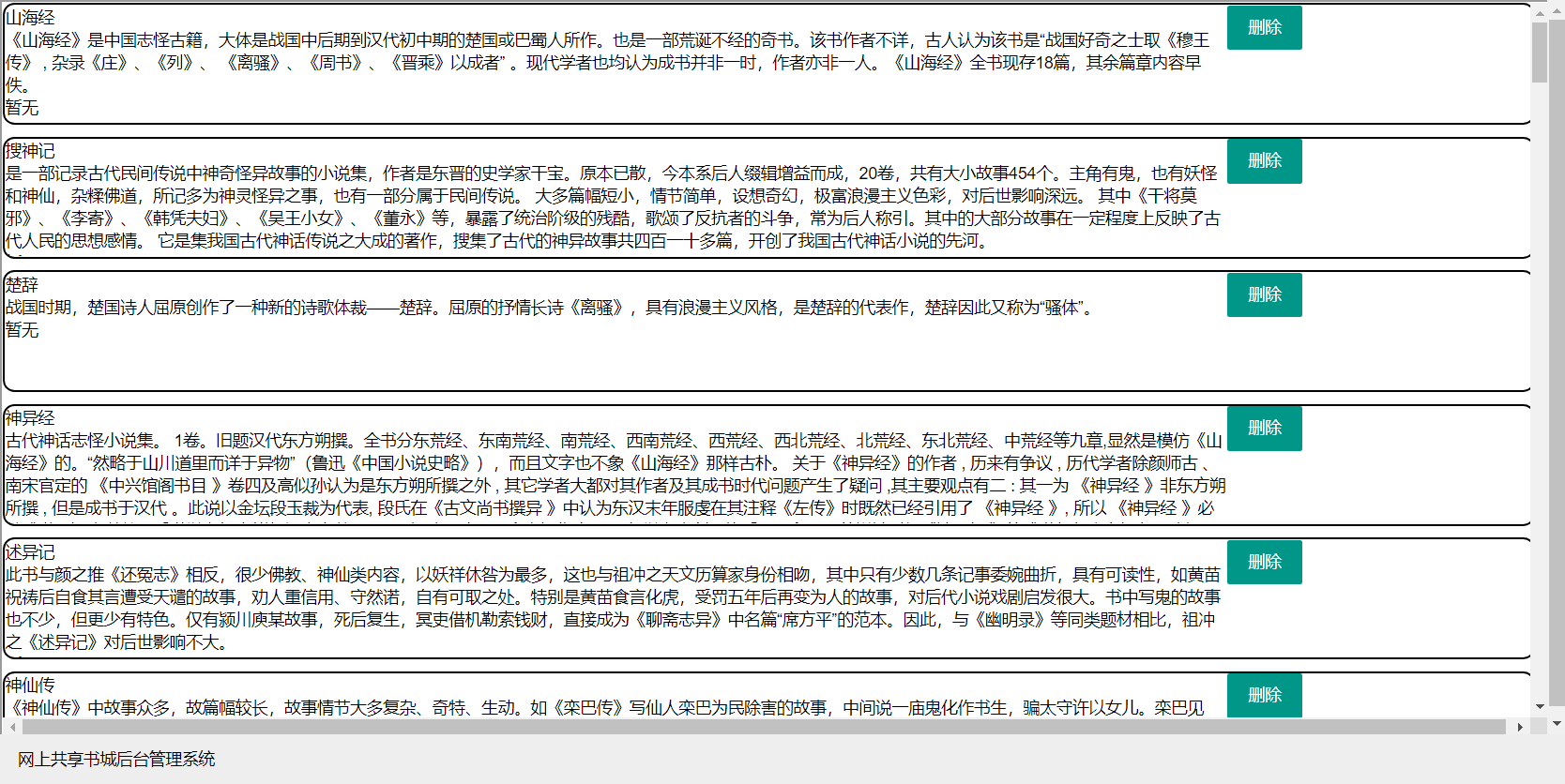


图3.26 书库管理页面

对相关书库进行删除操作。删除操作则是通过将对应书籍id传到后端，后端根据id进行数据库的删除操作，删除的书籍也就会即时渲染消失在页面上。

# 4 系统实现

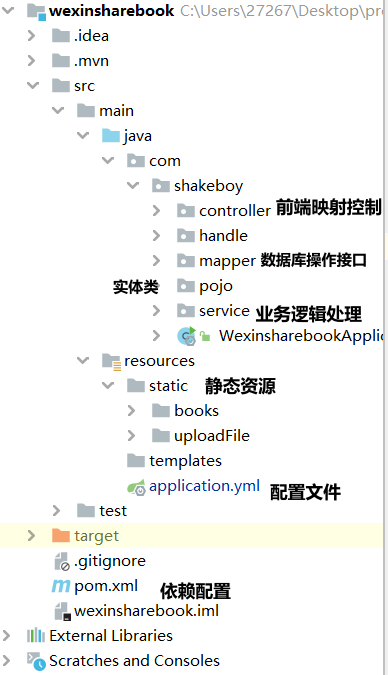
系统实现主要是介绍前后台的搭建和实现业务逻辑的整套流程，前端搭建利用layui和前端三要素（HTML，CSS，JavaScript）以及一些基本的jQuery和Ajax技术，实现基本的界面。后端则采用spring boot和mybatisplus+mysql实现基本的业务逻辑。

## 4.1 前端实现

微信小程序的数据绑定和请求api的调用，利用一些简单的组件，类似于HTML和css进行页面开发，前端搭建则是利用前端框架进行搭建。

### 4.1.1 前期配置

打开idea，新建初始化项目，然后新建基本的包，controller，service，mapper，pojo，util等，在配置文件中配置我们的相关环境，在pom.xml导入我们的依赖等等。

图4.1 Spring Boot项目搭建图

搭建好基本的框架后然后进行下面的基本配置。新建好开发的工程之后，在pom.xml文件导入相关依赖，数据库连接（MySQL）相关依赖，json数据解析依赖，热部署、实体快速创建等等相关依赖，如下列就是mybatis-plus的依赖，就是通过maven来构建项目的，大大简化了以前javaWeb,ssm,hibernate等等框架的配置工作。

<dependency>  
 <groupId>com.baomidou</groupId>  
 <artifactId>mybatis-plus-boot-starter</artifactId>  
 <version>3.4.2</version>  
</dependency>

接着就是在aapplication.yml就是相关的配置，如端口号、MySQL连接数据源、静态资源、控制打印SQL语句等相关信息。

spring:  
 datasource:  
 username: root  
 password:  
 url: jdbc:mysql://localhost:3306/sharebook?useUnicode=true&characterEncoding=utf-8&useSSL=true&serverTimezone=UTC  
 driver-class-name: com.mysql.cj.jdbc.Driver  
# 静态资源路径配置  
 resources:  
 static-locations: classpath:/META-INF/resources/,classpath:/resources/,classpath:/static/,classpath:/templates/,classpath:/public/  
# 配置slq打印日志  
mybatis-plus:  
# global-config:  
# field-strategy: 1  
 configuration:  
 log-impl: org.apache.ibatis.logging.stdout.StdOutImpl  
 mapper-locations: classpath\*:mybatis/\*.xml  
server:  
 port: 8888

### 4.1.2 登陆注册功能实现

页面编写完成之后，通过wx.request以表单的形式向后台服务器提交POST请求，服务器就会对发来的请求进行接收，通过controller调用service的业务逻辑处理函数，service会通过底层的mapper进行数据库用户表的插入，然后返回true给用户，这样注册的流程就完成了，注册成功之后就会跳转到登陆页面，用户在登陆时必须正确输入用户名和密码才能登陆到个人信息页面，具体过程就是，通过将用户和密码发送到后端接口，后端在数据库查找是否有该用户，如果有就返回true，允许登录，否则返回false，不允许登录。

### 4.1.3 个人信息渲染和个性签名功能实现

wx.request发送get请求，根据用户信息获取登录用户的基本信息，然后通过微信的数据渲染即时的渲染到用户个人界面上。

      wx.request({

        url: 'http://localhost:8888/user/getAUserById',

        method: "GET",

        data: {

          "user\_id": wx.getStorageSync('userInfo').userId

        },

        success: res => {

          // console.log(res.data)

          let user = res.data

          this.setData({

            username: user.userName,

            avator: user.userAvator,

            userInfo: user

          })

        }

      })

### 4.1.4 社区模块的渲染和模糊查询

在社区页面发送请求http://localhost:8888/ubook/getAllUBook，后台处理请求，去数据库查找所有数据，然后在controller层将数据通过fastjson的相关封装类将返回对象转换成json字符串，最后前端将获取的json字符串接收转换成json对象，然后渲染到页面中（发表最新的十条），还有一个功能就是模糊查询具体数据库操作如下：

<select id="searchByKeyvalue" resultType="com.shakeboy.pojo.Book">

select \* from books

where book\_name like "%" #{keyvalue} "%"

or book\_description like "%"#{keyvalue}"%"

or book\_author like "%"#{keyvalue}"%"

or book\_type http://localhost:8888/ubook/getAllUBook

like "%"#{keyvalue}"%"

</select>

原理就是，在输入的过程中向服务器发送请求并即时渲染到前端页面。

### 4.1.5 发表社区和百度地图api

form表单提交和跟社区模块类似的用户发表文章渲染。发表社区涉及到文章发表和书籍发表，他们都是一表单的形式进行提交给服务器端，服务器根据不同数据进行存储，存储后返回给客户success，接着就可以直接在自己的发表社区查看相关的文章，也可以在共享模块社区查看发表的响应书籍内容。百度地图api的调用则是通过发送关键在到百度相关接口，他们会根据关键字进行定位经纬度然后在地图显示相应的位置，这样就可以搜索酒店、学校、网吧之类的地方。

## 4.2 后端实现

### 4.2.1 用户管理

在前端页面发送Ajax请求到服务器，后台controller接收到请求后调用业务逻辑处理函数，service就可以通过注入的mapper操作数据库，把相关信息查询出来并一层层返回给前端data,再次通过json转换成json对象，就可以通过操作前端的DOM元素进行及时的渲染了，至于分页技术也是通过在渲染的过程中加一些判断的元素在里面，分页插件调用的是Layui的官方插件。

用户检索是通过模糊查询的方式查找相关的数据，查出来的用户可以通过点击进行操作，可以修改用户名，也可以删除该用户，同是也可以进行添加一个用户。具体的实现细节就是通过发送请求通过三成架构进行数据库的相关操作。

### 4.2.2 文章管理

在这一部分，管理员可以查看用户发表的文章，这里是通过Ajax实现的，在页面上可以进行逻辑删除和物理删除，也可以查看书籍。实现过程是通过发送Ajax请求给后台，后台接收到相关的请求之后进行业务逻辑的处理，流程如下：

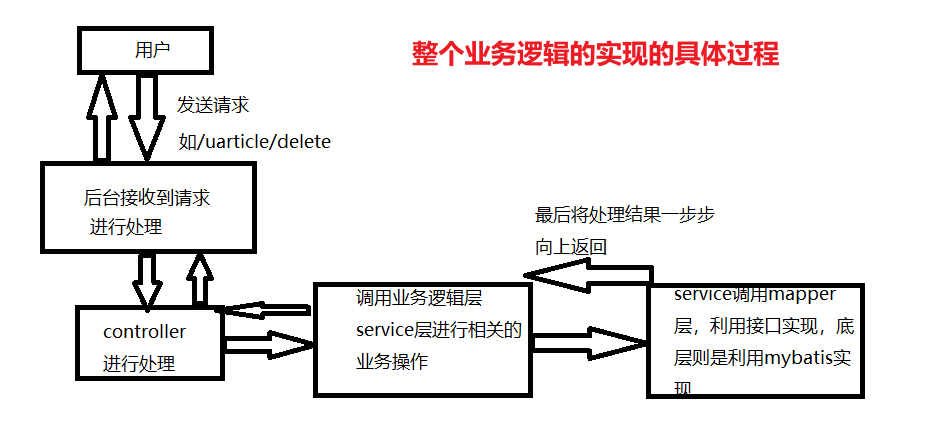


图4.2 文章管理业务逻辑图

用户在发表文章后会走上述的一套流程，然后服务器就会将相关信息返回给我们的用户，用户据可以根据返回的数据做相应的处理。

### 4.2.3 书库管理

在这一部分，只有一个删除功能。在页面布局上用一个div包裹，然后用Ajax请求即时渲染，将每一个请求回来的书籍对象用一个div包裹然后动态的渲染到页面山，主要用到的技术就是jQuery和DOM元素操作。

# 5 系统测试

## 5.1 服务端运行结果和测试

集成了spring security框架，这是个非常好用的安全框架，通过加入相关依赖即可使用，它会对一切请求进行拦截，只有管理员用户才能进行该系统，在后台编写了MyUserDetailsService，该类实现了UserDetailsService，可以在这个类面实现数据库验证的操作。

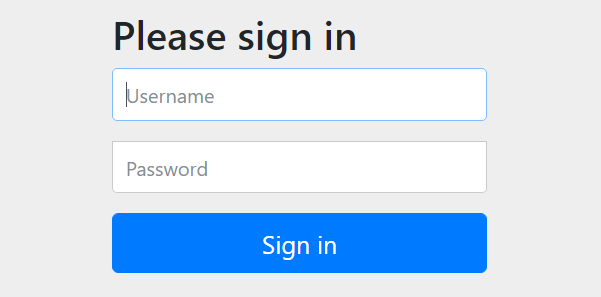


图5.1 Spring Security自带安全检验登录页面

这一部分管理员只有正确地输入用户名和密码才能实现我们基本的登录功能，在没有登陆之前我们不能访问服务内地任何资源（包括静态资源），所以通过Spring Security实现我们的安全框架，效果很好，操作也比较简单

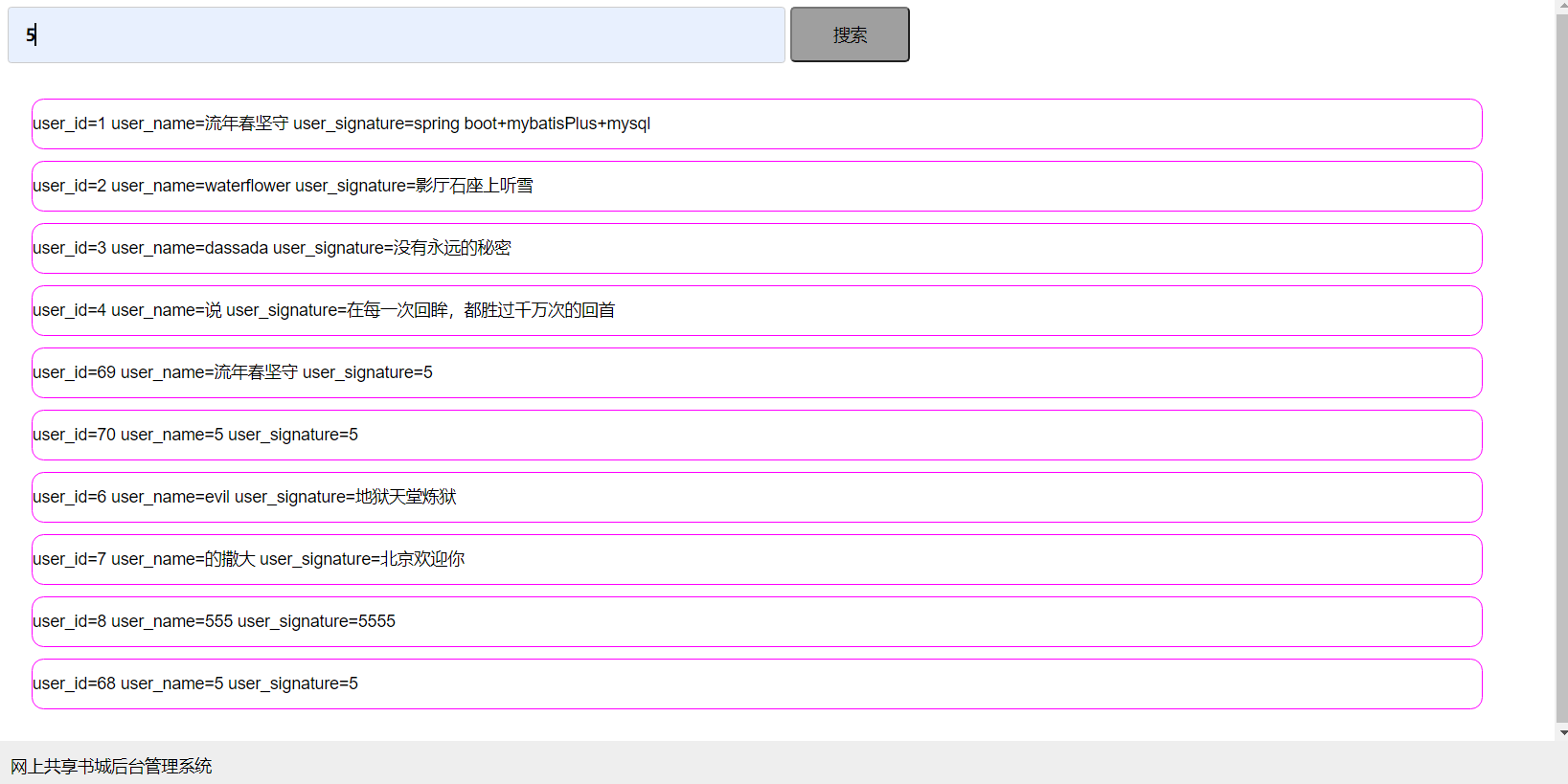


图5.2 后台用户操作页面

在这里可以进行模糊查询，搜索相关用户进行增删改查操作。管理员通过检索相关用户，然后点击该用户进入用户操作界面，对该用户进行相关的CRUD操作。也可以以表单的形式新增一个用户。

## 5.2 小程序端运行结果及测试

在小程序端，用户只有正确输入用户名和密码才能登录到小程序里面，当用户输入的账号或者密码错误时，就会有下图所示的弹框，登录功能地实现则是和数据库中地users表相关，数据校验也是根据users中字段进行比对。

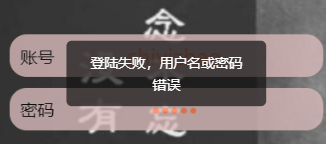


图5.3 用户登录页面

登陆成功后可以来到个人中心，在个人中心可以查询天气，修改个人信息，发表个性签名，去书库看书等等，登陆成功的用户也可以区社区模块查看其他用户发表的数据，并进行加好友操作。当然用户也可以自己发表文章和书籍，其他人可以通过用户发表的书籍访问发表书籍地用户。

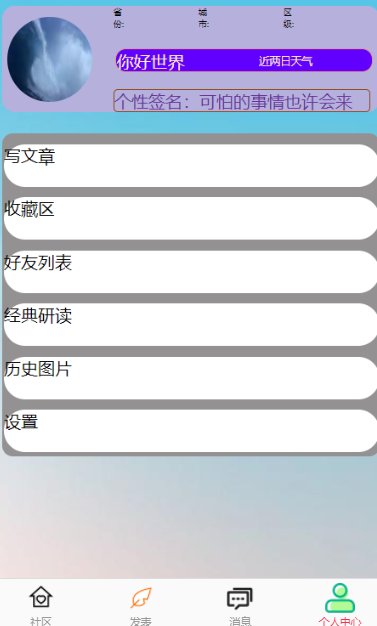


图5.4 默认注册用户页面

去社区看别人发表的书籍，也可以去加好友。这里还有百度api调用天气，用户输入省份和城市字段就可以通过发送请求将返回的json数据渲染在前端页面，这样就可以直接看今日天气了，也可以通过输入关键字搜索周边的饭店、商店之类地地方。

# 6 展望与总结

基于小程序的网上共享书城的设计与实现，后端是通过Spring Boot和MybBtis-Plus来搭建的，前端的微信小程序开发主要是根据微信开发文档介绍的组件化开发，网络请求，也数据绑定以及相关的渲染方法实现。整体上工作量适中，主要是抱着打造一个网上共享书城的想法，让更多的人获取资源更加方便，而且获得的资源是有效而且具有实际意义的，完全就是抱着一种打造共享的书城环境，具体的实现分为了前后端的页面搭建和业务逻辑实现。

Java作为一种主流的开发语言，对于Java开发者的框架也有很多，主流框架Hibernate、Struts2、SSM以及Spring Boot等等都是非常好的Java框架，对于选择Spring Boot框架的原因很简单，因为Spring Boot易学、容易上手，基本上不需要什么配置就可以就行开发了，在这段时间学习了这些框架之后，有了基本的逻辑思维，对于一个项目的整体开发的把握，以及开发前的项目搭建，服务器部署等等工作，都是有很多学习的地方。无论选择那种框架只要得心应手了，都不失为一个好的开发框架，Spring Boot作为主流框架的引领着，极大优化了配置，各种自动配置和maven依赖构建项目，让项目开发的效率大大提高。

微信小程序也是目前非常流行的一种线上产品，依赖于微信而存在，能够快速开发，快速实现线上的一个传播，也不需要卸载，只要打开微信，搜索小程序就可以使用，非常简便，免去了我们必须下载的繁琐过程。微信小程序开发用微信开发者工具进行开发，在wxml文件进行页面的编写，在wxss进行前端页面的美化，json文件用于页面的基础配置，例如头部的颜色、标题之类的，js文件用来动态渲染和处理一些相关的事件，也可以处理各种业务请求。整体上的微信小程序开发十分简易，组件化开发能够是我们的开发变得更加容易上手和学习。除此之外，还可以调用各种api，实现各种美化和实用的操作，云开发、可视化开发、公众号开发、小游戏开发都非常有意思。

参考文献

[1] 陈莹钰. 基于区块链技术的智能家居系统的设计与实现[D].南京邮电大学,2020.

[2] 李昊. 基于微信小程序的智能推荐点餐系统的设计与实现[D].南京邮电大学,2020.

[3] 李莹,鲍卫华,李海英.基于微信小程序的成绩查询系统的设计与实现[J].科技创新与应用,2021.

[4] 汪加楠,崔紫光,吴亮,崔小玉,王聪聪.基于微信小程序的甲骨文学习系统设计[J].电脑编程技巧与维护,2021.

[5] 齐鹏飞. 基于微信的智慧校园小程序设计[D].西安电子科技大学,2020.

[6] 褚晓洁. 基于微信平台的快时尚服装品牌传播研究[D].东华大学,2020.

[7] 于凤. 基于微信小程序的小学生“数学问题解决”能力分步训练研究[D].内蒙古师范大学,2020.

[8] 曾旺. 食用农产品溯源信息防篡改的区块链技术应用研究与实现[D].南昌大学,2020.

[9] 张宝虎. 基于微信小程序的CBMS桥梁巡查工具研发与应用[A]. 中国公路学会养护与管理分会.中国公路学会养护与管理分会第十一届学术年会论文集[C].中国公路学会养护与管理分会:中国公路学会养护与管理分会,2021.

[10] 九官.2020年度微信生态科技企业TOP50[J].互联网周刊,2021(07):48-49.

[11] [Ding Qifeng](https://scholar.cnki.net/home/search?sw=6&sw-input=Ding%20Qifeng" \t "https://schlr.cnki.net/zn/Detail/index/WWMERGEJLAST/_blank). Application of WeChat Mini Program in Xindong Gate Project[J].[IOP Conference Series: Earth and Environmental Science](https://scholar.cnki.net/journal/index/SIPD175513150027" \t "https://schlr.cnki.net/zn/Detail/index/WWMERGEJLAST/_blank)Volume 760, Issue 1. 2021.

[12] 陆禹文. 基于微信小程序的移动学习平台的设计、开发及应用[D].兰州大学,2020.

[13] 黄宇琼. 湘风云绣微信小程序营销平台设计[D].湖南师范大学,2020.

[14] 杜港.微信小程序在医院患者订餐中的应用[J].中国医院建筑与装备,2021.

[15] [Xinying Wang;](https://scholar.cnki.net/home/search?sw=6&sw-input=Xinying%20Wang" \t "https://schlr.cnki.net/zn/Detail/index/WZMERGEJLAST/_blank)[Haoyang Chen;](https://scholar.cnki.net/home/search?sw=6&sw-input=Haoyang%20Chen" \t "https://schlr.cnki.net/zn/Detail/index/WZMERGEJLAST/_blank)[Mengying Jia](https://scholar.cnki.net/home/search?sw=6&sw-input=Mengying%20Jia" \t "https://schlr.cnki.net/zn/Detail/index/WZMERGEJLAST/_blank). 基于微信Applet的线上线下混合教学平台的设计与实现[J].[Advances in Higher Education](https://scholar.cnki.net/journal/index/SJUU242484280006" \t "https://schlr.cnki.net/zn/Detail/index/WZMERGEJLAST/_blank)Volume 5, Issue 1. 2021.

[16] 陈甜甜. 综合档案馆档案信息移动服务研究[D].湘潭大学,2020.

[17] 汤礼锋.微信小程序在基层医院手术室手术登记中的应用[J].中医药管理杂志,2021.

[18] 罗宇东. “文件小助手”安全共享系统的微信小程序实现技术[D].广东技术师范大学,2020.

[19] 张一飞,左涛.基于微信小程序云开发测绘项目信息管理系统[J].测绘技术装备,2021.

[20] 倪旺. 基于微信小程序的车位预约设计[D].河北大学,2020.

附录一：application.yml配置文件

spring:  
 datasource:  
 username: root  
 password:  
 url: jdbc:mysql://localhost:3306/sharebook?useUnicode=true&characterEncoding=utf-8&useSSL=true&serverTimezone=UTC  
 driver-class-name: com.mysql.cj.jdbc.Driver  
 thymeleaf:  
 mode: HTML  
 prefix: classpath:/templates/  
 suffix: .html  
 encoding: utf-8  
 cache: false  
 resources:  
 static-locations: classpath:/META-INF/resources/,classpath:/resources/,classpath:/static/,classpath:/templates/,classpath:/public/  
 security:  
 user:  
 name: shakeboy  
 password: shakeboy  
mybatis:  
 mapper-locations: classpath:com/shakeboy/mapper/\*Mapping.xml  
 type-aliases-package: com.shakeboy.pojo  
  
# 配置slq打印日志  
mybatis-plus:  
 configuration:  
 log-impl: org.apache.ibatis.logging.stdout.StdOutImpl  
  
#showSql  
logging:  
 level:  
 com:  
 shakeboy:  
 mapper: debug  
server:  
 port: 8080

附录二：pom.xml配置文件

<dependencies>  
 <!-- thymeleaf热部署-->  
 <!-- devtools热部署-->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-thymeleaf</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <optional>true</optional>  
 <scope>true</scope>  
 </dependency>  
 <!-- web启动-->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.thymeleaf.extras</groupId>  
 <artifactId>thymeleaf-extras-springsecurity5</artifactId>  
 </dependency>  
 <!--整合Mybatis框架-->  
 <dependency>  
 <groupId>org.mybatis.spring.boot</groupId>  
 <artifactId>mybatis-spring-boot-starter</artifactId>  
 <version>1.3.2</version>  
 </dependency>  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
 <!-- Spring Security安全校验启动依赖 -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.security</groupId>  
 <artifactId>spring-security-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <!-- mybatis-plus-->  
 <dependency>  
 <groupId>com.baomidou</groupId>  
 <artifactId>mybatis-plus</artifactId>  
 <version>3.4.2</version>  
 </dependency>  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>com.alibaba</groupId>  
 <artifactId>fastjson</artifactId>  
 <version>1.2.54</version>  
 </dependency>  
</dependencies>

致 谢

作者在本论文开发过程中，受到了导师金红的悉心教导，可以说导师对作者的驱动力十分大，从前期的选题到后来选题报告，导师的细腻教学和监督让作者受益颇多，包括在中期的检查提出了许多宝贵的建议，可以说导师金红是本次论文的不二功臣。作者在导师的严谨而不失风雅的教导之下完成了本次论文工作，也得以完成本科的结业论文工作。所以，首先对尊敬的导师金红表以深深的歉意和真挚的感谢。

其次，还要感谢家里人对我的支持，作者在开发过程中，不用担心的生活起居，各方面的经济都不成问题，而且给予作者充足的时间和精力用于开发工作。

最后作者还要感谢自己在本次开发过程中坚持不懈，直到最后，作者也学习到了很多东西，从项目的需求分析和数据库设计，包括环境搭建等等，作者都受益颇多。