JavaEE 课程设计报告

# 网上商城系统

基于 Spring + SpringMVC + MyBatis + MariaDB 的在线购物系统

学院：计算机科学与技术学院

专业：软件工程

班级：软件2101班

学号：20210xxxx

姓名：张三、李四、王五

指导教师：赵老师

完成日期：2025年6月24日

# 2. 任务书

## 课程设计任务

设计并实现一个基于JavaEE技术的网上商城系统，要求具备以下功能：

1. 1. 用户管理：实现用户注册、登录、个人信息管理等功能
2. 2. 商品管理：实现商品展示、搜索、详情查看等功能
3. 3. 购物车管理：实现添加商品、修改数量、删除商品等功能
4. 4. 订单管理：实现订单创建、支付、查看等功能
5. 5. 系统管理：实现基本的系统配置和维护功能

## 技术要求

1. 1. 采用MVC架构模式，使用Spring + SpringMVC + MyBatis框架
2. 2. 数据库使用MariaDB，通过MyBatis实现ORM映射
3. 3. 前端使用JSP + Bootstrap + jQuery技术栈
4. 4. 实现RESTful风格的API接口
5. 5. 代码结构清晰，注释完整，具有良好的可维护性

## 完成时间

课程设计周期为4周，要求在2025年6月30日前完成并提交报告。

## 考核方式

1. 1. 系统功能完整性：40%
2. 2. 技术实现水平：30%
3. 3. 代码质量与规范：20%
4. 4. 文档完整性：10%

指导教师：赵老师

日期：2025年5月27日

# 3. 目录

1. 封面

2. 任务书

3. 目录

4. 前言，需求分析

4.1 项目背景

4.2 需求分析

4.3 可行性分析

5. 系统设计

5.1 总体架构设计

5.2 数据库设计

5.3 功能模块设计

5.4 接口设计

5.5 安全设计

6. 系统实现

6.1 开发环境

6.2 用户模块实现

6.3 商品模块实现

6.4 购物车模块实现

6.5 订单模块实现

6.6 搜索功能实现

6.7 前端实现

7. 系统测试

7.1 测试环境

7.2 功能测试

7.3 性能测试

7.4 兼容性测试

7.5 测试结果分析

8. 总结与展望

8.1 项目总结

8.2 技术亮点

8.3 存在问题

8.4 未来展望

9. 主要源代码

9.1 核心配置文件

9.2 实体类

9.3 数据访问层

9.4 业务逻辑层

9.5 控制器层

9.6 前端页面

# 4. 前言，需求分析

## 4.1 项目背景

（由张三完成）

随着互联网技术的快速发展和电子商务的普及，网上购物已成为人们日常生活中不可或缺的一部分。本项目旨在开发一个功能完善、用户体验良好的网上商城系统，为用户提供便捷的在线购物平台，同时也为JavaEE课程设计提供一个综合性的实践项目。

本系统采用主流的JavaEE技术栈，包括Spring、SpringMVC和MyBatis框架，结合MariaDB数据库，实现一个完整的电子商务平台。通过本项目的开发，可以深入理解和掌握JavaEE技术在实际应用中的运用，提升团队成员的实践能力和项目开发经验。

## 4.2 需求分析

（由张三完成）

根据市场调研和用户需求分析，本系统需要实现以下核心功能：

### 4.2.1 用户需求

1. 1. 用户管理

* - 用户注册：新用户可以通过提供用户名、密码、邮箱等信息进行注册
* - 用户登录：已注册用户可以通过用户名和密码进行登录
* - 个人信息管理：用户可以查看和修改个人信息，包括地址、电话等
* - 密码修改：用户可以修改自己的登录密码

1. 2. 商品浏览与搜索

* - 商品列表：用户可以浏览所有商品
* - 商品分类：商品按类别进行分类展示
* - 商品搜索：用户可以通过关键词搜索商品
* - 商品详情：用户可以查看商品的详细信息，包括图片、价格、描述等

1. 3. 购物车管理

* - 添加商品：用户可以将商品添加到购物车
* - 修改数量：用户可以修改购物车中商品的数量
* - 删除商品：用户可以从购物车中删除商品
* - 清空购物车：用户可以一键清空购物车

1. 4. 订单管理

* - 订单创建：用户可以将购物车中的商品生成订单
* - 订单支付：用户可以对订单进行支付
* - 订单查询：用户可以查询自己的历史订单
* - 订单详情：用户可以查看订单的详细信息

### 4.2.2 系统需求

1. 1. 性能需求

* - 系统响应时间：页面加载时间不超过3秒
* - 并发处理能力：支持至少100个用户同时在线操作
* - 数据库性能：查询响应时间不超过1秒

1. 2. 安全需求

* - 用户密码加密存储
* - 防止SQL注入攻击
* - 防止跨站脚本攻击(XSS)
* - 敏感操作需要验证用户身份

1. 3. 可用性需求

* - 系统7\*24小时可用
* - 界面友好，操作简单直观
* - 支持主流浏览器访问

## 4.3 可行性分析

（由李四完成）

### 4.3.1 技术可行性

本项目采用成熟的JavaEE技术栈，包括Spring、SpringMVC和MyBatis框架，这些技术已经在众多大型项目中得到验证，具有良好的稳定性和可靠性。团队成员在课程学习中已经掌握了这些技术的基本应用，具备开发此系统的技术能力。

前端采用Bootstrap和jQuery等主流框架，可以快速构建响应式界面，提供良好的用户体验。MariaDB数据库作为MySQL的分支，具有良好的性能和兼容性，适合作为本系统的数据存储方案。

### 4.3.2 经济可行性

本项目作为课程设计，主要目的是学习和实践，不涉及商业运营，因此不需要考虑投资回报等经济因素。开发过程中使用的技术和工具均为开源免费软件，不需要额外的软件许可费用。

### 4.3.3 操作可行性

系统设计注重用户体验，界面简洁直观，操作流程符合用户习惯，降低了用户的学习成本。同时，系统提供详细的操作提示和帮助信息，确保用户能够顺利完成各项操作。

# 5. 系统设计

## 5.1 总体架构设计

（由李四完成）

本系统采用经典的三层架构设计，包括表示层、业务逻辑层和数据访问层，结合MVC设计模式，实现了系统的高内聚、低耦合。

### 5.1.1 系统架构图

+------------------+  
| 表示层(View) |  
| JSP + Bootstrap |  
+--------+---------+  
 |  
+--------+---------+  
| 控制层(Controller)|  
| SpringMVC |  
+--------+---------+  
 |  
+--------+---------+  
| 业务层(Service) |  
| Spring |  
+--------+---------+  
 |  
+--------+---------+  
| 数据访问层(DAO) |  
| MyBatis |  
+--------+---------+  
 |  
+--------+---------+  
| 数据库 |  
| MariaDB |  
+------------------+

### 5.1.2 技术架构

* - 前端技术：JSP、Bootstrap、jQuery
* - 后端技术：Spring、SpringMVC、MyBatis
* - 数据库：MariaDB
* - 构建工具：Maven
* - 服务器：Tomcat

## 5.2 数据库设计

（由李四完成）

### 5.2.1 ER图

系统的核心实体包括用户(User)、商品(Product)、购物车(Cart)、订单(Order)和订单项(OrderItem)，它们之间的关系如下：

* - 一个用户可以有多个购物车项
* - 一个用户可以有多个订单
* - 一个订单包含多个订单项
* - 一个商品可以出现在多个购物车中
* - 一个商品可以出现在多个订单项中

### 5.2.2 数据库表设计

1. 1. 用户表(users)

CREATE TABLE IF NOT EXISTS users  
(  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 username VARCHAR(50) NOT NULL UNIQUE COMMENT '用户名',  
 password VARCHAR(100) NOT NULL COMMENT '密码(MD5加密)',  
 email VARCHAR(100) COMMENT '邮箱',  
 phone VARCHAR(20) COMMENT '手机号',  
 address TEXT COMMENT '地址',  
 create\_time DATETIME DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间'  
) COMMENT '用户表';

1. 2. 商品表(products)

CREATE TABLE IF NOT EXISTS products  
(  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(200) NOT NULL COMMENT '商品名称',  
 description TEXT COMMENT '商品描述',  
 price DECIMAL(10, 2) NOT NULL COMMENT '价格',  
 stock INT NOT NULL DEFAULT 0 COMMENT '库存',  
 sales INT NOT NULL DEFAULT 0 COMMENT '销量',  
 image VARCHAR(500) COMMENT '商品图片',  
 create\_time DATETIME DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间'  
) COMMENT '商品表';

1. 3. 购物车表(carts)

CREATE TABLE IF NOT EXISTS carts  
(  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 user\_id INT NOT NULL COMMENT '用户ID',  
 product\_id INT NOT NULL COMMENT '商品ID',  
 quantity INT NOT NULL DEFAULT 1 COMMENT '数量',  
 create\_time DATETIME DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  
 FOREIGN KEY (user\_id) REFERENCES users (id) ON DELETE CASCADE,  
 FOREIGN KEY (product\_id) REFERENCES products (id) ON DELETE CASCADE,  
 UNIQUE KEY uk\_user\_product (user\_id, product\_id)  
) COMMENT '购物车表';

1. 4. 订单表(orders)

CREATE TABLE IF NOT EXISTS orders  
(  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 user\_id INT NOT NULL COMMENT '用户ID',  
 total\_amount DECIMAL(10, 2) NOT NULL COMMENT '总金额',  
 status VARCHAR(20) NOT NULL COMMENT '订单状态',  
 username VARCHAR(50) NOT NULL COMMENT '订单镜像 用户名',  
 address TEXT NOT NULL COMMENT '订单镜像 收货地址',  
 phone VARCHAR(20) NOT NULL COMMENT '订单镜像 手机号',  
 create\_time DATETIME DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  
 FOREIGN KEY (user\_id) REFERENCES users (id)  
) COMMENT '订单表';

1. 5. 订单项表(order\_items)

CREATE TABLE IF NOT EXISTS order\_items  
(  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 oid INT NOT NULL COMMENT '订单ID',  
 pid INT NOT NULL COMMENT '商品ID',  
 price DECIMAL(10, 2) NOT NULL COMMENT '购买价格',  
 quantity INT NOT NULL COMMENT '数量',  
 subtotal DECIMAL(10, 2) NOT NULL COMMENT '小计',  
  
 product\_name VARCHAR(200) NOT NULL COMMENT '订单镜像 商品名称',  
 product\_description TEXT COMMENT '订单镜像 商品描述',  
 product\_price DECIMAL(10, 2) NOT NULL COMMENT '订单镜像 价格',  
 product\_image VARCHAR(500) COMMENT '订单镜像 商品图片'  
) COMMENT '订单项表';