****

系统设计报告

题 目： 出版社图书订购系统微信小程序

院（系）： 计算机与信息安全学院

专 业： 计算机科学与技术（卓越工程师班）

组 长： 1900300721 梁宏溢

组 员： 1900300720 梁海全

1900300716 黄世营

1900300932 吴俊杰

指导老师： 孙晋永

2022年7月24日

目录

[1 总体设计 2](#_Toc109856418)

[1.1概述 2](#_Toc109856419)

[1.2 小程序前端 2](#_Toc109856420)

[1.2.1 前端功能模块 2](#_Toc109856421)

[1.3 小程序后端 3](#_Toc109856422)

[2 数据库设计 3](#_Toc109856423)

[2.1 出版社管理员表 3](#_Toc109856424)

[2.2 图书信息表 3](#_Toc109856425)

[2.3 用户信息表 4](#_Toc109856426)

[2.4 订单信息表 4](#_Toc109856427)

[2.5 购物车信息表 5](#_Toc109856428)

[2.6 数据库E-R图 5](#_Toc109856429)

[3 项目详细设计 6](#_Toc109856430)

[3.1 系统设计 6](#_Toc109856431)

[3.2 原型设计 7](#_Toc109856432)

[3.3 项目技术栈 11](#_Toc109856433)

[3.3.1 前端技术栈 12](#_Toc109856434)

[3.3.2 后端技术栈 12](#_Toc109856435)

[3.3.3 协作与版本控制工具 13](#_Toc109856436)

# 1 总体设计

## 1.1概述

本小程序总体参照主流电商APP（淘宝、当当、京东等）功能来进行设计，包括前端页面展示和交互、后端API数据接口服务、图书和订单信息数据存储。设置图书商品分类展示功能、单个商品加入购物车功能、模拟支付功能、查看订单功能、添加修改收货地址功能、出版社图书管理功能、出版社订单发货功能等其他功能模块。

## 1.2 小程序前端

### 1.2.1 前端功能模块

前端主要获取后端数据库中的数据，并根据各功能模块实现相应页面的展示与交互，本项目包括小程序的所有页面和一个后台管理系统页面。主要功能模块和页面如下：

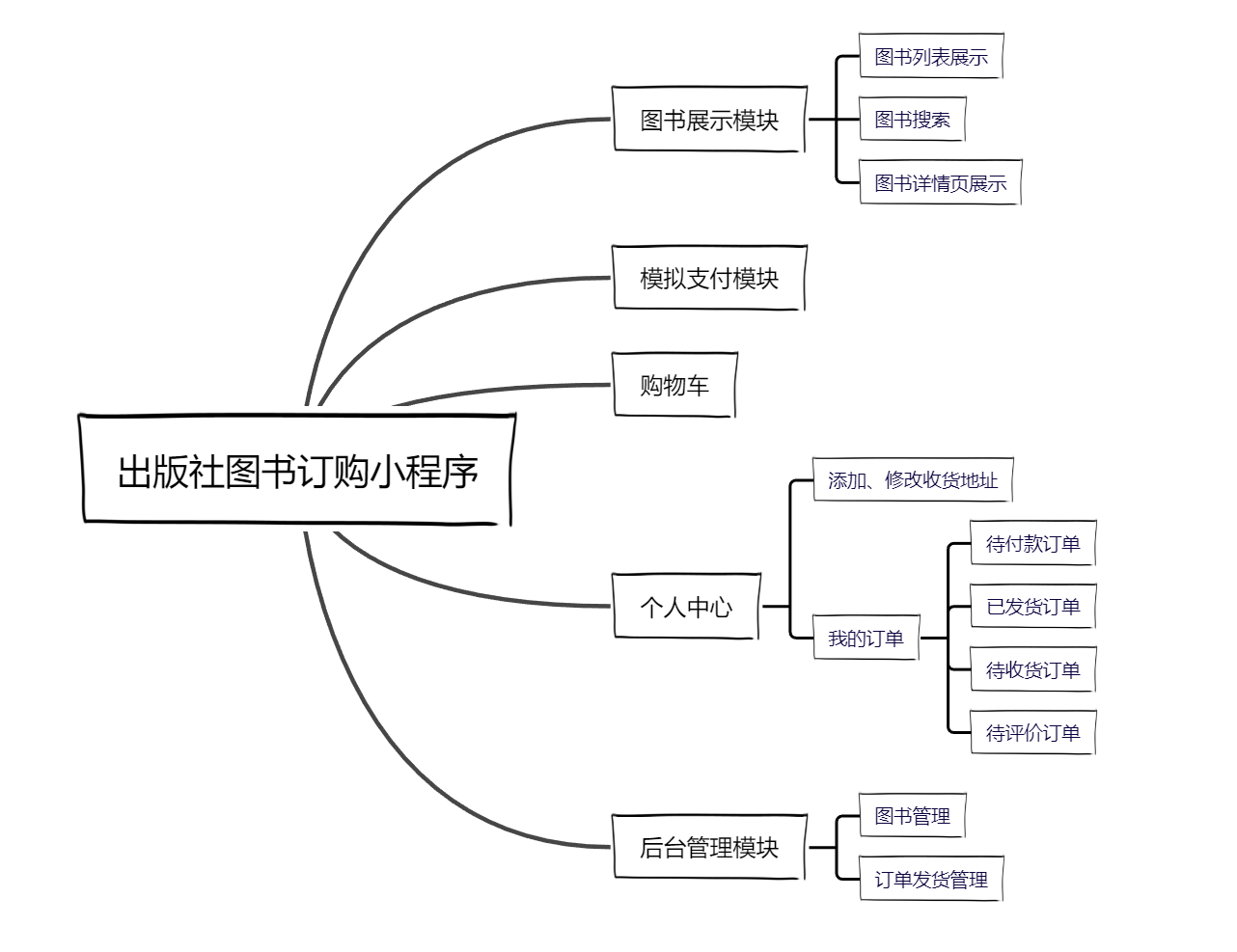


图1.1 出版社图书订购小程序功能模块结构图

## 1.3 小程序后端

后端主要是对数据库进行增删改查，提供相应的HTTP接口供前端调用以完成数据的可视化展示以及相应的交互。

后端需完成的数据接口包括但不限于以下全部：

* 登录接口（直接调用微信提供的API）
* 出版社图书信息的更新、删除接口
* 用户信息的更新、删除接口
* 订单信息的更新、删除接口
* 购物车更新、删除接口
* 图书搜索接口
* 订单搜索接口
* 出版社管理员信息更新、删除接口
* 出版社订单发货接口
* 模拟支付接口
* 模拟退款接口
* …

# 2 数据库设计

## 2.1 出版社管理员表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 必须 |
| 管理员账号 | adminID | Varchar(36) | 主键 |
| 用户名 | adminName | Varchar(36) | 索引 |
| 密码 | password | Varchar(16) | 是 |
| 电话号码 | phone | tinyint | 是 |
| 电子邮箱 | email | Varchar(36) | 否 |

## 2.2 图书信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 必须 |
| 图书ID | bookID | Varchar(36) | 主键 |
| 图书名称 | bookName | Varchar(36) | 是 |
| 图书作者 | author | Varchar(36) | 是 |
| 作者简介 | authorBrief | Varchar(255) | 是 |
| 图书简介 | briefInfo | Varchar(255) | 是 |
| 图书目录 | catalogue | Varchar(255) | 否 |
| 图书价格 | price | Int(4) | 是 |
| 图书出版社 | publisher | Varchar(36) | 是 |
| 图书出版年份 | year | Varchar(36) | 是 |
| 图书短评 | evaluation | Varchar(255) | 是 |
| 图书ISBN | isbn | longInt(20) | 是 |

## 2.3 用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 必须 |
| 用户id | openID | Varchar(36) | 主键 |
| 用户名 | username | Varchar(16) | 索引 |
| 联系电话 | userPhone | Int(11) | 否 |
| 收货地址 | address | Varchar(36) | 是 |
| 购物车id | shoppingCartID | Varchar(16) | 是 |

## 2.4 订单信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 必须 |
| 订单id | orderId | Varchar(36) | 主键 |
| 创建时间 | createTime | Varchar(36) | 索引 |
| 订单用户ID | orderUserId | Varchar(36) | 是 |
| 发货时间 | deliverTime | Varchar(36) | 是 |

## 2.5 购物车信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段代码 | 字段类型 | 必须 |
| 购物车id | shoppingCarId | Varchar(36) | 主键 |
| 图书ID | bookId | Varchar(36) | 是 |
| 购物车用户ID | shoppingCarUserId | Varchar(36) | 是 |

## 2.6 数据库E-R图

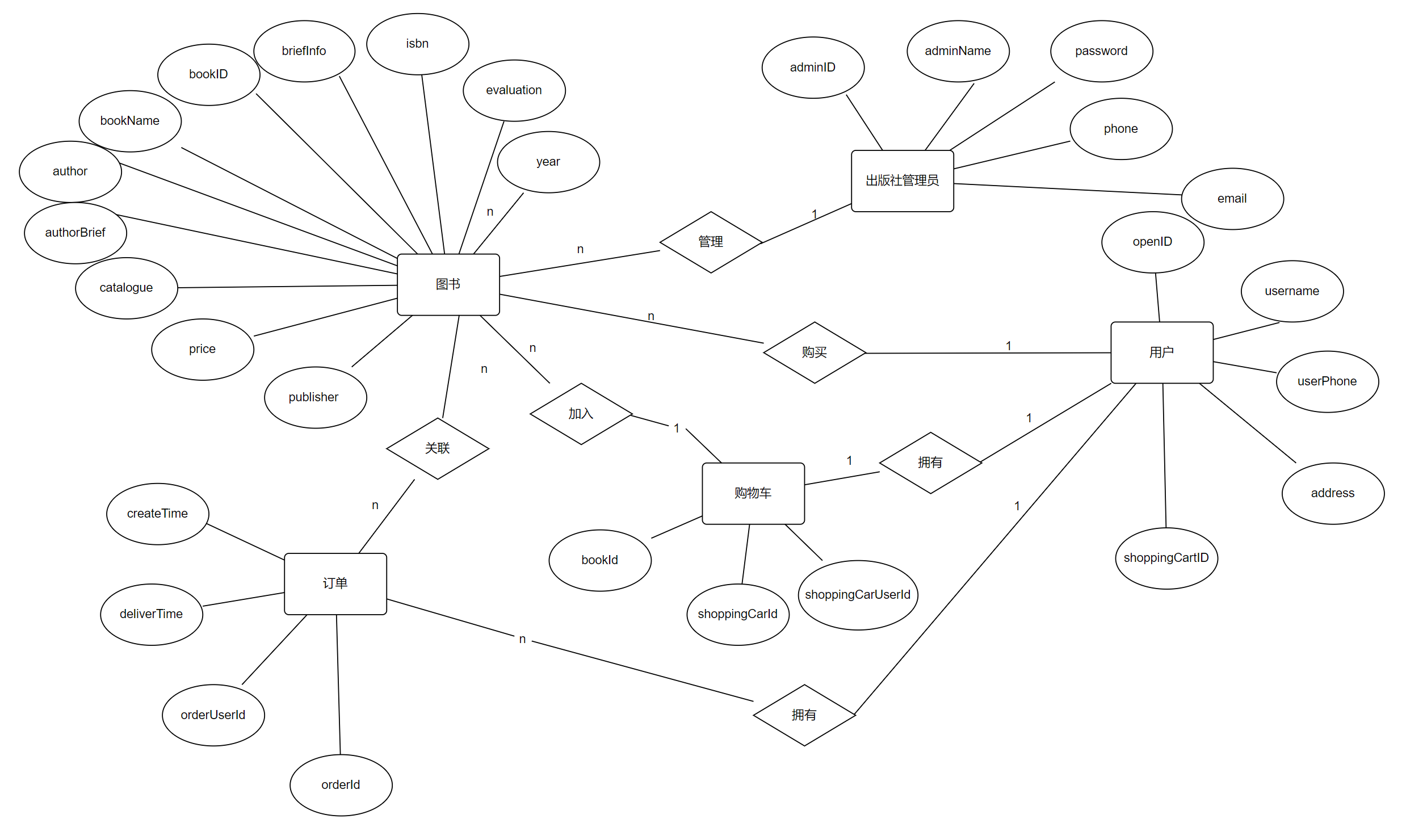


图2.1 出版社图书购买系统E-R图

本系统的数据库主要由4个数据表组成，分别是出版社管理员表、图书信息表、用户信息表、订单信息表、购物车信息表。其中各个表的字段如以上各数据表所示。一个用户可以购买多本图书，一个管理员可以管理多本图书，一个用户拥有一辆购物车，一辆购物车可以关联多本图书。一个订单可以关联多类图书，一类图书也可以关联多个不同的订单。一个用户拥有多份订单。

# 3 项目详细设计

## 3.1 系统设计

小程序的主要业务流程如下：

用户打开小程序，展示图书列表和图书搜索框，用户可以直接在图书列表中添加想要的图书到购物车中，但是要先授权登录。当用户登录后，才可以将相应商品加入到购物车中，用户也可以使用图书搜索功能搜索相关的图书，如果系统能匹配到用户搜索的图书，则以列表的形式展示给用户。用户可以点击图书进入图书详情页，查看图书的其他相关信息，在图书详情页的最下方提供加入购物车和购买按钮，同样，购买功能也需要用户授权登录。

用户点击购买之后开始进入模拟支付流程，如果用户在该流程中未完成支付，则订单将被加入到用户的待付款订单中，用户可以在「我的订单」中取消该订单。如果用户完成了支付，则系统将通知出版社发货，订单完成，后续用户还可以模拟退货退款。

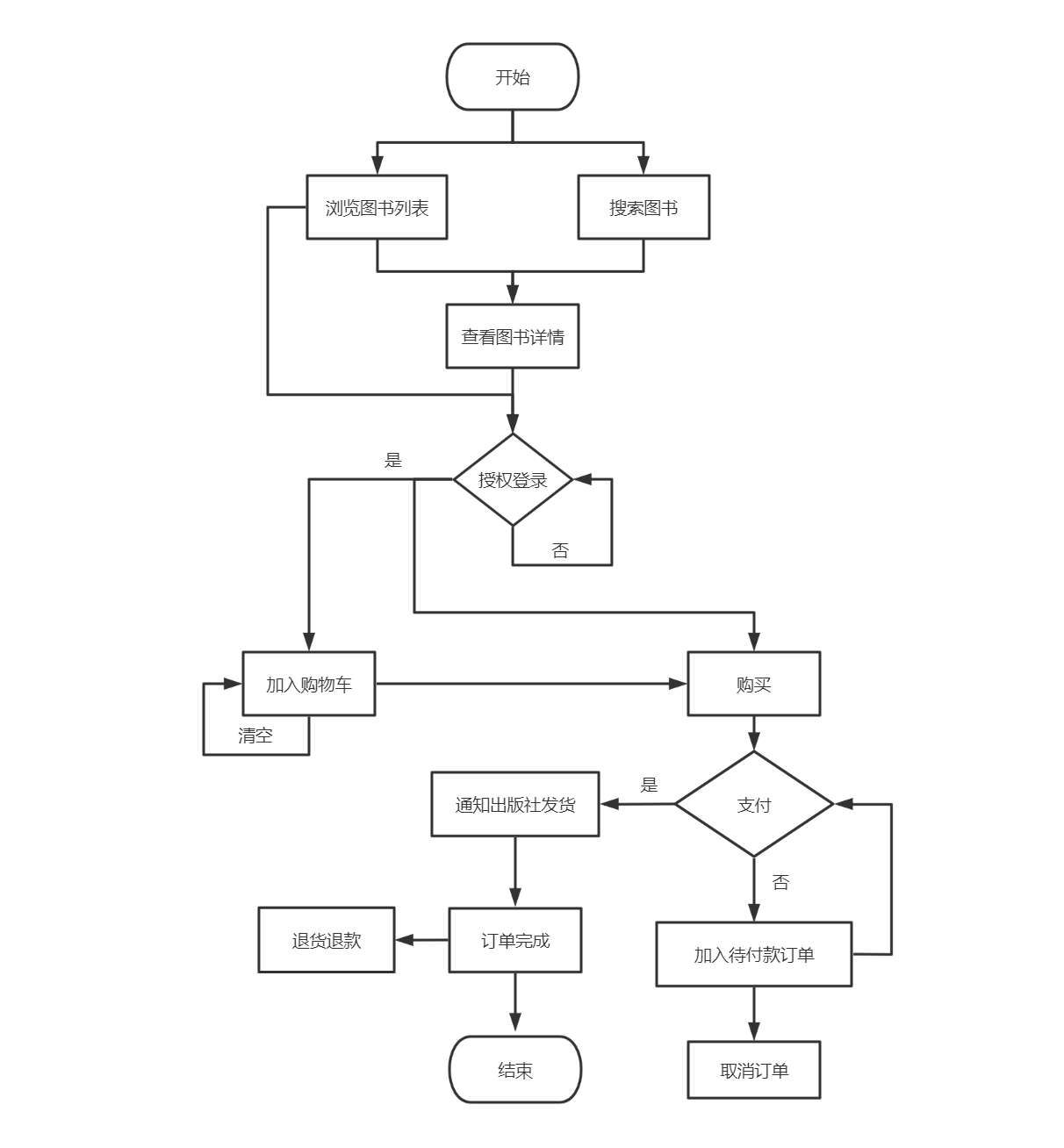


图3.1系统业务流程图

## 3.2 原型设计

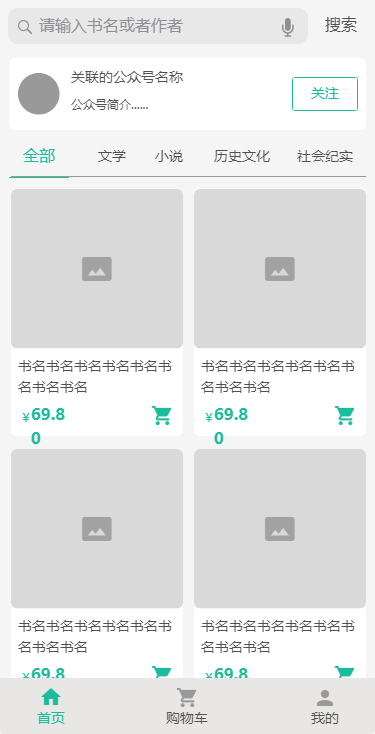
小程序原型参照各大电商平台的风格样式设计。

图3.2小程序首页

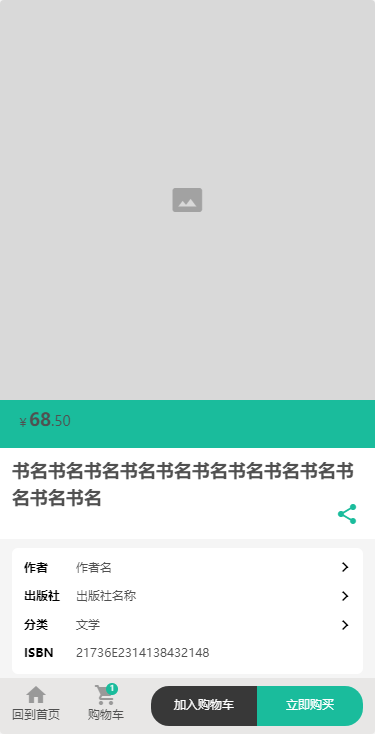


图3.3 图书详情页-商品

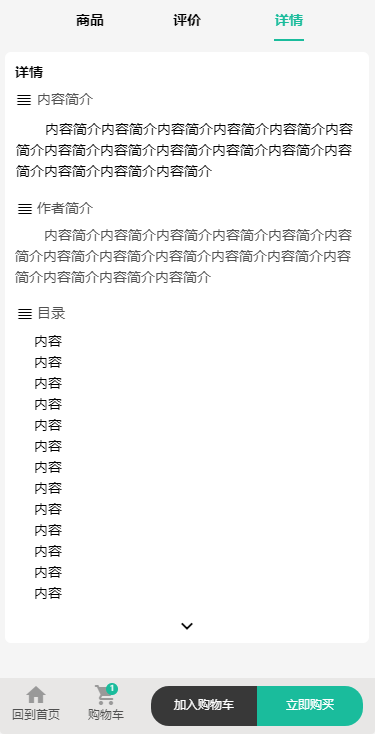


图3.5图书详情页-详情

图3.4图书详情页-评价



图3.7个人中心

图3.6购物车

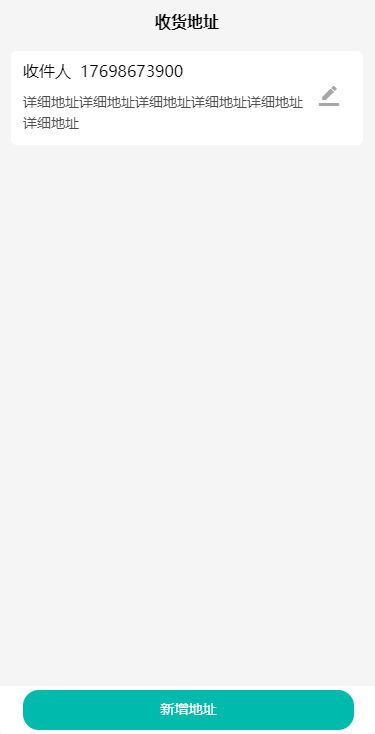


图3.9收货地址

图3.8 所有订单

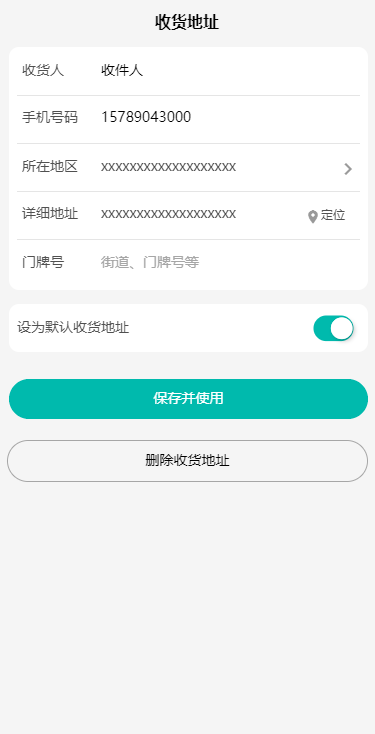


图3.10编辑收货地址

## 3.3 项目技术栈

核心思想是前端HTML页面通过AJAX调用后端的RESTFUL API接口并使用JSON数据进行交互。这样可以把前端与后端独立起来去开发，放在两个不同的服务器，需要独立部署。两个不同的工程，两个不同的代码库，不同的开发人员，前后端工程师需要约定交互接口，实现同步开发。开发结束后需要进行独立部署，前端通过接口来调用后端的API，前端只需要关注页面的样式与动态数据的解析和渲染，而后端专注于具体业务逻辑。

1、彻底解放前端。前端不再需要向后台提供模板或是后台在前端HTML中嵌入后台代。

2、提高工作效率，分工更加明确。前端只关注前端的事，后台只关心后台的活，两者开发可以同时进行，在后台还没有时间提供接口的时候，前端可以先将数据写死或者调用本地的JSON文件即可，页面的增加和路由的修改也不必再去麻烦后台，开发更加灵活。

3、局部性能提升。通过前端路由的配置，我们可以实现页面的按需加载，无需一开始加载首页便加载网站的所有的资源，服务器也不再需要解析前端页面，在页面交互及用户体验上有所提升。

### 3.3.1 前端技术栈

开发语言：

* WXML
* WXSS
* JavaScript

前端框架：

* Vue.js
* MINA

第三方库：

* vue-cookies：用于在 vue 中处理浏览器的 cookie 操作
* Axios：一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中，用来处理 vue 中的网络请求
* Echarts：前端可视化图表库
* Vue-router：Vue 官方路由库

开发工具：

* Visual Studio Code
* 微信开发者工具

### 3.3.2 后端技术栈

后端开发语言

* Java

后端框架

* SpringBoot: 快速构建JavaWeb应用程序
* Mybatis: 封装JDBC，以方便地增删改查数据库
* AlibabaDruid：用于快速生成与数据库的连接，提供实时的数据查询和低时延的数据插入

数据库

* mysql

### 3.3.3 协作与版本控制工具

版本控制工具：

* Git

远程代码仓库：

* Github