**致生活积分购物APP设计与实现**

**摘要**

随着计算机技术在各行各业日益广泛和深入的应用，网上的概念早已深入人心。网上在各行各业的发展战略中占据了重要的位置，成为商家不可分割的部分。商品的宣传已不只局限于电视和报纸，网上已成为商家展示自己的另一个舞台。商家建立网站，将商家各方面的宣传与服务展现于网中，通过网上实现网上购物、信息查询等功能，这些在改变我们原有经营方式与经营理念的同时，也为商家带来了更高的效益。因此，对商家来说，拥有一个属于自己的网站是至关重要的。网上商城实际上是运行在web服务器中的一个web运用程序。网上商店模拟一般的商店的经营模式。利用页面、脚本程序来实现网上商城的进货管理、销售管理、库存管理。互联网技术提供的不仅仅只是供需双方间的较低的交易成本，还有较低的选择费用和更多可供选择的商品，这些特点促使商家更多地通过使用网络来进行买卖交易。

本系统主要实现了购物系统中的积分购物子系统，积分兑换作为一种可控制的增加用户粘性手段，现今大多数商家和app都会建立一套这样的系统。因此也在各种购物软件中流行起来。积分兑换的发展归结原因在于现代社会关于促销、关于增强会员粘性、加强用户体验等各类商业手段的创新不断发展。

本系统前台开发使用html , css ,javascript等技术，后台使用java+servlet技术，通过Ajax+ JSON与后台进行交互，开源的关系型数据库Mysql用来保存数据。

关键词：accumulate points , product, App

**Abstract**

With the wide application of computer technology in all walks of life,The concept of the Internet has long been popular.Online in all walks of life development

strategy occupies an important position, become an integral part of the business.The promotion of goods has not only limited to television and newspapers, the Internet has become a stage to show their businesses.Businesses to establish a website, the business of all aspects of publicity and services in the network, through online shopping, information inquiries and other functions,hese changes in our original mode of operation and business philosophy, but also for businesses to bring higher benefits.Therefore, it is very important for businesses to have a website of their own.Online mall is actually running in the web server in a web application.Online store simulation of the general store business model.The use of the page, the script to achieve online shopping mall management, sales management, inventory management.Internet technology is not only the supply and demand side of the lower transaction costs, as well as lower selection costs and more options,These features encourage businesses to use the network to buy and sell transactions.

This system mainly realizes the integral shopping subsystem in the shopping system,Integral convertibility as a way to control the increase in user stickiness,Today,

most businesses and app will establish a system.it is also popular in a variety of shopping software.The reason for the development of integral exchange lies in the continuous development of modern society, such as promotion, on the strengthening of member stickiness, enhance the user experience and other commercial means of innovation.

The front of the system development using HTML, CSS, JavaScript and other technologies,background using JAVA technology,Interact with the background by Ajax+ JSON,open source relational database Mysql is used to save data.

**Key words**：accumulate points , product, App

目录

[第1章 绪论 1](#_Toc483237924)

[1.1 开发背景 1](#_Toc483237925)

[1.2 系统目标 1](#_Toc483237926)

[1.3 设计意义 1](#_Toc483237927)

[第2章相关技术介绍 3](#_Toc483237928)

[2.1 开发平台简介 3](#_Toc483237929)

[2.2 JSON简介 3](#_Toc483237930)

[2.3 Ajax技术简介 3](#_Toc483237931)

[2.4 Servlet技术简介 4](#_Toc483237932)

[2.5 Javascript技术简介 4](#_Toc483237933)

[第3章需求分析 6](#_Toc483237934)

[3.1 功能分析 6](#_Toc483237938)

[3.1.1系统前台 6](#_Toc483237942)

[3.1.2 系统后台 7](#_Toc483237943)

[3.2 各模块功能分析 8](#_Toc483237946)

[3.2.1 用户管理模块分析 8](#_Toc483237947)

[3.2.2 商品管理模块分析 9](#_Toc483237948)

[3.2.3 订单管理模块分析 9](#_Toc483237949)

[3.3 开发及运行环境 10](#_Toc483237950)

[3.4 可行性分析 10](#_Toc483237951)

[第4章系统总体设计 11](#_Toc483237952)

[4.1 体系结构 11](#_Toc483237956)

[4.2 系统功能结构 12](#_Toc483237957)

[4.3 概念结构设计 13](#_Toc483237963)

[4.4 数据库逻辑结构设计 16](#_Toc483237970)

[4.5 物理结构设计 18](#_Toc483237971)

[4.6 数据表之间的联系 19](#_Toc483237972)

[第5章系统功能模块详细设计与实现 21](#_Toc483237973)

[5.1 用户管理模块设计 21](#_Toc483237978)

[5.1.1登录界面及个人中心界面设计 21](#_Toc483237979)

[5.2 商品管理模块设计 24](#_Toc483237982)

[5.2.1加入购物车流程设计 24](#_Toc483237983)

[5.2.2购物车界面及商品分类界面设计 26](#_Toc483237984)

[5.3 订单管理模块设计 26](#_Toc483237990)

[5.3.1订单流程设计 26](#_Toc483237991)

[5.3.2订单流程界面设计 28](#_Toc483237992)

[第6章系统软件测试 30](#_Toc483237993)

[6. 1测试目的 30](#_Toc483238000)

[6.2测试方法 31](#_Toc483238001)

[6.3测试用例 31](#_Toc483238002)

[6.3.1测试用户订单提交的测试用例 31](#_Toc483238003)

[6.3.2测试商品添加的测试用例 32](#_Toc483238004)

[总结 33](#_Toc483238005)

[致谢 34](#_Toc483238006)

[参考文献 35](#_Toc483238007)

[附录 36](#_Toc483238008)

# 绪论

## 开发背景

随着互联网的不断发展，各大购物软件层出不穷，各大企业开始思考，如何让消费者，在完全消费条件和选择条件下，更倾向自己的产品，积分商城应运而生，积分商城的出现提高了用户粘度，入口流量以及用户体验等等。因此，大多数商家和App都会建立一套这样的体系。

宏观上，积分商城平台的建设将基于移动和商家现有的用户资源、网络资源，为企业更大规模的发展提供动力。不断完善了客户服务体系，尊重了用户个性化的需求，而且为短期的营销活动提供了更富有想象力的操作平台，并未商家的进一步发展开拓了新的应用方向。

微观上，积分平台的推出，将进一步提高用户积分兑换和品牌宣传的积极性，降低了实体门店操作的成本，彻底摆脱了时间和空间上对业务的限制，进一步提升了积分作为品牌凝聚剂的威力。

同时，积分商城的建设，还将为未来在竞争中发展用户、进一步扩大收入提供强大支持。[1]

## 系统目标

在早期的积分商城中，存在各种问题，如提供给用户的兑换商品品种远不能满足用户的个性需求，物种品类少，造成用户对积分不敏感；限于人员与合作的局限性，往往受用户欢迎的积分兑换商品不能长期的、持续的进行，只能搞运动式互动，长期以往，客户满意度下降，由于积分兑换的特殊性，基本只能在营业厅发放，客户只能跑到指定地点进行领取，个用户体验带来了极大的伤害。

因此，在系统开发中为解决上述问题，商城中包含商品分类较多，并且包含商品邮递，让用户在家就能轻松获得兑换的商品，以及加入限时抢购玩法，吸引客户在特定时间内登陆App.

## 设计意义

系统针对已有积分商城存在问题进行改进，同时对所学知识，进行巩固提升。系统开发平台为eclipse,前台用html5 , css , javascript 等技术进行编写，以及运用插件等；后台采用java +servlet。架构设计上，前台和后台交互数据，交互形式为Ajax+json，数据的存储以 mysql 为数据库，部署环境为Tomcat7.0。积分商城是一个成熟App的衍生品，因此无论是吸引流量，增加用户粘度，还是销售额度都要从App主体出发，本系统设计旨在完善积分商城，让积分体系化。

# 第2章相关技术介绍

## 2.1 开发平台简介

eclipse是著名的跨平台的自由集成开发环境（IDE）。最初主要用来java语言开发，通过安装不同的插件eclipse可以支持不同的计算机语言，比如C++和Python等开发工具。eclipse的本身只是一个框架平台，但是众多插件的支持使得eclipse拥有其他功能相对固定的IDE软件很难具有的灵活性。许多软件开发商以eclipse为框架开发自己的IDE。[2]

## 2.2 JSON简介

JSON（JavaScript Object Notation）是一种轻量级的数据交换格式，它基于ECMAScript的一个子集。JSON采用完全独立于语言的文本格式，但是也使用了类似于C语言家族的习惯（包括[C](http://baike.baidu.com/subview/10075/6770152.htm" \t "_blank)、C++、[C#](http://baike.baidu.com/view/6590.htm" \t "_blank)、[Java](http://baike.baidu.com/subview/29/12654100.htm" \t "_blank)、JavaScript、[Perl](http://baike.baidu.com/view/46614.htm" \t "_blank)、[Python](http://baike.baidu.com/view/21087.htm" \t "_blank)等）。这些特性使JSON成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成(一般用于提升网络传输速率)。

JSON 可以将 JavaScript 对象中表示的一组数据转换为字符串，然后就可以在函数之间轻松地传递这个字符串，或者在异步应用程序中将字符串从 Web 客户机传递给服务器端程序。这个字符串看起来有点儿古怪，但是[JavaScript](http://baike.baidu.com/view/16168.htm" \t "_blank)很容易解释它，而且 JSON 可以表示比"名称 / 值对"更复杂的结构。例如，可以表示[数组](http://baike.baidu.com/view/209670.htm" \t "_blank)和复杂的对象，而不仅仅是键和值的简单列表。[3]

## 2.3 Ajax技术简介

Ajax是异步的javascript和xml，通过后台与服务器进行少量数据交换，Ajax可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页进行部分更新。

Ajax不是一种新的编程语言，而是一种用于创建更好更快以及交互性更强的web应用程序的技术。使用javascript向服务器提出请求并处理响应而不阻塞用户，核心对象XMLHTTPRequest。通过这个对象，您的javascript可在不重载页面的情况与Web服务器交互数据。

Ajax前景非常乐观，可以提高系统性能，优化用户界面。Ajax现有直接框架AjaxPro，可以引入AjaxPro.2.dll文件，可以直接在前台页面JS调用后台页面的方法。[4]

## 2.4 Servlet技术简介

Servlet全称Java Servlet，是用java编写的服务器端程序。其主要功能在于交互式地浏览和修改数据，生成动态的Web内容。狭义的Servlet是指java语言实现的一个接口。

Servlet运行于支持java的应用服务器中，从原理上讲，servlet可以响应任何类型的请求，但大多数情况下Servlet只用来扩展基于HTTP协议的Web服务器。[5]

加载和实例化Servlet。这项操作一般是动态执行的。然而，Server通常会提供一个管理的选项，用于在Server启动时强制装载和初始化特定的Servlet，Server创建一个Servlet的实例，第一个客户端的请求到达Server，Server调用 Servlet的init()方法（可配置为Server创建Servlet实例时调用，在web.xml中 <servlet>标签下配置<load-on-startup>标签，配置的值为整型，值越小 Servlet 的启动优先级越高），一个客户端的请求到达Server，Server创建一个请求对象，处理客户端请求，Server创建一个响应对象，响应客户端请求，Server激活 Servlet的service()方法，传递请求和响应对象作为参数，service() 方法获得关于请求对象的信息，处理请求，访问其他资源，获得需要的信息，service() 方法使用响应对象的方法，将响应传回Server，最终到达客户端。service()方法可能激活其它方法以处理请求，如 doGet() 或 doPost() 或程序员自己开发的新的方法。[6]

对于更多的客户端请求，Server创建新的请求和响应对象，仍然激活Servlet的service()方法，将这两个对象作为参数传递给它。如此重复以上的循环，但无需再次调用init()方法。一般 Servlet 只初始化一次(只有一个对象)，当 Server 不再需要Servlet时（一般当Server关闭时），Server调用Servlet的destroy()方法。[7]

## 2.5 Javascript技术简介

javascript是一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为JavaScript引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在HTML（标准通用标记语言下的一个应用）网页上使用，用来给HTML网页增加动态功能。[8]

是一种属于网络的脚本语言,已经被广泛用于Web应用开发,常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。通常JavaScript脚本是通过嵌入在HTML中来实现自身的功能的。

Javascript脚本语言同其他语言一样，有它自身的基本数据类型，表达式和算术运算符及程序的基本程序框架。Javascript提供了四种基本的数据类型和两种特殊数据类型用来处理数据和文字。而变量提供存放信息的地方，表达式则可以完成较复杂的信息处理。[9]

# 第3章需求分析



## 3.1 功能分析

功能需求，主要对系统功能进行分析和整理，罗列系统整体的各个功能点，然后通过各个功能点的用例来详细分析各个需求。最后配置各个功能，设计出功能在哪里实现。[10]



### 3.1.1系统前台

系统为积分兑换App主要包括以下模块需求：

1. 用户在未登陆App时无法进入软件，进入积分商城的前提必须进行登陆；
2. 用户在积分商城中无法注册个人信息，即账户密码捆绑在主体App上；
3. 每日打卡，可增加相应的积分；
4. 用户登录进入App后可以根据账户积分进行积分兑换商品，超出积分余额无法兑换；
5. 用户进行购物的入口主要三个模块，即主页，分类页，以及抢购页。
6. 购物车模块展示用户购物时添加在购物车的所有商品信息，包含描述信息，规格，数量不限。
7. 订单模块主要展示用户付款后的商品，包含付款积分，以及图片信息描述和收货状态。
8. 订单模块中点击任意一商品可进入订单详情模块，展示商品详细信息，快递详情，收货地址信息等。
9. 收货地址模块主要包含我的收货地址，新建收货地址，编辑收货地址，选择收货地址等，根据用户操作，存储相应字段到数据库中。
10. 个人中心模块可以查看用户积分，订单信息，以及我的收货地址。

系统前台用例图如下：

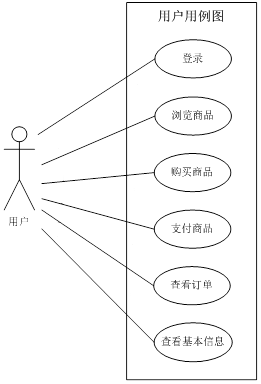


图 3-1 系统前台用户用例图

### 3.1.2 系统后台

系统后台根据功能分为三个模块，即用户管理模块，商品管理模块，以及订单管理模块。系统后台用例图如下：

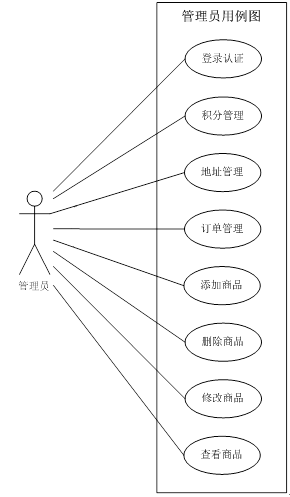


图 3-2 系统后台管理员用例图



## 3.2 各模块功能分析

### 3.2.1 用户管理模块分析

用户管理模块包括5个主要功能：

1）注册，用户注册账号，创建用户基本信息。

2）认证，即登录和调用其他接口的统一token令牌的获取功能

3）积分管理功能，积分的增删改查功能，是提供给业务逻辑调用的底层功能，比如签到接口调用增加积分，购买接口调用扣除积分，用户基本信息调用查询积分等。

4）地址信息管理，用户收货地址的管理功能，编辑地址，设置默认地址等。

5）基本信息管理，用户可以对基本信息进行管理，修改电话号码，用户昵称等。

用户管理

注册帐号

登录认证

积分管理

地址管理

基本信息

信息

图 3-3 用户管理功能图

### 3.2.2 商品管理模块分析

商品管理模块只是对商品的增删改查，但是对于商品来说，本身就比较复杂，一个商品除了名称、价格等固定的属性之外，还有像颜色、大小、重量、形状等等的不固定的属性。这些属性对于不同的商品不固定数量和类型。而且有时候出于营销的需要，希望对商品进行分类，有些商品要限时促销，有些要打折，这种情况下还要对商品进行多个维度的分类。

出于以上两点考虑，对于商品管理，一件商品的新增，要配置商品的基本属性和<K,V>形式的规格属性，在商品被购买时生成商品实例的时候，商品的规格才最终确定了，这时候也有了对应的规格实例，这样就可以解决不同商品的不同数量和类别的规格的问题。还要对商品添加标签来进行多维度分类，满足后者的需求。

### 3.2.3 订单管理模块分析

订单管理模块主要分为购物车，算费计费，订单流程三个部分：

1）购物车，用户可以将商品加入购物车中，最后选择商品下单购买，也可以删改购物车里的商品，如增加商品数量，从购物车中删除某件商品。

2）算费计费，在购物车中商品就可以下单购买了，下单购买之前会显示购买所选商品需要支付的价格，这个比较复杂，因为订单中可能有的商品是降价的，有的可能是限时促销的，有的可能是几件一起符合某种要求的情况下价格优惠或者采用其他特殊计费方式的，这种情况下需要有一个规则来算费，具体规则详细分析的时候再讨论，这里只设计一个算费功能，计费功能就是当订单提交前后，查询积分是否充足以及扣除相应要支付的积分的功能。

3）订单流程，订单从创建到最终流程结束，是应该有一个工作流的，体现这个过程的就是订单状态的改变，比如正常情况应该是生成订单到订单结束：生成订单->待支付->已支付（申请取消）->已确认。这个模块就是对订单流程进行管理，用户和后台管理员根据实际情况通过一些业务上的操作操作修改订单的状态。

订单管理

理

购物车

算费计费

订单流程

图

3-4订单管理功能图

## 3.3 开发及运行环境

系统开发平台为eclipse,前台用html5 , css , javascript 等技术进行编写，以及运用插件等；后台采用java +servlet。架构设计上，前台和后台交互数据，交互形式为Ajax+json，数据的存储以 mysql 为数据库，部署环境为Tomcat7.0。[11]

## 3.4 可行性分析

技术可行，现有技术可承担开发任务；操作可行，软件能被网上购物人员快速接受；经济可行，拥有建立系统所需的网络设备和软件，具备系统开发条件。[12]

# 第4章系统总体设计

系统设计工作应该自顶向下地进行。首先设计总体结构，然后在逐层深入，直至进行每一个模块的设计。总体设计主要是指在系统分析的基础上，对整个系统的划分（子系统）、数据的存贮规律以及整个系统实现规划等方面进行合理的安排。本章从数据库设计、系统功能、数据库逻辑结构设计、物理结构设计、数据表之间的联系等以下几个方面来对系统进行设计及说明。[13]



## 4.1 体系结构

系统体系结构如图4-1所示，是采用三层结构，通常意义上的三层架构就是将整个业务应用划分为：界面层、业务逻辑层、数据访问层。区分层次的目的是为了高内聚低耦合。[14]

[数据访问层](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E8%AE%BF%E9%97%AE%E5%B1%82)主要是对非原始数据（数据库或者文本文件等存放数据的形式）的操作层，而不是指原始数据，也就是说，是对数据库的操作，而不是数据，具体为业务逻辑层或表示层提供数据服务。业务逻辑层主要是针对具体的问题的操作，也可以理解成对数据层的操作，对数据业务逻辑处理。界面层主要是展示给用户看的，是数据的展示窗口和功能操作的触点。[15]

这种结构使得各个层次之间通过协议和接口规范解耦，单独的层次可以根据规范实现或者在规范下进行十分灵活的修改。比如无论表现层如何定义和更改，逻辑层都能完善地提供服务。[16]



图4-1系统结构图

## 4.2 系统功能结构

系统整体功能模块图如图4-2所示，主要分为用户管理模块和订单管理模块。

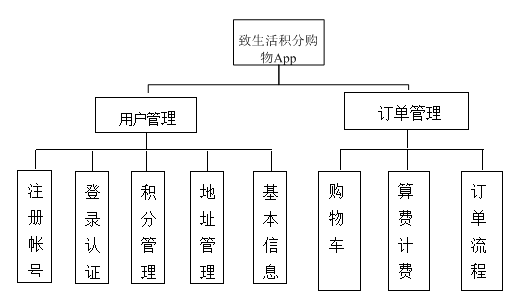


图 4-2系统功能模块图



## 4.3 概念结构设计



商品实体包括的属性有商品ID、当前价格、原始价格、商品图片、商品描述、商品名称等属性。图4-3为商品的实体属性图。

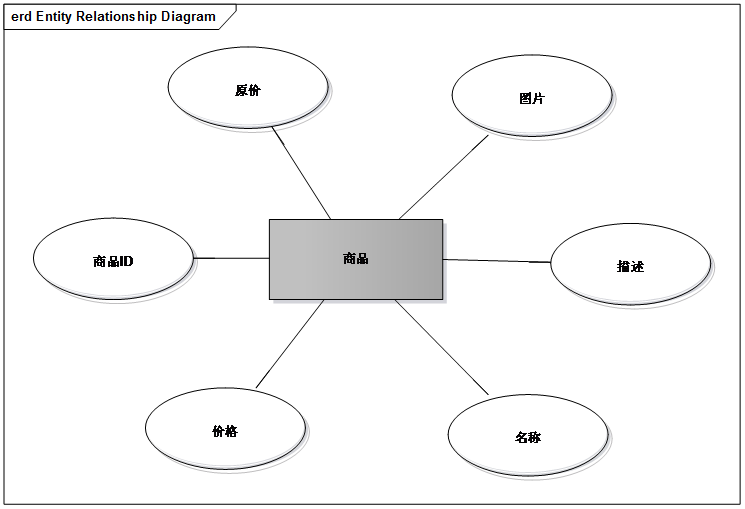


图4-3商品实体属性图

购物车实例：图4-4为购物车实例的属性图



图4-4购物车实例属性图

用户实例：图4-5为用户实例的属性图

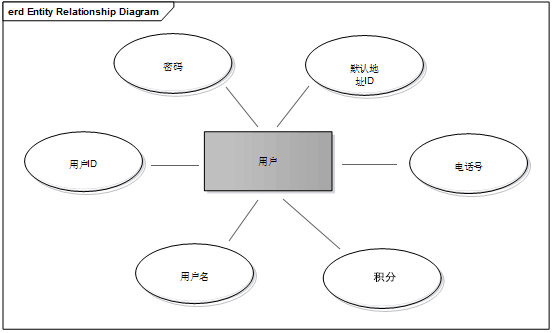


图4-5用户实例属性图

订单实例：图4-6为订单实例的属性图



图4-6订单实例属性图

积分实例：图4-7为积分实例的属性图



图4-7积分实例属性图

地址信息实例：图4-8为地址信息实例的属性图



图4-8用户实例属性图

同一种商品有根据颜色、尺寸等规格的不同有很多分类，当消费者购买商品时，由消费者选定一种或一系列规格值确定了一个商品实例。订单中包含地址信息和商品实例信息，商品加入购物车后才能下单购买，用户关联订单，购物车，地址信息和积分。

图4-9为系统的实体关系图：



图4-9系统实体关系图

## 4.4 数据库逻辑结构设计

表 4-10商品信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **英文字段名** | **中文字段名** | **数据类型** | **长度** | **关键字** | **备注** |
| 1 | prod\_id | 商品标识 | int | 11 | 主键 |  |
| 2 | prod\_name | 商品名称 | varchar | 50 |  |  |
| 3 | prod\_desc | 商品描述 | text |  |  |  |
| 4 | prod\_img | 商品图片 | text |  |  |  |
| 5 | prod\_price | 商品现价格 | int | 11 |  |  |
| 6 | prod\_old\_price | 商品原价格 | int | 11 |  |  |

表 4-11购物车信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **英文字段名** | **中文字段名** | **数据类型** | **长度** | **关键字** | **备注** |
| 1 | car\_id | 购物车标识 | int | 11 | 主键 |  |
| 2 | car\_user\_id | 用户标识 | int | 11 | 外键 | 用户id |
| 3 | car\_prod\_num | 商品数量 | int | 11 |  |  |
| 4 | car\_prod\_id | 商品标识 | int | 11 |  |  |

表 4-12用户信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **英文字段名** | **中文字段名** | **数据类型** | **长度** | **关键字** | **备注** |
| 1 | user\_id | 用户标识 | int | 11 | 主键 |  |
| 2 | user\_name | 用户名称 | varchar | 50 |  |  |
| 3 | user\_number | 用户号码 | varchar | 20 |  |  |
| 4 | user\_password | 用户密码 | varchar | 20 |  |  |
| 5 | user\_def\_addr | 用户默认地址 | varchar | 50 |  |  |
| 6 | user\_credit | 用户积分 | int | 11 |  |  |

表 4-13订单信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **英文字段名** | **中文字段名** | **数据类型** | **长度** | **关键字** | **备注** |
| 1 | order\_id | 订单标识 | int | 11 | 主键 |  |
| 2 | order\_number | 订单号码 | varchar | 20 |  |  |
| 3 | order\_user\_id | 订单用户标识 | int | 11 | 外键 | 用户id |
| 4 | order\_addr\_id | 订单地址标识 | int | 11 | 外键 | 地址id |
| 5 | order\_money | 订单金额 | double |  |  |  |
| 6 | order\_status | 订单状态 | varchar | 20 |  |  |
| 7 | order\_status\_code | 订单状态码 | int | 11 |  |  |
| 8 | order\_create\_time | 订单创建时间 | varchar | 20 |  |  |

表 4-14积分信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **英文字段名** | **中文字段名** | **数据类型** | **长度** | **关键字** | **备注** |
| 1 | inte\_id | 积分标识 | int | 11 | 主键 |  |
| 2 | inte\_user\_id | 用户标识 | int | 11 | 外键 | 用户id |
| 3 | inte\_value | 积分值 | int | 11 |  |  |
| 4 | inte\_grade | 积分等级 | int | 11 |  |  |

表 4-15地址信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **英文字段名** | **中文字段名** | **数据类型** | **长度** | **关键字** | **备注** |
| 1 | addr\_id | 地址标识 | int | 11 | 主键 |  |
| 2 | addr\_user\_id | 用户标识 | int | 11 | 外键 | 用户id |
| 3 | addr\_name | 地址用户名 | varchar | 50 |  |  |
| 4 | addr\_number | 地址用户号码 | varchar | 20 |  |  |
| 5 | addr\_region | 地址地区 | text |  |  |  |
| 6 | addr\_detail | 详细地址 | text |  |  |  |

## 4.5 物理结构设计

1）创建商品信息表

create table product(

prod\_id int(11) not null，--主键

prod\_name varchar(50)，--商品名称

prod\_desc text，--商品描述

prod\_img text，--商品图片

prod\_price int(11) ，--商品现价

prod\_old\_price int(11) ，--商品原价

prod\_type int(11)， --商品类型

)

2）创建购物车信息表

create table car(

car\_id int(11) not null，--主键

car\_user\_id int(11)，--购物车用户标识

car\_prod\_num int(11)，--购物车商品编号

car\_prod\_id int(11)，--购物车商品标识

)

3）创建用户信息表

create table user(

user\_id int not null，--主键

user\_name varchar(50)，--用户名称

user\_number varchar(50)，--用户号码

user\_password varchar(50)，--用户密码

user\_def\_addr int， --用户默认地址

)

4）创建订单信息表

create table order(

order\_id int(11) not null，--主键

order\_number varchar(20)，--订单号

order\_user\_id int(11)，--订单用户标识

order\_addr\_id int(11)，--订单地址标识

order\_money double，--订单金额

order\_status varchar(20)，--订单状态

order\_status\_code int(11)，--订单状态码

order\_create\_time varchar(20)，--订单创建时间

)

5）创建积分信息表

create table inte(

inte\_id int(11) not null，--主键

inte\_user\_id int(11)，--积分用户标识

inte\_value int(11)，--积分值

inte\_grade int(11)，--积分等级

)

6）创建地址信息表

create table address(

addr\_id int(11) not null，--主键

addr\_user\_id int(11)，--地址用户标识

addr\_name varchar(50)，--用户名

addr\_number varchar(20)，--用户号码

addr\_region text，--用户所在地区

addr\_detail text，--用户详细地址

)

## 4.6 数据表之间的联系

数据模型是数据特征的抽象，根据数据模型的设计，最终要转化为数据库。数据表中的字段和数据表之间关系，通过表结构可体现出来。[17] 如图4-17所示：

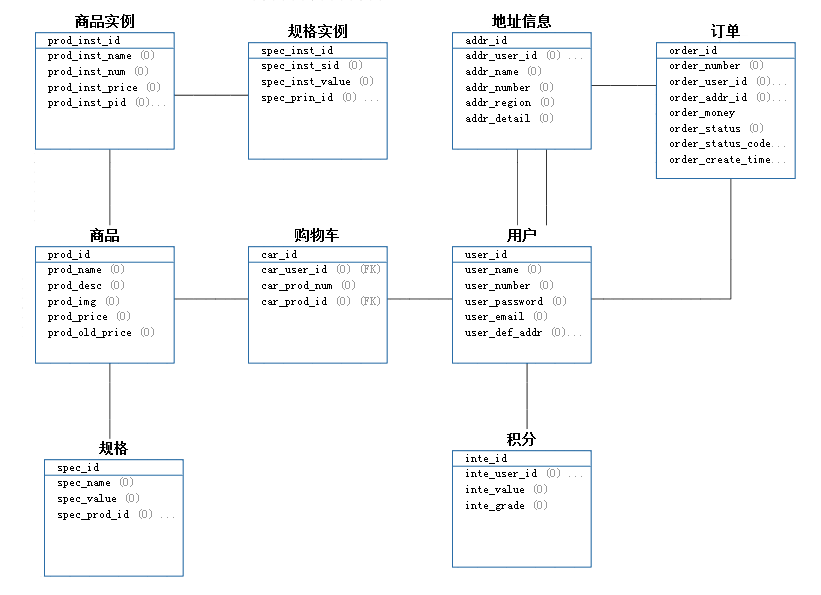


图4-16系统库表间关系图

# 第5章系统功能模块详细设计与实现



## 5.1 用户管理模块设计

### 5.1.1登录界面及个人中心界面设计

登录时需要返回标记为登录状态的token信息，作为后续执行功能的前提条件。如图5-1所示为登录界面，用户填写在主体App上注册时的手机号及服务密码，登录进入积分商城。

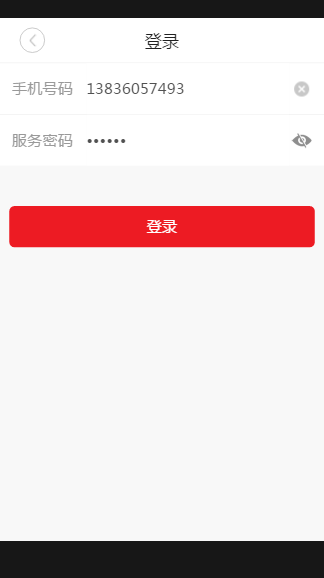


图5-1登录功能界面

如图5-2所示为个人中心界面，个人中心中可查看用户积分，订单信息，以及收货地址信息等。



图5-2个人中心界面

如图5-3所示为积分界面，显示用户积分总额。



图5-3用户积分界面

如图5-4所示为用户收货地址信息界面，显示用户收货地址。

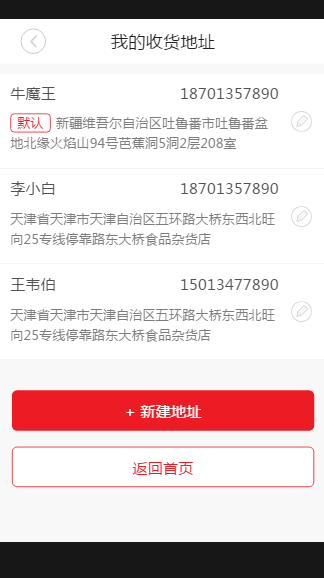


图5-4用户收货地址界面



## 5.2 商品管理模块设计

### 5.2.1加入购物车流程设计

如图5-5流程图所示，为用户进入App后对商品进行加入购物车操作的流程说明。用户进入app后浏览商品，点击商品，进入商品详情页，点击加入购物车，或立即购买，选择好商品规格，即可成功加入购物车。



图5-5加入购物车流程图

### 5.2.2购物车界面设计

如图5-6为购物车界面，包含我的积分，商品图片，信息规格，商家等，用户可自选数量，点击“去结算” 可进入订单结算。

恩恩



图5-6购物车界面



## 5.3 订单管理模块设计

### 5.3.1订单流程设计

如图5-7为订单流程图，用户进入App后，进入购物车，选择要购买的商品，选择收货地址，及填写订单备注（选填），点击提交订单，输入密码，密码正确及积分满足，则跳转到提交订单成功页面，否则，跳转到提交订单失败页面。



图5-7订单流程图

### 5.3.2订单流程界面设计

当前置条件都满足，后台调用相应接口时， 即可完成用户对商品的支付，生成相应信息保存到数据库中。如图5-8为订单结算界面，用户点击“去结算”跳入此页面，可查看当前积分，可选择商品邮寄地址，填写订单备注，然后点击提交订单，输入支付密码，即可完成对商品的结算，结算成功跳入成功页面。



图5-8订单结算界面

如图5-9为我的订单界面，商品付款成功后，商品信息在我的订单中可查看，包括厂家，图片信息，描述信息，实际付款积分，收货状态等。



图5-9我的订单界面

# 第6章系统软件测试

软件工程师设计出软件蓝图后，又经过编码而实现了软件产品。软件测试则尽力找出软件设计的失败与不足之处，再加以纠正，确保软件设计无差错的实现。表面看设计是建造，而测试是破坏。最终的任务是要建造高质量的软件。[18]



## 测试目的

基于不同的立场存在两种完全不同的测试目的，从用户的角度出发，普遍希望通过软件测试暴露软件中隐藏的错误和缺陷，以考虑是否可以接受该产品。而从软件开发者的角度出发，则希望测试成为表明软件产品中不存在错误的过程，验证软件已正确地实现了用户的要求，确立人们对软件质量的信心。鉴于此，就软件测试目的提出以下观点：

1. 测试是程序的执行过程，目的在于发现错误；
2. 一个好的测试用例在于能发现至今未发现的错误；
3. 一个成功的测试时发现了至今未发现错误的测试；

测试的目的是想以最少的时间和人力找出软件中潜在的各种错误和缺陷，如果成公地实施了测试，就能够发现软件中的错误。测试的附带收获是，它能够证明软件的功能和性能与需求说明相符。此外，实施测试收集到的测试结果数据为可靠性分析提供了依据。

根据这样的测试目的，软件测试的原则应该是：

1. 应当把“尽早地和不断地进行软件测试”作为软件开发者的座右铭；
2. 测试用例应由测试输入数据和与之对应的预期输出结果这两部分组成；
3. 程序员应避免检查自己的程序；
4. 在设计测试用例时，应当包括合理的输入条件和不合理的输入条件；
5. 充分注意测试中的群集现象；
6. 严格执行测试计划，排除测试的随意性；
7. 应当对每一个测试结果做全面检查；
8. 妥善保存测试计划，测试用例，出错统计和最终分析报告，为维护提供方便。[19]

## **6.2测试方法**

软件测试的测试分类有很多，如白盒测试，黑盒测试，自动化测试。

白盒测试：是把测试对象看作一个打开的盒子。利用白盒测试法进行动态测试时，需要测试软件产品的内部结构和处理过程，不需要测试软件产品的功能。白盒测试是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都有按预定要求正确工作，而不顾它的功能，白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。

黑盒测试：是根据软件的规格对软件进行测试，这类测试不考虑软件内部的运作原理，因此软件对用户来说就像一个黑盒子。软件测试人员以用户的角度，通过各种输入和观察软件的各种输出结果来发现软件存在的缺陷，而不关心程序具体如何实现的一种软件测试方法。

自动化测试：使用自动化测试工具来进行测试，测试一般不需要人干预，通常在GUI、性能测试和功能测试中用的比较多，通过录制测试脚本，然后执行这个测试脚本来实现测试过程的自动化。国内领先的自动化测试服务提供商是泽众软件，自动化测试工具有QTP、Testcomplete、AutoRunner和TAR等。

本系统测试方法为黑盒测试，即输入相应的数据，观察输出结果是否正确，从而测试软件是否已经达到预期目的效果，从而检测程序正确性。[20]

## 6.3测试用例

### 6.3.1测试用户订单提交的测试用例

所属模块：订单管理模块

用例目的：测试订单提交功能

前提条件：用户成功登陆App，在购物车中选择商品及数量，商品积分小于用户积分余额。

|  |  |  |
| --- | --- | --- |
| 输入 | 期望结果 | 实际情况 |
| 点击提交订单，输入密码 | 1. 密码验证正确  2. 跳转订购成功页面  3. 扣除相应积分  4. 在我的订单中增加一条订单信息 | 1．密码正确  2. 跳转成功  3. 用户积分余额扣款正确页面显示正确，数据库存储正确  4. 我的订单中商品记录添加成功，页面正确，数据库存储正确 |
| 切换账户，重新点击提交，输入密码 | 1. 密码验证正确  2. 跳转订购成功页面  3. 扣除相应积分  4. 在我的订单中增加一条订单信息 | 1．密码正确  2. 跳转成功  3. 用户积分余额扣款正确页面显示正确，数据库存储正确  4. 我的订单中商品记录添加成功，页面正确，数据库存储正确成功 |

### 6.3.2测试商品添加的测试用例

所属模块：商品管理模块

用例目的：测试用户购买产品时商品添加是否成功

前提条件：用户成功登陆App，在主页，我的抢购页，分类页中浏览商品信息，点击加入购物车，选择商品相应属性及数量。

|  |  |  |
| --- | --- | --- |
| 输入 | 期望结果 | 实际情况 |
| 浏览商品详情，点击加入购物车 | 1.出现商品加入成功提示，  2.数据库中相应账户增加相应商品信息  3.在购物车中页面中出现商品信息 | 1.信息提示正确  2.数据库中数据正确  3.购物车页面商品信息显示正确 |
| 切换账户，浏览商品详情，点击加入购物车 | 1.出现商品加入成功提示，  2.数据库中相应账户增加相应商品信息  3.在购物车中页面中出现商品信息 | 1.信息提示正确  2.数据库中数据正确  3.购物车页面商品信息显示正确 |

# 总结

本系统是一个积分商城购物App，主要包括前台页面设计展现，以及后台逻辑处理，及数据库存储。前台方面采用html5作为页面骨架，用css及javascrip完成页面相关样式及交互动作，并使用一些插件来完成特定功能，如首页轮播图，地址选择三联动等等。选择h5来开发前端相对于原生开发，更为高效简洁，且跨平台。后台开发主要是逻辑及数据，采用java +servlet。架构设计上，前台和后台交互数据，交互形式为Ajax+json，数据的存储以 mysql 为数据库，部署及运行环境为Tomcat7.0x。后台主要开发模块为用户管理模块、商品管理模块、以及订单管理模块。用户管理模块主要包括注册账号、登录认证、积分管理、地址管理、基本信息这5方面功能，各个功能调用相应接口实现；商品管理模块是对商品的增删改查的功能进行实现，订单管理模块包含购物车、算费计费、订单流程这三个功能，主要针对用户支付下单到确认付款的一些列，扣除相应积分。后台在开发过程中考虑的问题很多，虽然一开始比较盲目，不知道从何开始，但是一点一点做下来，没有想象的那么难，而且通过询问其他人，获得了许多开发上的经验，很感谢这次毕设，让我收获颇多。

在做毕业设计时，遇到很多问题，软件也存在一定不足，但只有发现问题面对问题才有可能解决问题，不足和遗憾不会给我打击只会更好的鞭策我前行，经过这次3个多月的努力终于完成了毕业设计，这次毕业设计不仅系统的复习了以前的知识，而且在设计中遇到很多难题通过查找资料，询问老师、同学，让我学会更过知识，为以后工作打下坚实的基础。

总体来说，本系统有良好的扩展性，且便于系统维护，使用简单，虽然离上线的软件标准还很远，但是基本流程，场景，大致相同。由于时间，知识有限，系统还有很多方面需要改进。我相信，在以后的学习中知识会更加的完善，所呈现出的系统也会越来越好。

# 致谢

本论文的工作是在导师宁静峰老师的指导下完成的，从论文的开题到资料收集，理论的建立，每一个程序、模块的完成以及最后论文的成稿都凝聚着老师的心血。宁老师，治学态度严谨，对待学生耐心，工作态度积极，给予我很大的帮助，让我学到很多东西。感谢老师的耐心指导和悉心教诲。

很感谢大学期间耐心传授知识的老师，虽然不是每节课都在讲技术相关的，但是很感谢老师们的倾心分享，分享自己的人生经历，态度，想法，教会我们要做一个怎样的人，应该要做一个怎样的人。最为感谢我的实验室老师郭建伟老师，在他的实验室，我学到了很多东西，让我在迷茫的时候知道应该怎么做，郭老师在生活和技术上都给与了我极大指导，在此表示由衷感谢。

最后，感谢父母家人一直以来的照顾和陪伴，感谢我这四年来的同学，无论什么时候，何时何地，我有困难，都会帮助我，真的很感谢你们，还有我的学长，无论问什么，无论忙不忙，都会回答我的问题，不厌其烦的。大学四年还不是终点，希望我们未来会更好，一起加油，努力，进步。

# 参考文献

[1] 阿瓦德《电子商务：从愿景到现实第三版》.人民邮电出版社.2009年01月.

[2] 李刚《疯狂Java讲义》电子工业出版社.2012年01月.

[3] [美]巴塞特《Json必知必会》. 人民邮电出版社. 2013年04月.

[4] [英]格雷恩《Ajax实战-实例详解》人民邮电出版社.2008年10月.

[5] 张峋, 杨三成《关键技术:JSP 与 JDBC 应用详解[M]》. 中国铁道出版社.2010.年11.

[6] FLANAGAN《Jsp技术手册》中国电力出版社.2002年06月.

[7] [法]内德尔库(Clement Nedelcu)《学习Nginx HTTP Server中文版》. 清华大学出版社.2012年 .

[8] David Flanagan《Javascript权威指南》机械工业出版社第6版.2012年04月.

[9] 腾讯前端Alloy Team 《Javascript 设计模式与开发实践》.人民邮电出版社.2015年01月.

[10] 李兴华《Java开发实战经典》清华大学出版社.2009年08月.

[11] 孙卫琴《Tomcat 与 Java Web 开发技术详解[M]》. 第 2 版. 电子工业出版社,2009年 01.

[12] [美]马克艾伦维斯《数据结构与算法分析》.机械工业出版社.2016年03月01日.

[13] 刘京华等《Java Web 整合开发王者归来[M]》. 清华大学出版社,2010 年 01 月 .

[14] 杜波依斯《MySQL 技术内幕[M]. 第 4 版.》人民邮电出版社.2011 年 07 月 .

[15] 姜承尧《Mysq技术内幕 InnoDB存储引擎》机械工业出版社.2011年01.

[16] [澳]麦斯阿塞克 《需求分析与系统设计》机械工业出版社.2009年09月.

[17] [美]爱德华 克劳利《系统架构》机械工业出版社.2016年12月.

[18] [美]Glenford J.Myers [美]Tom Badgett《软件测试的艺术》.机械工业出版社.2012年04月01日.

[19] 王朔韬《软件是这样炼成的》.清华大学出版社发行部.2016年01月01日.

[20] [美]Weinberg《颠覆完美软件》.电子工业出版社.2015年07月.

# 附录

/\*\*

\*

\* **@author** zhaoql

\*

\*/