

**网站购物系统**

**设计说明书**

|  |  |  |
| --- | --- | --- |
| **学号** | **姓名** | **本文档中主要承担的工作内容** |
| 14061209 | 嵇文麒 | 主稿 工作比例：0.4 |
| 14061210 | 孙镜涵 | 审理修正 工作比例：0.3 |
| 14061211 | 泰阳 | 审理修正 工作比例：0.3 |
|  |  |  |

北京航空航天大学

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 |
| V1.0 | 2016.11.29 | 嵇文麒 | 泰阳，孙镜涵 | 初版 |
| V2.0 | 2016.12.18 | 嵇文麒 | 泰阳 | 针对互评当中部分问题的修改 |
| V3.0 | 2016.12.27 | 嵇文麒 | 孙镜涵 | 针对各种评价当中问题以及系统初步实现的大型更新 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

1. 范围 1
   1. 标识 1
   2. 系统概述 1
   3. 文档概述 1
2. 引用文档 2
3. 需求概述 2
4. 体系结构设计 3
   1. 总体结构 3
5. 系统架构 3
6. 软件体系架构 3
7. 具体模块划分图 4
   1. 功能分配 4
   2. 关键问题及解决方案 5
      1. 数据安全 5
      2. 大数据量的存储及访问 5
8. 接口设计 5
   1. 用户界面设计 5
   2. 外部接口设计 8
   3. 内部接口设计 8
9. 数据结构设计 9
   1. 公共数据结构设计 9
   2. 数据库设计 9
10. 详细设计 13
    1. 各模块类及接口 13
    2. UML类图 19
11. 范围
    1. 标识

文档标识号：A2016-00-02-00

文档标题：网站购物系统需求规格说明书

文档版本号：2.0

项目/产品中文全称：网站购物系统

项目/产品英文全称：website shopping system

项目/产品英文简称：WSS\_1.0

项目/产品编码：201610090018

* 1. 系统概述

本系统的一般特性是能够实现用户在本网站上购物的全过程。本系统面向用户，依照现实中网上购物的基本流程，满足用户的浏览商品，下单订货，退单，查询自己订单，提交评价的需求，依照用户浏览，添加到购物车，提交订单，支付，收货，评价的流程来进行设计。用户需注册，本系统负责保管用户信息，提供商品目录。

开发方：计算机学院软件工程实践小组。

支持机构：北京航空航天大学软件工程研究所。

相关文档：《项目开发计划》、《软件设计说明书》、《软件测试计划》

* 1. 文档概述

本文档用于详细地阐述软件的设计，各种功能的实现。

包括软件的总体结构，功能分配，外部、内部接口设计，UI，还有数据的存储方式，数据库结构的设计。

1. 引用文档

[1] 王珊、萨师煊.《数据库系统概论》[M].高等教育出版社，2006,12

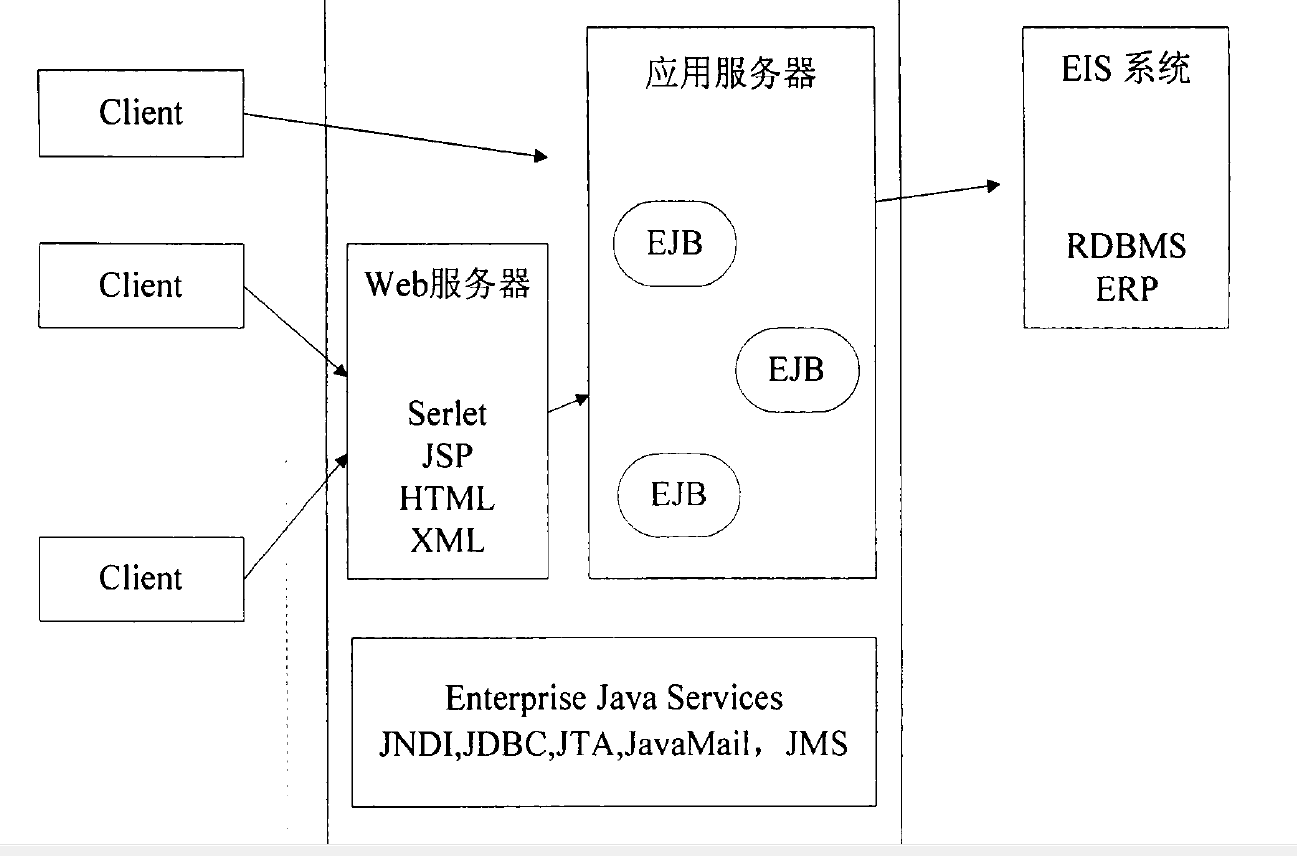
[2] 《Team17需求规格说明书\_SRS\_1.3》 2016,11

1. 需求概述
2. 对于已注册用户进入个人系统的功能，用户提供用户名和密码，完成登录操作。
3. 对于未注册用户要创建新账号。用户提供用户名，密码，密保问题信息，完成账号注册操作
4. 为用户界面提供相关商品信息
5. 处理用户对商品下订单的操作
6. 用户可将商品收藏入购物车，还包括对购物车商品的添加删除操作
7. 用户通过第三方支付平台完成支付操作
8. 用户对处于已完成状态的订单进行评价操作

新增行为需求用户密码找回需求，用户由于某种原因需要重置密码，系统会自动生成进入密保问题页面的链接并发送到用户邮箱中，然后从数据库中找到用户的密保问题。密保问题页面有一定有效时间，在此期间用户账号状态进入一个特别的安全状态，不接受其他的密码更改申请。

1. 体系结构设计
   1. 总体结构
2. 系统架构

系统架构，我们采用一个如下图所示的三层架构。三层架构在涉及到网络的开发中很常见，它将系统分割为自下而上的数据访问层（DAL）、业务逻辑层（BLL）和表示层（UI）。表现层即为展现给用户的UI界面，业务逻辑层是软件的核心了，包括处理用户请求和访问操作数据层等，数据访问层很直观了就是直接管理数据库的。在我们的项目开发中，我们将下图所画的三级结构表示为客户端层，模块层和数据层。

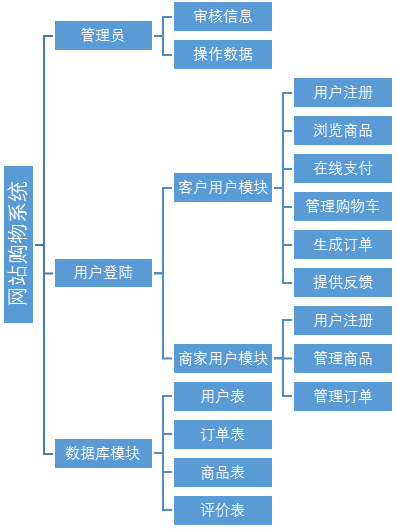


1. 软件体系架构

在上述系统架构的基础上，我们的软件体系结构应当说是一个以数据为中心的层次体系结构，所有商家、客户、商品、订单等等信息数据的存储驻留在这个体系结构的中心。客户端层次的需求通过中间模块层进行数据业务的逻辑处理来访问数据。这样仅由中间模块层完成数据的操作和提供应用软件的服务，可以提升用户数据的安全性，而当我们需要更新或者优化某一层结构的实现方式时，这种体系结构让我们可以减少或者不需要对其他层次结构的修改。比如我们要更新数据的存储方式，这时用户的客户端因为本就不直接访问数据层，所以并不需要修改。当然也存在有需要级联修改的隐患在。并且在我们的开发分工中，每个人可以更多的只关注我们某一层的开发以增加开发效率。

1. 具体模块划分图

具体的模块划分及关系如下，用户服务模块组全面针对用户的功能，包括买家客户和商家。通过部分工具提供相关信息的查询。系统管理模块组针对管理员的功能，提供日常管理和系统维护的工作，通过数据库的接口来操作用户数据。



* 1. 功能分配

用户登陆，注册，商品浏览，订单操作，购物车操作，支付操作和评价系统的功能全部放在在客户端层次，这些将会作为UI中的链接，按钮，图片及其事件来实现。

供应商的登陆，注册，上新申请，订单处理，评价反馈和移权操作功能同样在UI层实现，因为本网站购物系统的特点是对于系统管理员提供高权限，而对于商品供应商不提供自主销售权限，以实现具有鲜明特色的高集权度的网站购物系统，从而保障产品的高质量。

以上功能请求会发送给管理员模块，只有管理员具有向内层逻辑处理操作模块发送请求的权限。

* 1. 关键问题及解决方案
     1. 数据安全

由于我们要求用户在使用过程中提供了很多如付款信息，地址等真实信息，所以保证数据库的安全是很必要的。我们限制了用户所在的客户端层对数据库的访问，大部分的查询及修改数据的请求要通过管理员来完成。

* + 1. 大数据量的存储及访问

对于可能的大量的客户和商家的访问导致服务器崩溃以及高并发的修改操作数据库会导致的一些不可预测的问题。我们计划定期对服务器进行检查维护以及对数据库进行备份。

1. 接口设计
   1. 用户界面设计

本系统用户界面设计主要针对三类外部实体，用户，供货商，管理员。

1、针对用户的典型的用户界面大致如下：

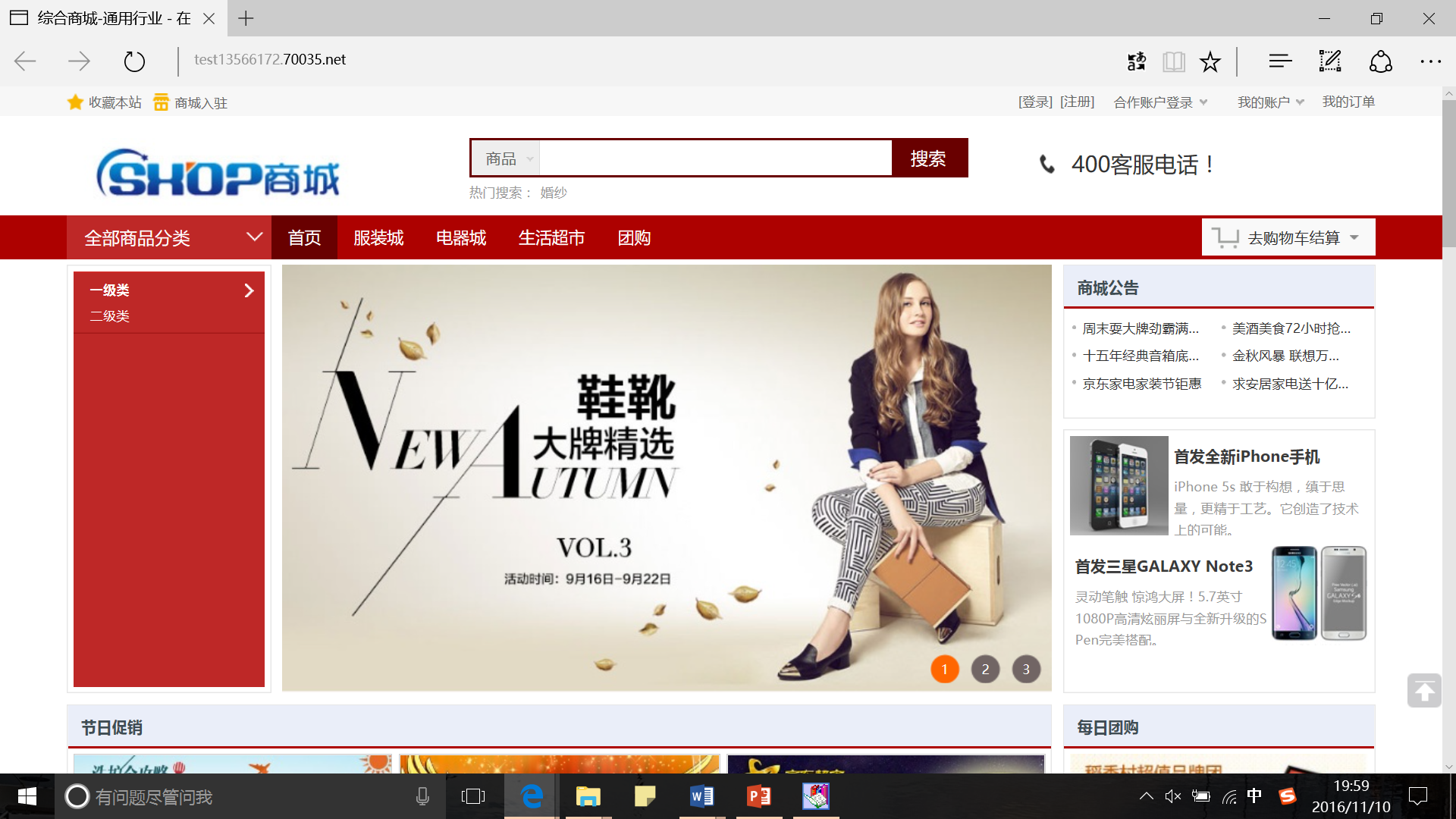


Figure 7

用户界面会为用户提供登录链接和注册链接，用户点击后进入相应页面以满足用户相应的需求。平台会在用户界面首页推送一些广告以及根据用户购买行为特征来推送满足用户口味的特定商品。同时用户界面首页会提供搜索引擎，供用户搜索想要的特定商品。用户界面还有供用户处理自身订单以及购物车的链接，用户可以在相应的页面对自己的订单进行相应的操作。在用户完成一次购买过程之后，平台会为用户提供评价以及支付的渠道。在支付行为发生的过程中，平台会从友商支付平台中获得并提供需要的信息。

2、针对供货商的用户界面大致如下：

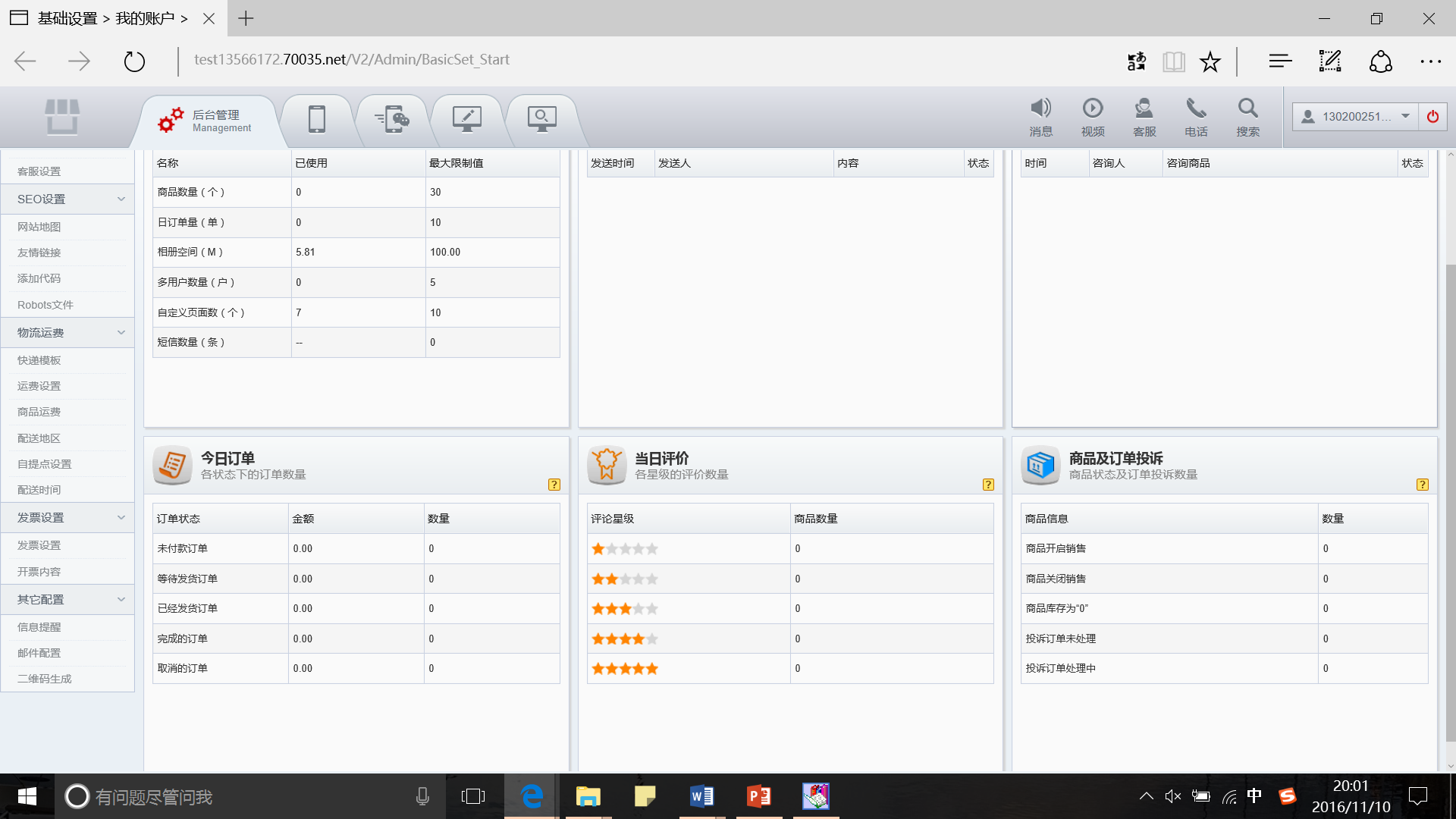


Figure 8

供货商的用户界面同样有登录，注册的接口，在注册的过程中，供货商需要提供详细的审核信息。供货商可以查看自身接受到的订单信息，以及对相应的订单的处理接口。与之对应的，供货商可以查看自身的评价，以及对评价的反馈渠道。在供货商收款过程中，需要与友商支付平台交换信息。

3、系统管理员的典型界面如下：

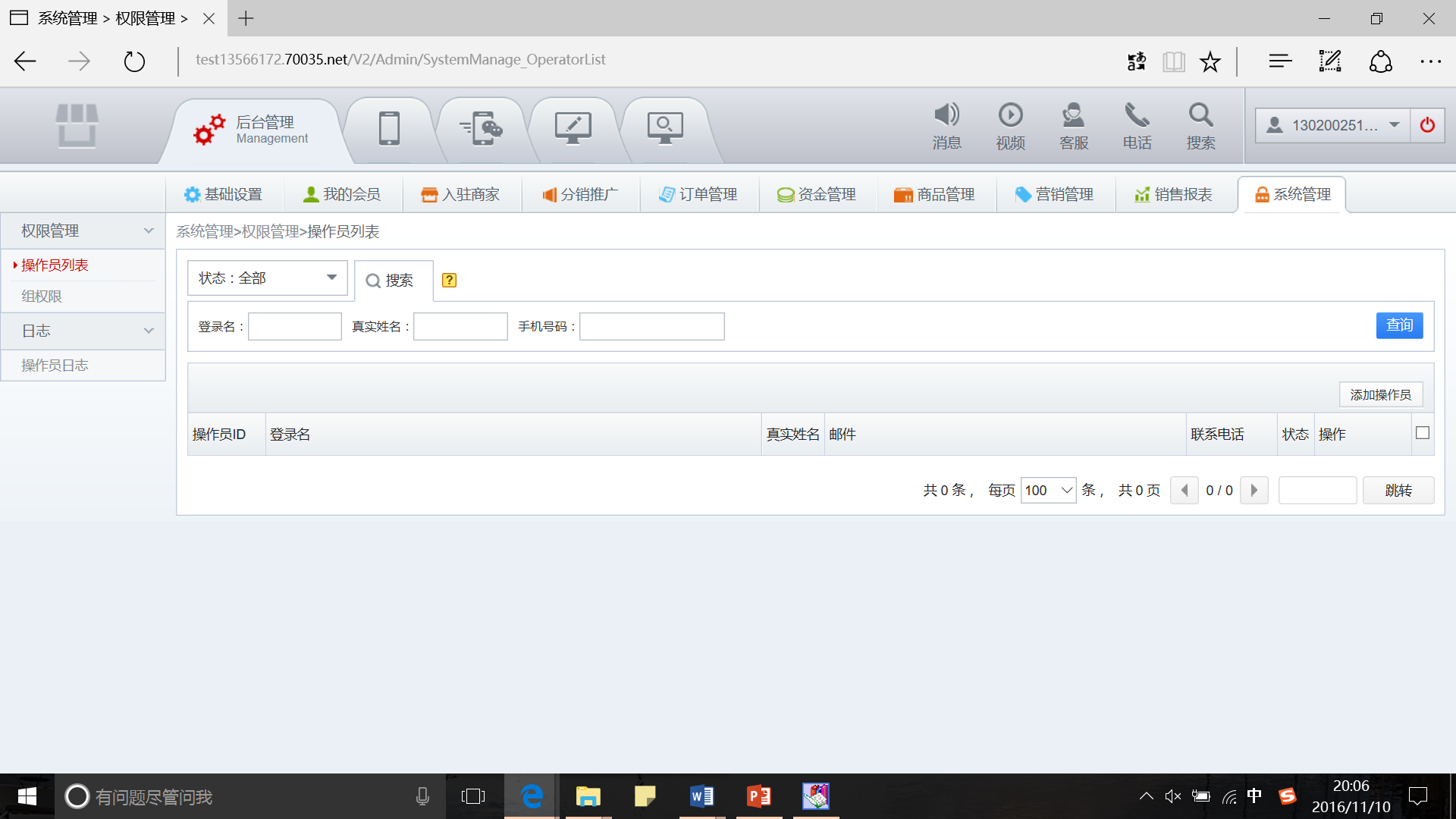


Figure 9

针对管理员的用户界面，需提供供货商，用户，商品，交易额等经相应权限管理后的信息。需给管理员提供供货商信息审核的接口，对相应商品进行审查的渠道，以及纠纷处理的必要渠道。对于管理托管供货商的管理员，还要提供供货商的相应信息。

* 1. 外部接口设计

硬件外部接口：

服务器：阿里云租赁足量资源服务器。

工作站：Pentium 4以上微机；内存1G；硬盘至少80以上；网络适配器10MＢ/100MＢ自适应。

网络交换机、网络适配器等。

软件外部接口：

数据库：Oracle

操作系统：Ubuntu及window10

逻辑层开发平台：J2EE平台

开发框架：EJB框架

* 1. 内部接口设计

用户使用前端与用户后台管理之间存在信息管理接口。

管理员前端与后台管理之间存在信息接口。

供货商前端与后台管理之间的信息接口。

用户后台与商家后台间关于评价，订单等的信息接口。

1. 添加用户注册信息的公共接口

Public int add\_User(int userid, String password, String nickname, String usertype, String Email)

1. 管理员操作用户列表的接口

Public int opr\_Data\_usr(enum opr\_type, int userid, String info)

/\*\*opr\_type

insert -> 添加用户数据

delete -> 删除用户数据

search -> 查询用户数据

\*/

1. 管理员修改用户信息的接口

Public int update\_Data\_usr()

1. 管理员修改评论信息的接口

Public int opr\_Data\_feedback(enum opr\_type, int id, String info)

/\*\*opr\_type

delete -> 删除

get -> 查询

\*/

1. 商家用户修改商品信息接口

Public int update\_Goods(String goodsid, float price)

1. 数据结构设计
   1. 公共数据结构设计

公共数据结构预计如下：

1. 数据库访问实体。

2、系统版本号等系统必须常量。

3、系统实现过程中必须的全局变量。

* 1. 数据库设计

对程序当中所采用的数据库相关的名称和标识符、在数据库当中的位置、定义、度量单位、格式和值域、敏感程度、数据项名、缩写词和代码，包括用于数据库的规格说明等进行定义。建议给出数据库表结构。

说明：数据库表结构可以参考如下，字段值域不够可以自己增加。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库的表结构 | 表名称 | 设计者 | 审核者 | 完成日期 |
| 商品 | 梁琛奇 | 泰阳 | 2016.11.10 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 备注 | 缩写词 |
| 1 | 商品号 | 整形 | 正整数 | 主码 |  |  |
| 2 | 名称 | 字符串 | 长度小于20的字符串 | 唯一 |  |  |
| 3 | 售价 | 实数 | 正实数 |  |  |  |
| 4 | 供应商工号 | 整形 | 正整数 | 外码 |  |  |
| 5 | 批号 | 整形 | 正整数 |  |  |  |
| 6 | 尺码 | 整形 | 正整数 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库的表结构 | 表名称 | 设计者 | 审核者 | 完成日期 |
| 供应商 | 梁琛奇 | 泰阳 | 2016.11.10 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 备注 | 缩写词 |
| 1 | 供应商工号 | 整形 | 正整数 | 主码 |  |  |
| 2 | 名称 | 字符串 | 长度小于20的字符串 | 唯一 |  |  |
| 3 | 联系人 | 字符串 | 长度小于20的字符串 |  |  |  |
| 4 | 地址 | 字符串 | 长度小于200的字符串 |  |  |  |
| 5 | 电话 | 整形 | 7位或11位正整数 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库的表结构 | 表名称 | 设计者 | 审核者 | 完成日期 |
| 客户 | 梁琛奇 | 泰阳 | 2016.11.10 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 备注 | 缩写词 |
| 1 | 客户号 | 整形 | 正整数 | 主码 |  |  |
| 2 | 名称 | 字符串 | 长度小于20的字符串 |  |  |  |
| 3 | 邮编 | 字符串 | 长度为7位的字符串 |  |  |  |
| 4 | 地址 | 字符串 | 长度小于200的字符串 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库的表结构 | 表名称 | 设计者 | 审核者 | 完成日期 |
| 系统管理销售员 | 梁琛奇 | 泰阳 | 2016.11.10 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 备注 | 缩写词 |
| 1 | 工号 | 整形 | 正整数 | 主码 |  |  |
| 2 | 姓名 | 字符串 | 长度小于20的字符串 | 唯一 |  |  |
| 3 | 部门号 | 整形 | 正整数 |  |  |  |
| 4 | 地址 | 字符串 | 长度小于200的字符串 |  |  |  |
| 5 | 电话 | 整形 | 7位或11位正整数 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库的表结构 | 表名称 | 设计者 | 审核者 | 完成日期 |
| 订货单 | 泰阳 | 泰阳 | 2016.11.30 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 备注 | 缩写词 |
| 1 | 进货号 | 整形 | 正整数 |  |  |  |
| 2 | 进货量 | 整形 | 正整数 |  |  |  |
| 3 | 货物编码 | 整形 | 正整数 | 外码 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库的表结构 | 表名称 | 设计者 | 审核者 | 完成日期 |
| 订单 | 泰阳 | 泰阳 | 2016.11.30 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段代号** | **名称** | **类型** | **值域** | **索引或键** | **备注** | **缩写词** |
| 1 | 订单号 | 整形 | 正整数 | 主码 |  |  |
| 2 | 销售工号 | 整形 | 正整数 |  |  |  |
| 3 | 客户编号 | 整形 | 正整数 |  |  |  |
| 4 | 订单日期 | 日期 |  |  |  |  |
| 5 | 发货方式 | 字符串 |  |  |  |  |
| 6 | 付款方式 | 字符串 |  |  |  |  |
| 7 | 总金额 | 整形 |  |  |  |  |

1. 详细设计
   1. 各模块类及接口

* 用户类

public class User implements java.io.Serializable {

private Integer id;

private String userName;

private String userPass;

private String nickName;

private String realName;

private String sex;

private Date birthday;

private String idCard;

private String cellPhone;

private String email;

private String city;

private Set orderses = new HashSet(0);

private Set addresses = new HashSet(0);

private Set feedbacks = new HashSet(0);

定义基本客户的使用操作和信息存取，基本模块包括

-用户名（包括id，昵称，用户名）

-密码

-相关信息（包括性别，生日，真实姓名，身份证，手机号码，邮件地址，所在城市）

-购物车列表（即订单列表）

-地址列表

-评价列表

-注册信息方法

-重置信息方法

-付款方法

-登陆方法

-管理评论方法

-管理地址方法

-管理购物车方法

* 商家类

public class Merchant implements java.io.Serializable {

private Integer id;

private String userName;

private String userPass;

private String shopName;

private String theMan;

private String phoneNum;

private String email;

private String addr;

private Set orderses = new HashSet(0);

private Set Goods = new HashSet(0);

定义商家的使用操作和信息存取，基本模块包括

-用户名（包括id，商家名，用户名）

-密码

-相关信息（包括负责人，邮件地址，真实地址，联系电话）

-订单列表

-商品列表

-注册信息方法

-重置信息方法

-登陆方法

-管理商品方法

-管理订单方法

-收款方法

* 管理猿类

public class Admin implements java.io.Serializable {

private Integer id;

private String adminName;

private String adminPass;

private Integer cellPhone;

private String email;

定义管理员账号（管理员账号不可被注册，我们预先直接设定管理员账号）及其可使用的高权限操作

-用户名

-密码

-登录

-存用户及上架商品数据（ALL）方法

-查询用户数据方法

-修改用户数据方法

* 商品类

public class Goods implements java.io.Serializable {

private Integer id;

private Goodstype goodstype;

private String name;

private Integer price;

private String status;

private Integer discount;

private Date uptime;

private Integer countNumber;

private Integer sellNumer;

private Date produceDate;

private Integer timeLength;

private String describe;

private String image;

private Set orderdetailses = new HashSet(0);

-商品名（包括id，名称）

-商品类型

-商品价格

-商品上架时间

-商品状态

-商品折扣信息

-商品数量统计（包括卖出数量和总数）

-商品图片

-商品描述

-管理商品信息方法

* 订单类

private Integer id;

private User user;

private Address address;

private Date time;

private Integer money;

private String sendMode;

private String status;

private Set orderdetailses = new HashSet(0);

-用户名

-地址

-时间（接受订单操作的系统时间）

-金额

-配送方式

-状态

-订单细节（包括评论信息，商品信息）

-管理订单状态的方法

* 评价反馈类

public class Feedback implements java.io.Serializable {

private Integer id;

private User user;

private Orderdetails orderdetails;

private String message;

private Date time;

评价操作针对订单，由用户执行

-评价id

-用户

-订单信息

-评价信息

-时间（最后一次接受修改信息操作的系统时间）

-修改评价信息的方法

-删除本条评价的方法

* 消息类

public class Notice implements java.io.Serializable {

private Integer id;

private String describe;

private Date time;

private String status;

-消息编号

-消息内容

-时间（消息进入服务器数据库的系统时间）

-消息状态

-修改消息内容的方法

-管理消息状态的方法

* 数据库接口

package dao;

public interface BaseDAO

public interface GoodsDAO extends BaseDAO

public interface UserDAO extends BaseDAO

* Web服务接口

package service;

public interface BaseService

public interface AdminService extends BaseService

public interface UserService extends BaseService

public interface MerchantService extends BaseService

* 1. UML类图

