解决方案说明书

Solution Model Description

**编号**：**TMP-SMD**

**版本 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| 作者： | 杨涛 | 日期： | 2015.4.10 |
| 审批： |  | 日期： |  |

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 变更说明 | 作者 |
| 2015.4.10 | 1.0 | 创建 | 杨涛 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1． 简介 3](#_Toc416688798)

[1.1 内容 4](#_Toc416688799)

[1.2 适用对象 4](#_Toc416688800)

[1.3 参考文档 4](#_Toc416688801)

[2． 系统整体架构 4](#_Toc416688802)

[3． 资源层解决方案 6](#_Toc416688803)

[4． 业务层解决方案 8](#_Toc416688804)

[5． 表示层解决方案 10](#_Toc416688805)

## 简介

内 容编写此说明书，实现基于node的POS机交易系统，重点实现购物网站的用户选购商品，提交订单，查询订单，取消订单，订单支付等功能。

### 内容

本系统实现基于node的POS机交易。

1. 用户往购物车添加商品；
2. 用户在购物车增减已有商品的数量；
3. 用户提交订单；
4. 用户查询订单；
5. 用户取消已经生成但未完成订单。

### 适用对象

本性用户为：电商购物平台

### 参考文档

1. 《node.js开发指南》
2. 《node与express开发》
3. 《node.js实战》
4. 《mongodb权威指南》
5. 《javascript权威指南》

## 系统整体架构



## 资源层解决方案

**3.1 数据库设计的概念模型**

数据访问层负责对业务层提供数据操作。业务层或者通用层中所有的对象都通过数据访问层的对象访问数据库。数据访问层中的类是按业务对象来组织的，每个业务对象中包含的数据可能存在不同的几种数据表中，它由数据访问类统一组织成一个概念中的对象，它相当于是一个面向对象的数据库层，负责映射面向对象与关系数据库间的关系。对数据库的所有操作均由存储过程完成，数据层只是在前台调用后台的存储过程。

**3.2 资源层组件**

资源层组件是完成数据库处理的组件，例如数据源的管理、执行数据库的查询、CRUD。

**3.2.1 资源层类图**



**3.2.2 类详细设计**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 类名称 | User | | | | 所属包名称 |  | | |
| 类变量定义(先Public->protected->private) | | | | | | | | |
| 名称 | | | 类型 | | | | 描述 | |
| User | | | User | | | | 用户 | |
| Userinfo | | | Userinfo | | | | 用户信息 | |
| 方法(先Public->protected->private) | | | | | | | | |
| 名称 | | 参数 | | 返回值 | | | | 简单描述 |
| UpdateUser | | User | | User | | | | 更改用户信息 |
| SearchUser | | User | | Datatable | | | | 查询用户信息 |
| DeleteUser | | User | | User | | | | 删除用户信息 |
| AddUser | |  | | User | | | | 添加用户信息 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类名称 | Goods | | 所属包名称 |  | |
| 类变量定义(先Public->protected->private) | | | | | |
| 名称 | | 类型 | | | 描述 |
| GoodsID | | long | | | 商品ID |
| GoodsName | | string | | | 商品名称 |
| Goodsprice | | float | | | 商品价格 |
| Goodstype | | string | | | 商品类别 |
| GoodsStock | | int | | | 商品库存 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类名称 | Order | | 所属包名称 |  | |
| 类变量定义(先Public->protected->private) | | | | | |
| 名称 | | 类型 | | | 描述 |
| OrderID | | long | | | 订单号 |
| UserID | | long | | | 用户编号 |
| OrderStage | | string | | | 订单状态 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类名称 | Shoppingcart | | 所属包名称 |  | |
| 类变量定义(先Public->protected->private) | | | | | |
| 名称 | | 类型 | | | 描述 |
| UserName | | String | | | 用户名 |
| GoodsName | | string | | | 商品名称 |
| GoodsAmount | | int | | | 商品数量 |
| Total | | float | | | 总额 |

## 业务层解决方案

4.1 业务层概述

业务层是专门处理软件业务需求的一层，处于数据库之上，服务层之下，完成一些列对Domain Object的 CRUD，作为一组微服务提供给服务层来组织在暴露给表现层。集中了所有业务处理的代码.；处理了应用程序的业务逻辑和业务校验.；管理事物 ；与其它层相互作用的接口

4.2 业务层类图



4.3 业务层类详细设计

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 类名称 | UserBiz | | | | | | 所属包名称 |  | | |
| 类变量定义(先Public->protected->private) | | | | | | | | | | |
| 名称 | | | | 类型 | | | | | 描述 | |
| **User** | | | | User | | | | | 用户信息对象 | |
| 方法(先Public->protected->private) | | | | | | | | | | |
| 名称 | | | 参数 | | | 返回值 | | | | 简单描述 |
| Login | | | String Username，string password | | | int | | | | 登录用户信息，病判断是否有效 |
| UpdatePassword | | | String Password | | | int | | | | 修改用户密码 |
| SubmitOrder | | | Order | | | Order | | | | 提交订单 |
| 类名称 | OrderBiz | | | | | | 所属包名称 |  | | |
| 类变量定义(先Public->protected->private) | | | | | | | | | | |
| 名称 | | | | 类型 | | | | | 描述 | |
| OrderID | | | | long | | | | | 订单号 | |
| UserID | | | | long | | | | | 用户编号 | |
| OrderStage | | | | string | | | | | 订单状态 | |
| 方法(先Public->protected->private) | | | | | | | | | | |
| 名称 | | 参数 | | | 返回值 | | | | | 简单描述 |
| submitOrder | | Goods | | | Orderdatatable | | | | | 提交订单 |
| SearchOrder | | int | | | Orderdatatable | | | | | 查询订单 |
| CancelOrder | | int | | | Orderdatatable | | | | | 取消订单 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 类名称 | ShoppingcartBiz | | | 所属包名称 |  | | |
| 类变量定义(先Public->protected->private) | | | | | | | |
| 名称 | | 类型 | | | | 描述 | |
| UserName | | String | | | | 用户名 | |
| GoodsName | | string | | | | 商品名称 | |
| GoodsAmount | | int | | | | 商品数量 | |
| Total | | float | | | | 总额 | |
| 方法(先Public->protected->private) | | | | | | | |
| 名称 | | 参数 | 返回值 | | | | 简单描述 |
| Balance | | Goods | int | | | | 结算 |
| AddGoods | |  | Shoppingcart | | | | 添加商品 |
| AddGoodsAmount | | Goods | int | | | | 增加商品数量 |
| ReduceGoodsAmount | | Goods | int | | | | 减少商品数量 |

## 表示层解决方案

**5.1 表示层概述**

表示层负责：管理用户的请求，做出相应的响应 ；提供一个controller，调用业务逻辑和其它上层接口 ；处理异常 ；简单地验证。

5.2 表示层类图

5.3 用例实现

5.3.1 用例一

（1）．用例名称及编号：

用例名称：添加商品到购物车

编号：101

（2）．用例概述：

用户往购物车添加数据库中存在且数量大 于0的商品

（3）．页面流程图

（4）．视图设计（ejs,express）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ejs页面说明** | | | | | | |
| 页面名称 | | AddGoods.ejs | | | | |
| 功能简述 | | 添加商品到购物车  显示上面mongodb传递过来的商品信息 | | | | |
| 进入的URL | | /shoponline/pos/views/AddGoods | | | 进入方式 | Respond |
| 数据来源定义 | | | | | | |
| 编号 | 参数名称 | | 来源 | 说明 | | |
| 1 | Goods | | mongodb | 获取商品的信息，与上表的输出相对应 | | |
| 页面外观 | | | | | | |
|  | | | | | | |

（5）．控制器（express）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **express类说明** | | | | | | |
| 类名称 | | | Pos.views.AddDoods | | | |
| 功能简述 | | | 用户添加商品到购物车  执行的操作如下：   1. 从session中得到商品的ID信息 2. 生成Goods对象，将商品信息赋值给Goods对象 3. 调用Goods的load方法，从数据库中得到对应商品的所有信息 4. 将这些商品信息放到Request的Attribute中，并通过respond方法转发给后面的ejs 5. 用户点击“加入购物车”按钮之后，通过connect-mongp由mongoose对数据库mongodb进行相关操作，数据库信息更新，将更新后的信息放到request的attribute中，并通过respond方法传递到后面的ejs, | | | |
| 输入定义 | | | | | | |
| # | | 名称 | | 类型 | | 说明 |
| 1 | | GoodsID | | long | | 商品的ID号，从用户点击“加入购物车”得到 |
| 处理流程 | | | | | | |
| 取出该商品的信息，传递给后面views中的ejs，商品加入购物车，数据通过connect-mongo传给mongodb,数据库信息更新，购物车信息更新 | | | | | | |
| 输出定义 | | | | | | |
| # | 名称 | | | 类型 | 说明 | |
| 1 | Goods | | | Goods | 将商品信息从数据库中取到Goods数据对象中，并将该Goods对象存放到Request中，传递给下面的ejs页面显示 | |
| 2 | AddGoods | | | Goods | 将商品加入购物车，数据通过connect-mongo传给mongodb,数据库信息更新，购物车信息更新 | |

5.3.2 用例二

（1）．用例名称及编号：

用例名称：在购物车中增减商品数量

编号：102

（2）．用例概述：

用户在购物车中增减已经存在于购物车中的商品的数量

（3）．页面流程图

（4）．视图设计（ejs,express）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ejs页面说明** | | | | | | |
| 页面名称 | | ChangeGoodsAmount.ejs | | | | |
| 功能简述 | | 用户增减商品数量  显示更改过的商品数量以及商品价格总额 | | | | |
| 进入的URL | | shoponline/pos/views/ChangeGoodsAmount | | | 进入方式 | respond |
| 数据来源定义 | | | | | | |
| 编号 | 参数名称 | | 来源 | 说明 | | |
| 1 | Goods，User | | Shoppingcart | 获得购物车的信息，与传来的数据相对应 | | |
| 页面外观 | | | | | | |
|  | | | | | | |

（5）．控制器（express类）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **express类说明** | | | | | | |
| 类名称 | | | Pos.views.ChangeGoodsAmount | | | |
| 功能简述 | | | 用户在购物车中增减已有商品数量  执行的操作如下：   1. 从session中得到用户的购物车信息 2. 生成ShoppingCart对象，将购物车信息赋值给ShoppingCart 对象 3. 调用ShoppingCart的load方法，从数据库中得到对应用户的ShoppingCart的所有商品信息 4. 将这些商品信息放到request的attribute中，并通过respond方法将request转发给后面的购物车ejs 5. 将增减商品数量的操作信息通过connect-mongo传给mongoose,mongoose对mongodb进行操作，mongodb数据更新,更新后的数据返回给购物车ejs 6. 更新购物车信息 | | | |
| 输入定义 | | | | | | |
| # | | 名称 | | 类型 | | 说明 |
| 1 | | GoodsAmount | | long | | 商品的数量，从用户的点击操作得到 |
| 处理流程 | | | | | | |
| 用户取得购物车信息，更改选中的商品的数量，数据传递到mongodb，更新mongodb数据，传递给后面的购物车ejs | | | | | | |
| 输出定义 | | | | | | |
| # | 名称 | | | 类型 | 说明 | |
| 1 | ShoppingCart | | | String | 将用户更改后的购物车信息从数据库中取到ShoppingCart数据对象中，并将该ShoppingCart对象存放到Request中，传递给下面的ejs页面显示 | |

5.3.3 用例三

（1）．用例名称及编号：

用例名称：提交订单

编号：103

（2）．用例概述：

用户提交订单

（3）．页面流程图

（4）．视图设计（ejs，express）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ejs页面说明** | | | | | | |
| 页面名称 | | SubmitOrder.ejs | | | | |
| 功能简述 | | 提交订单  显示上面Order传递过来的用户订单信息 | | | | |
| 进入的URL | | shoponline/pos/views/submitorder | | | 进入方式 | Respond |
| 数据来源定义 | | | | | | |
| 编号 | 参数名称 | | 来源 | 说明 | | |
| 1 | ShoppingCart | | 资源层 | 构成订单的购物车信息，与上表的输出相对应 | | |
| 页面外观 | | | | | | |
|  | | | | | | |

（5）．控制器（express类）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **express类说明** | | | | | | |
| 类名称 | | | Pos.views.submitorder | | | |
| 功能简述 | | | 提交订单  执行的操作如下：  1. 从session中得到用户的购物车信息  2. 生成Order对象，将购物车信息赋值给Order对象   1. 调用Order的load方法，从数据库中得到对应购物车的所有信息 2. 将这些购物车信息放到Request的Attribute中，并通过respond方法将Request转发给后面的ejs 3. 用户点击“提交订单”后，生成订单数据通过connect-mongo传到数据库mongodb， 4. 更新订单列表信息，更新购物车信息 | | | |
| 输入定义 | | | | | | |
| # | | 名称 | | 类型 | | 说明 |
| 1 | | ShoppingCart | | String | | 购物车信息，从用户ID关联中得到 |
| 处理流程 | | | | | | |
| 从ShoppingCart得到构成订单的购物车信息，生成订单页面，传递给后面的ejs，用户点击“提交订单”后，更新购物车信息，更新订单列表信息 | | | | | | |
| 输出定义 | | | | | | |
| # | 名称 | | | 类型 | 说明 | |
| 1 | Order | | | Order | 将购物车信息通过mongo-connect从mongodb数据库中取到Order数据对象中，并将该Order对象存放到Request中，传递给下面的ejs页面显示 | |

5.3.4 用例四

（1）．用例名称及编号：

用例名称：查询订单

编号：104

（2）．用例概述：

用户查询已经生成但未完成的订单或者历史订单

（3）．页面流程图

（4）．视图设计（ejs,express）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ejs页面说明** | | | | | | |
| 页面名称 | | searchorder.ejs | | | | |
| 功能简述 | | 用户查询订单  显示上面mongodb传递过来的订单信息 | | | | |
| 进入的URL | | shoponline/pos/SearchOrder.ejs | | | 进入方式 | Respond |
| 数据来源定义 | | | | | | |
| 编号 | 参数名称 | | 来源 | 说明 | | |
| 1 | Order | | 资源层 | 用户的订单信息，与上表的输出相对应 | | |
| 页面外观 | | | | | | |
|  | | | | | | |

（5）．控制器（express类）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **express类说明** | | | | | | |
| 类名称 | | | Pos.views.searchorder | | | |
| 功能简述 | | | 用户查询订单  执行的操作如下：   1. 从session中得到用户的OrderID信息 2. 生成Order对象，将ID信息赋值给Order对象 3. 调用Order的load方法，通过connect-mongo从数据库mongodb中得到对应ID的所有信息 4. 将这些信息放到Request的Attribute中，并通过respond方法将Request转发给后面的ejs | | | |
| 输入定义 | | | | | | |
| # | | 名称 | | 类型 | | 说明 |
| 1 | | OrderID | | long | | 用户的OrderID号，从用户的输入中得到 |
| 处理流程 | | | | | | |
| 取得用户的OrderID信息，将mongodb中对应ID数据传递给后面的ejs | | | | | | |
| 输出定义 | | | | | | |
| # | 名称 | | | 类型 | 说明 | |
| 1 | Order | | | Order | 将OrderID信息从数据库中取到Order数据对象中，并将该Order对象存放到Request中，传递给下面的ejs页面显示 | |

5.3.5 用例五

（1）．用例名称及编号：

用例名称：取消订单

编号：105

（2）．用例概述：

用户取消已经生成但未完成的订单

（3）．页面流程图

（4）．视图设计（ejs,express）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ejs页面说明** | | | | | | |
| 页面名称 | | cancelOrder.ejs | | | | |
| 功能简述 | | 用户取消订单  显示上面mongodb传递过来的用户订单信息 | | | | |
| 进入的URL | | shoponline/pos/cancelOrder | | | 进入方式 | Respond |
| 数据来源定义 | | | | | | |
| 编号 | 参数名称 | | 来源 | 说明 | | |
| 1 | OrderID | |  | 用户的订单信息，与上表的输出相对应 | | |
| 页面外观 | | | | | | |
|  | | | | | | |

（5）．控制器（ejs, express类）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **express类说明** | | | | | | |
| 类名称 | | | Pos.views.cancelorder | | | |
| 功能简述 | | | 用户取消订单  执行的操作如下：   1. 从session中得到用户的OrderID信息 2. 生成Order对象，将ID信息赋值给Order对象 3. 调用Order的load方法，从mongodb数据库中得到对应OrderID的所有信息 4. 将这些信息放到Request的Attribute中，并通过respond方法将Request转发给后面的ejs 5. 用户点击“取消订单”按钮后，通过connect-mongo由mongoose对mongodb进行相关操作，更新数据库信息，更新订单信息列表 | | | |
| 输入定义 | | | | | | |
| # | | 名称 | | 类型 | | 说明 |
| 1 | | OrderID | | long | | 用户的订单OrderID号，从用户的输入中得到 |
| 处理流程 | | | | | | |
| 取出OrderID相应订单信息，传递给后面的ejs， | | | | | | |
| 输出定义 | | | | | | |
| # | 名称 | | | 类型 | 说明 | |
| 1 | Order | | | Order | 将用户的OrderID信息从数据库中取到Order数据对象中，并将该Order对象存放到Request中，传递给下面的ejs页面显示 | |