长 沙 学 院

课程设计说明书

|  |  |
| --- | --- |
| 题目 | 网上商店前台功能的设计与实现 |
| 学院 | 计算机工程与应用数学学院 |
| 专业(班级) | 16级软件1班 |
| 姓名 | 黄文威 |
| 学号 | B20160304108 |
| 指导教师 | 段晓娟、马丽 |
| 起止日期 | 2019.5.20～2019.5.31 |

**目录**

[1 设计内容与要求 1](#_Toc468700606)

[2需求分析 2](#_Toc468700607)

[2.1需求描述 2](#_Toc468700608)

[2.2用例规约 2](#_Toc468700609)

[3概要设计 3](#_Toc468700610)

[3.1总体设计 3](#_Toc468700611)

[3.2功能模块设计 4](#_Toc468700612)

[3.3数据库设计 4](#_Toc468700613)

[3.4页面设计 4](#_Toc468700614)

[3.5接口设计 5](#_Toc468700615)

[3.6运行设计 5](#_Toc468700616)

[3.7出错处理设计 5](#_Toc468700617)

[4详细设计 6](#_Toc468700618)

[4.1 方法流程设计 6](#_Toc468700619)

[5 编码 7](#_Toc468700620)

[6 结论 8](#_Toc468700621)

[参考文献 9](#_Toc468700622)

[附录 10](#_Toc468700623)

[附录A 本系统的组成 10](#_Toc468700624)

[附录B 程序源代码 10](#_Toc468700625)

课程设计任务书

**课程名称：**Java Web应用开发软件课程设计

**设计题目：**网上商店前台功能的设计与实现

### 已知技术参数和设计要求：

1. 问题描述（功能要求）：

1. 分析设计一个网上商店的前台功能部分。完成本系统的如下功能：
   * 用户注册和登陆
   * 展示特定的某类商品：展示的方式有多种，可以按类别、按销售排行榜、按上架时间等，有层次的展示商品，提高用户的购买兴趣。对于某种商品，除显示商品信息以外，还可以附加评论、推荐相关类型商品。
   * 商品查询：根据多种条件来查询，比如商品名称、价格、颜色等特征信息。能进行模糊匹配，包含关键字的商品都能查到。
   * 购物车：只有注册用户才能购买。为用户提供购物车，购物车内同类商品的数量可以修改，也可以删除购物车里已有的商品。购物车信息保存，便于下次购物时继续选购商品。

**2. 运行环境要求：**

正确安装、配置和运行JDK。

正确安装、配置和运行Tomcat。

正确安装、配置和运行Eclipse。

正确安装、配置和运行数据库和JDBC驱动程序。

3. 技术要求：

要求使用Java语言，利用面向对象的方法、编程思想来完成系统的设计；在设计的过程中，建立清晰的类层次；在系统设计中要分析和定义各个类，每个类中要有各自的属性和方法。写出需求分析，UML设计图，数据库设计图。

设计工作量：

40课时

工作计划：

1.班级

16软件1~2班

2.课时及教室安排

10课时：相关知识介绍。机房内多媒体教室

10课时：设计。普通教室。

16课时：上机、调试。计算机系机房

4课时：调试答辩。计算机系机房

**注意事项**

**■提交文档**

* 长沙学院课程设计任务书 （每学生1份）
* 长沙学院课程设计说明书 （每学生1份）
* 长沙学院课程设计鉴定表 （每学生1份）

# 2需求分析

## 2.1需求描述

本系统商品为手机，平板电脑等电子产品，商品具有以下属性：

1. id int类型，商品编号
2. name varchar类型，商品名称
3. relation int类型， 表示类型与品牌
4. price double类型 商品价格
5. volume int类型 商品销量
6. imagpath varchar类型 商品图片地址

商品根据类型与品牌分类，类型与品牌为同级，类别名称有小米手机，苹果手机，华为平板电脑等.

查询根据商品名字查询，输入的关键字与商品的name属性匹配.

功能模块主要有：登录、注册、查看商品、搜索商品、购物车管理。其中查看商品可以按照商品的品牌和类型进行查看，并可以根据商品编号，商品价格，商品销量排序后查看，购物车管理包括查看购物车，将商品加入购物车，将商品从购物车车移除，清空购物车

## 2.2用例规约

登录功能：用户输入账号密码，点击登录按钮，若账号密码正确则会跳转至商品展示页面，账号密码错误或不存在则会跳转至登录失败界面。

注册功能：用户输入用户名，密码，手机号码，电子邮箱，若输入符合规范则跳转到注册成功页面，不符合或用户名已存在则跳转到注册失败页面

注销功能：用户点击注销按钮，退出登录

查看商品功能：用户登录成功后进入展示商品页面，默认展示所有商品（展示商品id，商品名称，价格，销量，商品图片），点击页面上的两个下拉框可以展示该类别下的商品（展示的商品属性与上相同），点击商品编号进入展示商品信息页面（展示商品id，商品名称，商品品牌，商品类型，价格，销量，商品图片），并会展示推荐商品（相同类别的其他商品）.

搜索商品：除了登录和注册页面，所有页面上方都有搜索框，在搜索框中输入关键字，点击搜索，会展示符合关键字的商品（展示商品id，商品名称，价格，销量，商品图片）

购物车管理：1.添加商品至购物车：在所有商品展示页面，商品信息旁边都会有一个购买链接，点击可将该商品加入登录用户的购物车，可将同件商品多次加入. 2.展示购物车：将登录用户的购物车中的商品信息展示出来（展示商品id，商品名称，在购物车中的数量，价格，商品图片） 3.从购物车中移除商品 商品信息最右边有个移除链接，点击可将该商品全部移除出购物车 4.清空购物车：页面右上方有个清空购物车链接，点击后会弹出来一个确认窗口，点击确认后将购物车中所有商品都移除出购物车.

# 3概要设计

## 3.1总体设计

购物系统

购物车管理

查询商品

用户管理

浏览商品

按类别浏览

用户注销

移除购物车。。

清空购物车

添加购物车

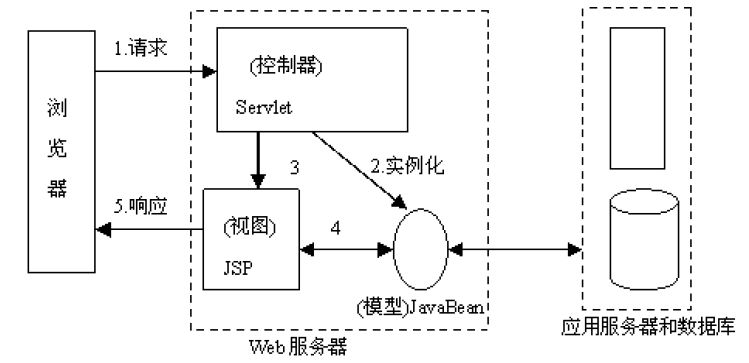
按名字查询

按品牌浏览

用户注册

用户登录

3.1总体设计图



3.2框架设计图

画出每个功能模块对应的的视图层、控制器层和模型层。对前述的框架设计图进行具体化的应用。

假设此系统的组成如下：（请完善下图，掉其中的。。。）

.servlet

ShowShopCartSevlet

LogoutSevlet

FindAllProductSevlet

-

RegisterSevlet

ShopCartSevlet

FindProducInfoSevlet

SortSevlet

SearchSevlet

LoginSevlet

dao

SearchDao

LoginDao

-

SortDao

ProductDao

RegisterDao

index.html

bean

Navigationbar.jsp

\*.jsp,\*.html

Product

Register.html

-

User

showProduct.jsp

ShopCart

showProductInfo.jsp

DataBaseAccess

showShopCart.jsp

DataBaseAccess

登录功能：控制器：LoginServlet 模型：LoginDao 视图: index.html

注册功能: 控制器：RegisterServlet 模型：RegisterDao 视图: Register.html

查看商品：控制器：findAllproductServlet,findProductInfoServlet,SortServlet 模型：ProductDao,SortDao 视图: showProduct.jsp, showProductInfo.jsp, Navigationbar.jsp

搜索功能：控制器：SortSevlet 模型：Sort Dao 视图: showProduct.jsp,.

购物车管理： 控制器：ShopCartSevlet，ShowShopCartSevlet 模型：ProductDao视图:showShopCart.jsp

注销功能： 控制器：LogoutServlet 模型： 视图: index.html

## 3.2功能模块设计

用户管理模块：此模块功能包括：用户注册、登录、注销。用户注册时填写用户名（判断唯一性）、密码，手机号邮箱。用户登录成功后可查看、搜索商品，管理购物车，用户可以选择注销，注销时返回首页.

浏览商品模块：用户登录后可以按照商品的类型或者品牌来查看商品，商品的信息包括商品可以根据销量，价格，编号来进行排序。

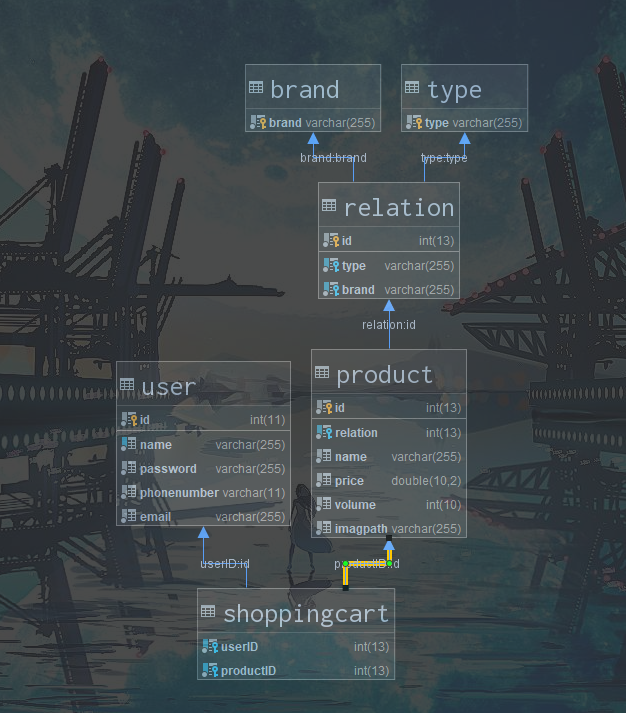
搜索模块：用户输入关键字并点击搜索，系统将根据关键字返回展示相关商品的页面

购物车管理：用户登录后可以管理购物车，在商品展示页面可以将商品加入购物车，在购物车展示页面可以将商品移除出购物车，并可以将购物车直接清空

## 3.3数据库设计

本系统数据库中需要存储的有用户信息，商品信息，购物车信息，数据库中总共有user,product,relation,brand,type,,shoppingcart 6张表，用户表中有用户ID用户名，密码，手机号码，电子邮箱，商品信息表中有商品的编号，商品名，价格，销量，关系，图片位置。商品的品牌和类型组成商品的关系属性，商品表中保存商品的关系信息，relation表中保存类型和品牌的不同组合形成的关系，type表保存类型信息，brand表保存品牌信息，购物车表保存用户的购物信息，由保存用户id和商品id来实现.

E-R图：分析其中存在的实体之间的关系。



表结构 例如：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用户信息表(user) | | | | | |
| 字段名 | 字段类型 | 类型说明 | 约束检查 | 是否为空 | 字段含义 |
| Id | int(11) | 整型 | 主键 | 否 | 用户编号 |
| Name | VARCHAR(255) | 字符型 | Unique | 否 | 用户名 |
| Password | VARCHAR(255) | 字符型 |  | 否 | 用户密码 |
| phonenumber | VARCHAR(11) | 字符型 |  | 否 | 用户电话号码 |
| email | Varchar(255) | 字符型 |  | 否 | 用户电子邮箱 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 商品信息表(product) | | | | | |
| 字段名 | 字段类型 | 类型说明 | 约束检查 | 是否为空 | 字段含义 |
| Id | int(11) | 整型 | 主键 | 否 | 商品编号 |
| Name | VARCHAR(255) | 字符型 |  | 否 | 商品名 |
| Relation | int(13) | 整型 |  | 否 | 商品关系 |
| Price | double(10) | 浮点数型 |  | 否 | 商品价格 |
| Volume | int(10) | 整型 |  | 否 | 商品销量 |
| imagpath | VARCHAR(255) | 字符型 |  | 否 | 商品图片地址 |
| 购物车表(shopcart) | | | | | |
| 字段名 | 字段类型 | 类型说明 | 约束检查 | 是否为空 | 字段含义 |
| userId | int(13) | 整型 |  | 否 | 用户编号 |
| ProductID | int(13) | 整型 |  | 否 | 商品编号 |

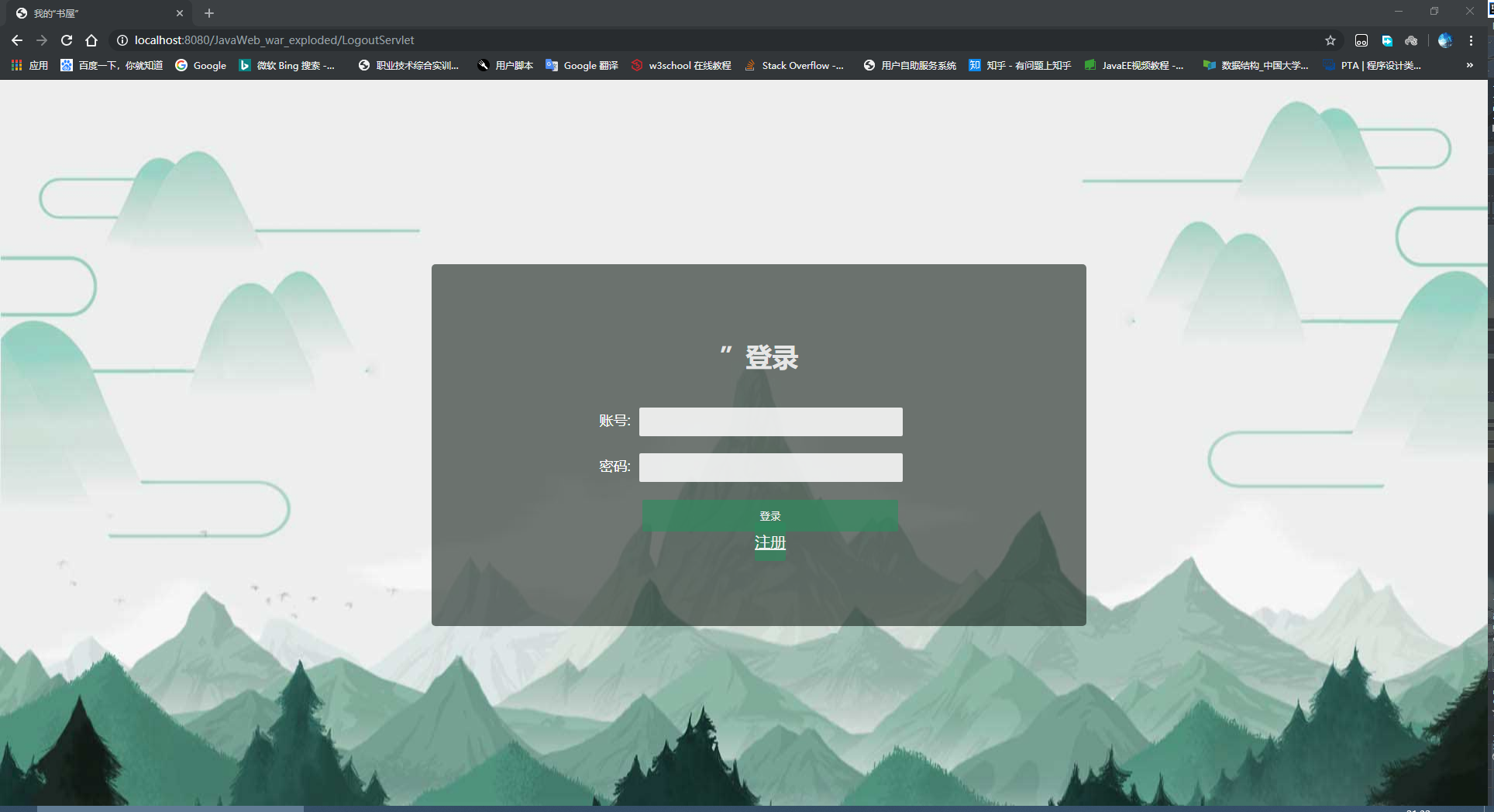
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 关系表(relation) | | | | | |
| 字段名 | 字段类型 | 类型说明 | 约束检查 | 是否为空 | 字段含义 |
| Id | int(13) | 整型 |  | 否 | 关系编号 |
| Type | VARCHAR(255) | 字符型 |  | 否 | 商品类型 |
| Brand | VARCHAR(255) | 字符型 |  | 否 | 商品品牌 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 品牌表(relation) | | | | | |
| 字段名 | 字段类型 | 类型说明 | 约束检查 | 是否为空 | 字段含义 |
| Brand | VARCHAR(255) | 字符型 |  | 否 | 商品品牌 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类型表(relation) | | | | | |
| 字段名 | 字段类型 | 类型说明 | 约束检查 | 是否为空 | 字段含义 |
| Type | VARCHAR(255) | 字符型 |  | 否 | 商品类型 |

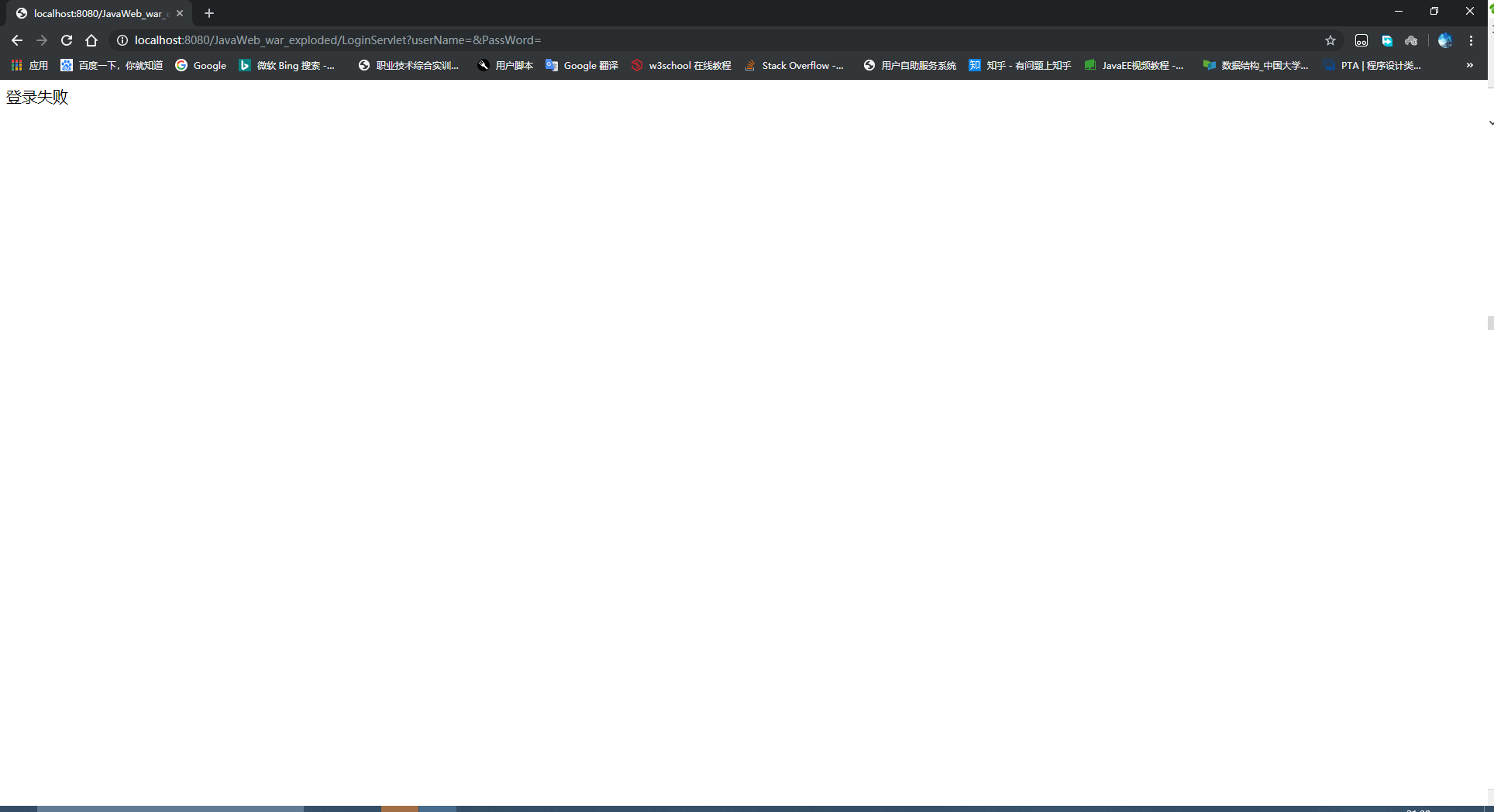
## 3.4页面设计

此系统中涉及的所有页面，包含输入的表单和输出的结果页面的各种元素（比如链接，表格）的组成，以及它们分别是用来做什么用的（比如链接是作为导航使用，或者是用于链接到商品详情页面）。



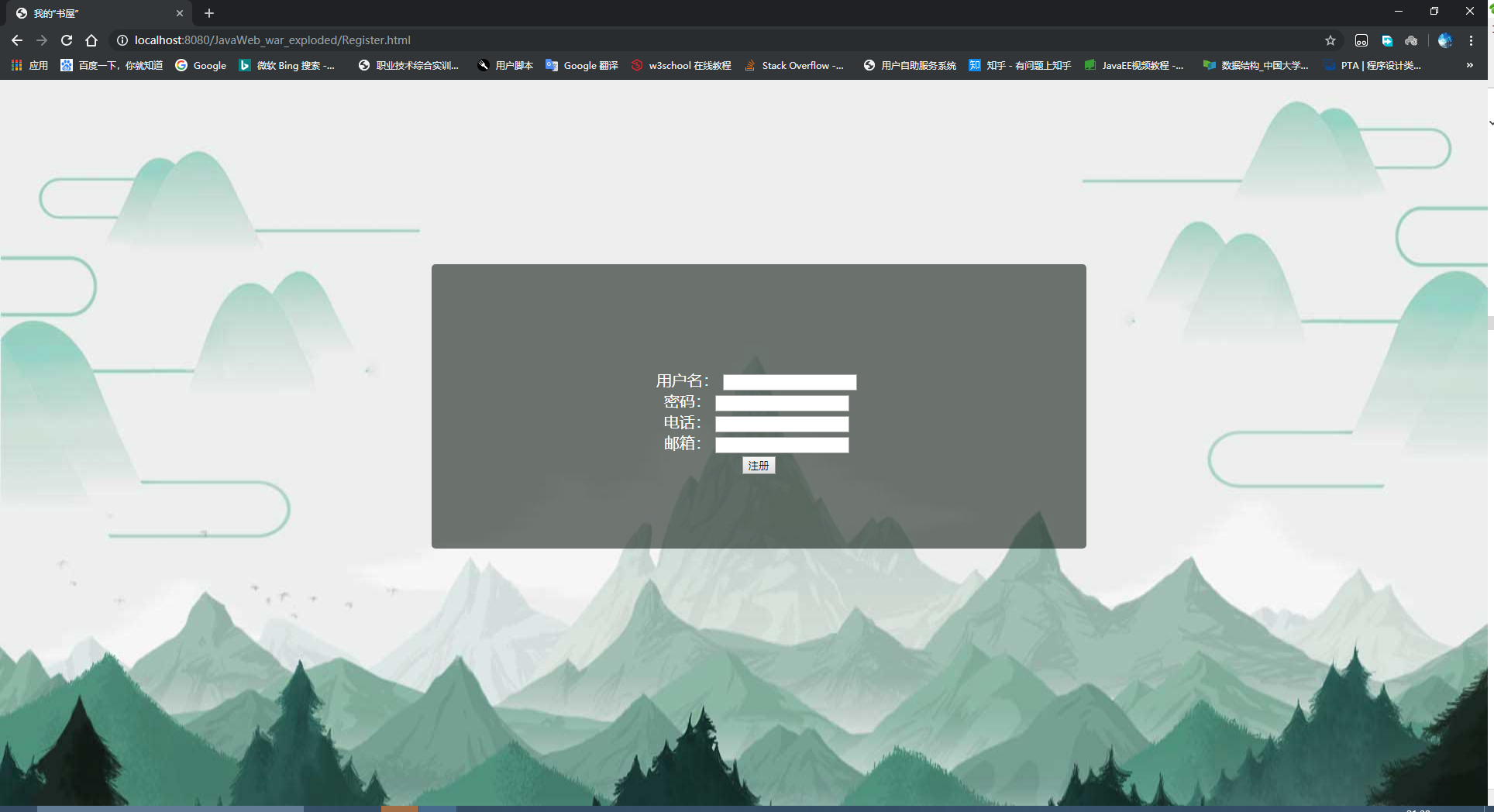
登录界面

一个输入表单用于输入用户名和密码，俩个链接，一个用于验证登录及跳转至展示商品页面，另一个用于跳转至注册界面。



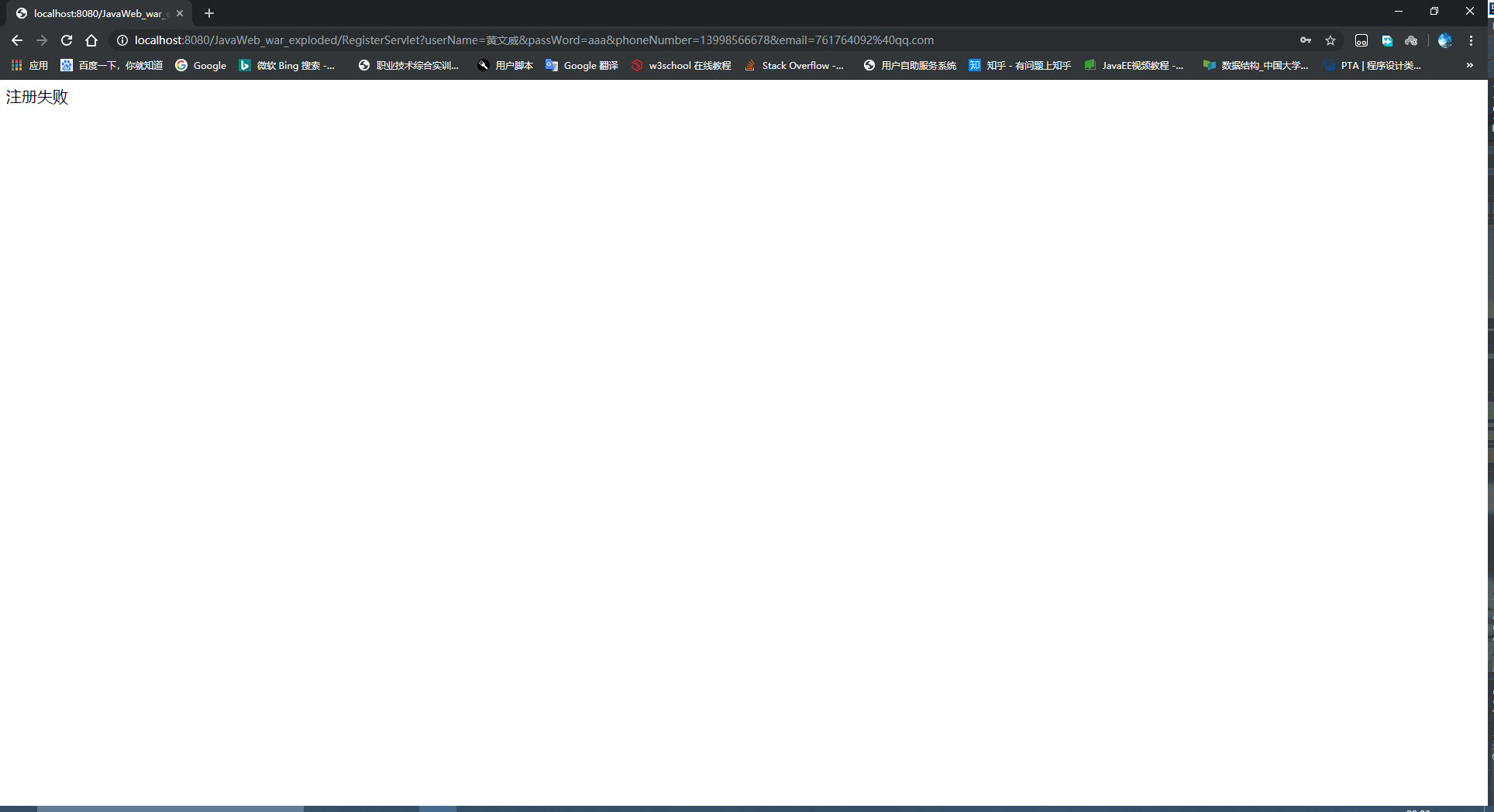
登录失败页面

通知用户登录失败



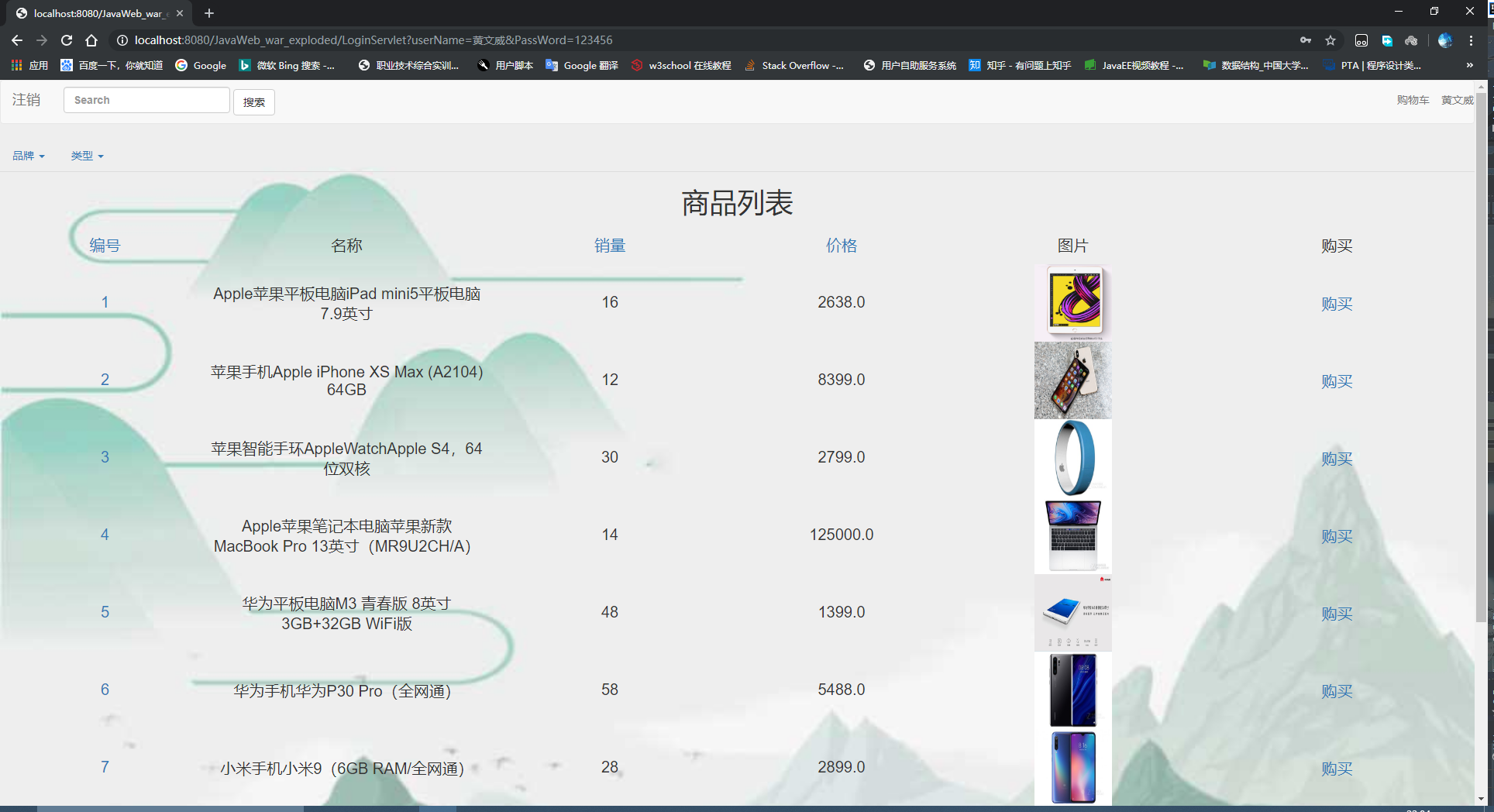
注册页面

一个表单用于用户输入数据，一个链接调用RegisterServlet实现注册



注册失败页面

通知用户注册失败。

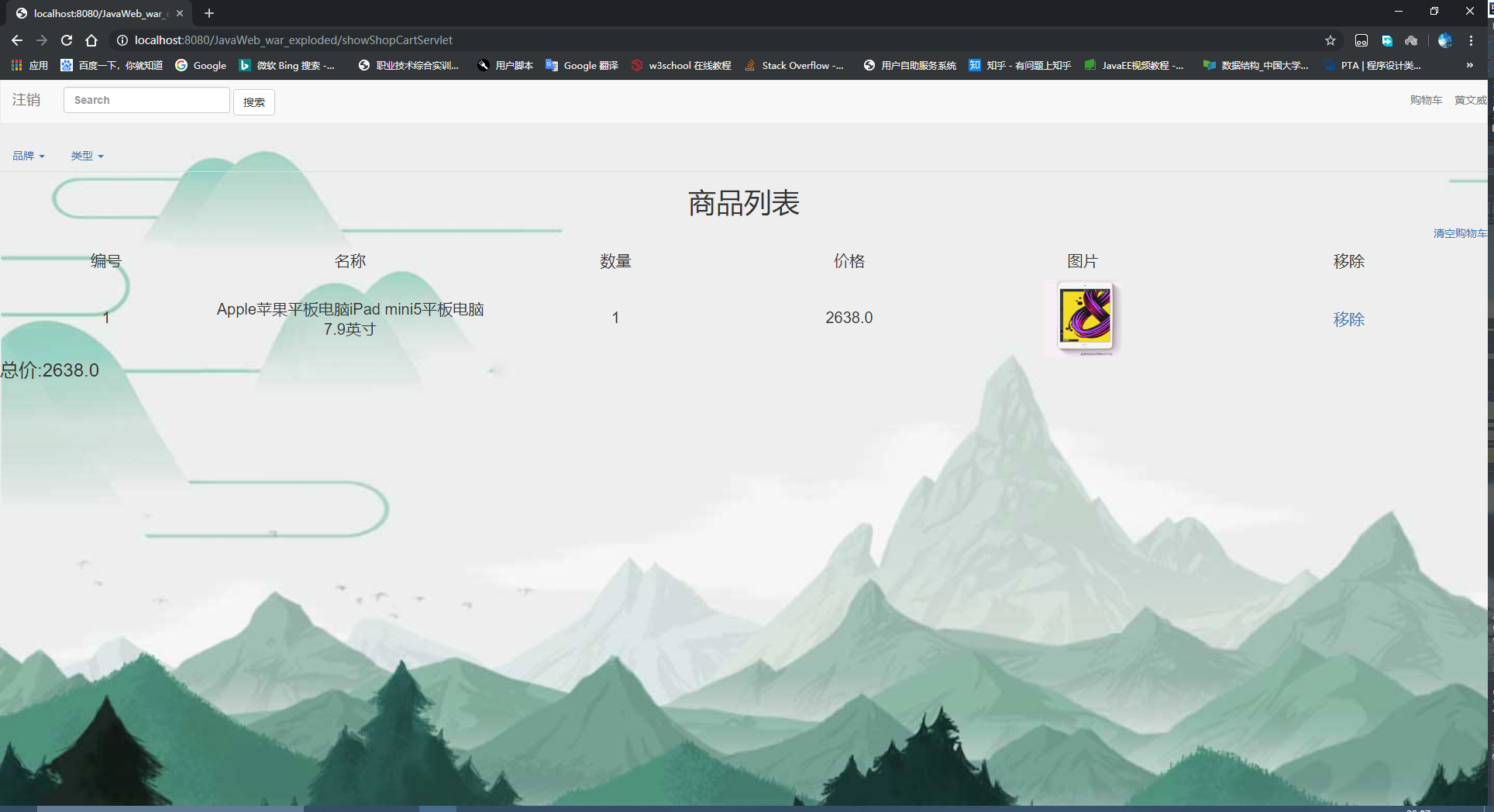


展示商品页面

包括一个导航栏，一个表格，导航栏里存在一个表单用于输入搜索的关键字，导航栏最左边的注销可以点击然后退出登录，跳转至登录界面，导航栏右边显示用户名和一个跳转至购物车的连接，表格用于展示商品信息，表格上面有两个下拉框，可以根据品牌和类型来显示商品，表格中第一行的编号，销量，价格可以点击，点击后会分别根据编号，销量，价格进行排序，表格最右边的购买可以点击将商品加入购物车。

商品信息页面

用一个表格展示商品信息，另一个表格展示推荐商品的信息，推荐商品为同一类型的不同产品



购物车页面

用一个表格展示用户购物车中的商品信息，如果一个商品在购物车中存在多件，数量上会显示出来，点击表格右上方的清空购物车链接会弹出一个警告框，确认后会将购物车中所有商品移除，点击表格最右端的移除链接会将该件商品从购物车中移除，不管是一件还是多件。

## 3.5接口设计 （写出类名，类的方法名，方法的参数的定义和返回值含义。）

例如：1）控制层(Servlet)接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 01 | | 备注 |
| 接口类名 | LoginServlet | | 登录控制器 |
| 接口描述 | 登录页面请求LoginServlet类的doget接口完成登录验证跳转 | | |
| 接口方法名 | doget | | 登录控制器 |
| 接口参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest |  |
| 2 | Resp | HttpServletResponse |  |
| 返回类型 | 用户登录成功返回商品展示页面  登录失败返回登录失败页面 | |  |

2）LoginDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 02 | | 备注 |
| 接口类名 | LoginDao | | 用户管理服务 |
| 接口描述 | 用于用户管理，此处描述登录验证接口方法 | | |
| 接口方法名 | isLogin | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | userName | Strng | 登录名 |
| 2 | password | String | 密码 |
| 返回类型 | boolean | | 成功返回true，失败返回false |

3）RegisterServlet接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 03 | | 备注 |
| 接口名 | RegisterServlet | | 注册控制器 |
| 接口描述 | 注册页面请求RegisterServlet的doget接口完成注册验证跳转 | | |
| 接口方法名 | Doget | | 注册控制器 |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest | . |
| 2 | Resp | HttpServletResponse | . |
| 返回类型 | . 用户登录成功返回商品展示页面  登录失败返回登录失败页面 | | . |

4）RegisterDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 04 | | 备注 |
| 接口类名 | RegisterDao | | 用户管理服务 |
| 接口描述 | 用于用户管理，此处描述注册验证接口方法 | | |
| 接口方法名 | isRegister | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | userName | Strng | 登录名 |
| 2 | password | String | 密码 |
| 3 | Phonenumber | String | 电话号码 |
| 4 | Email | String | 电子邮箱 |
| 返回类型 | boolean | | 成功返回true，失败返回false |

5）findAllProductServlet接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 05 | | 备注 |
| 接口类名 | findAllProductServlet | | 展示所有产品 |
| 接口描述 | 展示页面请求findAllProductServlet的doget接口完成商品展示 | | |
| 接口方法名 | Doget | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest | . |
| 2 | Resp | HttpServletResponse | . |
| 返回类型 | Void | | 成功：在req中设置一Arraylist<product> |

6）ProductDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 06 | | 备注 |
| 接口类名 | ProductDao | | 产品管理 |
| 接口描述 | 用于商品管理，此处描述查找商品接口方法 | | |
| 接口方法名 | findAllProduct | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Void |  | . |
| 返回类型 | ArrayList<Product> | | 成功返回保存了所有产品信息的Arraylist<product> |

7）LogoutServlet接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 07 | | 备注 |
| 接口类名 | LogoutServlet | | 注销控制器 |
| 接口描述 | 页面请求LogoutServlet的doget方法实现注销 | | |
| 接口方法名 | Doget | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest |  |
| 2 | Resp | HttpServletResponse |  |
| 返回类型 | Void | | 成功：清除seesion中的信息，用户需重新登录 |

8）findProductInfoServlet接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 08 | | 备注 |
| 接口类名 | findProductInfoServlet | | 展示商品信息控制器 |
| 接口描述 | 页面请求findProductInfoServlet的doget接口展示商品信息 | | |
| 接口方法名 | Doget | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest |  |
| 2 | Resp | HttpServletResponse |  |
| 返回类型 | Void | | 成功：返回产品信息 |

9）ProductDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 09 | | 备注 |
| 接口类名 | ProductDao | | 产品管理 |
| 接口描述 | 用于商品管理，此处描述查找商品接口方法 | | |
| 接口方法名 | findProduct | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Pid | int | 产品号 |
| 返回类型 | Product | | 成功返回保存了产品所有信息的product |

10）SearchServlet接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 10 | | 备注 |
| 接口类名 | SearchServlet | | 搜索控制器 |
| 接口描述 | 页面请求SearchServlet的doget接口展示商品信息 | | |
| 接口方法名 | Doget | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest |  |
| 2 | Resp | HttpServletResponse |  |
| 返回类型 | Void | | 成功：返回产品信息 |

11）SearchDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 11 | | 备注 |
| 接口类名 | SearchDao | | 产品搜索 |
| 接口描述 | 用于商品搜索，此处描述搜索商品接口方法 | | |
| 接口方法名 | search | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Key | String | 关键字 |
| 返回类型 | ArrayList<Product> | | 成功返回符合关键字的product集合 |

12）SortServlet接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 12 | | 备注 |
| 接口类名 | SortServlet | | 排序控制器 |
| 接口描述 | 页面请求SortServlet的doget接口实现排序 | | |
| 接口方法名 | Doget | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest |  |
| 2 | Resp | HttpServletResponse |  |
| 返回类型 | Void | | 成功：返回完成排序的产品集合 |

13）SortDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 13 | | 备注 |
| 接口类名 | SortDao | | 产品排序 |
| 接口描述 | 用于商品排序，此处描述搜索商品排序方法 | | |
| 接口方法名 | sort | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | HowToSort | String | 排序要求 |
| 2. | products | ArrayList<Product> | 被排序的产品集合 |
| 返回类型 | ArrayList<Product> | | 成功返回完成排序的product集合 |

14）showShopCartServlet接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 14 | | 备注 |
| 接口类名 | showShopCartServlet | | 购物车展示控制器 |
| 接口描述 | 页面请求showShopCartServlet的doget接口实现排序 | | |
| 接口方法名 | Doget | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest |  |
| 2 | Resp | HttpServletResponse |  |
| 返回类型 | Void | | 成功：返回购物车信息 |

15）productDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 15 | | 备注 |
| 接口类名 | productDao | | 展示购物车 |
| 接口描述 | 用于展示购物车，此处描述展示购物车内产品信息方法 | | |
| 接口方法名 | showShopCart | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | ShopCart | Shopcart | 购物车实体类 |
| 返回类型 | ArrayList<Product> | | 成功返回购物车中存在的产品的product集合 |

16）ShopCartServlet接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 16 | | 备注 |
| 接口类名 | ShopCartServlet | | 购物车展示控制器 |
| 接口描述 | 页面请求ShopCartServlet的doget接口实现购物车管理 | | |
| 接口方法名 | Doget | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | Req | HttpServletRequest |  |
| 2 | Resp | HttpServletResponse |  |
| 返回类型 | Void | | 成功：完成购物车操作 |

17）productDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 17 | | 备注 |
| 接口类名 | productDao | | 添加购物车 |
| 接口描述 | 用于将商品插入购物车，此处描述展示插入购物车方法 | | |
| 接口方法名 | insertShopCart | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | UserID | Int | 用户编号 |
| 2 | Pid | Int | 产品编号 |
| 返回类型 | Void | | 成功在该用户的购物车中插入该商品 |

18）productDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 18 | | 备注 |
| 接口类名 | productDao | | 移除购物车 |
| 接口描述 | 用于将商品移除出购物车，此处描述展示移除购物车方法 | | |
| 接口方法名 | removeShopCart | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | UserID | Int | 用户编号 |
| 2 | Pid | Int | 产品编号 |
| 返回类型 | Void | | 成功将该用户的购物车中的该件商品移除 |

19）productDao接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 19 | | 备注 |
| 接口类名 | productDao | | 清空购物车 |
| 接口描述 | 用于将用户的购物车清空，此处描述展示清空购物车方法 | | |
| 接口方法名 | emptyShopCart | |  |
| 接口方法参数 | 参数名 | 类型 | 备注 |
| 1 | UserID | Int | 用户编号 |
| 返回类型 | Void | | 成功清空该用户的购物车 |

## 3.6运行设计 （即系统运行时服务器是用的什么？浏览器用了什么？）

系统运行服务器为：Tomcat 9.0.16

浏览器：：谷歌浏览器

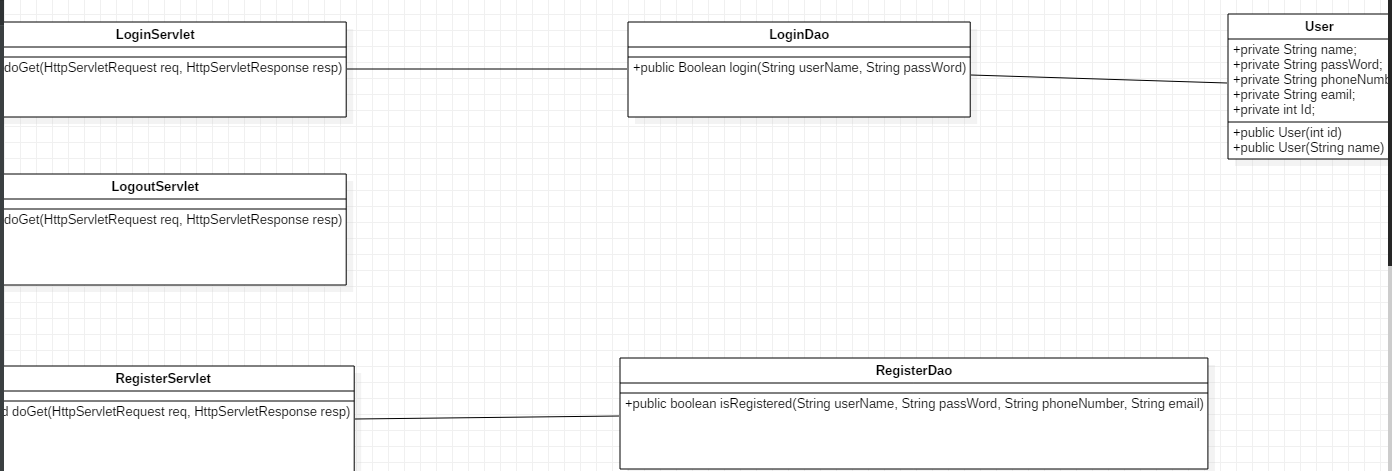
## 3.7出错处理设计 （写出出错的原因和错误的详细信息）

出错处理设计

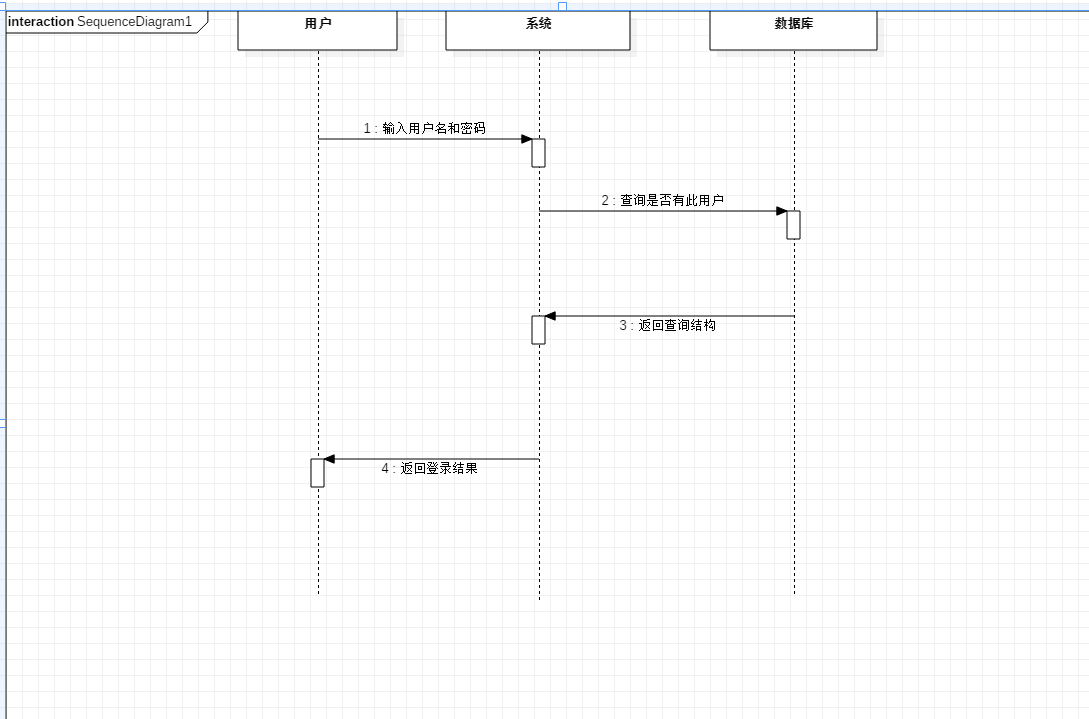
出错原因：用户注册时输入的信息不合法，如用户名或密码为空，手机号码或电子邮箱不符合规范，用户登录时输入的数据在数据库中不存在

# 4详细设计

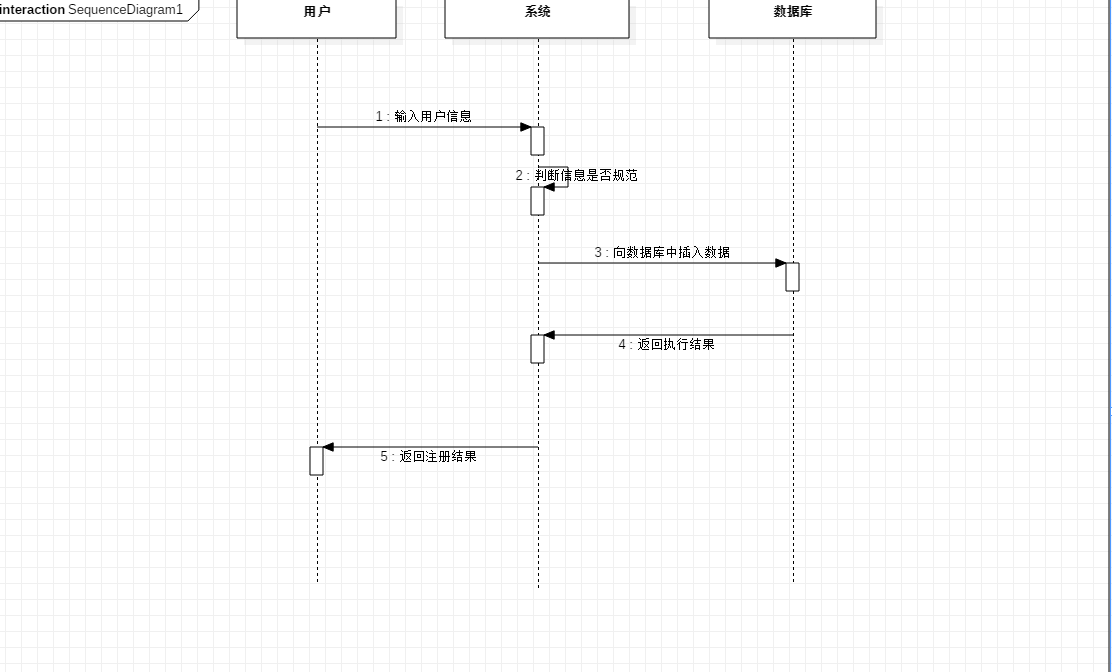
## 4.1 方法流程设计

用户管理功能：

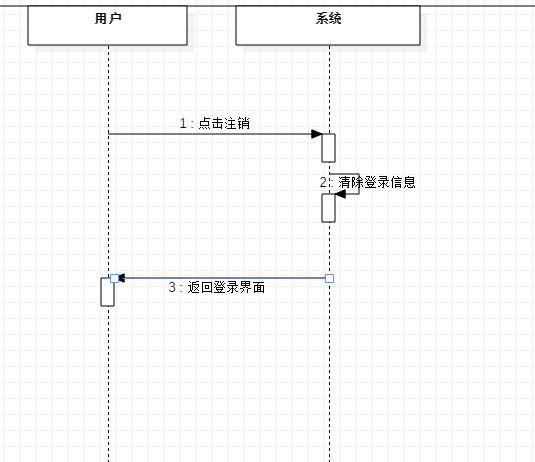
类图



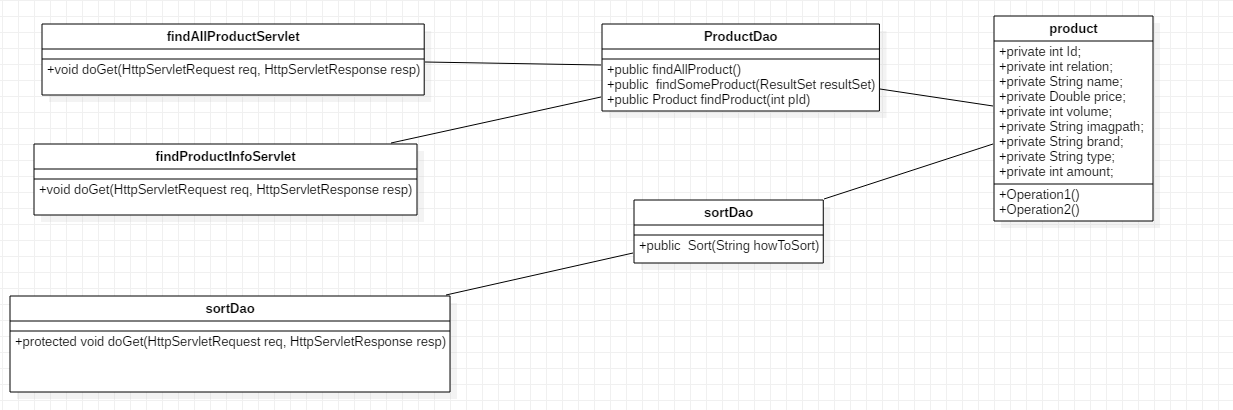
用户登录



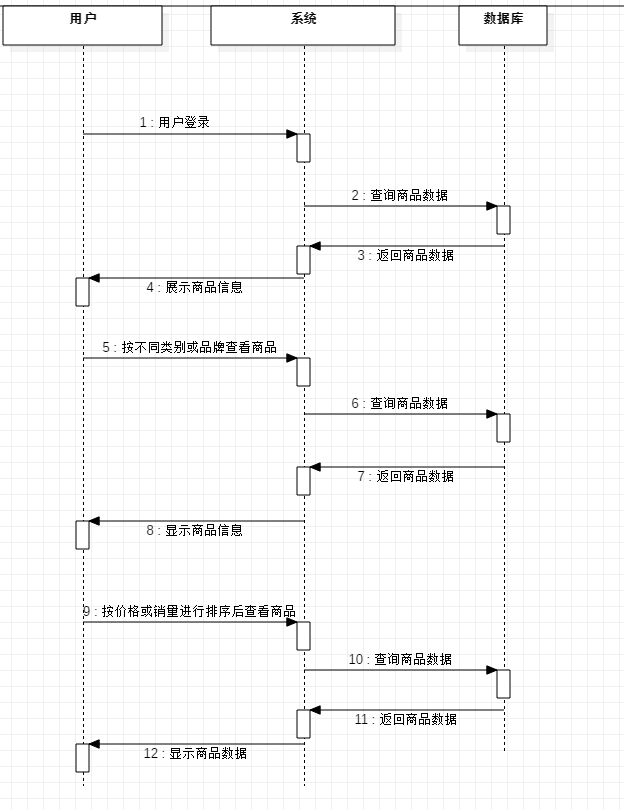
用户注册



用户注销

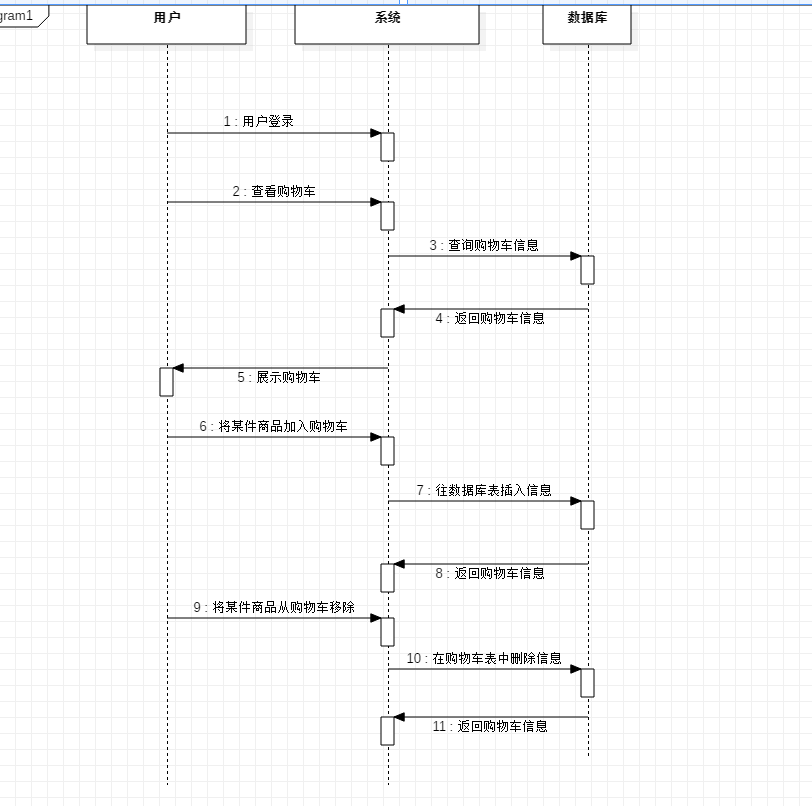
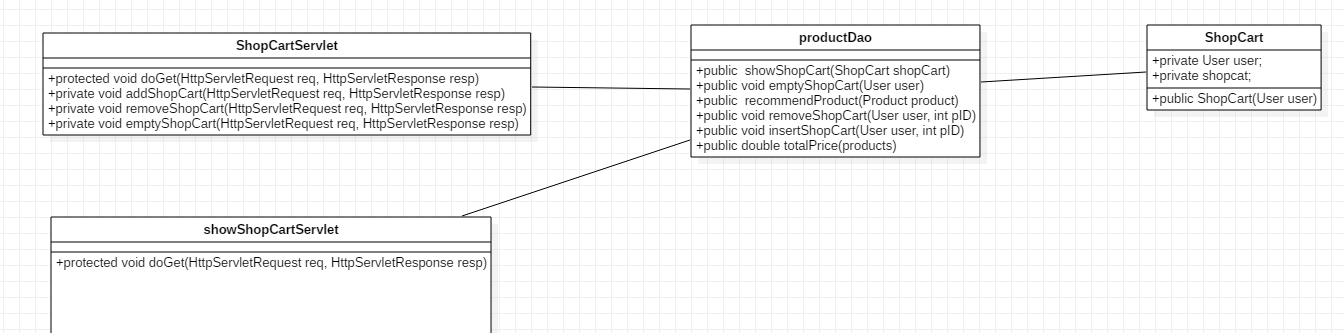
商品浏览功能：

类图



浏览商品

购物车管理功能：



**购物车操作**

# 5 编码

登录：6 结论

采用了什么技术，所设计的系统（模块）实现了哪些功能，性能怎么样。系统（模块）有哪些要改进的。

# 参考文献

[1] 贾素玲. JSP应用开发技术[M]. 北京：清华大学出版社，2007.

以上一条为例，写出实际参考的文献

# 附录

## 附录A 本系统的组成

**列出本系统Web应用程序的组成；**

## 附录B 程序源代码(经过注释的关键代码)

//根据登录名查询用户

public User findUserById(String userId) {

Connection conn = DBConnection.getConnection(); //获得连接对象

String findByIDSQL = "select \* from " +"User where userLogin = ?"; //SQL语句

PreparedStatement ps = null; //声明预处理对象

ResultSet rs = null; //声明结果集对象

User user = null; //声明用户对象

try {

ps = conn.prepareStatement(findByIDSQL); //获得预处理对象并赋值

ps.setString(1, userId); //设置第一个SQL占位符参数

rs = ps.executeQuery(); //执行查询

if(rs.next()) { //若找到用户

user = new User(); //创建用户对象

user.setUserLogin(rs.getString(1)); //设置用户登录名

user.setUserName(rs.getString(2)); //设置用户姓名

user.setUserPwd(rs.getString(3)); //设置用户密码

user.setNote(rs.getString(4)); //设置用户简介

user.setFlag(rs.getInt(5)); //设置删除状态

}

} catch (SQLException e) { //捕获异常

e.printStackTrace(); //处理异常

} finally {

DBConnection.close(rs); //关闭结果集对象

DBConnection.close(ps); //关闭预处理对象

DBConnection.close(conn); //关闭连接对象

}

return user;

}

//修改用户信息

public void updateUser(User user) {

String sql="update User set userName=?,userPwd=?,note=?,flag=? where userLogin=?";

PreparedStatement ps = null; //声明预处理对象

Connection conn = DBConnection.getConnection(); //获得连接对象

try {

ps = conn.prepareStatement(sql); /获得预处理对象并赋值

ps.setString(1,user.getUserName()); //设置第一个SQL占位符参数

ps.setString(2,user.getUserPwd());//设置第二个SQL占位符参数

ps.setString(3,user.getNote()); //设置第三个SQL占位符参数

ps.setInt(4,user.getFlag()); //设置第四个SQL占位符参数

ps.setString(5,user.getUserLogin()); //设置第五个SQL占位符参数

ps.executeUpdate();//执行更新(修改记录)

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBConnection.close(ps); //关闭预处理对象

DBConnection.close(conn); //关闭连接对象

}

}