|  |  |
| --- | --- |
|  | **广州航海学院** |

课程设计（实训）报告

课 程 企业项目综合设计

题 目 快马外卖点餐系统设计与实现

专 业 软件工程

指导教师 严健武

|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | **姓名** | **角色** | **分工** |
| 例如：20220101101 | 例如：张三 | 例如：组员|组长 | 例如：注册登录；个人信息维护；历史订单维护和统计。 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

2025 年 9 月 8 日《企业项目综合设计》课程设计任务书

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 系 别 | | 软件工程 | 专 业 | 软件工程 |
| 题 目 | | 快马外卖点餐系统设计与实现 | | |
| 指导教师 | | 严健武 | | |
| 设计时间 | | 2025年 9 月 8 日 至 2025年 9 月 30 日 共40课时 | | |
| 课  程  设  计  要  求 | **一、课程设计目的**  通过本次“快马外卖点餐系统”的企业项目综合设计，使软件工程专业学生能够将所学的软件开发相关理论知识与实际项目开发相结合，熟悉软件开发的完整流程，包括需求分析、系统设计、编码实现、测试部署等环节。同时，培养学生的团队协作能力、问题解决能力以及对主流开发技术的应用能力，为后续从事实际软件开发工作奠定坚实基础。  **二、设计项目概述**  本项目旨在为专门提供中式菜品外卖的店铺定制一款 “快马外卖点餐系统”，该系统分为用户端和管理员端两大部分。用户端基于移动端浏览器，采用 Vue3 开发（熟悉微信小程序者也可选择微信小程序开发），为用户提供注册登录、个人信息维护、历史订单查看和统计、菜品浏览和点餐以及支付等功能；管理员端为 Web APP，实现用户管理、员工管理、菜品管理分类、订单查询与统计等功能。后端接口使用 Spring Boot 实现，数据库采用 MySQL，数据库框架使用 MyBatis Plus 和 MyBatis Plus Join，技术文档由 Swagger 生成，菜品数据和购物车数据使用 Redis 和 Spring Cache 实现缓存。  边学边做：参考黑马程序员的“苍穹外卖”技术方案，边学边做。  <https://www.bilibili.com/list/ml3304073830?oid=315565756&bvid=BV1TP411v7v6&p=2>  **三、设计任务(系统功能需求)**  **（一）用户端功能需求**  **1.注册登录**  用户能够通过输入用户名和密码完成注册操作，注册信息需进行合法性校验，如用户名长度、密码复杂度等。  已注册用户能够通过输入正确的用户名和密码进行登录，登录成功后进入系统首页，登录失败需给出相应提示信息。  **2.个人信息维护**  用户登录后可进入个人信息页面，查看并修改个人基本信息，如姓名、联系电话等。  用户可添加、修改、删除送货地址，添加地址时需调用高德地图 API，实现地址的选择录入，确保地址信息的准确性。  **3.历史订单查看和统计**  用户可查看自己的历史订单列表，订单列表需展示订单编号、下单时间、订单金额、订单状态等信息。  提供日期范围选择功能，用户可选择特定时间段，系统根据选择的时间段统计该时间段内的消费总额、订单数量等数据，并以图表或文字形式展示统计结果。  **4.菜品浏览和点餐**  系统按菜品类别（如热菜、凉菜、汤品、套餐等）对菜品进行分类展示，用户可点击相应类别查看该类别下的菜品单品或套餐。  每个菜品需展示菜品图片、名称、价格、简介等信息，用户可将菜品加入购物车，在购物车中可修改菜品数量或删除菜品。  **5.支付**  用户在购物车中确认订单信息后，可选择支付方式。若为基于移动端浏览器的 APP，使用支付宝沙箱功能实现支付；若为微信小程序，使用微信支付。支付过程需保证数据的安全性和稳定性，支付成功或失败均需给出明确提示，并更新订单状态。  **（二）管理员端功能需求**  **1.用户管理**  管理员可查看系统内所有用户的列表，列表展示用户 ID、用户名、注册时间、账号状态（启用 / 禁用）等信息。  管理员有权对违规用户进行账号禁用操作，禁用后用户无法登录系统；也可对已禁用的用户账号进行启用操作，恢复用户的正常使用权限。  **2.员工管理**  管理员可添加店员账号，设置店员的用户名、初始密码、所属店铺等信息，并为店员分配相应的操作权限（如接单、拒单、处理配送状态等）。  管理员可查看所有店员的列表信息，包括店员 ID、用户名、权限范围、账号状态等，同时可对店员账号进行修改、删除或重置密码等操作。店员登录系统后，可处理用户订单，包括接单、拒单操作，并能手动更新订单的配送状态（如已接单、配送中、已送达等）。  **3.菜品管理分类**  单品/套餐分类管理：管理员可添加新的菜品分类（如新增 “特色小吃” 分类），设置分类名称、分类排序等信息；也可对已有的分类进行修改名称、调整排序或删除（删除分类需确保该分类下无菜品）操作。  菜品单品管理：管理员可在相应的菜品分类下添加菜品单品，录入菜品名称、价格、图片、简介、库存数量、是否上架等信息；可对已添加的菜品单品信息进行修改，如调整价格、更新库存、更换图片等；还可对下架的菜品进行上架操作，或对热销菜品进行优先展示设置。  套餐管理：管理员可创建菜品套餐，选择套餐内包含的菜品单品及对应的数量，设置套餐价格（套餐价格可低于套餐内各单品价格之和，以吸引用户）、套餐图片、套餐简介、是否上架等信息；可对已创建的套餐进行修改或下架操作。  **4.订单查询与统计**  订单查询：管理员可根据订单编号、下单时间范围、订单状态（取消 / 等待中/已发货/已完成等）、用户信息等条件查询相应的订单列表，订单列表展示订单详细信息，如订单编号、用户信息、下单时间、菜品信息、订单金额、支付状态、配送状态等。  订单统计：管理员可选择特定的日期范围，系统统计该时间段内已完成订单的总数量、总金额、平均订单金额等数据；同时可分别查看取消订单、等待中订单、已发货订单的数量及相关详细信息，并以报表或图表形式展示统计结果，便于管理员了解店铺的经营状况。  **四、技术要求**  **前端技术**  用户端：使用 Vue3 框架进行开发，若选择微信小程序开发，则需遵循微信小程序的开发规范和技术要求。需合理使用 Vue3 的 Composition API、组件化开发思想，实现页面的布局和交互功能，确保页面响应式设计，适配不同尺寸的移动设备屏幕。同时，集成高德地图 API 实现地址选择功能，集成支付宝沙箱 SDK（或微信支付 SDK）实现支付功能。  管理员端：采用 Web APP 开发模式，选择Vue3+ElementUI或AntD，实现页面的美观布局和便捷操作，确保管理员能够高效地进行系统管理工作。  **后端技术**  使用 Spring Boot 框架搭建后端项目，实现 RESTful 风格的 API 接口，为前后端交互提供数据支持。  集成 Spring Security 安全框架，实现用户认证和授权功能，确保系统的安全性，防止未授权访问。  数据库采用 MySQL，使用 MyBatis Plus 和 MyBatis Plus Join 作为数据库框架，简化数据库操作，提高开发效率。需设计合理的数据库表结构，确保数据的完整性和一致性。  使用 Redis 和 Spring Cache 实现缓存功能，将菜品数据和购物车数据缓存到 Redis 中，减少数据库访问次数，提高系统的响应速度。  集成 Swagger 生成 API 文档，方便前后端开发人员进行接口对接和测试，文档需清晰展示接口的请求参数、响应数据格式、接口功能说明等信息。  **五、开发小组组成及分工建议**  开发小组由 5 人组成，具体分工如下：  **组长（1 人）**：负责整体任务安排、项目进度把控，兼后端开发工作，为系统添加 Spring Security 安全框架，在项目各模块开发完成后进行系统整合，确保系统各部分能够协调运行。  **前端用户端开发（2 人）**  **成员A**：负责实现用户端功能（1）注册登录（用户名和密码注册和登录）、（2）个人信息维护（填写必要的个人信息，其中送货地址使用高德地图选择录入）、（3）历史订单查看和统计（根据日期范围进行消费查看和统计）。  **成员B**：负责实现用户端功能（4）菜品浏览和点餐（用户可以按分类浏览菜品单品或套餐，可以放入购物车）、（5）支付（使用支付宝沙箱功能实现支付，如果是微信小程序，则使用微信支付）。  **前端管理员端开发（3 人）**  **成员C：**负责实现管理员端功能（1）用户管理（禁用或启用用户账号）、（2）员工管理（分配员工账号，店员可以处理订单，即接单 / 拒单；手动处理配送状态）。  **成员D：**负责实现管理员端功能（3）菜品管理分类（单品 / 套餐分类管理；菜品单品管理；套餐管理）。  **成员E：**负责实现管理员端功能（4）订单查询与统计（查询和统计日期范围内的已完成订单的数据；查看取消 / 等待中 / 已发货的订单数据）。 六、进度安排  | **阶段** | **时间** | **主要任务** | | --- | --- | --- | | 第一阶段 | 6学时 | 需求分析与系统设计，明确系统功能需求和非功能需求，进行系统架构设计、数据库设计、模块划分，确定技术方案和开发计划，完成需求分析文档和系统设计文档。 | | 第二阶段 | 16学时 | 各模块编码实现，前端开发人员完成用户端和管理员端页面布局和交互功能开发，后端开发人员（组长）完成后端 API 接口开发、Spring Security 安全框架集成、Redis 缓存功能实现等，各成员定期沟通开发进度，解决开发过程中遇到的问题。 | | 第三阶段 | 6学时 | 系统测试与修改，对系统进行单元测试、集成测试和功能测试，根据测试结果修复系统中的 bug 和问题，优化系统性能，确保系统功能的正确性和稳定性。 | | 第四阶段 | 6学时 | 系统整合与部署，组长负责将各模块进行系统整合，确保系统各部分能够协调运行，完成系统部署工作，生成系统部署文档。 | | 第五阶段 | 6学时 | 文档整理与成果提交，整理系统源代码、数据库脚本、技术文档、测试报告、部署文档和总结报告等交付成果，进行最终检查和完善，按时提交所有交付成果。 |   **七、设计成果提交（**提交截止时间：2025-9-30）  **1.系统源代码**  完整的 “快马外卖点餐系统” 源代码，包括前端用户端、前端管理员端和后端代码，代码需结构清晰、注释完整，符合软件开发规范。  系统数据库脚本，包括数据库表创建语句、初始数据插入语句等，确保能够顺利搭建系统数据库环境。  **2.各阶段设计文档**  系统技术文档，由 Swagger 生成，包含所有 API 接口的详细说明，同时需补充系统的整体设计文档，包括需求分析、系统架构设计、数据库设计、模块设计等内容。  系统测试报告，包括测试用例、测试过程、测试结果等，测试用例需覆盖系统的主要功能点，确保系统功能的正确性和稳定性。  系统部署文档，详细说明系统的部署环境要求、部署步骤、注意事项等，确保能够顺利将系统部署到实际运行环境中。  课程设计总结报告，每个小组成员需提交个人总结报告，总结在课程设计过程中的工作内容、遇到的问题及解决方法、收获与体会等。  **3.课程设计报告**  每个小组统一提交1份设计报告包括：需求分析、用例分析与数据库设计、详细设计、系统功能实现、测试、总结（总结项目开发的整体情况、团队协作情况、项目存在的不足及改进方向等）。  每个小组提供单独的接口文档。  **最后，将以上所有文档和源代码作为压缩包提交，压缩包命名格式为：班级-组号（2位）组.rar，例如：软工222-01组.rar。** | | | |

|  |
| --- |
| 《企业项目综合设计》是软件工程专业一个重要的综合实践环节，本课程通过一个综合性的企业级软件项目，在企业导师和老师的指导下，进行项目实战训练及软件工程职业实践，在项目实践中巩固技术、增长能力、提升素质、丰富经验，真正达到具有解决工程实践问题能力的目标。采用“项目小组”的形式，结合具体的项目按照企业软件工程规范开发一个软件系统。培养学生按照软件工程的原理、方法、技术、标准和规范，进行软件开发和软件项目管理的能力。培养学生解决工程实践问题的能力，对技术文档的编写能力，以及合作意识和团队精神。 |
| **第二部分：实训准备(包含实训设备及器材)**   * 机房提供电脑，或自备笔记本电脑。 * 操作系统：Windows7以上。 * 数据库：MySQL，或 SQL Server等。 * 开发工具和运行环境：IntelliJ IDEA、VSCode、Vue.js等。 |
| **第三部分：实训要求**  1.提前5-10分钟到达机房，做好实训前的准备工作，不得迟到、早退。如有特殊情况不能按时到课，需提前向组长或老师请假并说明原因。  2.保持机房安静，不得大声喧哗、打闹、随意走动或交头接耳，以免影响他人学习。有问题需要讨论时，应尽量小声，且不得干扰其他同学的正常操作。  3. 实验过程中，应集中精力，认真完成实验任务，不得浏览与实验无关的网页、玩游戏、观看视频等，严禁从事与实验教学无关的活动。  4. 因病、因事不能参加实训的同学，需提前向任课教师提交书面请假条，并附上相关证明材料（如病假需提供医院诊断证明，事假需提供家长签字的请假条等），经任课教师批准后方可请假。未经批准擅自缺课者，按旷课处理。  5. 未经教师允许，不得擅自更改设备的配置和设置，不得私自拆卸、安装或移动设备。  6. 注意用电安全，不得随意插拔电源插头、网线等，不得在机房内使用任何未经允许的电器设备。当次实训结束后，应按照正确的顺序关闭计算机及相关设备，并切断电源，确保设备处于安全状态。  7. 保持机房内的环境卫生，不得在机房内吃东西、喝饮料、乱扔垃圾。当次实训结束后，应将座位周围的垃圾清理干净，将椅子摆放整齐，确保机房整洁有序。 |
| **第四部分：实训内容(步骤及程序)** |
| **第五部分：实训结果(结果分析)** |
| **第六部分：实训小结(心得及体会)** |
| **【说明：第四、五、六部分独立分节，见下文】** |

**目录**

[1 需求分析（所有红色字体，成文后删除） 1](#_Toc200914400)

[1.1 功能需求分析 1](#_Toc200914401)

[1.2 非功能需求分析 1](#_Toc200914402)

[2 用例文档与数据库设计 2](#_Toc200914403)

[2.1 用例文档 2](#_Toc200914404)

[2.1.1 登录用例文档 2](#_Toc200914405)

[2.1.3 商品展示用例文档 3](#_Toc200914406)

[2.1.4 商品详情用例文档 3](#_Toc200914407)

[2.1.5 购物车用例文档 3](#_Toc200914408)

[2.1.6 XXX用例文档 3](#_Toc200914409)

[2.1.7 XXX用例文档 3](#_Toc200914410)

[2.2 数据库设计 3](#_Toc200914411)

[2.2.1 概念模型设计 3](#_Toc200914412)

[2.2.2 数据库表设计 3](#_Toc200914413)

[3 系统设计 6](#_Toc200914414)

[3.1 登录模块详细设计 6](#_Toc200914415)

[3.2 商品展示模块详细设计 6](#_Toc200914416)

[3.3 商品详情模块详细设计 6](#_Toc200914417)

[3.4 购物车模块详细设计 7](#_Toc200914418)

[3.5 XXXX模块详细设计 7](#_Toc200914419)

[3.6 XXXX模块详细设计 7](#_Toc200914420)

[3.7 XXXX模块详细设计 7](#_Toc200914421)

[4 系统实现 8](#_Toc200914422)

[4.1 登录模块的实现 8](#_Toc200914423)

[4.2 商品展示模块的实现 9](#_Toc200914424)

[4.3 商品详情模块的实现 9](#_Toc200914425)

[4.4 购物车模块的实现 9](#_Toc200914426)

[4.5 XXX模块的实现 9](#_Toc200914427)

[4.6 XXX模块的实现 10](#_Toc200914428)

[4.7 XXX模块的实现 10](#_Toc200914429)

[5 系统测试 11](#_Toc200914430)

[5.1 测试环境和测试方法 11](#_Toc200914431)

[5.2 测试用例和测试结果 11](#_Toc200914432)

[5.2.1 登录功能测试 11](#_Toc200914433)

[5.2.2 商品展示模块的功能测试 11](#_Toc200914434)

[5.2.3 商品详情模块的功能测试 12](#_Toc200914435)

[5.2.4 购物车模块的功能测试 12](#_Toc200914436)

[5.2.5 XXXX模块的功能测试 12](#_Toc200914437)

[5.2.6 XXXX模块的功能测试 12](#_Toc200914438)

[5.2.7 XXXX模块的功能测试 12](#_Toc200914439)

[结论（总结） 13](#_Toc200914440)

[小结 14](#_Toc200914441)

1 需求分析（所有红色字体，成文后删除）

系统的实现来源于用户的需求，通过需求分析，对用户的需求进行归纳、总结得出需求模型，为系统实现提供有力支持。

## 1.1 功能需求分析

经过需求分析，本系统应该……

提示：功能需求分析要画用例图（面向对象UML建模）。用例图如图1.1所示。

**图1.1 系统用例图**

## 1.2 非功能需求分析

提示：描述系统的软硬件运行环境：包括客户端和服务器端的开发与运行的平台和环境、开发语言、数据库管理系统软件等。描述性能需求等。

2 用例文档与数据库设计

## 2.1 用例文档

提示：通过用例文档对用例图中各个用例进行业务分析和描述，可在用例文档中使用类图、活动图或顺序图等加以说明。

### 2.1.1 登录用例文档

用例文档的描述格式，参考表1。

**表2.1 “登录”用例文档**

|  |  |
| --- | --- |
| 描述项 | 说明 |
| **用例名称** | 登录 |
| **用例描述** | 用户利用该用例登录系统，通过身份认证后获得相应的操作权限 |
| **参与者** | 用户（后端管理员、前端用户在此被统称为“用户”） |
| **相关用例** | 无 |
| **前置条件** | 无（访问用例前必须满足的条件列表） |
| **后置条件** | 如果登录成功，则显示相应权限的操作界面（用例完成以后得到满足的条件列表） |
| **基本事件流** | (1)用例起始于用户需要登录到该系统  (2)系统显示欢迎界面,并要求用户输人用户名和密码  (3)用户输入用户名和密码  (4)系统验证用户名和密码,允许用户登录系统(A-1)  (5)系统根据用户类型启动不同的主操作界面  （描述用例中各项工作都正常进行时用例的工作方式） |
| **备选事件流** | A-1用户名错误或密码错误  (1)系统显示用户名错误或密码错误的提示信息,并进人第(2)步  (2)用户可以重新输入用户名和密码(B-1),也可以选择结束该用例  （描述变更工作方式.出现异常或发生错误的情况下所遵循的路径） |
| **补充约束-业务规则** | B-1系统允许用户重试3次登录操作,超过3次后系统自动结束,不允许用户重试 |
| **补充约束-非功能需求** | 安全性:密码应该采用加密的方式存储,有关密码的加密算法待定 |
| **补充约束-数据需求** |  |
| **频率[可选]** | （参与者访问此用例的频率，如用户是每日访问一次还是每月访问一次） |
| **待解决问题** | 关于用户名和密码的管理与维护功能还需要进一步明确 |
| **相关图** | 无(可画出登录过程的活动图,此处省略) |

### 2.1.3 商品展示用例文档

### 2.1.4 商品详情用例文档

### 2.1.5 购物车用例文档

### 2.1.6 XXX用例文档

### 2.1.7 XXX用例文档

## 2.2 数据库设计

为了实现系统的各项功能，必须有数据的支撑。基于所需功能对数据保存的需求，设计出数据模型，并创建数据库表。

### 2.2.1 概念模型设计

提示：画局部E-R图和全局E-R图。例如：本系统的全局E-R图如图2.1所示。

**图2.1 全局E-R图**

### 2.2.2 数据库表设计

例如：系统数据库中总共定义了……张表，根据数据库的E-R结构模型的分析结果，转换为各个数据基本表如表2.2至表……所示。

**表2.2 XXX表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 描述 | 是否为空 | 是否主键 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**表2.3 XXX表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 描述 | 是否为空 | 是否主键 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**表x XXX表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 描述 | 是否为空 | 是否主键 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**表x XXX表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 描述 | 是否为空 | 是否主键 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**表x XXX表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 描述 | 是否为空 | 是否主键 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**表x XXX表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 描述 | 是否为空 | 是否主键 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

3 系统设计

提示：详细设计：顺序图、活动图、构件图等（面向对象UML建模）；或者程序流程图、PAD图等（结构化分析设计）。

## 3.1 登录模块详细设计

例如：客户端登录功能：在此模块中，用户可以通过输入正确的登录账号和登录密码，再点击登陆按钮实现登陆操作。在登录模块中，每个用户都拥有独一无二的登录账号和登录密码，且数据存储在服务端的MySql数据库服务器中。如果用户为新用户，则首先需要通过注册模块获得账号。

在登陆界面，用户输入登录账号密码之后，系统会对用户输入的账号密码进行判断是否非空，如果数据为空，则出现提示信息以弹框形式提醒用户，如果不为空，则会通过URL访问系统服务端的MySql数据库，与数据库用户表数据进行比较，比对成功则成功跳转进入系统主页面。登录模块的顺序图如图3所示。

**图3.1 登录模块的顺序图**

## 3.2 商品展示模块详细设计

作为主页，所有用户进入网站时度可以浏览器商城展示的商品，并可以查询商品。商品展示的活动图如图3.2所示。

**图3.2 商品展示活动图**

## 3.3 商品详情模块详细设计

该模块用于查看所选中商品的详细系信息。商品详情的通信图如图3.3所示。

**图3.3 查商品详情通信图**

## 3.4 购物车模块详细设计

已登录用户可以购买商品，即将商品放入购物，并可以展示购物车的商品（订单），下面对购物车部分功能进行建模。查询订单信息的活动图如图3.4所示。

**图3.4 订单信息查询活动图**

其他功能模块自行完善，例如针对管理员的后台管理系统，对商品信息进行增删改查操作。

## 3.5 XXXX模块详细设计

## 3.6 XXXX模块详细设计

## 3.7 XXXX模块详细设计

4 系统实现

## 4.1 登录模块的实现

提示：用语言描述各模块实现的功能简介。

界面截图（注意，每张图都需要标识图题，例如图x）

核心关键代码，需要插入代码的部分，应该背景设为灰色，如下所示。

<!--保存页面模块-->

<action name="saveVideo"

</action>

例如：用户拥有账号后才可完成登录操作。用户登录界面如图4.1所示。



**图4.1 用户登录界面图**

核心关键代码，如下所示。

try { // 连接数据库

con = DriverManager.getConnection(url, user, password);

String queryUser = "select \* from userinfo where account=" + account + " and pwd=" + pwd;

String queryCart = "select \* from cartinfo where account="+account;

String queryCom = "select \* from commodityinfo where comId=";

pre = con.prepareStatement(queryUser);

rs = pre.executeQuery();

session.setAttribute("class", " ");

// 匹配成功则更新userBean的信息，并跳转至main.jsp

if (rs.next()) {

UserBean userBean = (UserBean) session.getAttribute("userBean");

if(userBean == null) {

userBean = new UserBean();

session.setAttribute("userBean", userBean);

}

userBean.setAccount(account);

userBean.setName(account);

Updata.userBeanUpdata(userBean);

session.removeAttribute("tip");

RequestDispatcher dispatcher = request.getRequestDispatcher("main.jsp");

dispatcher.forward(request, response);

} else {

// 匹配失败则回到登录页面并给出提示信息

session.setAttribute("tip", "账号或密码错误");

RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");

dispatcher.forward(request, response);

}

} catch (Exception e) {

session.setAttribute("class", " ");

session.setAttribute("tip", e.toString());

RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");

dispatcher.forward(request, response);

}

## 4.2 商品展示模块的实现

## 4.3 商品详情模块的实现

## 4.4 购物车模块的实现

## 4.5 XXX模块的实现

## 4.6 XXX模块的实现

## 4.7 XXX模块的实现

5 系统测试

任何一个新开发的软件都需要经过系统且全面的测试，这样的测试最主要的是为了验证所开发出来的系统是否与预期的客户需求相一致，以及提前发现软件本身存在的缺陷。当然测试仅仅只是开发软件本身的一环，并不是最终目的。软件测试人员应当以一个正常用户的角度去测试，在用户必用或常用的功能上，多做测试。

## 5.1 测试环境和测试方法

## 5.2 测试用例和测试结果

### 5.2.1 登录功能测试

例如：用户登录时，当输入登录账号或者密码为空时，会出现密码或账号为空提示。当输入账号密码与服务端数据库用户表相应数据不匹配时，会出现密码或账号错误提示。系统的用户登录功能的黑盒测试的测试用例设计如表5.1所示。

**表5.1 登录功能测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 账号 | 密码 | 预期结果 | 测试结果 |
| dl-1 | abc | 输入正确密码 | 登录成功 | 通过 |
| dl-2 | abc | 888888（错误登录密码） | 密码输入错误提示 | 通过 |
| dl-3 | abc | （为空） | 登录密码为空提示 | 通过 |

### 5.2.2 商品展示模块的功能测试

商品维护的测试用例以及测试结果如表5.2所示。

**表5.2 育秧记录测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 操作 | 输入数据 | 预期结果 | 测试结果 |
| wh-1 | 新增记录 | 输入新增记录 | 新增记录成功 | 通过 |
| wh-2 | 删除记录 | 无 | 删除记录成功 | 通过 |
| wh-3 | 修改记录 | 输入修改记录 | 修改记录成功 | 通过 |
| wh-4 | 查询记录 | 输入查询记录条件 | 若记录存在则显示，否则无显示 | 通过 |

### 5.2.3 商品详情模块的功能测试

### 5.2.4 购物车模块的功能测试

### 5.2.5 XXXX模块的功能测试

### 5.2.6 XXXX模块的功能测试

### 5.2.7 XXXX模块的功能测试

结论（总结）

简明概要的一段话，总结本系统满足了用户的哪些需求，实现了哪些功能，是否达到了预期结果。系统有哪些经济和社会效益或价值。

小结

主要负责的工作内容；开发过程中遇到的问题及其解决方法：（简明概要地描述，并说明个人的研发能力是否得到了提升）；自我评价；

老师评语：

严健武

2025年 6月 30 日