

**Web编程技术课程设计报告**

题 目 旧货交流平台

专 业 软件工程

组 长 姬龙龙

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号 | 姓名 | 班级 | 分工 | 成绩 |
| 201521122082 | 姬龙龙 | 软件1511 | 项目架构及主要功能实现 |  |
| 201521122061 | 吴一楠 | 软件1511 | 单点登录系统的实现 |  |
| 201521122092 | 费洪玺 | 软件1513 | 数据库的设计 |  |

起止时间：2016年 月 日——2017年 月 日

1. 研究意义

淘淘二手交易商城是一个综合性的B2C平台，类似京东商城、天猫商城。会员可以在商城浏览商品、下订单，以及参加各种活动。

管理员、运营可以在平台后台管理系统中管理商品、订单、会员等。

客服可以在后台管理系统中处理用户的询问以及投诉。

1. 需求分析

后台管理系统：管理商品、订单、类目、商品规格属性、用户管理以及内容发布等功能。

前台系统：用户可以在前台系统中进行注册、登录、浏览商品、首页、下单等操作。

会员系统：用户可以在该系统中查询已下的订单、收藏的商品、我的优惠券、团购等信息。

订单系统：提供下单、查询订单、修改订单状态、定时处理订单。

搜索系统：提供商品的搜索功能。

单点登录系统：为多个系统之间提供用户登录凭证以及查询登录用户的信息。

1. 总体设计（概要设计）



分布式架构：

把系统按照模块拆分成多个子系统。

优点：

1. 把模块拆分，使用接口通信，降低模块之间的耦合度。
2. 把项目拆分成若干个子项目，不同的团队负责不同的子项目。
3. 增加功能时只需要再增加一个子项目，调用其他系统的接口就可以。

可以灵活的进行分布式部署。

缺点：

系统之间交互需要使用远程通信，接口开发增加工作量。

### 技术选型（主要技术）

* Spring、SpringMVC、Mybatis
* JSP、JSTL、jQuery、jQuery plugin、EasyUI、KindEditor（富文本编辑器）、CSS+DIV
* Redis（缓存服务器）
* Solr（搜索）
* httpclient（调用系统服务）
* Mysql
* Nginx（web服务器）
* Nginx静态技术（Linux的Crond定时任务）
* RabbitMQ消息队列技术

## 开发工具和环境

Eclipse 4.5.0(Luna)，自带maven插件，需要手工安装svn插件。

Maven 3.3.3（开发工具自带）

Tomcat 7.0.53（Maven Tomcat Plugin）

JDK 1.7

Mysql 5.6

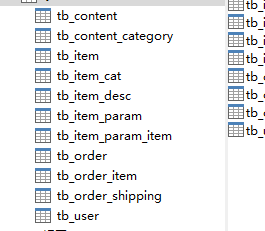
Nginx 1.8.0

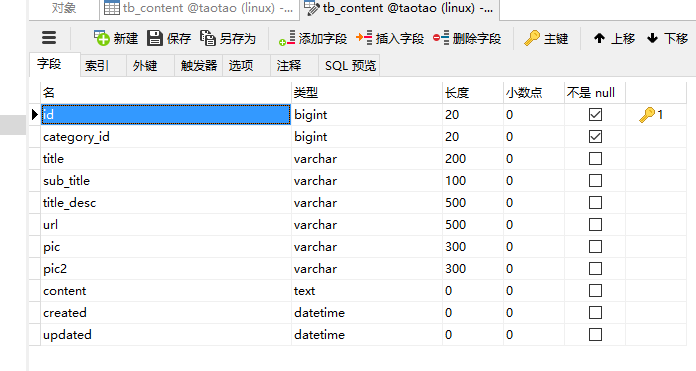
Redis 3.0.0

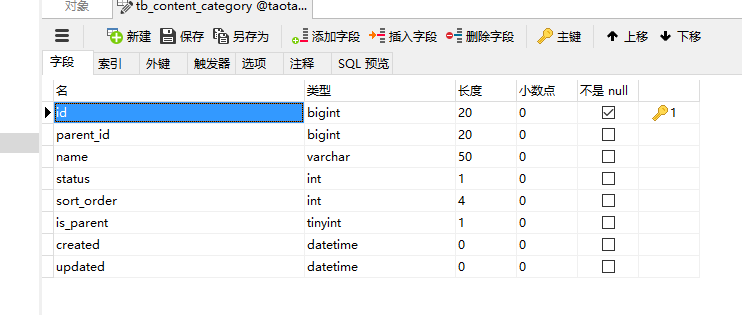
Win10 操作系统

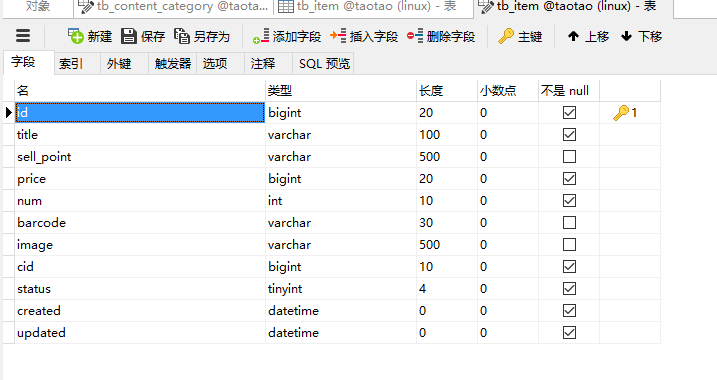
Linux-centOS

1. 数据库设计









其他表格见数据库文件。

1. 主要功能与实现代码分析

注册用户成功后插入数据库

**public** TaotaoResult createUser(TbUser user) {

user.setUpdated(**new** Date());

user.setCreated(**new** Date());

//md5加密

user.setPassword(DigestUtils.*md5DigestAsHex*(user.getPassword().getBytes()));

userMapper.insert(user);

**return** TaotaoResult.*ok*();

}

登录验证的实现代码

**public** TaotaoResult userLogin(String username, String password,

HttpServletRequest request, HttpServletResponse response) {

TbUserExample example = **new** TbUserExample();

Criteria criteria = example.createCriteria();

criteria.andUsernameEqualTo(username);

List<TbUser> list = userMapper.selectByExample(example);

//如果没有此用户名

**if** (**null** == list || list.size() == 0) {

**return** TaotaoResult.*build*(400, "用户名或密码错误");

}

TbUser user = list.get(0);

//比对密码

**if** (!DigestUtils.*md5DigestAsHex*(password.getBytes()).equals(user.getPassword())) {

**return** TaotaoResult.*build*(400, "用户名或密码错误");

}

//生成token

String token = UUID.*randomUUID*().toString();

//保存用户之前，把用户对象中的密码清空。

user.setPassword(**null**);

//把用户信息写入redis

jedisClient.set(REDIS\_USER\_SESSION\_KEY + ":" + token, JsonUtils.*objectToJson*(user));

//设置session的过期时间

jedisClient.expire(REDIS\_USER\_SESSION\_KEY + ":" + token, SSO\_SESSION\_EXPIRE);

//添加写cookie的逻辑，cookie的有效期是关闭浏览器就失效。

CookieUtils.*setCookie*(request, response, "TT\_TOKEN", token);

//返回token

**return** TaotaoResult.*ok*(token);

1. 测试与改进

测试

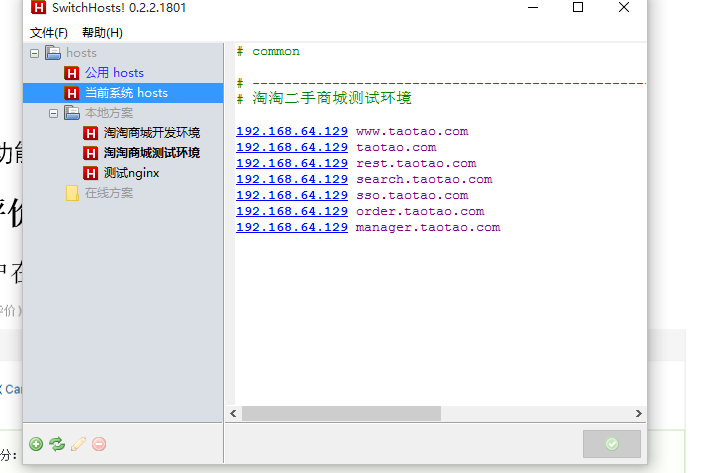
## 域名的分配

一级域名：taotao.com

二级域名：[www.taotao.com](http://www.taotao.com)

|  |  |
| --- | --- |
| 系统 | 域名 |
| Taotao-manager | manager.taotao.com |
| Taotao-rest | rest.taotao.com |
| Taotao-portal | www.taotao.com |
| Taotao-search | search.taotao.com |
| Taotao-sso | sso.taotao.com |
| Taotao-order | order.taotao.com |

所有的域名应该指向反向代理服务器。



服务器规划：

Taotao-manager、图片服务器 、Redis集群 、Solr集群、Mysql :一台服务器

Taotao-portal：1台

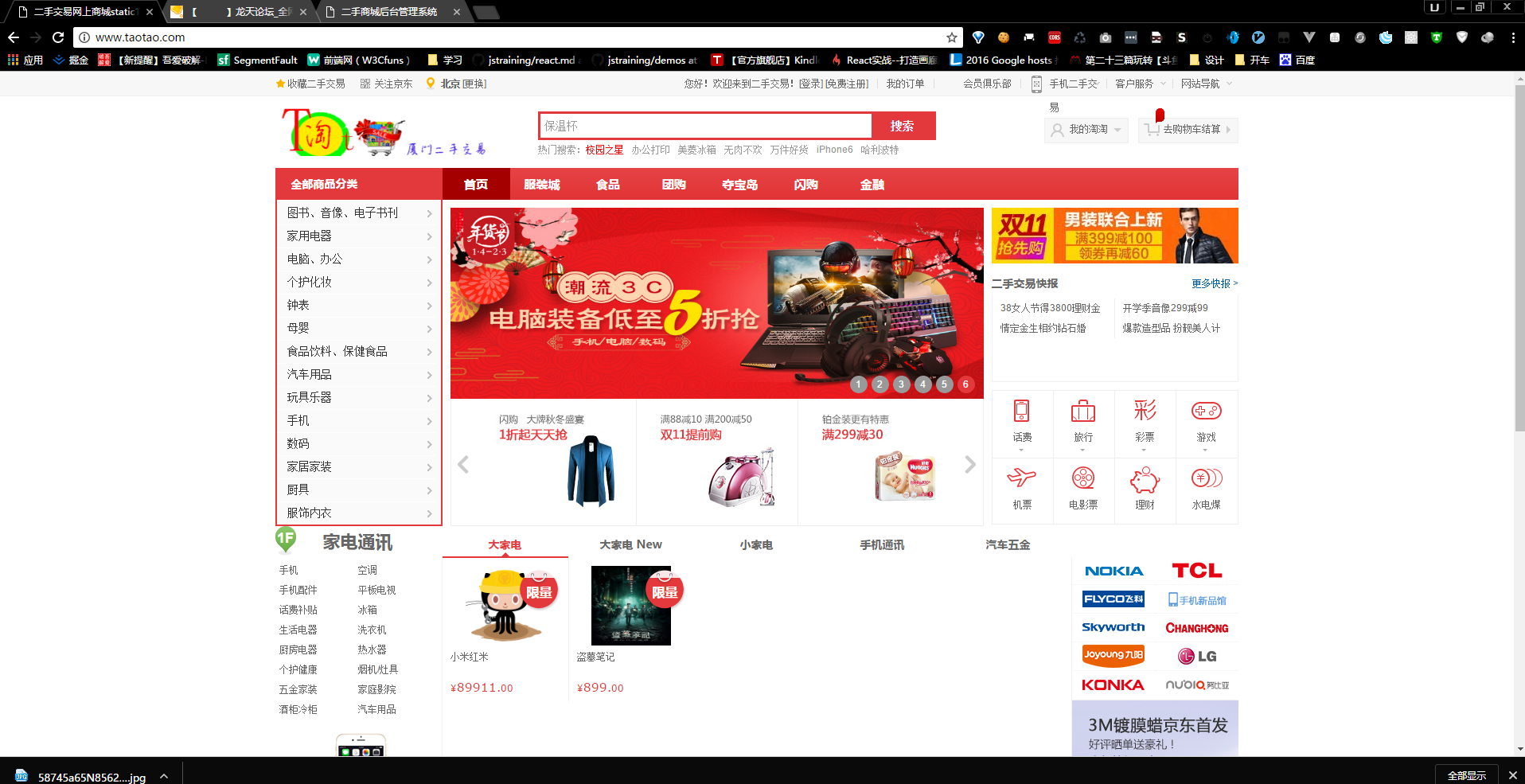
Taotao-rest、taotao-search、taotao-sso、taotao-order：1台（4个tomcat实例）

Nginx：1台

共需要4台虚拟机。

加上本机的RabbitMQ环境。

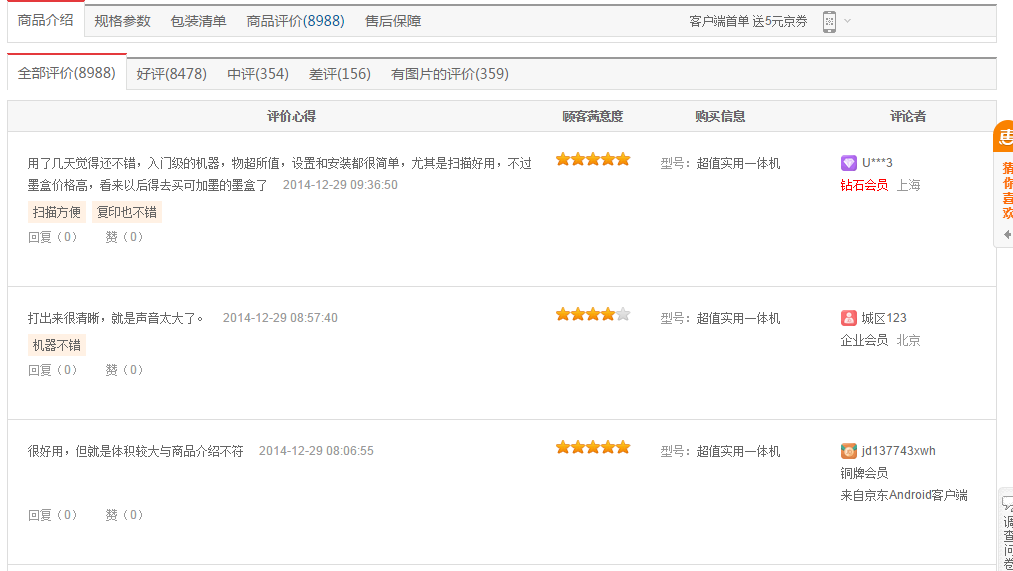
# 系统部署完成



待新增的功能

# 商品评价

用户在购物完成后可以对该商品进行评价；  


在商品详情页中显示可以商品评价。  


评分规则：4~5星：好评，2~3星：中评，1星：差评。

1. 结语

使用maven构建工程。Maven的继承、聚合、依赖管理。

mybatis逆向工程。

RESTFUL接口规则实现

后台管理系统搭建，前台技术使用EasyUI框架。了解easyUI的dategrid的使用方法。

图片服务器的搭建。使用ftp+http配合，实现图片服务。ftp服务使用vsftpd实现，http服务使用nginx实现

理解分布式系统架构思想。

Jsonp解决ajax跨域调用的问题

CMS系统的实现思路

Httpclient的使用方法

首页大广告位的实现

redis的使用方法及redis集群的搭建

系统中添加缓存逻辑

缓存的同步方法

solr服务在linux环境的搭建

使用solr实现搜索功能。

索引库中商品的同步

商品详情页面内容异步加载

规格参数按需加载

商品详情页面添加缓存及缓存同步

服务接口的开发

在分布式环境中使用redis实现session共享

使用cookie在多个系统中共享。

拦截器的使用方法

使用cookie保存购物车信息

创建订单系统。发布服务，支持post请求提交json格式的数据。

在用户提交订单之前，检查用户的登录状态（拦截器）

nginx的虚拟机的配置

Nginx的反向代理

Nginx的负载均衡

Nginx的高可用

Mysql数据库在linux环境的安装

使用maven进行系统的热部署。

Nginx静态技术（Linux的Crond定时任务）

RabbitMQ消息队列技术

8.附录