# 目 录

1. **绪论 1**
   1. [研究的背景及意义 1](#_bookmark0)
      1. [选题的背景 1](#_bookmark1)
      2. [国内外研究现状 1](#_bookmark2)
      3. [研究的意义 2](#_bookmark3)

[1.2 系统目标 2](#_bookmark4)

1. [**需求分析 3**](#_bookmark5)

[2.1 业务需求 3](#_bookmark6)

[2.1.1 主要业务流程 3](#_bookmark7)

[2.2 功能需求 5](#_bookmark8)

[2.2.1 角色分析 5](#_bookmark9)

[2.2.2 业务功能 5](#_bookmark10)

[2.3 非功能需求 15](#_bookmark11)

[2.3.1 环境需求 15](#_bookmark12)

[2.3.2 性能需求 15](#_bookmark13)

[2.3.3 安全需求 15](#_bookmark14)

1. [**总体设计 16**](#_bookmark15)
   1. [系统设计的原则 16](#_bookmark16)
   2. [系统体系结构设计 16](#_bookmark17)
   3. [系统功能结构设计 16](#_bookmark18)
2. [**数据库设计 19**](#_bookmark19)
   1. [概念结构设计 19](#_bookmark20)

[4.1.1 设计思路 19](#_bookmark21)

[4.1.2 E-R 图 19](#_bookmark22)

* 1. [逻辑结构设计 22](#_bookmark23)

[4.2.1 设计思路 22](#_bookmark24)

[4.2.2 逻辑模型 23](#_bookmark25)

* 1. [物理结构设计 25](#_bookmark26)

[4.3.1 存取方式 25](#_bookmark27)

[4.3.2 存储结构 25](#_bookmark28)

1. [**界面设计 28**](#_bookmark29)
   1. [界面关系图或工作流图 28](#_bookmark30)
   2. [界面设计成果 29](#_bookmark31)

[5.2.1 主界面 29](#_bookmark32)

* + 1. [轮播图子界面 30](#_bookmark33)
    2. [专题推荐子界面 31](#_bookmark34)
    3. [图书分类子界面 32](#_bookmark35)
    4. [图书详情子界面 33](#_bookmark36)
    5. [购物车子界面 34](#_bookmark37)
    6. [个人信息子界面 35](#_bookmark38)
    7. [地址管理子界面 36](#_bookmark39)
    8. [订单详情子界面 37](#_bookmark40)

1. [**详细设计 38**](#_bookmark41)
   1. [系统主要功能模块介绍 38](#_bookmark42)
   2. [地址信息模块设计 38](#_bookmark43)
      1. [功能结构设计 38](#_bookmark44)
      2. [功能类设计 39](#_bookmark45)
      3. [顺序图设计 41](#_bookmark46)
      4. [核心处理流程 41](#_bookmark47)
   3. [轮播图模块设计 42](#_bookmark48)
      1. [功能结构设计 42](#_bookmark49)

[6.3.2 功能类设计 42](#_bookmark50)

[6.3.3 顺序图设计 43](#_bookmark51)

[6.3.4 核心处理流程 43](#_bookmark52)

* 1. [专题管理模块设计 44](#_bookmark53)
     1. [功能结构设计 44](#_bookmark54)

[6.4.2 功能类设计 44](#_bookmark55)

[6.4.3 顺序图设计 44](#_bookmark56)

[6.4.4 核心处理流程 45](#_bookmark57)

* 1. [图书管理模块设计 47](#_bookmark58)
     1. [功能结构设计 47](#_bookmark59)

[6.5.2 功能类设计 47](#_bookmark60)

[6.5.3 顺序图设计 48](#_bookmark61)

[6.5.4 核心处理流程 48](#_bookmark62)

* 1. [订单详情模块设计 49](#_bookmark63)
     1. [功能结构设计 49](#_bookmark64)

[6.6.2 功能类设计 50](#_bookmark65)

[6.6.3 顺序图设计 50](#_bookmark66)

[6.6.4 核心处理流程 51](#_bookmark67)

[**7 编码 52**](#_bookmark68)

* 1. [代码实现与核心算法 52](#_bookmark69)
  2. [代码优化分析 55](#_bookmark70)

[**8 测试 57**](#_bookmark71)

* 1. [测试方案设计 57](#_bookmark72)

[8.1.1 测试策略 57](#_bookmark73)

[8.1.2 测试进度安排 57](#_bookmark74)

[8.1.3 测试资源 57](#_bookmark75)

[8.1.4 关键测试点 58](#_bookmark76)

* 1. [测试用例构建 58](#_bookmark77)
     1. [测试用例编写约定 58](#_bookmark78)
     2. [测试用例设计 58](#_bookmark79)
     3. [关键测试用例 59](#_bookmark80)
     4. [测试用例维护 59](#_bookmark81)

1. [总结与展望 60](#_bookmark82)
   1. [设计工作总结 60](#_bookmark83)
   2. [未来工作展望 60](#_bookmark84)

[谢 辞 61](#_bookmark85)

[参考文献 62](#_bookmark86)

[附录 A 外文翻译—原文部分 64](#_bookmark87)

[附录 B 外文翻译—译文部分 70](#_bookmark88)

[附录 C 软件使用说明书 73](#_bookmark89)

[附录 D 主要源代码 82](#_bookmark90)

# 绪论

### 研究的背景及意义

* + 1. 选题的背景

书店相信每个人都逛过，我们可以将逛书店的人分为三类：根本不买书只是看看的人； 对书进行比较货比三家的人；看到哪本感兴趣直接买的人。对于第一二类用户而言，逛书店需要耗费很多往返的时间和精力。所以电子书店随着互联网的发展就逐渐兴起了，并紧跟互联网技术的发展而展现不同的形式，这里就不得不提到微信小程序了。

微信小程序依托于腾讯的金字招牌一出世便吸引了极大的关注量。很多人都对微信小程序的抱有很大的期望，而小程序的表现也没有让大家失望。截至 2018 年 3 月,微信小程

序的月活跃用户就已经达到四亿,而截至到 2019 年末,月活跃用户一举突破 6 亿,这带给店家的是巨大的流量,网上书店的模式也十分贴合小程序的使用场景。

根据调查报告数据显示,微信小程序中使用平吕最高的一大类就是电商，也是用户认为比 app 更实用的类别，图书商城本质上也是属于电商中的一类，同时随着我国人均文化素养的提升，每年人均阅读量显著上升，而一般用户网上购书的渠道都是淘宝等购物平台， 图书在这类购物平台中只是一小类，而进入这些购物平台都是琳琅满目的活动，看的人眼花缭乱，但大部分活动都没什么优惠，反而会造成卡顿影响用户的体验，微信小程序网上书城的优势在于依托于微信，只需要简单的向下一拉就可以进入商城，没有其他多余的东西影响购物体验，不需要下载其他购物 app。而且可以一键分享给微信好友，好友可以直接打开而不用复制链接给好友，好友再复制信息到购物 app 打开，偶尔手机卡顿还要重新复制，过程过于麻烦。而现在大部分用户最怕的就是麻烦。

* + 1. 国内外研究现状

国内现状

APP 人口红利尾声，用户增长迈入瓶颈期。潮退之时，互联网巨头最先做出应对，以自身已经拥有的超级 APP 为基搭建小程序或轻应用的分发平台。截至 2020 年 3 月，微信月活跃设备数高达 12 亿，因“水之积也厚”以大水为托，微信小程序受到众多开发者和用户的热情拥抱。

截至现在，电商、美食外卖行业百强小程序数保持了前二，并且稳中有升。月独立设备数 TOP100 的小程序中，排名前五的行业占去 71%的席位，其中电子商务行业拥抱大片河山，占据百强 20 席。

在电子商务小程序中，图书商城小程序也是其中发展趋势较好的一类，依托于微信小程序的火爆，图书商城的前景是非常可观的，对比一般的网上书城，微信图书商城最大的优势就是依托于微信的高曝光率以及方便无需下载 app，也不需要去特地打开浏览器

查找网址，更加方便也更加高效。国外现状

一家名为亚马逊的网上书店于 1995 年 7 月悄然成立，得益于较之同行更加便捷的购物流程，图书种类的丰富多样，更加便宜的价格，和更为完善的售后服务，在短短几年内就脱颖而出，超过上千名同行业竞争者，在网上书店行业的一骑绝尘。自从 1995 年 7 月亚

马逊书店出售第一本书以来，每年的销量都在稳步上升。截至到 2020 年 3 月，亚马逊的市

值已经达到了夸张的 1.1 万亿美元，当之无愧的网上购物巨头。

* + 1. 研究的意义

图书商城小程序比之一般的网上书城更加的方便，曝光率更高，不需要用户下载 app 或者专门在手机浏览器输入网址，只需要用户在微信界面往下一拉就可以找到小程序，而且页面也更贴合手机，看起来更加简洁大方，运行速度也更快一些。

图书商城小程序的优势主要体现在：

（1）快速建立品牌优势

图书商城小程序可以通过搜索附近、公众号、分享等裂变式的营销模式，有利于商家建立起属于自己的品牌。一旦品牌的建立，客户源就会源源不断，这样可以增加商家品牌的曝光度，同时也可以增加移动端流量的引流和转化率。

（2）

微信最大的优势就是依靠强大的社交属性，来获取大量的客户群体，将流量进行转化。另外，小程序图书商城背靠微信流量池，可以帮助企业快速实现裂变。同时它还可以通过群聊分享、好友分享、朋友圈实现裂变群体分享。比如大型的活动，双十一，双十二等， 可以将活动信息一键分享给好友，达到快速裂变的目的。

（3）成本低

图书商城小程序利用微信小程序将线上线下的联系进一步加强，有了微信强大的社交属性，图书商城小程序不会因为巨额的推广费而困扰，所以，企业在推广商城时所花费的成本低，可以帮助企业快速解决线上线下流量的转化，实现 o2o 模式。

### 系统目标

本系统的设计目标是能够完成基本的图书购买流程以及对数据的维护，具体功能如下： 管理员可以维护书籍信息，管理轮播图和推荐专题管理，处理订单信息。用户可以登

录书城，选择书籍分类，查看首页推荐书籍，查看推荐专题书单，查看书籍信息，将书籍加入购物车，清空购物车，生成订单，支付订单等操作。

# 2 需求分析

### 业务需求

* + 1. 前台能够实现图书展示，轮播图滑动推荐，书单推荐，图书分类，购物车管理，个人信息管理等基本业务流程。后台管理系统支持维护业务流程产生的各种数据，能够对商品信息，用户权限，系统日志等信息进行管理与维护。
    2. 系统页面跳转与获取数据的速度合理，系统功能运行稳定，对输入的错误信息有一定的验证能力。
    3. 主要业务流程
       1. 管理员填写图书基本信息将图书上架，商品展示活动图如下图 2-1 所示：

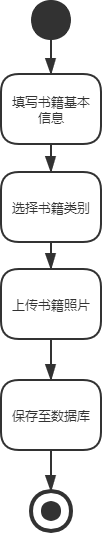


图 2-1 商品展示活动图

* + - 1. 用户进入图书商城主页面可以选择书籍分类或者推荐书单寻找感兴趣的书籍， 通过这三种方式的推荐选择心仪的图书并加入购物车，然后生成订单。

图书下单活动图如下图 2-2 所示：

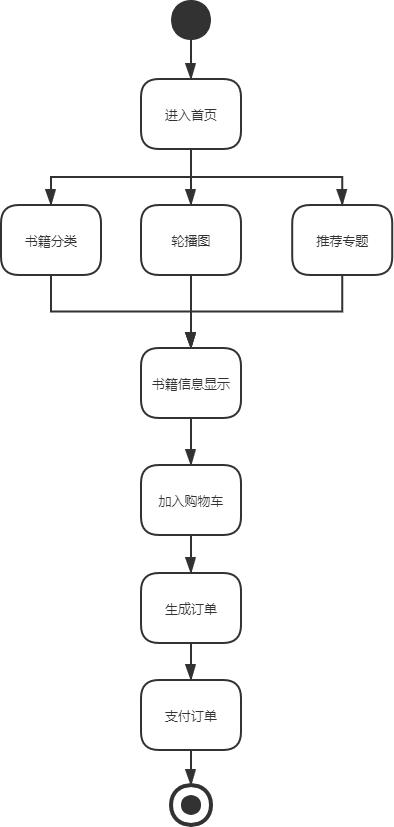


图 2-2 图书下单活动图

* + - 1. 用户点击我的按钮即可进入我的信息，点击登陆按钮便可以一键登录，系统会自动获取用户的微信头像及昵称等基本信息，并有订单信息和地址信息页面可以对订单和地址进行管理。用户信息管理活动图如下图 2-2 所示：

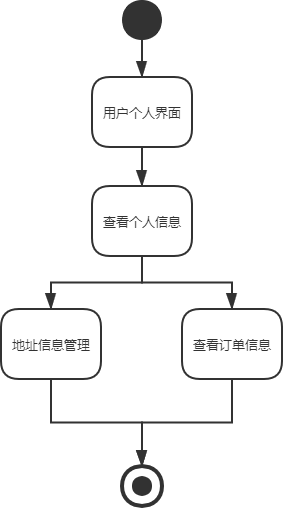


图 2-3 用户信息管理活动图

### 功能需求

* + 1. 角色分析

从网上购物系统的实际需求角度进行分析，本系统会涉及到以下角色：

表 2-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 用户 | 通过微信直接登陆可以浏览图书，轮播图，推荐专题，购物车管理，  以及个人信息管理。 |
| 管理员 | 可以对图书，轮播图，专题，用户权限进行管理。 |

* + 1. 业务功能

从系统角色角度出发，给出了系统的总体用例图，对于用户而言可以进行书籍选购， 个人信息管理，购物车管理，订单支付等操作。对于管理员而言，可以进行轮播图管理， 图书管理，专题管理，用户权限管理等操作。系统总体用例图如下图 2-4 所示：

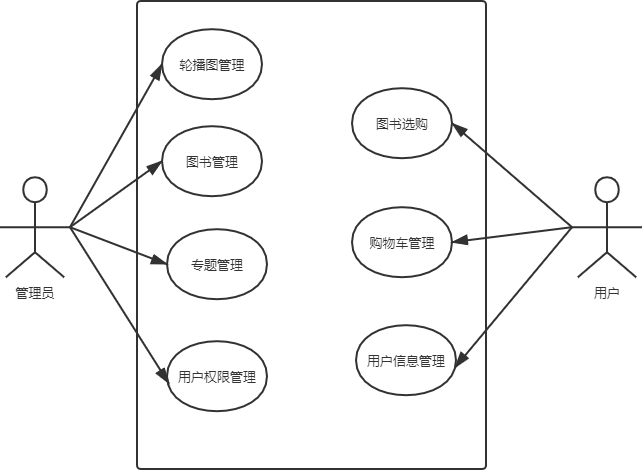


图 2-4 系统总体用例图

* + - 1. 商品选购

商品选购用例图如图 2-5 所示：

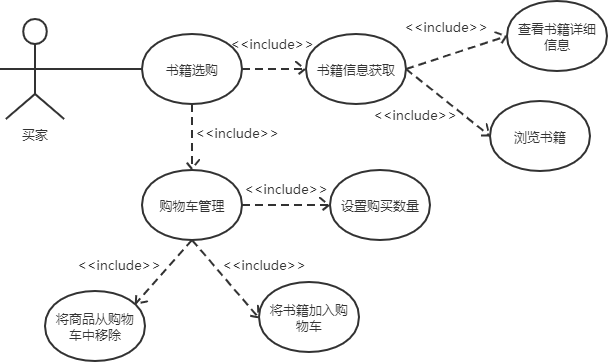


图 2-5 商品选购用例图

1. 商品选购用例描述如表 2-2 所示：

表 2-2 商品选购用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 商品选购 |
| 用例描述 | 用户选购商品流程 |
| 前置条件 | 用户已登录系统 |
| 后置条件 | 系统给出操作成功提示 |
| 基本操作流 | 1. 进入首页。 2. 选择心仪的图书； 3. 点击加入购物车按钮； 4. 返回上一界面继续选购其他商品； |

1. 图书信息获取用例描述如表 2-3 所示：

表 2-3 图书信息获取用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 商品信息获取 |
| 用例描述 | 用户获取图书的详细信息 |
| 参与者表 | 客户（买家） |
| 前置条件 | 用户已登录系统 |
| 后置条件 | 系统显示所获取图书信息 |
| 基本操作流 | 1. 用户进入分类页面。 2. 选择分类。 3. 选择图书。 4. 进入图书详情页面 |
| 可选操作流 | 1. 用户点击轮播图。 2. 进入图书详情。 3. 进入图书详情页面。 |

1. 加入购物车用例描述如表 2-4 所示：

表 2-4 加入购物车用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 加入购物车 |
| 用例描述 | 描述了将图书放入购物车的整个过程 |
| 参与者表 | 用户 |
| 基本操作流 | 1. 选择书籍 2. 点击“加入购物车”按钮 |
| 可选操作流 | 1．一次性加入多本图书 |

1. 设置购买数量用例描述如表 2-5 所示：

表 2-5 修改购买用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 修改购买数量 |
| 用例描述 | 用户修改购物车中图书的购买数量 |
| 参与者表 | 用户 |
| 前置条件 | 用户进入购物车界面 |
| 后置条件 | 系统给出操作成功提示 |
| 基本操作流 | 1. 用户找到需要进行操作的书籍 2. 点击右侧的加减号进行操作 |
| 被包含用例表 | 购物车管理 |

1. 查看书籍详细信息用例描述如表 2-6 所示：

表 2-6 查看图书详细信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 查看图书详细信息 |
| 用例描述 | 用户查看图书的详细信息 |
| 参与者表 | 用户 |
| 前置条件 | 用户已登录系统 |

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 后置条件 | 系统给出用户所查看商品详细信息 |
| 基本操作流 | 1. 选择某一商品类别 2. 选择某一具体商品 3. 点击“查看详情”按钮 |

* + - 1. 图书管理

图书管理用例图如图 2-6 所示：

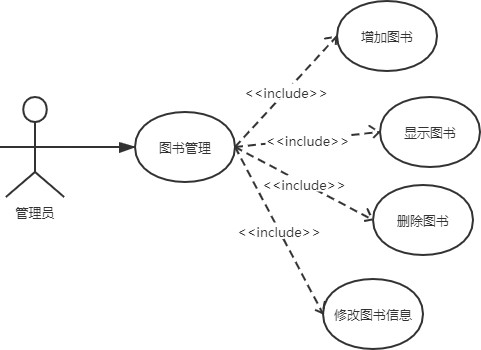


图 2-6 图书管理用例图

1. 增加图书用例描述如表 2-7 所示：

表 2-7 增加图书用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 增加图书 |
| 用例描述 | 填写图书的基本信息新增图书 |
| 参与者表 | 管理员 |
| c 前置条件 | 管理员登陆 |
| 后置条件 | 将对图书信息的操作的结果映射到数据库中 |

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 基本操作流 | 1. 进入后台管理页面 2. 点击新增图书按钮 3. 输入图书基本信息以及上传图片 4. 保存操作 |
| 可选操作流 | 1. 新增的图书信息不符合验证器验证标准，报错 2. 重新填写 |

1. 显示图书用例描述如表 2-8 所示：

表 2-8 显示图书用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 显示图书 |
| 用例描述 | 管理员查看图书列表显示所有图书信息 |
| 参与者表 | 管理员 |
| 前置条件 | 管理员登陆 |
| 后置条件 | 图书信息显示成功 |
| 基本操作流 | 1. 管理员进入图书管理页面 2. 点击显示图书列表 3. 查看图书信息 |

* + - 1. 用户信息管理

用户信息用例图如图 2-7 所示：

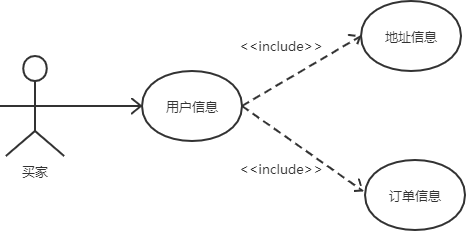


图 2-7 用户信息用例图

1. 用户信息用例如表 2-9 所示：

表 2-9 用户信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 用户信息 |
| 用例描述 | 用户获取个人信息 |
| 参与者表 | 用户 |
| 前置条件 | 用户打开个人信息页面点击登陆按钮登陆小程序 |
| 后置条件 | 登陆成功，成功获取用户信息 |
| 基本操作流 | 1. 用户打开个人信息页面 2. 点击登陆按钮 3. 成功获取用户信息 |

1. 地址信息用例如表 2-10 所示：

表 2-10 地址信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 地址信息 |
| 用例描述 | 用户选择对地址信息进行增删查改。 |
| 参与者表 | 用户 |
| 前置条件 | 用户打开个人信息页面 |
| 后置条件 | 操作成功 |
| 基本操作流 | 1. 查看地址信息 2. 增加一条地址信息 |
| 可选操作流 | 1. 删除一条地址信息 2. 修改一条地址信息 |
| 被包含用例表 | 个人信息 |

1. 订单信息用例如表 2-11 所示：

表 2-11 订单信息用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 订单信息 |
| 用例描述 | 用户个人的订单信息 |
| 参与者表 | 用户 |
| 前置条件 | 用户打开个人信息页面 |
| 后置条件 | 操作成功 |
| 基本操作流 | 1. 显示个人订单 2. 查看订单 |
| 可选操作流 | 1.支付未支付的订单 |

* + - 1. 轮播图管理

轮播图管理用例图如图 2-8 所示：

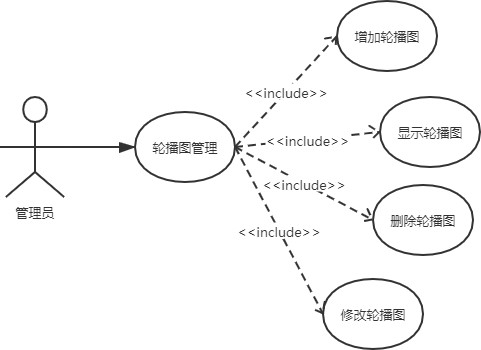


图 2-8 轮播图管理用例图

1）删除轮播图用例如表 2-12 所示：

表 2-12 删除轮播图用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 删除轮播图 |

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例描述 | 管理员点击删除按钮删除当前轮播图 |
| 参与者表 | 管理员 |
| 前置条件 | 管理员登陆后台管理系统，并打开轮播图列表 |
| 后置条件 | 将对轮播图信息的操作的结果映射到数据库中 |
| 基本操作流 | 1. 管理员进入轮播图管理页面 2. 点击轮播图列表按钮，显示所有轮播图 3. 选择轮播图，点击删除按钮 |
| 可选操作流 | 1.当前无轮播图，无法删除 |

* + - 1. 专题管理

1）专题管理用例图如图 2-9 所示：

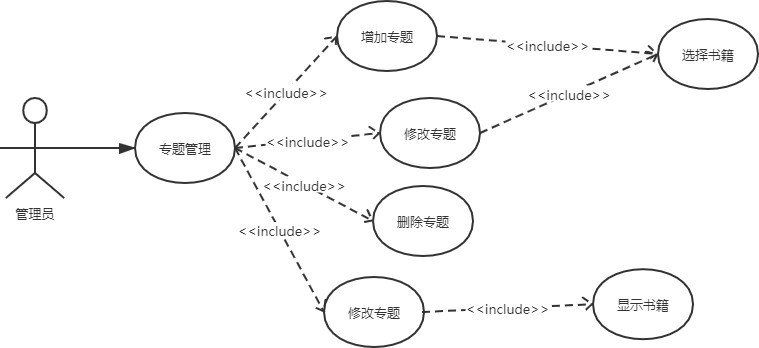


图 2-9 专题管理用例图

5）修改专题用例如表 2-13 所示：

表 2-13 修改专题用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 修改专题 |
| 用例描述 | 管理员修改专题信息 |

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 参与者表 | 管理员 |
| 前置条件 | 管理员登陆后台管理系统，并打开专题列表 |
| 后置条件 | 将对专题信息的操作的结果映射到数据库中 |
| 基本操作流 | 1. 管理员进入专题管理页面 2. 点击专题列表按钮，显示所有专题 3. 选择专题，点击修改按钮 4. 弹出修改页面，输入修改后的专题信息 5. 保存修改操作 |
| 可选操作流 | 1. 修改后的专题信息不符合验证器验证标准，报错 2. 重新填写修改信息 |

* + - 1. 用户权限管理

用户权限管理用例图如图 2-10 所示：

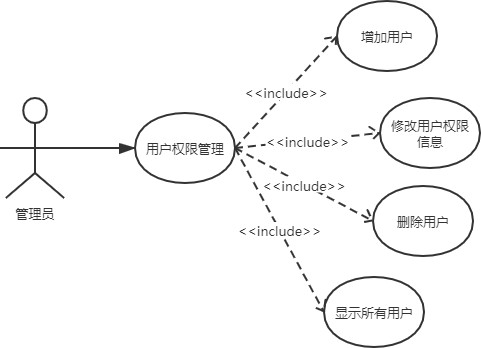


图 2-10 用户权限管理用例图

1）增加用户用例如表 2-14 所示：

表 2-14 增加用户用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 增加用户 |

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例描述 | 管理员注册新的用户数据，并为用户分配后台管理权限 |
| 参与者表 | 管理员 |
| 后置条件 | 注册完成用户能正常登陆 |
| 基本操作流 | 1. 管理员进入后台管理页面 2. 点击用户管理页面，点击新增用户 3. 注册新用户的账号密码并为其分配权限 4. 保存账号信息 |

### 非功能需求

* + 1. 环境需求

本系统服务端环境操作系统采用 Windows 10 系统，Web 服务器版本为 Apache 2.4.39，数据库采用 Mysql 5.7.26，客户端环境 Android/IOS 操作系统支持最新版微信即可。

* + 1. 性能需求
       1. 响应速度

API 请求的响应时间应低于 2s, 页面跳转速度 6s 以下。

* + - 1. 稳定性

系统需合理的利用资源，保证前后台数据操作的效率，以及在数据响应和界面承载方面都要达到不会出现界面混乱、数据报错、触发按钮功能缺失、操作频繁或者快速容易崩溃的问题。

* + - 1. 兼容性

具有兼容微信 5.3 版本以上的能力。

* + 1. 安全需求

1. 容错性

当用户输入非法数据时，系统应该对输入的数据进行验证，确保数据的合理性，保证系统的容错性。在这种情况下，系统应给出友好的提示，提示用户重新输入。

1. 权限

管理员将后台管理的权限分配给其他用户，没有得到对应权限的用户无法对后台数据进行操作。

# 总体设计

### 系统设计的原则

系统的设计严格遵循以下的设计原则：

* + 1. 实用性原则。系统数据保证全面且贴合用户实际需求，可以满足各类用户的基本需求。
    2. 简洁性原则。前后端页面保证简洁美观，各项功能分配明确，操作简单易懂， 给用户带来良好的体验。
    3. 标准化和规范化的原则。本系统按照相关标准，包括代码、界面、操作等，以达到标准化、规范化统的风格。

### 系统体系结构设计

本系统采用 ThinkPHP5.0 框架，以 ORM 的方式查询数据库，使用 TP5 验证器 Validate 来构建整个验证层用来验证数据的正确性，由于 TP5 框架自带的验证规则还是存在一定局限性，可以添加一些自定义的验证规则完善验证层，前端代码的话采用了 Class 和 Module 面向对象的思维来构建，为了方便不同的前端设备和后端交互，采用 RESTFul API 的 API 设计风格，因为是小程序服务的都是微信用户，所以不需要设计登陆注册页面，直接使用Token 令牌来构建用户授权体系。对于小程序的后台采用 cms 框架搭建一个与小程序配套的简单的后台管理系统，支持对于系统各类数据的一些简单的操作。

### 系统功能结构设计

系统功能结构图如图 3-1 所示：

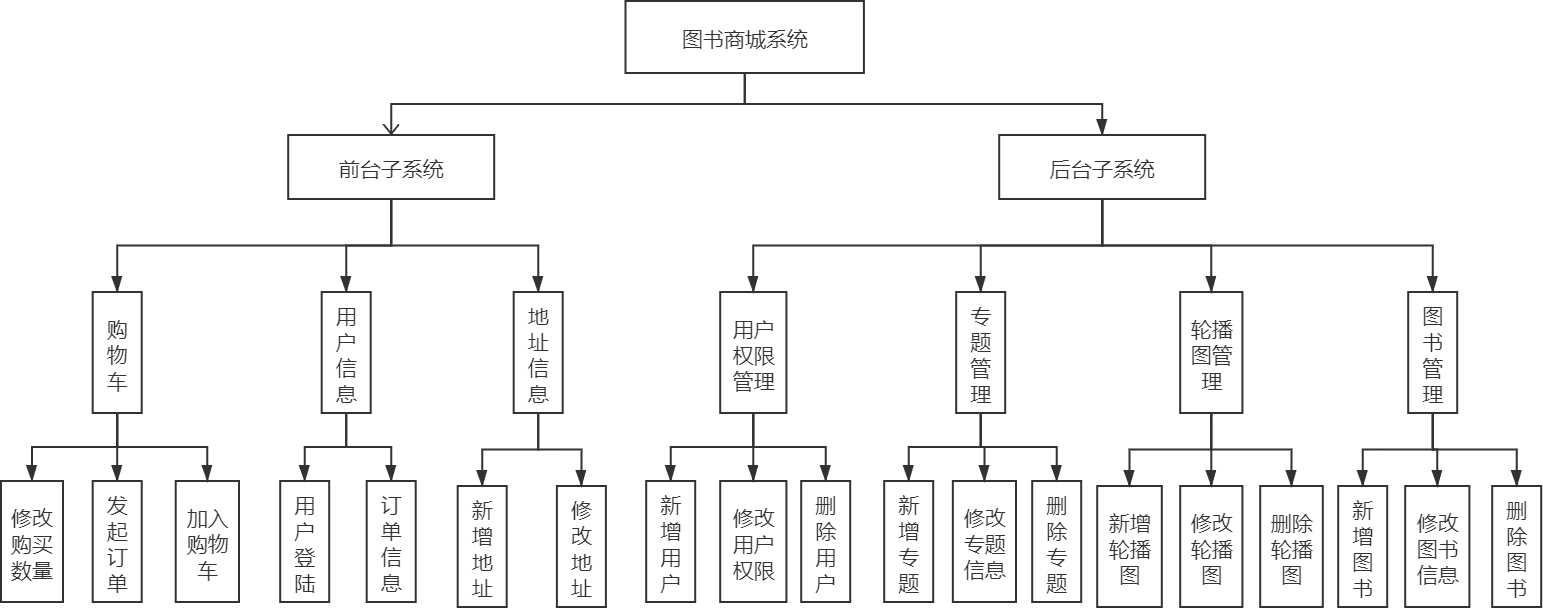


图 3-1 系统功能结构图

如上图所示，本系统由前台子系统和后台管理子系统两个子系统组成，两个子系统下的各个模块介绍如下：

1. 购物车模块

购物车模块由三个功能组成分别为修改购买数量，发起订单，加入购物车。

1. 修改购买数量：修改购物车中图书的购买数量。
2. 发起订单：用户可以再购物车中选择想要的图书在左侧点击选定，然后点击发起订单，系统将会计算所购买图书的总价格并生成订单。
3. 加入购物车：用户在图书页面点击加入购物车。
4. 用户信息模块

用户信息模块由用户登陆和订单信息组成。

1. 用户登陆：因为本系统是微信小程序，所以用户可以直接通过微信授权一键登陆本系统，系统会自动获取用户的基本信息。
2. 订单信息：在我的信息页面中查看订单状态。
3. 地址信息模块

保存用户的地址信息，在发起订单时调用地址信息，支持新增地址与修改地址。

1. 新增地址：用户点击新增按钮，跳转新增地址页面，输入地址信息时，后台验证器会验证地址信息是否合理。
2. 修改地址：用户选择地址信息，并对其进行修改，修改后的地址信息同样必须通过后台验证器验证格式是否正确。
3. 用户权限管理模块

管理员可以注册一些用户账号并给与这些账号一些权限，方便管理后台信息。支持新增用户，修改用户权限，删除用户等操作。

1. 新增用户：点击新增按钮，填写用户信息。
2. 修改用户权限：编辑按钮，编辑用户信息。
3. 删除用户：点击删除按钮，删除该用户账号。
4. 专题管理模块

管理员可以自行将一些比较值得推荐的图书整理成一个个书单，定期推荐给用户，后台管理系统支持对专题进行维护。

1. 新增专题：管理员点击新增专题，输入专题的基础信息，然后选择图书加入专题中。
2. 修改专题信息：管理员点击编辑按钮，弹出编辑页面，修改专题的信息以及所包含的图书。
3. 删除专题：管理员打开专题列表，选择想要删除的专题，点击专题最右边的删除按钮，一键删除当前专题
4. 轮播图管理模块

管理员将一些销售火热的书籍放在首页轮播图上，用户可以一眼就看到，同样轮播图管理模块也支持新增轮播图，修改轮播图信息，删除轮播图等操作。

1. 新增轮播图：管理员新增按钮，跳转新增轮播图页面，输入轮播图的封面，简介等基础信息，然后选择轮播图所指向的图书。
2. 修改轮播图信息：管理员打开轮播图列表，选择想要修改的轮播图，点击轮播图右边的编辑按钮，弹出编辑页面，修改轮播图的信息以及所指向的图书。
3. 删除专题：管理员打开轮播图列表，选择想要删除的轮播图，点击轮播图最右边的删除按钮，一键删除当前轮播图。
4. 图书管理模块

管理员通过图书管理页面可以对图书进行管理。

1. 新增图书：管理员点击新增按钮，编辑图书信息，编辑完毕点击提交后台验证器会验证图书信息是否合理，如果不合理则重新编辑。
2. 编辑图书信息：管理员打开图书列表，点击编辑按钮，跳转图书编辑页面，编辑图书信息。
3. 删除图书：管理员选则图书点击右边的删除按钮，删除图书。

# 数据库设计

### 概念结构设计

* + 1. 设计思路

本系统开发时采用自顶向下的策略，首先整理好整体架构的设计思路，再逐渐细化到每一个子模块具体的设计。

本系统概念结构设计如下：

* + - 1. 轮播图表（id,名称，描述，关联的书籍）
      2. 图书专题表（id,名称，描述，封面，包含的图书，列表页头图）
      3. 图书表（id，名称，描述，封面，价格，库存，所属分类）
      4. 地址信息表（id，收货人姓名，手机号，省，市，县，详细地址）
      5. 订单表（id，订单号，用户 id，价格，订单状态，订单快照）
      6. 图书分类表（id，名称，描述，封面）
    1. E-R 图

图书商城系统总体 E-R 图如下图 4-1 所示：

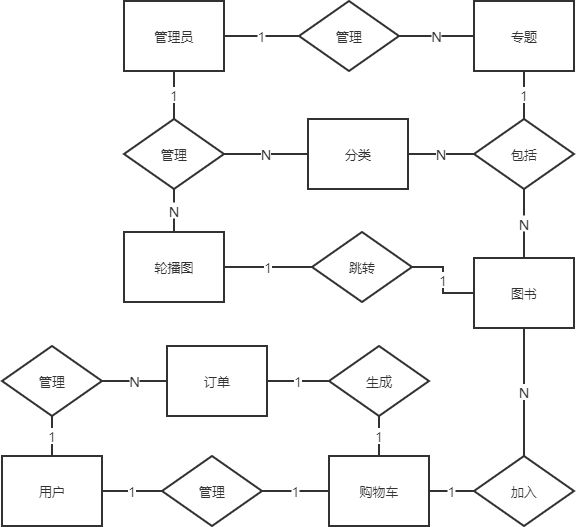


图 4-1 图书商场系统总体E-R 图

1. 轮播图实体 E-R 图实体：书籍

属性：id,名称，描述，关联的书籍。轮播图 ER 图如下图 4-2 所示。

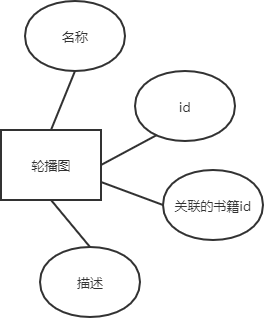


图 4-2 轮播图实体 E-R 图

1. 图书实体 E-R 图实体：图书

属性：id，名称，描述，封面，价格，库存，所属分类 id。图书实体 E-R 图如图 4-3

所示：

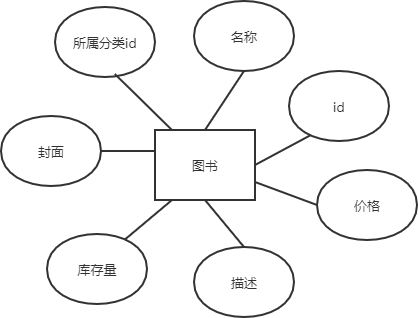


图 4-3 书籍实体 E-R 图

1. 图书专题实体 E-R 图实体：图书专题

属性：id，名称，描述，封面，包含的书图书，列表页头图。图书专题实体 E-R 图如图 4-4 所示：

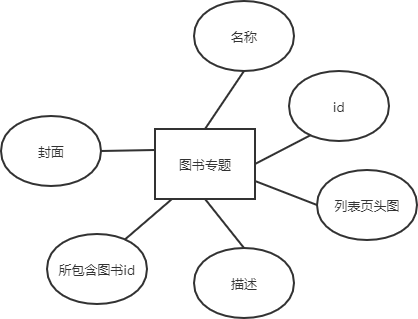


图 4-4 图书专题实体 E-R 图

1. 地址信息实体 E-R 图实体：地址信息。

属性：id，收货人姓名，手机号，省，市，县，详细地址。地址信息实体 E-R 图如图

4-5 所示：

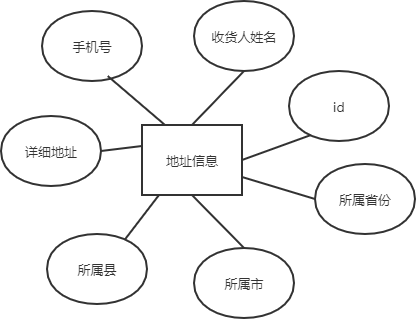


图 4-5 地址信息实体 E-R 图

1. 订单实体 E-R 图实体：订单。

属性：id，订单号，用户 id，价格，订单状态，订单快照。订单实体 E-R 图如图 4-6 所

示：

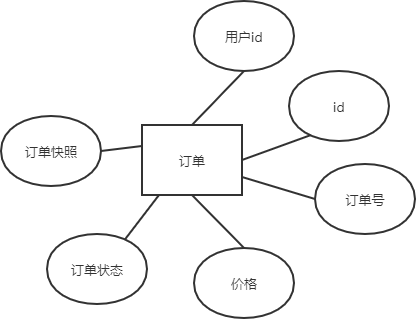


图 4-6 订单实体 E-R 图

1. 图书分类实体 E-R 图实体：图书分类

属性：id，名称，描述，封面。图书分类实体 E-R 图如图 4-7 所示：

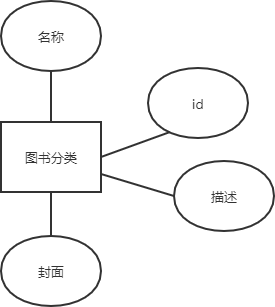


图 4-7 图书分类实体 E-R 图

### 逻辑结构设计

* + 1. 设计思路

将概念设计阶段设计好的基本 E-R 图转换为与数据模型相符合的逻辑结构。

* + 1. 逻辑模型

1. 用户信息表如表 4-1 所示：

表 4-1 用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | User\_id | 整型 | 20 | 是 | 用户 ID |
| 2 | nickname | 字符串 | 20 | 否 | 用户名 |

1. 用户地址表如表 4-2 所示：

表 4-2 用户地址表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | use\_id | 整型 | 20 | 是 | 用户 ID |
| 2 | name | 字符串 | 20 | 否 | 收货人姓名 |
| 3 | mobile | 整型 | 20 | 否 | 手机号 |
| 4 | province | 字符串 | 20 | 否 | 省 |
| 5 | city | 字符串 | 20 | 否 | 市 |
| 6 | country | 字符串 | 20 | 否 | 区 |
| 7 | detail | 字符串 | 100 | 否 | 详细地址 |

1. 轮播图表如表 4-3 所示：

表 4-3 轮播图表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | bannerID | 整型 | 11 | 是 | 轮播图 ID |
| 2 | img\_id | 整型 | 11 | 否 | 外键，关联 image 表 |

1. 分类表如表 4-4 所示：

表 4-4 分类表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | categoryID | 整型 | 11 | 是 | 分类 ID |
| 2 | categoryName | 字符串 | 50 | 否 | 分类名称 |
| 3 | topic\_img\_id | 整型 | 11 | 否 | 外键，关联 image 表 |
| 4 | description | 字符串 | 100 | 否 | 分类描述 |

1. 图表如表 4-5 所示：

表 4-5 图表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | Image\_id | 整型 | 11 | 是 | 图片 ID |
| 2 | image\_url | 字符串 | 255 | 否 | 图片路径 |

1. 订单信息表如表 4-6 所示：

表 4-6 订单信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | Order\_id | 整型 | 20 | 是 | 订单 ID |
| 2 | order\_no | 字符串 | 20 | 否 | 订单号 |
| 3 | user\_id | 整型 | 11 | 否 | 外键，用户 id |
| 4 | status | 整型 | 4 | 否 | 1:未支付 2:已支付 3:已发货 |
| 5 | snap\_img | 字符串 | 255 | 否 | 订单快照图片 |
| 6 | snap\_name | 字符串 | 80 | 否 | 订单快照名称 |

1. 书籍信息表如表 4-7 所示：

表 4-7 书籍信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | product\_id | 整型 | 11 | 是 | 书籍 ID |
| 2 | product\_name | 字符串 | 100 | 否 | 书籍名称 |
| 3 | product\_price | 整型 | 11 | 否 | 价格 |
| 6 | main\_img\_url | 字符串 | 255 | 否 | 主图 ID 号 |
| 7 | summary | 字符串 | 2550 | 否 | 书籍简介 |
| 8 | img\_id | 整型 | 11 | 否 | 图片外键 |

1. 商品属性表如表 4-8 所示：

表 4-8 商品属性表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | property\_id | 整型 | 20 | 是 | 属性 ID |
| 2 | property\_name | 字符串 | 30 | 否 | 详情属性名称 |
| 3 | property\_detail | 字符串 | 255 | 否 | 详情属性 |
| 4 | product\_id | 整型 | 11 | 否 | 商品 id，外键 |

1. 专题表如表 4-9 所示：

表 4-9 专题表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | theme\_id | 整型 | 20 | 是 | 专题 ID |
| 2 | theme\_name | 字符串 | 30 | 否 | 专题名称 |
| 3 | description | 字符串 | 255 | 否 | 专题描述 |
| 4 | topic\_img\_id | 整型 | 11 | 否 | 主题图，外键 |
| 5 | head\_img\_id | 整型 | 11 | 否 | 专题列表页，头图 |

### 物理结构设计

* + 1. 存取方式

索引方法，采用 B+树索引。因为 B+树为多分平衡树，存取效率高既能随机查找、又能顺序查找增删改操作，保持平衡。

决定使用 mysql 数据库，数据类型整型采用 int，字符串采用 varchar。

* + 1. 存储结构

1. 用户信息表如表 4-10 所示：

表 4-10 用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | User\_id | int | 20 | 是 | 用户 ID |
| 2 | nickname | varchar | 20 | 否 | 用户名 |

1. 用户地址表如表 4-11 所示：

表 4-11 用户地址表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | use\_id | int | 20 | 是 | 用户 ID |
| 2 | name | varchar | 20 | 否 | 收货人姓名 |
| 3 | mobile | int | 20 | 否 | 手机号 |
| 4 | province | varchar | 20 | 否 | 省 |
| 5 | city | varchar | 20 | 否 | 市 |
| 6 | country | varchar | 20 | 否 | 区 |
| 7 | detail | varchar | 100 | 否 | 详细地址 |

1. 轮播图表如表 4-12 所示：

表 4-12 轮播图表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | bannerID | int | 11 | 是 | 轮播图 ID |
| 2 | img\_id | int | 11 | 否 | 外键，关联 image 表 |

1. 分类表如表 4-13 所示：

表 4-13 分类表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | categoryID | int | 11 | 是 | 分类 ID |
| 2 | categoryName | varchar | 50 | 否 | 分类名称 |
| 3 | topic\_img\_id | int | 11 | 否 | 外键，关联 image 表 |
| 4 | description | varchar | 100 | 否 | 分类描述 |

1. 图表如表 4-14 所示：

表 4-14：图表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | Image\_id | int | 11 | 是 | 图片 ID |
| 2 | image\_url | varchar | 255 | 否 | 图片路径 |

1. 订单信息表如表 4-15 所示：

表 4-15：订单信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | Order\_id | int | 20 | 是 | 订单 ID |
| 2 | order\_no | varchar | 20 | 否 | 订单号 |
| 3 | user\_id | int | 11 | 否 | 外键，用户 id |
| 4 | status | int | 4 | 否 | 1:未支付 2:已支付 3:已发货 |
| 5 | snap\_img | varchar | 255 | 否 | 订单快照图片 |
| 6 | snap\_name | varchar | 80 | 否 | 订单快照名称 |

1. 书籍信息表如表 4-16 所示：

表 4-16：书籍信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | product\_id | int | 11 | 是 | 书籍 ID |

续表 4-16

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 2 | product\_name | varchar | 100 | 否 | 书籍名称 |
| 3 | product\_price | int | 11 | 否 | 价格 |
| 4 | main\_img\_url | varchar | 255 | 否 | 主图 ID 号 |
| 5 | create\_time | int | 11 | 否 | 创建时间 |
| 6 | summary | varchar | 2550 | 否 | 书籍简介 |
| 7 | img\_id | int | 11 | 否 | 图片外键 |

1. 商品属性表如表 4-17 所示：

表 4-17：商品属性表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | property\_id | int | 20 | 是 | 属性 ID |
| 2 | property\_name | varchar | 30 | 否 | 详情属性名称 |
| 3 | property\_detail | varchar | 255 | 否 | 详情属性 |
| 4 | product\_id | int | 11 | 否 | 商品 id，外键 |

1. 专题表如表 4-18 所示：

表 4-18：专题表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段  长度 | 是否  主键 | 说明 |
| 1 | theme\_id | int | 20 | 是 | 专题 ID |
| 2 | theme\_name | varchar | 30 | 否 | 专题名称 |
| 3 | description | varchar | 255 | 否 | 专题描述 |
| 4 | topic\_img\_id | int | 11 | 否 | 主题图，外键 |
| 5 | head\_img\_id | int | 11 | 否 | 专题列表页，头图 |

# 界面设计

### 界面关系图或工作流图

首先用户进入主页面，可以点击图标或图片选择跳转到轮播图页面，专题推荐页面， 图书分类页面，购物车页面，个人信息页面等页面。如果用户点击的是轮播图，专题推荐或图书分类页面，都会跳转到图书详情页面，将图书加入购物车后，点击购物车图标可以跳转到购物车页面查看图书，下单后可以到个人信息页面查看订单详情和进行地址管理。界面关系图如图 5-1。

页面 UI 设计整体风格偏冷色调，系统看起来更加简洁大方，功能按钮图片选择简洁风的，符合当下流行的极简主义，分类上方放有符合分类风格的头图，这样不会让分类页面显得过于空洞，增加美感首页的轮播图和推荐专题会占首页一大部分面积不仅仅是为了推荐图书，同时也会让系统看起来更加的舒服，不会一眼望过去全是密密麻麻的图书，让用户体验不佳。

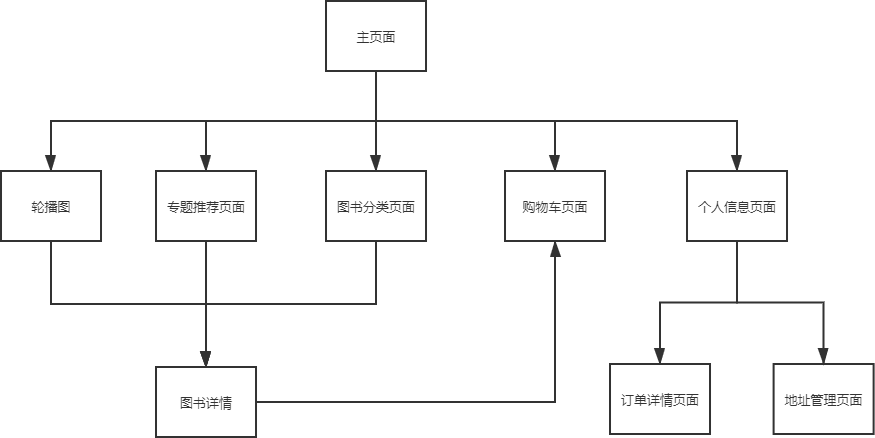


图 5-1 界面关系图

### 界面设计成果

* + 1. 主界面

主界面最顶部为小程序名字图书商城，点击名字右侧三个点的按钮可以一键分享小程序，往下就是轮播图，这里一共有四幅轮播图，支持滑动，点击轮播图会进入图书详情， 轮播图下方为精选主题，可以点击查看推荐主题，最下面是功能栏分别对应首页，图书分类，购物车，我的信息。小程序项目主界面如图 5-2 所示：



图 5-2 项目主界面

* + 1. 轮播图子界面

点击首页的轮播图如我们点击第一张轮播图《活着》，可以跳转到图书详情界面。轮播图页面如图 5-3 所示：

图 5-3 轮播图界面

* + 1. 专题推荐子界面

如果点击首页的第三个专题推荐，用户可以跳转到校园青春漫画推荐专题页面专题推荐界面如图 5-4 所示：



图 5-4 专题推荐界面

* + 1. 图书分类子界面

点击首页最下方的分类按钮，用户可以跳转到图书分类子界面图书分类界面给各类图书分好了类别，可以直接查看各类型图书。图书分类界面如图 5-5 所示：

图 5-5 图书分类界面

* + 1. 图书详情子界面

前面几个子界面都是为了让用户更好的找到感兴趣的图书，用户找到感兴趣的图书点击图书，跳转图书详情页面.可查看图书详细信息并选择加入购物车的数量。图书详情界面如图 5-6 所示：

图 5-6 图书详情界面

* + 1. 购物车子界面

用户进入购物车。可以查看购物车中的图书并修改图书的购买数量，点击购物车中图书左边的红点可以选择是否选择该图书，选择完图书后点击下单即可生成订单。

购物车界面如图 5-7 所示：

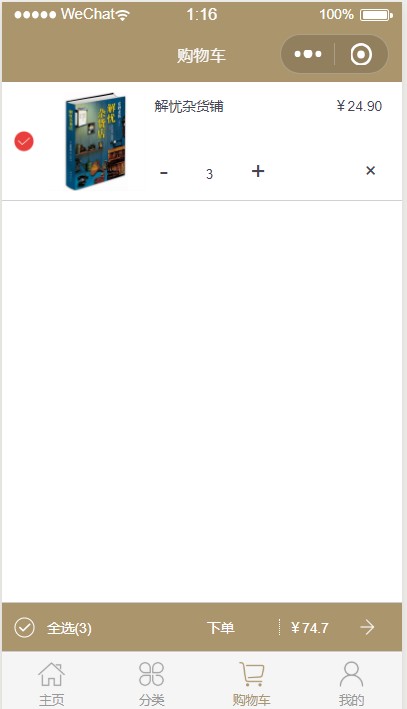


图 5-7 购物车界面

* + 1. 个人信息子界面

用户点击最下面的我的按钮就可以到达个人信息子界面，点击“请单击获取你的个人信息”会弹出微信授权，点击接收即可获取微信个人信息，同时下面也有用户的地址信息与订单信息。个人信息子界面如图 5-8 所示：

图 5-8 个人信息子界面

* + 1. 地址管理子界面

点击个人信息页面中的地址管理即可进入地址管理子界面，可以在这选择默认的地址或者新增地址。

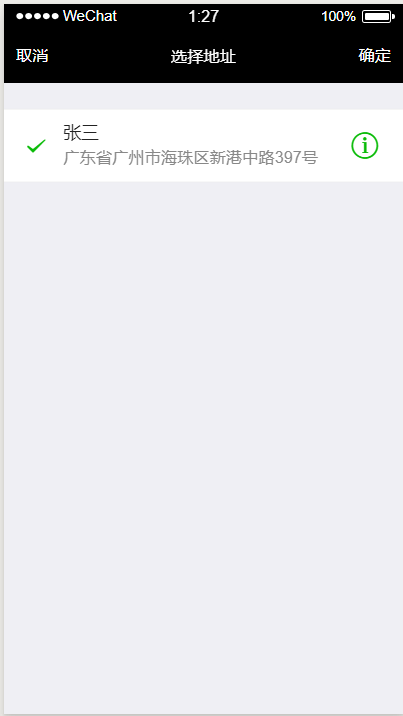
地址管理子界面如图 5-9 所示：

图 5-9 地址管理子界面

* + 1. 订单详情子界面

点击个人信息页面中的订单详情中任意一个订单即可进入当前订单详情页面，可以在这查看订单各种信息，对于没有付款的订单，可以选择支付订单。

订单详情子界面如图 5-10 所示：

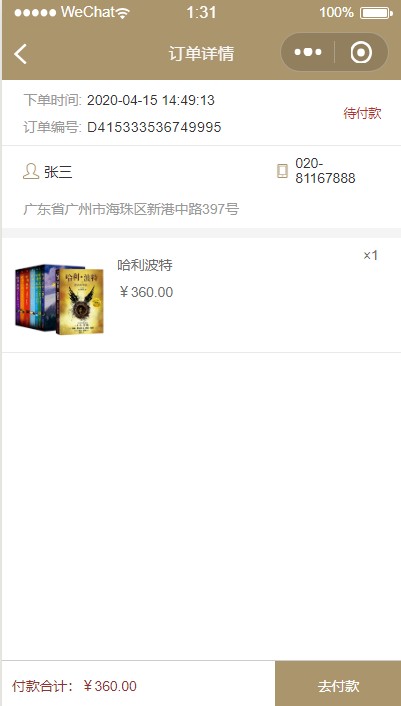


图 5-10 订单详情子界面

# 详细设计

### 系统主要功能模块介绍

子系统关系图如图 6-1 所示：

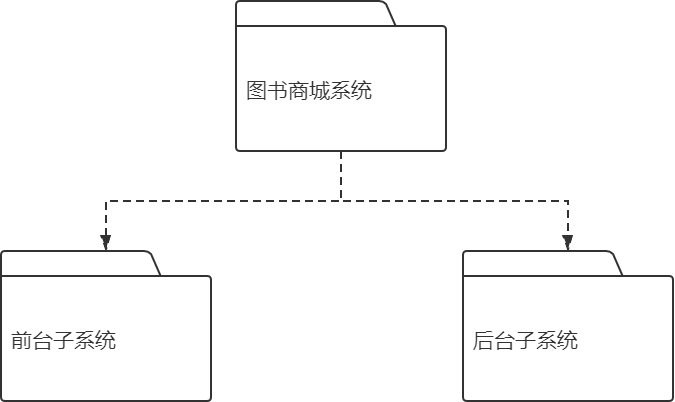


图 6-1 子系统关系图

前台子系统共包括购物车，用户信息，地址信息三个模块。

* + 1. 购物车模块：购物车模块由三个功能组成分别为修改购买数量，发起订单，加入购物车。
    2. 用户信息模块：用户信息模块由用户登陆和订单信息组成。
    3. 地址信息模块：保存用户的地址信息，在发起订单时调用地址信息，支持新增地址与修改地址。

后台子系统共包括：书籍管理、交易信息管理、用户信息信息管理三个模块。

1. 书籍管理：包含查找书籍、删除书籍、新增书籍、修改书籍信息等操作。
2. 用户账户信息管理：包含修改用户各项信息、新增或删除用户等操作。
3. 交易信息管理：包含查看交易信息、删除交易信息、修改交易信息、新增交易信息等操作。

### 地址信息模块设计

* + 1. 功能结构设计

地址信息子功能保存用户的地址信息，在发起订单时调用地址信息，支持新增地址与修改地址。

* + - 1. 新增地址：用户可以在个人信息页面新增地址信息，提交地址信息时，后台验证器会验证地址信息是否合理。
      2. 修改地址：用户可以在个人信息页面选择地址信息，并对其进行修改，修改后的地址信息同样必须通过后台验证器验证格式是否正确。地址信息功能结构设计图如图 6-2 所示：

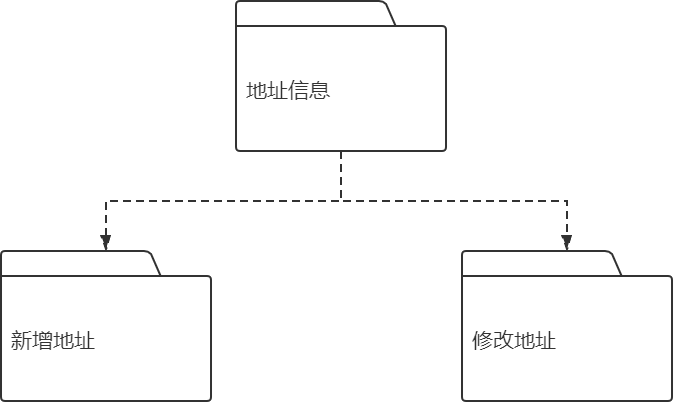


图 6-2 地址信息功能结构设计图

* + 1. 功能类设计

地址信息类图包含于下图 6-3 系统总类图中：

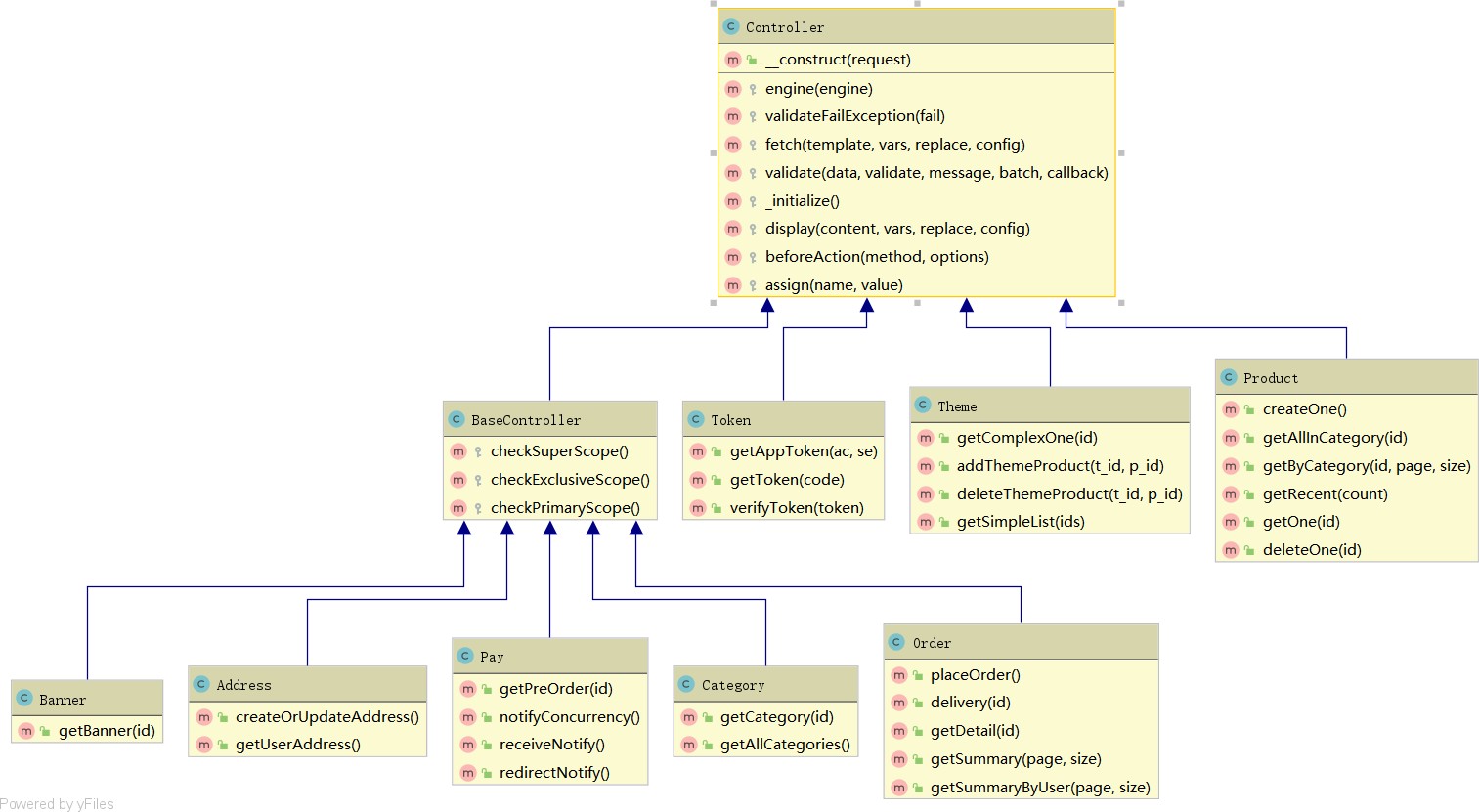


图 6-3 系统总类图

Controller 类描述如表 6-1 所示:

表 6-1 Controller 类描述

|  |  |  |
| --- | --- | --- |
| Controller 类 | | |
| 类的描述：所有类的基类，负责实现一些类的共有方法 | | |
| 操作描述 | \_initialize（） | 初始化类的基本属性 |
| beforeAction（） | 前置操作 |
| fetch(self) | 加载模板输出 |
| Display（） | 渲染内容输出 |
| Assign（） | 模板变量赋值 |
| Engine（） | 初始化模板引擎 |
| validateFailException（） | 设置验证失败后是否抛出异常 |
| Validate（） | 验证数据 |

Address 类描述如表 6-2 所示:

表 6-2 Address 类描述

|  |  |  |
| --- | --- | --- |
| Address 类 | | |
| 类的描述：地址信息类 | | |
| 操 | getUserAddress（） | 获取用户地址信息 |
| 作 |
|  |  |
| 描 | createOrUpdateAddress（） | 更新或者创建用户收获地址 |
| 述 |  |  |

Token 类描述如表 6-3 所示:

表 6-3 Token 类描述

|  |  |  |
| --- | --- | --- |
| Token 类 | | |
| 类的描述：获取令牌，相当于登录 | | |
| 操作描  述 | getToken（） | 用户获取令牌（登陆） |
| getAppToken（） | 第三方应用获取令牌 |
|  | verifyToken（） | 验证令牌 |

* + 1. 顺序图设计

地址信息顺序图如图 6-4 所示：

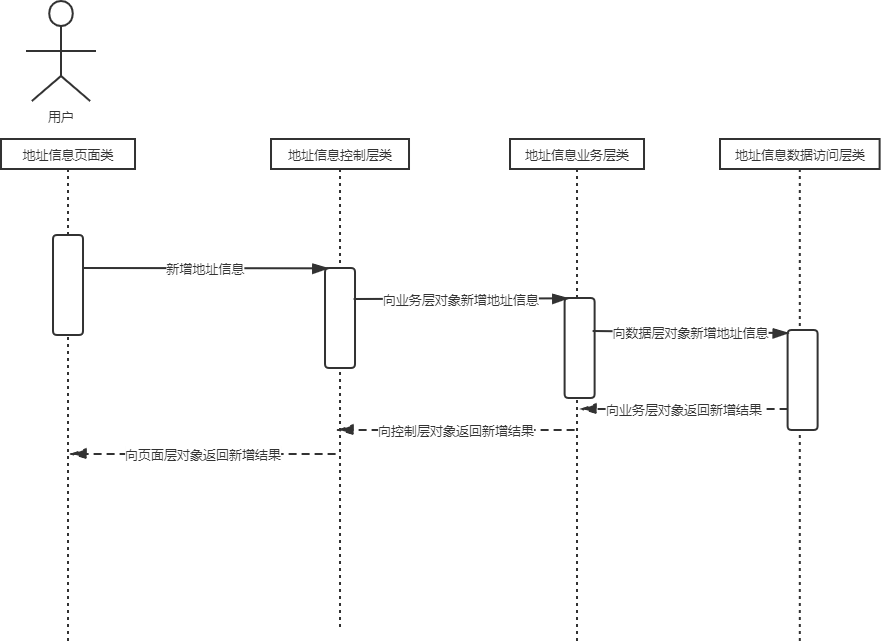


图 6-4 地址信息功能顺序图

* + 1. 核心处理流程

地址信息核心处理流程设计图如图 6-5 所示：

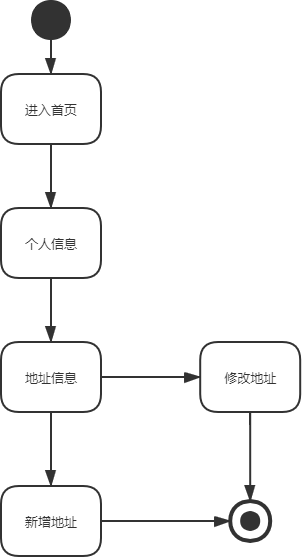


图 6-5 地址信息核心处理流程设计图

### 轮播图模块设计

* + 1. 功能结构设计

轮播图功能结构设计图如图 6-6 所示：

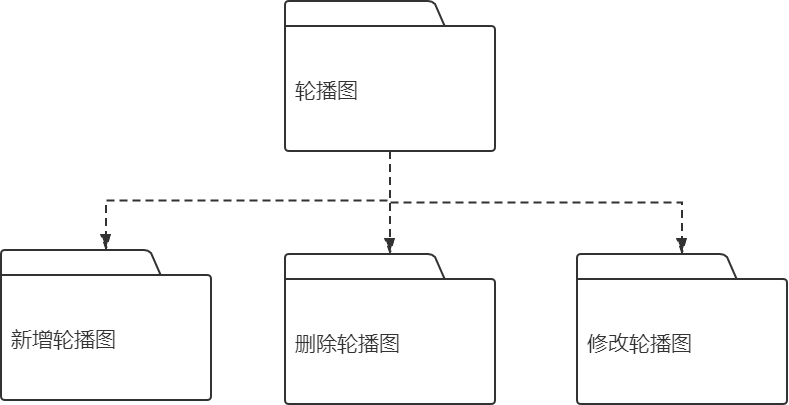


图 6-6 轮播图功能结构设计图

* + 1. 功能类设计

轮播图类图包含在上面的类图 6-3 中。

Banner 类描述如表 6-4 所示:

表 6-4 Banner 类描述

|  |  |  |
| --- | --- | --- |
| Banner 类 | | |
| 类的描述：轮播图类，轮播图管理类 | | |
| 操作描述 | getBanner（） | 获取轮播图信息 |
| addBanner() | 新增轮播图接口 |
| delBanner() | 删除轮播图 |
| editBannerInfo($id) | 编辑轮播图基础信息 |
| addBannerItem() | 新增轮播图元素 |
| editBannerItem() | 编辑轮播图元素 |
| delBannerItem() | 删除轮播图元素 |

* + 1. 顺序图设计

地址信息顺序图如图 6-7 所示：

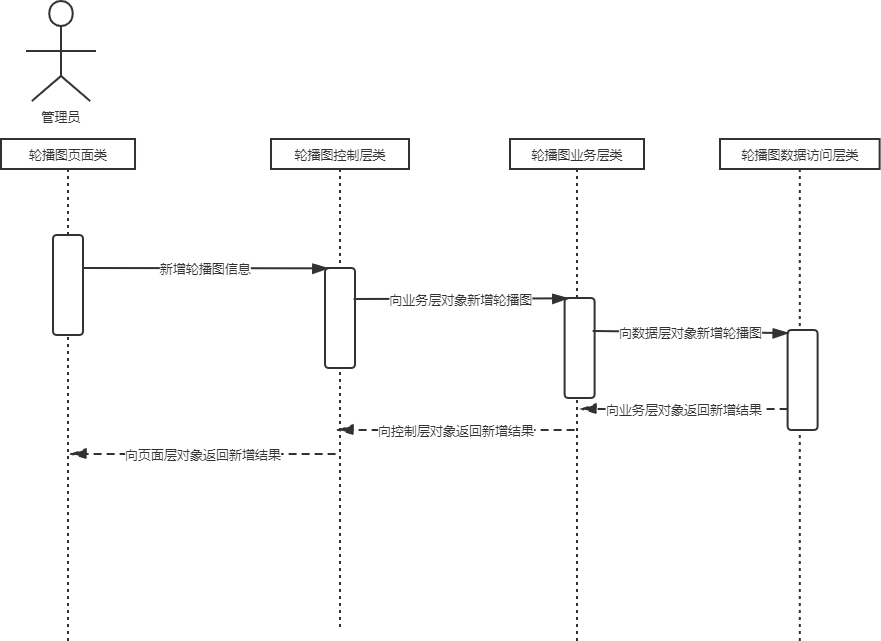


图 6-7 轮播图功能时序图

* + 1. 核心处理流程

轮播图核心处理流程设计图如图 6-8 所示：

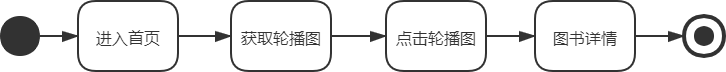


图 6-8 轮播图核心处理流程设计图

### 专题管理模块设计

* + 1. 功能结构设计

专题管理功能结构设计图如图 6-9 所示：

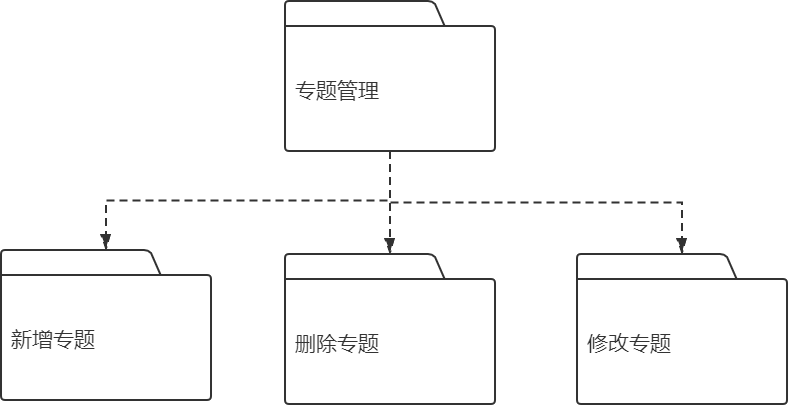


图 6-9 专题管理功能结构设计图

* + 1. 功能类设计

专题管理类图包含在上面的类图 6-3 中，Theme 类描述如表 6-5 所示:

表 6-5 Theme 类描述

|  |  |  |
| --- | --- | --- |
| Theme 类 | | |
| 类的描述：图书专题类，专题管理类 | | |
| 操作描述 | getSimpleList（） | 获取专题列表 |
| addThemeProduct（） | 新增专题元素 |
| deleteThemeProduc（） | 删除专题元素 |
| editThemeProduc（） | 编辑专题元素 |
| addTheme（） | 新增专题 |
| deleteTheme（） | 删除专题 |
| editTheme（） | 编辑专题 |

* + 1. 顺序图设计

专题管理顺序图如图 6-10 所示：

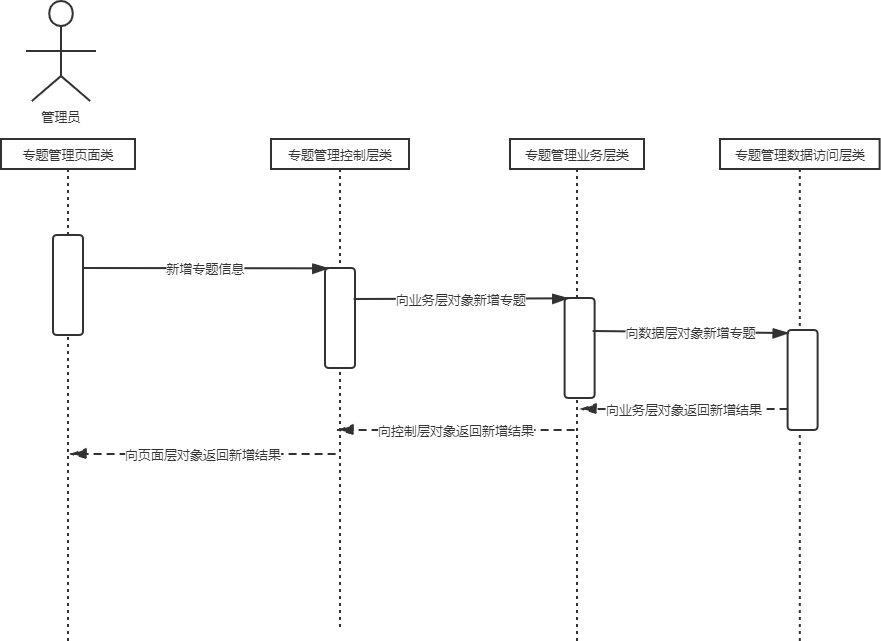


图 6-10：专题管理功能顺序图

* + 1. 核心处理流程

图书专题核心处理流程设计图如图 6-11 所示：

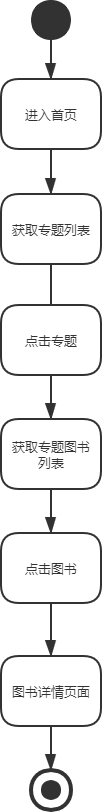


图 6-11 图书专题核心流程设计图

### 图书管理模块设计

* + 1. 功能结构设计

图书管理功能结构设计图如图 6-12 所示：

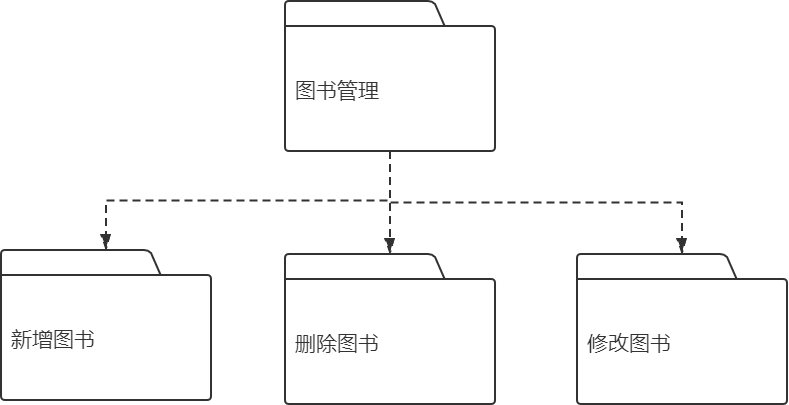


图 6-12：图书管理功能结构设计图

* + 1. 功能类设计

图书管理类图包含在上面的类图 6-3 中，Product 类描述如表 6-6 所示:

表 6-6 Product 类描述

|  |  |  |
| --- | --- | --- |
| Theme 类 | | |
| 类的描述：图书管理类，对图书进行管理。 | | |
| 操作描述 | getByCategory（） | 根据类目 ID 获取该类目下所有图书(分页） |
| getAllInCategory（） | 获取某分类下全部图书(不分页） |
| getRecent（） | 获取指定数量的最近图书 |
| getOne（） | 获取图书详情 |
| createOne（） | 编辑图书详情 |
| deleteOne（） | 删除图书 |

Category 类描述如表 6-7 所示:

表 6-7 Category 类描述

|  |  |  |
| --- | --- | --- |
| Theme 类 | | |
| 类的描述：图书管理类，对图书进行管理。 | | |
| 操作描  述 | getAllCategories（） | 根据类目 ID 获取该类目下所有图书(分页） |
| getAllInCategory（） | 获取某分类下全部图书(不分页） |
| getCategory | 获取分类 |

* + 1. 顺序图设计

图书管理顺序图如图 6-13 所示：

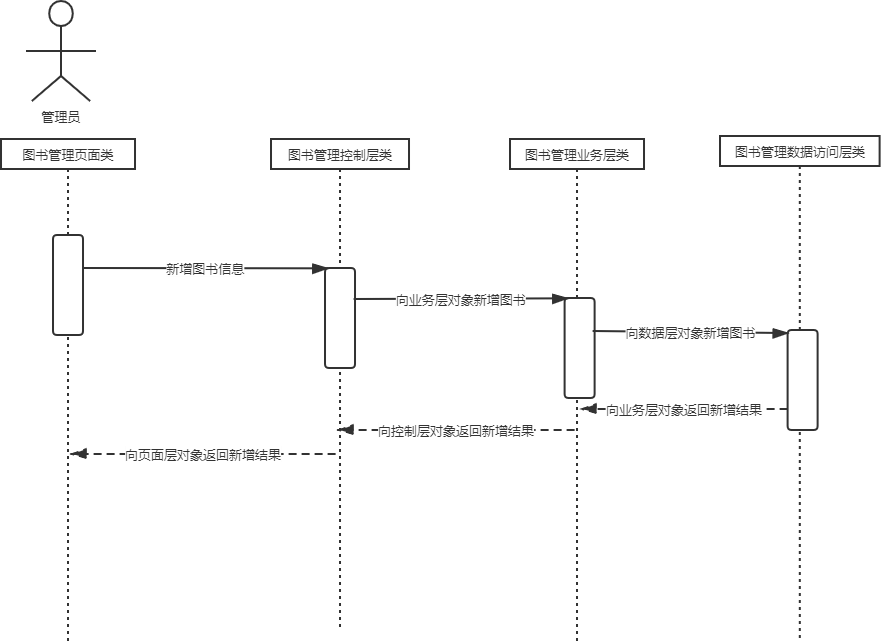


图 6-13：图书管理功能时序图

* + 1. 核心处理流程

图书详情核心流程设计设计图如图 6-14 所示：

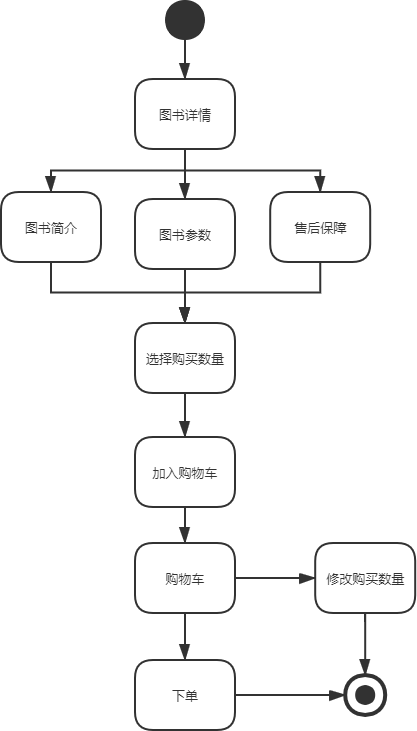


图 6-14 图书详情核心流程设计图

### 订单详情模块设计

* + 1. 功能结构设计

订单详情功能结构设计图如图 6-15 所示：

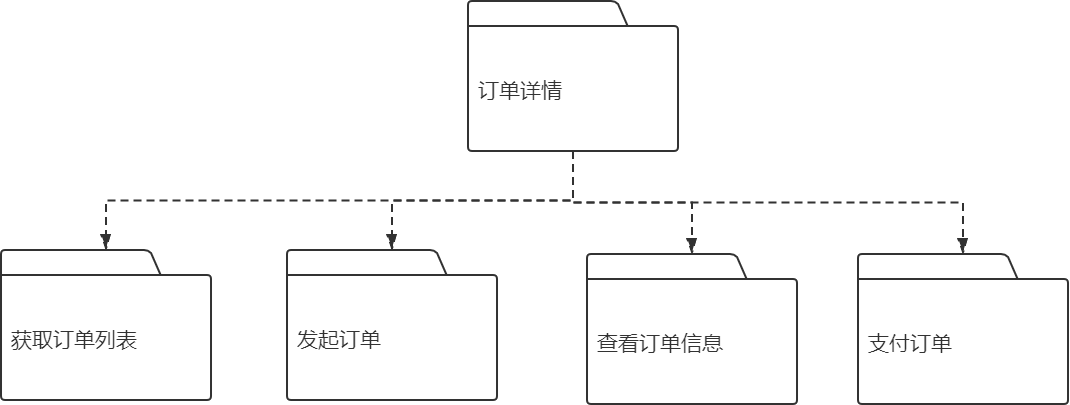


图 6-15：书籍分类管理功能结构设计图

* + 1. 功能类设计

订单详情类图包含在上面的类图 6-3 中，Order 类描述如表 6-7 所示:

表 6-7 Order 类描述

|  |  |  |
| --- | --- | --- |
| Order 类 | | |
| 类的描述：订单详情类。 | | |
| 操作描述 | placeOrder（） | 下单 |
| getDetail（） | 获取订单详情 |
| getSummaryByUser（） | 根据用户 id 分页获取订单列表（简要信息） |
| getSummary（） | 获取全部订单简要信息（分页） |
| delivery（） | 支付订单 |

Pay 类描述如表 6-8 所示:

表 6-8 Pay 类描述

|  |  |  |
| --- | --- | --- |
| Pay 类 | | |
| 类的描述：微信支付类。 | | |
| 操作描述 | getPreOrder（） | 获取订单信息 |
| redirectNotify（） | 重定向通知 |
| notifyConcurrency（） | 并发通知 |
| receiveNotify（） | 确认通知 |

* + 1. 顺序图设计

订单详情功能顺序图如图 6-16 所示：

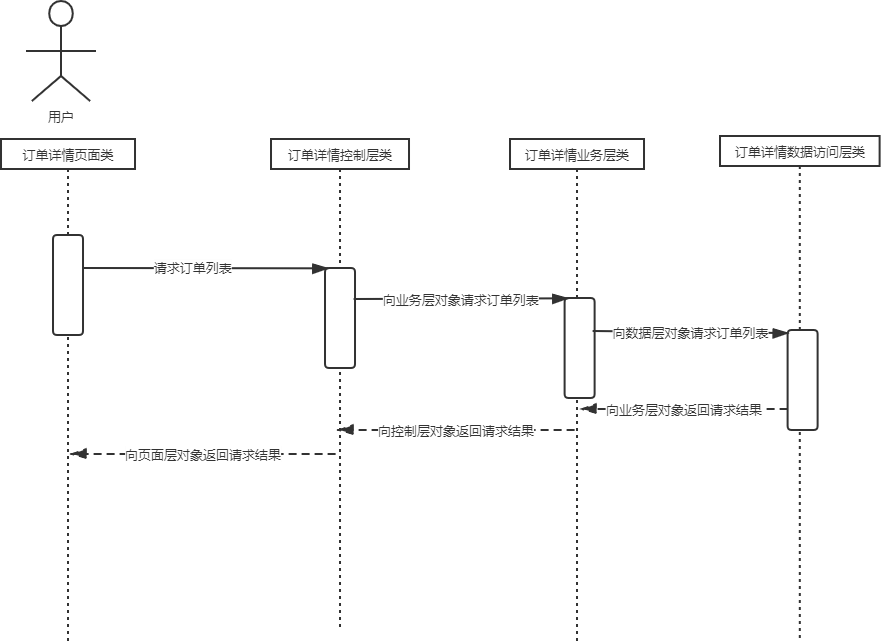


图 6-16：订单详情功能时序图

* + 1. 核心处理流程

订单详情核心处理流程设计图如图 6-17 所示：

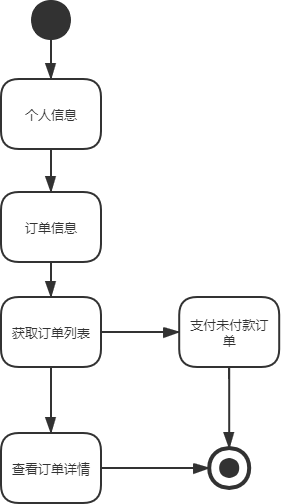


图 6-17 订单详情核心处理流程设计图

# 编码

### 代码实现与核心算法

以下为 Theme 控制器方法的具体说明。

其中 getSimpleList()方法是出查询所有主题。通过在 ThemeModel 定义的 select 方法获取所有主题，其中 with 中的 topicImage 和 headImg 方法是在 ThemeModel 中定义的，作用是把当前 Theme 模型中的 topic\_image\_id,head\_image\_id 两个字段声明与 image 模型的 id 字段关联，最后验证一下是否获取到了主题内容，如果没有获取到，则返回提示。

getThemeById()方法是根据 id 查询指定专题，调用 ThemeModel 中定义的 get 方法获取，这里 with 中比之前多了一个 products 方法，这个方法是用来声名两个模型之间的关联关系，以为图书和分类之间是多对多的关系。

addTheme()方法是新增精选主题。调用 ThemeModel 的静态方法 create()来新增一个精选主题。create()方法成功时返回的是模型的对象实例，这里我们调用实例内置的 isEmpty() 方法来检测下这次新增操作是否成功 。

delTheme() 方法是删除主题， 之前在 ThemeModel 下已经封装了一个静态方法delTheme(),在这个方法里面实现了删除的逻辑并返回执行的结果，这里直接调用就好了。同时因为删除操作的特殊性，加上了一条日志语句，保存删除的日志信息。

增删查都实现了，只剩下修改了，修改操作比较麻烦分成三个方法实现。分别为

updateThemeInfo()，removeThemeProduct()，addThemeProduct()。

其中 updateThemeInfo()方法用于实现更新精选主题的基础信息，通过定义的 patch 方法获取更新专题的信息，并通过 get 方法查询当前专题是否存在，不存在就返回提示，存在的话就更新专题信息。

removeThemeProduct()方法用于实现从精选主题中移除某个商品的关联，这里使用注解参数验证，要求传递一个精选主题的 id 和一个由商品 id 组成的数组。由于之前在

Theme 模型类下定义了一个 products()方法声明了主题表和商品表的多对多关系，所以这里直接使用$theme->products()->detach($products)这种方式，框架会自动删除中间表中

theme\_id 为$id，product\_id 为$products 数组内元素值的记录。

addThemeProduct()方法实现新增商品关联到指定的精选主题中，这里同样使用了注解参数验证，要求传递一个精选主题的 id 和一个由商品 id 组成的数组。同样借助之前Theme 模型类下定义的 products()方法，实现中间表的记录插入。

以下代码为 Theme 控制器的核心代码:

class Theme

{

/\*\*查询所有精选主题\*/

public function getSimpleList(){

$result = ThemeModel::with('topicImg,headImg')->select(); if ($result->isEmpty()){

throw new ThemeException([

'msg'=>'没有查询到主题内容'

]);

}

return $result;

}

/\*\*

* 查询指定主题详情
* @param('id','精选主题 id','require|number')

\*/

public function getThemeById($id){

$theme = ThemeModel::with('topicImg,headImg,products')->get($id); return $theme;

}

/\*\*

* 新增精选主题
* @validate('ThemeForm')
* @throws ThemeException

\*/

public function addTheme()

{

// 获取 post 请求参数内容

$params = Request::post();

// 调用模型的 create()方法创建 theme 表记录，内容是获取到的参数内容，并仅允许写入数据表定义的字段数据

$theme = ThemeModel::create($params, true); if ($theme->isEmpty()) {

throw new ThemeException([

'msg' => '精选主题新增失败'

]);

}

return writeJson(201, ['id' => $theme->id], '精选主题新增成功！');

/\*\*

* @auth('删除精选主题','精选主题管理')
* @param('ids','待删除的主题 id 数组','require|array|min:1')

\*/

public function delTheme()

{

$ids = Request::delete('ids');

// 调用模型内封装好的方法

$res = ThemeModel::delTheme($ids);

if (!$res) throw new ThemeException(['msg' => '精选主题删除失败']);

// 记录本次行为的日志

Hook::listen('logger', '删除了 id 为' . implode(',', $ids) . '的精选主题'); return writeJson(201, [], '精选主题删除成功！');

}

/\*\*

* 更新精选主题信息
* @validate('ThemeForm.edit')

\*/

public function updateThemeInfo($id)

{

$themeInfo = Request::patch();

$theme = ThemeModel::get($id);

if (!$theme) throw new ThemeException(['msg' => '指定的主题不存在']);

$theme->save($themeInfo);

return writeJson(201, [], '精选主题基础信息更新成功！');

}

/\*\*

* 移除精选主题关联商品
* @param('id','精选主题 id','require|number')
* @param('products','商品 id 列表','require|array|min:1')

\*/

public function removeThemeProduct($id)

{

$products = Request::post('products');

$theme = ThemeModel::get($id);

if (!$theme) throw new ThemeException(['msg' => '指定的主题不存在']);

$theme->products()->detach($products);

return writeJson(201, [], '精选专题删除商品成功');

}

/\*\*

* 新增精选主题关联商品
* @param('id','精选主题 id','require|number')
* @param('products','商品 id 列表','require|array|min:1')

\*/

public function addThemeProduct($id)

{

$products = Request::post('products');

$theme = ThemeModel::get($id);

if (!$theme) throw new ThemeException(['msg' => '指定的主题不存在']);

$theme->products()->saveAll($products);

return writeJson(201, [], '精选专题新增商品成功');

}

}

### 代码优化分析

TP5 框架内置的验证规则只能校验参数是不是数组，而轮播图管理模块的需求是除了校验是不是数组，我们还要校验数组的内容，而且不同场景（新增、编辑）下，这个验证内容的规则也有点不一样，这就需要自定义验证规则了。

我们定义了两条自定义验证规则——checkAddItem()和 checkEditItem()，规则的内容就是遍历参数的值，判断值是否存在或者合法。从方法名称我们可以看出，这是分别对 应新增轮播图元素和编辑轮播图元素时用的。那么怎么让这两条规则在不同场景下生效呢？ 方法就是上面的 sceneEdit()和 sceneAdd()，这两个方法定义了两个场景，分别是 edit 和add，每个场景内都执行了一个 return $this->append('参数名', '规则名')这段代码， 这段代码的作用就是，当这个场景触发时，就给指定的参数名追加一条规则。这里我们默 认只有对 items 参数的长度和类型做校验，当触发了 edit 或者 add 场景时，对应的自定 规则就追加进去。

class BannerItemForm extends BaseValidate{

protected $rule = [

'items' => 'array|require|min:1',

];

public function sceneEdit()

{

return $this->append('items', 'checkEditItem');

}

public function sceneAdd()

{

return $this->append('items', 'checkAddItem');

}

protected function checkAddItem($value)

{

foreach ($value as $k => $v) { if (!empty($v['id'])) {

return '新增轮播图元素不能包含 id';

}

if (empty($v['img\_id']) || empty($v['key\_word']) || empty($v['type']) || empty($v['banner\_id'])) {

return '轮播图元素信息不完整';

}

}

return true;

}

protected function checkEditItem($value)

{

foreach ($value as $k => $v) { if (empty($v['id'])) {

return '轮播图元素 id 不能为空';

}

if (empty($v['img\_id']) || empty($v['key\_word']) || empty($v['type'])) { return '轮播图元素信息不完整';

}

}

return true;

}}

# 测试

### 测试方案设计

* + 1. 测试策略
       1. 单元测试。
       2. 黑盒测试 。
       3. 边开发边测试。
    2. 测试进度安排

测试进度安排表如图 8-1 所示：

表 8-1 测试进度安排表表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试阶段 | 测试任务 | 工作量估  计 | 人员分配 | 起止时间 |
| 第一阶段功能测试 | 地址信息管理测试用户信息获取测试获取订单信息测试购物车管理测试  后台轮播图管理测试后台专题管理测试 后台图书管理测试  后台用户权限管理测试 | 2 日 |  | 2019 年 4 月 12 日至  2019 年 4 月 13 日 |
| 第二阶段系统测试 | 所有模块的组合测试确  定所有业务流向和数据都是正确的 | 2 日 |  | 2019 年 4 月 14 日至  2019 年 4 月 15 日 |
| 第三阶段  性能测试 | 多用户进行访问，交替  进行测试 | 1 日 |  | 2019 年 4 月 16 日 |
| 第四阶段  安装部署测试 | 对与项目部署进行测试 | 1 日 |  | 2019 年 4 月 17 日 |
| 第五阶段  兼容测试 | 软件的运行情况 | 1 日 |  | 2019 年 4 月 18 日 |

* + 1. 测试资源

谷歌浏览器，微信

* + 1. 关键测试点

1. 兼容性测试

测试在不同系统的微信客户端是否显示正常。微信端需要考虑 IOS 和 Android 的尺寸和分辨率的影响。根据友盟和百度流量研究院选取主流的分辨率和尺寸进行测试。

1. 系统功能测试
2. 微信登陆测试：用户点击登陆按钮，测试是否会发起弹窗，弹窗点击确定后是否能获取到正确的用户信息。
3. 购物车测试：加入多件图书测试总价计算是否正确，依次减少图书购买数量，测试当图书数量减少到 0，图书是否会消失在购物车中，测试当购买数量超过库存量时，是否能正常购买。
4. 专题管理测试：测试增删改查功能是否正确，如果新增和修改中传入空数据是否可以通过。测试新增修改删除后专题列表是否发生变化。

### 测试用例构建

* + 1. 测试用例编写约定

1. 全面性。全面测试系统的各个功能以及整体业务流程，并测试在各种情况下系统的返回结果，与预期结果对比，检查系统是否会出现错误。
2. 符合正常业务规则。测试的数据要尽量符合用户实际工作中的业务流程，贴合用户实际。
3. 地名、人名，电话号码等应具有模拟功能，符合正常的命名规则；
   * 1. 测试用例设计

测试项目表如下：

表 8-1 测试项目表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 内容和目的 | 测试结果 |
| 1 | 轮播图管理  （新增，修改，删除） | 新增和修改操作输入正  确信息 | 成功保存，轮播图列表正  确显示对应操作结果 |
| 新增和修改操作输入错  误信息 | 操作失败，重新输入 |
| 2 | 书籍分类 | 添加书籍到分类 | 添加成功 |
| 3 | 订单查看 | 发起订单后查看个人信  息页面测试订单信息 | 订单信息正确 |

续表 8-1

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 内容和目的 | 测试结果 |
| 4 | 专题管理  （新增，修改，删除） | 新增和修改操作输入正  确信息 | 成功保存，专题列表正确  显示对应操作结果 |
| 新增和修改操作输入错  误信息 | 操作失败，重新输入 |
| 5 | 图书管理  （新增，修改，删除） | 新增和修改操作输入正  确信息 | 成功保存，图书列表正确  显示对应操作结果 |
| 新增和修改操作输入错  误信息 | 操作失败，重新输入 |
| 6 | 用户权限管理  （新增，修改，删除） | 注册用户账号并分配权  限后，测试当前账号能否登陆，权限信息是否正确 | 账号密码能正常登陆，权限与分配的一致 |
| 修改用户信息和权限后测试账号修改前的密码能否登陆，修改后的密码能否登陆，修改后权限是  否正确 | 修改前的账号密码无法登陆，使用修改后的密码可以正常登陆。账号权限与修改后的保持一致。 |
| 删除用户账号测试是否  还能登陆 | 无法登陆 |
| 7 | 购物车管理 | 测试购物车增删改功能  是否能正确返回结果 | 正确返回 |

* + 1. 关键测试用例

1. 对某一书籍库存量 x,下单 x+1 件，观察发货情况。
2. 对用户地址输入空字符串，观察是否会报错。
3. 输入轮播图信息后，测试轮播图列表是否显示新增的轮播图信息，点击编辑按钮，测试输入空字符串是否保存。修改后测试轮播图列表是否更新信息。点击删除按钮测试列表中是否当前轮播图是否存在。
   * 1. 测试用例维护
4. 库存量实验完毕后及时更新库存量
5. 将测试用的地址与书籍信息删去

# 总结与展望

### 设计工作总结

经过这段时间的学习和实践，终于自己完成了一个完成的项目，在毕设期间学到了非常多的细节，之前写代码只顾着逻辑，只要能实现一个功能就行了，后面实际做项目的时候才明白代码复用和代码规范的重要性。还有对于用户数据的验证，因为现实中做项目不仅仅知识完成这一个功能还要考虑各种影响，如果不对用户数据进行全面的验证，很容易在实际操作中出现各种 bug，开发过程中碰到 bug 确实是非常折磨人，记得刚开始写后台管理系统的时候，页面一直没法显示，查了半天才知道是跨域的问题把环境改一下就好了。

这次毕业设计不仅是对自己知识掌握程度的一次检验也是对自己心态的一次锻炼。

### 未来工作展望

完成本系统后，我对于工程有了一个初步的理解，虽然功能还有很多地方不够完善， 但基本的业务流程还是实现了，这个系统还是一个比较粗糙的系统，还有很多地方可以去完善，小程序的前端页面还是不够美观，个人信息那里因为微信的 api 改了，虽然照着最新的 api 要求实现了，但还是留了一大横杠在那，而且获取完之后没法消失。

系统功能方面也只是实现了最基本的功能，还有很多可以扩充的功能比如图书的优惠卷，活动打折，微信支付等功能，还有很多安全性的东西还可以继续实现，比如对于数据的验证，验证器里还是比较简单的规则验证，考虑到用户数据的多样性，应该多自定义一些验证规则，比如对于上架图书的信息核对，以及用户已经下了订单但是一直不付款，应该设计一个订单的倒计时。还有对用户购书信息的总结，比如销量总结可以看出哪一类的图书更受欢迎，可以多进一些这种种类的图书，还有对用户开放会员，会员的具体权限。

大学结束但是也是职业生涯的起点，工作中还有许多知识需要不断地学习，不断充实自己，大学期间从一开始从没接触过代码，连 hello world 都打的磕磕绊绊的，到每天坚持写代码刷题，享受写代码途中思考的过程，找到了感兴趣的方向，并为之努力，虽然在其他人看来很累，没什么意思，但是自己能感觉到比起以前浑浑噩噩没有目标的日子现在过的要更充实更开心，期间收获了一些荣誉虽然更多的还是遗憾，毕竟不可能所有的结局都是美好的，未来不管自己是否会从事相关的工作，希望自己永远能记得自己大学期间写代码时无忧无虑的样子。

# 谢 辞

# 参考文献

* + 1. 彭静. 论微信平台中的商标侵权行为及责任认定[J]. 法制博览，2017,06:1-5.

[2]李松佳，徐飞. 基于微信公众号的高校微课平台的设计[J]. 价值工程，2017,08:86-87.

[3]段任. 基于互联网开放平台的微信公众号构建[J]. 中国管理信息化，2017,05:160-162.

[4] 田学芝. 微信公众平台价值营销模式研究--基于典型案例的分析[J]. 绿色科技，

2017,04:170-172.

[5]张琰. 基于大数据的多功能智能手环与微信平台的结合[J]. 电脑编程技巧与维护，

2017,05:31-32+47.

[6]杨霄飞. 对微信公众平台着作权问题的几点思考[J]. 法制博览，2017,02:29-30+21.

[7]毛殷. 提升学报内容在微信平台上传播质量的路径[J]. 编辑学报，2017,01:73-76.

[8]姚修杰. 微信公众账号服务平台微服务的设计与研究[J]. 通讯世界，2017,06:84-85.

[9]陈薇. 掌上社保的微信公共服务平台构建[J]. 中国管理信息化，2017,06:154-155.

[10]张艳. 微信平台在护理管理中的应用研究[J]. 中西医结合护理（中英文），2017,02:143- 145.

[11]郭艳茹. 微信商业模式探析--“内容+平台+数据”模式[J]. 商场现代化，2017,05:42-43. [12]雷磊.微信小程序开发入门与实践[M].北京: 清华大学出版社，2017

[13]张翔.微信小程序：分享微信创业 2.0 时代千亿红利[M].北京：清华大学出版社，2017 [14]方蓓工作室.微信公众平台开发最佳实践[M].北京：机械工业出版社，2015.5

[15]李炜康.基于微信的网上购物系统的设计与实现[D].南京大学，2015 [16]储广昕.分析软件开发中数据库设计理论的实践[J].信息通信，2015

[17]沈炜，徐慧，汤倩.Mysql 数据库编程技术与实例[M].北京:人民邮电出版社，2005 [18]吴挺. 基于微信公众平台的可扩展架构研究与实现[D].兰州大学，2016.

[19] 沈溶溶. 微课在 PHP 口课程教学中的应用[J]. 信息系统工程, 2019(3).

1. Wang H , Xu H , Li Q , et al. PHP-based collaborative education and management system for water hydraulic laboratory[J].Computer Applications in Engineering Education, 2018, 26(2):259-271.
2. Misek J,Zavoral F . Semantic analysis of mbiguous types in dynamic languages[J]. Journal of Ambient Intelligence & Humanized Computing, 2019, 10(7):2537-2544.
3. Lorenc A,Szkoda M ,Szarata A ,et al. Evaluation of the effectiveness of methods and criteria for product classification in the warehouse[J]. European Journal of Industrial Engineering, 2020, 14.
4. Mershad K.MQL: Mixed Query Language for Querying MySQL and HBase Databases[C]// 2019 International Conference on Innovative Trends in Computer Engineering (ITCE). IEEE, 2019.
5. Shen L . Design of data tracing method for PC component based on distributed MySQL[J]. journal of the hebei academy of sciences, 2019.
6. Brooks A . SQL tester: an online SQL assessment tool and its impact[J]. Computing reviews, 2019, 60(2):80-80.
7. Robin Nixon. Learning PHP, MySQL, and Javascript[J]. oreilly media, 2018. [27] Bora A , Bezboruah T . Some Aspects of Reliability Evaluation of Multi Service

Multi-Functional SOAP Based Web Services[J]. International journal of information retrieval research, 2018, 8(4):24-38.

1. Tao Y , Weidong Z , Xiaoqing Z ,et al. Research on and Implementation of MySQL Accessing Based on C Programming Language[J]. journal of chengdu university(natural science edition), 2017.
2. Yanfen L . Access to the MySQL Database Based on C++ and Applications of It in Geomatics Work[J]. beijing surveying and mapping, 2017.
3. Usmani Z A , Khan S , Kazi M , et al. ZAIMUS: A department automation system using data mining and web technology[C]// 2017 International Conference on Nascent Technologies in Engineering (ICNTE). IEEE, 2017.

## 附录A 外文翻译—原文部分

外文出处：

Autor：Martin Ngobye

The Source of Article：Getting php to talk to Mysql

## PHP Language Basics

PHP Language Basics Active Server Pages (PHP) is a proven, well-established technology for building dynamic Web applications, which provides the power and flexibility you need to create anything from a personal, Web based photo gallery to a complete catalogue and shopping cart system for your next eCommerce project。One unique feature of PHP is that it lets you choose your favourite scripting language, be it JavaScript or PHP ; however, PHP is by far the most popular choice. In this article, I'll bring you up to speed on the basic syntax of the PHP language, including variables, operators, and control structures. This article is the second in a series teaching PHP. Specifically, the goal of this series is to teach you all you need to know to create dynamic Web sites using PHP. This article picks up right where the previous article in the series, Getting Started with PHP, left off. Variables Here is the listing for the first PHP script I helped you create in the previous article:

<html>

2 <head>

3 <title> My First PHP Page </title> 4 </head>

5 <body> 6 <?php

7 ' Write out a simple HTML paragraph

8 Echo "<p>This is a test of PHP.</p>" 9 ?>

10 </body> 11 </html>

As I admitted in that article, this is a pretty uninteresting example of an PHP script. When it comes right down to it, this script doesn't do anything a plain, old HTML page couldn't do. Oh sure, I gave a slightly more interesting example that displayed the current server time, but to be really useful a script needs

to perform some form of calculation, or manipulate dynamic information to present it in some interesting way. The language used for writing most PHP programs, and which I'll be using throughout this series, is called PHP . Like most programming languages, PHP lets you store data in variables. A variable may be thought of simply as a named location in memory where data may be stored. PHP is what is known as a loosely typed language, which means that a particular variable may store any kind of information, be it a number, a piece of text, a date, or some more complicated chunk of data (as opposed to strictly typed languages where you can only store one kind of information

in each variable). Before you can use a variable, though, you must declare it; that is, you must let PHP know that you want to create a variable with a particular name. Let's look at a basic example to help solidify these concepts in your mind. Say you were writing a Web page that performed conversions between Celsius and Fahrenheit temperatures. In countries where Celsius is used, 20&deg;C is commonly accepted as the value for room temperature. The following code creates a variable called intRoomTempC, and then assigns it a value of 20: New Revised 2nd Edition Out NOW! "Build Your Own Database Driven Website Using PHP & MySQL" Fully updated for PHP 4.3. Installation instructions for Mac OS X Full index provided New wider book size Enhanced fonts New cover design Lay- flat spine All content revisited Download the First 4 Chapters FREE Tell me more about this top-selling book. $ intRoomTempC ' Create a variable intRoomTempC = 20 ' Assign the variable a value of 20 The keyword $ in the above is short for $ension, and is used to tell PHP to create a variable with the name specified (in this case, intRoomTempC). Why '$ension', you ask? I agree, it's not the most obvious choice, but basically it refers to what you're asking PHP to do. When creating a variable, PHP needs to assign some space in memory to store whatever value(s) will be placed in the variable, and part of that task is to figure out the size ($ension) of the space that needs to be allocated. In any case, creating a variable is as simple as typing $ followed by the name of the variable. The second line of the above example assigns a value to the variable that was just created; specifically, it stores the number 20 in the variable. The equals sign (=) is called the assignment operator because it is used to assign values to variables. During the course of this article, you'll meet many other operators that do other weird and wonderful things to variables and the values they store. You should always create a

variable before assigning it a value, and you'll usually want to assign the variable a value before putting it to use. Trying to assign a value to a variable that does not exist, however, will cause PHP to automatically create a new variable with the given name. This is called implicit declaration, because a new variable is declared implicitly as a result of your trying to assign a value to a variable that doesn't exist. Since you are free to use implicit declaration for all of your variables, you may be wondering what the point is of using the

$ command to create each and every variable by hand. The answer has to do with how easy you want it to be to find typing mistakes in

your code. PHP provides another command, Option Explicit, which causes PHP to disallow implicit declarations and instead display an error message whenever you try to assign a value to a non-existent variable. Why would you want this to happen? Consider the following example: $ intRoomTempC ' Create a variable intRomTempC = 20 ' Assign the variable a value of 20 If you have a keen eye, you may have noticed that the variable name is misspelled on the second line. This is the kind of mistake that even experienced programmers make all the time. With implicit declaration enabled, the second line will create another new variable called intRomTempC and will store the value in that variable instead. Now, if the rest of your script expects that value to be stored in intRoomTempC, you're going to run into trouble. In a large script, tracing such a problem back to one little typing mistake can be very time consuming. That's where Option Explicit comes in: Option Explicit ' Disable implicit declaration $ intRoomTempC ' Create a variable intRomTempC = 20 ' Assign the variable a value of 20 This time, PHP will report the typing mistake as an illegal implicit declaration, displaying an error message to that effect with the exact line number where the typing mistake was made. For this reason, I tend to explicitly declare all my variables with

$ and specify Option Explicit on the first line of all of my PHP scripts. It might take slightly longer to type, but it saves a lot of headaches when something goes wrong. A shortcut exists for creating several variables at once on the same line. For instance, the following line would create two variables, intRoomTempC, and intFreezingC: $ intRoomTempC, intFreezingC ' Two variables in one line By now you may be wondering about my naming convention for variables. The two variables created in the above snippet both begin with int. I'm using this prefix to indicate that these variables will contain integers (whole

numbers). You can feel free to name your variables whatever you like and store whatever kind of data you like in them, but I prefer to use this convention as a helpful reminder of the type of information in each variable. This practice of prefixing variable names with a clue as to their type is known as Hungarian notation, and I'll introduce additional prefixes for other data types as they arise over the course of this series. The Web has grown beyond the point where an online brochure will satisfy a typical company's needs for its Web presence. If you aim to market yourself as a Webmaster these days, you need to have some skill building online applications – Web sites that users can interact with, whether to get something done (e.g. send email), get information targeted to their specific needs (e.g. a real-time stock quote), or to interact with other users (e.g. an online community). In this series of articles, I’ll guide you through the process of learning one of the most popular frameworks for creating dynamic Web sites such as these – Active Server Pages (PHP). If you can secure a strong knowledge of PHP, as well as some

practical experience building Web sites with it, you should never have trouble getting work as a Web developer. A quick search of your favourite online job directory with the keyword 'PHP' should be more than enough to convince you of that. In this first article, I'll help you get your feet wet by introducing the PHP programming language, and how to use it to write dynamic Web pages with PHP. Before I get to that, I shall stop to explain how server-side scripting, and PHP in particular, differs from other Web scripting technologies that you may be familiar with, such as client-side JavaScript. This will get you armed with the proper vocabulary and ensure that we're on the same page before launching headlong into the brave, new world of PHP. Server-Side Scripting To understand where PHP fits into the big picture of Web development, you need to understand the concept of a server-side scripting language. If you've programmed Web pages in Perl, PHP, JSP, or Cold Fusion before, you can safely skip this section – all of those are server-side scripting languages, and PHP works in much the same way. If you're coming to PHP armed only with knowledge of HTML (and perhaps with some CSS and/or JavaScript experience) then you'll find that server-side scripting is quite a bit different. Let me begin by giving you a quick review of how standard, non-PHP Web pages work. As shown in Figure 1, the Web browser on the client computer (the computer belonging to the user) makes a request for a page, say file.html . Assuming the requested file exists on the

Web host computer where the Web Server software can find it, that software replies to the request by sending the file back to the browser (2). Any additional files (images, for example) required to display the page are requested and received in the same way. The protocol used for this exchange, and indeed for all communication between Web browsers and Web servers is called Hypertext Transfer Protocol (HTTP). If you've ever used any JavaScript in your pages, you know that the requested Web page (file.html) can contain, in addition to plain HTML code, small programs written in JavaScript. These programs, or scripts, are read and executed by the Web browser while the page is displayed in the browser. So the Web browser must understand not only how to read HTML and display text and images, but it must also be able to run JavaScript programs appearing inside Web pages. This arrangement, where the Web browser runs the script after receiving it from the Web server, is called client-side scripting. The name makes sense – all of the script runs on the client-side – the right- hand side of Figure 1. The Web server is completely oblivious to whether the file it is sending contains a script or not; it’s all up to the browser (the client) to handle execution of the script. PHP fits into a different category of technologies, called server-side scripting, where it is no longer the browser running on the client that is responsible for running the script; instead, it is the Web server that runs the script. This process is illustrated in Figure 2. As before, the Web browser requests a file . In this case, however, the filename ends with .php (file.php, for example), branding it as a file containing an PHP script that needs to be processed by the server. The server recognizes this, and

instead of directly sending the requested file back to the browser, it sends the file to the PHP scripting engine . The engine is a component of the Web server software that can interpret PHP scripts and output the results as HTML. The trick here is that any given script can output different HTML each time it is run, so what comes out of the PHP engine can be different for each client (browser) request. That dynamically generated page is then sent to the browser in response to its request , in exactly the same way as the static page was sent in the previous example. Just as when the page contained client-side JavaScript and the server was completely unaware of this fact, when the page contains server-side PHP script, the browser does not know this at all. The PHP code contained in the page is interpreted and converted to plain HTML by the

PHP engin

## 附录B 外文翻译—译文部分

PHP Language Basics

我承认在文章中，这是一个 PHP 脚本完整而没有兴趣的例子。当写下例子，这个脚本不能做普通传统网页不可能做的任何事情。当我给了一个稍微有兴趣的例子，它显示当前服务器的时间，但是真正有用的脚本使用有兴趣的方式需要执行一些计算的格式，操纵目前动态信息。

<html>

2 <head>

3 <title> My First PHP Page </title> 4 </head>

5 <body> 6 <?php

7 ' Write out a simple HTML paragraph 8 Echo "<p>This is a test of PHP.</p>" 9 ?>

10 </body> 11 </html>

写 PHP 程序使用最多的语言，我们通过这个系列叫做 PHP 。像许多其它程序语言一样, PHP 让你用变量存储数据。一变量可以被认为是内存里的一个位置，这个位置是用来存储数据的。PHP 被认为是最自由的一类程序语言，这意味着一特殊的变量可以存储任何类型的信息，可以是一个数字，一篇文章， 一个数据或者一些复杂的数据堆（与严格程序语言相对的，你只能用每一个变量来存储一种类型的信息）。在你使用一个变量时，虽然你必须声明它，即你必须让 PHP 了解你想用一个特殊的名字创建一个变量。

让我们看一个例子来巩固在你脑子里的概念。即你在做一张网页是执行 Celsius 和 Fahrenheit 温度的转换。Celsius 在乡村被使用，20&deg。C 被认为是一个房间温度的值。下面的代码是创建一个intRoomTempC 的变量，然后赋值为 20： New Revised 2nd Edition Out NOW! "Build Your Own Database Driven Website Using PHP & MySQL" Fully updated for PHP 4.3. Installation instructions for Mac OS X Full index provided New wider book size Enhanced fonts New cover design Lay-flat spine All content revisited Download the First 4 Chapters FREE Tell me more about this top-selling book. $ intRoomTempC ' Create a variable intRoomTempC = 20 ' Assign the variable a value of 20 在上面关键字$的范围比较小，在使用 PHP 创建一个名字确定的变量（就如 intRoomTempC）你会问为什么要范围呢?我同意，这不是最清晰的选择，但是基本上涉及了 你问 PHP 所做的一切。当创建一个变量，PHP 需要在内存中留一些空间来存储放在变量中的任何值，它的部分任务是明确所分配的空间大小。在任何情况下，创建一个变量就像打$ 后跟变量

名。

上面例子的第二行赋了一个值给刚创建的变量，明确了它变量赋值是数字 20。等号是一个明确的操作符,因为它来确定变量的赋值。在本文的课程学习中，你将碰到许多其他的操作符，用来给变量另外离奇而有意思的东西，以及它们存储的值。 你应该在确定一个值时，先创建一个变量，通常在使用它前给想给它一个确定的值。试着给变量一个不存在的值，不管怎么样，将引起自动创建一个全名的新变量。这个叫隐式说明,因为新的变量被隐式的说明由于你试者给一个不存在的变量赋值。由于你能自

由的对你所有的变量进行隐式说明，你可以知道的关键是手动的用$命令创建任何一个和每一个变量。

如何容易处理的找出在你的代码中的打字中错误的解答。PHP 提供了另一个命令 Option Explicit ，在你试着给一个不存在的变量赋值时，它将导致不允许的隐式说明和代替显示错误信息。你想为什么会发生这种情况呢 ？考虑一下下面的例子： $ intRoomTempC ' Create a variable intRomTempC = 20 ' Assign the variable a value of 20 如果你有一双敏锐的眼睛，你可可以注意到第二行的变量名拼写错误。这种类型的错误甚至在有经验的程序员中也一直存在。在变量能隐式说明的帮助下，第二行将创建另一个新的变量 intRomTempC，然后代替那个变量存储值。现在，你的剩余的脚本希望在 intRoomTempC 存储值，你的运行将出现问题。在更大的脚本中，对像这样一个小的打字中错误的追踪是非常的费时间的。Option Explicit 的来源： 这时，PHP 将用非法的隐式说明报告打 印错误，用精确的每一行数字显示打字中错误信息。因为这个原因，我往往用$来隐式说明所有的我的 变量以及明确我的PHP 脚本中第一行的 Option Explicit。它可能花更多的时间打，但是它省去当问题出现时的许多头痛的问题。 在同一行上创建几个变量的一个捷径。例如，下面一行将创建两个变量

ntRoomTempC, 和 intFreezingC： $ intRoomTempC, intFreezingC ' Two variables in one line

到目前为止你可以知道我对变量命名的的惯例。在上面的小片段中用 int 开头创建了两个变量。我使用的这些变量都包含整型前缀（全部数字）的。你可以自由的命名你的变量，无论你喜欢什么以及你喜欢在他们中存储什么的数据，但是我更喜欢用这个惯例作为每一个变量的信息类型的有用的提醒者。关于前缀变量名字的线索的实行是以 Hungarian notation 而著称的，对上面这一些列都了解的前提下， 我将介绍其它的数据类型的附加前缀。

Web 发展已经超出了在线小册子将满足典型公司对 Web 存在的需求的关键点。如果你作为 Web 站点管理员的市场瞄准的是自己，你需要有构建在线应用的一些技能—Web 站点能使用户相互作用,或者做一些事情(如发邮件),得到他们明确需要的目标信息(如实时库存引文),或者和别的用户相互作用(如在线社团)。

在这一系列文章，我通过学习最流行的框架的过程引导你创建动态 Web 站点像这些 PHP。如果你能够获得 PHP 大量知识，还有一些实践用它来构建 Web 站点，你作为一个 Web 开发者应该做工作决不会问题。用关键字 PHP 来快速查找你喜爱的在线工作目录应该更能够使你相信。 在文章的第一部分，我将帮助你通过介绍 PHP 程序语言让你感觉你是新手，然后怎么使用 PHP 来写动态 Web 网页。在我得到那个以前，我将停止解释服务器端脚本，特别是 PHP，与你可能熟悉其它的 Web 脚本技术不同，像客户端Java 脚本语言。这使你用适当的词汇武装自己确保我们在头向前的勇敢进入 PHP 新世界。

服务器端的脚本 为了理解 PHP 适合 Web 开发的大图片，你需要理解服务器端的脚本语言的概念。如果你用 Perl, PHP, JSP 或者 Cold Fusion 编写 Web 页面前，你能够安全略过这部分—所有这些是服

务器端的脚本语言，PHP 工作在许多相同的方式。如果你只是用 HTML 知识来辅助 PHP(也许有过一些 CSS

或者 Java 脚本的经历),那么你将发现服务器端的脚本是有点不同之处的。

让我开始给你一个标准的快速回顾，非 PHP Web 网页工作。图 1 中显示客户端的计算机（计算机属于用户）Web 浏览器对页面做请求即 html 文件。假定在 Web 主机上存在被请求的文件，Web 服务器软件能发现它，软件能通过发回文件给浏览器来回应请求。任何附加文件（如图象文件）要求显示页面是用同样的方式请求和接收。协议用做交易，的确 Web 浏览器和 Web 服务器通信是叫做 HTTP。 如果你曾经在你的网页上用过 Java 脚本，你知道请求 Web 网页（html 文件）能包含除了普通 HTML 代码，用Java 脚本写的小程序。这些程序或者脚本通过 Web 浏览器是可读可执行的当页面显示在浏览器上时。所以 Web 浏览器必须理解怎么读 HTML 和显示文本和图象，但是它必须能运行显示在 Web 页面的 Java 脚本程序。浏览器收到 Web 服务器这个安排后运行脚本，这个安排叫客户端脚本。在图 1 右手边所有名字有意义的脚本在客护端运行。Web 服务器文件发送是否包含脚本是完全健忘的，这是浏览器（客户端） 处理执行脚本。

PHP 适合不同的技术种类，叫客户端脚本，它不在浏览器运行客户端而是在 Web 服务器上负责运行脚本。这个过程在图 2 中描绘。 像前面一样，Web 浏览器请求文件。然而在这样的情况下，文件名末尾用 PHP（如文件 PHP），标志着这个文件包含 PHP 脚本需要服务器来处理。服务器认识这个，代替直接送请求文件给浏览器，它送文件给 PHP 脚本引擎。引擎是 Web 服务器的重要软件，它能够解释 PHP 脚本和作为 HTML 输出结果。这里的诀窍是任何已给的脚本每次运行能输出不同的 HTML，所以 PHP 引擎的出现能对每一个客户（浏览器）请求是不同的。那动态生成的网页被送到浏览器以响应它的请求，这跟以前的静态网页有相同的方式。

正像页面包含客户端的 Java 脚本和服务器是完全没有意识到这个事实，当页面包含服务器端的PHP 脚本，浏览器是根本不知道的。PHP 代码包含在页面被解释和传送到 PHP 页面看起来像其它任何普通的 Web 页面。所有的工作在服务器端，所以名字就是服务器端脚本。

## 附录C 软件使用说明书

主界面最顶部为小程序名字图书商城，点击名字右侧三个点的按钮可以一键分享小程序，往下就是轮播图，这里一共有四幅轮播图，支持滑动，点击轮播图会进入图书详情， 轮播图下方为精选主题，可以点击查看推荐主题，最下面是功能栏分别对应首页，图书分类，购物车，我的信息。小程序项目主界面附录 C 图 1 所示：



附录 C 图 1 项目主界面

点击首页的轮播图如我们点击第一张轮播图《活着》，可以跳转到图书详情界面。轮播图页面如附录 C 图 2 所示：

附录 C 图 2 轮播图界面

如果点击首页的第三个专题推荐，用户可以跳转到校园青春漫画推荐专题页面专题推荐界面如附录 C 图 3 所示：



附录 C 图 3 专题推荐界面

点击首页最下方的分类按钮，用户可以跳转到图书分类子界面图书分类界面给各类图书分好了类别，可以直接查看各类型图书。图书分类界面如附录 C 图 4 所示：

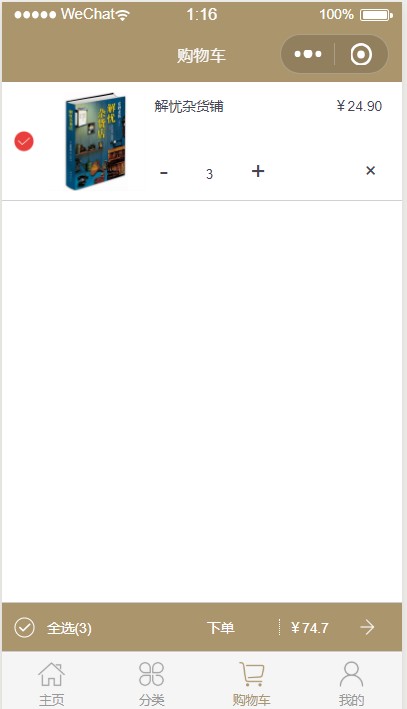


附录 C 图 4 图书分类界面

前面几个子界面都是为了让用户更好的找到感兴趣的图书，用户找到感兴趣的图书点击图书，跳转图书详情页面.可查看图书详细信息并选择加入购物车的数量。图书详情界面附录 C 图 5 所示：

附录 C 图 5 图书详情界面

用户进入购物车。可以查看购物车中的图书并修改图书的购买数量，点击购物车中图书左边的红点可以选择是否选择该图书，选择完图书后点击下单即可生成订单。

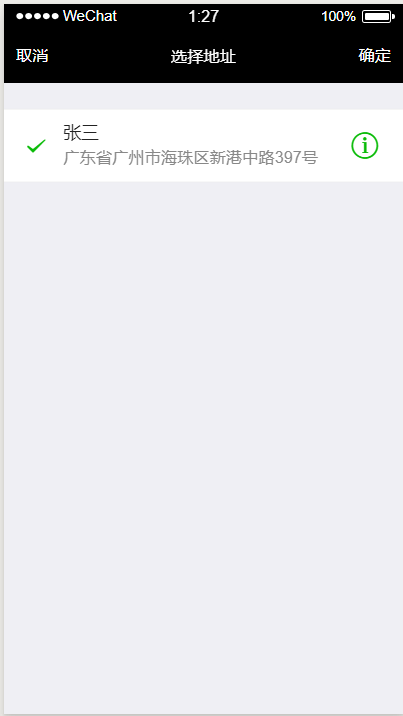
购物车界面如附录 C 图 6 所示：

附录 C 图 6 购物车界面

用户点击最下面的我的按钮就可以到达个人信息子界面，点击“请单击获取你的个人信息”会弹出微信授权，点击接收即可获取微信个人信息，同时下面也有用户的地址信息与订单信息。个人信息子界面如附录 C 图 7 所示：

附录 C 图 7 个人信息子界面

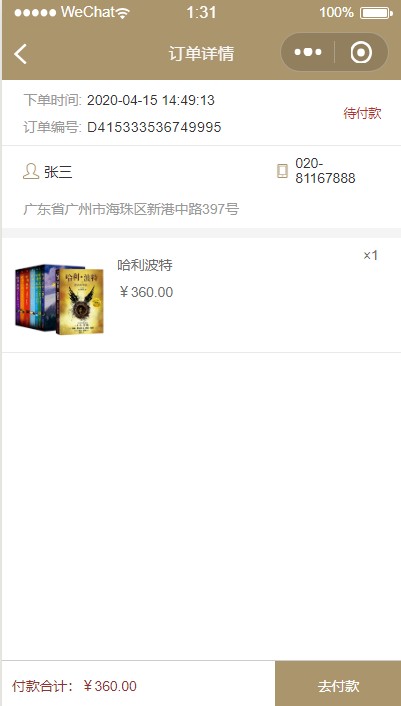
点击个人信息页面中的地址管理即可进入地址管理子界面，可以在这选择默认的地址或者新增地址。

地址管理子界面如附录 C 图 8 所示：

附录 C 图 8 地址管理子界面

点击个人信息页面中的订单详情中任意一个订单即可进入当前订单详情页面，可以在这查看订单各种信息，对于没有付款的订单，可以选择支付订单。

订单详情子界面如附录 C 图 9 所示：



附录 C 图 9 订单详情子界面

项目初始化代码如下：

## 附录D 主要源代码

import { Home } from 'home-model.js'; var home = new Home(); //实例化 首页 对象Page({

data: {

loadingHidden: false

},

onLoad: function () { this.\_loadData();

},

/\*加载所有数据\*/

\_loadData:function(callback){ var that = this;

// 获得 bannar 信息

home.getBannerData((data) => { that.setData({

bannerArr: data,

});

});

/\*获取主题信息\*/ home.getThemeData((data) => {

that.setData({ themeArr: data, loadingHidden: true

});

});

/\*获取单品信息\*/ home.getProductorData((data) => {

that.setData({ productsArr: data

});

},

});

callback&&callback();

/\*跳转到商品详情\*/

onProductsItemTap: function (event) {

var id = home.getDataSet(event, 'id'); wx.navigateTo({

url: '../product/product?id=' + id

})

},

/\*跳转到主题列表\*/

onThemesItemTap: function (event) {

var id = home.getDataSet(event, 'id');

var name = home.getDataSet(event, 'name'); wx.navigateTo({

url: '../theme/theme?id=' + id+'&name='+ name

})

},

/\*下拉刷新页面\*/ onPullDownRefresh: function(){

this.\_loadData(()=>{ wx.stopPullDownRefresh()

});

},

//分享效果

onShareAppMessage: function () { return {

title: '图书商城', path: 'pages/home/home'

}

}

})

购物车管理源代码：

class Cart extends Base{ constructor(){

super(); this.\_storageKeyName='cart';

};

/\*

\* 获取购物车

\*/ getCartDataFromLocal(flag){

var res = wx.getStorageSync(this.\_storageKeyName); if(!res){

res=[];

}

//在下单的时候过滤不下单的商品， if(flag){

var newRes=[];

for(let i=0;i<res.length;i++){ if(res[i].selectStatus){

newRes.push(res[i]);

}

}

res=newRes;

}

return res;

};

/\*

\*获得购物车商品总数目,包括分类和不分类

\*/ getCartTotalCounts(flag){

var data=this.getCartDataFromLocal(), counts1=0,

counts2=0;

for(let i=0;i<data.length;i++){ if (flag){

if(data[i].selectStatus) { counts1 += data[i].counts; counts2++;

}

}else{

counts1 += data[i].counts; counts2++;

}

}

return {

counts1:counts1, counts2:counts2

};

};

/\*本地缓存 保存／更新\*/ execSetStorageSync(data){

wx.setStorageSync(this.\_storageKeyName,data);

};

add(item,counts){

var cartData=this.getCartDataFromLocal(); if(!cartData){

cartData=[];

}

var isHadInfo=this.\_isHasThatOne(item.id,cartData);

//新商品if(isHadInfo.index==-1) {

item.counts=counts;

item.selectStatus=true; //默认在购物车中为选中状态cartData.push(item);

}

//已有商品else{

cartData[isHadInfo.index].counts+=counts;

}

this.execSetStorageSync(cartData); //更新本地缓存return cartData;

};

/\*

\* 修改商品数目

\* \*/

\_changeCounts(id,counts){

var cartData=this.getCartDataFromLocal(), hasInfo=this.\_isHasThatOne(id,cartData);

if(hasInfo.index!=-1){ if(hasInfo.data.counts>1){

cartData[hasInfo.index].counts+=counts;

}

}

this.execSetStorageSync(cartData); //更新本地缓存

};

/\*

\* 增加商品数目

\* \*/ addCounts(id){

this.\_changeCounts(id,1);

};

/\*

\* 购物车减

\* \*/ cutCounts(id){

this.\_changeCounts(id,-1);

};

/\*购物车中是否已经存在该商品\*/

\_isHasThatOne(id,arr){ var item,

result={index:-1};

for(let i=0;i<arr.length;i++){ item=arr[i]; if(item.id==id) {

result = {

index:i, data:item

};

break;

}

}

return result;

}

/\*

\* 删除某些商品

\*/ delete(ids){

if(!(ids instanceof Array)){ ids=[ids];

}

var cartData=this.getCartDataFromLocal(); for(let i=0;i<ids.length;i++) {

var hasInfo = this.\_isHasThatOne(ids[i], cartData); if (hasInfo.index != -1) {

cartData.splice(hasInfo.index, 1); //删除数组某一项

}

}

this.execSetStorageSync(cartData);

}

}