**显卡购物商城后台**

**设**

**计**

**文**

**档**

**小组：第六组**

**成员：李雪辉、文振中、陈秀萍**

# 需求分析

1. 实现后台用户增删改查功能
2. 实现后台商品增删改查功能
3. 实现后台商品类型增删改查功能
4. 实现后台订单修改查询删除功能

# 架构设计

设计框架图：

jsp请求

判断请求类型

Servlet实现业务功能

Service+ServiceImpl实现数据库交互

DAO辅助处理

数据处理完成通过session转交给JSP页面渲染

完成业务处理并返回给用户

# 模块设计

## 用户增删改查

采用路由方式，减少Servlet的使用。

首先建立用户管理的后台JSP页面，页面必须有用户的增删改查入口。

然后在admin包中新增一个Servlet用于路由，再将所有的一类业务逻辑写在Action包中。

1、增加用户可使用注册的Dao方法。

2、删除用户需要在UserService中添加一个 deleteUser方法与数据库交互

3、修改用户需要在UserService 中添加一个updateUser 方法与数据库交互

4、查询用户可直接使用queryAll方法获得所有用户，然后使用for循环和String自带的indexof方法判断是否包含需要查询的关键字。

## 商品类型增删改查

采用普通的一个功能对应一个Servlet的方式完成。

首先建立商品类型管理的后台JSP页面，页面必须有增加商品类型、修改商品类型、查询商品类型的入口。

1. 商品类型增加：新建一个Servlet取名 AddNewProductTypeServlet.java 主要用于商品类型的增加，在productService 中新增一个接口方法 insertProductType(ProductType pt)并在productServiceImpl 中完成实现，在其中完成数据库交互的功能。
2. 商品类型修改：新建一个Servlet 取名为 ChangeProductTypeServlet.java，在其中实现业务逻辑，并在productService 中新增一个接口方法 updateProductType 并在productServiceImpl中完成实现，在其中完成修改商品的数据库交互。
3. 商品类型查询：新建一个SearchProductTypeServlet.java 主要用于商品类型查询的业务逻辑，并在productService 中新增一个接口方法 getProductTypes() 并在productServiceImpl中完成实现，此方法获取所有的商品类型，在SearchProductTypeServlet.java中使用for循环比对完成搜索功能。

## 商品增删改查

采用普通的一个功能对应一个Servlet的方式完成。

首先建立商品管理的后台JSP页面，页面必须有增加商品类型、修改商品类型、查询商品类型的入口。

1. 增加商品：在admin包中新建一个AddNewProductServlet.java 实现业务逻辑，收集需要新增的商品信息，封装成一个对象，然后在ProductDao中新建一个方法insert(Product p)完成数据库交互功能
2. 删除商品：为了数据完整性，不直接删除商品信息，而是新增一个字段记录商品删除信息，is\_delete，如果值为1则此商品被删除，否则未删除。

业务逻辑实现：新增一个DeleteProductServlet.java 并在其中实现收集需要删除的商品id，在productService中实现deleteProduct（Integer id）方法，在其中完成数据库交互功能。

1. 修改商品：新建一个Servlet 取名为 ChangeProductServlet.java，在其中实现业务逻辑，并在productService 中新增一个接口方法 updateProduct（Product p） 并在productServiceImpl中完成实现，在其中完成修改商品的数据库交互。
2. 查询商品：新建一个SearchProductServlet.java 主要用于商品类型查询的业务逻辑，并在productService 中新增一个接口方法 queryAll() 并在productServiceImpl中完成实现，此方法获取所有的商品，在SearchProductServlet.java中使用for循环比对完成搜索功能。

## 订单修改查询

采用普通的一个功能对应一个Servlet的方式完成。

首先建立订单管理的后台JSP页面，页面必须有修改订单状态、查询订单的入口。

1. 订单的修改：新建一个Servlet 取名为 ChangeOrderServlet.java，在其中实现业务逻辑，并在OrderService 中新增一个接口方法 updateStatus（Order 0,String Status） 并在OrderServiceImpl中完成实现，在其中完成修改商品的数据库交互。
2. 订单的查询: 新建一个Servlet 取名为 SearchOrderServlet.java，在其中实现业务逻辑，并在OrderDao 中新增一个接口方法 queryAllOrder() 并在OrderDaoImpl中完成实现，在其中完成查询所有商品的数据库交互，再在SeachOrderServlet中完成比对搜索功能。

# 分工实现

|  |  |
| --- | --- |
| 李雪辉 | 负责Servlet和Service层的开发 |
| 陈秀萍 | 负责JSP页面与后台Servlet的交互 |
| 文振中 | 负责Dao层的开发 |