中山大學
SUN YAT-SEN UNIVERSITY

本科生毕业论文（设计）

Undergraduate Graduation Thesis（Design）

题目 Title：基于 Mpvue 的在线书城小程序

院 系
School (Department)：数据科学与计算机学院

专 业
Major：软件工程(移动信息工程)

学生姓名
Student Name：蔡丽芝

学 号
Student No.：15352006

指导教师(职称)
Supervisor（Title）：张永民(高级讲师)

时间： 2019 年 5 月 7 日

Date: Month May Day 7 Year 2019

表一：毕业论文（设计）开题报告

Form 1: Research Proposal of Graduation Thesis (Design)

论文（设计）题目
Thesis (Design) Title：基于 Mpvue 的在线书城小程序

选题的目的：

随着国民文化素质的提高，阅读渐渐成为我们必不可缺的一部分，我们对阅读载体要求也越来越高，过去在移动端浏览器中由于文字过小，不适应手机屏幕的阅读方式已经多次被人诟病，而微信小程序的出现，不仅可以解决移动端文字适配问题，优化阅读体验，还可以减小手机内存存储负担。不同于 app 的系统兼容性问题，无论是 ios 系统，还是安卓系统，我们只需开发一套微信小程序，即可在这两种系统上运行，基于以上的原因，设计开发一个书城小程序尤为重要。

思路

该书城大体有以下几块功能

1. 书籍首页：图书展示，提供搜索图书和随机推荐书籍的功能。
2. 阅读器页： 可实现阅读器功能，同时可播放文字内容
3. 书架页：收藏图书功能，可进行分组和移入书架和移出书架

方法

1. 前端技术栈

小程序自带技术栈，但官方并不推荐开发复杂应用，而书城小程序涉及到许多复杂的功能，要求做到组件和模块化，自动化构建，代码复用等，因此，我打算通过 vue 来开发小程序，但由于小程序有自己的开发框。因此，为了能够在小程序中使用 vue，需要借助 mpvue 这一开源的前端 开发框架来进行小程序的开发，有了它，即可使用 vue 进行开发小程序

2. 后端技术栈

需要在阿里云租一个云服务器，在云服务器上搭建 Nginx 服务，并使 node 作为后端实现服务

Student Signature： 蔡丽芝 Date: 2018/11/18

指导教师意见 Comments from Supervisor： 同意开题   1.同意开题 2.修改后开题 3.重新开题 1.Approved( ) 2. Approved after Revision ( ) 3. Disapproved( )   Supervisor Signature： Date: 2018/11/8

表二：毕业论文（设计）过程检查情况记录表
Form 2：Process Check-up Form

指导教师分阶段检查论文的进展情况（要求过程检查记录不少于 3 次）
The supervisor should check up the working process for the thesis（design）and fill up the following check-up log. At least three times of the check-up should be done and kept on the log.

第 1 次检查（First Check-up）：

学生总结
Student Self-summary：对当下已有的书城应用 app，小程序进行调研，对项目总体内容进行需求分析，并确定基础的功能模块。考虑整体项目难度

指导教师意见
Comments of Supervisor：当下书城应用很多，要注意突出创新点，可以考虑增加一些新的功能模块。

第 2 次检查（Second Check-up）：

学生总结
Student Self-summary：完成项目开发环境搭建，完成项目前端界面开发，和部分后台接口。

指导教师意见
Comments of Supervisor：建议完善后台架构的整体性

第 3 次检查（Third Check-up）：

学生总结
Student Self-summary：完成所有功能开发，完成功能测试，撰写毕业论文，定下毕业论文初稿

指导教师意见
Comments of Supervisor：注意论文结构，需求分析时建议使用用例图和时序图

第 4 次检查
Fourth Check-up

学生总结

Student Self-summary：完成毕业论文，并通过查重，查重率为 0.2%。项目功能开发与测试完成。

指导教师意见（Comments of Supervisor）：注意目录格式和参考文献格式

学生签名（Student Signature）：　　日期（Date）：

指导教师签名（Supervisor Signature）：　　日期（Date）：

总体完成情况（Overall Assessment）	指导教师意见 Comments of Supervisor： 1、按计划完成，完成情况优（Excellent）：　（ ） 2、按计划完成，完成情况良（Good）：　（ ） 3、基本按计划完成，完成情况合格（Fair）：（ ） 4、完成情况不合格（Poor）：　（ ） 指导教师签名（Supervisor Signature）： 日期（Date）：

表三：毕业论文（设计）答辩情况登记表

Form 3: Thesis Defense Performance Form

答辩人 **Student Name**	蔡丽芝	专 业 **Major**	软件工程（移动信息工程）
论文（设计）题目 **Thesis（Design）Title**	基于 **Mpvue** 的在线书城小程序		
答辩小组成员 **Committee Members**			

答辩记录

Records of Defense Performance：

记录人签名（Clerk Signature）：　　日期（Date）：

学术诚信声明

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名：　　蔡丽芝　　日期：2019/5/7

Statement of Academic Integrity

I hereby acknowledge that the thesis submitted is a product of my own independent research under the supervision of my supervisor, and that all the data, statistics, pictures and materials are reliable and trustworthy, and that all the previous research and sources are appropriately marked in the thesis, and that the intellectual property of the thesis belongs to the school. I am fully aware of the legal effect of this statement.

Student Signature：　　蔡丽芝　　Date：2019/5/7

【摘　要】

本文设计了一款基于Mpvue框架开发的微信小程序“看书吧”。在技术方面：使用mysql作为数据库，使用nodejs进行网络爬虫获取真实线上数据；前端使用Mpvue框架，实现代码在h5项目和小程序项目之间的复用以及组件的复用，“看书吧”不仅可以作为小程序项目在微信端运行，经过少量代码的调整，可转化为h5项目在移动端浏览器中运行。UI使用iview weapp框架，统一项目界面风格；后端使用Koa2框架，提供前端所需数据接口API，并使用redis技术，增加缓存层，提高性能。在功能方面：实现了书城首页，阅读器，我的书架，评论模块，书籍详情，搜索，个人中心。同时使用css3动画技术，提高用户体验友好度。用户可以通过书城首页查看书籍排行榜，通过阅读他人对书籍的评论，找到喜欢的书籍进行阅读，并根据个人喜好设置界面UI，也可将书籍存入我的书架中，稍后阅读。“看书吧”是一款打通前后端，使用真实线上数据的小程序应用。

【关键词】小程序；Mpvue；nodejs；Koa2；redis

[ABSTRACT]

"Reading Bar" is a WeChat mini programe based on the Mpvue framework. In terms of technology: uses mysql as the database, nodejs technology for web crawling to obtain real online-data; Front-end uses the Mpvue framework to achieve the reuse of the code between the h5 project and the applet project as well as the components ,it makes the Reading bar not only run as a small program project on the WeChat side, but after a small amount of code adjustment, it can be converted into a h5 project to run in the mobile browser. The UI uses the iview weapp framework to unify the project interface style; The back-end uses the Koa2 framework to provide the front-end data interface , and uses redis technology to increase the cache layer and improve performance. In terms of functions: there are seven modules, Book City Home, Reader, My Bookshelf, Comment Module, Book Details, Search, and personal Center. At the same time, it uses css3 animation technology to improve user's experience. The user can view the book leaderboard through the book home page, find the favorite books for reading through other people's comments on the book, and set the interface UI according to personal preference, or save it in my bookshelf and read it later. Reading Bar is a mini program application that uses real-time online data to get through the front and back.

[Keywords] wechat small Programe; Mpvue; nodejs; koa2; redis

目　录

第一章 引言

1.1 研究背景

随着国民素质的不断提高，阅读逐渐成为我们生活中不可或缺的部分。有了移动设备的出现，人们阅读的载体不再仅仅只是局限于纸质，通过手机，平板，我们可以轻轻松松随时随地进行阅读。便捷的获取知识途径同时，人们对阅读体验的要求也在不断的提高。早期由于大多数网页都是基于大屏幕的 pc 端设计，人们在手机浏览器端阅读时，常常会出现小说文字内容过小，页面内容与屏幕大小不适配的问题。

随着移动端设备的普及，Ethan Marcotte 提出了一种自适应网页设计[1]的设计思想，极大提高了用户体验，随后 app 的出现，流畅的动画效果，人性化的阅读体验，强大的数据库资源，更是让人们爱上了阅读。然而，由 app 所带来的内存占用问题，同样让用户困扰。这时，小程序横空出世，负载于社交平台微信，以其足以媲美 app 功能，却不占移动设备内存的特点[2]，迅速让人们深陷其中。因而当下许多小说网站都陆续推出小说阅读小程序，借助微信的社交性，提高自身用户量。

1.2 研究现状与目的

书城小程序依托于社交平台微信之中，具有极大的用户流量群。越来越多的小说网站都将自己的产品迁徙到小程序中。然而，小程序开发所需的技术栈与当下大多公司前端的技术栈不一致，学习成本不够低。对于一个业务繁重的复杂应用，往往要求开发方式能够做到组件和模块化，自动构建和集成，代码复用[3]等，然而小程序本身的开发规范限制了这部分功能。 同时，为了小程序而开发的书城代码无法再其他业务场景，如浏览器端使用。因此，”书吧”致力于做到更加工程化的开发体验，并做到代码多端复用的高效性，实现小程序和 H5 项目的灵活转换

1.3 论文结构简介

基于研究背景和目的，将本文的章节安排如下：

第一章：引言，在此章节中分为研究背景和研究现状目的两个方面进行介绍，通过介绍当下移动端阅读应用的背景和流行趋势，分析微信小程序的优势与劣势，引出设计出一款多端复用的书城小程序的重要性。

第二章：相关知识介绍，首先介绍前端框架 Mpvue 和微信小程序的基本原理，接着分享后端框架的基本使用原理，最后介绍缓存层 redis 的原理使用

第三章：书城系统分析，本章节是对系统进行需求和功能分析，通过流程图展示大致需求功能和使用流程介绍，通过用例图展示功能需求模块，最后介绍系统的开发环境。

第四章：系统后台设计，首先对书城后台设计进行总结，接着介绍网络爬虫部分，后端架构设计与缓存层实现，最后讲述数据库的分析与实现

第五章：系统实现，这一部分主要介绍几个功能模块，登录鉴权模块，个人中心模块，我的书架，书城首页，书籍详情页的代码前后端实现。

第六章：系统测试，对小程序的功能进行测试，并展示测试情况

第七章：总结与展望，最后对完成本系统所做的工作，书城系统存在的问题进行总结，并指出系统存在的不足以及未来可发展的方向。

第二章 相关知识介绍

2.1 前端框架 Mpvue 和微信小程序

Mpvue 是一款使用 Vue.js 开发微信小程序的前端框架[4]，开发者只需初步了解小程序的开发规范，熟悉 Vue.js 即可着手进行开发，Mpvue 会将代码转为小程序并确保其正确运行。之所以选择使用 Vue.js 来开发微信小程序，一方面是因为 Vue 如今是大多公司的前端技术栈，另一方面是 Vue.js[5]和小程序的底层原理基本一致，它们都是典型的逻辑视图层框架[6]，小程序底层实现原理如图 2-1 所示。

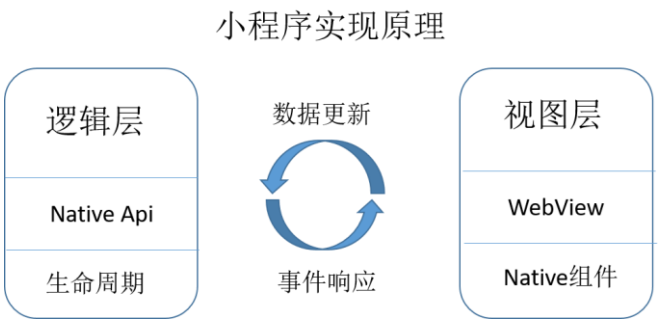

图 2-1 小程序实现原理

和 Vue 工作原理类似，逻辑层和视图层之间的工作方式为：当逻辑层数据发生变化时，会驱动视图层发生视图更新；视图层视图交互触发事件时，事件响应函数会修改数据从而再次触发视图更新。

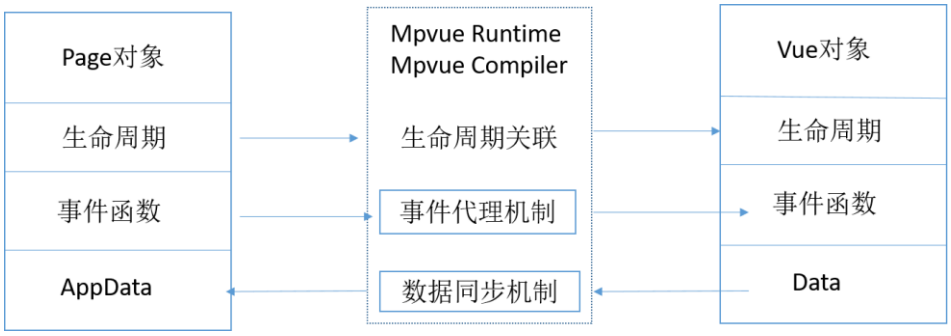

图 2-2 Mpvue 实现原理

如图 2-2 所示，基于 Vue.js 和小程序一致的工作原理，Mpvue 将小程序的功能托付给 Vue.js，它将小程序和 Vue 的生命周期进行关联，并在正确的时机将数据变更同步到小程序中。基于这种机制，我们就可以通过 Vue.js 的形式来进行微信小程序的开发。

2.2 后端框架 Koa2 介绍

Koa2，是基于 Node.js 平台的下一代 web 后台开发框架[7]，通过 async，await 方式优雅的处理异步问题，丢弃可怕的回调地狱问题。Koa 的中间件采用了洋葱圈模型，如图 2-3

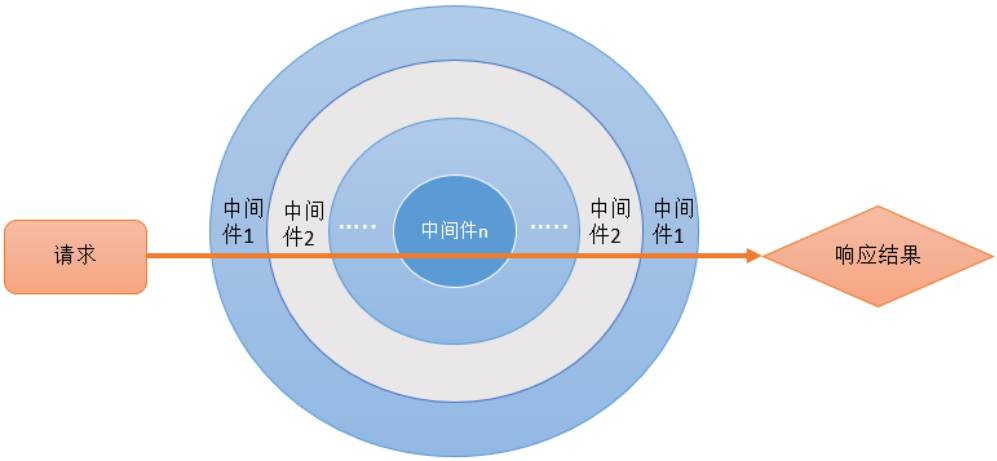

图 2-3 洋葱圈模型

Koa 的洋葱圈中间件模式[8]，中间件可以比作为一个过滤器，它是在客户端和应用程序服务端之间的一个处理请求和响应的方法， 所有来自前端的请求经过一个中间件的时候都会执行两次。利用 Koa2 开发后端，可以避免出现复杂的函数嵌套，使得异步开发 Web 程序，实现实时数据资源存储及获取，权限控制等后端逻辑处理，变得更加简洁明了

2.3 Redis(Remote Dictionary Server)缓存技术

在本次开发中所用到的关系型数据库 MySql，作为一种持久化型数据库，MySql 将表和数据等信息资源全部存储在服务器上的磁盘中。而对于一个上线的 web 应用来说，服务器的读写效率往往是提升应用运行速度的重要条件之一。即提高数据库的处理速度可以很大程度提高应用的的运行速度。通过优化 sql 语句可以提高数据库处理数据的能力[9]，但是，当应用的访问量很大的时候，此时数据库的压力会非常大，应用对数据的处理效率也会大大的降低。因此，此时通过引入缓存技术，来减低数据库的读写次数，提高应用的运行效率，成为一种常用的优化方式。

Redis 是一个高性能的 key-value 非关系型数据库，通过将常用的数据存储在内容分中，并直接在内存中实现数据的增删改查，减少数据库的读写次数[10]，从而保证了数据的处理速度。它可以支持的 value 数据类型非常灵活， 除了常规的 String 类型，还可以保存复杂的数据类型，如 list(链表)，set(集合)，zset(有序集合)和 hash（散列表）。一个典型的使用 redis 作为缓存层的架构如图 2-4 所示。

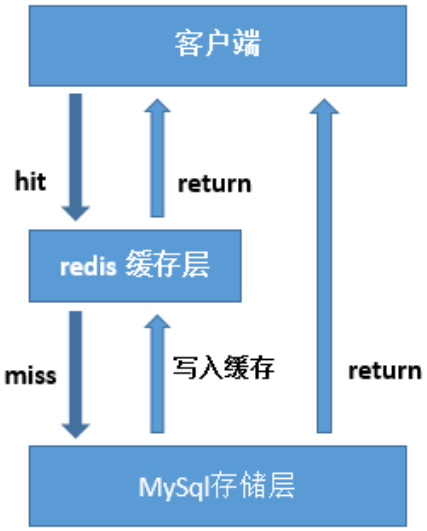

图 2-4 典型缓存架构

由于缓存是存于内存之中，因此通过缓存，可以有效的提高读写速度，优化用户体验，同时当出现一些复杂的 Sql 查询语句的时候[11]，加入缓存，可以帮助后端数据库减少访问量和复杂计算问题，降低后端负载压力，提高应用的运行效率。

第三章 书城系统分析

3.1 需求分析

大数据时代的今天，阅读的载体不再仅仅只是纸质书籍，人们可以通过各种网络设备进行阅读，而网络阅读也逐渐成为人们获取知识的主要途径。作为一款合格高效的在线书城小程序：需要有简洁优雅的界面，可收藏书籍的书架功能，可个性化设计的阅读器以及强大的后背小说数据资源支持。

因此，作为一个在线书城，必需具备以下几个功能

（1）需要有一个界面进行书籍汇总，展示小说排行榜内容，或者对数据库中的资源进行分类，提供全局搜索的功能，使得用户可以快速找到自己喜欢的书籍

（2）需要书籍详情页，去展示该书籍相关信息，并展示出其他人对该本书的评论，同时自己也可以发表评论

（3）需要有阅读器的功能，这是作为书城系统最基础的一个功能，用户可以在查看目录，设置阅读器的风格和字体，并提供加入书架的功能

（4）需要书架功能，用来展示用户收藏的书籍，并且在这里可以对书籍进行删除或者新增书籍

（5）个人中心，用户必须登录才可以使用书城小程序，展示用户的个人信息，同时可以进行阅读设置

（6）作为一个在线的书城小程序，必须要有前台和后台，前台拿到后台的数据进行页面展示，同时后台为前台提供 Api 数据接口，那么小说资源的来源可以通过网络爬虫的方式获取。

3.2 功能分析

基于上述的需求分析，我主要将书城系统的功能分为以下六个模块：书城首页，书籍详情页，阅读器，我的书架，登录鉴权和个人中心。如图 3-1

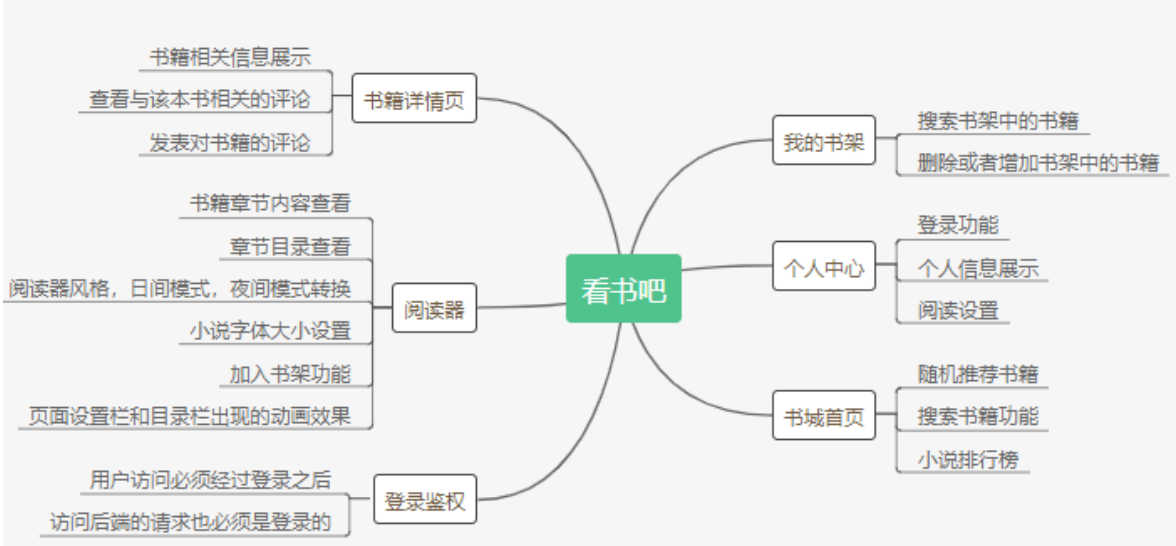

图 3-1 功能模块图

3.2.1 登录鉴权模块

书籍的随机推荐，评论都是以用户为单位，同时对于私人的个人资料与信息，以及小说书籍内容的阅读，我们必须通过鉴权的方式来进行控制，只有登录之后的人，后端才会返回小说的数据，个人信息资料等。如图 3-2

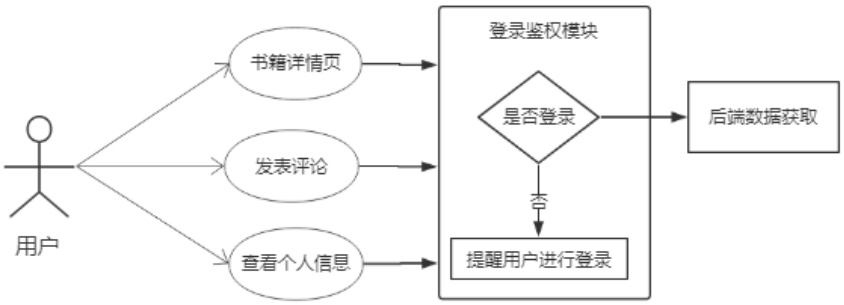

图 3-2 登录鉴权模块用例图

3.2.2 书城首页模块

书城首页模块中包含着许多子模块，比如说，随机推荐模块，精选阅读模块和排行榜模块，随机推荐模块会根据用户曾经阅读过的书籍推荐相关，类似的书籍，通过网络爬虫的方式，抓取排行榜数据，让用户可以看到小说排行，从而找到自己的阅读书籍，图 3-3 为书城首页模块的用例图

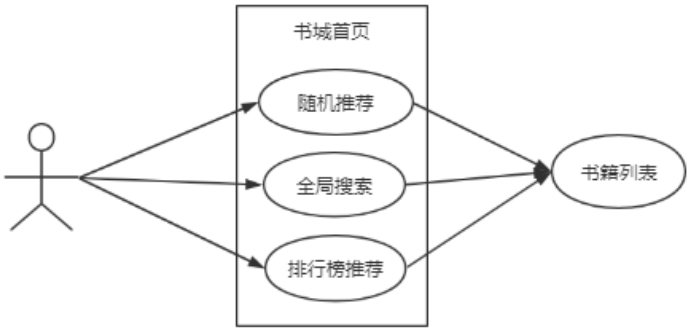

图 3-3 书城首页模块用例图

3.2.3 书籍详情页模块

当用户进入书籍详情页时，首先可以可到书籍详细的相关信息，如作者，简介等内容，同时可以看到所有用户对这本书的评论，自己也可以对书发表自己的评论，最后提供按钮，让用户选择阅读该书籍，还是选择回到首页，重新选择书籍进行阅读。图 3-4 为书籍详情模块用例图

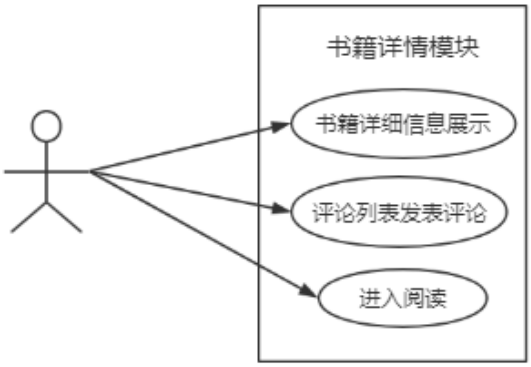

图 3-4 书籍详情模块用例图

3.2.4 阅读器模块

作为书城应用，阅读器是最基本的功能模块，在阅读器中阅读，用户可以查看该本小说的目录，同时可以利用目录定位到对应的章节进行阅读，阅读器模块中有设置阅读器风格的功能，可以随时切换日间模式和夜间模式，用户可以根据个人需求，动态设置字体大小，为了提高用户使用的流畅性，增加设置栏和目录栏的动画效果，当点击屏幕，设置栏会以动画的效果从下到上出现，而不是突然出现。当阅读完当前这一章，可跳到下一章进行阅读，阅读的过程中，如果喜欢这本书，用户可以选择是否加入书架。如图 3-5 是阅读器模块的用例图

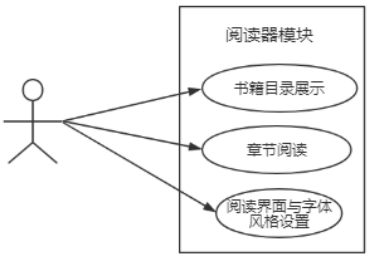

图 3-5　阅读器模块用例图

3.2.5 我的书架模块

在我的书架模块中，首先必需具有书籍展示的功能，能展示用户曾经加入书架中的书籍，同时具有搜索功能，当用户加入书架中的书籍内容较多的时候，加入搜索功能，能够使得用户快速定位找到自己想要找的书籍。同时对书架内容可以进行改动，当长按书架上的书籍时，可以进入编辑状态，选择进行删除书籍。既然可以进行删除，当让也可以增添书架中的内容。如图 3-6 我的书架模块用例图

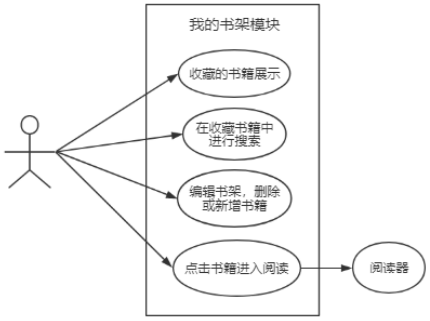

图 3-6 我的书架模块用例图

3.2.6 个人中心模块

在个人中心模块中，首先要展示用户的个人信息，包括用户头像，用户昵称。其次用户能在个人中心中查看自己曾经发表过的评论，查看最近阅读的书。如图 3-7 为个人中心模块的用例图

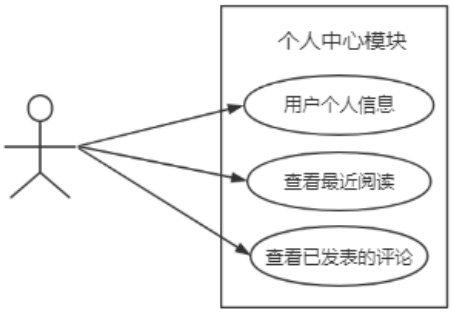

图 3-7 个人中心模块用例图

3.3 系统使用流程图设计

基于上述的功能需求分析之后，整体功能流程图如图 3-8 系统流程图设计

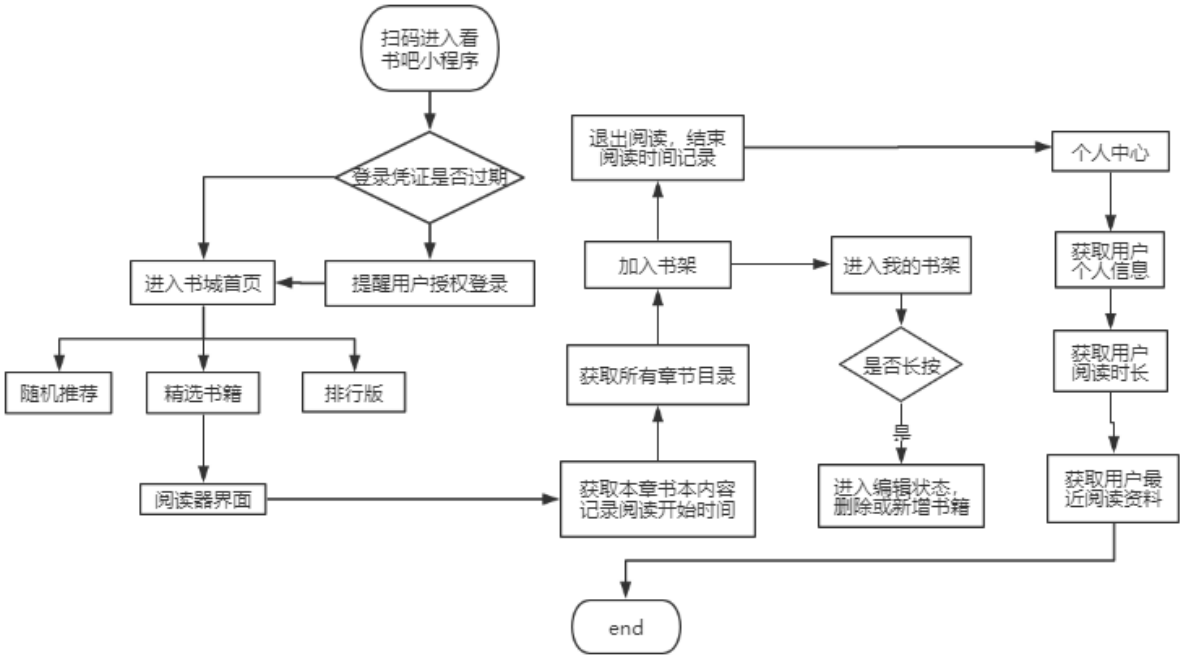

图 3-8 系统流程图设计

3.4 系统开发环境

看书吧所需要的开发环境以及开发工具如下：

（一）开发工具：sublime text3

（二）微信小程序开发工具：微信 web 开发者工具

（三）开发使用的数据库操作软件：Navicat Premium, MySql,

（四）命令行工具： Cmder

第四章 系统后台设计

4.1 全栈书城系统后台设计

看书吧作为一款在线小程序，必须包括前台和后台，小程序前台是用户和小程序应用交互最直接的部分，然而前台中的许多数据和功能离不开后台的支持，比如书城首页，书城详情页，我们要展示排行榜信息，为用户推荐小说，用户发表评论，用户进行登录等等功能，都需要与后台进行交互，前台需要发送各种请求，如图片资源，小说内容请求，登录请求到后台，后台接收到前台的请求，按照特定的逻辑处理，返回响应。

作为看书吧小程序的后台，主要分为以下四个方面：

一. 数据库中小说数据资源的获取，不同于当下其他成熟的书城小程序，拥有强大的小说资源支持，看书吧的小说资源需要通过网络爬虫的方式进行网络上的获取，爬虫的内容包括两类：小说和排行榜信息。爬虫后，存入 mysql 数据库中

二. 具有路由功能，能解析前台发送的请求，根据前台发送请求的 url 地址路由到不同的控制器中，处理业务逻辑，如登录鉴权等。与数据库进行连接，对数据库进行增删改查，从而为前台 api 接口

三. 数据库设计，如何设计表，使得用户可以快速进行数据库资源的增删改查操作

四. 为了提高用户体验，提高后台的响应速度，提高系统的性能，必须尽量减少磁盘 IO 操作，也就是尽量减少数据库的操作，因此，需要加入缓存的功能，当前台发送获取小说排行榜数据的请求时，该请求需要先经过缓存层，若可以找到对应排行榜信息的缓存，则直接丢缓存返回响应，若无再进行数据库操作，从而提高了响应速度。

4.2 爬虫

看书吧小程序的小说数据来源于网络上的小说发布网站。按照功能划分，主要分为两个阶段：

一，爬虫阶段：对网站内容解析，将所需要的小说资源数据爬取下来，

二，数据保存阶段：当我们拿到爬下来的小说资源数据，应该存入数据库中，进行数据持久化操作。

4.2.1 技术选型

“看书吧”小程序主要使用 Node.js 进行网络爬虫[12]，MySql 作为数据库保存数据。

网络上现在有许多语言版本的网络爬虫，如 Python， Java，这些都是非常好用的语言，但是鉴于”看书吧”小程序的后台是基于 Node.js 语言进行开发，为了保持技术的一致性，我选择用 Node.js 进行网络爬虫，获取网络小说资源和排行榜信息。

使用 Node.js 进行爬虫主要使用 request，cheerio，async 三个模块

（一）request 是 Node.js 的一个模块，这个模块的存在，使得 http 请求变得简单方便。Node.js 发送请求的时候，需要用到 request 这个模块，首先要使用 request 发送请求，必须引用 request 这个模块，发送请求的时候，传入两个参数，第一个参数：传入一个配置对象，在这个对象中，可以设置请求 url，编码方式，超时时间等。第二个参数：请求结果回调函数，会传入 3 个参数，第一个参数是错误，第二个参数是响应对象，第三个是返回的请求结果数据。

（二）cheerio 模块是类似前端操作 dom 节点的 Jquery 库，可以对爬虫获取到的 html 内容进行解析，类似 Jquery 对 html 中的 dom 进行节点操作，从而筛选数据，将多余的 html 标签去掉，留下我们需要的数据。更优秀的一点是，cheerio 的语法结构与 JQuery 的基本一致，

开发者只要注意差异，即可上手

（三） async 异步处理模块，在发送请求的过程中，涉及到许多异步的问题，有了 async 模块，可以更加优雅的解决异步流程控制问题。其中在 nodejs 爬虫设计中，主要用到 async 模块的 series 函数，如图 4-1

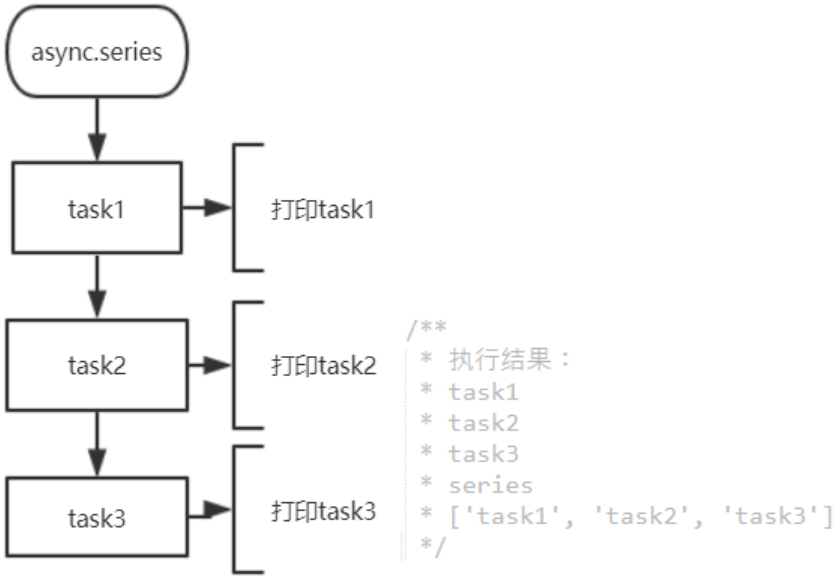

图 4-1　async 模块 series 函数

Series 函数会接收两个参数，第一个参数是函数执行列表，第二个参数是回调函数，利用 async 的 series 方法，可以使得每一个 task，尤其是 task 里都包含着异步操作的时候，让这些函数按照顺序依次从上到下执行。因此如图 4-2 的执行结果是按照从上到下的顺序输出，并且 series 方法对错误的处理是，如果中途发生错误，会马上停止后面函数的执行，将错误传递到回调函数中。

4.2.2 爬虫实现

爬虫主要分为以下几个模块：数据爬取模块，数据保存模块，主程序模块

（一）数据爬取模块是整个爬虫实现的主要功能模块，分为两个子模块：请求模块和数据解析模块。

请求模块：首先需要我们配置好请求的 url 地址， 调用 node.js 的 request 模块，传入我们配置好的 url 地址，会向该 url 发送请求，在

request 模块的回调函数中可以得到我们请求的 html 内容。如图 4-2 为网络爬虫请求模块流程图

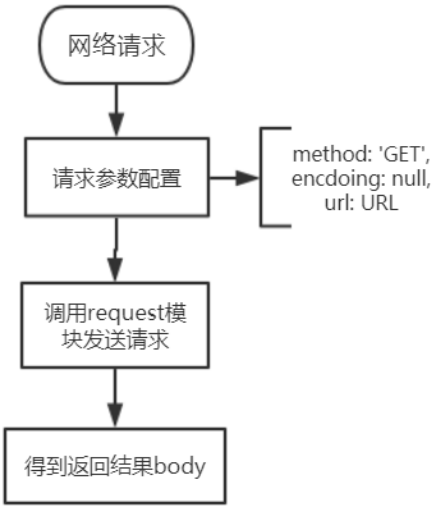

图 4-2 网络爬虫请求模块实现

当获取到返回结果的时候，使用 iconv 模块的 docode 方法，是为了防止编码出现错乱问题，因此进行 jbk 编码。

数据解析模块：当成功抓取到数据之后，需要调用 cheerio 库，通过以 jquery 获取选择 dom 内容的方式，过滤出所要的内容。

1. 书籍基本信息的解析：分析 html 源码，使用 cherrio 去解析获取的 html，并通过选择器的方式，获取书籍的书籍名，封面图片，作者，简介，以及这本书的章节列表地址

   在分析 html 源码的时候，应该找到网页中小说名，封面，简介等信息的 id，并通过 css 选择器，利用 cheerio 去获取对应的内容，从而得到想要的数据。

2. 章节列表内容的爬取：章节列表内容的获取方式和书籍信息的获取方式基本一致，也是通过分析源码，找到对于的 css 选择器，并利用 cheerio 进行获取内容，这里不多加赘述

（二）数据保存

当我们通过网络爬虫的方式，爬取到需要的小说信息和章节列表信息后，下一步要做的是对数据的持久化，也就是要将保存到数据库中。数据库选择 mysql，建了”booklist”和”bookcontent”两张表，用

于存储小说详细信息和章节内容，具体表结构在后续数据库设计中会详细叙述。在该数据保存模块中，封装了对书籍信息和章节内容两个表数据的增删改查操作。

（三）主程序

主程序是后台进行网络爬虫的主入口，在主程序中会调用之前封装好的两个模块：爬虫模块和数据操作保存模块，无论是爬虫，还是数据库操作都涉及到异步回调的问题，为了确保是爬取到数据之后，才存入数据库中，使用了 nodejs 的 Async 模块的 series 函数，如图 4-3 所示

图 4-3 主程序调用流程图

4.3 后端架构设计与缓存层实现

经过需求分析，为了提高后端响应速度，在原有的后端基础架构上，加入 Redis 缓存层，后端架构流程图如图 4-4 所示

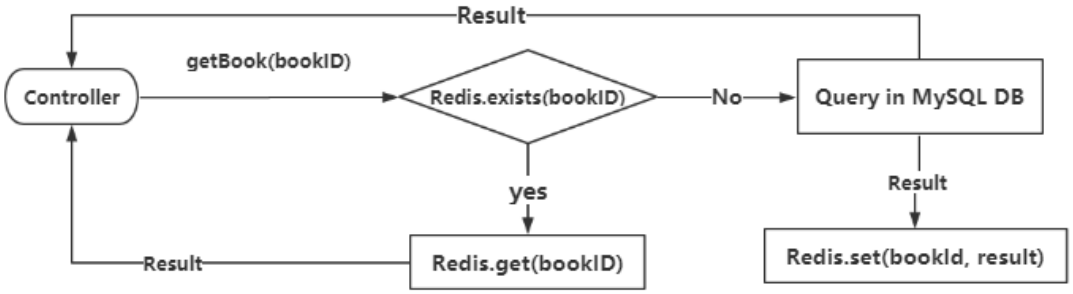

图 4-4 后端架构设计

当 Node Server 端接收到来自小程序端发送的请求之后，首先经过会路由器解析请求 url，然后被分配到对应的控制器中，控制器是实现业务逻辑如登录，获取书籍信息，发表评论等主要场所，当涉及到数据库相关操作的时候，会调用 Model 数据层接口，在 Model 层中分为两个层级，一个是 Redis 缓存层，一个是数据库 Mysql,所有访问 Model 层的调用，都必须经过 Redis Server。查看在 Redis 中是否存在相应的缓存。

在 Redis 缓存层模块中，封装了一个 Redis 操作类，当需要进行 Redis 操作时，只需将该类引用即可。使用 Redis 缓存机制的流程图如图 4-5 所示

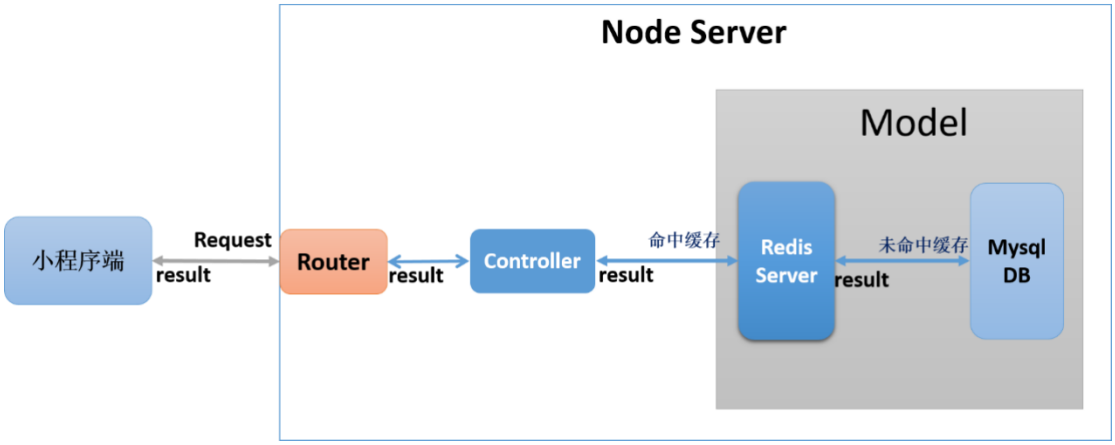

图 4-5 缓存机制流程图

当涉及到数据获取问题，如小说信息获取，小说章节内容获取，小说评论获取，都会经过如图 4-7 的流程图逻辑过程，调用封装好的 Redis 类，进行缓存查

询判断，如果存在，则直接返回缓存内容，如果不存在，则到数据库中进行查询，在数据库查询得到结果之后，将结果存在 Redis Server 中，并返回结果给 Controller 层。

涉及到缓存问题，就不可避免会遇到缓存更新问题，”看书吧”小程序后台主要通过以下两种方式进行缓存更新。

（一）设置缓存过期时间，在设置缓存内容的同时，调用 redis 的 expire 方法设置过期时间，如书籍的作者信息，简介信息等缓存内容。

（二）服务器端主动更新缓存内容，保持数据库和缓存信息的一致性。如当小说更新新的章节内容的时候，这时应主动更新小说章节内容的缓存。

当服务器端进行主动更新缓存时，为了避免由于出现并发操作带来的脏数据问题，缓存更新的策略采用了论文<<Scaling Memcache at Facebook>>中所提出的更新策略 Cache Aside Pattern[13]，该模式进行更新的逻辑流程是：当数据进行更新的时候，先将新更新的数据存储到数据之中，数据存储成功之后，最后再让缓存失效。

4.4　数据库分析与实现

4.4.1 数据库关系 E-R 图

“看书吧”小程序使用 MySql 作为数据库，经过需求和功能分析，为了建立一个完善的书城数据库结构，一共规划 4 个实体，分别是：书籍信息实体，章节信息主体，评论内容主题，用户信息实体，数据关系的 E-R[14] 图如图 4-6 所示

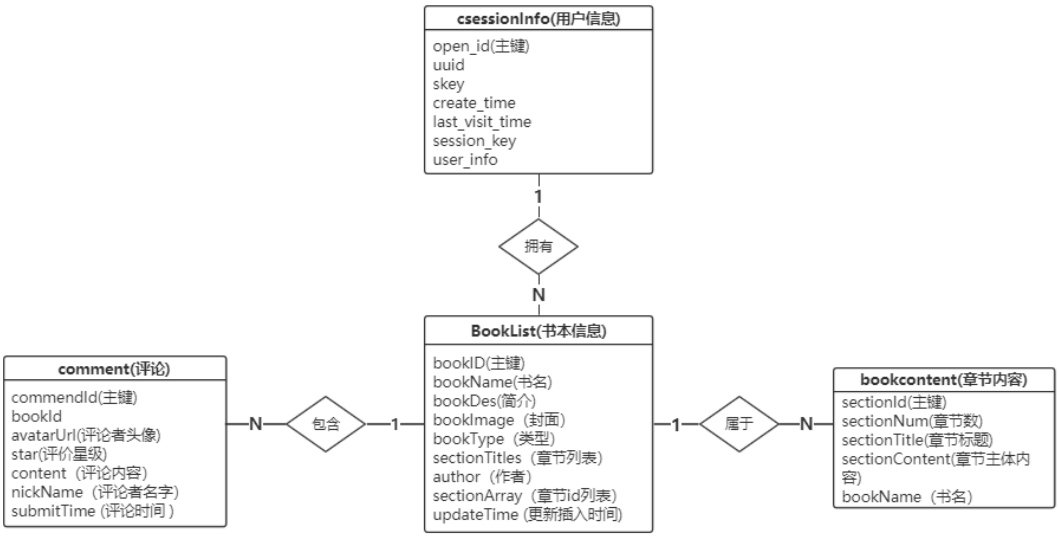

图 4-6 系统数据库 E-R 关系图

4.4.2 数据库表结构设计

用户信息表存储了用户登录之后的个人信息，包括了用户唯一标示 openid，会话密钥 session_key，用户最初登录时间，最近访问时间和用户的个人信息对象，这个对象包含了用户的头像，昵称，性别的信息。如表 4-1 所示:

表 4-1：用户信息表

字段名	类型	是否主键	说明
open_id	Varchar	是	用户唯一标识
Uuid	Varchar	否	用户 id
Skey	Varchar	否	自定义登陆态
create_time	Timestamp	否	登录时间
last_visit_time	Timestamp	否	最近访问时间
session_key	Varchar	否	会话密钥
user_info	Varchar	否	用户信息

书籍信息表存储的是小说的基本信息，包括小说的小说名，小说的封面小说的简介，小说类型，作者，章节列表标题和更新时间。表结构如表 4-2 所示

表 4-2: 书籍基本信息表

字段名	类型	是否主键	说明
bookId	Int	是	书籍唯一 ID
bookName	Varchar	否	书名
bookDes	Text	否	简介
bookImage	Varchar	否	封面
bookType	Varchar	否	类型
sectionTitles	Text	否	章节标题列表
Author	Varchar	否	作者
sectionArray	Text	否	章节 ID 列表
updateTime	Timestamp	否	更新时间

章节内容列表是用来存储小说章节内容的信息，包括章节的标题，章节内容，章节数，小说名等字段，其表结构如表 4-3

表 4-3：章节内容信息表

字段名	类型	是否主键	说明
sectionId	Int	是	章节唯一 ID
sectionNum	Int	否	章节数
sectionTitle	Varchar	否	章节标题
sectionContent	Text	否	章节内容
bookName	Varchar	否	书名

评论表保存了一本书下所有用户对其的评论，它包含了评论书籍的 bookId 用户的头像，评价的等级，评价的内容，用户名和评论的时间，评论表(comment)的表结构如表 4-4

表 4-4: 评论表结构

字段名	类型	是否主键	说明
commentId	Int	是	评论唯一 Id
bookId	Int	否	书籍的 ID
avatarUrl	Varchar	否	用户头像
Star	Int	否	评价的等级
Content	Text	否	评论内容
nickname	Varchar	否	用户名
submitTime	Timestamp	否	提交时间

第五章 系统实现

5.1 登录鉴权模块

微信小程序的一个优势是用户不需要进行显示的注册和登录，通过授权登录，以微信账号的方式即可登录小程序。登录的作用在于让开发者服务器知道当前使用应用的用户是谁？在如 app，浏览器等传统 web 应用中，使用者必须通过注册账号密码，登录输入账号密码的方式来告知服务器登录用户对象时谁？然而，在微信小程序中，我们可以借助微信服务器来完成这个操作[16]，微信服务器发送请求获取当前用户的唯一标识 OpenId，其中每个用户相对于每个微信应用的 OpenId 是唯一的。

由于小程序端没有 cookie 机制，因此小程序的登录态维护与传统的 web 应用有些不同，大致的流程如下：

1. 小程序前端调用接口 wx.login()获取临时的登录凭证 code，并调用 wx.request 将 code 发送到开发者服务器中
2. 开发者服务器接收到临时凭证 code 之后，发送携带临时凭证 code 的 http 请求到接口 auth.code2Session,用 code 换取用户唯一标示 OpenID 和会话密钥 session_key.
3. 在开发者服务器中，将 openid+session_key 关联加密生成一个 skey，将 skey 作为 key，对应的 openid+session_key 作为 value，并设置过期时间为 7 天，存入 redis 缓存中，同时存入数据库中，随后，将 skey 返回给小程序端
4. 小程序端将接收到 skey 存入缓存中，在之后小程序的每一个请求的头部都都携带自定义登录态 skey
5. 开发者服务器接收到小程序端的请求之后，首先会拿到请求头部中的自定义登录态 skey，然后在 redis 缓存中进行查询，查询是否有对应的 openid 和 session_key，并判断登录态是否过期，如果可以查到对应的 openid 和 session_key，并且仍然在有效时期内，则允许继续执行业务

逻辑，返回业务数据；否则，则不允许继续执行，告知小程序端，通知用户进行登录操作，重复执行上述的 1-5 操作。登陆鉴权模块时序图如图 5-1 所示

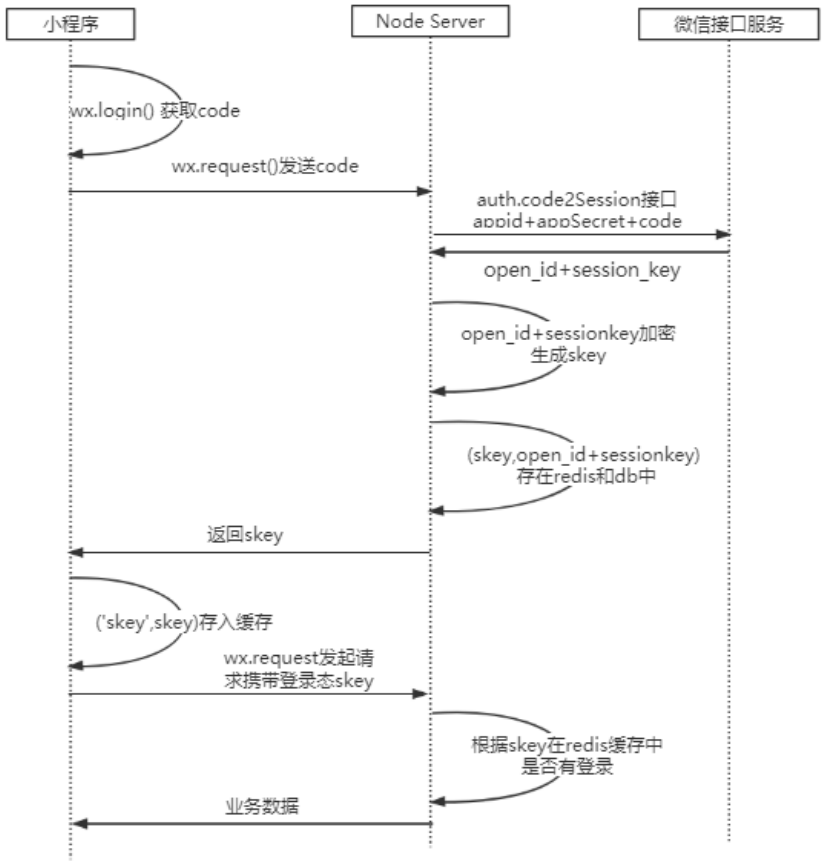

图 5-1 登录鉴权时序图

5.2 阅读器模块

阅读器模块是书城小程序中最主要的功能更，当用户点击一本小说开始进行阅读的时候，即会跳转到阅读器界面。阅读器模块按照功能划分为两个个子模块分别是：章节内容查看，阅读器风格设置和目录查看

5.2.1 章节内容查看

当用户点击某本书籍的时候，会跳转到阅读界面进行小说阅读，而此时看到

的小说章节数有两种情况：第一种是用户之前曾经看过这本小说，此时应该直接跳转到用户上次阅读的地方，第二种是用户从未看过这本小说，此时应该跳转到小说的第一章。章节内容的核心流程图如图 5-2 所示

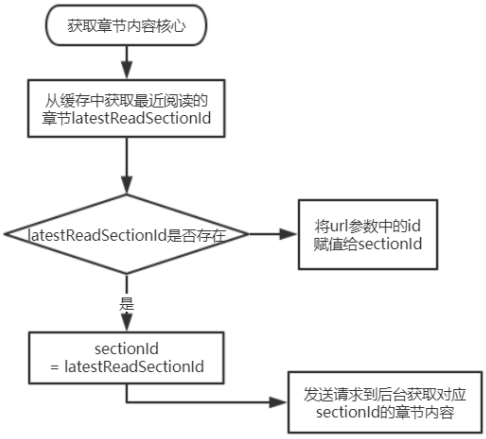

图 5-2 章节内容核心流程图

从代码的角度来实现，也就是：首先必需从缓存中查找是否有该本书的最近阅读章节数，若有，即将 sectionId 设为缓存章节数，若没有相应的缓存记录，则从 query 参数中获取 sectionId，而在书籍详情页跳转到阅读器页时传的参数 sectionId 一般默认值是小说的第一章。因此实现了即可保留阅读记录，也可正常阅读小说的功能。

通过 sectionId，前端会向后端发送章节内容获取请求，获取章节内容，如图 5-3 所示

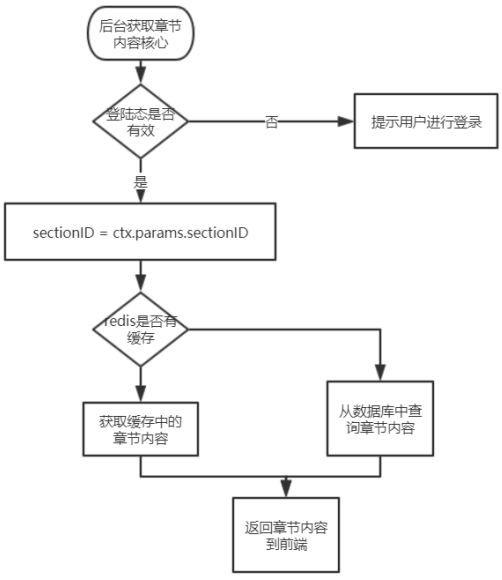

图 5-3 章节获取核心流程图

5.2.2 阅读器风格设置和章节目录查看

对于一个阅读器来说，必须要做到一点是章节目录的查看，当用户点击目录按钮，会从左侧弹出目录，用户可以在弹出的章节目录中，选中自己想要阅读的章节，相对应在内容区会跳转到对应的章节内容。

为了提高界面的流畅性，目录栏的出现是以动画的效果从左侧弹出。并且当用户 touch 目录按钮的时候，目录栏出现，而当用户 touch 屏幕内容的时候，目录栏会自动收起。一本书的目录的高度是屏幕的高度，高度是一致，但是目录章节数可能很长，因此在使用了<scroll-view/>标签，使得目录可以在纵向上进行滚动，这样即可阅读到所有的章节标题。

为了能让用户有更好的阅读体验，看书吧小程序设置了两种阅读模式，一种是日间模式，一种是夜间模式，同时，用户可以进行个人喜好调整字体的大小。为了实现界面风格的切换，首先需要将日间模式和夜间模式的 styles 属性保存在 data 属性中，当触发阅读器界面风格日间模式按钮的时候，就会将对应的样式应用在标签元素上。同时阅读器应该具有保存用户之前离开当前界面时候的风格选择，即用户如果离开时的模式是夜间模式，字体大小是 24px，则再次回到该界面的时候，应保持一致， 进入阅读器界面一开始，即从小程序缓存中获取字体大小和风格模式，并将获取到的属性值应用到界面上，应用对应的样式

5.3 我的书架模块

我的书架模块，展示的是用户加入到书架中的书籍，所以首先需要从缓存中获取书架小说列表。在小程序缓存中有一个字段为存 bookshelfItems”，其存储的是被加入书架中的书籍信息，包括其 bookId， 封面等信息，缓存如图 5-4 所示

图 5-4 缓存信息

对书架上的书进行增添：给所有的 bookshelfItems 加上一个属性 type，type = 1 表示正常无编辑状态，type = 2 表示此时处于编辑状态，封面右下角出现编辑图标，type = 3 表示该元素为空，该元素应该是增加书籍的封面。

长按屏幕，会变成编辑状态，此时所有 bookshelfItems 元素的 type 属性会变为 2，此时所有封面的右下角会出现编辑图标，根据 type 的值，当为 1，则跳转到书籍详情页，当为 2，则表示选中需要被删除，为 3，跳转到书城首页。

5.4 书城首页模块

书城首页模块分为随机推荐，精选，搜索，排行榜信息三个部分。随机推荐后台会会根据用户的书架中书籍的类型推荐相同类的书籍。

排行榜信息模块，网络爬虫的时候，将小说发布网站上的排行榜信息爬取下来，同时保存对应小说的基本信息，以及小说对应的章节内容模块，因此在进行排行榜信息展示的时候，一共需要发送 2 个请求，分别是排行榜类别获得，书籍详细信息获得，涉及到两个数据库表的 IO 操作，分别是排行榜表(ranking_list)和书籍详细信息表(bookLlist).

搜索模块，在搜索功能中，用户点击搜索会跳到搜索界面，根据用户填入输入框的信息，进行模糊查询，并将搜索记录保存在缓存中，以历史搜索的方式展

示出来。

搜索模块中的流程如下：

(1) 在小程序缓存中查找是否有 history 关键字的缓存，将曾经搜索的关键字展示在历史搜索视图中

(2) 用户在搜索输入框中输入查询书籍，前端发送 post 请求到后端，后端拿到前端 post 过来的查询条件，到数据库中进行模糊查询，返回数据列表给前端

（3）小程序端拿到查询数据之后，展示在搜索 view 中。

流程图如 5-5 所示

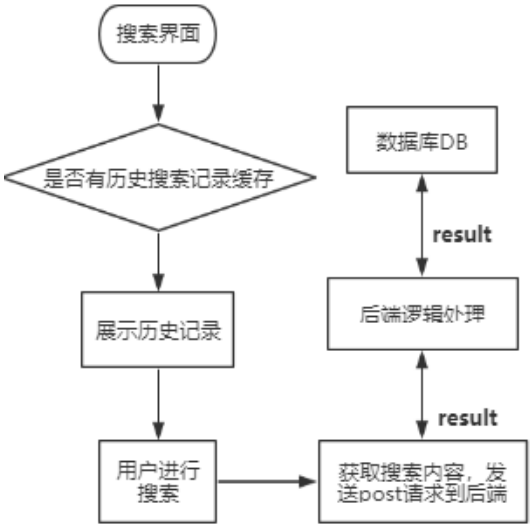

图 5-5 搜索模块流程图

无论是随机推荐列表，排行榜信息列表，当用户 touch 书本封面的时候，都会触发 touch 事件，跳转到书籍详情页中.

5.5 书籍详情模块

书籍详情页一共分为四个部分，分别是小说基本信息展示模块，发表评论模块，评论列表展示以及跳转阅读部分。小说基本展示模块：即是数据库中书籍基本信息表的获取。

发表评论模块，当用户点击星星等级的时候，会弹出评论框，用户在评论框中输入对该本书的评论内容，当触发确认事件时，前端将评论内容作为 data 发送请求到后端，保存到数据库 Comment 表中，发表评论的核心代码流程图如图 5-6 所示

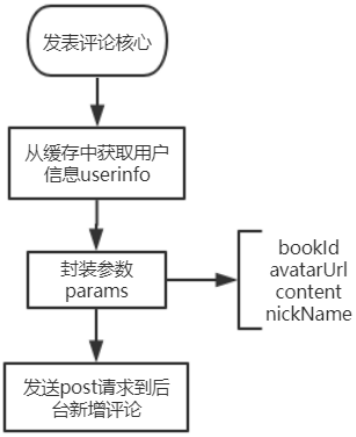

图 5-6 发表评论核心流程图

需要新增一条评论信息到表 Comment 中，首先应该封装好 data，将用户信息，bookId, avatarUrl，star，content 封装在 params 参数中，然后发起 post 请求到后端，将评论内容新增到数据库表 comment 中。

评论列表展示模块，主要涉及到的是数据库表 comment 的内容获取。根据 bookId 获取关于这本书的全部评论。在展示评论内容，涉及到时间展示的时候，由于数据库中评论时间是以时间戳的形式存在，因此在拿到评论时间的时候，需要进行时间格式的转化。

第六章系统测试

6.1 缓存性能测试

在后端架构设计中，加入了缓存层，当前端发送请求到后端获取数据时，响应速度应该比未加入缓存时的快，通过进入书籍详情页，前端向后台发送请求时，两种设计响应速度对比如下如图 6-1：

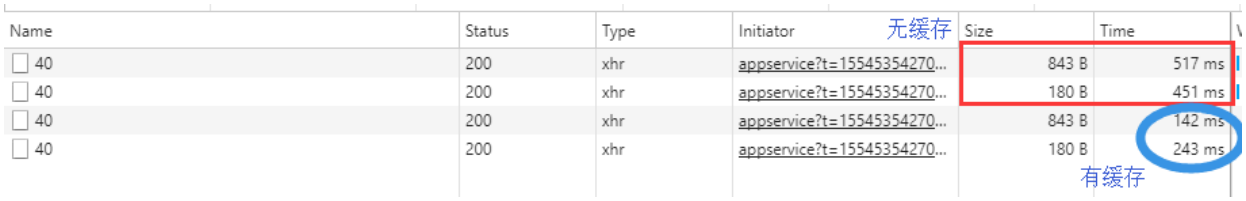

图 6-1 响应速度对比图

图 6-1 中第一行和第二行表示请求 bookId 为 40 的书籍基本信息和 bookId 为 40 的所有评论，同样第三和第四行请求的内容分别和第一行和第二行的一致，通过控制变量法的方式，比较两者的响应时间，如图 6-1 红框圈出的为无缓存时的响应时间，而蓝色框圈出的为有缓存时的响应时间，显然加入缓存层后响应速度加快，性能得到了优化。

6.2 书籍详情页

在书城首页中选择一本书籍，会跳转到书城详情页，在书籍详情页应该可以成功展示对应书籍的基本详细信息，比如小说书名，小说作者，小说的封面和小说的内容简介，如图 6-2（a）

图 6-2（a）书籍详情页

由于一般来说，一个小说的简介内容会偏长，如果全部放在界面中，会影响美观，因此，“看书吧”设置了只有点击简介内容，会弹出一个弹框展示所有的简介内容如图 6-2(b)．展示所有用户对这本小说的评论如图 6-2(c)所示

图 6-2(b)简介

图 6-2（c）评论

用户发表评论时，用户可以点评小说的评价星级，同时评论内容和用户信息保存到数据库中,测试结果如图 6-2(d)

commentId	bookId	avatarUrl	star	content	nickName	submitTime
2	38	https://wx.qlo	3	不错的一本书	小优	2019-04-06 20:0
3	38	https://wx.qlo	5	很有民国的味	兰小野	2019-04-06 20:1
4	45	https://wx.qlo	3	男主很有意思	美美丽	2019-04-06 20:1
5	38	https://wx.qlo	3	这是一部不错	菜荔枝	2019-04-06 20:0

图 6-2（d）评论内容

点击返回首页，能够返回到书城首页，点击开始阅读，跳转到阅读器界面进行小说内容的阅读。

6.3　阅读器界面

点击阅读书籍会跳转到阅读器界面，点开一个未曾阅读过的小说，阅读器显示的是小说第一章的内容如图 6-3（a）所示，点开一个曾经阅读到第三章的小说，跳转到阅读器时，应该显示第三章的内容如图 6-3（b）

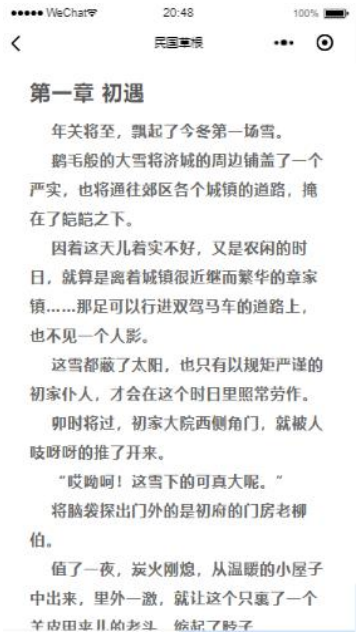

图 6-3（a）阅读器

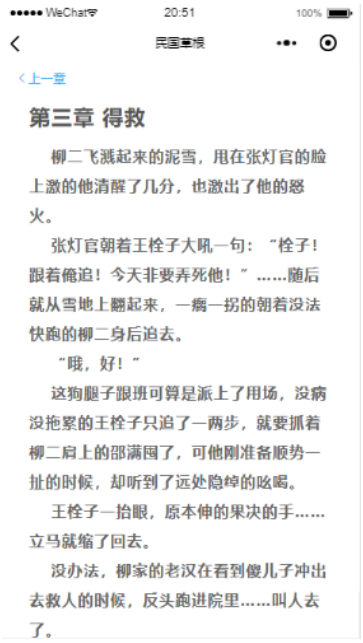

图 6-3（b）阅读器

切换夜间风格如图 6-3（c），　日间风格与调整字体大小 6-3（d）

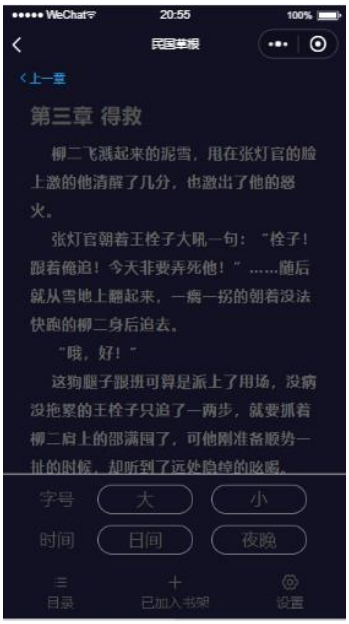

图 6-3（c）夜间模式

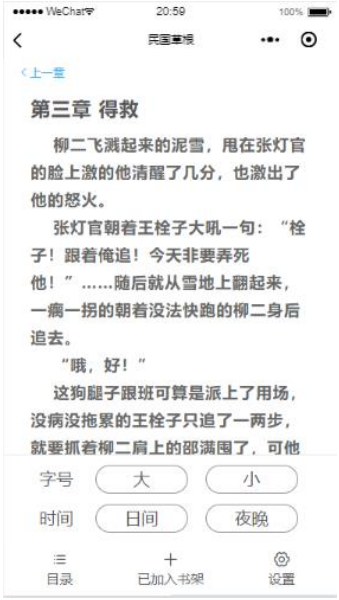

图 6-3（d）日间模式

实现目录查看如图 6-3（e）

图 6-3（e）目录

6.4　书城首页

书城首页模块，提供用户书籍选择如图 6-4(a),进行全局搜索，历史搜索展式如图 6-4（b），搜索结果展示如图 6-4（c）

图 6-4（a）首页

图 6-4（b）搜索

图 6-4（c）搜索结果

6.5　我的书架模块

在我的书架模块中，首先具有陈列之前用户加入书架的书籍，如图 6-5（a），长按屏幕进入编辑状态，可选择要删除的书本，并显示删除数，成功删除选择的书籍，如图 6-5（b）

图 6-5(a)　书架

图 6-5(b)　编辑书架

6.6　个人中心模块

用户需进行主动登录操作，登录后可以看到最近阅读的书籍等信息，如图 6-6 个人中心所示。

图 6-6　个人中心

第七章 总结与展望

微信小程序现如今已经成为大多数公司拓展商业圈的重要应用，基于小程序的书城系统让人们的阅读变得更加便利，在本课题“基于 Mpvue 的在线书城小程序”的设计与开发中，一共完成了以下几项工作：

1. 学习研究 Mpvue 框架技术，使得书城系统可以同时作为小程序项目和 h5 项目在多个平台中使用，代码得到高效的复用。
2. 学习研究 Koa2 后端框架技术，构建后台 api 服务，加入缓存层，实现请求响应速度的提升
3. 构建全栈书城系统，实现用户浏览书城，发表评论，查看图书介绍，阅读小说等功能。

“看书吧”小程序的开发涉及到许多理论和技术，虽然目前书城应用已经开发完毕并且已投入使用，但是依然存在着一些新的问题，需要在不断地用户实践中进行修正和完善，同时，“看书吧”小程序也需要多方面提高使用性能，成为当下人们手中更加友好便利的掌上阅读应用。

参考文献：

[1]. Ethan Marcotte . Responsive Web Design[EB/OL]. https://alistapart.com/article/reponsive-web-design. 2010.

[2]. 郭全中. 小程序及其未来[J]. 新闻与写作, 2017(3):28-30.

[3]. 一种开发和执行均衡高效的 Web 前端框架的研究与实现[D]. 北京邮电大学, 2015.

[4]. 美团技术团队. 开源框架 mpvue 解析[EB/OL]. https://blog.csdn.net/MeituanTech/details. 2018.

[5]. 刘博文. 深入浅出 Vue.js [M].人民邮电出版社. 2019.

[6]. 陈明，李猛坤. 一种基于扩展 MVVM 模式的面向服务软构件模型 [J]. Science Technology and Engineering. 2011. 1671—1815.

[7]. 阮一峰. Koa 框架教程[EB/OL]. http://www.ruanyifeng.com/blog/2017/08/koa.html. 2017.

[8]. Nicholas C.Zakas. Professional JavaScript for Web Developers 3rd Edition[M]. 人民邮电出版社.2016.

[9]. 李仕伟, 周坤, 刘新蕊,等. MySQL 数据库优化技术[J]. 信息与电脑(理论版), 2016(12):173-174.

[10]. 邱书洋. Redis 缓存技术研究及应用[D]. 2016.

[11]. Gao X, Fang X. High-Performance Distributed Cache Architecture Based on Redis[J]. Lecture Notes in Electrical Engineering, 2014, 270:105-111.

[12]. Tilkov S , Vinoski S . Node.js: Using JavaScript to Build High-Performance Network Programs[J]. IEEE Internet Computing, 2010, 14(6):80-83.

[13]. Nishtala R , Fugal H , Grimm S , et al. Scaling Memcache at Facebook[C]// Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2013.

[14]. Abraham Silberschatz, Henry F. Korth, S. Sudarshan. 数据库系统概念[M]. 2008.

[15]. 一斤代码.微信小程序中用户登录和登录态维护[EB/OL]. https: //www.jianshu.com/p/c5f6c98b2685. 2018.

致　谢

从去年 10 月份定下毕业论文题目，到现在完成所有论文设计将近半年的时间里，离不开我的毕业设计指导老师张永民老师的帮助，从论文的选题，到论文初稿的修改，张老师一直给予了我不少帮助与鼓励。论文选题期间，张老师不仅提供了大量的选题方向，而且也允许我自己定题，同时给予意见。因此，在这里非常感谢张永民老师，感谢他在我创作的过程中所给予的帮助。同时，也要感谢师兄师姐们，感谢他们在开发过程中提供的资料，让我可以更加顺畅地完成设计开发。

毕业论文（设计）成绩评定记录
Grading Sheet of the Graduation Thesis (Design)

指导教师评语
Comments of Supervisor：

成绩评定
Grade：

指导教师签名
Supervisor Signature ： Date:

答辩小组意见
Comments of the Defense Committee：

成绩评定
Grade：

签名： Date:
Signatures of Committee Members

院系负责人意见
Comments of the Academic Chief of School：

成绩评定
Grade：

签名 院系盖章
Signature： Stamp: Date: