

**本 科 毕 业 设 计**

题 目 基于微信公众平台图书销售

系统的设计与实现

# 学 院 信息科学与工程学院

专 业 软件工程

学生姓名 程斌

学 号 1122005 班级 2012-3

指导教师 刘永红 职称 副教授

企业导师 职称

2016 年 3 月 12 日

基于微信公众平台图书销售系统的设计与实现

专 业：信息科学与工程学院 学 号：1122005

学 生：程 斌 指导教师：刘永红

摘要：

关键词：

Design and Implementation of WeChat Reservation System Based on WeChat Public Platform

Specialty: Computer science and technology Student Number: 1122005

Student: Cheng Bin Supervisor: Liu Yonghong

Abstract: (Times New Roman字体，5号)

Key words: WeChat; PHP; MySQL

绪 论

1. 研究背景和意义

随着微信软件在我们日常生活中的持续走俏，许多商家和品牌已经将营销的战场投向了微信。此时的微信已经不仅仅是一款流行的即时聊天工具，它同时还具有营销价值，微信不仅改变了人们的生活方式，也对移动互联网的发展产生了重大的影响。因此,腾讯官方也根据众多企业和商业用户的需求，推出了微信公众平台这个新的功能模块，通过这一平台，企业和商家可以更方便地在微信平台上进行多种多样的互动活动。当前，众多的企业和商家都希望能够找到一个最好的方式在微信上进行营销推广活动，利用微信平台实现品牌宣传和产品推广的目的。不过，由于微信公众平台提供的基础功能过于简陋，大多数的企业和商家都是简单在微信平台上发布广告宣传信息，没有很好地利用公众平台提供的高级功能设计并进行营销活动，所以很难取得良好的推广效果针对以上情况。

在当前主流移动互联网社交平台中，腾讯公司的微信平台是我国最成功的平台。据统计，微信已在全国范围内拥有6亿以上用户。腾讯公司瞄准电子商务领域，通过微信好友的开放体系，把无数个交际圈、人脉圈，发展成为服务企业的电商平台。截至2013年12月，微信公众账号用户已经超过200万，业务领域涉及到销售、通信、金融、银行、政府、物流、媒体等。亚马逊、京东、苏宁、国美等传统电商巨头纷纷抢滩微信平台，利用微信具有一定用户规模的优势进行移动化尝试，微信正逐渐演变成为一大商业交易平台。

我们需要在充分了解微信公众平台的基础上总结移动互联网环境下网络营销的特点、现状以及微信带来的影响，了解当前微信公共平台的各种功能和提供的第三方接口.以此满足商家和用户在微信平台上营销的基本需求。最终，帮助企业和商家更好地利用微信平台宣传自己的品牌，推广自己的产品，让更多的企业和商家更快地拥抱移动互联网的浪潮。

1. 国内外研究现状

通过对现有资料的收集和整理发现，国外关于微信营销模式的提法较少，从 2012 年微信诞生至今国外也有一少部分的人开始应用微信进行通话、聊天等。现在海外微信比较火的还是在东南亚地区，依照他们的使用情况来看，微信在马来使用率应该是最高的，另外在中东巴基斯坦、泰国、印度、印尼使用人数也较多。与 Line 相比商务人士更倾向于使用微信，特别是与中国商人打交道时。美国目前主要还是中国人在用，但是加拿大数据中心正在架设,未来还是能有一定市场，外来软件进入美国还不是很容易的。

目前国内当下最受关注的移动互联网产品非微信莫属，在各种媒体、媒体人和深度微信用户看来，微信似乎能给现存的绝大多数互联网业务及其商业模式带来冲击，同时还会对几乎所有能与互联网结合的传统企业产生影响。微信，它让很多人感到恐惧，同时它也让很多人看到了“弯道超车”的希望。

1. 发展趋势

传统电子商务的消费模式已经深入人心并有了相当大的受众群体，移动电子商务目前仍然处于起步阶段，在未来还需经历相当的磨合时期。如今微信需要鼓励、刺激消费者对移动电子商务进行深度体验，逐步占领电子商务市场。微信自开启移动电商之路后，一直都努力地把社交流量转化成电商流量，2014年1月微信将“滴滴打车”电商板块植入微信“我的银行卡”，将生活服务进行线上线下的结合，结果是初步改变用户的消费习惯，2014年春节的微信红包笼络了近500万人密切关注微信，凭借微信良好的用户黏性及年轻群体对新产品的热衷度，微信红包一炮而红，结果是微信支付井喷式发展。然而，微信用户的体验并没有在购物上得以加深，微信里的微生活、微购物、微团购模块难以发现甚至发现不了。

在微信移动销售诞生和发展的过程中必须看到，随着微信的商业化趋势，微信已经逐渐把腾讯品牌从原先的社交属性里延伸出来，在不久的将来，微信移动电子商务必然可以和成熟的淘宝一较高下，微信也将通过不同的方法将线上平台和线下商务连成一片，打破线上和线下的障碍，实现“二维码+账号体系+基于位置服务+支付+关系链”一体化，形成一个O2O闭环，并引进优秀的生活服务O2O企业，围绕微信支付打造良好的微信生活服务生态圈，并以此带动其它行业的发展。

1. 系统需求分析和可行性分析

本章主要对系统可行性分析以及系统需求分析进行详细介绍，以便更好的进行可行性论证和需求分析，这为后续系统设计与实现提供良好前期保障。

* 1. 需求分析

需求分析是指对要解决的问题进行详细的分析，弄清楚问题的要求，包括需要输入什么数据，要得到什么结果，最后应输出什么。做好需求分析，能够更好的完成系统的功能设计和实现。

1.1.1 功能需求

1. 用户管理：微信用户的基本信息管理，例如姓名、性别、年龄、手机号、收货地址、QQ号等。
2. 图书浏览：微信用户可通过点击该平台公众号的相应菜单进入购书页面，还可通过向公众账号发送关键词获取回复，根据回复进入系统查看图书列表，进行选择后查看该商家的商品信息，并对其进行选择下单。
3. 订单管理：用户通过该公众平台向服务器提交订单后，后端服务器接收到用户的订单数据保存至数据库,新订书数据同步至服务器端和Android端, 方便商家查看订书订单详情，用户联系方式，便于安排配送。
4. 图书管理：采用MySQL数据库存储图书数据和用户信息。商家通过服务器端,可以对购书订单、图书分类信息和详细信息进行管理并查看。

1.1.2 性能需求

为了保证系统能够长期、稳定、可靠、安全、高效的运行，本系统应满足以下性能需求。

1. 系统处理的准确性和及时性

系统处理的准确性和及时性是任何操作系统的必要性能。在系统设计和开发过程中，要充分考虑到该系统当前和将来可能承受的工作量，使系统的处理能力和响应时间能够满足餐厅对订单信息处理的需求。

本系统的在线下单功能、查询功能是整个系统中的重要功能，在系统开发过程中，必须采用有效方法保证录入数据、查询数据的准确性。

1. 系统响应速度

信息管理管理系统在日常处理中的速度为秒级，要达到实时要求，以及时反馈信息。当然也会存在一定的限制。在进行统计分析时，要根据所需数量的不同而从秒级到分钟级，原则是保证操作人员不会因速度问题而影响工作效率。

* 1. 可行性分析

可行性分析是在用户要求以及系统调研基础上进行的，从新系统的开发以及技术、经济、操作等各方面进行分析。

1.2.1 经济可行性分析：

本设计实现基于微信软件开发，对用户来说，不需要安装任何第三方APP和购买昂贵的设备就能够实现轻松购物；对商家来说，微信官方提供了丰富的编程接口，不需要花费大量的时间和金钱去做开发和设计。因此，本系统在经济上是可行的。

1,2.2 技术可行性论证：

就技术难度而言，本系统采用微信公众平台JS-SDK和ThinkPHP开发，它们拥有丰富的内库、接口和文档，实现容易，开发效率高。本系统还会利用JQuery Mobile和Google提供的第三方AngularJS库，它们集成了大量可复用的模块和工具类，利用这些优势可大大降低开发难度。

1.2.3 操作可行性分析：

该系统设计简单，系统界面友好，符合大多数用户的操作习惯，不需要特别的培训。用户可以根据平时使用微信其它功能的经验来使用本系统。因此，本系统在操作上是可行的。

1.2.4 运行可行性分析：

本系统基于微信公众平台开发，拥有稳定的架构，所耗费资源非常小，只要装有微信的移动端都能满足条件。因此，本系统在运行上是可行的。

1.2.5 可行性研究结论：

通过以上的经济、技术、操作、运行等可行性分析得出一个很重要结论是可以正式研发并可很好的实现其各个功能。以上的研究和分析表明基于微信公众平台图书销售系统开发所产生的效益将大于投入，所以开发该项目是可行的、必要的。

* 1. 开发环境及硬件需求

软件类需求：

1. 服务端操作系统：Linux操作系统（如Ubuntu、openSUSE、CentOS等）
2. 客户端操作系统：Android 4.0+
3. 数据库：MySQL(5.6)
4. 客户端请求服务器：Nginx 1.9
5. 后端开发框架：ThinkPHP 2.3

硬件类需求：

1. 服务端内存：2G以上
2. 客户端运存：512M以上
3. 开发平台及相关技术

2.1 微信公众平台

本节将着重介绍微信公众平台及其功能概述。由于本系统是建立在微信工作平台之上，故将着重介绍其开放接口和调用过程。

2.1.1 微信公众平台介绍

[微信公众平台](http://baike.baidu.com/view/9212662.htm)（WeChat public platform，简称微信平台，微信公众号，公众号）是腾讯公司于2012年08月23日正式推出上线，主要面向名人、政府、媒体、企业等机构推出的合作[推广](http://baike.baidu.com/view/193748.htm)业务。通过这一平台，每一个人都可以用一个 QQ 号码，打造自己的一个微信的公众号，并在微信平台上实现和特定群体的文字、图片、语音的全方位沟通、互动。曾命名为“官号平台”和“媒体平台”，最终定位为“公众平台”，无疑让我们看到一个微信对后续更大的期望。和新浪微博早期从明星战略着手不同，微信此时已经有了亿级的用户，挖掘自己用户的价值，为这个新的平台增加更优质的内容，创造更好的粘性，形成一个不一样的生态循环，是平台发展初期更重要的方向。目前微信公众平台支持PC，并可以绑定私人帐号进行群发信息。

2.1.2 微信公众号类型

2014年9月18日，[微信公众平台](http://baike.baidu.com/view/9212662.htm)升级，将微信公众平台分成订阅号、服务号和企业号三种类型。

订阅号主要偏于为用户传达资讯（类似报纸杂志），认证前后都是每天只可以群发一条消息；服务号主要偏于服务交互（类似银行，114，提供服务查询），认证前后都是每个月可群发4条消息；企业号主要用于公司内部通讯使用，需要先有成员的通讯信息验证才可以关注成功企业号。

2.1.3 微信公众号订阅与发布

通过发布公众号的二维码，让微信用户随手扫描订阅。任何微信公众帐号用户，都可以在设置中找到一个如下格式的二维码，品牌ID会放到二维码的中部。 　　你也可以有其他方式来订阅微信公众帐号。比如，通过微信号进行订阅，在微信上直接点按“添加朋友”——“按号码查找”，输入“帐号(字母，数字，下划数组合且必须以字母开头)”就可以查找并关注您感兴趣的内容。但是对于目前开放申请的微信公众号，大部分微信号格式类似gh\_5a8ce1d1bc8b，并不利于记忆和查找，此外，微信平台上面还可以通过发送名片的方式把你喜欢的微信公众帐号发送给朋友。

2.1.4 微信公众平台接口类型

微信公众平台目前分成消息接口和通用接口两大模块。接入消息接口的微信公众账号，当关注该公众账号的粉丝向其发送消息，微信服务器会对公众账号所对应的服务器地址推送一个特定结构的消息体，公众账号开发者可以通过响应包进行对该条消息的回复。通用接口的作用是实现诸如获取粉丝资料，上传媒体文件以便消息接口回复图片、语音和视频等多媒体消息等功能，丰富消息接口的同时支持主动推送消息给用户。使用接口的任何一个公众账号都拥有唯一的Appid和Appsecret，对于该公众账号都有一个使用等级，使用等级高的将享有更高的接口调用频率。开发者应评估自己的应用，并对公众号等级进行相应的申请。

2.2 使用技术

本节会针对性地介绍与实现微信公众平台图书销售系统中运用到的相关技术及其背景资料。包括采用的PHP开发语言，MySQL数据库，JQuery Mobile开发库。

2.2.1 PHP

PHP是英文Hypertext Preprocessor(超文本预处理器)的缩写。它是一种服务器端的脚本编程语言，其特点是简单、面向对象、解释执行、健壮安全、独立于架构、可移植、动态。自PHP5正式发布以来，PHP以其方便快速的风格、丰富的函数功能和开放的源代码，迅速在 web 系统开发中占有了重要地位，正成为世界上最流行的Web应用程序编程语言之一。PHP 最初于1994 年开始设计。1995 年发布第一版，定名为PHP Tools，提供了留言本、计数器等简单功能。同年第二版 PHP 问市，定名为 PHP/FI ，在其中加入了mSQL 的支持，自此奠定了 PHP 在动态网页开发上的影响力。在 1996 年底，有一万五千个 Web 网站使用PHP/FI，1997 年则增加到超过五万个。1997 年开始了第三版的开发计划，发布时定名为PHP3。PHP3 跟Apache服务器紧密结合，几乎支持所有数据库系统，执行效率极高，这使得在1999年使用 PHP 技术的网站超过了十五万。 2000年5月，PHP4.0正式发布。它使用了Zend引擎，采用了一些新的语言结构，提供更高的性能，支持更多的 Web 服务器，支持 HTTP Sessions，支持输出缓存，处理用户输入更安全，运行速度更快。2004 年 7 月，PHP5 问世。无论对于 PHP 语言本身还是 PHP 的用户来讲，PHP5 发布都算是一个里程碑式的版本。PHP5 的诞生，使 PHP语言进入了一个新时代，给整个Web开发带来新的革命性的发展。在编写动态Web页面时，用户可以混合使用 PHP 和 HTML 。当客户端浏览到这种页面时，服务器端首先对页面中的 PHP 命令进行处理，然后把处理后的结果连同 HTML 内容一起传送到客户端。 PHP 作为一种源代码开放程序，拥有很好的跨平台兼容性。用户可以在诸如 Windows 系统、Unix系统等操作系统上运行 PHP ，而且可以将 PHP 作为 Apache 服务器的内置模块或 CGI 程序运行。

2.2.2 JQuery Mobile

jQuery Mobile是jQuery 在手机上和平板设备上的版本。jQuery Mobile不仅会给主流移动平台带来jQuery核心库，而且会发布一个完整统一的jQuery移动UI框架。jQuery Mobile框架包括构建完整移动Web应用程序和网站所需的所有UI 组件。页面、对话框、工具栏、不同类型的列表视图，各种表单元素和按钮等。jQuery Mobile构建于jQuery内核之上，因此您可以访问关键设备，包括 HTML 和 XML 文档对象模型 (DOM) 的遍历操作；事件处理；使用 Ajax 服务器通信；以及Web页面的动画和图像效果。有了 jQuery Mobile，您就可以轻而易举地编写基础Web应用程序。由于jQuery Mobile是一个非常全面的基础架构，提供了一些高级事件和 API，所以您还可以编写高级移动Web应用程序和网站。

2.2.3 MySQL

MySQL是一个小型关系型数据库管理系统，开发者为瑞典MySQL AB公司。在2008年1月16号被Sun公司收购。而2009年,SUN又被Oracle收购.对于MySQL的前途,没有任何人抱乐观的态度.目前MySQL被广泛地应用在Internet上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库。目前Internet上流行的网站构架方式是LANP（Linux+Nginx +MySQL+PHP），即使用Linux作为操作系统，Nginx作为Web服务器，MySQL作为数据库，PHP作为服务器端脚本解释器。由于这四个软件都是自由或开放源码软件（FLOSS)，因此使用这种方式不用花一分钱就可以建立起一个稳定、免费的网站系统。

2.3.4 Nginx

Nginx("engine x")是一个高性能的HTTP和反向代理 服务器，也是一个IMAP/POP3/SMTP代理服务器。Nginx是由Igor Sysoev为俄罗斯访问量第二的 Rambler.ru 站点开发的，第一个公开版本0.1.0发布于2004年10月4日。其将源代码以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而闻名。2011年6月1日，Nginx1.0.4发布。

与Apache相比，在同样的Web服务下，Nginx占用更少的内存及资源，此外，Nginx处理请求是异步非阻塞的，而Apache是阻塞型的，在高并发的情况下Nginx能保持低资源、低消耗和高性能。据W3Techs在2013年07月公布的数据显示，在排名前1000的高流量网站中，Nginx占据了34.9%，取代了Apache（34.5%）第一名的位置。

第三章 系统概要设计

概要设计也称总体设计，其基本目标是能够针对软件需求分析中提出的一系列软件问题，概要地回答如何解决。例如，软件系统将采用什么样的体系构架、需要创建哪些功能模块、模块之间的关系、使用的数据结构等。

3.1 总体功能设计

该系统分为客户端和服务端：客户端主要是用户对图书的浏览、分类筛选、加入购物车、支付订单；服务端主要是管理员对用户、图书、图书分类、订单信息的管理。在整个系统中它们有着不同的功能和权限。如下图3.1所示：

微信

公众平台图书销售系统

客户端

服务端

用户信息

图书浏览

加入购物车

支付及查看订单

管理员登录

用户管理

图书管理

订单管理

姓名

联系电话

收货地址

邮件地址

图书名称

图片

价格

数量

图书描述

图3.1 总体功能图

3.2 E-R图

微信公众平台客户端主要包含了ID，昵称，姓名，联系电话等属性；图书分类主要包含了分类ID及名称等属性；图书主要包含了图书ID、图书名称和价格等属性；订单主要包含了订单ID和订单状态等属性。系统管理员主要管理用户、收货地址、图书分类、图书和订单五个内容。该系统的E-R如图3.2所示：

用户ID

用户昵称

联系电话

电子邮件

性别

用户

图书类型

类型ID

类型名

类型封面

图书

图书ID

图书名称

图书价格

图书数量

图书封面

销售数量

图书类型

订单

收货地址

地址ID

省份

城市

街道

订单ID

创建时间

总价

用户姓名

收货地址

订单状态

购物车

用户编号

商品编号

数量

价格

用户类型

用户ID

管理员

管理

管理

包含

管理

组成

图书描述

管理

管理

1

m

m

1

1

1

1

1

m

m

m

1

m

图3.2 系统E-R图

3.4 数据库设计

本系统采用的是MySQL数据库，为了将在系统页面中显示的动态数据存放在后台数据库中，便于修改、更新和信息隐藏，因此，将E-R图转换成关系模型，即在数据库中建立实体关系表。

1. 用户表

用户表具有用户昵称、姓名、电话、邮件等基本信息。设置uid为主键，主键的生成策略采用自动生成方式。用户的属性图如图3.3所示：

用户

图3.3 用户属性图

根据图3.3可定义出用户的数据表user，用户表用来保存所有用户的信息。结构如表3-1所示：

表3-1 用户表(user)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| uid | int | 用户ID | primary key |
| username | varchar | 用户名 | not null |
| name | varchar | 姓名 | not null |
| phone | varchar | 联系电话 | not null |
| email | varchar | 电子邮箱 | not null |
| gender | bit | 性别 |  |
| age | tinyint | 年龄 |  |
| create\_time | date | 关注时间 | default current\_timestamp |

1. 收货地址表

收货地址表具有用户编号、省、市、街道、邮编属性。设置id为主键，主键的生成策略采用自动生成方式。地址表的属性图如图3.4所示：

收货地址

图3.4 收货地址属性图

根据图3.4可定义出收货地址的数据表user\_address，用户表用来保存所有收货的信息。结构如表3-2所示：

表3-2 收货地址表(user\_address)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| id | int | 收货地址ID | primary key |
| uid | int | 用户ID | foreign key(uid)  references user(uid) |
| province | varchar | 省 | not null |
| city | varchar | 城市 | not null |
| street | varchar | 街道 | not null |
| zipcode | varchar | 邮编 | not null |
| remark | varchar | 备注 |  |

1. 图书分类表

图书分类表具有分类编号、分类名称、分类封面、描述、备注属性。设置id为主键，主键的生成策略采用自动生成方式。分类表的属性图如图3.5所示：

图书分类表

图3.5 图书分类属性图

根据图3.5可定义出图书分类的数据表book\_category，图书分类表用来保存所有图书分类的信息。结构如表3-3所示：

表3-3 图书分类表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| id | int | 图书分类ID | primary key |
| name | varchar | 分类名称 | not null |
| cover | varchar | 分类图片 | not null |
| description | varchar | 分类描述 |  |
| Remark | varchar | 备注 |  |

1. 图书表

图书表具有图书编号、图书名称、图书封面、图书描述、出版时间、上架时间、发出社、点击数、销售数、原价、现价、作者属性。设置id为主键，主键的生成策略采用自动生成方式。图书表的属性图如图3.6所示：

图书

图3.6 图书属性

根据图3.6可定义出图书的数据表book，图书表用来保存所有图书的所有信息。结构如表3-4所示：

表3-4 图书表(book)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| id | int | 图书ID | primary key |
| book\_category\_id | int | 图书分类ID | foreign key |
| name | varchar | 图书名称 | not null |
| author | varchar | 作者 | not null |
| cover\_id | int | 封面ID | foreign key |
| price\_normal | decimal | 原价 | not null |
| price\_now | decimal | 现价 | not null |
| publish\_time | date | 出版时间 | default current\_timestamp |
| total\_count | int | 数量 | default 0 |
| sell\_count | int | 销售量 | default 0 |
| click\_count | int | 点击数 | default 0 |
| press | varchar | 出版社 | not null |
| description | varchar | 内容描述 | not null |

1. 购物车表

购物车表具有购物车编号、图书编号、用户编号、数量、创建时间、订单编号、备注属性。设置id为主键，主键的生成策略采用自动生成方式。表的属性图如图3.7所示：

购物车

图3.7 购物车属性

根据图3.7可定义出购物车的数据表shop\_cart，购物车表用来保存用户添加欲购买的图书信息。结构如表3-5所示：

表3-5 购物车表(shop\_cart)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| id | int | 购物车ID | primary key |
| book\_id | int | 图书id | foreign key |
| user\_id | int | 用户id | foreign key |
| amount | int | 数量 | default 1 |
| create\_time | date | 创建时间 | default current\_timestamp |
| order\_id | int | 订单id |  |
| remark | varchar | 备注 |  |

1. 订单表

订单表具有订单编号、订单序列号、收件人、收货地址、创建时间、总价、订单状态、联系电话、用户编号、备注属性。设置id为主键，主键的生成策略采用自动生成方式。表的属性图如图3.8所示：

订单

图3.8 订单属性

根据图3.8可定义出订单数据表order，订单表用来保存用户购买的图书信息。结构如表3-6所示：

图3-6 订单表(order)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| id | int | 订单ID | primary key |
| order\_id | varchar | 订单序列号 | not null |
| username | varchar | 收件人名字 | not null |
| address | varchar | 收件人地址 | not null |
| create\_time | date | 创建时间 | default current\_timestamp |
| total\_price | decimal | 总价 | not null |
| order\_status | bit | 订单状态 | not null |
| phone | varchar | 联系电话 | not null |
| user\_id | int | 用户ID | foreign key |
| remark | varchar | 备注 |  |

1. 管理员表

管理员表具有管理员ID、账号、密码、备注属性。设置id为主键，主键的生成策略采用自动生成方式。表的属性图如图3.9所示：

管理员

图3-9 管理员表

根据图3.9可定义出管理员数据表manager，管理员表用来保存后台用户的登录信息。结构如表3-7所示：

表3-7 管理员表(manager)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| id | int | 管理员ID | primary key |
| account | varchar | 管理员号 | not null |
| password | varchar | 管理员密码 | foreign key |
| remark | varchar | 备注 |  |

1. 附件表

附件表具有附件ID、附件路径、实体编号属性。设置id为主键，主键的生成策略采用自动生成方式。表的属性图如图3.10所示：

附件

图3-10 附件表

根据图3.10可定义出附件数据表attachment，附件表用来保存用户上传的图片、文件等信息。结构如表3-8所示：

表3-8 附件表(attachment)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| id | int | 附件ID | primary key |
| path | varchar | 存放路径 | not null |
| entity\_id | int | 实体编号 | foreign key |

1. 操作日志表

操作日志表具有日志ID、操作类型、操作用户、操作时间属性。设置id为主键，主键的生成策略采用自动生成方式。表的属性图如图3.11所示：

操作日志表

图3-11 操作日志表

根据图3.11可定义出操作日志数据表log，操作日志表用来记录用户或管理员对系统进行的操作信息。结构如表3-9所示：

表3-9 操作日志表(log)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | 说明 |
| id | int | 操作ID | primary key |
| do | varchar | 操作动作 | not null |
| create\_author | int | 用户编号 | foreign key |
| create\_time | date | 操作时间 | default current\_timestamp |

第四章 详细设计

4.1 总体工作流程设计

登入微信平台图书销售系统

登入服务端

用户信息

浏览图书

购买图书

我的订单

用户信息

图书信息

图书分类信息

订单信息

数据库（MySQL）

图4.1 系统工作流程图

工作流程如图4.1所示：用户只需要关注该微信图书销售系统，无需账号注册，便可进行图书浏览、加入购物车、购买、支付等。管理员登录后台，对图书及图书分类进行录入，系统将录入的数据存储到数据库中，管理员可根据订单信息选择是否发货等功能。

4.2 系统逻辑结构设计

4.2.1 客户端逻辑结构设计

1. 商品分类搜索，如图4.2所示：

点击分类Item

将分类请求发送至服务端

服务端查询数据库，返回分类数据至客户端

用户在客户端显示图书列表

图4.2 客户端图书分类搜索流程图

图书分类主要按用户想购买的类型来搜索，在以上图书搜索中，选择一个类型可实现多重查询，显示多本书籍在界面。

1. 商品搜索，如图4.3所示：

用户输入图书名称

客户端将输入的数据用字节流发送至服务端

服务端处理逻辑，查询数据库，返回图书列表至客户端

用户在客户端选择具体图书

显示图书详细信息

选择数量，加入购物车或购买

图4.3 客户端图书搜索流程图

客户端将用户填写的书籍名称发送至服务端查询，服务端根据图书名称对其进行模糊查询出多本书籍数据，并将数据返回至客户端。客户端可对图书进行详细信息查看、加入购物车和购买。

1. 购物车，如图4.4所示：

点击购物车

客户端本地请求，读取本地文件并构造成Model

将构造的书籍列表数据显示在客户端

点击结算以及支付，将数据序列化发送至服务端

服务端读取数据，调用支付接口，存储数据至库

服务端返回订单支付结果

图4.4 客户端购物车流程图

用户点击购物车，客户端读取本地图书数据并显示到界面，点击结算按钮，生成订单及图书详细信息，输入数量、金额后，提交数据至服务端，服务端跳转至支付界面，支付处理后返回订单状态给客户端。

1. 订单，如图4.5所示：

用户点击订单菜单

将订单请求发送至服务端

服务器响应请求，将订单数据列表发送至客户端

将订单列表在客户端显示

图4.5 客户端订单列表流程图

点击结算按钮后，客服端发送请求服务端，将用户填写的订单信息添加至数据库。通过后台来实现对订单的管理。管理员在确定发货后，数据库将订单的状态更改为已发货。客户端页面中，返回数据库中订单状态。

4.2.2 服务端逻辑结构

1. 管理员登录，如图4.6所示：

管理员登录

向服务器发送数据，检验用户信息

返回至登录界面

登录成功

True

False

图4.6 管理员登录流程图

管理员在后台进行登登录，输入的账号密码，将发送到数据库进行验证，如果正确，将直接登入到主界面中，如果错误，对其进行错误提示并返回至登录界面重新登入。

1. 图书列表，如图4.7所示：

图书列表

向数据库发送查询请求，将数据显示到页面

添加图书

图4.7图书管理流程图

修改图书

删除图书

查看图书

在管理员点击图书列表菜单的时候，数据库将所有图书信息分页显示在页面中，管理员可对图书进行增加，修改，删除，查询的功能。

1. 订单列表，如图4.8所示：

订单列表

向数据库发送查询请求，将数据显示到页面

设定为发货

图4.8订单管理流程图

取消订单

管理员点击订单列表，页面使用分页显示出所有客户的订单，管理员可以对这些订单进行操作，实现订单状态的修改，发货或关闭交易等操作。

1. 用户列表，如图4.9所示：

用户列表

向数据库发送请求，将用户信息显示在界面

强制取消关注

图4.9 用户管理流程图

删除用户

查看用户信息

管理员点击用户菜单，数据库将用户信息分页且显示在页面中，管理员可对某个用户进行强制取消关注、删除、查看详细信息等功能。以此实现对系统更好的管理。

第五章 系统实现

5.1 微信平台主要功能实现

5.1.1图书分类模块

图书分类模块分为两个板块，即包含分类列表和显示系统最新书籍的banner栏目。图书分类界面如图5.1所示：

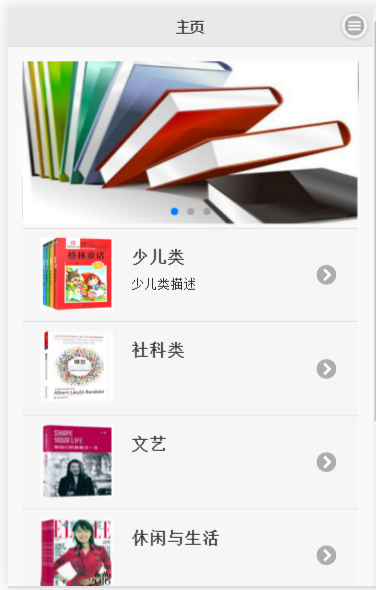


图5.1 图书分类界面

图书分类模块展示了图书销售系统的所有图书分类和最新上架书籍，同时也是系统的主页，设计的目的是为了让用户可以方便购买分类书籍。

5.1.2图书模块

图书模块由图书列表和图书详细信息组成。图书列表类似于分类列表，图书的详细信息界面如图5.2所示：



图5.2 图书分类界面

当公众平台用户点击某图书项时，客户端请求服务端的getBookById()方法来获取图书的基本信息并将结果渲染到界面。getBookById()方法核心代码如下：

public function getBookById() {

...

$params['id'] = $\_POST['bookId'];

$dao = new BookDao();

$bookInfo = $dao->findBookById($params);

...

}

5.1.3购物车模块

购物车模块中，主要包含以下几个功能，即：添加购物车、查看购物车、生成订单。购物车的界面如图5.3所示：



图5.3 购物车界面

添加购物车就是把用户选中的书籍暂时存放在购物车中，当用户在首页中单击图书展示区的“购买”按钮时，系统会将该商品的详细信息展示在查看物品清单页面中。用户在单击物品清单页面下方的“加入购物车”链接，便可以将该图书放入购物车中。

添加图书的代码流程是从session中读取书籍对象，如果为空则说明还没有进行购物或者已经清空了购物车，需要新建购物车对象；将图书名称与购物车列表中的图书名称对比，如果已经存在，则把商品数量加1。

为了方便用户随时查看购物情况，在网站的首页加入了查看购物车菜单，通过它可以将用户所有选中的书籍信息放入购物车中显示出来。在程序中使用了一组文本框记录用户购买的图书数量，用户可以在文本框中输入想要购买的数量。如果欲删除该商品，可以在对应的图书项里点击删除按钮。系统每次只会将1个商品放入购物车中，如果用户需要多个同种书籍，可以通过修改书籍信息右侧相应文本框的值来完成。

5.1.4订单模块

订单模块中，主要包含以下几个功能，即：收货地址的填写、地址的添加、提交订单、查看订单详细信息及状态。订单模块的界面如图5.4和5.5所示：



图5.4 提交订单界面

生成订单是购买书籍的最终目的，前面所有功能的实现都是为最后生成一个用户满意的订单作基础，在此要生成一个可供用户随时查询的订单号，还要保存用户订单中所购买的图书信息。当用户确认对购物车不再改变以后，就可以点击提交订单。提交订单的流程为：从购物车中读取书籍名称、数量、价格信息，生成一个唯一的订单号，同时也把用户注册的基本信息读取出来，形成一个完整的订单写入数据库。

当用户点击提交按钮后，客户端会向服务端发送支付请求。服务端调用支付接口的jsapipay()方法进行支付。支付接口方法jsapipay()的核心代码如下：

public function jsapipay($order = array()) {

...

//获取用户openid

$tools = new JsApiPay();

$openId = $tools->GetOpenid();

//下单

$input = new WxPayUnifiedOrder();

$input->SetBody($order['names']);

$input->SetAttach($order['names']);

$input->SetOut\_trade\_no($order['trade\_no']);

$total\_fee = $order['total\_fee'];

$input->SetTotal\_fee($total\_fee);

$input->SetTime\_start(date("YmdHis"));

$input->SetTime\_expire(date("YmdHis", time() + 600));

$input->SetGoods\_tag($order['names']);

$input->SetNotify\_url("/Home/Main/success.html");

$input->SetTrade\_type("JSAPI");

$input->SetOpenid($openId);

$order = WxPayApi::unifiedOrder($input);

$jsApiParameters = $tools->GetJsApiParameters($order);

...

}



图5.5 订单信息界面

用户提交订单后，通过产生的订单号查询订单信息及执行状态。用户在购物中心首页中单击书籍的订单链接，就可以数据表中查询处对应的书籍信息并转到订单查询显示页面将书籍信息显示出来。

5.2 服务端主要功能实现

5.2.1登录模块

设计登录模块是为了防止对数据库服务端的恶意操作。管理员登录界面如图5.6所示：

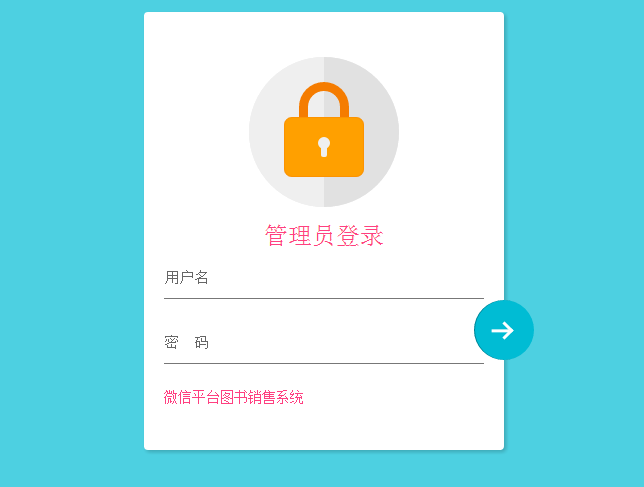


图5.6 管理员登录界面

登录功能是对管理员身份进行验证，在管理登录页面中，是通过JavaScript的validate()方法来实现录入数据完整性验证的。用户输入登录ID和密码后单击“登录”按钮时，如果录入数据完整性验证通过，系统会将数据提交到服务端检验输入信息在数据库manager表中是否存在提交的用户名和密码，如果登录用户名及密码有效，就保存用户名信息，管理员进入主页，否则提示错误信息。Validata()方法的核心代码如下：

function validate() {

var username = $("#username").val();

var pwd = $("#password").val();

if(username === "" || pwd === "") {

return false;

}

return true;

}

5.2.2用户管理模块

用户管理模块主要包括对用户的删除和取消关注，用户管理模块如图5.7所示：



图5.7 用户管理界面

当管理员点击用户管理菜单时，服务端调用findList()方法来获取用户的分页列表并将结果渲染到界面。findList()方法核心代码如下：

public function book() {

...

$u = new User();

$userList = $u->findList();

$this->assign('users', $userList);

$this->display();

...

}

5.2.3图书分类管理模块

图书分类管理模块，主要包括：添加图书类型、修改图书类型、删除图书类型。其添加界面如图5.8所示，类型列表类似于图书列表，如5.10所示：

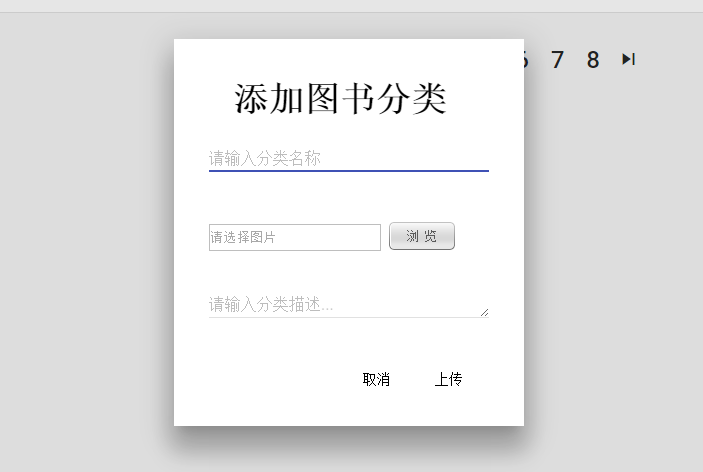


图5.8 图书分类界面

单击图书类型界面中的“添加类型”超链接，系统将打开添加图书页面，该页面主要用于将图书信息添加到数据库中，以提供微信客户端展示所需要的数据依据。为了减少人为错误，在提交数据信息时，需要对数据的合法性进行验证，输入的分类名称不能为空，上传只能是图片且大小小于3M等，这些任务由ThinkPHP的upload()方法来完成。upload()方法核心代码如下：

public function upload() {

...

$upload = new \Think\Upload();

$upload->maxSize = 3292200;

$upload->allowExts = explode(',', 'jpg,gif,png,jpeg');

$upload->savePath = './categorys/';

$info = $upload->upload();

if (!$info) {

$this->error($upload->getError());

} else {

$filename = $info[upload]['savename'];

}

...

}

5.2.4图书管理模块

图书管理模块包含对图书的添加、删除、修改功能。添加图书的界面类似于图书类型的添加，效果如图5.8所示，图书列表界面如图5.9所示：



图5.9 图书管理界面

当用户点击了图书管理菜单时，服务端请求查询数据库，数据库返回查询结果，并使用模板技术渲染页面。页面展示了图书的基本信息和对图书的操作按钮。

5.2.5 订单管理模块

订单管理模块，主要包括：订单状态设置、取消订单、查看详细订单功能。界面效果如图5.10所示：



图5.10 订单界面

用户在微信平台端所产生的订单并不能立即执行，需要系统管理员在后台进行订单信息管理审核以确认它的执行性。用户单击菜单中的订单管理链接进入用户订单处理页面，该页面上会显示近期处理的订单。用户订单处理页面显示的是用户和订单基本信息，如果想知道订单中所涉及的书籍，需要单击对应订单号的超链接打开查看详情页面才行。

第六章 系统测试与分析

经过本文三四五章介绍，微信公众平台图书销售系统的设计和开发过程已经完成。本章将对微信公众平台图书销售系统进行测试，包括功能测试、压力测试和兼容性测试。

6.1 测试目的

系统测试是为了发现错误而执行程序的过程，成功的测试是发现了至今尚未发现的错误的测试。

测试的目的就是希望能以最少的人力和时间发现潜在的各种错误和缺陷。应根据开发各阶段的需求、设计等文档或程序的内部结构精心设计测试用例，并利用这些实例来运行程序，以便发现错误。信息系统测试应包括软件测试、硬件测试和网络测试。硬件测试、网络测试可以根据具体的性能指标来进行，此处所说的测试更多的是指软件测试。

系统测试是保证系统质量和可靠性的关键步骤，是对系统开发过程中的系统分析系统设计和实施的最后复查。根据测试的概念和目的，在进行信息系统测试时应遵循以基本原则。

6.2 测试环境

6.2.1 服务端

将系统部署到Coding服务器，服务端具体环境如表6-1所示。

表6-1服务端测试环境

|  |  |  |
| --- | --- | --- |
| 软件 | 服务器操作系统 | Linux |
| 本地WEB服务器 | Nginx |
| PHP版本 | PHP5 |
| ThinkPHP版本 | ThinkPHP 2.3 |
| 数据库 | MySQL 5.6 |
| 硬件 | CPU | 2G |
| 内存 | 1G |

6.2.2 客户端

测试时，采用不同平台使用本系统，这样可以测试系统对于不同平台的兼容性。使用的网络为移动通信3G/4G网络以及WLAN无线网。测试时选取的移动智能终端信息如表6-2所示。

表6-2 测试所使用的移动智能终端

|  |  |  |  |
| --- | --- | --- | --- |
| 手机名字 | 尺寸 | 分辨率 | 操作系统 |
| 小米5 | 5.7英寸 | 1920×1080像素 | Android 6.0 |
| 红米Note3 | 5.5英寸 | 1920×1080像素 | Android 5.x |
| OPPO R821T | 4.0英寸 | 480×800像素 | Android 4.0 |
| iPhone 6s plus | 5.5英寸 | 1920 x 1080像素 | IOS |

6.3 测试内容

6.3.1功能测试

功能测试主要内容包括：是否正常稳定启动本应用，各功能模块是否能正常运行、是否能够正常链接数据库并且准确读取数据库信息测试，系统获取数据的准确性，是否和原系统获取数据一致。

1. 关注该微信平台测试

基于微信公众平台的图书销售系统建立了用户数据库，用户只需要关注该公众号即可进行购买书籍。客户端读取该用户的数据保存在服务端用户数据表中，当用户取消关注该公众号后用户表数据不会删除，方便用户日后再次关注。关注公众平台的测试步骤如表6-3所示。

表6-3 关注公众平台测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 输入数据 | 预期输出 | 实际输出 |
| 1 | 关注该公众号 | 添加或扫描二维码 | 关注成功，推送一条新用户欢迎信息 | 关注成功，接收到欢迎消息 |
| 2 | 取消关注 | 选择取消关注 | 取消关注成功，从用户服务号列表删除 | 取消关注成功 |
| 3 | 再次关注 | 添加或扫描二维码 | 关注成功，推送一条欢迎再次关注信息 | 关注成功，接收到欢迎消息 |

1. 购买图书测试

用户选择某类图书进入图书选择页面，选择好欲购买图书后，需要选择图书数量，图书的数量不能小于1，也不能大于库存总数，根据购买数量的改变价格也会跟着改变。加入购物车后，购物车能够查询到选购的数据。购买图书的测试步骤如表6-4所示：

表6-4 购买图书测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 输入 | 预期输出 | 实际输出 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

1. 添加收货地址测试
2. 登录测试
3. 添加图书测试
4. 订单测试

6.3.2 压力测试

压力测试主要目的是测试软件在可能的最大负载的情况下系统的运行效果，以发现在最大负载的条件下应用程序可能存在的缺陷。压力测试的内容就是测试在用户数量及系统并发量较大的情况下系统的运行情况。压力测试往往需要组织大量的用户进行测试。由于实际中要组织大量人员来测试不方便，所以在系统测试过程中采用的是模拟压力测试方式，即采用压力测试软件来模拟操作。

使用ApacheBench软件对系统模拟了50人左右的压力测试（本地测试）。压力测试结果如表6-3所示，系统基本能够承受50人左右的并发访问。但在并发数达到55左右时，服务器的响应速度已经在5s左右，超出了微信公众平台接收响应的时间。

表6-3 压力测试结果

|  |  |  |
| --- | --- | --- |
| 并发数 | 用户登录响应时间 | 信息查询响应时间 |
| 1 | 0.5s | 0.5s |
| 5 | 1s | 0.9s |
| 10 | 1.8s | 2s |
| 25 | 2.6s | 2.8s |
| 50 | 4.3s | 4.9s |

6.3.3 兼容性测试

兼容性测试包括两方面，一是系统对于移动平台的兼容性测试。系统需要兼容Android、IOS两大主流平台，因此需要测试在不同的平台下各功能是否正常运行，效果是否一样。二是系统与移动智能终端上的其他软件的兼容性测试。测试本系统是否能与移动智能终端上的其他软件同时使用。

在测试中，客户端采用了型号、操作系统、屏幕尺寸、分辨率等不同的移动智能终端进行测试，结果表明本系统都可兼容，运行稳定、流畅。和其他应用同时使用时也可以兼容。

# 结束语

通过本次微信公众平台的图书销售系统的设计和论文的撰写对自身有了很大的提高，对于四年所学的专业知识进行再次的巩固和不断的实践。系统中所运用到的这些知识是我在平时学习理论课时所学不到的。虽然自己在平时利用一些时间去看书，但还是感觉到自己所学到的东西太少，感谢学院老师给了我们一个动手实践的机会，同时在指导老师的耐心指导之下，使自己能够把所学到的理论知识与实践结合起来。在设计中，发现了自己存在了很多不足，在指导老师的帮助下，我能够认清自己的不足，并能很好的解决这些问题。

本系统处于试运行阶段，还存在许多问题与不足,如安全性能还不完善，缺少网上支付认证加密系统等。因此在今后的工作里，这些功能与技术还需要加以提高和实现。微信平台系统无论是在设计过程中，还是建成后的日常维护过程中，都需要进行严格的检测，以保证购物系统的服务质量。

随着网上购物系统技术的日趋成熟、网上购物的优势日趋明显，会有更多的人参与到网上购物的浪潮中来。在今后的日子里，我会进一步加强自我实践能力，并在以后工作岗位中发挥我所长。