**一、需求分析**

1.系统概述：本系统通过管理员、快递员、注册用户的互动实现鲜花、水果等商品的在线预订和配送功能。管理员可以对在线货架进行管理，实现商品上架和下架的功能；管理员可以对订单进行管理，实现订单配货、订单发货、订单配送完成确认、订单分类查询的功能；管理员可以对快递员进行管理，实现快递员管理（添加、修改、删除快递员）功能。快递员可以对配送进行管理和确认，实现在线收货接单、签收签收确认、历史送货查询功能。注册的用户可以浏览货架、查询商品、购物车管理、购物车结算、在线支付、取消订单、订单确认、历史订单查询的功能。匿名用户可以在线浏览货架产品的功能。

2.系统的角色：管理员、快递员、注册用户、匿名用户；

3.各角色功能

(1)**管理员**的功能：登录、商品上架、商品下架、订单配货、订单发货、订单配送完成确认、订单分类查询、快递员管理；

(2)**快递员**的功能：登录、在线收货接单、订单签收确认、历史送货查询；

(3)**注册用户**的功能：登录、浏览货架、浏览货架、查询商品、购物车管理、购物车结算、在线支付、取消订单、订单确认、历史订单查询；

(4)**匿名用户**的功能：在线浏览货架产品。

4.数据要求

**实体1**：管理员=登录名+密码+联系电话+是否有效；

**实体2**：注册用户=登录名+密码+联系电话+是否有效；

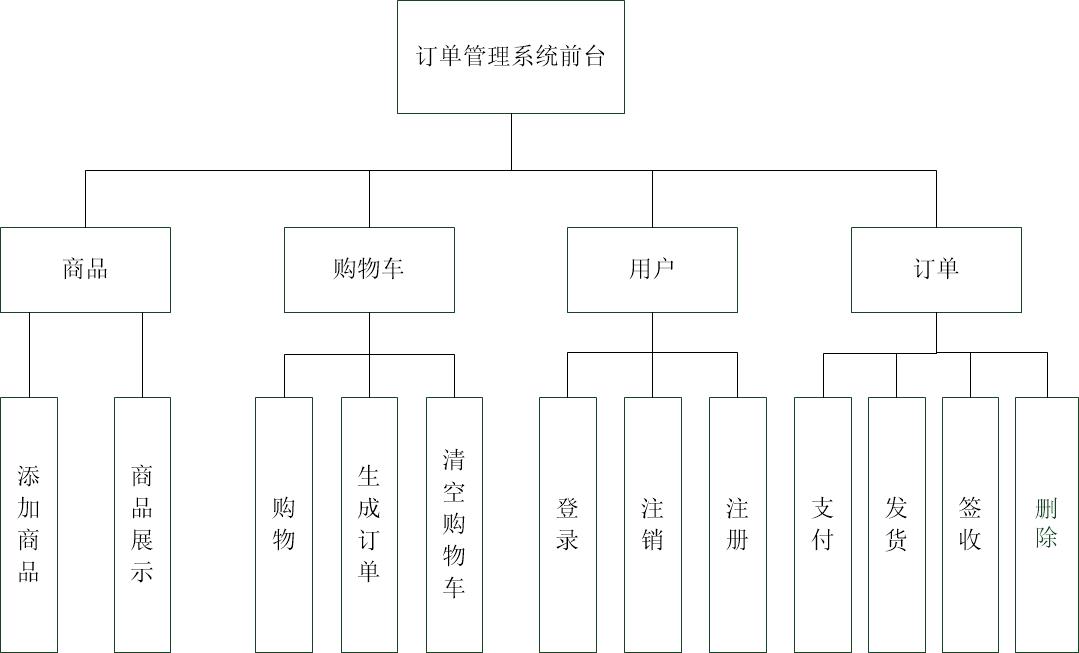
**实体3**：快递员=登录名+密码+联系电话+是否有效；

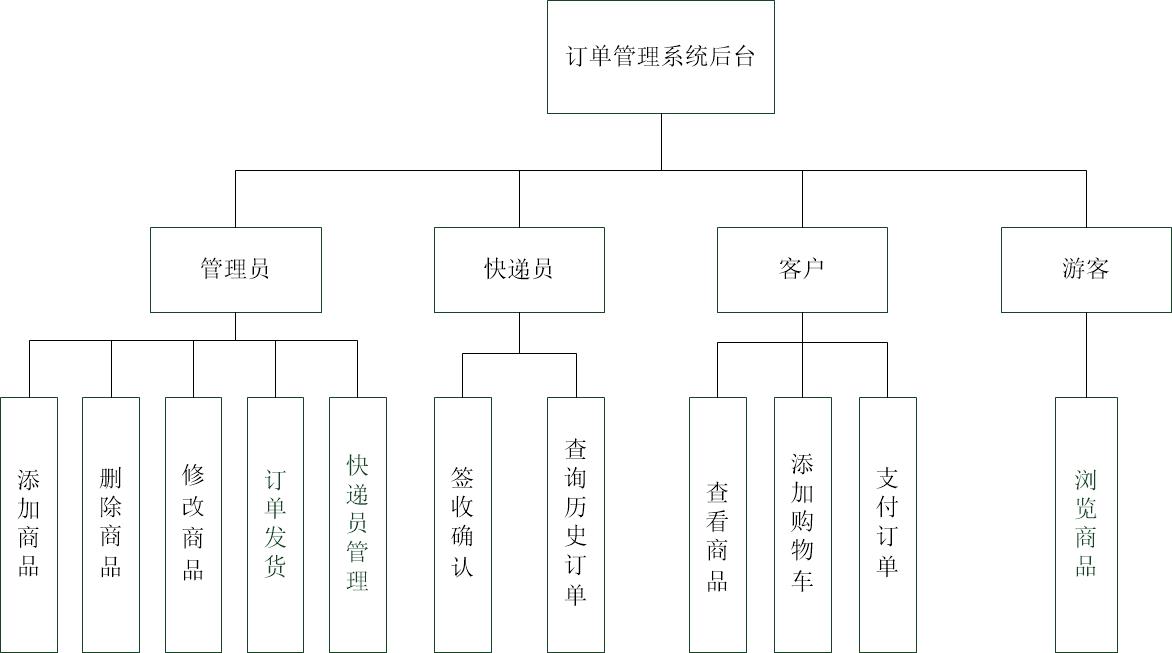
**实体4：**商品=商品名称+样品图片+价格+供货开始时间+供货结束时间；

**实体5：**在线货架={商品}；

**实体6：**订单=订货人ID+配货管理员ID+快递员ID+收货人姓名+收货人地址+收货人电话+预期送货时间+{商品}+支付金额+订单建立时间+订单发货时间+订单签收时间+订单配送完成时间；

**二、总体设计**





**三、系统设计**

create database OrderManager;

用户：

create user OrderManager identified by 'OrderManager';

授权：

grant all on OrderManager.\* to OrderManager;

use OrderManager;

用户表

create table users (

id int primary key auto\_increment，

username varchar(100)，

password varchar(100)，

telnum varchar(40)，

role\_id int， #0---管理员，1-----快递员，2------普通用户

foreign key(role\_id) references roles(id)，

cash int

);

角色表

create table roles (

id int primary key，

roleName varchar(20)

);

功能表：

create table function (

id int primary key，

funcName varchar(100)，

path varchar(255)

);

权限表：

create table privilege (

role\_id int，

func\_id int，

primary key(role\_id， func\_id)，

foreign key(role\_id) references roles(id)，

foreign key(func\_id) references function(id)

);

商品表：

create table products(

id varchar(100) primary key ，

name varchar(40)，

imgurl varchar(100)，

price double，

pnum int ，

startTime date，

endTime date，

admin\_id int，

foreign key(admin\_id) references users(id)

);

订单表

create table orders(

id varchar(100) primary key，

money double，

address varchar(255)，

payState int，

admin\_id int，

courier\_id int，

orderTime datetime，

sendTime datetime，

signTime datetime，

sendFlag int， #1---已发货，0---未发货

signFlag int， #1---已签收，0---未签收

user\_id int ，

foreign key(user\_id) references users(id)

);

订单——商品映射表

create table orderitem(

order\_id varchar(100),

product\_id varchar(100),

buynum int,

primary key(order\_id，product\_id) #联合主键，两列的值加在一起作为这张表的主键使用

);

javaee经典三层架构+工厂类实现解耦

包结构:

com.order.web-----------------------------------------控制层

.service-------------------------------业务逻辑层

.dao-----------------------------------数据库访问层

.util------------------------------------存放工具类

.domian------------------------------数据实体(持久化对象)

.factory------------------------工厂类，实现业务逻辑与数据控制解耦

.exception

.test

.filter---------------------------过滤器

.listener-----------------------监听器，主要用于购物车

导入第三方jar包

\*junit

\*JSTL

beanutils

mysql驱动

c3p0

dbutils

commns-fileupload

配置文件

c3p0-config.xml

config.properties

**四、主要功能模块设计**

1.用户注册

index.jsp -- 如果用户没有登录，则提示“欢迎光临游客 注册 登录”；如果用户已经登录，提示“欢迎回来xxx，注销”

regist.jsp -- 提供注册表单 (验证码/js实现表单校验)

regist.do -- 校验验证码 封装数据校验数据 调用Service注册用户 重定向到主页

UserService -- 注册用户

UserDao -- 添加客户方法

2.用户登录(记住用户名，7天内自动登陆)

index.jsp -- 如果用户没有登录，则提示 欢迎光临游客 注册 登录 如果用户已经登录 提示 欢迎回来xxx，注销

login.jsp -- 提供用户登录表单输入用户名密码

login.do -- 1.获取用户名密码，2.调用Service中根据用户名密码查找用户的方法 3.登录用户重定向到主页

UserService -- 提供根据用户名 密码查找用户的方法 调用dao中对应方法

UserDao -- 提供根据用户名密码查找用户的方法

3.用户注销

Kill session

4.添加商品(文件上传)

admin.jsp -- 提供添加商品功能

addProd.jsp -- 提供添加商品的表单，这个表单应该是文件上传的表单，其中允许上传商品图片

addProd.do -- 实现文件上传，将商品的图片上传到服务器中.并且向数据库的商品表中增加一条记录

Service -- 添加商品

Dao -- 添加商品信息的方法

5.商品列表

admin.jsp， customer.jsp -- 提供商品列表

pageProdList.do -- 调用Service中查询所有商品的方法，查到后存入request域带到页面展示（分页）

prodList.jsp -- 从request域中拿出所有的商品做展示

6..加入购物车

在商品详细信息页面中点击加入购物车，将商品加入购物车

设置监听器，在Session创建时，就将cartmap加入到session中

addOrder.do -- 根据id查找商品，存入购物车，如果购物车中还不存在这个商品，则存入，数量为1，如果已经存在则在原有数量上+1

cart.jsp -- 遍历购物车map，遍历cartmap显示当前用户所有的购物车信息

7.删除购物车

在购物车页面，中点击删除时，触发

delCart.do，根据id找到要删除的商品后，从购物车map中删除

8.修改购买数量

在购物车页面，修改购物数量时触发，利用js控制输入的数字必须正整数

changeCar.do，根据id找到要删除的商品后，修改购物车中商品的数量

9.清空购物车

找到购物车map，清空map

10.生成订单

在购物车中，当购物完成后，用户点击“生成订单”

访问一个addOrder.jsp -- 列出订单的基本信息，要求用户输入收货地址和支付方式

addOrder.do -- 创建Order对象设置基本值，其中Money需要在后台根据购物车实时计算出来 调用OrderService中生成订单的方法生成订单，清空购物车，回到主页

OrderService -- 中生成订单的方法 在订单表中插入一条记录 在订单项表中插入记录保存此订单和商品之间的关系 从商品表中的库存数量中扣除购买数量 需要进行事务管理

OrderDao -- 增加订单的方法 增加订单项的方法

prodDao -- 增加扣除商品数量的方法

11.订单查询

在主页中提供订单查询

OrderListServlet（客户使用） 获取当前客户的用户号，调用Service中查询指定用户订单的方法，查询出这个用户所有订单，存入request域，带到页面显示

OrderManaServlet（管理员使用）

OrderSendServlet（管理员使用，正在派送的订单）

OrderSentServlet（管理员使用，历史派送的订单）

OrderService 提供根据用户id查询订单的方法

OrderDao 提供根据用户id查询订单的方法

orderList.jsp 展示所有订单

12.订单删除

在订单列表，对于未支付的订单提供订单删除

DelOrderServlet 获取订单id 调用service中根据订单id删除订单的方法，删除后再重新回到订单列表页面

OrderService 提供根据订单id删除订单的方法 将订单相关商品的库存数量加回去 删除订单项表中的记录 删除订单记录 -- 事务控制

OrderDao中提供根据订单编号删除订单项和 订单的方法

ProdDao中提供修改库存数量的方法

13.在线支付

在订单列表页面，对于未支付的订单提供在线支付

pay.jsp 页面中展示订单和支付金额，若账户余额不足则提示

PayServlet 扣除客户相应的金额，同时增加卖家相等的金额

16.权限控制