**淘京东宝网上购物平台**

**设计说明**

|  |  |  |  |
| --- | --- | --- | --- |
| **项目组成员信息** | | |  |
| **小组名称** | **Team1** | | |
| **学号** | **姓名** | **本文档中主要承担的工作内容** | **贡献率** |
| **14061129** | 韩世依 | 编写文档 | 0.4 |
| **14061139** | 刘铮 | 检查修改文档 | 0.2 |
| **14061145** | 李子涵 | 检查修改文档 | 0.2 |
| **14061131** | 王震 | 检查修改文档 | 0.2 |

2016-11

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 1.0 | 2016-11-15 | 韩世依 | 刘铮 | 设计说明初版 |
| 1.1 | 2016-11-20 | 韩世依 | 王震 | 详细设计的完善 |
| 1.2 | 2016-11-25 | 韩世依 | 李子涵 | 数据库设计的完善 |
| 1.3 | 2016-11-30 | 韩世依 | 刘铮、王震 | 关键问题和解决方案总结 |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[1. 范围 1](#_Toc265683273)

[1.1 标识 1](#_Toc265683274)

[1.2 项目概述 1](#_Toc265683275)

[1.3 文档概述 1](#_Toc265683276)

[1.4 术语和缩略词 1](#_Toc265683277)

[1.5 引用文档 1](#_Toc265683278)

[2. 需求概述 2](#_Toc265683279)

[3. 体系结构设计 2](#_Toc265683280)

[3.1 总体结构 2](#_Toc265683281)

[3.2 功能分配 2](#_Toc265683282)

[3.3 关键问题及解决方案 2](#_Toc265683283)

[4. 接口设计 2](#_Toc265683284)

[4.1 用户界面设计 2](#_Toc265683285)

[4.2 外部接口设计 3](#_Toc265683286)

[4.3 内部接口设计 3](#_Toc265683287)

[5. 数据结构设计 3](#_Toc265683288)

[5.1 公共数据结构设计 3](#_Toc265683289)

[5.2 数据库设计 3](#_Toc265683290)

[6. 详细设计 3](#_Toc265683291)

# 范围

## 标识

文档标识号：Team1\_淘京东宝网上购物平台\_SDD1.3

文档标题：《淘京东宝网上购物平台》设计说明书

版本号：1.3

## 项目概述

本系统是一个在线互联网电商网站，给注册用户提供在在线购物的平台和提供便捷的物流快递服务。用户可以在网站上挑选自己喜欢的商品购买，同时支持在线支付服务，为用户提供一站式服务。由于运营规模小，成本有限，和阿里巴巴集团合作，支付方式使用支付宝，并且整个平台计划以web作为前段手段，暂未提供app的支持，待到用户量和反响得到一定收获，再做进一步开发。

这是网站初次进行开发和运行。

投资方是淘京东宝科技公司，需方是淘京东宝科技公司，用户为所有网络消费者，开发方为140615科技公司，支持机构为140615科技公司，管理员为淘京东宝科技公司。

计划运行在淘京东宝的云服务器上。

## 文档概述

### 编写目的

为明确软件需求、安排项目规划与进度、组织软件开发与测试，撰写本文档。

### 文档用途

软件需求规格说明书的编制是为了便于用户、开发人员进行理解和交流，反映出用户问题的结构，使用户和软件开发者双方对该软件的初始规定有一个共同的理解，可以作为软件开发工作的基础和依据，并作为确认测试和验收的依据。

### 文档内容

包含硬件、功能、性能、输入输出、接口需求、警示信息、保密安全、数据与数据库、文档和法规的要求。

### 文档保密性与私密性

本文档的预期读者是与购物平台开发有联系的决策人，项目承担者，开发组成人员，支持本项目的领导和公司人员，软件验证者。

## 术语和缩略词

暂无。

## 引用文档

1. 国家标准《项目开发计划（GB856T——88）》
2. 《淘京东宝网上购物平台》开发计划2.0
3. Team1\_淘京东宝网上购物平台\_SRS1.3

# 需求概述

该系统包括以下功能：登录；注册；浏览目录；搜索商品；查看商品；购物车管理；订单处理；会员信息管理；客户服务；商家商品管理；商家订单处理；商家信息管理等。

其中，购物车管理包括：加入购物车；查看购物车；修改购物车中的商品；删除购物车中的商品；清空购物车等。

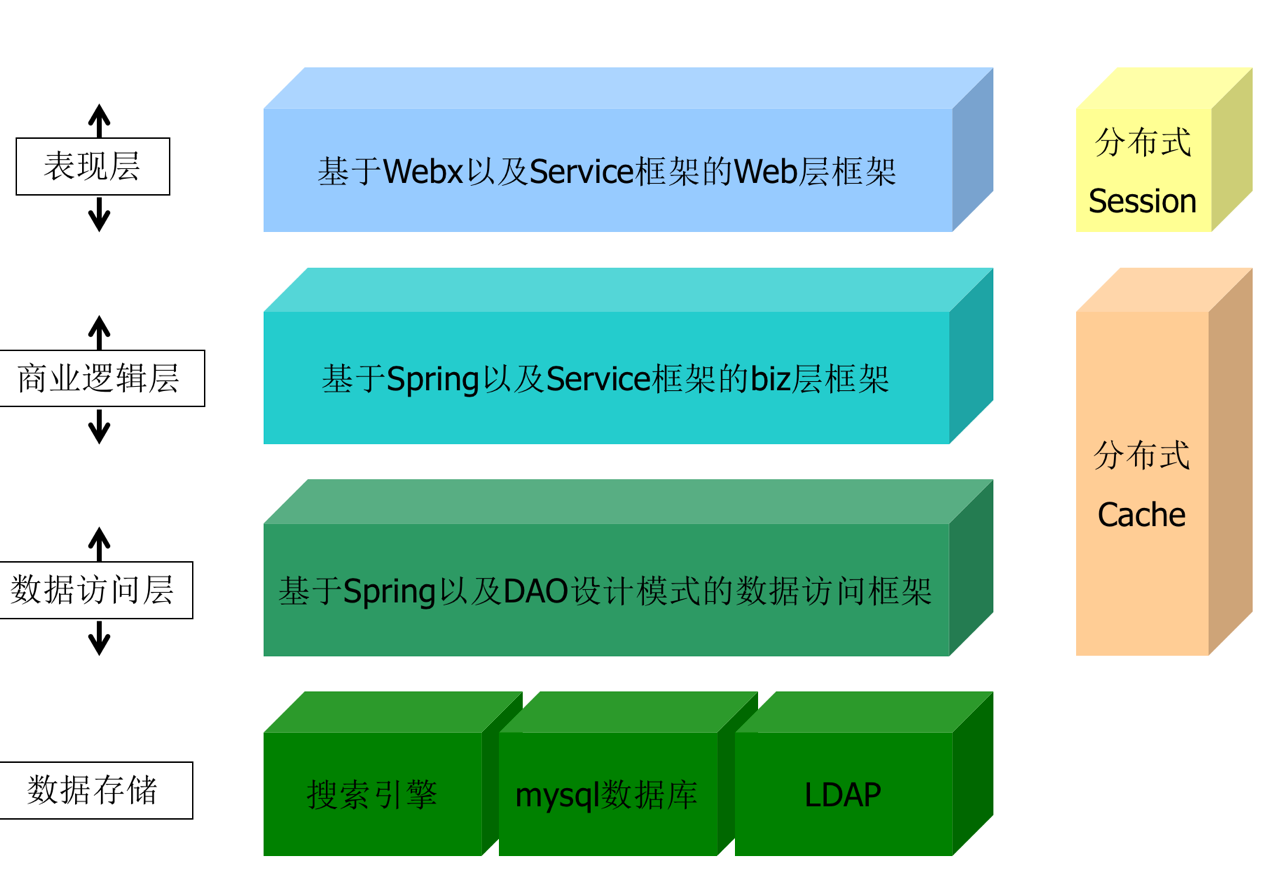
订单处理包括：结账；配置收货人信息；配置送货方式；配置付款方式；确认订单；查看订单；修改订单；删除订单等。

会员信息管理包括：修改个人基本信息；修改收货地址；查看消费记录；查看积分等。

商家商品管理包括：添加商品；完善商品信息；删除商品等。

# 体系结构设计

## 总体结构



在表现层，我们使用了Webx框架，Webx是一个在[阿里巴巴集团](http://baike.baidu.com/subview/1247049/1247049.htm)内部广泛使用的，层次化、模块化的一个Web框架。 Webx是基于经典MVC设计模式的WEB框架，推崇页面驱动和约定胜于配置的理念。 Webx是一个基于[Spring](http://baike.baidu.com/view/23023.htm)的组件框架。组件是一个软件包，它可以被其它组件扩展，也可以扩展其它组件。在我们的系统中，使用webx以及service框架作为web层框架来实现高度可扩展的网站开发模式，高效利用框架实现系统。

在商业逻辑层，区别于传统的SLSB，我们的应用采用了SPRING架构来实现业务逻辑对象的控制，使用别人搭建好的框架节省了开发时的逻辑复杂度。

同时，我们采用分布式cache和分布式session的设计，最大化分布式系统的优势，节省硬件开销，加快访问速度。

数据存储层，我们使用搜索引擎来帮助用户搜索到他们想要的物品，搜索的时候，先通过LDAP（Lightweight Directory Access Protocol）。LDAP目录以树状的层次结构来存储数据，快速的找到数据所在数据库，然后进行对应的数据库操作，返回搜索数据。

本应用的硬件体系结构相对简单，由于规模较小，采用小型的数据库进行数据储存 。

## 功能分配

需求文档中说明了以下功能：

登陆功能： 登陆功能作为用户购买物品的资格接口，没有登陆的用户只能够拥有访问商品查看商品的权限，并不能购买商品。

注册功能：没有账户的用户需要注册以得到登陆的账号密码，账号可以使用邮箱，用户名或者手机号。

访问商品：作为本应用的核心模块，访问商品让用户直接的可以查看商品的属性和价格。

购物车功能：用户将商品放入购物车是购买物品的唯一途径。

购买提交订单功能：将用户购物车的商品订单提交到数据库。

支付功能：支付订单的总额的对应金钱，实现网站收益。

**说明需求文档当中各项功能同总体结构的对应关系**。

## 关键问题及解决方案

### Webx的使用

**对于webx这样的新技术，开始的设置和使用是我们的最大难点**

* + - * 安装JDK

Webx需要JDK 5.0以上的版本。从这里下载并安装它：http://www.oracle.com/technetwork/java/javase/。10.1.2. 安装和配置maven

Webx需要maven 2或更高版本。从这里下载并安装它：http://maven.apache.org/。

不需要对maven进行特殊的配置，因为运行Webx应用所需要的所有包都存放在全世界共享的中心Maven仓库（Central Maven Repository）中。Maven将从那里自动获取所有的jar包、源代码和javadoc。

* + - * 安装集成开发环境
      * 安装如下插件：

• Maven eclipse插件：http://eclipse.org/m2e/

• Git eclipse插件：http://eclipse.org/egit/

• SpringExt eclipse插件：http://openwebx.org/updatesite/

* + - * 创建应用

请打开命令行工具（Windows cmd或Unix/Linux bash），输入如下命令：

mvn archetype:generate \

-DgroupId=com.alibaba.webx \anchor co.firstapp.archetype.groupId

-DartifactId=tutorial1 \anchor co.firstapp.archetype.artifactId

-Dversion=1.0-SNAPSHOT \anchor co.firstapp.archetype.version

-Dpackage=com.alibaba.webx.tutorial1 \anchor co.firstapp.archetype.package

-DarchetypeArtifactId=archetype-webx-quickstart \

-DarchetypeGroupId=com.alibaba.citrus.sample \

-DarchetypeVersion=1.8 \

-DinteractiveMode=false

由于Windows下不支持命令换行，请改用非换行版：

mvn archetype:generate -DgroupId=com.alibaba.webx -DartifactId=tutorial1 -Dversion=1.0-SNAPSHOT -Dpackage=com.alibaba.webx.tutorial1 -DarchetypeArtifactId=archetype-webx-quickstart -DarchetypeGroupId=com.alibaba.citrus.sample -DarchetypeVersion=1.8 -DinteractiveMode=false

命令执行完后，你会看见一个新目录：tutorial1。它就是我们刚刚创建的新项目。

* + - * 运行应用

进入刚创建的tutorial1目录，在此目录下执行maven命令：

mvn jetty:run

这条命令会启动Jetty Server，默认的端口是8081。

**3.3.2 Spring的使用**

spring框架在使用的时候方便高效，但是在开始设置的时候花了我们一番功夫。

1. jar包引入

　　Spring 2.5.6：spring.jar、spring-webmvc.jar、commons-logging.jar、cglib-nodep-2.1\_3.jar

　　Hibernate 3.6.8：hibernate3.jar、hibernate-jpa-2.0-api-1.0.1.Final.jar、antlr-2.7.6.jar、commons-collections-3.1、dom4j-1.6.1.jar、javassist-3.12.0.GA.jar、jta-1.1.jar、slf4j-api-1.6.1.jar、slf4j-nop-1.6.4.jar、相应数据库的驱动jar包

2. web.xml配置（部分）

<!-- Spring MVC配置 -->

<!-- ====================================== -->

<servlet>

<servlet-name>spring</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<!-- 可以自定义servlet.xml配置文件的位置和名称，默认为WEB-INF目录下，名称为[<servlet-name>]-servlet.xml，如spring-servlet.xml

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/spring-servlet.xml</param-value> 默认

</init-param>

-->

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>spring</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

<!-- Spring配置 -->

<!-- ====================================== -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- 指定Spring Bean的配置文件所在目录。默认配置在WEB-INF目录下 -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:config/applicationContext.xml</param-value>

</context-param>

3. spring-servlet.xml配置

　　spring-servlet这个名字是因为上面web.xml中<servlet-name>标签配的值为spring（<servlet-name>spring</servlet-name>），再加上“-servlet”后缀而形成的spring-servlet.xml文件名，如果改为springMVC，对应的文件名则为springMVC-servlet.xml。

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.springframework.org/schema/p"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd

http://www.springframework.org/schema/context <A href="http://www.springframework.org/schema/context/spring-context-3.0.xsd">http://www.springframework.org/schema/context/spring-context-3.0.xsd</A>">

<context:annotation-config />

<context:component-scan base-package="controller"></context:component-scan>

<bean class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter" />

<!-- 对转向页面的路径解析。prefix：前缀， suffix：后缀 -->

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver" p:prefix="/jsp/" p:suffix=".jsp" />

</beans>

4. applicationContext.xml配置

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="

http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd

http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">

<bean id="sessionFactory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">

<property name="configLocation">

<value>classpath:config/hibernate.cfg.xml</value>

</property>

</bean>

<bean id="transactionManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">

<property name="sessionFactory">

<ref local="sessionFactory"/>

</property>

</bean>

<tx:annotation-driven transaction-manager="transactionManager" proxy-target-class="true"/>

<bean id="loginService" class="service.LoginService"></bean>

<bean id="hibernateDao" class="dao.HibernateDao">

<property name="sessionFactory" ref="sessionFactory"></property>

</bean>

</beans>

**3.3.3 MySQL数据库的使用**

1）首先检查[系统](http://www.2cto.com/os/)中是否已经安装了MySQL

在终端里面输入 sudo netstat -tap | grep [mysql](http://www.2cto.com/database/MySQL/)

若没有反映，没有显示已安装结果，则没有安装。否则表示已经安装。

2）如果没有安装，则安装MySQL.

在终端输入 sudo apt-get install mysql-server mysql-client

在此安装过程中会需要输入root用户(管理MySQL[数据库](http://www.2cto.com/database/)用户，非Linux系统用户)密码，按照要求输入即可。

3）测试安装是否成功：

在终端输入 sudo netstat -tap | grep mysql

4）也可通过登录MySQL测试

在终端输入 mysql -uroot -p 接下来会提示你输入密码，输入正确密码，即可进入。

5）MySQL的一些简单管理：

启动MySQL服务： sudo start mysql

停止MySQL服务： sudo stop mysql

修改 MySQL 的管理员密码： sudo mysqladmin -u root password newpassword

设置远程访问(正常情况下，mysql占用的3306端口只是在IP 127.0.0.1上监听，拒绝了其他IP的访问（通过netstat可以查看到）。取消本地监

听需要修改 my.cnf 文件：)：

sudo vi /etc/mysql/my.cnf

bind-address = 127.0.0.1 //找到此内容并且注释

6）MySQL安装后的目录结构分析(此结构只针对于使用apt-get install 在线安装情况)：

数据库存放目录： /var/lib/mysql/

相关配置文件存放目录： /usr/share/mysql

相关命令存放目录： /usr/bin(mysqladmin mysqldump等命令)

启动脚步存放目录： /etc/rc.d/init.d/

# 接口设计

## 用户界面设计



图1 登录页面图

登陆界面需要包括用户名输入和密码输入，对于没有注册的用户，提供了注册接口，登陆界面信息较少，所以作为模态框的形式显示，我们也同时提供了多种登陆方式，包括qq和微信等，以免用户忘记自己的信息而导致无用账号等情况的发生。



图2 注册页面图

有两层页面，第一层让用户选择是注册会员用户还是注册商家用户。第二层页面为表单页面，让对应类型的用户填 各个属性项。为了保证用户不重复注册和友商进行网页攻击，我们增加了手机验证环节和验证码功能。



图3 浏览目录页面1

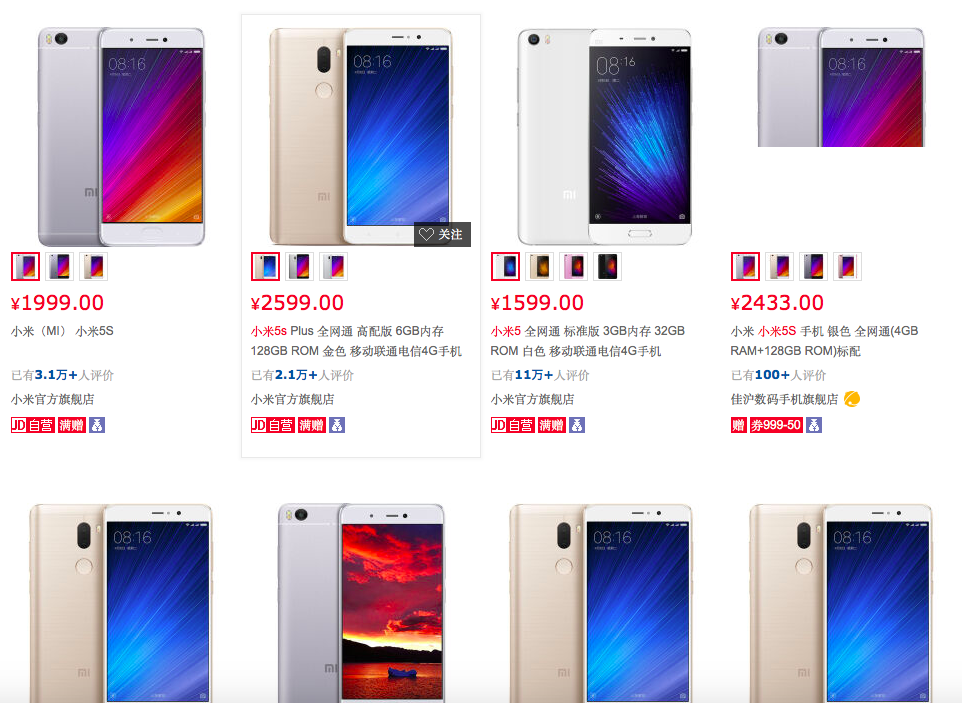


图4 浏览用户页面2

浏览目录页面包括两种，一种是一般商品浏览，在一般商品浏览界面中，会对热销的商品和应季火爆的商品进行推荐，以增长对应商品的销量，一种是搜索商品浏览，会根据用户的搜索关键词来寻找对应的上皮。

一般商品浏览页面如图3所示。其中包括一个商品类别选择类别框，用户可以选择一个类别的商品，并进入另一个浏览目录页面。此页面中还包括各种类别中对会员用户推荐商品的浏览界面。

搜索商品浏览中列举用户搜索的商品，如图4所示。

搜索商品

此功能没有特定页面，但在会员用户的浏览目录页面、查看商品页面中的顶部都出现一个搜索框供用户搜索商品。

查看商品页面



图5 查看商品页面

此页面供用户浏览一个具体商品，页面中需要包括商品详细介绍信息、商品价格信息、供用户选择商品属性的选择框、商品的评论信息已经供用户加入购物车的按钮。

购物车管理页面



图6 购物车管理页面

购物车管理页面需要显示用户已加入购物车的商品信息、商品总价，并有按钮和输入框供用户选择商品数量，还有按钮可以删除购物车内商品，还有按钮让用户进入结算页面。

结算界面直接链接支付宝，剩下的交给支付宝来完成，完成支付再返回此页面显示支付成功或者失败。

## 外部接口设计

## 4.2.1硬件接口

用户：

本软件由于是个web应用，硬件接口要求较低，无论是移动端还是pc端还是mac端都可以运行。但依旧有推荐配置和最低配置以达到软件的性能最优方案。

最低配置要求：

CPU Intel(R) P4/PE双核

内存 2GB

推荐配置要求：

CPU Intel(R) Core(TM)2 Duo CPU

内存 3GB

服务器：

由于本软件使用用户和商家们长时间高频率的访问，本程序需要一个高能的服务器，推荐配置如下，如果服务器实际配置低于推荐配置，所有的后果由管理者承担。

推荐配置：

英特尔® 至强® 处理器 E5-2603 v3

• 4GB内存 / 1TB 7.2K 3.5英寸热插拔硬盘

• PERC H330 RAID控制器/iDRAC8 Express/

•单个, 热插拔电源 (1+0), 495瓦

## 4.2.2软件接口

本web应用经过程序员们的努力，适应了主流的浏览器，并经过了以下内核的测试，流畅度均能达到主流标准。

Trident内核代表产品Internet Explorer

Gecko内核代表作品Mozilla FirefoxGecko

WebKit内核代表作品Safari、Chromewebkit

Presto内核代表作品OperaPresto是由Opera Software

本软件不同于很多基于老的flash驱动的网页，全面支持html5（对于商品，可以使用视频来展示商品。

## 内部接口设计

本应用的数据库采用分布式方式，所有的数据分布在各个数据中心，方便各个模块分开管理。

登陆接口和账号数据库连接，通过ajax方式保密的传递账号密码，然后对应后段对比数据库后再穿回登陆成功失败信息。

注册接口和账号数据库连接，通过ajax方式保密的传递账号密码，然后用sql语句操作数据库，设置了严密的触发器来为新用户开通各种权限。

查看商品界面要和商品数据库进行连接，通过jsp传信息至前端页面。该页面还要和用户信息数据库进行连接，先读取用户的偏好信息和购买记录，

# 数据结构设计

## 公共数据结构设计

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项名含义 | 别名 | | 数据类型 | 长度 | 取值范围 | 取值含义 | 与其他数据项的逻辑关系 | |
| 昵称 | 会员的昵称 | nickName | | TEXT | 20 | ASCII | 须为合法字符 | 无关 | |
| 性别 | 会员的性别 | sex | | TEXT | 1 | ASCII | 男或女 | 无关 | |
| 生日 | 会员的生日 | birthday | | DATE | 50 | ASCII | 须为合法日期 | 无关 | |
| 兴趣爱好 | 会员的兴趣爱好 | hobby | | TEXT | 50 | ASCII | 须为合法字符 | 无关 | |
| 会员名 | 会员的名字 | name | | TEXT | 10 | ASCII | 须为合法字符 | 无关 | |
| 会员积分 | 会员的积分 | score | | INT | 100 | ASCII | 须为合法字符 | 无关 | |
| 商品名称 | 商品的名称 | CommodityName | TEXT | | 10 | ASCII | 需为合法字符 | | 无关 |
| 商品价格 | 商品的单价 | commodityPrice | TEXT | | 100 | NUMBER | 须为合法数字串 | | 无关 |
| 商品类型 | 商品的种类 | commodityType | TEXT | | 20 | ASCII | 须为合法 串 | | 无关 |
| 商品优惠价格 | 商品的优惠后单价 | commodityCutPrice | TEXT | | 10 | NUMBER | 须为合法数字串 | | 无关 |
| 商品折扣 | 商品的折扣 | commodityPriceOff | TEXT | | 10 | NUMBER | 须为合法数字串 | | 无关 |
| 商品描述 | 商品的信息说明 | commodityDescription | TEXT | | 500 | ASCII | 须为合法 串 | | 无关 |
| 商品品牌 | 商品的品牌 | commodityBrand | TEXT | | 15 | ASCII | 须为合法 串 | | 无关 |
| 系统时间 | 系统的时间 | currentSystemTime | Time | | 12 | number | 需要和服务器时间统一 | | 无关 |

## 数据库设计

### 逻辑结构设计

C:\Users\zihan\Desktop\ER.pngER

图7：ER图

表格1：会员基本信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 数据项名 | 索引或键 | 备注 |
| 1 | nickName | TEXT | 20 | 昵称 | 键 | 会员的昵称 |
| 2 | sex | TEXT | 1 | 性别 |  | 会员的性别 |
| 3 | birthday | DATE | 50 | 生日 |  | 会员的生日 |
| 4 | hobby | TEXT | 50 | 兴趣爱好 |  | 会员的兴趣爱好 |
| 5 | name | TEXT | 10 | 会员名 |  | 会员的名字 |
| 6 | score | INT | 100 | 会员积分 |  | 会员的积分 |
| 7 | expenseCalendar | TEXT | 50 | 消费记录 |  | 会员的消费记录 |
| 8 | address | TEXT | 100 | 收货地址 |  | 会员你的收货地址 |
| 9 | Email | TEXT | 50 | 邮箱 |  | 会员你的邮箱 |
| 10 | realName | TEXT | 10 | 真实姓名 |  | 会员真实姓名 |
| 11 | phoneNumber | TEXT | 50 | 联系电话 |  | 会员联系电话 |

表格2：消费记录表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 数据项名 | 索引或键 | 备注 |
| 1 | nickName | TEXT | 20 | 昵称 | 键 | 会员的昵称 |
| 2 | Num | INT | 200 | 订单数量 |  | 订单的数量 |

表格3：订单信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 数据项名 | 索引或键 | 备注 |
| 1 | id | TEXT | 20 | 订单编号 | 键 | 订单的编号 |
| 2 | time | DATE | 20 | 时间 |  | 订单产生的时间 |
| 3 | buyerID | TEXT | 20 | 买家编号 |  | 买家的编号 |
| 4 | sellerID | TEXT | 20 | 卖家编号 |  | 卖家的编号 |

表格4：购物车信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 数据项名 | 索引或键 | 备注 |
| 1 | name | TEXT | 20 | 会员昵称 | 键 | 会员的昵称 |
| 2 | Id | TEXT | 20 | 商品编号 |  | 购物车里商品的编号 |
| 3 | time | DATE | 20 | 添加时间 |  | 添加进入购物车的时间 |

表格5：商品信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 数据项名 | 索引或键 | 备注 |
| 1 | ID | INT | 20 | 商品编号 | 键 | 商品的编号 |
| 2 | CommodityName | TEXT | 10 | 商品名称 |  | 商品名称 |
| 3 | commodityPrice | TEXT | 100 | 商品价格 |  | 商品单价 |
| 4 | commodityType | TEXT | 20 | 商品类型 |  | 商品的种类 |
| 5 | commodityCutPrice | TEXT | 10 | 商品优惠价格 |  | 商品的优惠后单价 |
| 6 | commodityPriceOff | TEXT | 10 | 商品折扣 |  | 商品的折扣 |
| 7 | commodityDescription | TEXT | 500 | 商品描述 |  | 商品的信息说明 |
| 8 | commodityBrand | TEXT | 15 | 商品品牌 |  | 商品的品牌 |
| 9 | isPost. | TEXT | 10 | 是否包邮 |  | 说明是否包邮 |
| 10 | date | DATE | 20 | 上架日期 |  | 商品的上架时间 |

表格6：商家信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 数据项名 | 索引或键 | 备注 |
| 1 | accountNumber | TEXT | 30 | 商家账号 | 键 | 商家的账号 |
| 2 | name | TEXT | 20 | 商家名 |  | 商家的名字 |
| 3 | creditValue | INT | 100 | 信誉值 |  | 商家的信誉值 |
| 4 | phoneNumber | TEXT | 20 | 联系方式 |  | 商家的联系方式 |
| 5 | passWord | TEXT | 10 | 密码 |  | 商家的密码 |

表格7：数据库设计信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 表名称 | 设计者 | 审核者 | 完成日期 |
| 会员基本信息表 | 王震 | 李子涵 | 2016/11/20 |
| 消费记录表 | 李子涵 | 刘铮 | 2016/11/20 |
| 订单信息表 | 刘铮 | 韩世依 | 2016/11/20 |
| 购物车信息表 | 韩世依 | 王震 | 2016/11/20 |
| 商品信息表 | 王震 | 李子涵 | 2016/11/20 |
| 商家信息表 | 李子涵 | 刘铮 | 2016/11/20 |

### 物理结构设计

数据库的文件物理结构为：二叉树结构

### 数据结构与程序的关系

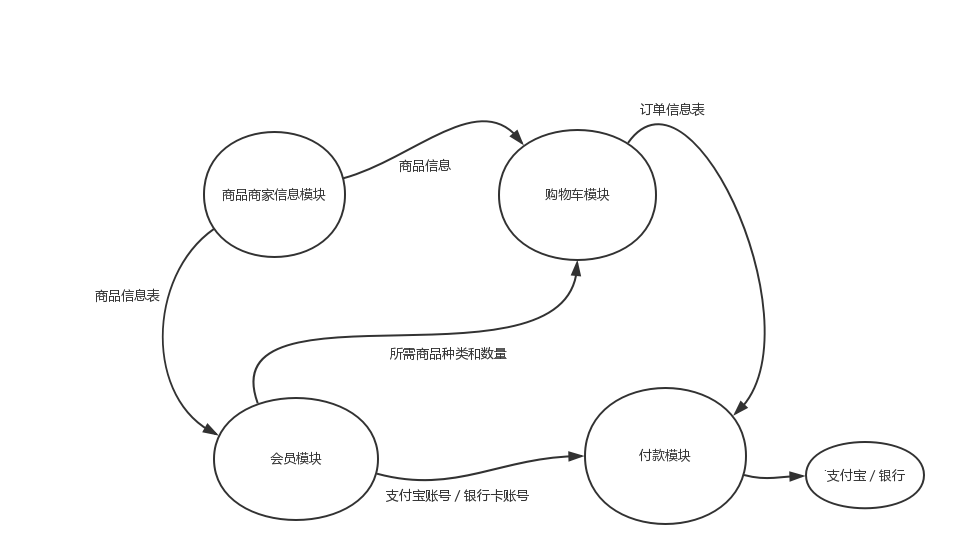
数据结构为关系型数据库，所以，在程序中可以通过标准的SQL语句与数据结构进行交互，交互过程中采用通用的数据库访问接口。为了保持良好的程序架构，对数据库访问采用DAO设计模式实现，提高维护性及扩展性。

表格8：数据结构与程序的关系表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 登录程序 | 注册程序 | 购物车 | 查询商品 | 购买商品 |
| 会员基本信息表 | √ | √ |  |  |  |
| 消费记录表 |  |  | √ | √ | √ |
| 订单信息表 |  |  | √ |  | √ |
| 购物车信息表 |  | √ |  | √ |  |
| 商品信息表 |  |  | √ | √ | √ |
| 商家信息表 | √ | √ |  | √ | √ |

# 详细设计

## 整体dfd图



以上为整体的dfd图，表述了各模块顶层的信息交换，下面，将从各个模块开始分别介绍设计模式。

## 购物车模块

### 模块概述

用户将想购买的商品放入购物车，购物车模块独立于商品浏览和用户信息，作为一个单独设置的模块。

### 接口说明

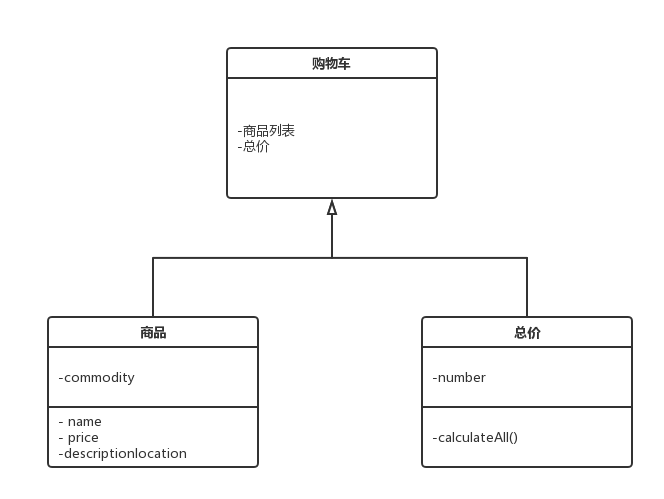
输入：用户想购买的商品种类和商品数量，商品的信息介绍和商品主页链接

输出：订单的信息（包含总价和订单时间等信息，订单信息表设计已在见表格3。

### 内部设计说明

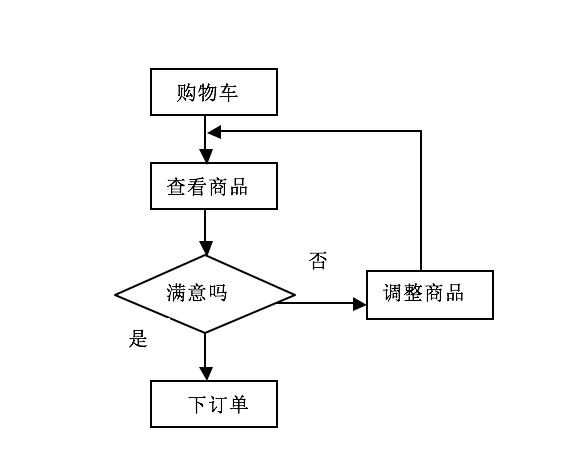
#### 静态结构

类图



#### 动态结构

顺序图



外部与内部功能流程图



## 商品商家信息模块

### 模块概述

商品和商家信息在同一个模块，用户在浏览商品的时候，也可以参考商户的信息（包括信誉度，历史销量等）

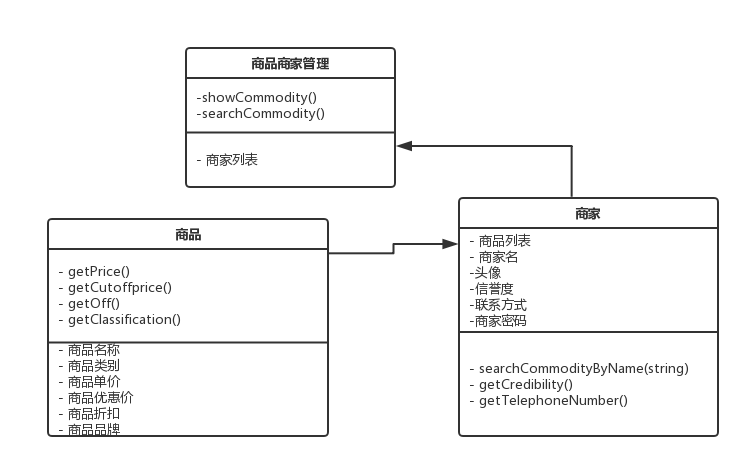
### 接口说明

输入：由于商品和商家输入同一模块，而所有商品信息由商家设计提供，所以没有外部输入信息。

输出：商品商家模块会将商品的信息以列表的形式展示给会员，在购物车中，商品商家模块会将用户选中的商品的信息和部分图片以固定的格式传过去。

### 内部结构设计

静态结构，以下是其类图



由于商品和商家的操作和会员密不可分，所以他们的动态结构将下会员的时序图一起展示。

## 会员模块

### 模块概述

会员模块拥有会员区别于普通用户的不同是能否有使用购物车，并且只有会员可以真正的购买商品，这个模块包括用户的注册和登陆等功能。

### 接口说明

输入：登陆时的用户名和密码

注册时的账户密码和各种用户信息

修改用户信息时的输入信息

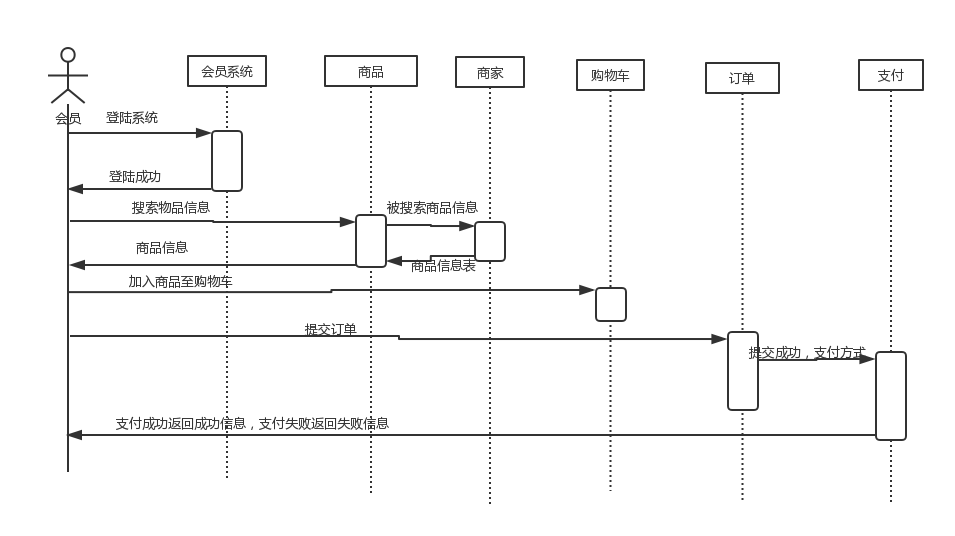
输出：查看用户信息的时候显示的用户信息

### 内部结构设计

#### 静态结构



#### 动态结构



## 付款模块

### 模块概述

付款模块，基本不暴露在外，一切由后台服务器负责，来保证用户信息和系统财产的安全。

### 接口说明

输入：

购物车中的商品种类和数量

输出：

购买成功提示信息或支付失败信息

### 内部结构设计

付款模块主要是在后台，并且没有固定的属性，顾没有静态结构，主要设计由动态结构展示。

动态结构

