# 详细设计-附表

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| **2017//11/30** | V0.1 | 针对总体设计HIPO结构的第一次详细设计 | 吴安之 |
| **2017/12/24** | V1.2 | 根据实际情况更新 | 吴安之 |

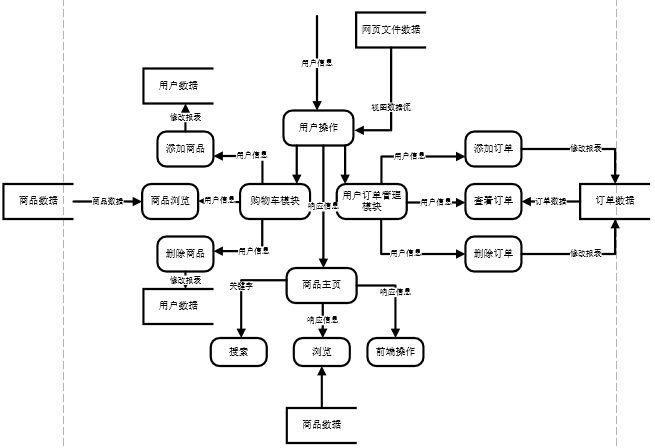
模块名：响应事件处理模块

模块主要功能概述：该模块主要面向用户，实现登录用户在网上的浏览和操作。

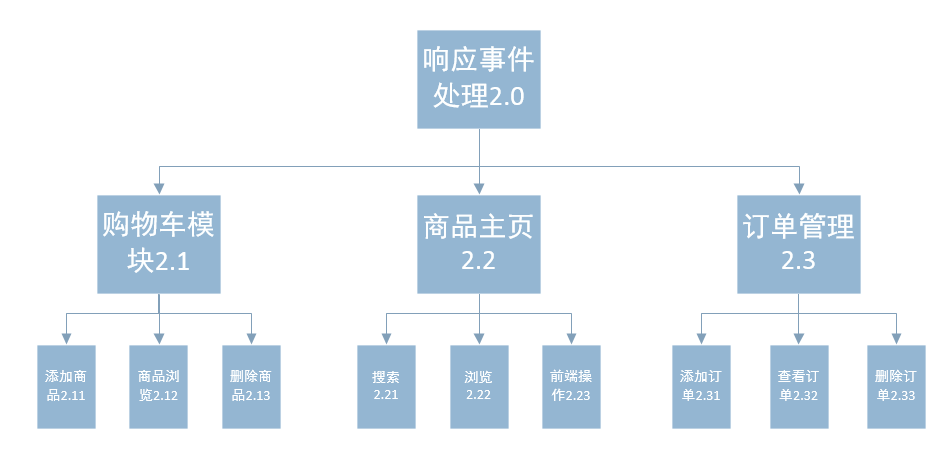
负责人：吴安之

## 子模块HIPO结构：

### 精化模块数据流图：



### 输出HIPO结构：



### IPO表：

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号： | 2.1.1 | 功能名称： | 添加商品 |
| 功能描述：  对用户提供将商品添加进购物车功能 | | | |
| 输入：  选中的商品 | | | |
| 处理：  1、验证是否登录  2、调用事件操作“添加商品至购物车” | | | |
| 输出：  是否添加成功 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号： | 2.1.2 | 功能名称： | 商品浏览 |
| 功能描述：  对用户提供查看购物车内商品功能 | | | |
| 输入：  查看指令 | | | |
| 处理：  1、验证是否登录  2、调用事件操作“查看购物车” | | | |
| 输出：  购物车内商品 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号： | 2.1.3 | 功能名称： | 删除商品 |
| 功能描述：  对用户提供删除购物车内商品功能 | | | |
| 输入：  购物车内待删除的商品 | | | |
| 处理：  调用事件操作“删除购物车内商品” | | | |
| 输出：  是否删除成功 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号： | 2.2.1 | 功能名称： | 搜索 |
| 功能描述：  对用户提供商品查询功能 | | | |
| 输入：  关键字 | | | |
| 处理：  根据关键字信息对商品进行检索 | | | |
| 输出：  检索出的商品信息 | | | |
| 功能编号： | 2.2.2 | 功能名称： | 浏览 |
| 功能描述：  对用户提供网页栏目的浏览检索 | | | |
| 输入：  关键字 | | | |
| 处理：  根据关键字信息对网页相关功能模块检索 ,提供分页分类的浏览辅助 | | | |
| 输出：  检索出相应的功能模块 | | | |

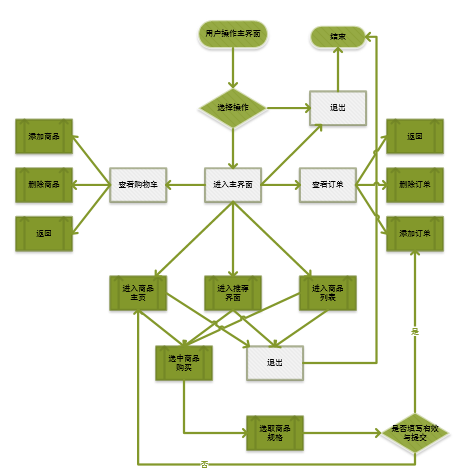
|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号： | 2.2.3 | 功能名称： | 前端操作 |
| 功能描述：  对用户提供网页帮助与可使用前端行为 | | | |
| 输入：  响应事件 | | | |
| 处理：  js事件处理 | | | |
| 输出：  处理结果 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号： | 2.3.1 | 功能名称： | 添加订单 |
| 功能描述：  对用户提供提交订单功能 | | | |
| 输入：  1、选中的商品  2、收货地址 | | | |
| 处理：  调用事件操作“提交订单” | | | |
| 输出：  返回订单信息 | | | |

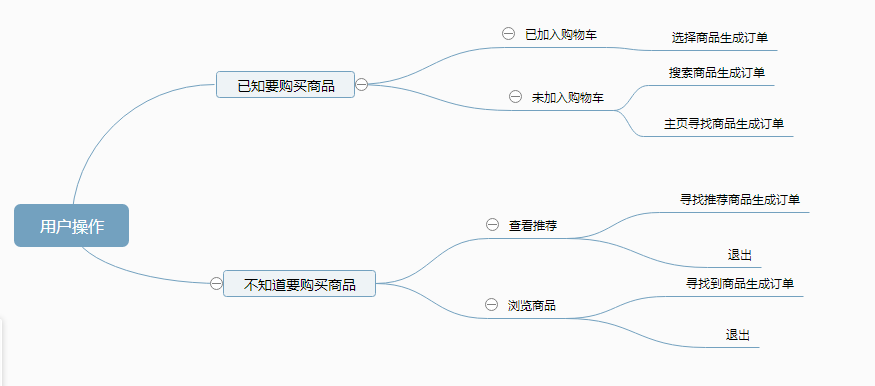
|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号： | 2.3.2 | 功能名称： | 修改订单 |
| 功能描述：  对用户提供修改订单功能 | | | |
| 输入：  要修改地址 | | | |
| 处理：  1、验证订单状态，状态为未处理才能更改  2、调用事件操作“修改” | | | |
| 输出：  是否修改成功 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号： | 2.3.3 | 功能名称： | 删除订单 |
| 功能描述：  对管理员提供删除订单功能 | | | |
| 输入：  1、待删除订单  2、删除指令 | | | |
| 处理：  1、调用事件操作“删除订单” | | | |
| 输出：  是否删除成功 | | | |

## 程序流程图：



## 判定树：



## PDL:

1.1：

procedure Cart\_add interface<http requset> is

begin

if(user is logged in)

if (user want to add Goods to cart)

then

Add an instance of a commodity to a shopping

cart

which based on this user info.

insert above add\_message into MongoDB

(

insert

(productid,productname,size,prive,color,amount)

to table carts

which is in customers

)

return add Goods succeed

else

defult

else

User\_operation

end

1.2：

procedure Cart\_browse interface<http request> is

begin

if(user is logged in)

if (user want to browse Goods from cart)

then

instance of a commodity info=all\_cart(Good\_item)

select above Goods\_message from MongoDB

(

ref carts=select carts from customers where

customers.register\_info=register\_info

select \* from carts from customer.carts

Commodity display

)

Commodity display

else

defult

end

1.3:

procedure Cart\_delete interface<http request> is

begin

if(user is logged in)

if (user want to delete Goods from cart)

then

instance of a commodity info=select(Good\_item)

delete above delete\_message from MongoDB()

（

delete Goods

(productid,productname,size,price,color,amount)

from carts from customer=reg\_customer

return delect Goods succeed

else

defult

end

2.1:

procedure Goods\_search interface<http request> is

begin

do

kw=input("Commodity keywords")

select above kw from MongoDB

select \* from product\_info from products

(

Good\_item.add

(ID,name,class,size,color,storge,agent\_price,

sale\_priv e,raw\_price,time\_import.time\_end\_sale)

)

instance of a commodity info=all\_rs(Good\_item)

Commodity display

while(search)

end

2.2

procedure Goods\_browse interface<http request> is

begin

do

instance of a commodity info=all\_rs(Good)

Good=

products.product\_info.id

+products.product\_info.name

+products.product\_info.class

+products.product\_info.size

+products.product\_info.color

+products.product\_info.storage

+products.product\_info.agent\_price

+products.product\_info.sale\_price

+products.product\_info.raw\_price

+products.product\_info.time\_import

+products.product\_info.time\_end\_sale

Commodity display

while(browse)

end

2.3

procedure User\_operation interface<http request> is

begin

if(user want to know the operation)

Click the "help" button

(

get help info

)

end

3.1

procedure Order\_Submit interface<http request> is

begin

Good\_info=select(Good\_item)

Good=

products.product\_info.id

+products.product\_info.name

+products.product\_info.class

+products.product\_info.size

+products.product\_info.color

+products.product\_info.storage

+products.product\_info.agent\_price

+products.product\_info.sale\_price

+products.product\_info.raw\_price

+products.product\_info.time\_import

+products.product\_info.time\_end\_sale

Harvest address=select(harvest\_adress)

harvest\_adress=

customer.addInfo.address

which the user has filled out

Order\_message=Good\_info + Harvest address

if(user want to submit the order)

insert above Order\_message into MongoDB

(

insert

(orderID,productid,addressid,amount,payment,time)

to

orders

)

return Submit\_Order succeed

end

3.2

procedure Order\_Modify interface<http request> is

begin

Harvest address=select(harvest\_adress)

harvest\_adress=

customer.addInfo.address

modify Harvest address=input

(modify\_Harvest\_address)

which the user has filled out

if(user want to modify the order)

update above Modify\_Order\_info into MongoDB

（

Update addressID from orders from customer where ID = reg\_ID

）

return Modify\_Order succeed

end

3.3

procedure Order\_delete interface<http request> is

begin

if (user want to delete Order from cart)

then

instance of a Order info=select(Order\_item)

Order\_item=

orderID

+productID

+addressID

+amount

+payment

+time

delete above delete\_Order from MongoDB

(

Delete orders(productIDmaddressID，amount,payment,time) from customer

)

return delect Goods succeed

else

defult

end

Main

while(1)

switch(user\_operation)

case dddtocart:Cart\_add

case browsecart:Cart\_browse

case deletefromcart:Cart\_delete

case searchgoods:Goods\_search

case operation:User\_operation

case submit\_order:Order\_Submit

case modify\_order:Order\_Modify

case delete\_order:Order\_delete

defult:Cart\_browse