**东软机密**

文件编号：D05-PDT07A

概要设计报告模板

版本：0.0.0-1.1.0

2019-3-15

东软集团股份有限公司 软件开发事业部

**(版权所有，翻版必究)**

文件修改控制

|  |  |  |  |
| --- | --- | --- | --- |
| **修改编号** | **版本** | **修改条款及内容** | **修改日期** |
| 1 | 0.0.0-1.0.0 | 新建 | 2019-3-15 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**东软机密**

文件编号：项目编号PDR顺序号 第1版

分册名称： 第1册/共1册

项目名称（项目编号）

概要设计报告

**(软件开发事业部)**

东软集团股份有限公司

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 总页数 |  | 正文 |  | 附录 |  | 生效日期 |  |
| 编制 |  | | | 批准 |  | | |

修改记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 变更控制报告  编号 | 更改条款及内容 | 更改人 | 审批人 | 更改日期 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

1 引言 6

1.1 目的 6

1.2 背景 6

1.3 术语表 6

1.4 参考资料 7

2 概要设计决策 7

2.1 设计原则 8

2.2 设计决策 8

2.2.1 决策1 8

2.2.2 决策2 8

2.2.3 决策3 9

2.2.4 决策4 9

2.2.5 决策5 9

3 总体设计 10

3.1 软件体系结构 10

3.2 系统技术构架 10

3.3 系统运行环境 11

4 子系统设计 11

5 接口说明 13

5.1 外部接口 13

5.2 内部接口 23

# 引言

随着经济的飞速发展，人们的物质需求逐渐提高，传统的商店渐渐无法满足人们轻松，方便购物的需要。近些年兴起的互联网技术，网上购物提供了可能。本项目是一个网上商城，为顾客提供购买服务，为商家提供发布商品信息等服务。本项目运用SSM框架作为基础进行设计，为项目提供低耦合、高效率。

## 目的

本报告主要阐述本项目京西商城的概要设计决策，总体设计，子系统设计和接口说明。制定本项目的设计概况，保证项目顺利进展。

## 背景

叙述该项软件开发的意图、应用目标、作用范围以及其他应向读者说明的有关该软件开发的背景材料。

本项目的开发意图旨在给顾客和商家提供一个网上商城的平台，将实现用户通过互联网完成商品购买的整个过程。顾客可以进行平台上查看商品种类，查看商品详细信息，将商品添加至购物车，下单生产订单等操作。

## 术语表

列出本报告中专门术语的定义、英文缩写词的原词组和意义、项目组内达成一致意见的专用词汇，同时要求继承全部的先前过程中定义过的词汇。

|  |  |  |
| --- | --- | --- |
| 术语 | 含义 | 备注 |
| SSM | Java后端框架，包括Spring, SpringMVC, Mybatis |  |
| Spring | Spring是一个开源框架，它由Rod Johnson创建。它是为了解决企业应用开发的复杂性而创建的 |  |
| SpringMVC | SpringMVC是Spring框架的一个模块,是基于mvc的webframework模块。 |  |
| Mybatis | MyBatis 是一款优秀的持久层框架,它支持定制化 SQL、存储过程以及高级映射 |  |
| Mysql | MySQL是一个关系型数据库管理系统 |  |

## 参考资料

* https://spring.io/
* http://blog.mybatis.org/
* https://www.mysql.com/

# 概要设计决策

本节记录对概要设计具有重要影响的软件需求和约束，描述相关的重要决策和决策的原因。这些决策一般对设计有很大影响,它们的改变很难通过设计技巧来屏蔽，例如所选用的开发语言,所选用的平台、组件和框架，对关键业务，类和方法的处理等。

## 设计原则

本项目设计才用前后端分离的方式。后端采用Java来设计，并且使用SSM框架。Java的面向对象的特性，可以实现代码的复用；Java的自动垃圾回收机制，可以有效防止内存泄露；Java的跨平台性为用户的使用提供便利；此外Java有较多的现有框架可供使用，为开发的效率、安全性、健全性提供了保证。

前端部分我们采用了html，css，同时采用了vue框架。vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

## 设计决策

本节记录系统的所有设计决策。

### 决策1

|  |  |  |  |
| --- | --- | --- | --- |
| **设计决策** | 后端采用语言 | **标识号** | 1 |
| **问题描述** | 后端采用哪种语言进行开发 | | |
| **假设条件** | 后端项目可以用Java实行 | | |
| **可选方案** | 无 | | |
| **决策** | 使用Java 作为后端开发的语言 | | |
| **理由** | Java有现行的Web框架可以使用 | | |
| **影响** | 无 | | |
| **相关决策** | 项目IDE统一使用Intellij IDEA | | |

### 决策2

|  |  |  |  |
| --- | --- | --- | --- |
| **设计决策** | 前端采用语言 | **标识号** | 2 |
| **问题描述** | 前端采用哪种语言进行开发 | | |
| **假设条件** | 前端项目可以用时html，css语言开发 | | |
| **可选方案** | 无 | | |
| **决策** | 使用html，css作为前端开发的语言 | | |
| **理由** | html 和css 为主流的前端开发语言，有较多资料和例子提供参考 | | |
| **影响** | 无 | | |
| **相关决策** | 前端使用vue框架 | | |

### 决策3

|  |  |  |  |
| --- | --- | --- | --- |
| **设计决策** | 后端采用框架 | **标识号** | 3 |
| **问题描述** | 后端采用哪种框架进行开发 | | |
| **假设条件** | 后端采用框架进行框架可行 | | |
| **可选方案** | 无 | | |
| **决策** | 使用SSM框架进行开发 | | |
| **理由** | SSM框架较为成熟，提供了从Web后端处理到数据库的方案 | | |
| **影响** | 无 | | |
| **相关决策** | 后端采用Java语言 | | |

### 决策4

|  |  |  |  |
| --- | --- | --- | --- |
| **设计决策** | 前端采用框架 | **标识号** | 4 |
| **问题描述** | 前端采用哪种框架进行开发 | | |
| **假设条件** | 前端采用框架进行框架可行 | | |
| **可选方案** | 无 | | |
| **决策** | 使用vue框架进行开发 | | |
| **理由** | vue框架较为成熟，vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，vue 被设计为可以自底向上逐层应用 | | |
| **影响** | 无 | | |
| **相关决策** | 前端采用html，css语言 | | |

### 决策5

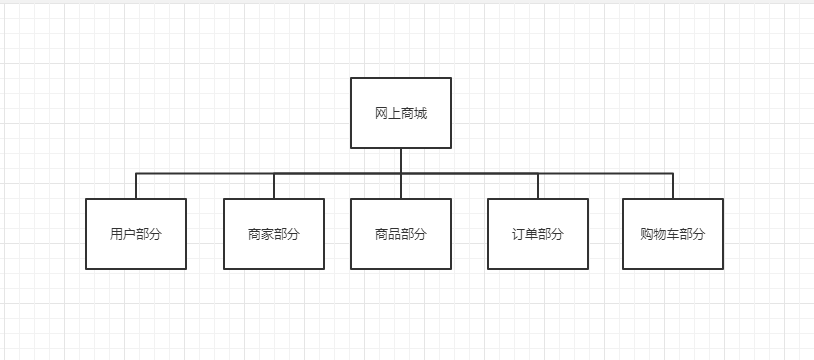
|  |  |  |  |
| --- | --- | --- | --- |
| **设计决策** | 数据库使用 | **标识号** | 5 |
| **问题描述** | 使用哪种数据库 | | |
| **假设条件** | 项目数据量需要数据库支持 | | |
| **可选方案** | 无 | | |
| **决策** | 使用MySQL数据库 | | |
| **理由** | ,MySQL被广泛地应用在Internet上的中小型网站中，其体积小、速度快、总体拥有成本低 | | |
| **影响** | 无 | | |
| **相关决策** |  | | |

# 总体设计

## 软件体系结构

本节说明对系统的层次结构（layer）和子系统的设计，说明各子系统的职责，相互之间的关系、操作和属性。对体系结构的设计应该满足系统的所有功能需求的要求。

本项目主要分为以下几个部分：顾客部分、商家部分、商品部分、订单部分、购物车部分。



用户部分：登录、注册、查看信息、查看订单信息

商家部分：添加商铺、添加商铺、查看商品

商品部分：显示全部商品、根据分类显示商品、显示商品详情、根据商品属性排序

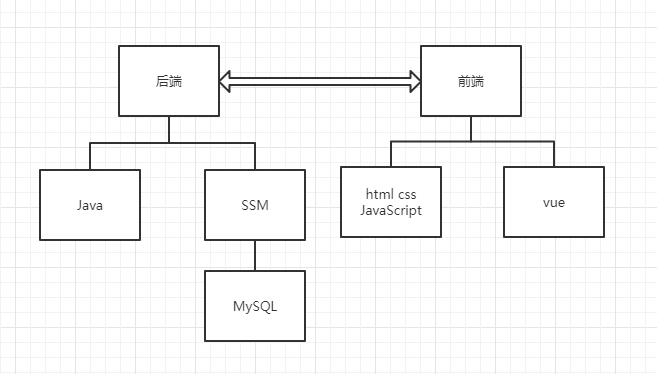
订单部分：添加订单、更新订单状态

购物车部分：添加购物车、购物车表项更新

## 系统技术构架

本项目使用了采用前后端分离的方式进行开发。后端采用Java语言进行开发，运用SSM框架。前端运用了html，css，javaScript语言，并且采用vue框架进行开发。

所用IDE为IntelliJ IDEA。服务部署于阿里云。



## 系统运行环境

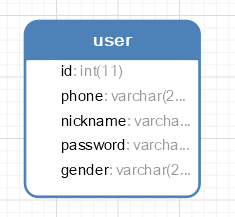
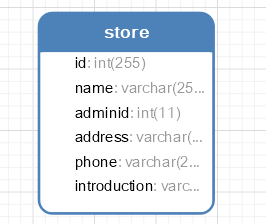
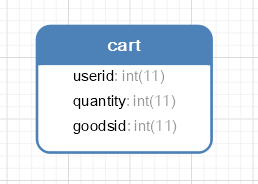
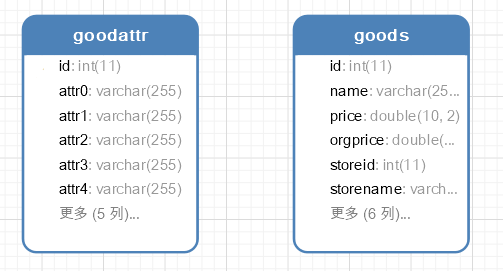
后端：部署于阿里云

数据库：部署于阿里云

前端：部署在本地

注意事项：演示时部署在本地

# 子系统设计

* 用户部分
* 名称：登录、注册
* 说明：用户通过输入用户名、密码登录到网上商城；用户通过按照格式提供信息进行网上商城的注册
* 相关类：User
* 相关类图：
* 商家部分
* 名称：登录、注册
* 说明：商家通过输入用户名、密码登录到网上商城；商家通过按照格式提供信息进行网上商城的注册
* 相关类：Admin
* 相关类图：
* 名称：显示所有商店所有商品
* 说明：根据商店id显示商店内所有商品
* 相关类：Store、Goods
* 相关类图：
* 订单部分
* 名称：对于订单的添加和更改状态
* 说明：可以对于订单的状态进行更改、下单时添加订单
* 相关类：Order、Goods
* 相关类图：
* 购物车部分
* 名称：对于购物车的增、查、删、改
* 说明：根据用户操作对于购物车进行增、查、删、改
* 相关类：Cart、Goods
* 相关类图：
* 商品部分
* 名称：对于商品和商品信息的增、查、删、改
* 说明：根据用户和商家的操作对于商品和商品信息的增、查、删、改
* 相关类：Goods Good\_attr
* 相关类图：

# 接口说明

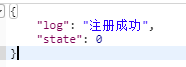
## 外部接口

------------------------------**User部分**-------------------------------------

//ps：接口未说明的情况下，请求方式默认为POST，返回值为json

//ps2:默认情况下，state0表示成功，state不为0表示失败，goods的state属性表示是否存在，1为存在，0为已删除

1. 注册

* URL：http://localhost:8080/user/register (POST)
* 参数:
  + nickname: 用户名
  + password：密码
* 返回值 json:
  + state: 0 log：注册成功 
  + state: 1 log：注册失败
  + state: 2 log：用户名重复
* 备注：希望前端做一下密码的格式检查

1. 登录

* URL：http://localhost:8080/user/login (POST)
* 参数:
  + nickname: 用户名
  + password：密码
* 返回值 json:
  + state:1 (登录失败，用户名或密码错误时返回)
  + state:1 和id（登录成功 ）

1. 获取用户信息

* URL：http:// localhost:8080/user/queryUserById (POST)
* 参数:
  + id：用户的id
* 返回值:
  + 用户id存在时返回用户信息和state:0 
  + 用户id不存在返回state:1

1. 修改用户信息

* URL：http://localhost:8080/user/updateUser (POST)
* 参数:
  + id：用于指定被修改用户的id
  + phone：电话
  + gender：性别
* 返回值:
  + state:0 （修改成功）

1. 修改密码

* URL：http://localhost:8080/user /updatePassword (POST)
* 参数:
  + nickname：指定修改的用户名
  + oldPassword：原密码
  + newPassword：新密码
* 返回值:
  + state:0 修改成功
  + state:1 密码错误（oldPassword错误时）
  + state:2 修改失败（无对应用户时）

1. 获取用户列表（仅用于测试）

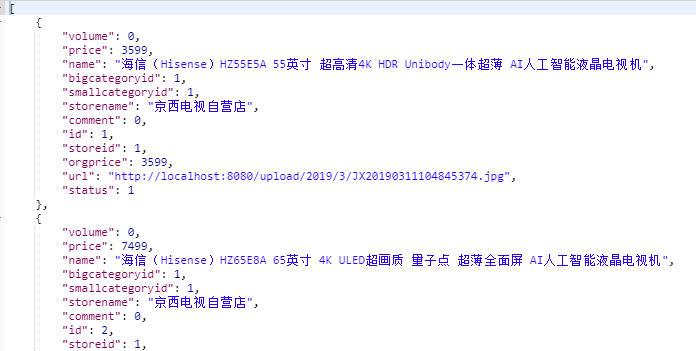
* URL：http://localhost:8080/user /queryUserById (POST)
* 参数: 无
* 返回值:
  + 返回所用用户信息的jsonarray：

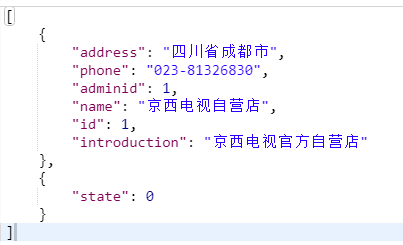
------------------------------**Cart部分**--------------------------------------

1. 获取购物车列表

* URL：http://localhost:8080/user /updatePassword (POST)
* 参数:
  + nickname：指定修改的用户名
  + oldPassword：原密码
  + newPassword：新密码
* 返回值:
  + state:0 修改成功
  + state:1 密码错误（oldPassword错误时）
  + state:2 修改失败（无对应用户时）

------------------------------**Store部分**-------------------------------------

* 添加商店
* URL：http://localhost:8080/store/addStore(POST)
* 参数:
  + name: 商店名
  + adminId：店主id
  + address：商店地址
  + phone：商店电话
  + introduction：介绍
* 返回值 json:
  + state: 0 添加成功
  + state: 1 添加失败
* 加载商店的商品列表
* URL：http://localhost:8080/store/showAllGoodsById(POST)
* 参数:
  + id：商店的id
* 返回商店里的商品列表：
  + 
* 查询商店信息
* URL：http://localhost:8080/store/showAllAttributesOfStore (POST)
* 参数:
  + id：商店的id
* 返回值 json:
  + state: 0和商店信息（查询成功）
  + state: 1 （查询失败）
* 通过商店名称查找商店（模糊查找）
* URL：http://localhost:8080/store/showStoreByName (POST)
* 参数
* name: 商店的名字的一部分
* 返回值：jsonarray



------------------------------Good**部分**-------------------------------------

1. 查询热销商品

* URL：http://localhost:8080/good/findHotestGoods (POST/GET)
* 参数:
  + 无
* 返回值 json:
  + 销量前10的商品的信息 json（按销量降序）
  + 

1. 添加商品（未完成）

* URL：http://localhost:8080/good/addGoods (POST)
* 参数:
  + name：商品名
  + price：商品现价
  + orgprice：商品原价
  + storeid：商店id
  + storename：商店名
  + bigcategoryid：大类id
  + smallcategoryid：小类id
  + url：商品主图的url
* 返回值 json:
  + 添加成功：state：0
  + 添加失败：state：1

1. 获取大类商品列表

* URL：http://localhost:8080/good/findGoodsByBigcategory (POST)
* 参数:
  + bigId：大类的id
* 返回值 json:
  + 返回商品列表的json（goodList）和销量最高的4个商品的json（hostList）
  + 
  + 

1. 获取小类商品列表

* URL：http://localhost:8080/good/findGoodsBySmallcategory (POST)
* 参数:
  + smallId：小类的id
* 返回值 json:
  + 返回商品列表的json（goodList）和销量最高的4个商品的json（hostList）
  + 

1. 获取商品详情

* URL：http://localhost:8080/good/goodsPage (POST/GET)
* 参数:
  + id：商品的id
* 返回值 json:
  + 返回商品详情的jsonarray，包含三个json对象
    - attr：包含商品的详细属性（内有attr0~9，共10个属性）
    - info：包含商品的主要信息
    - pic：包含商品的图片list（jsonarray）
    - 

1. 搜索商品（未完成，打算做一个简单的模糊搜索）

------------------------------Comment**部分**------------------------------------

1. 发送评论

* URL：http://localhost:8080/comment/addComment (POST)
* 参数:
  + userid：发送者的id
  + goodsid：评论商品的id
  + content：评论内容
  + rating：评价（int型）（1：差评，2：好评，3：差评）
* 返回值 json:
  + state：0 发送成功，发送成功会自动给商品的评论数+1
  + state：1 发送失败

1. 获取商品评论

* URL：http://localhost:8080/comment/getComment (POST)
* 参数:
  + goodsid：评论商品的id
* 返回值 json:
  + 查询成功返回： state：0 和评论信息
  + 
  + 查询失败返回 state：1

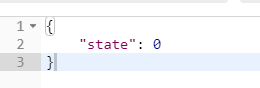
------------------------------Order部分-------------------------------------

1. 添加订单

* URL：http://localhost:8080/order/addOrder (POST)
* 参数：
* userid: 用户的id
* goodsid: 商品的id
* quantity：商品的数量
* totalprice：总价
* adminid：商家的id
* state：订单状态（verchar字符串）
* unitprice：商品单价
* 返回值：json

添加成功返回 state：0

添加失败返回 state:1

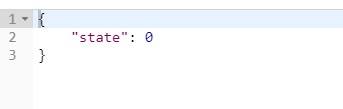


1. 更改订单

* URL：http://localhost:8080/order/updateOrder (POST)
* 参数：
* id: 订单的id
* quantity：商品的数量
* totalprice：总价
* state：订单状态（verchar字符串）
* unitprice：商品单价
* 返回值：json

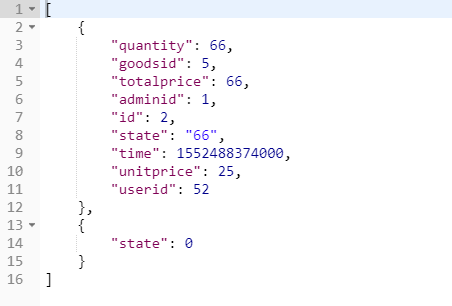
添加成功返回 state：0

添加失败返回 state:1



1. 查找用户的订单

* URL：http://localhost:8080/order/showOrderByUserId (POST)
* 参数：
* userid: 用户的id
* 返回值：jsonarray



1. 查找商家的订单

* URL：http://localhost:8080/order/showOrderByAdminId（POST)
* 参数：
* adminid: 商家的id
* 返回值：jsonarray



## 内部接口

* Dao
* CartDao
* CommonDao
* GoodDao
* OrderDao
* PicDao
* StoreDao
* UserDao
* Service
* CartService
* CommonService
* GoodServie
* OrderService
* PicService
* StoreService
* UserService