深圳大学考试答题纸

(以论文、报告等形式考核专用)

二○ 二三 ～二○ 二四 学年度第 一 学期

课程编号	1500720002	课程名称	软件工程		主讲教师	朱安民	评分
	2021192010		王曦		2021 级信息与计算科学（数学与计算机实验班）		
	2021150024		张艺轩		2021 级计算机科学与技术（数学与计算机实验班）		
	2021280462		叶可禾		2021 级软件工程		
学 号	2021040074	姓名	陈婷	专业年级	2021 级软件工程		

教师评语：

题目： 《冰酒销售计算机系统管理系统》

组员名单（学号及姓名）、签名：

学号	姓名	签名
2021192010	王曦	
2021150024	张艺轩	
2021280462	叶可禾	
2021040074	陈婷	

说明：

本课程以大作业的成绩代替考试方式。即以一个软件开发项目为实例，按照软件项目的实际开发过程，分成不同的阶段，贯穿于整个教学过程。在不同的阶段，学生都有机会完成相应的阶段任务。学生自始至终在不同的阶段都围绕着软件工程的概念思考，并提交相应阶段的阶段报告。课程结束之时，提交总体报告，这也就意味着学生实际完成了一个实际的软件开发项目。

评分根据是：1）系统分析报告；2）系统原型建立报告；3）系统设计实现报告；4）课堂演示评价；5）本次提交的系统开发汇总报告。

冰酒销售管理系统

开发报告

课程：软件工程

小组名：壹零空

小组成员：王曦、张艺轩、叶可禾、陈婷

提交时间：2024 年 01 月 02 日

目录

修订日志 ..... **vi**
1. 前言 **7**
1.1 目的 7
1.2 文档约定 7
1.3 目标读者和阅读建议 7
2. 简介 **8**
2.1 功能概述 8
2.2 用户界面 8
2.3 数据安全与隐私 8
2.4 预期效益 8
2.5 结论 8
3. 目标 **9**
3.1 系统功能定义与规范 9
3.2 用户界面设计 9
3.3 数据安全和隐私 9
3.4 系统可靠性与性能 9
3.5 用户需求确认 9
3.6 技术可行性验证 9
3.7 用户反馈和改进 10
3.8 项目可行性评估 10
3.9 为开发团队提供指导方 10
3.10 制定下一步计划 10
4. 范围 **11**
4.1 系统概述 11
4.2 用户群体 11
4.3 功能范围 11
4.4 技术架构 11
4.4.1 技术框架 11
4.4.2 技术组件 12
4.5 安全和合规性 13
4.6 限制和假设 13
4.7 未来发展 13
5. 定义及缩略语 **14**
5.1 冰酒 (Ice Wine) 14
5.2 销售管理系统 (Sales Management System, SMS) 14
5.3 客户关系管理 (Customer Relationship Management, CRM) 14
5.4 库存管理系统 (Inventory Management System, IMS) 14
5.5 用户界面 (User Interface, UI) 14
5.6 应用程序接口 (Application Programming Interface, API) 14
5.7 数据库管理系统 (Database Management System, DBMS) 14
5.8 GDPR (General Data Protection Regulation) 15
5.9 AI (Artificial Intelligence) 15
5.10 电子商务 (E-commerce) 15
6. 需求分析报告 **16**
6.1 总述 16
6.1.1 产品视角 16
6.1.2 产品特点 16
6.1.3 用户类别和特点 16
6.1.4 操作环境 17
6.1.5 设计与实现限制 17
6.1.6 用户文档 17

6.1.7 假设和依赖性.....18
6.2 外部接口需求.....18
6.2.1 用户界面.....18
6.2.2 硬件接口.....18
6.2.3 软件接口.....19
6.2.4 通信接口.....19
6.3 系统功能需求.....20
6.3.1 产品图片的显示和浏览.....20
6.3.2 销售订单管理.....20
6.3.3 库存管理.....21
6.4 其它非功能需求.....21
6.4.1 性能要求.....21
6.4.2 安全要求.....22
6.4.3 数据保护要求.....22
6.4.4 软件质量属性.....23
6.4.5 业务规则.....24
6.4.6 权限级别.....24
6.5 其它要求.....25
6.5.1 数据库要求.....25
6.5.2 国际化要求.....25
6.6 分析模型.....26
6.6.1 数据流图（DFD）.....26
6.6.2 类图.....27
6.6.3 状态转换图.....28
6.6.4 实体关系图（ERD）.....28
6.7 待定部分.....29
7. 系统原型报告.....31
7.1 登录页面.....31
7.2 注册页面.....31
7.3 推荐页面.....32
7.4 全部商品页面.....33
7.5 购物车页面.....34
7.6 支付页面.....35
7.7 我的订单页面.....36
7.8 公司介绍页面.....37
7.9 我的页面.....38
7.10 商品详情页面.....39
7.11 员工管理页面.....40
7.12 产品管理页面.....41
7.13 订单分析页面.....42
7.14 新增产品页面.....43
7.15 新增员工页面.....44
7.16 推广管理页面.....45
7.17 总览.....46
8. 系统设计与实现报告.....47
8.1 用户管理 UML 图.....47
8.1.1 用例图.....47
8.1.2 类图.....48
8.1.3 顺序图：用户登录.....49
8.1.4 活动图：用户登录.....50
8.2 订单管理 UML 图.....51
8.2.1 用例图.....51

8.2.2 类图 52
8.2.3 顺序图：订单创建与处理 53
8.2.4 活动图：订单创建与处理 54
8.3 产品管理 UML 图 56
8.3.1 用例图 56
8.3.2 类图 57
8.3.3 顺序图 58
8.3.4 活动图 60
8.4 库存管理 UML 图 62
8.4.1 用例图 62
8.4.2 类图 63
8.4.3 顺序图 64
8.4.4 活动图 65
8.5 系统实现 67
8.5.1 前端 67
8.5.2 数据库 67
8.5.3 后端 67
8.5.4 维护 68
8.5.5 单元测试 69
8.5.6 集成测试 69
8.5.7 性能测试 70
8.5.8 云服务 70
9. 系统测试 71
9.1 系统测试计划 71
9.1.1 范围 71
9.1.2 参考 71
9.1.3 测试计划 71
9.1.4 测试说明 74
9.2 系统测试报告 114
9.2.1 范围 114
9.2.2 参考 114
9.2.3 测试结果概述 114
9.2.4 详细测试结果 115
9.2.5 评价 201
10. 系统使用说明书 203
10.1 前台 203
10.1.1 注册 203
10.1.2 登录 203
10.1.3 首页 204
10.1.4 商品详情页 205
10.1.5 购物车 205
10.1.6 我的订单 206
10.1.7 商品筛选页 208
10.1.8 支付界面 209
10.2 后台 210
10.2.1 管理首页 210
10.2.2 分类管理 210
10.2.3 产品管理 212
10.2.4 用户管理 213

10.2.5 订单管理 213
10.2.6 我的账户 215
10.2.7 登录 215
11. 参考 217
11.1 参考资料 217
11.2 技术文档 217
11.3 研究和报告 217
11.4 相关法规 217
附录 219
附录 1 小组成员及贡献比例 219

修订日志

姓名	Date	Reason For Changes	Version
小组全体成员	2024.01.02	初稿 1.0	初稿

1. 前言

1.1 目的

冰酒销售管理系统旨在帮助农户冰酒提供商管理和跟踪其冰酒销售的各个方面。

本报告旨在描述和概括冰酒销售管理系统的需求分析、系统原型设计、系统实现、系统测试的软件开发全过程。

1.2 文档约定

在本软件需求规范的最新版本中，没有文档约定。

1.3 目标读者和阅读建议

目标读者是将负责实施和测试冰酒销售管理系统的软件工程课程学生。此外，本文档将供软件工程教授使用，用于评估软件的设计和功能。阅读建议为学生应充分理解系统的需求，以便有效地进行开发和测试工作。教授应使用本文档来监督和指导学生的项目，并评估其在满足需求方面的成功程度。

2. 简介

2.1 功能概述

该系统将包括以下核心功能：

① 库存管理：追踪冰酒库存，包括各个批次的数量、存放位置和有效期。
② 订单处理：处理客户订单，生成发货单和发票，跟踪订单状态并支持支付和退款。
③ 客户管理：记录客户信息、购买历史和偏好，以提供个性化服务和营销。
④ 报告与分析： 提供销售数据、库存状况和客户趋势等报告，以支持决策和策略制定。

2.2 用户界面

系统将提供直观友好的用户界面，易于导航和使用，支持多平台访问，包括桌面和移动设备。

2.3 数据安全与隐私

保障数据安全是该系统设计的重点，包括加密敏感信息、权限管理和备份策略等措施，以确保客户信息和交易数据的安全性和保密性。

2.4 预期效益

通过该系统的实施，预期将提高销售效率，优化库存管理，改善客户服务，并为决策者提供更准确的数据支持。

2.5 结论

本报告介绍了一个冰酒销售管理系统的原型设计，强调了系统的核心功能、用户界面、数据安全性以及预期效益，以期为设计与开发提供一个全面的指导方向。

3. 目标

3.1 系统功能定义与规范

详细定义系统所需的功能，包括库存管理、订单处理、客户管理和报告分析等，确保这些功能满足用户需求。

3.2 用户界面设计

设计易于使用、直观友好的用户界面，使用户能够轻松浏览、操作系统功能，并实现跨平台访问。

3.3 数据安全和隐私

确保系统具备强大的数据加密、权限管理和备份策略，保障客户数据和交易信息的安全性和保密性。

3.4 系统可靠性与性能

着重于系统的稳定性和高性能，确保系统能够长时间稳定运行，处理大量数据并保持高效率。

3.5 用户需求确认

通过原型设计，验证和确认用户需求，以确保系统满足用户期望并解决他们的业务挑战。

3.6 技术可行性验证

确保所选技术和架构符合项目需求，考虑系统的扩展性和未来发展。

3.7 用户反馈和改进

利用原型阶段收集用户反馈，以便在后续开发阶段进行改进和优化，确保系统符合用户期望。

3.8 项目可行性评估

综合考虑资源投入、时间安排和预期效益，评估项目实施的可行性和潜在风险。

3.9 为开发团队提供指导方

提供明确的原型设计，作为开发团队的指导方向，确保开发工作与最终期望的系统一致。

3.10 制定下一步计划

根据原型设计结果和评估，制定明确的下一步行动计划，包括开发、测试和实施等阶段的计划安排。

4. 范围

4.1 系统概述

冰酒销售管理系统旨在提高销售效率，简化库存管理，以及提供数据分析以支持决策过程。

4.2 用户群体

冰酒销售管理系统面向冰酒制造商、分销商、零售商、顾客。

4.3 功能范围

① 销售管理：包括订单处理、发票生成、支付处理等功能。
② 库存管理：追踪冰酒库存状态，自动补货提醒等。
③ 客户关系管理：客户信息管理，购买历史记录，客户服务等。
④ 数据分析与报告：销售数据分析，库存状况报告，市场趋势预测等。
⑤ 多平台接入：系统能够支持不同的操作平台，如电脑端、手机端等。

4.4 技术架构

4.4.1 技术框架

（1）前端：

① JSP (JavaServer Pages)
② JQuery (Cross-platform JavaScript Library)

（2）数据库：

MySQL

（3）后端：

① Java
② Spring

③ MyBatis
④ MVC 模式 (Model - View - Controller)

（4）维护：

① 版本管理：Git
② 代码托管：Github

（5）单元测试：

① JUnit Jupiter 框架
② AAA 方法 (Arrange – Act - Assert)

（6）集成测试：

① Spring Boot 框架
② MyBatis Plus
③ MySQL 8.0
④ JSP
⑤ JQuery

（7）性能测试：

① Apache JMeter
② LoadRunner

（8）云服务：

AWS (Amazon Web Services)

4.4.2 技术组件

① 用户管理模块： 提供用户注册、登录、密码管理、角色和权限管理等功能。
② 产品管理模块： 提供冰酒产品的增删查改、分类、排名等功能。
③ 库存管理模块： 提供库存查看和管理、预警设置等功能
④ 订单管理模块： 提供订单创建、查询、支付、退款、物流追踪等功能。
⑤ 营销及推广模块： 提供促销活动、优惠券管理、商品推荐等功能。
⑥ 报表统计模块： 提供销售数据、用户行为、库存变动等数据报表。
⑦ 支付接口： 与支付宝、微信支付等支付平台进行接口对接，提供在线支付功能。
⑧ 物流接口： 与各大物流公司进行数据对接，实现实时物流状态查询。

4.5 安全和合规性

确保系统遵守相关的数据保护法规，如 GDPR 等，并采取必要的安全措施来保护用户数据和交易信息。

4.6 限制和假设

明确在设计和实施过程中可能遇到的限制条件和基本假设。

4.7 未来发展

提出可能的扩展功能或未来发展方向，比如增加人工智能驱动的销售预测，或集成电子商务功能等。

5. 定义及缩略语

5.1 冰酒 (Ice Wine)

在葡萄完全成熟并在自然环境下结冰时收获和压榨的一种甜葡萄酒。

5.2 销售管理系统 (Sales Management System, SMS)

一个用于管理产品销售过程的软件系统，包括订单处理、客户管理、库存控制等功能。

5.3 客户关系管理 (Customer Relationship Management, CRM)

用于管理企业与现有和潜在客户之间互动的技术和策略。

5.4 库存管理系统 (Inventory Management System, IMS)

用于监控库存水平、订单、销售和配送的系统。

5.5 用户界面 (User Interface, UI)

系统与用户之间交互的部分，包括图形界面、命令行界面等。

5.6 应用程序接口 (Application Programming Interface, API)

允许不同软件应用程序相互通信的一套规则和工具。

5.7 数据库管理系统 (Database Management System, DBMS)

用于存储、检索、管理和处理数据的软件。

5.8 GDPR (General Data Protection Regulation)

欧盟的数据保护和隐私法规。

5.9 AI (Artificial Intelligence)

模拟人类智能过程的机器和软件系统。

5.10 电子商务 (E-commerce)

通过电子网络进行的商业交易。

6. 需求分析报告

6.1 总述

6.1.1 产品视角

冰酒订购系统旨在简化冰酒供应商接受和履行冰酒订单的流程。它旨在取代将订单记录在卡片上并存储在数据库中的手动方法。有了这个系统，顾客可以通过电话或直接访问商店下单。收到订单后，冰酒供应商输入客户的详细信息、冰酒要求，并使用系统计算成本。这确保了准确的定价和高效的订单管理。该系统中描述的功能旨在在初始版本 1.0 中完全实现。然而，该系统被设计为适应未来的增强功能和后续版本中可能需要的附加功能。这种灵活性确保了冰酒订购系统能够适应不断变化的业务需求和客户要求。

6.1.2 产品特点

冰酒订购系统提供了一系列功能来简化冰酒的订购流程，来确保订单准确无误，配送高效。顾客可以通过电话或亲自前来订购冰酒，提供他们的个人信息和冰酒要求。系统会根据冰酒选择和配送距离自动计算订单的价格，将订单放置在数据库中，以便需要时能够轻松找到特定的订单。系统可以跟踪订单的状态，确保及时配送并满足顾客的需求。最后，系统具备适应未来扩展的能力，例如在线下单和自动通知等功能。

6.1.3 用户类别和特点

冰酒供应商是果园农民，也是冰酒供应商。他向市场供应多种类的冰酒。客户可以通过电话或亲自到店订购冰酒。客户可以是商店、超市或个人。冰酒供应商会填写一份订单，记录客户的详细信息、冰酒要求和计算费用。应当允许所有冰酒供应商的员工都可以通过系统为客户填写订单，并且很有可能会在某些时间段内出现大量订单的数据处理。同样的，应当允许客户查看自己的订单运送情况。

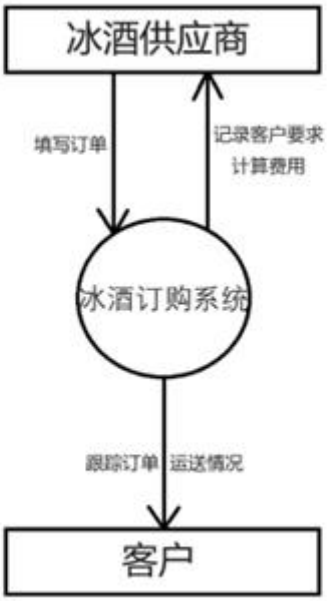


图 6.1.3.1 ：冰酒订购系统用户图

6.1.4 操作环境

OE-1：冰酒订购系统应能在笔记本电脑或者台式主机上允许。系统要求为 Windows 11 平台上运行。

OE-2：冰酒订购系统应在 Windows 11 和指定存储订单信息的 Linux 服务器之间进行接口。此时尚未建立 Web 服务器和数据库软件。本文档的更新版本将包括与这些特定操作环境相关的信息。

6.1.5 设计与实现限制

CO-1：分配给这个项目的时间将限于 2023 年秋季学期。

CO-2：项目语言为 C++，开发环境为 Microsoft Visual Studio。

CO-3：用户手册的所有 HTML 代码将符合 HTML 4.0 标准。

6.1.6 用户文档

UD-1：系统将提供 HTML 格式的在线用户手册，其中描述了用户可用的功能和选项。

UD-2：系统将提供用户手册的硬拷贝，该硬拷贝与 UD-1 中概述的手册相同

6.1.7 假设和依赖性

AS-1：在此软件需求规格说明文档中提到的数据库已经事先管理好，其中包含了冰酒订购系统所需的正确信息。

AS-2：对于客户的订单，在发出运送之前能够通过系统进行订单要求的修改。

DE-1：用于给客户展示商品的数据数据依赖于一个由冰酒订购系统无法管理的外部数据库中的信息。

DE-2：用户订单信息依赖于员工对每个订单都始终使用系统进行记录。

6.2 外部接口需求

6.2.1 用户界面

UI-1: 系统应在产品信息页面上显示产品图片，以帮助用户更容易识别产品。

UI-2: 用户可以使用鼠标单击产品图片，以查看更多产品详细信息、库存状况和价格。

UI-3: 所有对库存的修改操作应通过拖拽或点击图形化界面进行，键盘仅用于输入密码进行身份验证。

UI-4: 系统将提供一个帮助按钮，点击后将打开在线用户手册，以支持用户在使用系统时解决问题。

UI-5: 系统应提供一个“销售统计”页面，显示当前的销售趋势、库存水平和利润统计数据。

6.2.2 硬件接口

HI-1: 系统将在 POS（Point of Sale）终端设备上运行，这些设备包括商店收银机、扫描枪、打印机、条形码阅读器等。

HI-2: POS 终端设备将通过 USB 接口连接到计算机系统，以实现销售、库存管理和打印收据等功能。

HI-3: 针对移动销售和库存管理，系统将支持移动设备，如平板电脑或智能手机，这些设备可以通过 Wi-Fi 或蓝牙接口连接到系统。

HI-4: 为了实现支付功能，系统需要与信用卡处理终端进行连接，通过 POS 终端设备或互联网接口进行交互。

HI-5: 系统需要与金库设备（例如钱箱）进行连接，以存储现金和硬币。

HI-6: 为了备份和保护数据，系统将连接到网络存储设备或云存储，通过以太网或无线

网络接口进行数据传输。

HI-7：如果需要，系统可能需要支持显示屏、客户显示器、POS 键盘等外部设备，这些设备可以通过 USB、HDMI 或 VGA 接口连接。

6.2.3 软件接口

SI-1：冰酒销售商店管理系统

SI-1.1：该软件将通过网络接口与后端数据库进行通信，以实现以下功能：将产品信息、库存数据和销售记录上传到数据库。从数据库中检索产品信息、库存数据和销售记录。

SI-1.2：系统将允许授权用户随时修改产品信息、库存数据和销售记录。

SI-1.3：如果用户在更新库存或销售记录时遗漏了信息，系统将提供自动数据同步功能，确保数据库与系统保持一致。

SI-2：后端数据库

冰酒销售商店管理系统将与后端数据库进行数据交换，以实现以下功能：

SI-2.1：允许用户向数据库添加新的产品信息、库存数据和销售记录。

SI-2.2：允许用户从数据库中更新或修改产品信息、库存数据和销售记录。

SI-2.3：允许用户查询数据库以检索产品、库存和销售数据，以生成报告和统计信息。

SI-3：电子邮件通知系统

如果有新订单或特定的事件发生，系统将与电子邮件通知系统进行集成，以实现以下功能：

SI-3.1：当有新订单时，系统将自动生成电子邮件通知，通知相关部门或员工。

SI-3.2：如果库存低于特定水平，系统将发送电子邮件通知以提醒采购或库存管理。

SI-3.3：当销售记录被修改或更新时，系统将发送电子邮件通知给相关人员，以保持信息同步。

6.2.4 通信接口

CI-1：系统必须确保用户安全性通过密码的一致性。

CI-2：当需要向客户发送订单确认、支付提醒或其他通知时，系统将通知邮件服务模块，以触发电子邮件通知的发送。

CI-3：系统将通过无线以太网系统与数据库进行通信，以实现库存管理、销售记录更新和客户数据的同步。

6.3 系统功能需求

6.3.1 产品图片的显示和浏览

6.3.1.1 说明和优先事项

通过该功能，用户可以浏览和查看产品图片，以帮助他们选择和购买产品。这是高优先级功能，因为产品图片对于用户选择和购买产品非常重要。

6.3.1.2 刺激/反应序列

用户打开产品页面。
系统加载产品信息，包括图片。
用户滚动或点击以浏览不同的产品图片。

6.3.1.3 功能要求

REQ-1: 系统应能够显示产品图片，以便用户可以在浏览产品时查看图片。
REQ-2: 系统应允许用户点击图片以查看更多产品细节，如价格、库存状况和描述。
REQ-3: 用户应能够在产品页面上使用放大和缩小功能来查看更大或更小的产品图片。
REQ-4: 系统应根据产品的可用性状态显示不同的图片标识，如“有库存”或“售罄”。

6.3.2 销售订单管理

6.3.2.1 说明和优先事项

通过该功能，用户可以创建、编辑和管理销售订单，确保订单准确且满足客户需求。这是高优先级功能，因为订单管理是销售业务的核心。

6.3.2.2 刺激/反应序列

用户登录系统。
用户选择“创建新订单”选项。
用户添加所需产品到订单。
用户编辑订单细节，如数量、送货地址等。
用户确认订单并进行付款。
系统生成订单确认页面。

6.3.2.3 功能要求

REQ-5: 用户应能够创建新销售订单，选择所需的产品，并添加到购物车。

REQ-6: 用户应能够编辑订单细节，包括产品数量、送货地址、付款方式等。
REQ-7: 系统应计算订单总额，包括产品费用、税费和运费。
REQ-8: 用户应能够选择不同的付款方式，如信用卡、PayPal 等。
REQ-9: 系统应生成订单确认页面，显示订单细节和订单号。

6.3.3 库存管理

6.3.3.1 说明和优先事项

通过该功能，员工可以管理库存，包括添加、编辑和删除产品信息，以及跟踪库存数量和价格。这是高优先级功能，因为库存管理对于销售商店的运营至关重要。

6.3.3.2 刺激/反应序列

Stimulus: 员工登录系统。
Response: 系统验证员工身份并加载库存管理页面。
Stimulus: 员工选择“添加新产品”选项。
Response: 系统显示产品信息表单，允许员工输入新产品的详细信息。
Stimulus: 员工编辑产品信息，如价格、库存数量和描述。
Response: 系统更新产品信息并显示最新库存状态。

6.3.3.3 功能要求

REQ-7: 员工应能够登录系统，并系统应验证员工身份。
REQ-8: 员工应能够添加新产品信息，包括产品名称、价格、库存数量、描述等。
REQ-9: 员工应能够编辑产品信息，如价格、库存数量和描述。
REQ-10: 系统应根据产品的可用性状态显示不同的图片标识，如“有库存”或“售罄”。
REQ-11: 系统应提供报告和分析页面，显示销售趋势、库存状况和财务数据的图表和图形。

6.4 其它非功能需求

6.4.1 性能要求

6.4.1.1 时间准确，消息处理及时

系统应准确记录销售时间，并确保通知邮件能够及时发送给客户。
实现策略：在系统中使用可靠的时间戳记录销售时间，并使用高效的邮件发送机制，确保消息的及时性。

6.4.1.2 处理大量销售记录和统计数据的高效性能

系统应能够高效地处理大量销售记录和统计数据，以保持系统的稳定性和可用性。

实现策略：使用适当的数据库索引和查询优化技术，合理分配系统资源，如内存和处理器，并进行系统性能测试和优化。

6.4.1.3 处理大量销售记录和统计数据的高效性能

系统应具备良好的并发处理能力，能够同时支持多个销售人员访问和操作系统。

实现策略：使用并发编程技术，如线程池或异步处理，以处理并发请求，确保系统的响应性能和稳定性。

6.4.2 安全要求

6.4.2.1 销售记录和客户信息的完整性

系统应确保销售记录和客户信息不被篡改或损坏。

实现策略：使用数据校验和加密技术，确保数据的完整性和安全性。同时，使用访问控制和权限管理机制，限制对数据的修改和访问。

6.4.2.2 身份验证和访问控制

只有经过身份验证的用户才能访问系统，并根据其角色和权限执行相应的功能。

实现策略：实施安全的用户身份验证机制，如用户名和密码、双因素身份验证等。使用访问控制列表或角色基础的访问控制，确保只有授权用户可以执行相关操作。

6.4.2.3 防止未经授权访问和恶意攻击

系统应具备防止未经授权访问和恶意攻击的安全机制。

实现策略：使用防火墙、入侵检测系统、安全审计等安全工具和技术，及时更新系统补丁和安全策略，以防止未经授权访问和恶意攻击。

6.4.3 数据保护要求

6.4.3.1 符合隐私法规和政策

系统应符合适用的隐私法规和政策，保护用户的个人数据和隐私。

实现策略：严格遵守相关隐私法规的要求，采用数据加密、匿名化处理等技术手段，确保用户数据的合规性和安全性。

6.4.3.2 安全的用户身份验证

系统应采用安全的用户身份验证机制，确保只有合法用户可以访问系统和敏感数据。

实现策略：使用强密码策略、加密传输用户凭证等安全措施，以防止身份伪造和密码泄露。

6.4.3.3 安全的用户身份验证

敏感数据在传输和存储过程中应进行加密，以防止数据泄露和篡改。

实现策略：使用安全的传输协议，如 HTTPS，对数据进行加密传输。在存储时，使用加密算法对数据进行加密保护。

6.4.4 软件质量属性

6.4.4.1 可用性和易用性

系统应具备良好的可用性和易用性，使销售人员能够快速上手并高效地使用系统。

实现策略：提供直观和用户友好的界面设计，简化操作流程，提供清晰的错误提示和帮助文档，以提升系统的可用性和易用性。

6.4.4.2 准确性和可靠性

系统应确保销售记录和统计数据的准确性，避免数据丢失或错误。

实现策略：使用数据验证和合理的错误处理机制，进行数据备份和恢复策略，进行系统稳定性和可靠性的测试和优化。

6.4.4.3 可测试性

系统应具备良好的可测试性，以便进行单元测试、集成测试和系统测试，确保软件质量。

实现策略：设计可测试的模块和接口，使用自动化测试工具和技术，编写清晰的测试用例和测试文档，以提升系统的可测试性和测试效率。

6.4.4.4 扩展性

系统应具备良好的扩展性，能够根据业务需求进行功能的扩展和修改。

实现策略：使用模块化和松耦合的架构设计，遵循设计原则和设计模式，以便于系统的扩展和维护。

6.4.4.5 Bug 反馈入口

系统应提供用户反馈机制，以便用户报告 Bug 和问题，并及时进行处理和修复。

实现策略：在系统中提供 Bug 反馈入口，如问题反馈表单、电子邮件等方式，确保用户可以方便地报告 Bug 和问题，并进行及时的跟踪和解决。

6.4.5 业务规则

1. 定价以及折扣规则：确定产品定价、定义折扣的条件和金额，例如购买数量折扣、促销折扣、会员等级折扣等。
2. 订单处理规则：规定订单的处理流程，包括验证订单信息、库存检查、支付处理、订单状态更新等。
3. 库存管理规则：定义库存的管理策略，包括库存预警、库存补货、库存调拨等。
4. 客户优惠规则：制定与客户相关的优惠政策，例如特定客户的折扣、积分兑换、生日礼品等。
5. 销售报告规则：定义生成销售报告的方式和内容，包括销售额统计、产品销售排名、客户购买历史等。

6.4.6 权限级别

6.4.6.1 管理员/高级经理权限

对所有业务规则具有完全访问和管理权限。
可以创建、编辑和删除定价和折扣规则。
可以定义和配置订单处理规则，包括验证、库存检查、支付和状态更新等。
具有完全访问和管理库存管理规则的权限，包括库存预警、补货和调拨策略的设置。
可以制定客户优惠规则，包括特定客户的折扣、积分兑换和生日礼品等。
可以定义和生成销售报告，包括销售额统计、产品销售排名和客户购买历史等。

6.4.6.2 销售经理权限

可以查看和编辑定价和折扣规则。
可以查看和编辑订单处理规则。
可以查看和编辑库存管理规则，但无法进行删除操作。
可以查看和编辑客户优惠规则。
可以查看销售报告和生成部分销售报告。

6.4.6.3 销售员权限

可以查看定价和折扣规则，但无法进行编辑或修改。
可以提交订单并查看订单状态和更新信息。
可以查看库存信息，但无法进行库存管理操作。
可以应用客户优惠规则，但无法进行编辑或修改。
可以查看销售报告的部分内容。

6.4.6.4 客户权限

可以查看产品定价和折扣信息。

可以下单和查看订单状态。
可以查看库存信息。
可以享受特定客户的优惠政策，如折扣和积分兑换。
可以查看自己的购买历史。

6.5 其它要求

6.5.1 数据库要求

DA-1：系统应包括一个主要数据库，用于存储与冰酒销售相关的数据。

DA-2：主数据库应包含产品信息，包括产品名称、描述、定价、库存量以及与每种产品相关的销售数据。

DA-3：数据库应具备客户信息，包括客户姓名、联系信息、历史购买记录等。

DA-4：系统还需要一个订单数据库，用于跟踪订单信息，包括订单日期、订单号、客户信息、所购买的产品、数量和总金额。

DA-5：数据库应支持员工信息的存储，包括员工姓名、职位、联系信息以及与销售操作相关的数据。

DA-6：该系统应能够生成销售报告，并将其存储在数据库中，以便分析销售趋势和库存管理。

DA-7：数据库应具备安全性措施，以确保敏感销售数据的机密性和完整性。

DA-8：系统还需要备份和恢复功能，以防止数据丢失或损坏。

DA-9：数据库应具备性能和可扩展性，以满足未来可能的数据增长和系统需求变化。

6.5.2 国际化要求

IA-1：系统应支持多语言用户界面，以满足不同地区和语言的用户需求。

IA-2:用户界面应具备本地化能力,使不同语言和文化背景的用户能够方便地使用系统。

IA-3：系统应支持日期和时间的多时区表示，以确保时间相关的功能在全球范围内正确运行。

IA-4：系统应提供语言切换功能，允许用户根据自己的首选语言选择界面语言。

IA-5：系统应支持国际化字符集，以处理不同语言所涉及的字符和符号。

IA-6：用户界面应具备可调整的日期和时间格式，以适应不同地区的习惯和格式要求。

IA-7：系统应提供多语言文档和帮助文档，以帮助用户理解和使用系统。

IA-8：系统应考虑不同地区的法律、文化和商业规范，以确保系统在全球范围内的合规性和适应性。

IA-9：系统应提供全球范围内的货币和货币符号支持，以适应不同地区的货币需求。

IA-10：系统应支持跨时区的数据同步和共享，以便在不同地区的用户之间共享和访问数据。

6.6 分析模型

在本附录中，我们提供了一些相关的分析模型，以帮助更好地理解和可视化"冰酒销售管理系统"的需求。以下是一些可能的分析模型，包括数据流图、类图、状态转换图和实体关系图。

6.6.1 数据流图（DFD）

数据流图是一种图形化工具，用于描述系统内数据的流动和处理。它有助于识别数据如何在系统中传递和转换。

本系统的数据流图如下图所示。

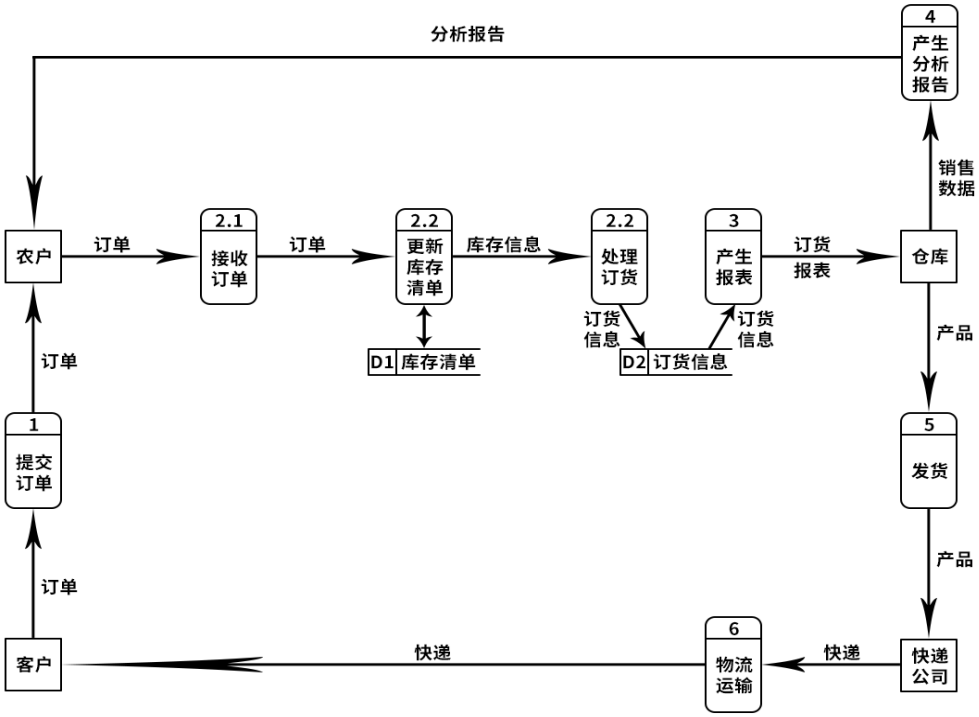


图 6.6.1.1 ：数据流图

6.6.2 类图

类图用于显示系统中的类和它们之间的关系。在"冰酒销售管理系统"中，我们可以绘制类图以描述不同类别的对象，如产品、客户、订单和员工，以及它们之间的关系。

本系统的类图如下图所示。

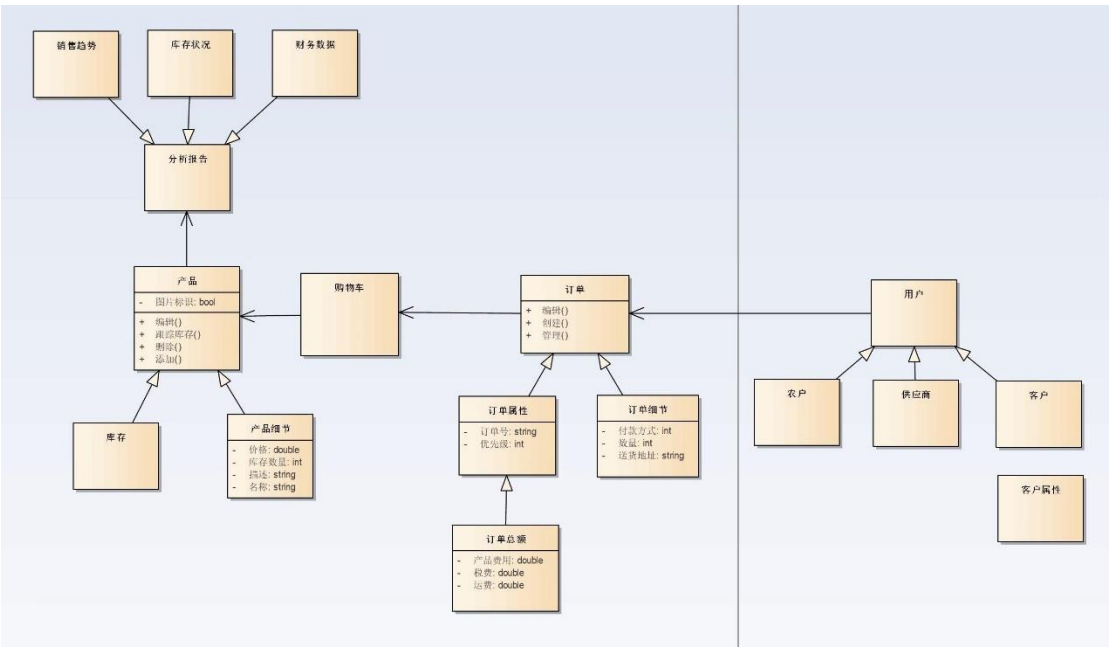


图 6.6.2.1 ：类图

6.6.3 状态转换图

状态转换图用于表示对象在不同状态之间的转换以及导致这些转换的事件。在"冰酒销售管理系统"中，状态转换图可以用于描述订单的生命周期，包括订单的创建、处理、完成和关闭等状态。

本系统的状态转换图如下图所示。

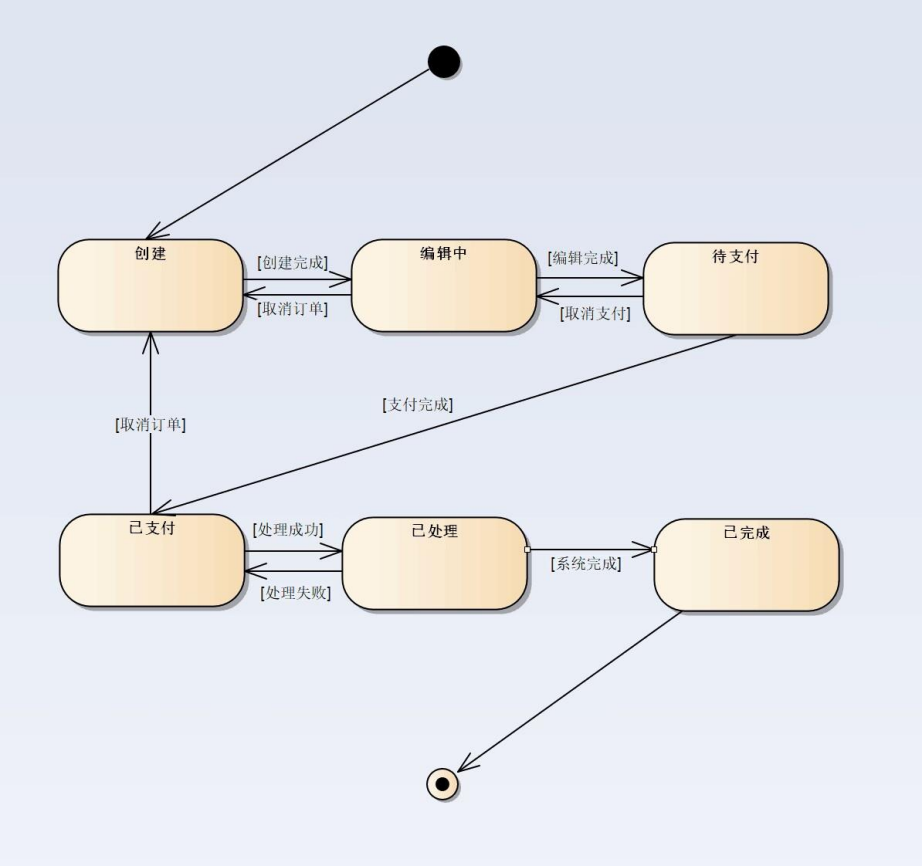


图 6.6.3.1 ：状态转换图

6.6.4 实体关系图（ERD）

实体关系图用于显示系统中各个实体之间的关系。在本系统中，我们可以使用 ERD 来表示不同实体，如产品、客户、订单和员工，以及它们之间的联系，如产品与订单之间的关联。

这些分析模型是帮助理解"冰酒销售管理系统"需求的工具，有助于项目相关人员更清晰地看到系统的结构和功能。请注意，具体的分析模型可以根据项目的需要进行调整和扩展。

本系统的数据流图如下图所示。

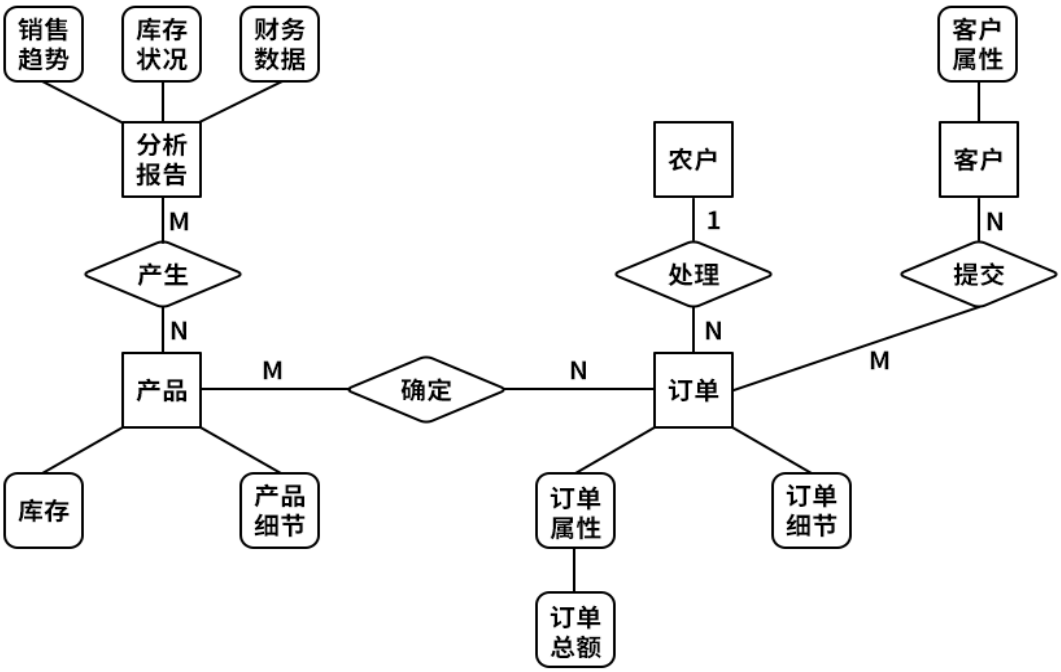


图 6.6.4.1 ：实体关系图

6.7 待定部分

以下是"冰酒销售管理系统"需求分析报告中待确定（TBD）的引用项的编号列表，这些项需要进一步确认和详细定义，以便在后续的项目开发中得到解决：

TBD-1：安全性要求 – 需要详细定义关于数据安全性、用户身份验证和访问控制的具体要求。

TBD-2：报表生成 – 需要确定系统报表的具体内容、格式和生成方式。

TBD-3：用户界面设计 – 需要进一步讨论和规划用户界面的设计元素和布局。

TBD-4：数据备份和恢复 – 需要明确定义有关数据库备份和恢复的策略和流程。

TBD-5：性能要求 – 需要详细规定系统的性能指标和相关测试方法。

TBD-6：安全性措施 – 需要进一步讨论有关密码管理、数据加密和用户权限的安全性措施。

TBD-7：集成要求 – 需要确定与其他系统的集成需求和接口规范。

TBD-8：数据库管理 – 需要明确定义数据库的结构、表格和字段。

TBD-9：通知系统 – 需要进一步规划与客户和员工的通信方式和通知系统。

TBD-10：数据校验 – 需要详细定义数据输入和输出的校验规则和流程。

这些待定项将在项目的后续阶段进行详细讨论和确定，以确保"冰酒销售管理系统"的所有需求都能得到满足。在每个 TBD 项解决之前，项目团队应跟踪和记录它们的状态，以确保项目进展受到适当的控制。

7. 系统原型报告

7.1 登录页面

直接登录是以用户权限来操作，若要切换到后台可以选择后台登录。

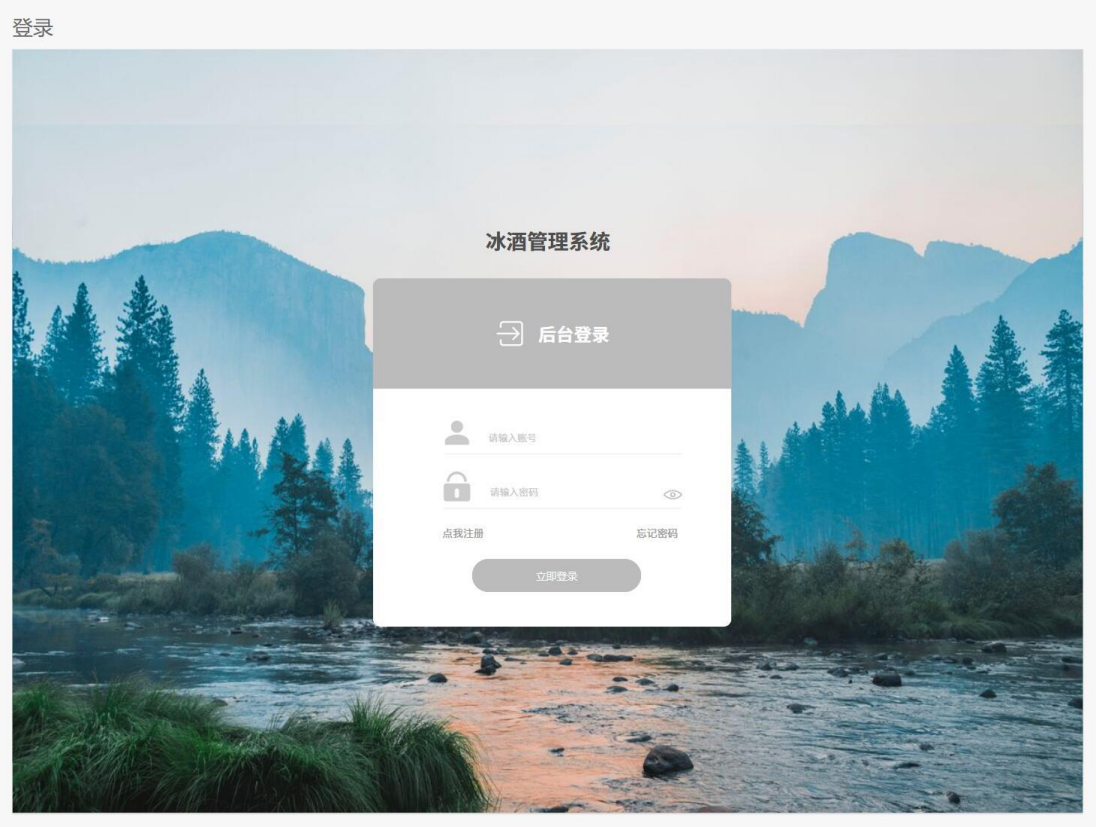


图 7.1.1 ：登录页面

7.2 注册页面

点击注册即跳转到注册界面，输入手机号，密码验证码进行注册。

图 7.2.1 ：注册页面

7.3 推荐页面

这页也是首页，主要展示一些近期热卖的商品，以及根据客户的偏好进行商品推荐。此外还有搜索框和公告栏。

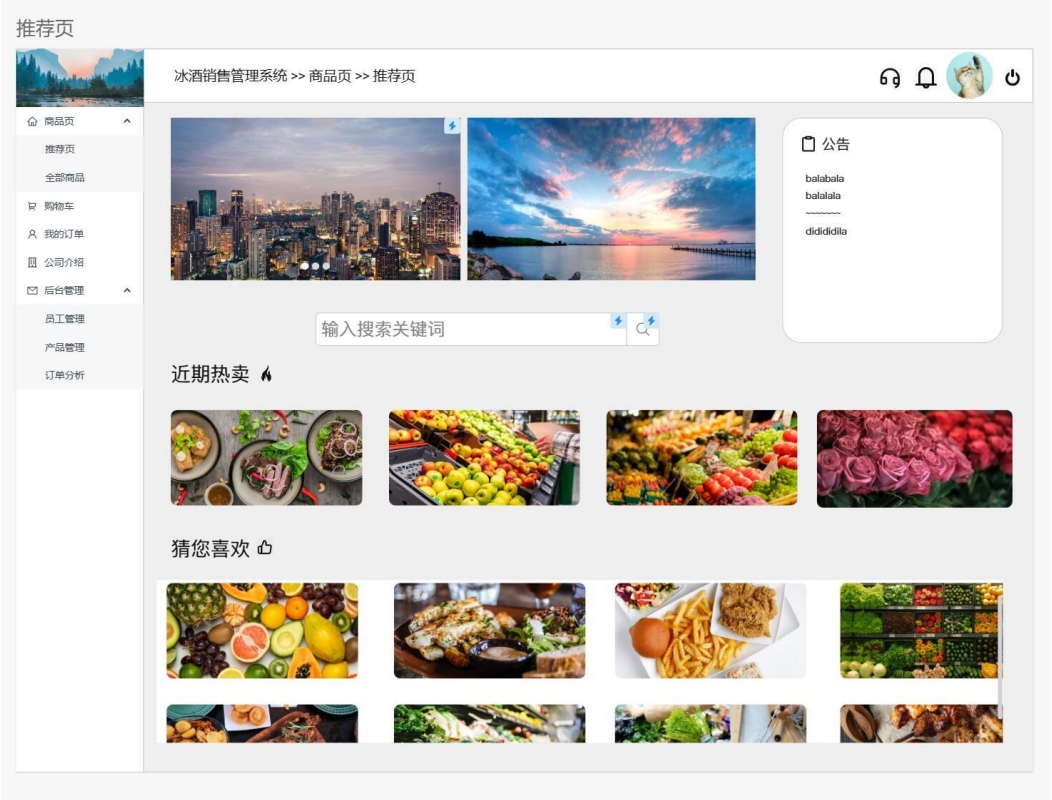


图 7.3.1 ：推荐页面

7.4 全部商品页面

全部商品在此页展示，包含多种筛选搜索功能。

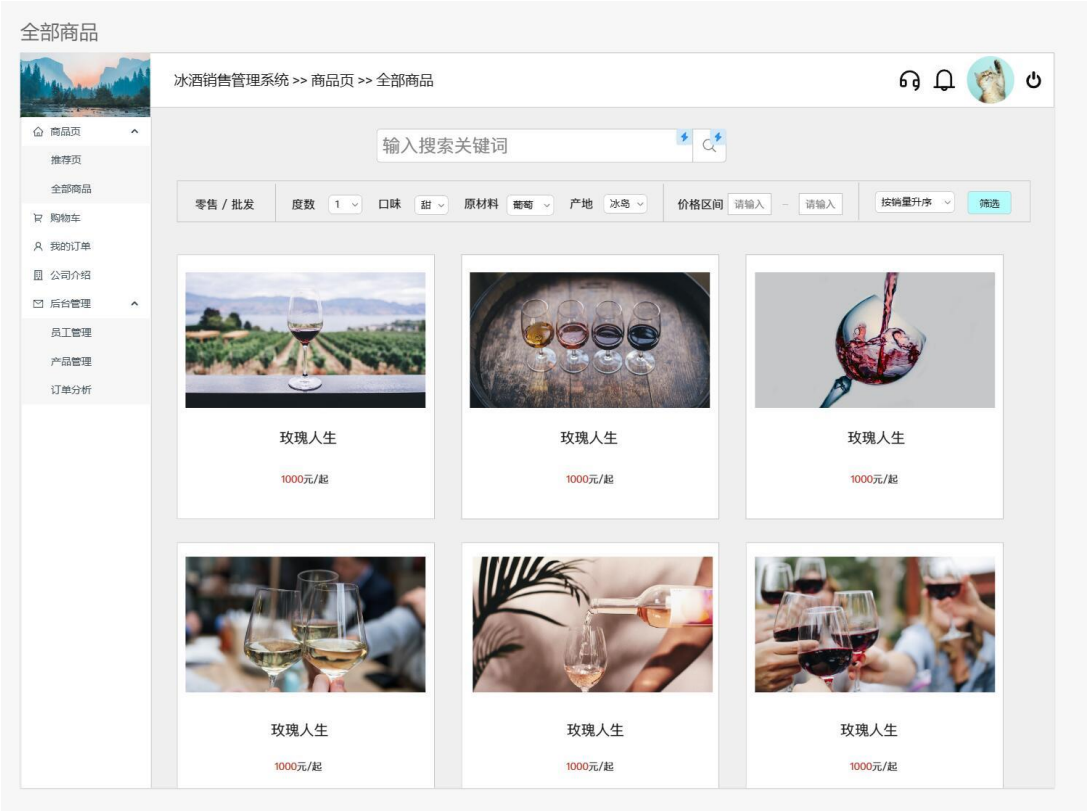


图 7.4.1 ：全部商品页面

7.5 购物车页面

勾选商品以进行编辑，可结算。

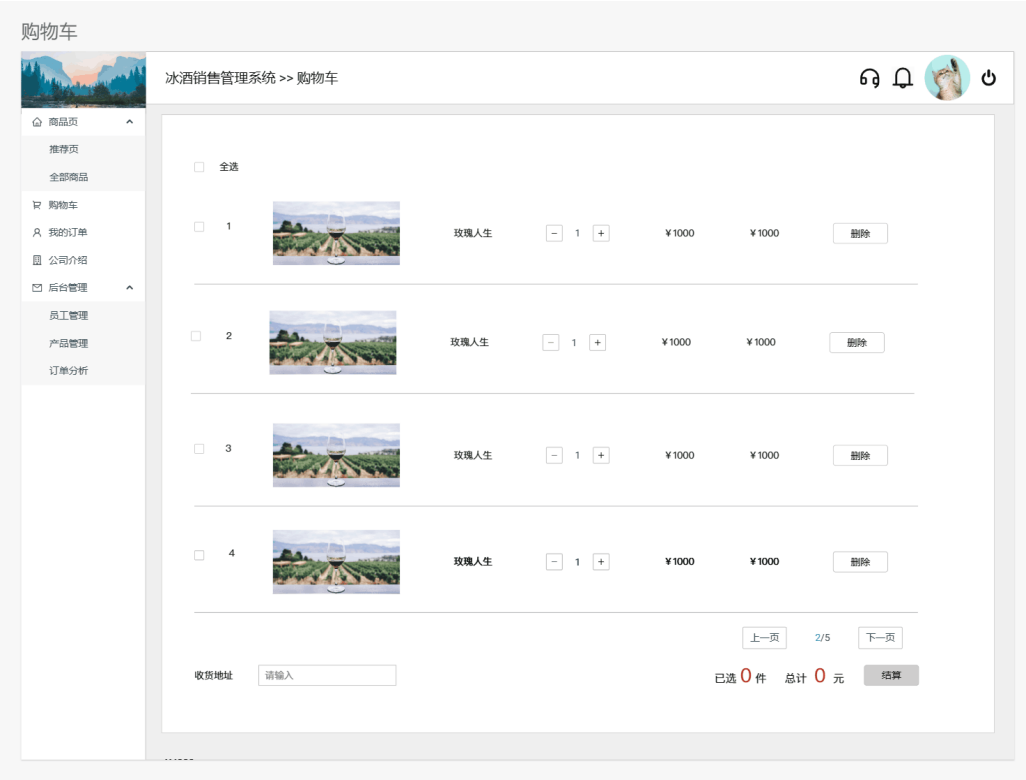


图 7.5.1 ：购物车页面

7.6 支付页面

支付界面，填写收货信息，支付方式，配送方式，是否开发票，以及确认商品信息无误，然后提交订单。

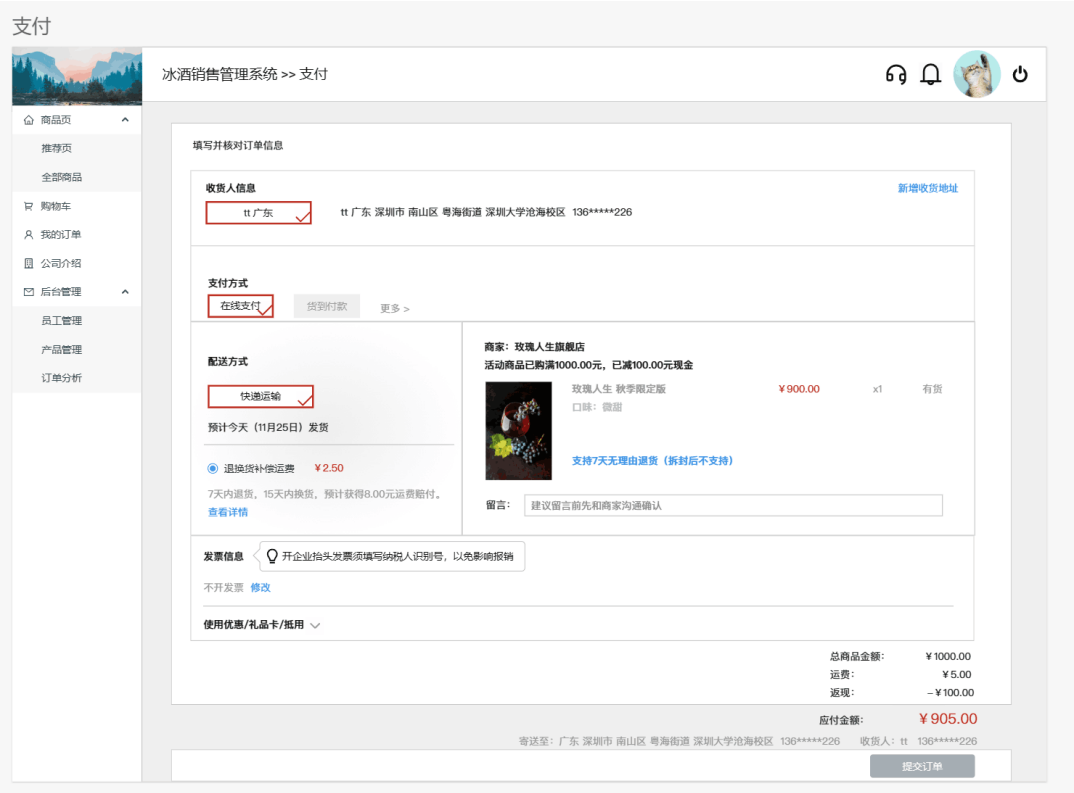


图 7.6.1 ：支付页面

7.7 我的订单页面

展示所有历史订单。

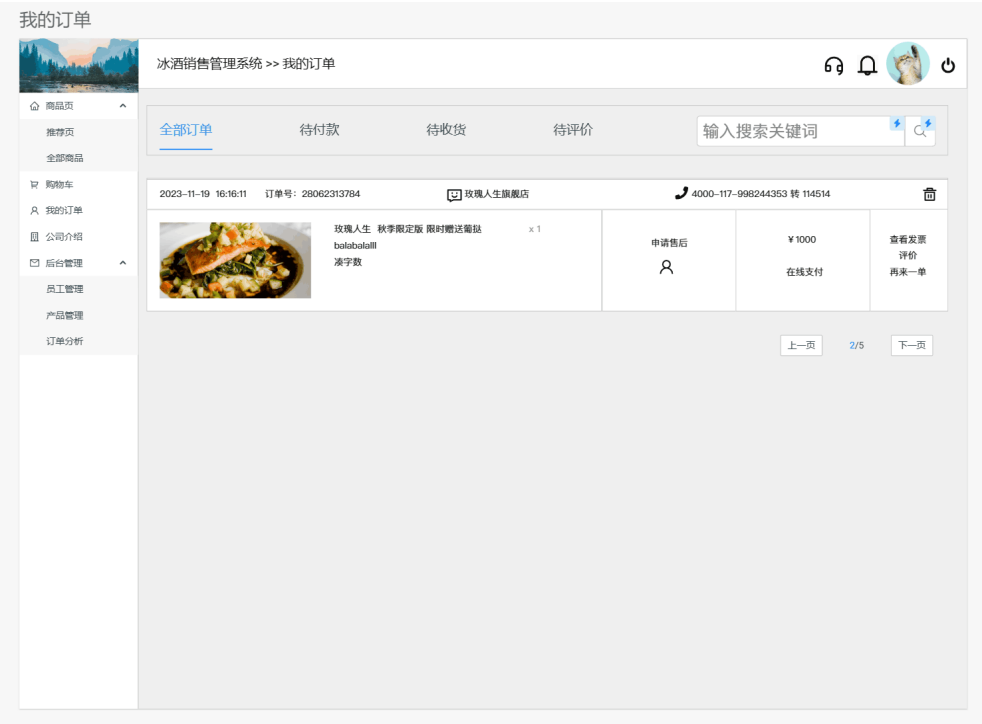


图 7.7.1 ：我的订单页面

7.8 公司介绍页面

可上传有关公司的介绍，吸纳更多人才。

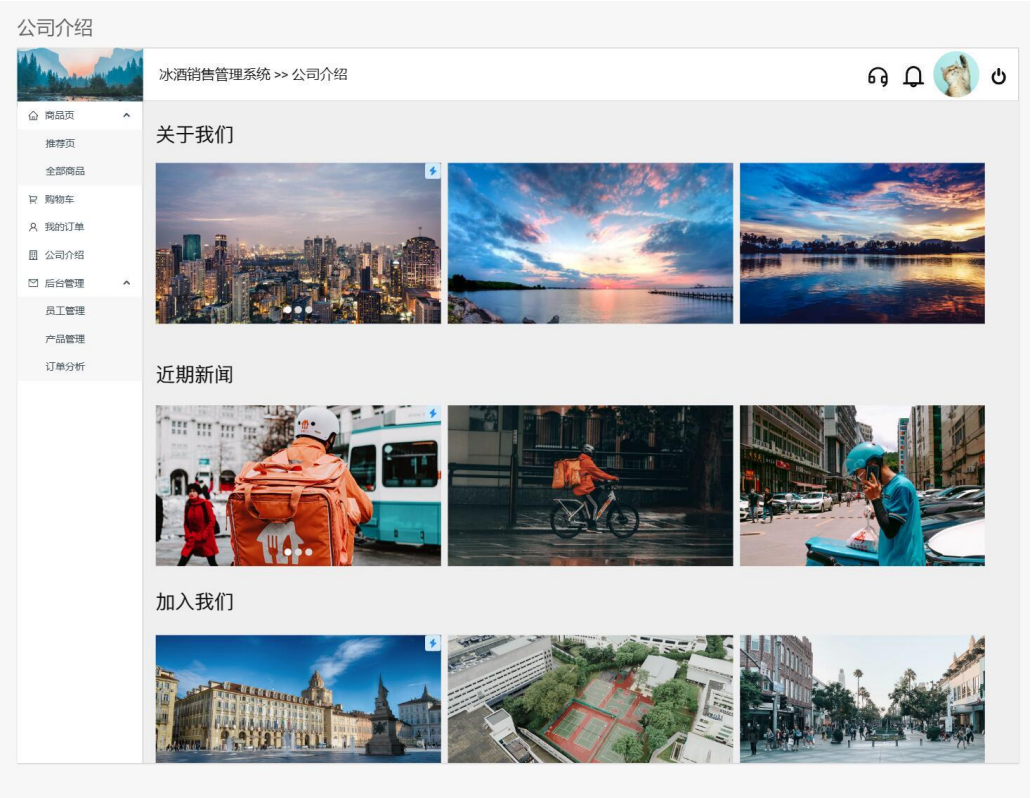


图 7.8.1 ：我的订单页面

7.9 我的页面

个人信息及账号相关资料编辑。

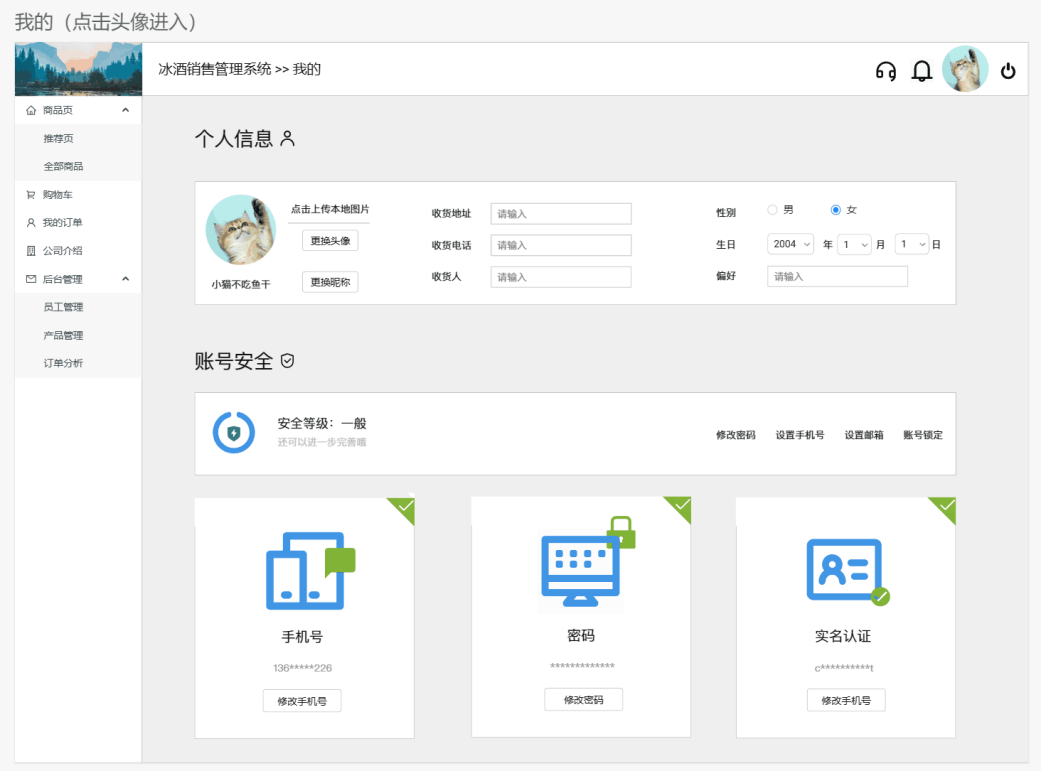


图 7.9.1 ：我的页面

7.10 商品详情页面

包含该商铺的详细信息以及一些相关推荐，下拉查看更多商品详情。

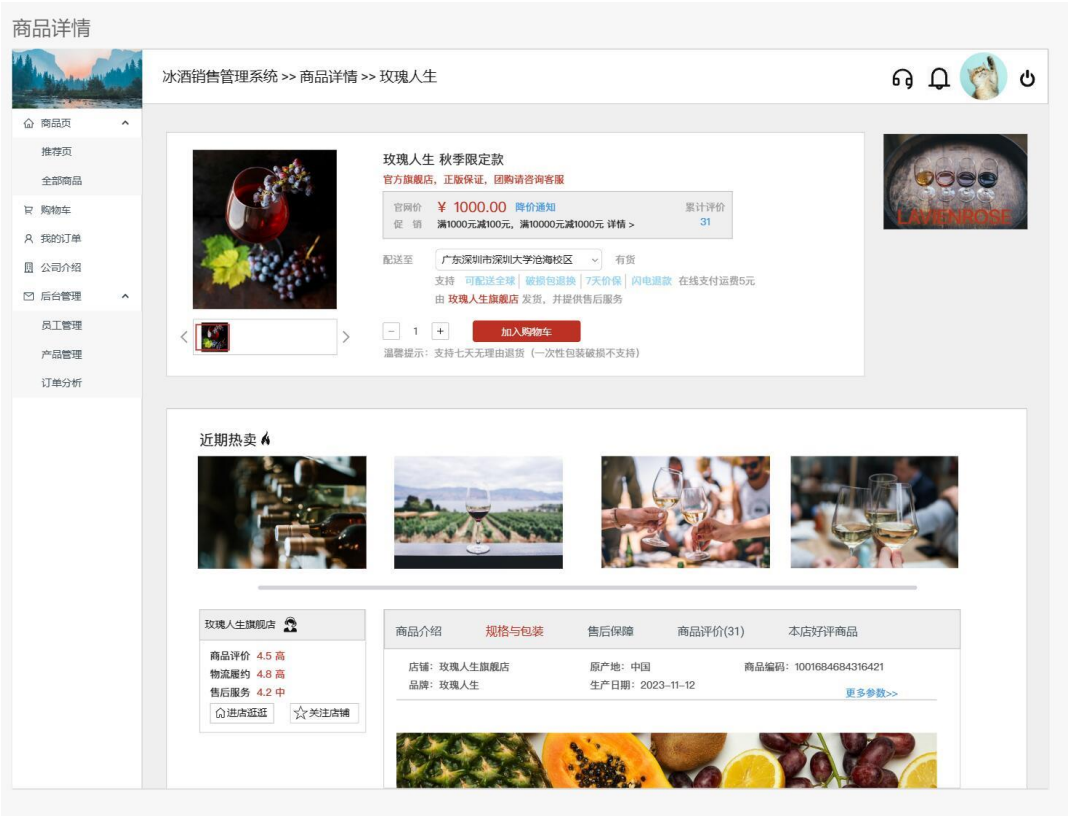


图 7.10.1 ：商品详情页面

7.11 员工管理页面

员工信息管理，登入后台后根据一定权限基于解锁对应功能。

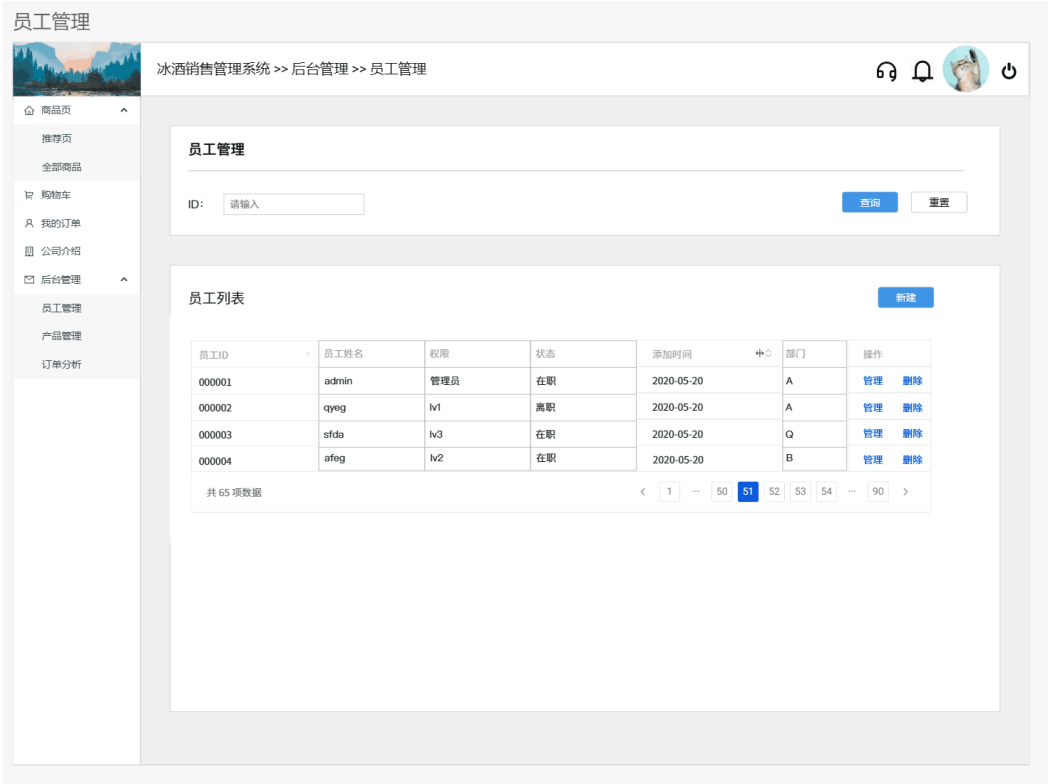


图 7.11.1 ：员工管理页面

7.12 产品管理页面

库存管理，商品上架等相关信息。

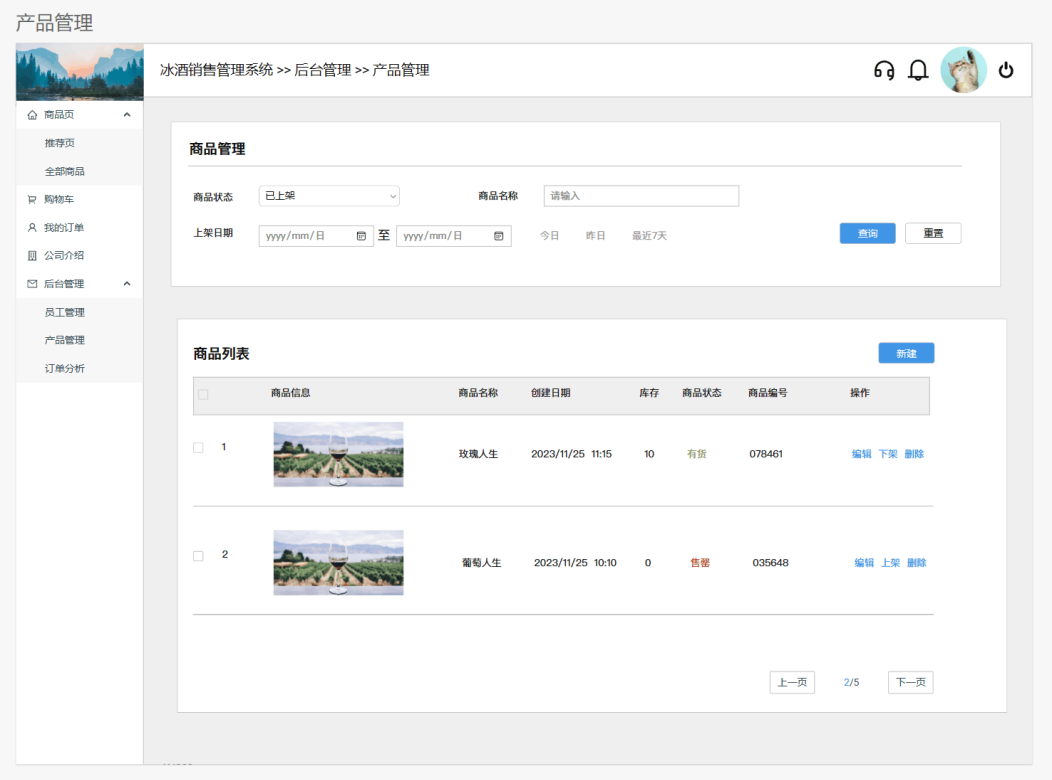


图 7.12.1 ：产品管理页面

7.13 订单分析页面

订单分析，多角度解析销售情况，以便及时抓住客户需求，市场风向。

图 7.13.1 ：订单分析页面

7.14 新增产品页面

上传新产品的详细信息。

图 7.14.1 ：新增产品页面

7.15 新增员工页面

上传新员工的详细信息。

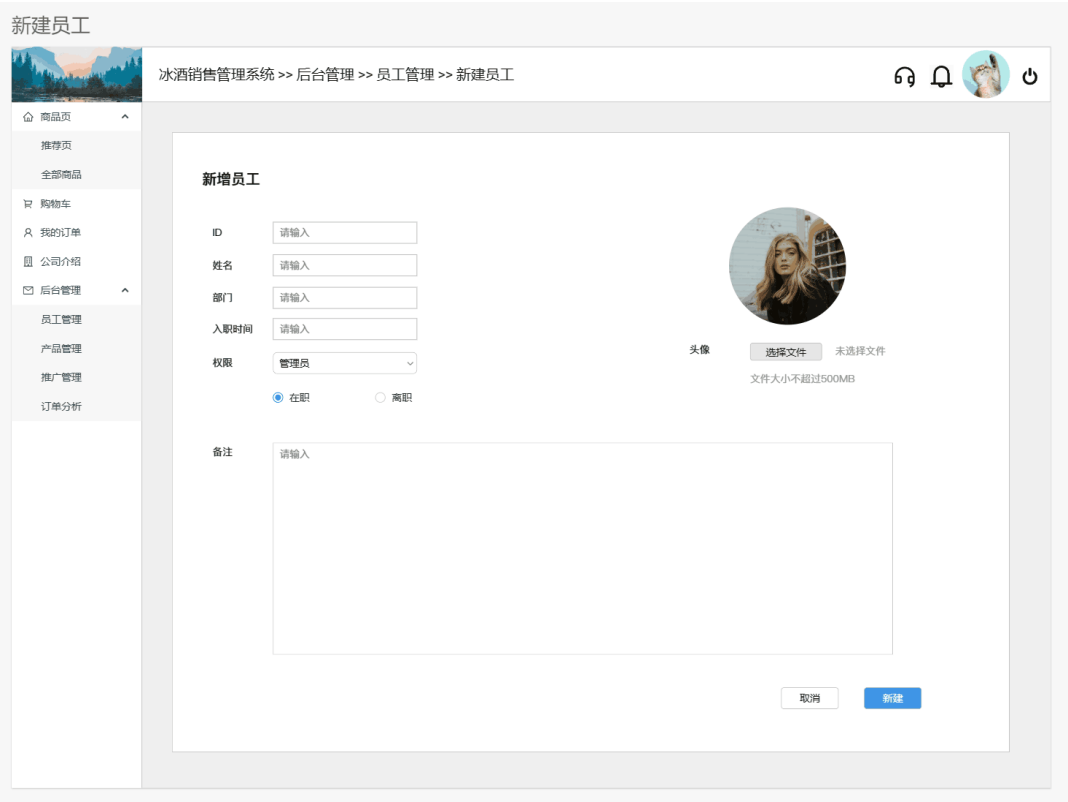


图 7.15.1 ：新增员工页面

7.16 推广管理页面

产品的推广管理，主要应用于主页推荐页，商品详情页等。

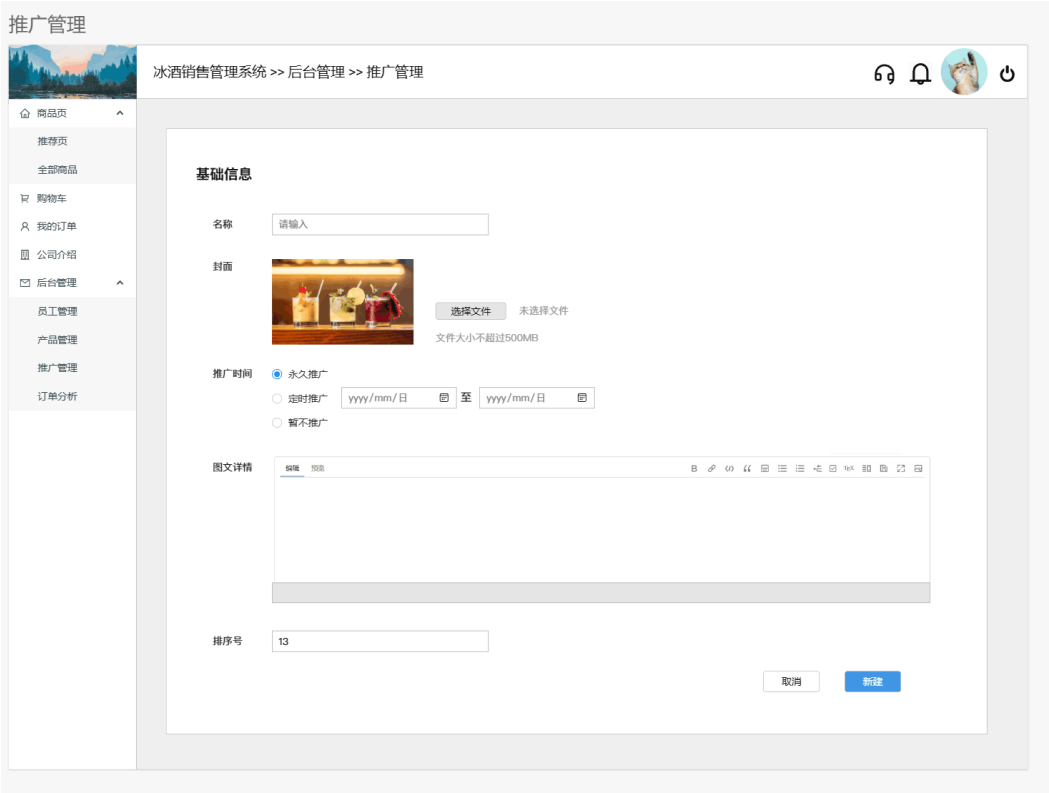


图 7.16.1 ：推广管理页面

7.17 总览

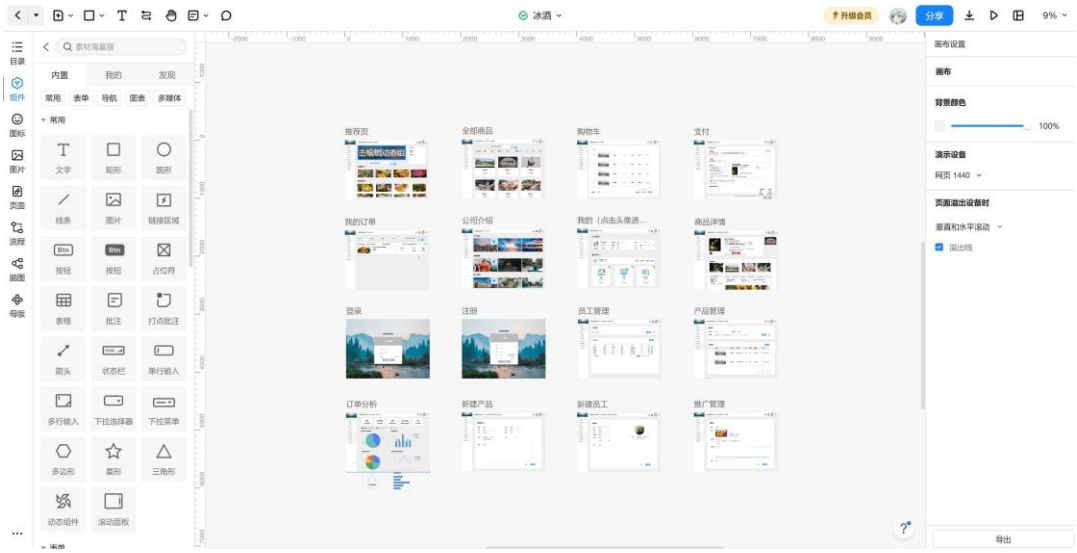


图 7.17.1 ：总览

8. 系统设计与实现报告

8.1 用户管理 UML 图

8.1.1 用例图

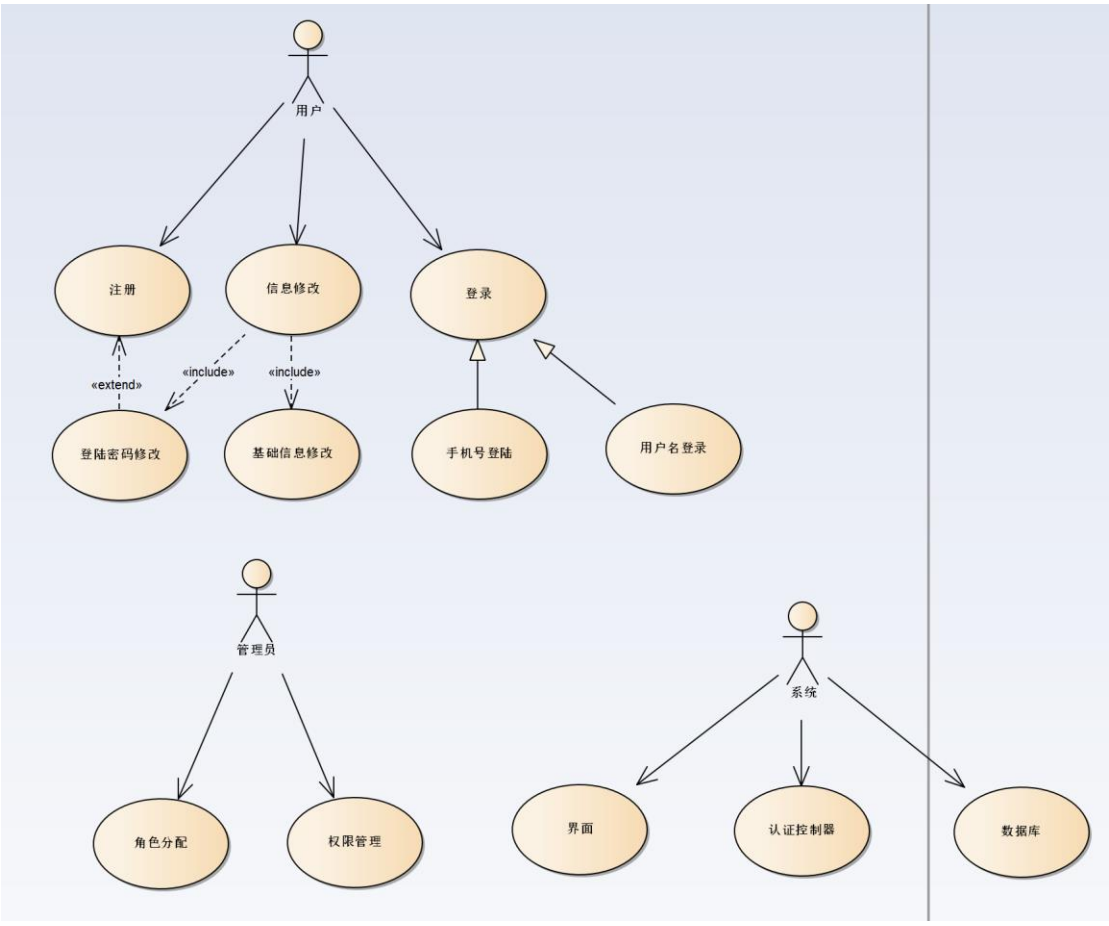


图 8.1.1 ：用户管理 – 用例图

参与者（Actors）:
用户（User）：直接与系统交互的人，使用系统的基本功能。
管理员（Admin）：负责管理和维护系统的人，使用高级功能。
用例（Use Cases）:
注册（Register）：用户创建新账户的过程。
登录（Login）：用户进入系统的过程。
密码管理（Manage Password）：用户更改或重置密码的功能。
角色分配（Assign Roles）：管理员为用户分配角色的功能。
权限管理（Manage Permissions）：管理员管理角色权限的功能。

关系（Relationships）：
用户与用例的关系：用户与注册、登录和密码管理用例相关联。
管理员与用例的关系：管理员与角色分配和权限管理用例相关联。

8.1.2 类图

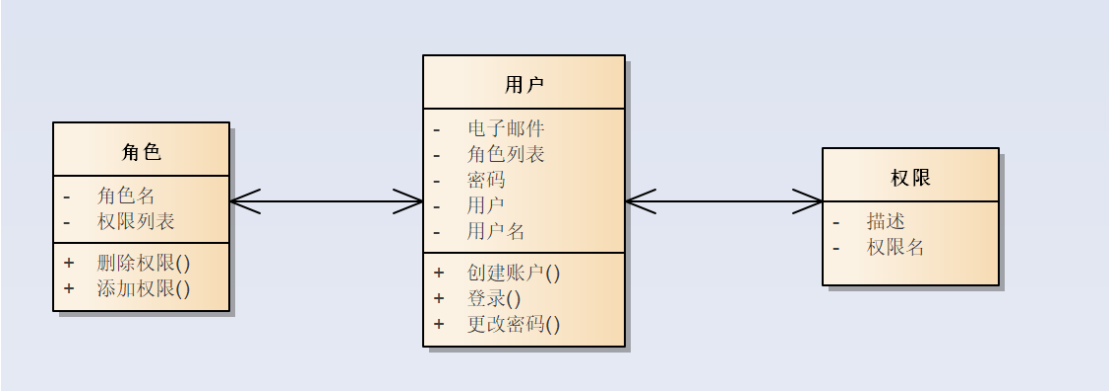


图 8.1.2 ：用户管理 – 类图

类（Classes）：
用户（User）：
属性：用户名（username）、密码（password）、电子邮件（email）、用户 ID（userId）、角色列表（roles）。
方法：创建账户（createAccount）、登录（login）、更改密码（changePassword）。
角色（Role）：
属性：角色名（roleName）、权限列表（permissions）。
方法：添加权限（addPermission）、删除权限（removePermission）。
权限（Permission）：
属性：权限名（permissionName）、描述（description）。
方法：无特定方法，主要用于访问控制。
关系（Relationships）：
用户与角色：多对多关系（一个用户可以有多个角色，一个角色可以分配给多个用户）。
角色与权限：多对多关系（一个角色可以拥有多个权限，一个权限可以分配给多个角色）。

8.1.3 顺序图：用户登录

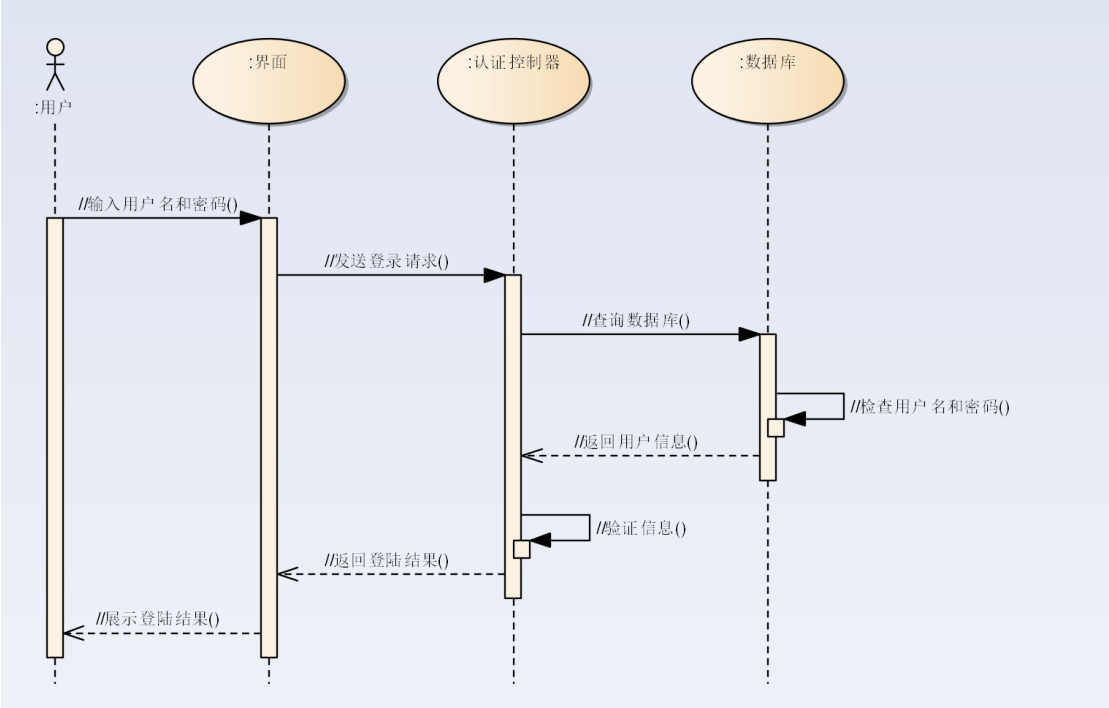


图 8.1.3 ：用户管理 – 顺序图

参与者和对象：
用户（User）：与系统交互的人。
界面（UI）：用户界面，用户与系统交互的入口。
认证控制器（Authentication Controller）：处理登录请求的后端控制器。
数据库（Database）：存储用户数据的数据库。
交互过程：
用户在界面上输入用户名和密码。
界面将登录请求发送给认证控制器。
认证控制器查询数据库，检查用户名和密码。
数据库返回用户信息给认证控制器。
认证控制器验证信息，然后向界面返回登录结果（成功或失败）。

8.1.4 活动图：用户登录

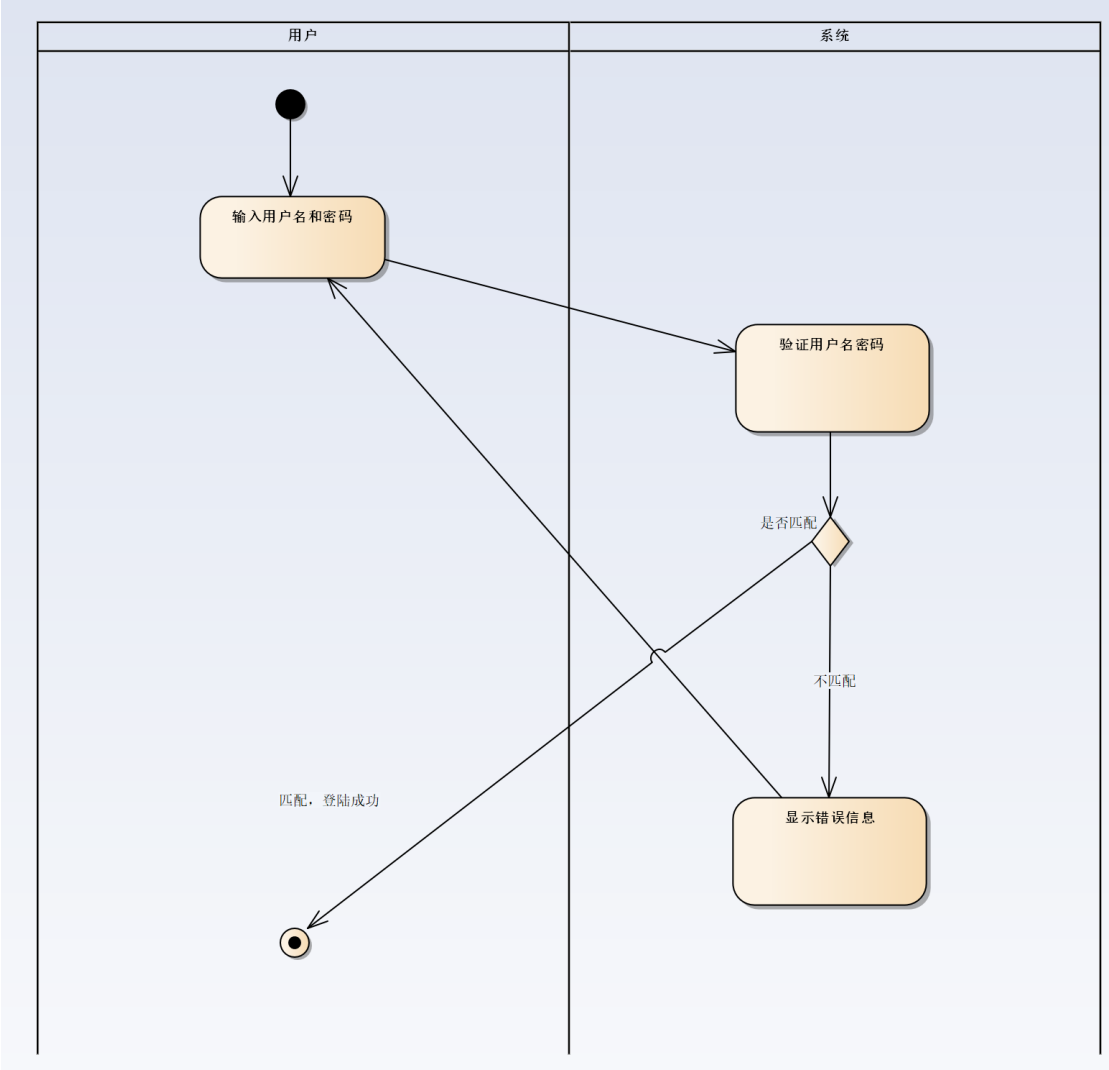


图 8.1.4 ：用户管理 – 活动图

活动和决策点：
开始：表示流程的开始点。
用户输入用户名和密码：用户在界面上输入登录凭据。
系统验证凭据：系统检查输入的用户名和密码是否匹配。
决策点 - 凭据是否有效？
如果是（有效），则进行下一步。
如果否（无效），则转到“显示错误消息”。
显示错误消息：如果登录信息无效，显示错误消息并返回到“用户输入用户名和密码”。
登录成功：用户成功登录系统。
结束：表示流程的结束点。

8.2 订单管理 UML 图

8.2.1 用例图

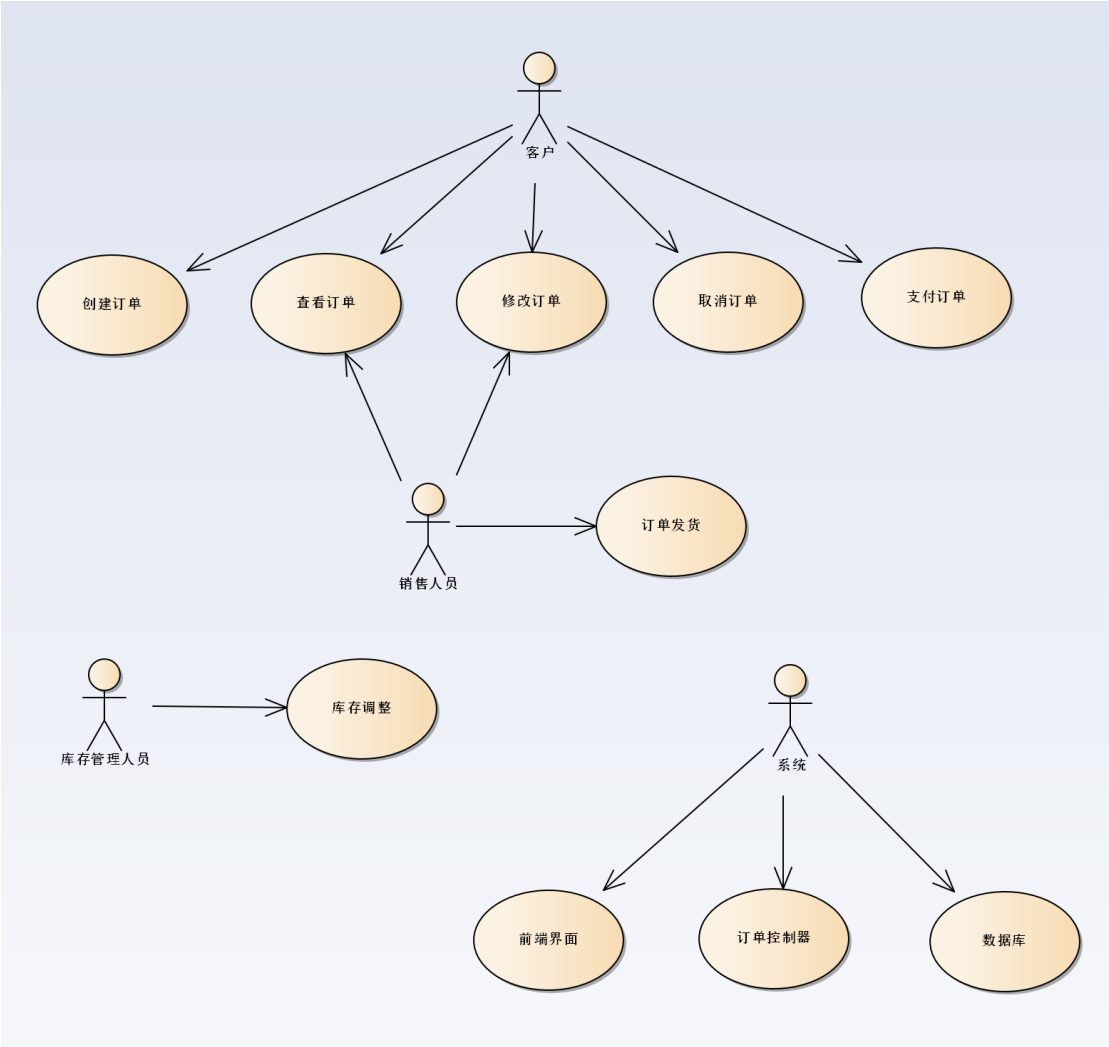


图 8.2.1 ：订单管理 – 用例图

参与者（Actors）:
客户（Customer）：在系统中购物并创建订单的人。
销售人员（Sales Staff）：负责订单的处理和管理。
库存管理人员（Inventory Staff）：处理与订单相关的库存问题。
用例（Use Cases）:
创建订单（Create Order）：客户选择产品并创建新订单。
查看订单（View Order）：客户和销售人员查看订单详情。
修改订单（Modify Order）：客户或销售人员更改订单内容。
取消订单（Cancel Order）：客户取消已创建的订单。
支付订单（Pay for Order）：客户对订单进行支付。
订单发货（Ship Order）：销售人员处理订单发货。

库存调整（Adjust Inventory）：库存管理人员根据订单调整库存。
关系（Relationships）：
客户与用例的关系：客户主要与创建订单、查看订单、修改订单、取消订单和支付订单相关联。
销售人员与用例的关系：销售人员参与查看订单、修改订单、订单发货等。
库存管理人员与用例的关系：库存管理人员涉及库存调整。

8.2.2 类图

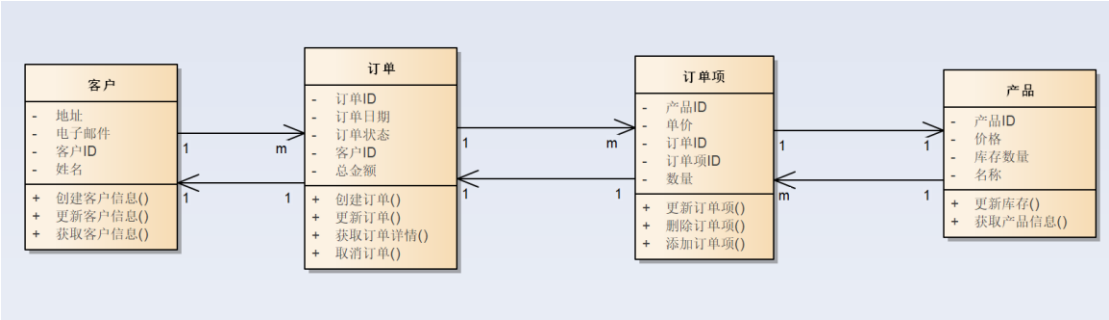


图 8.2.2 ：订单管理 – 类图

类（Classes）：
订单（Order）：

属性: 订单 ID（orderId）、订单日期（orderDate）、订单状态（status）、客户 ID（customerId）、总金额（totalAmount）。
方法: 创建订单（create）、更新订单（update）、取消订单（cancel）、获取订单详情（getDetails）。
客户（Customer）：

属性: 客户 ID（customerId）、姓名（name）、地址（address）、电子邮件（email）。
方法: 创建客户信息（create）、更新客户信息（update）、获取客户信息（getDetails）。
订单项（OrderItem）：

属性: 订单项 ID（orderItemId）、订单 ID（orderId）、产品 ID（productId）、数量（quantity）、单价（unitPrice）。
方法: 添加订单项（addItem）、更新订单项（updateItem）、删除订单项（removeItem）。
产品（Product）：

属性: 产品 ID（productId）、名称（name）、价格（price）、库存数量（inventoryQuantity）。
方法: 更新库存（updateInventory）、获取产品信息（getDetails）。
关系（Relationships）：
订单与客户：一个客户可以拥有多个订单，但每个订单只属于一个客户（一对多关系）。
订单与订单项：一个订单可以包含多个订单项，每个订单项只属于一个订单（一对多关系）。

订单项与产品：每个订单项对应一个产品，产品可以在多个订单项中出现（多对一关系）。

8.2.3 顺序图：订单创建与处理

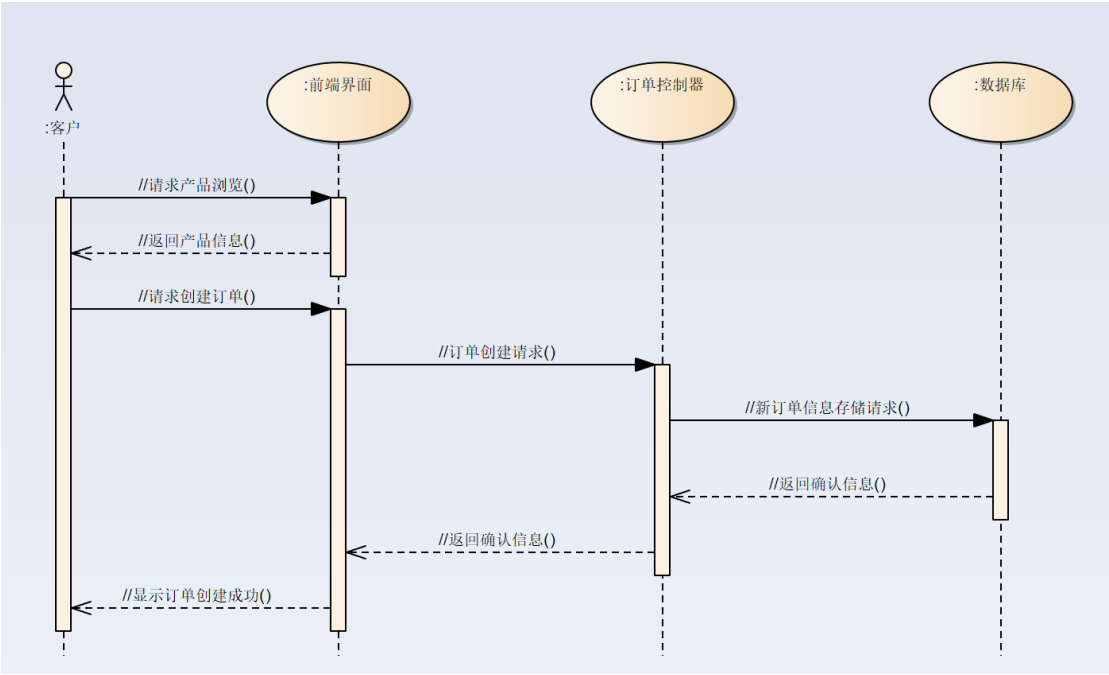


图 8.2.3 ：订单管理 – 顺序图

参与者和对象：
客户（Customer）：发起订单创建的用户。
前端界面（Frontend Interface）：客户与系统交互的界面。
订单控制器（Order Controller）：后端服务，处理订单相关的逻辑。
数据库（Database）：存储订单和客户数据。
交互过程：
客户在前端界面上选择产品并发起创建订单的请求。
前端界面将创建订单的请求发送到订单控制器。
订单控制器请求数据库以存储新订单信息。
数据库确认订单信息已存储并返回确认信息给订单控制器。
订单控制器将订单确认信息发送回前端界面。
前端界面向客户显示订单创建成功的消息。

8.2.4 活动图：订单创建与处理

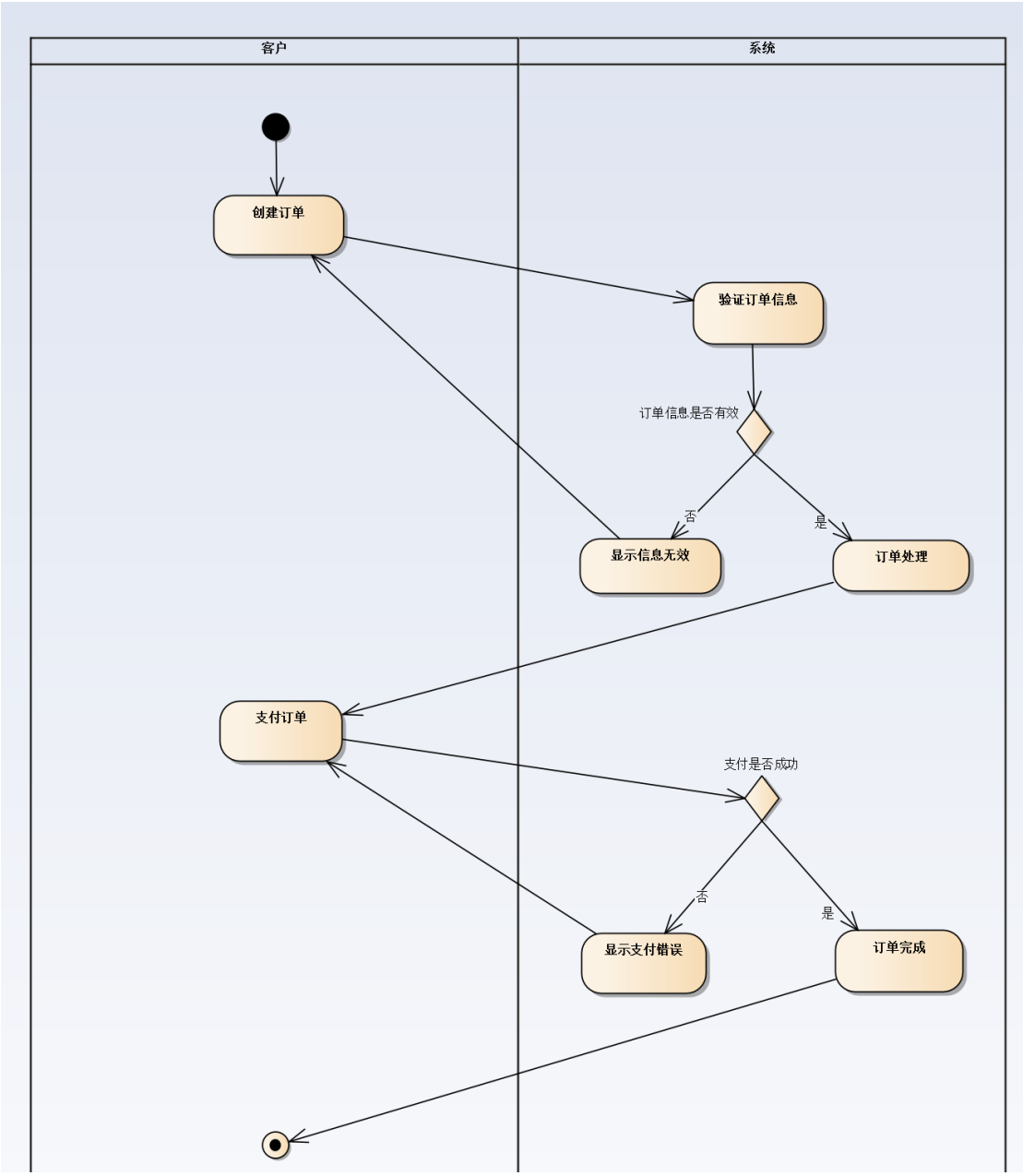


图 8.2.4 ：用户管理 – 活动图

主要活动和决策点：
开始：表示流程的开始点。
客户创建订单：客户在系统中选择产品并发起订单。
系统验证订单信息：系统检查订单信息是否完整和正确。
决策点 - 订单信息是否有效？
如果是（有效），则进入下一步。
如果否（无效），则转到“显示错误消息”。
显示错误消息：如果订单信息无效，向客户显示错误消息，并返回到“客户创建订单”。

系统处理订单：系统对订单进行处理，如计算总价、更新库存等。
订单支付：客户对订单进行支付。
决策点 - 支付是否成功？
如果是（成功），则进入下一步。
如果否（失败），则转到“显示支付错误”。
显示支付错误：如果支付失败，向客户显示错误消息，并返回到“订单支付”。
订单完成：订单处理完成，系统更新订单状态。
结束：表示流程的结束点。

8.3 产品管理 UML 图

8.3.1 用例图

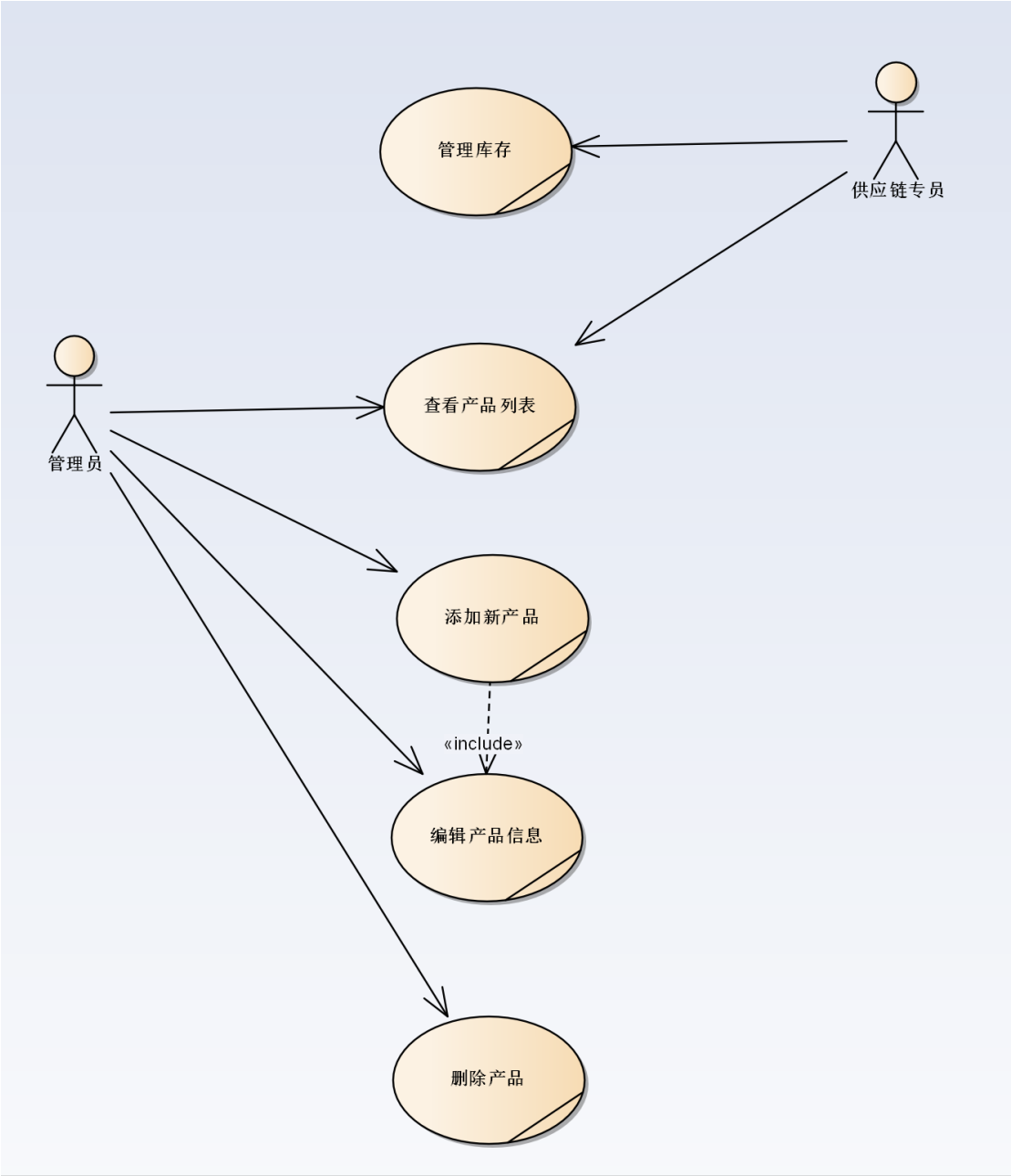


图 8.3.1 ：产品管理 – 用例图

参与者（Actors）：
管理员：负责管理产品信息。
供应链专员：需要访问库存数据来管理供应链。

用例（Use Cases）：

添加新产品：管理员可以添加新产品到系统。
编辑产品信息：管理员可以更新产品的价格、描述等信息。
查看产品列表：管理员和供应链专员可以查看所有产品的列表。
管理库存：供应链专员可以查看和更新库存信息。
删除产品：管理员可以从系统中删除产品。

关系（Relationships）：
关联（Associations）：表示参与者与用例之间的交互。
包含（Include）：如果一个用例是另一个用例的必要部分。
扩展（Extend）：如果一个用例在某些条件下可以扩展或补充另一个用例。

8.3.2 类图

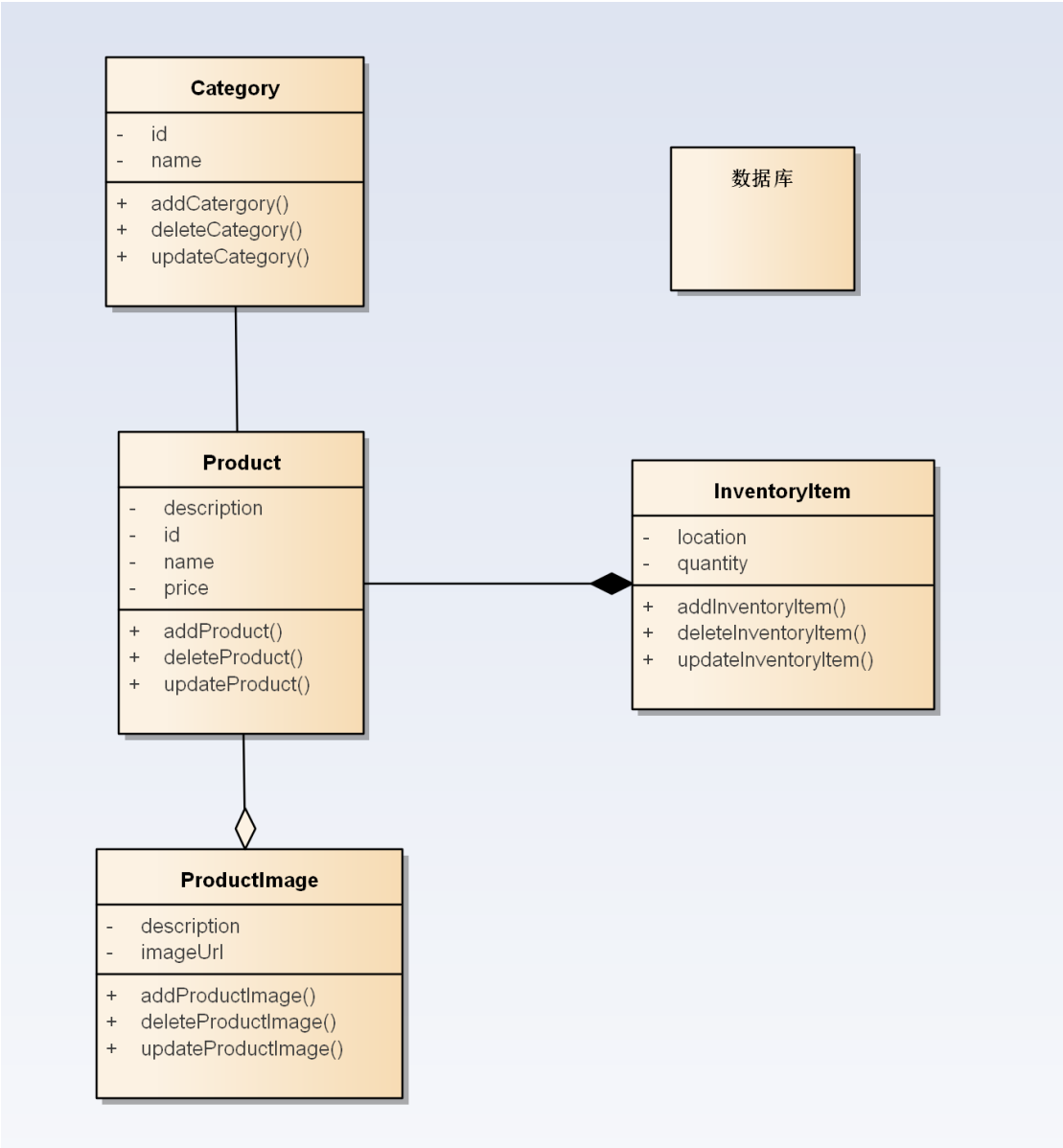


图 8.3.2 ：产品管理 – 类图

主要类:
Product：代表产品，具有属性如 id、name、description、price 等。
Category：代表产品分类，具有属性如 id、name。
InventoryItem：代表库存项，与特定产品相关联，具有属性如 quantity、location。
ProductImage：代表产品图片，具有属性如 imageUrl、description。

关系:
关联（Association）: 例如，Product 与 Category 之间有关联，表示产品属于某个分类。
聚合（Aggregation）: Product 聚合多个 ProductImage，表示产品可以有多张图片。
组合（Composition）: InventoryItem 与 Product 之间是组合关系，表示库存项直接依赖于产品。

方法:
每个类可以有相应的方法，如 addProduct(), updateProduct(), deleteProduct() 在 Product 类中，用于管理产品的生命周期。
示意图描述
Product 类位于中心，包含多个属性，如 id, name, description, price。
Category 类与 Product 类通过关联相连，表示产品分类。
InventoryItem 类与 Product 类通过组合关系相连，表示库存项的依赖。
ProductImage 类与 Product 类通过聚合关系相连，表示产品可以有多张图片。

8.3.3 顺序图

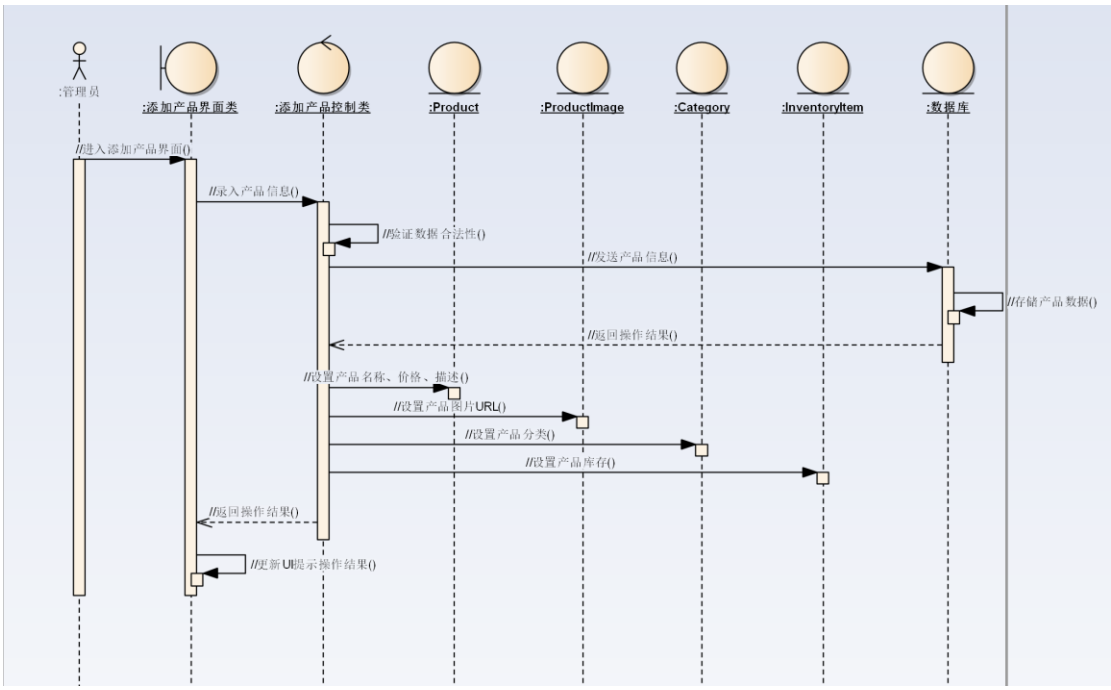


图 8.3.3 ：产品管理 – 顺序图

参与对象：
管理员（User）：与系统交互的用户。
前端界面（UI）：用户界面，用于输入产品数据和显示响应。
控制器（Controller）：处理前端发送的请求。
产品服务（ProductService）：负责产品相关的业务逻辑。
数据库（Database）：存储产品信息。

交互流程：
管理员发起添加产品请求：管理员在 UI 上填写产品信息，如名称、价格、描述，并提交。
UI 发送数据到控制器：UI 捕获输入数据，发送请求到后端的控制器。
控制器请求产品服务处理数据：控制器接收请求，调用产品服务来处理产品数据。
产品服务与数据库交互：产品服务验证数据合法性，然后将数据存储到数据库。
返回操作结果：一旦数据存储完成，产品服务将操作结果返回给控制器，控制器再将结果返回给 UI。
UI 显示响应：UI 根据控制器返回的结果，向管理员显示操作成功或错误信息。

8.3.4 活动图

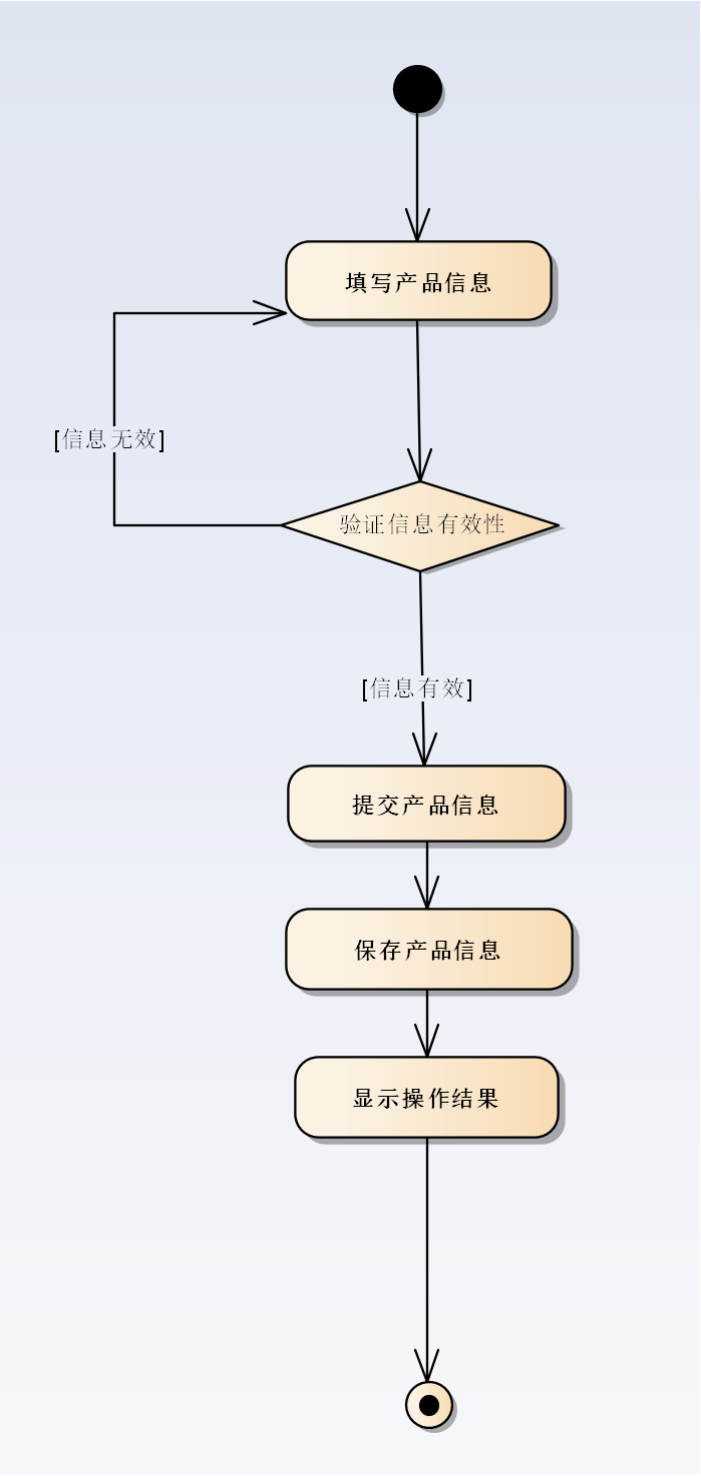


图 8.3.4 ：产品管理 – 活动图

主要活动： 开始添加产品：这是流程的起点。 填写产品信息：管理员在用户界面填写产品的详细信息，如名称、描述、价格、分类等。

验证信息有效性：系统检查输入的信息是否完整且有效。 判断信息是否有效： 如果信息有效，继续下一步。 如果信息无效，返回重新填写。 提交产品信息：将有效的产品信息提交到系统。 保存产品信息：系统处理提交的信息，并将其保存到数据库。 显示操作结果：根据保存操作的结果，向管理员显示成功或错误消息。 结束流程：添加产品的流程结束。

8.4 库存管理 UML 图

8.4.1 用例图

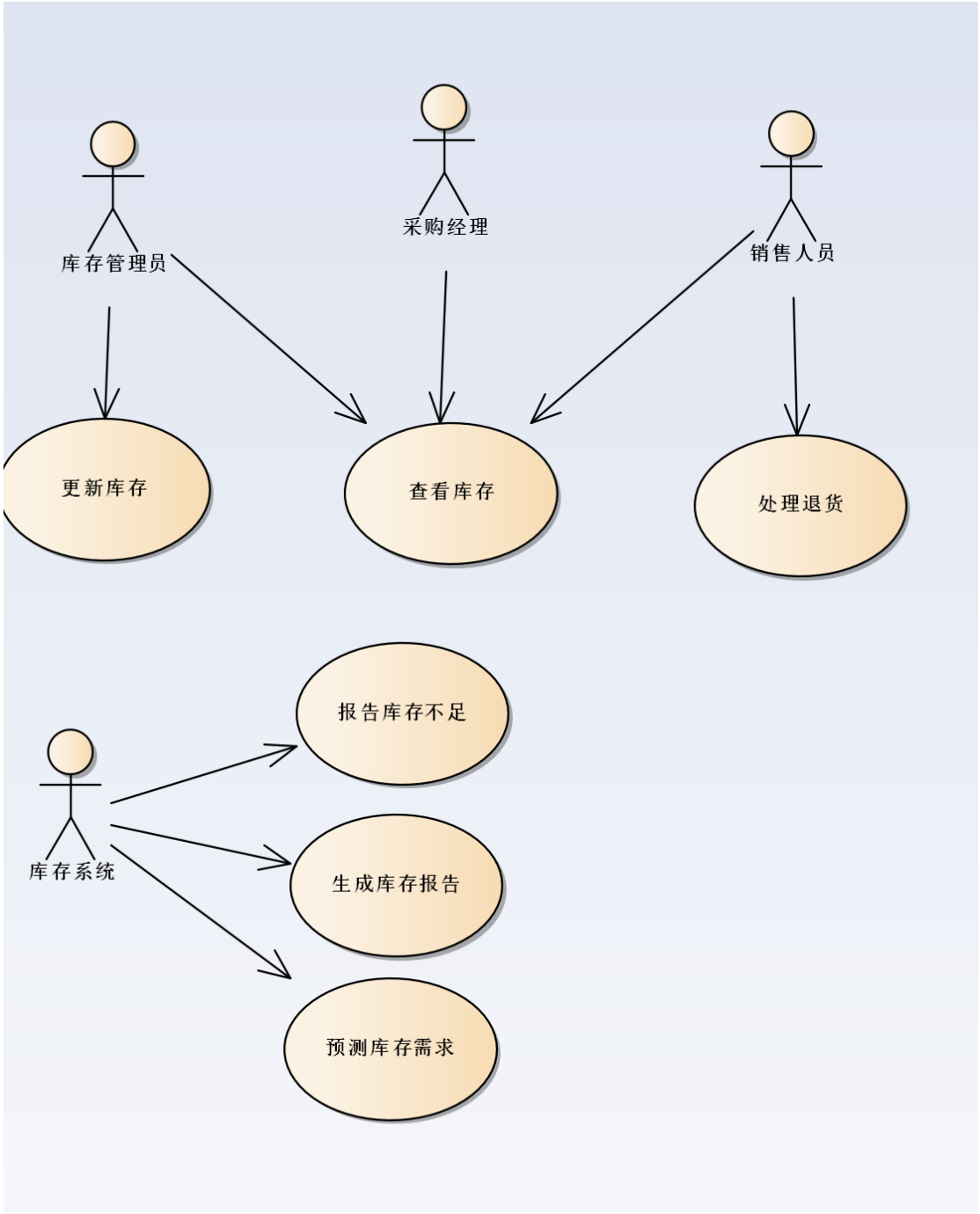


图 8.4.1 ：库存管理 – 用例图

参与者（Actors） 库存管理员：负责管理库存的日常操作。 销售人员：需要查看库存信息，以了解产品的可用性。

采购经理：负责补充库存，需要查看库存水平和历史数据。

用例（Use Cases）
查看库存：用户可以查看当前的库存水平。
更新库存：库存管理员更新库存数量，例如在收到新货物或销售产品后。
报告库存不足：系统自动报告低库存水平。
生成库存报告：生成关于当前库存状态的报告。
预测库存需求：基于历史数据预测未来的库存需求。
处理退货：管理退回的产品并更新库存。

8.4.2 类图

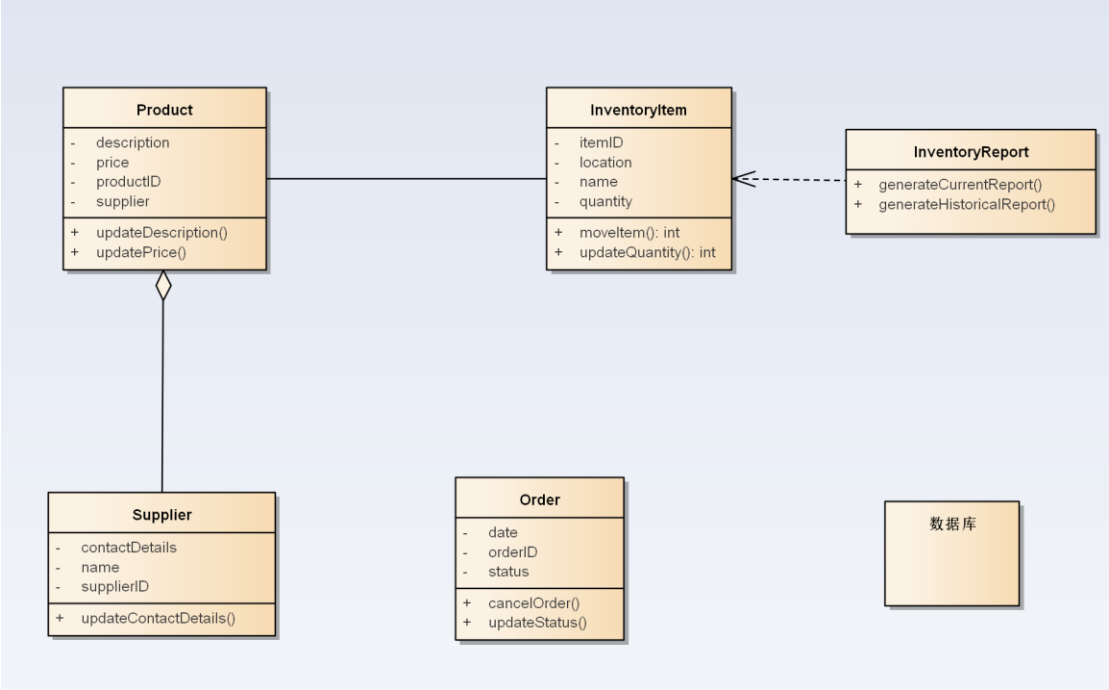


图 8.4.2 ：库存管理 – 类图

主要类
InventoryItem: 代表库存中的单个物品。
- 属性 itemID, name, quantity, location 等。
- 方法 updateQuantity(), moveItem() 等。
Product: 与 InventoryItem 相关联的产品详情。
- 属性 productID, description, price, supplier 等。
- 方法 updatePrice(), updateDescription() 等。
Supplier: 提供产品的供应商信息。
- 属性 supplierID, name, contactDetails 答。
- 方法 updateContactDetails() 等。
InventoryReport: 生成库存报告。

- 方法　generateCurrentReport(), generateHistoricalReport()　等。
Order: 如果库存管理涉及订单处理。
- 属性　orderID, date, status　等。
- 方法　updateStatus(), cancelOrder()　等。

关系
关联（Association）: InventoryItem　和　Product　之间有关联，表示库存项对应特定的产品。
聚合（Aggregation）: Supplier　与　Product　有聚合关系，表示供应商提供多种产品。
依赖（Dependency）: InventoryReport　依赖于　InventoryItem，以获取生成报告所需的数据。

8.4.3 顺序图

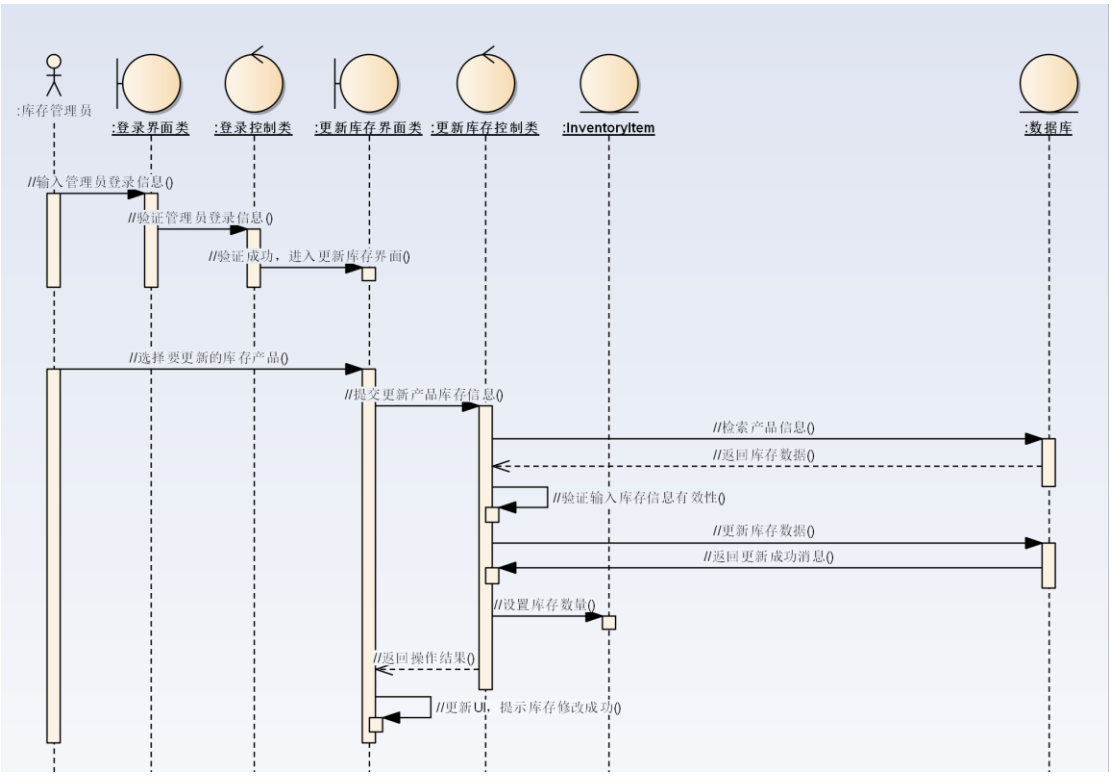


图 8.4.3 ：库存管理 – 顺序图

参与对象：
库存管理员：执行更新库存操作的用户。
库存管理系统：库存管理模块的主要组件，负责处理库存数据。
产品：代表库存中的一个产品。
数据库：用于存储库存数据。

交互流程：
库存管理员请求更新库存：库存管理员启动更新库存操作。
库存管理系统验证管理员身份：系统验证管理员的身份，确保他们有权限执行此操作。

库存管理员选择产品：管理员选择要更新库存的产品。
库存管理系统检索产品信息：系统从数据库中检索所选产品的信息，包括当前库存量。
库存管理员更新库存数量：管理员输入新的库存数量。
库存管理系统验证库存数量：系统验证输入的库存数量的有效性。
库存管理系统更新库存信息：如果库存数量有效，系统更新数据库中的库存信息。
库存管理系统生成更新确认：系统生成更新成功的确认消息。
库存管理员接收更新确认：管理员接收并查看更新成功的确认消息。

8.4.4 活动图

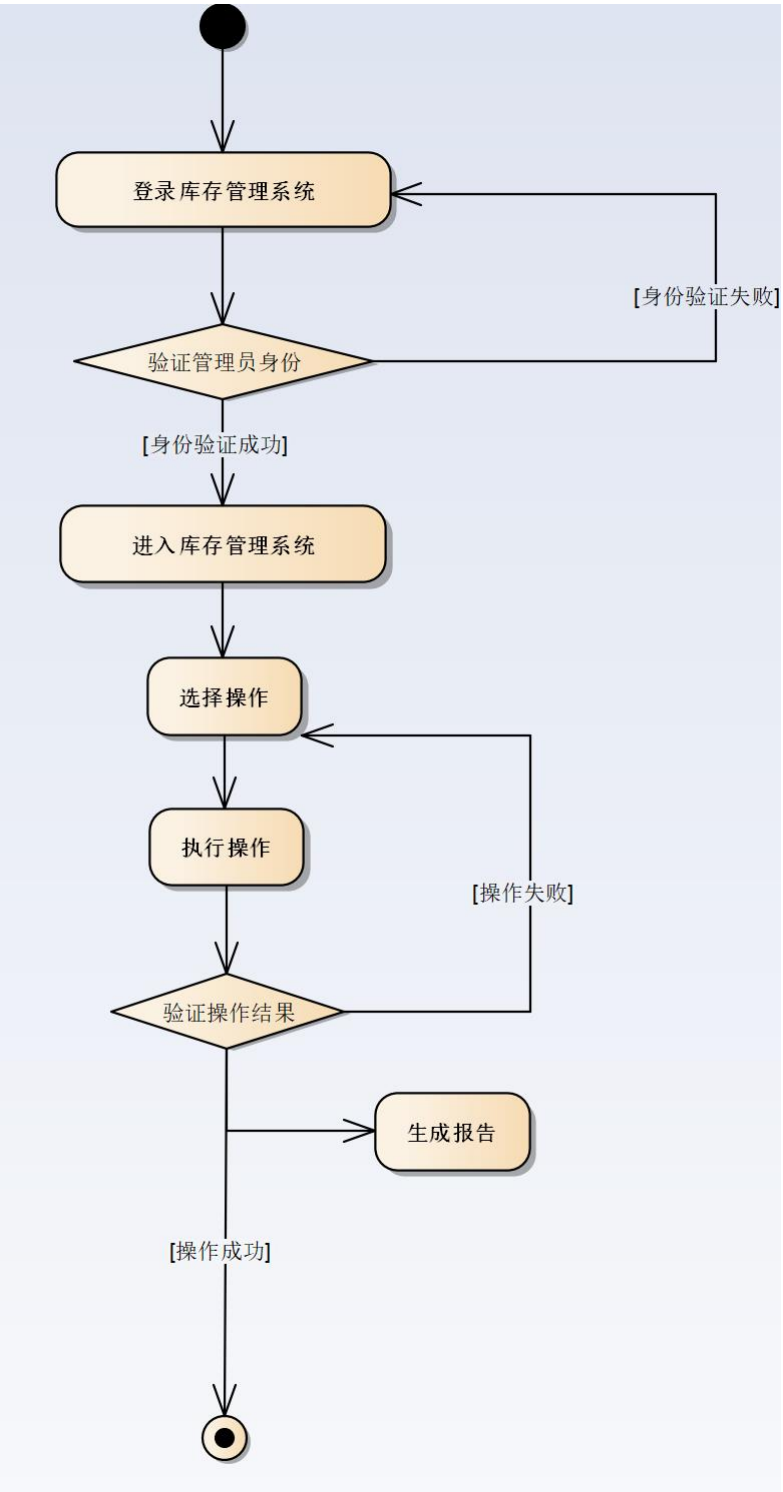


图 8.4.4 ：库存管理 – 活动图

主要活动：
开始操作：这是流程的起点。
登录库存管理系统：管理员登录库存管理系统以执行操作。
选择操作：管理员选择要执行的库存管理操作，例如更新库存、生成库存报告等。
执行操作：管理员执行所选操作的具体步骤。
验证操作结果：系统验证操作的结果，如果成功，进入下一步。

生成报告：如果需要，系统可以生成库存报告。 结束操作：操作流程结束。

8.5 系统实现

8.5.1 前端

（1）JSP (JavaServer Pages)

JSP 的全称是 JavaServer Pages，中文译为 Java 服务器页面。它是一种用于在服务器端生成动态网页的 Java 技术。通过 JSP，开发人员可以在 HTML 页面中嵌入 Java 代码，从而实现动态内容的生成和展示。JSP 技术使得开发者可以将 Java 代码和静态的标记语言（如 HTML、XML）组合在一起，以创建动态的、基于 Java 的 Web 应用程序。

（2）JQuery (Cross-platform JavaScript Library)

JQuery 的全称是"Cross-platform JavaScript Library"，中文翻译为"跨平台的 JavaScript 库"。它是一个快速、简洁的 JavaScript 库，设计用于简化 HTML 文档的遍历、事件处理、动画等操作。jQuery 使得在处理 HTML 文档中的 DOM 元素、执行动画、处理事件、发起 Ajax 请求等方面变得更加容易和便捷。它是一种流行的 JavaScript 库，被广泛用于 Web 开发中。

8.5.2 数据库

MySQL

MySQL 是一种开源的关系型数据库管理系统（RDBMS），广泛应用于 Web 开发和其他数据驱动的应用程序。作为一款强大而灵活的数据库，MySQL 采用表格形式存储数据，使用 SQL 进行数据管理和查询。它具有多用户、多线程的特性，能够同时处理多个用户的请求和多个查询操作。MySQL 的开源性质使其拥有庞大的社区支持，用户可以免费获取社区版，同时也提供了企业版以满足更高级的商业需求。由于其跨平台性、良好的性能和广泛的应用领域，MySQL 已经成为最流行的数据库之一，特别是在 LAMP 技术栈中与 PHP 等编程语言的集成中表现出色。

8.5.3 后端

（1）Java

Java 是一种流行且多用途的编程语言，于 1995 年由 Sun Microsystems 推出。其主要特点包括跨平台性、面向对象的设计、自动内存管理（垃圾回收）和丰富的库支持。Java 被广泛应用于企业级应用、移动应用开发（特别是 Android 平台）、大数据处理和云计算等领域。其安全性和可移植性使其成为构建稳健应用程序的首选语言之一。

（2）Spring

Spring 框架是一个开源的、轻量级的应用开发框架，为 Java 应用程序提供了广泛的基础设施支持。它简化了企业级应用的开发，通过依赖注入（DI）和面向切面编程（AOP）等特性，提供了松耦合、可测试和可扩展的开发环境。Spring 包括许多模块，如 Spring Core、Spring MVC、Spring Boot 等，提供了各种功能，从简化数据库访问到处理事务管理和构建 RESTful Web 服务。Spring 的灵活性和广泛的社区支持使其成为构建健壮、可维护和可扩展应用程序的首选框架之一。

（3）MyBatis

MyBatis 是一款开源的 Java 持久层框架，它简化了数据库操作的开发过程。通过将 SQL 语句与 Java 代码解耦，MyBatis 提供了灵活的映射方式，使得数据库交互变得更加直观和可维护。开发者可以使用 XML 或注解配置 SQL 映射，而不必编写繁琐的 JDBC 代码。MyBatis 支持动态 SQL、存储过程调用等特性，同时提供了高度的可定制性，使得开发者能够更好地控制 SQL 的执行过程。这使得 MyBatis 成为构建数据库访问层的常用工具之一。

（4）MVC 模式 (Model - View - Controller)

MVC（Model-View-Controller）模式是一种软件设计模式，用于组织和实现用户界面交互。其中，模型（Model）负责应用程序的数据和业务逻辑，视图（View）负责显示数据并与用户进行交互，而控制器（Controller）充当模型和视图之间的中介，处理用户输入并相应地更新模型和视图。MVC 的分离关注点和模块化设计使得应用程序更易于维护、扩展和测试，是许多软件开发框架和应用中广泛采用的设计范式。

8.5.4 维护

（1）版本管理：Git

Git 是一种分布式版本控制系统，广泛用于协作软件开发。由 Linus Torvalds 创立，Git 通过追踪文件的变化、支持多个并行开发分支、具备高效的合并能力等特性，使团队能够协同工作、追溯项目历史，确保代码的一致性和可追溯性。Git 的分布式架构允许开发者在本地进行工作，而后将变更推送到远程仓库，提供了强大的分支管理和版本控制能力，成为开发者日常版本控制的首选工具。

（2）代码托管：Github

GitHub 是一个基于 Git 的代码托管平台，为开发者提供了协作、版本控制和项目管

理的集成服务。在 GitHub 上，用户可以创建远程仓库存储其代码，进行版本控制，并通过分支管理实现并行开发。除此之外，GitHub 还提供了问题追踪、团队协作、持续集成等功能，使团队能够更加高效地协作开发。开源项目在 GitHub 上得到广泛分享和贡献，使其成为全球最大的开源社区之一。

8.5.5 单元测试

（1）JUnit Jupiter 框架

JUnit Jupiter 是 Java 生态系统中用于编写和运行单元测试的测试框架之一。作为 JUnit 5 的一部分，JUnit Jupiter 引入了新的扩展模型，支持注解驱动的测试，并提供了更丰富的功能，如参数化测试、动态测试注册等。它的扩展性和灵活性使得开发者能够更轻松地编写清晰、模块化的测试代码，同时利用新的测试引擎，JUnit Jupiter 提供了更好的性能和更丰富的测试功能，成为现代 Java 应用开发中常用的测试框架之一。

（2）AAA 方法 (Arrange – Act - Assert)

AAA（Arrange – Act - Assert）是一种测试方法论，通常用于编写单元测试。它的核心思想是将测试分为三个阶段：安排（Arrange）、执行（Act）和断言（Assert）。在安排阶段，准备测试环境，设置必要的先决条件；执行阶段进行被测试方法的调用或操作；最后，在断言阶段验证预期结果是否符合预期。AAA 方法框架有助于测试的组织和可读性，使得单元测试更加清晰、可维护，有利于快速定位和解决问题。

8.5.6 集成测试

（1）Spring Boot 框架

Spring Boot 是一个用于构建基于 Spring 框架的 Java 应用程序的开发框架。它通过提供默认配置和约定大于配置的原则，简化了 Spring 应用的搭建和开发过程。Spring Boot 集成了大量常用的第三方库，提供了开箱即用的特性，包括自动配置、嵌入式 Web 服务器、健康检查、安全性等。这使得开发者能够更专注于业务逻辑的实现而不必过多关注框架的配置。Spring Boot 同时支持各种构建工具，如 Maven 和 Gradle，并且提供了丰富的扩展点，使得开发者能够灵活定制和拓展应用。

（2）MyBatis Plus

MyBatis Plus 是基于 MyBatis 框架的一个增强工具包，简化了 MyBatis 的使用，提供了更便捷的持久层开发方式。MyBatis Plus 通过代码生成器、通用 Mapper、分页插件等功能，减少了开发者的重复工作，提高了开发效率。它支持注解和 XML 两种方式配置映射关系，提供了方便的 CRUD 操作接口，同时具备强大的查询构建和条件构造器，简化了复杂查询的编写。MyBatis Plus 的设计理念是简单、方便、高效，使得 Java 应用的数据访问层更

加轻松和灵活。

（3）MySQL 8.0

（4）JSP

（5）JQuery

8.5.7 性能测试

（1）Apache JMeter

Apache JMeter 是一个开源的 Java 框架，用于性能测试和负载测试。它被广泛用于评估和测量应用程序的性能，特别是在 Web 应用程序的开发和测试过程中。JMeter 支持多种协议，包括 HTTP、FTP、JMS、SOAP、数据库等，使其能够模拟多种场景下的用户行为和请求。其图形化的用户界面简化了测试脚本的创建和管理，同时具备强大的分析和报告功能，帮助开发者和测试人员深入了解应用程序的性能状况。Apache JMeter 的灵活性和可扩展性使其成为一个强大的性能测试工具，适用于各种复杂的测试需求。

（2）LoadRunner

LoadRunner 是一款由 Micro Focus 公司推出的性能测试工具，用于模拟和评估应用程序在不同负载条件下的性能表现。LoadRunner 支持多种协议，包括 Web、数据库、消息队列等，能够模拟大量用户并监测系统资源的利用情况。其独特之处在于分布式负载生成器的能力，使得可以模拟真实场景中大规模用户同时访问的情况。LoadRunner 提供了直观的用户界面和强大的脚本编辑器，使测试人员能够轻松创建、执行和分析性能测试。通过实时监测和分析结果，LoadRunner 帮助开发者发现潜在的性能瓶颈，确保应用程序在各种条件下都能提供良好的用户体验。

8.5.8 云服务

AWS (Amazon Web Services)

Amazon Web Services（AWS）是亚马逊公司提供的全球领先的云计算平台。AWS 提供了丰富的云服务，包括计算、存储、数据库、人工智能、机器学习、物联网等，帮助个人、企业和政府机构构建灵活、可扩展且安全的云基础设施。其弹性计算、自动化部署和按需支付等特性，使用户能够根据实际需求弹性地使用计算资源，同时降低了硬件和维护成本。AWS 的全球性质和强大的生态系统使其成为业界领先的云服务提供商，为创新和业务发展提供了可靠的技术基础。

9. 系统测试

9.1 系统测试计划

9.1.1 范围

9.1.1.1 系统概述

冰酒订购系统旨在简化冰酒供应商接受和履行冰酒订单的流程。它旨在取代将订单记录在卡片上并存储在数据库中的手动方法。有了这个系统，顾客可以通过电话或直接访问商店下单。收到订单后，冰酒供应商输入客户的详细信息、冰酒要求，并使用系统计算成本。这确保了准确的定价和高效的订单管理。该系统中描述的功能旨在在初始版本 1.0 中完全实现。然而，该系统被设计为适应未来的增强功能和后续版本中可能需要的附加功能。这种灵活性确保了冰酒订购系统能够适应不断变化的业务需求和客户要求。

9.1.1.2 文档描述

编写这份测试计划是为了给软件测试提供执行依据。文档描述了测试计划、测试用例和测试过程。

9.1.2 参考

本文档参考如下文档进行编写：

（一） 《冰酒销售管理系统需求分析报告》
（二） 《冰酒销售管理系统系统原型报告》
（三） 《计算机软件文档编制规范》

9.1.3 测试计划

9.1.3.1 测试进度

（一）单元测试

（1）计划时间:

开始日期: 2023 年 12 月 18 日
结束日期: 2023 年 12 月 20 日

（2）目标:

确保各个模块和组件在孤立的环境中运行正常，验证其功能的正确性和稳定性。

（3）活动：

1）编写单元测试用例，覆盖每个模块的所有功能。
2）开展单元测试，发现并修复潜在的代码缺陷。
3）评估每个模块的性能和资源利用情况。

（4）里程碑：

单元测试完成报告提交：2023 年 12 月 20 日

（二）集成测试

（1）计划时间：

开始日期：2023 年 12 月 21 日
结束日期：2023 年 12 月 23 日

（2）目标：

验证各个模块之间的接口和交互，确保整个系统在集成时能够正常协同工作。

（3）活动：

1）编写单元测试用例，测试模块间的数据流和信息交换。
2）模块集成后，进行功能性和性能方面的测试。
3）识别和解决模块集成引起的问题。

（4）里程碑：

单元测试完成报告提交：2023 年 12 月 23 日

（三）系统测试

（1）计划时间：

开始日期：2023 年 12 月 24 日
结束日期：2023 年 12 月 25 日

（2）目标：

在模拟生产环境中对整个系统进行综合测试，验证系统的稳定性、性能和安全性。

（3）活动：

1）编写系统测试用例，覆盖业务流程和用户场景。
2）执行系统测试，模拟真实用户操作，包括插入、删除、查询和更新等操作。
3）进行性能测试，确保系统在负载和压力下仍能正常运行。

（4）里程碑：

单元测试完成报告提交：2023 年 12 月 25 日

9.1.3.2 测试方案

（一）单元测试

（1）编写单元测试用例，覆盖每个模块的所有功能。
（2）开展单元测试，发现并修复潜在的代码缺陷。
（3）评估每个模块的性能和资源利用情况。

（二）集成测试

（1）编写单元测试用例，测试模块间的数据流和信息交换。
（2）模块集成后，进行功能性和性能方面的测试。
（3）识别和解决模块集成引起的问题。

（三）系统测试

（1）编写系统测试用例，覆盖业务流程和用户场景。
（2）执行系统测试，模拟真实用户操作，包括插入、删除、查询和更新等操作。
（3）进行性能测试，确保系统在负载和压力下仍能正常运行。

9.1.3.3 测试环境

（一）硬件环境

（1）服务器

1）用途：托管冰酒销售管理系统
2）操作系统：Windows Server
3）处理器：双核
4）内存：8 GB RAM
5）存储：100 GB 可用硬盘空间

（2）数据库服务器

1）用途：存储系统数据的数据库服务器
2）操作系统：Windows Server
3）处理器：双核
4）内存：8 GB RAM
5）存储：200 GB 可用硬盘空间

（3）客户端

1）用途：用于访问冰酒销售管理系统的计算机
2）浏览器：Google Chrome 、Mozilla Firefox 、Microsoft Edge

（二）软件环境

（1）应用服务器

1）用途：托管冰酒销售管理系统
2）操作系统：Windows Server
3）处理器：双核
4）内存：8 GB RAM
5）存储：100 GB 可用硬盘空间

（2）数据库

1）数据库管理系统：MySQL 5.7.31
2）数据库连接池：Apache Commons DBCP

（3）测试工具

1）测试框架：JUnit 、TestNG
2）浏览器自动化测试工具：Selenium WebDriver

（4）版本控制

1）版本控制系统：Git
2）代码托管平台：Github

（三）网络环境

（1）内部网络

用途：用于应用服务器和数据库服务器之间的通信

（2）外部网络

用途：用于模拟真实用户通过互联网访问系统的情况

9.1.3.4 测试人员安排

测试板块		测试人员
单元测试		张艺轩、叶可禾
集成测试		张艺轩、叶可禾
系统测试	后台测试	样例设计：王曦　　测试：陈婷
	前台测试	样例设计：陈婷　　测试：王曦
	性能测试	王曦、陈婷

9.1.4 测试说明

在系统测试阶段，我们对冰酒销售管理系统的各个方面进行了详细的测试，涵盖了用户管理、产品管理、销售订单、库存管理和报表生成等核心功能。以下是对每个方面的详细测试结果的简要介绍：

（一） 用户管理

用户管理模块经过测试，包括用户注册、登录、密码修改等功能。测试覆盖了不同用户角色的权限验证，确保系统能够正确地限制用户对敏感信息和操作的访问。

（二） 产品管理

产品管理模块测试主要集中在添加、编辑和删除产品信息的功能上。此外，我们验证了产品分类和标签的管理，确保系统在产品信息方面的准确性和完整性。

（三） 销售订单

销售订单模块经过测试，包括订单的创建、编辑和删除。我们特别关注了订单状态的更新，以确保系统能够及时和正确地反映销售订单的实际状态。

（四） 库存管理

库存管理模块测试覆盖了库存数量的准确性，以及入库和出库操作。我们验证了系统对库存变化的追踪和管理，确保库存数据的一致性。

（五） 报表生成

报表生成模块的测试主要包括销售报表和库存报表的生成。我们验证了报表的准确性和及时性，确保用户能够方便地获取系统关键数据进行分析和决策。

（六） 缺陷报告

在测试过程中，我们发现了一些缺陷，主要涉及订单状态更新的延迟和跨浏览器兼容性。这些缺陷已被详细记录，并将在后续版本中进行修复和改进。

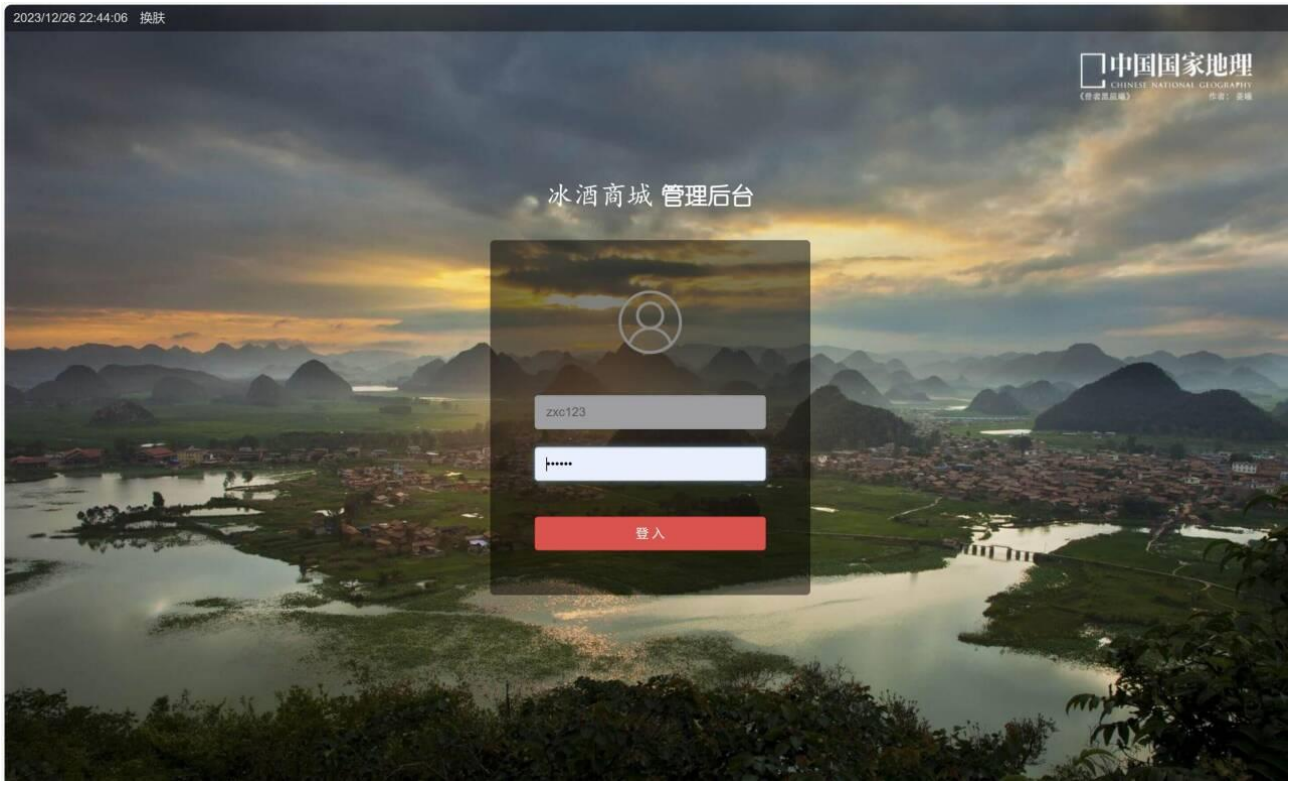
（七） 建议和改进

基于测试结果，我们提出了改进建议，包括优化订单状态更新流程和进行跨浏览器兼容性测试。这些建议旨在提升系统的质量和用户体验。

详细测试结果的分析将为开发团队提供有力的反馈，以便他们能够更好地理解系统的强项和改进的空间，进而提高系统的整体质量。

9.1.4.1 后台功能测试

9.1.4.1.1 系统进入测试



测试用例清单

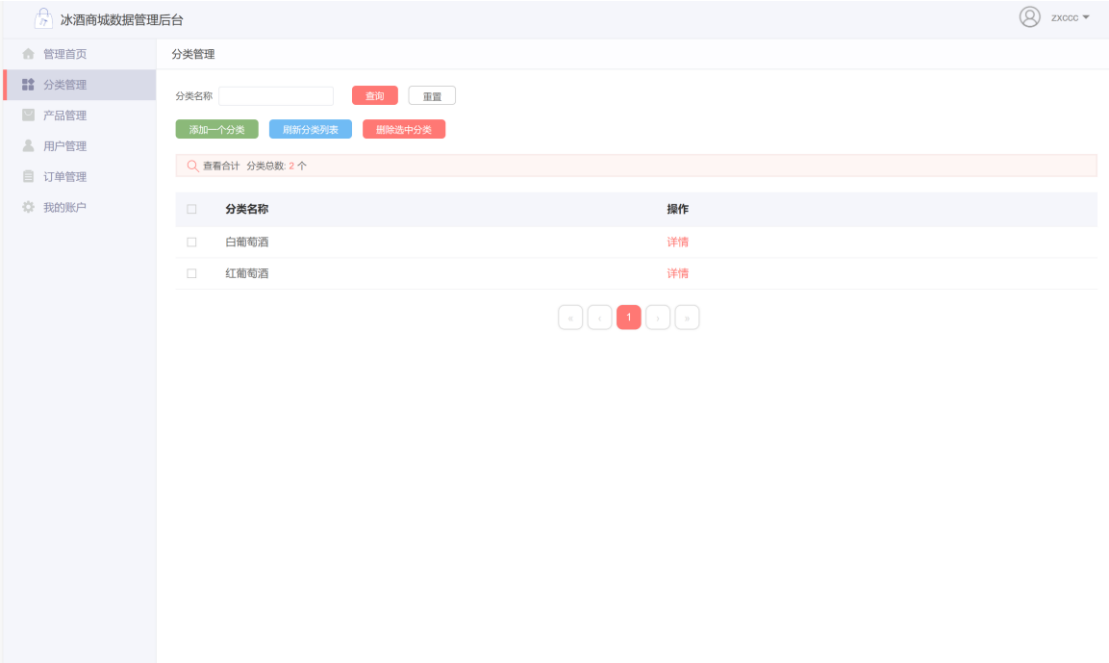
序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	目的：检测系统登录操作成功	用例 1 （正确）： 用户名： 密码：	登录进入后台	
2	目的：检测系统登录操作不成功及其反馈	用例 2 （不正确）： 用户名： 密码：	提示出错信息	

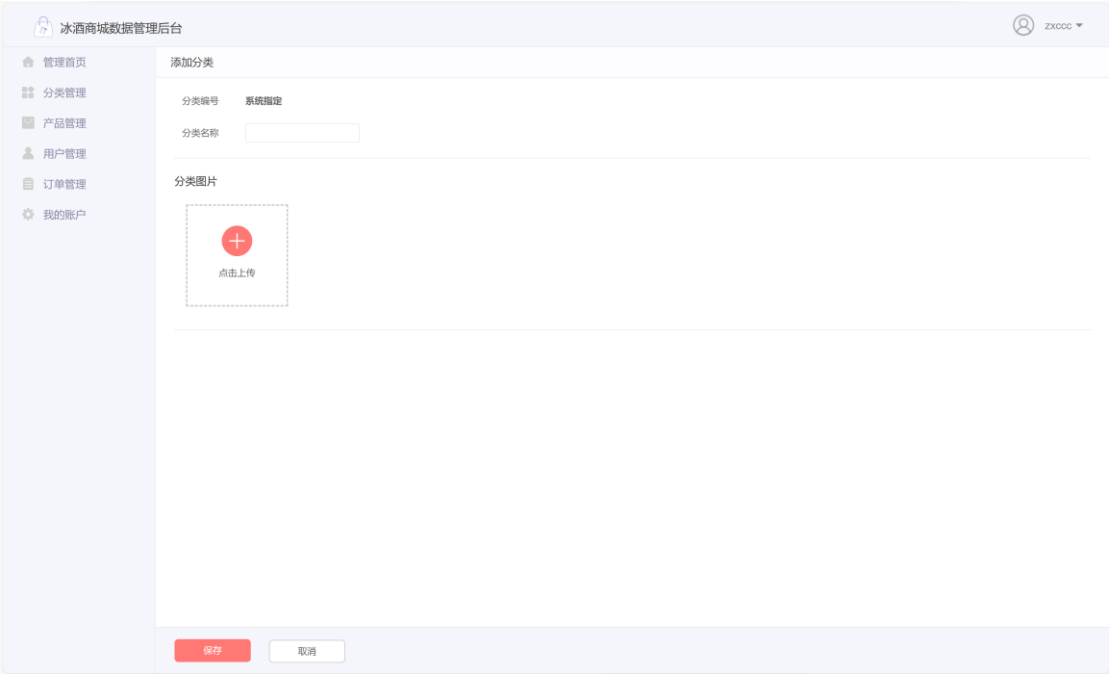
9.1.4.1.2 管理首页测试

测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	目的：检测一段时间内对上架产品数量、注册用户数量、成交订单数量、商城订单交易额的统计是否正确	用例 1： 统计时间区间选择 2023/12/19 到 2023/12/25	上架商品数量：21 注册用户数量：2 成交订单数量：0 商城订单交易额：0	
		用例 2： 上架一件商品后，统计时间区间选择 2023/12/25 到 2023/12/25	上架商品数量：22 注册用户数量：2 成交订单数量：0 商城订单交易额：0	
		用例 3： 注册一个新用户后，统计时间区间选择 2023/12/25 到 2023/12/25	上架商品数量：22 注册用户数量：3 成交订单数量：0 商城订单交易额：0	

9.1.4.1.3 分类管理测试



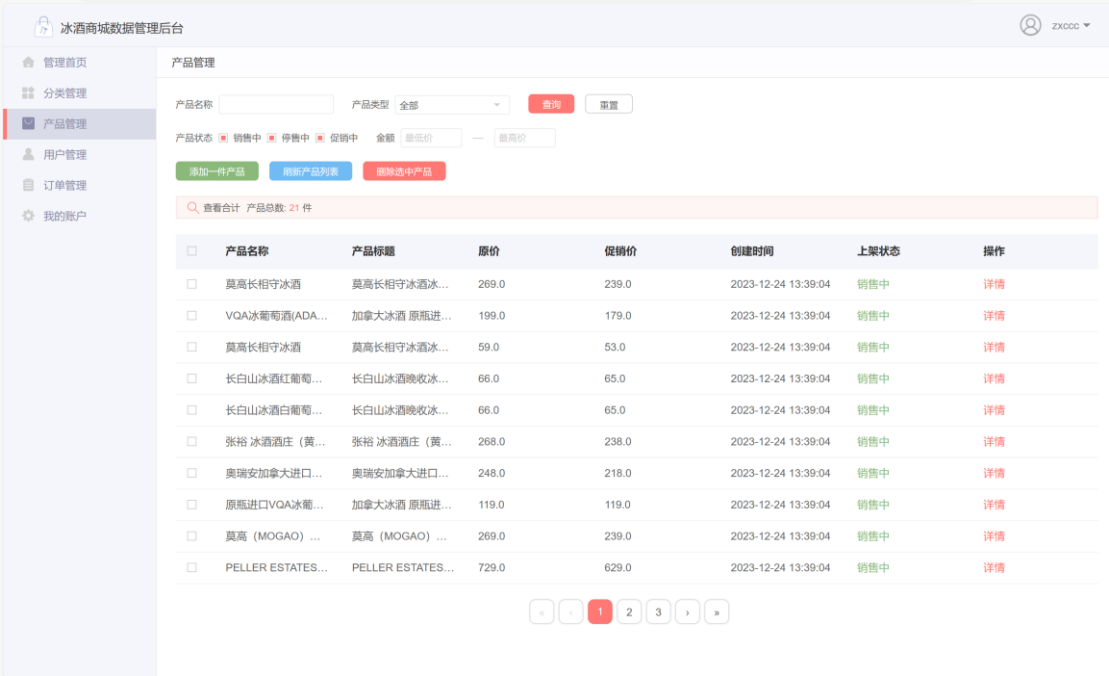


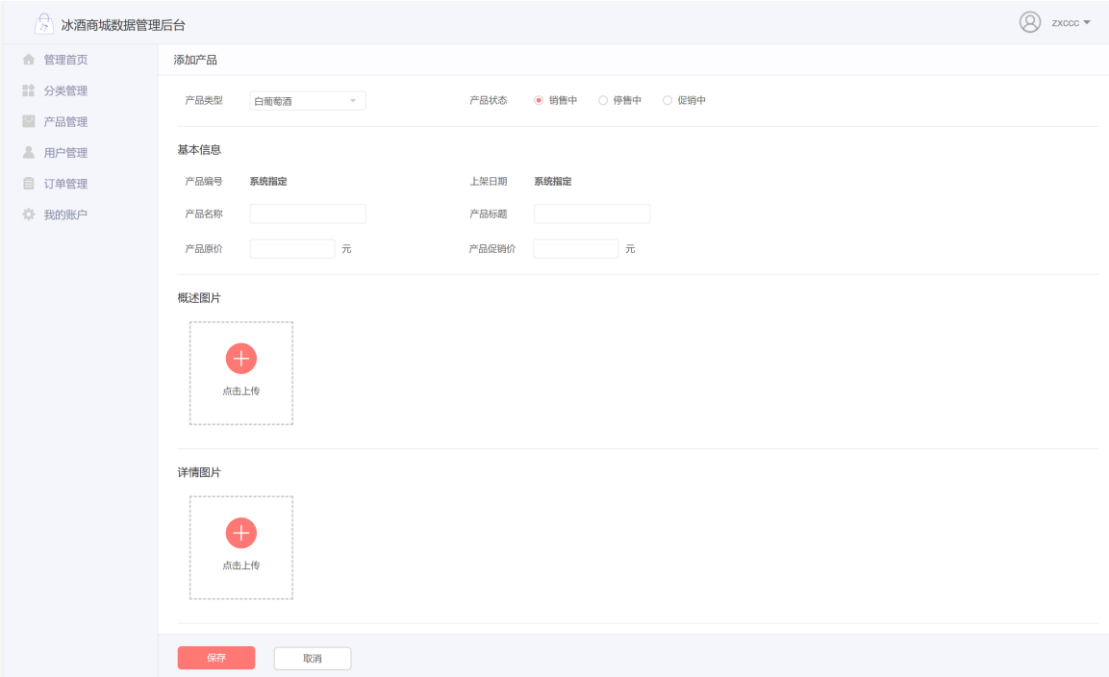
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 检测分类管理页面的“详情 ”操作是否正确	**用例 1 ：** 点击分类“常温保存”右侧的“详情”	显示分类“常温保存”的详情	
2	**目的：** 检测分类管理页面的“查询 ”操作是否正确	**用例 2 ：** 在“分类名称”的文本框中输入“保存”后，点击右侧的“查询”按钮	列表显示名称有关键词“保存”的分类	

		用例 3 ： 在“分类名称”的文本框中输入空值后，点击右侧的“查询”按钮	列表显示所有分类	
3	**目的：** 测试分类管理页面的“重置 ”操作是否正确 先决条件：先前进行过查询	**用例 4 ：** 在“分类名称”的文本框中输入“保存”后，点击右侧的“查询”按钮，再点击“重置”按钮	点击“查询”按钮后，列表显示名称有关键词“保存”的分类；点击“重置”按钮后，“分类名称”的文本框清空，列表显示所有分类	
4	**目的：** 测试分类管理页面的“添加一个分类 ”操作是否正确	**用例 5 ：** 点击“添加一个分类”，在“添加分类”页面的“分类名称”中输入“测试分类”，并上传“分类图片”后，点击下方的“确定”按钮	成功创建“测试分类”，列表中显示该分类，在详情中可看到该分类的图片为上传的图片	
		用例 6 ： 点击“添加一个分类”，在“添加分类”页面的“分类名称”中输入空值，并上传“分类图片”后，点击下方的“确定”按钮	显示提示信息：“分类名称”不能为空	
		用例 7 ： 点击“添加一个分类”，在“添加分类”页面的“分类名称”中输入“测试分类”，不上传“分类图片”，点击下方的“确定”按钮	显示提示信息：“分类图片”不能为空	
5	**目的：** 测试分类管理页面的“刷新分类列表 ”操作是否正确	**用例 8 ：** 用命令行添加分类“测试分类”后，点击“刷新分类列表”按钮	列表刷新后显示用命令行添加的分类	
6	**目的：** 测试分类管理页面的“删除选中分类”操作是否正确	**用例 9 ：** 在列表中选择分类“测试分类”后，点击“删除选中分类”按钮，点击“刷新分类列表”按钮	列表刷新后不再显示分类“测试分类”	

9.1.4.1.4 产品管理测试





测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试“产品管理”页面中“查询”功能是否正确	**用例 1 ：** 在“产品名称”的文本框中输入“长白山”，点击“查询”按钮	列表显示“产品名称”含关键字“长白山”的产品	
		用例 2 ： 在“产品类型”的选择框	列表显示类别为“白葡萄酒”的产品	

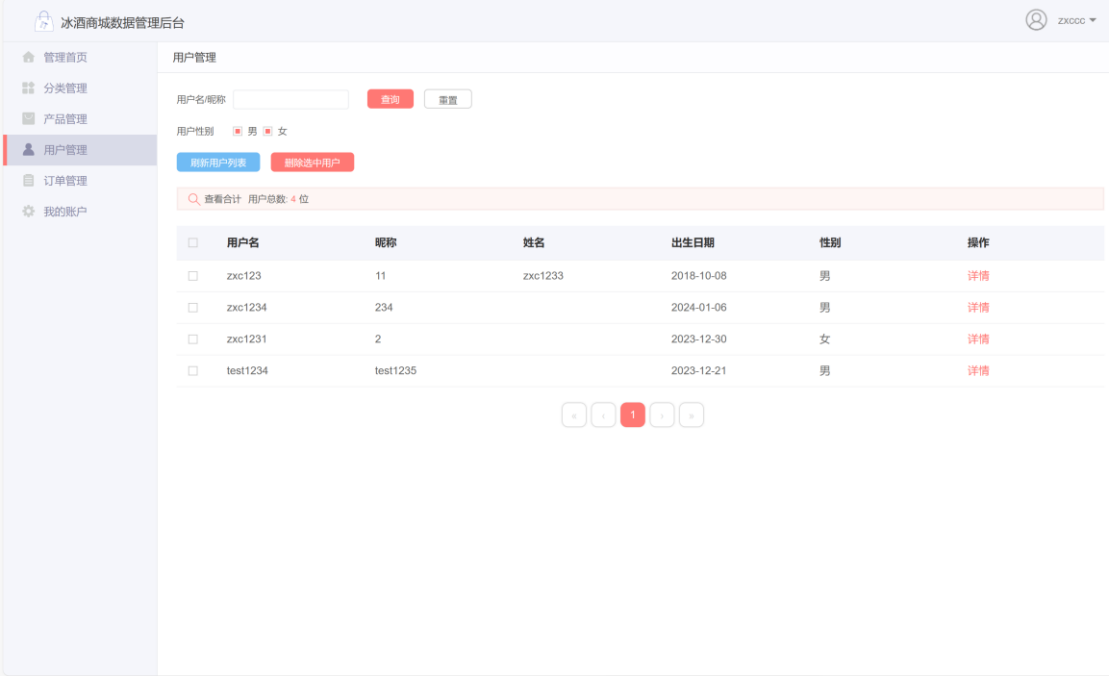
		中选择“白葡萄酒”，点击“查询”按钮		
		用例 3 ： 在“产品状态”的复选框中选择“销售中”，点击“查询”按钮	列表显示“产品状态”为“销售中”的产品	
		用例 4 ： 在“金额”的“最低价”和“最高价”中分别输入“300”和“500”，点击“查询”按钮	列表显示“销售价”介于 300 到 500 的产品	
		用例 5 ： 在“产品名称”的文本框中输入“长白山”，点击“查询”按钮，再点击“重置”按钮	点击“查询”按钮后列表显示“产品名称”含关键字“长白山”的产品；点击“重置”按钮后列表显示所有产品	
2	**目的：** 测试“产品管理”页面中“详情”功能是否正确	**用例 1 ：** 点击 产品“莫高长相守冰酒”右侧的“详情”	显示产品“莫高长相守冰酒”的详情	
3	**目的：** 测试“产品管理”页面的“添加一件产品”操作是否正确	**用例 1 ：** 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选择“停售中”，在“产品名称”的文本框中输入“测试白葡萄酒名称”，在“产品标题”的文本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮	列表中出现“产品名称”为“测试白葡萄酒名称”、“产品标题”为“测试白葡萄酒标题”、“原价”为 400 、“促销价”为 300 、创建时间为测试时刻、上架状态为“停售中”的产品	
		用例 2 ： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择 框中选择“白葡萄酒”，在 “产品状态”的单选框中选	显示提示信息：“产品名称”不能为空	

		择“停售中”，在“产品名称”的文本框中输入“空值”，在“产品标题”的文本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮		
		用例 3： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选择“停售中”，在“产品名称”的文本框中输入“测试白葡萄酒名称”，在“产品标题”的文本框中输入“空值”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮	显示提示信息：“产品标题”不能为空	
		用例 4： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选择“停售中”，在“产品名称”的文本框中输入“测试白葡萄酒名称”，在“产品标题”的文本框中输入	显示提示信息：“产品原价”不能为空	

		“测 试白葡萄酒标题”，在“产品原价”中输入“空值”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮		
		用例 5： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“测 试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“空值”，上传“概述图片”和“详情图片”，点击“保存”按钮	显示提示信息：“促销价”不能为空	
		用例 6： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“测 试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，不上传“概述图片”，上传“详情图片”，	显示提示信息：“概述图片” 不能为空	

		点击“保存”按钮		
		用例 7 ： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“测 试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图片”，不上传“详情图片”，点击“保存”按钮	显示提示信息：“详情图片” 不能为空	
4	**目的：** 测试“产品管理”页面的“刷新产品列表”操作是否正确	**用例 1 ：** 用命令行添加一个“产品类型”为“红葡萄酒”、“产品状态”为“促销中”、“产品名称”为“测试红葡萄酒名称”、“产品标题”为“测试红葡萄酒标题”、“产品原价”为“500”、“促销价”为“400”、“概述贴图”和“详情贴图”任意设 置的产品后，点击“刷新产品列表”按钮	列表刷新后显示新加入的产品	
5	**目的：** 测试“产品管理”页面的“删除选中产品”操作是否正确	**用例 1 ：** 在列表中选中产品“测试红葡萄酒名称”，点击“删除选中产品”按钮	列表中产品“测试红葡萄酒名称”被删除	

9.1.4.1.5 用户管理测试

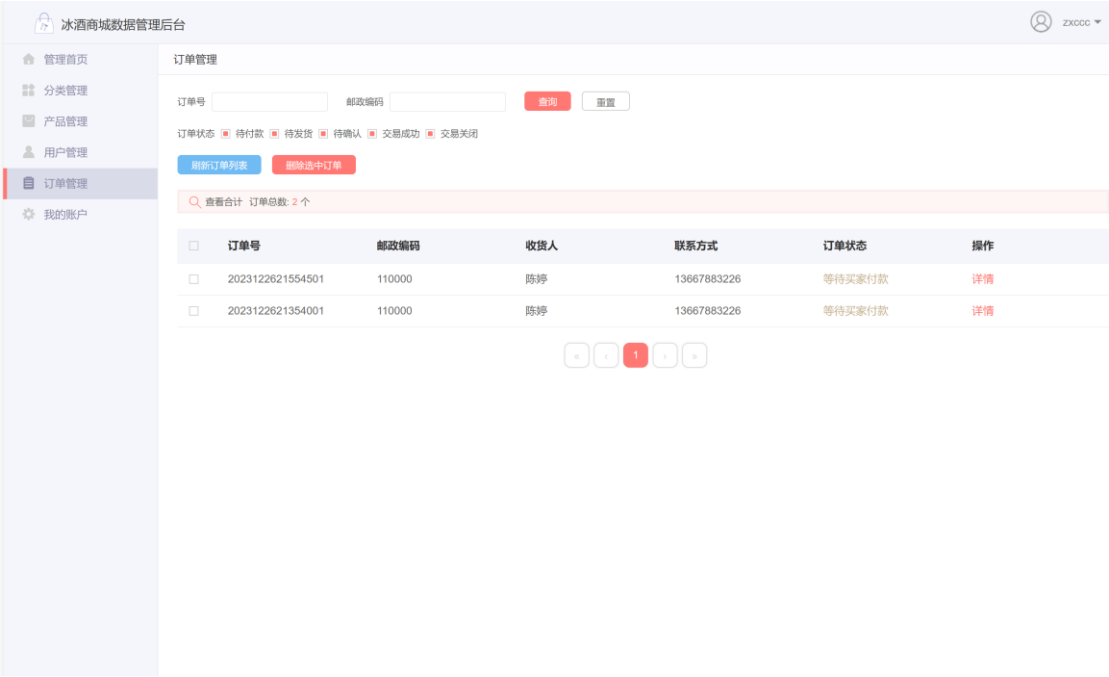


测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试“用户管理”页面的“查询”操作是否正确	**用例 1：** 在“用户名/昵称”的文本框中输入“zxc”，点击“查询”按钮	列表显示“用户名/昵称”含关键词“zxc”的用户	
		用例 2： 在“用户性别”的复选框中选择“男”，点击“查询”按钮	列表显示“用户性别”为“男”的用户	
2	**目的：** 测试“用户管理”页面的“重置”操作是否正确	**用例 1：** 在“用户名/昵称”的文本框中输入“zxc”，点击“查询”按钮，再点击“重置”按钮	点击“查询”按钮后列表显示“用户名/昵称”含关键词“zxc”的用户；点击“重置”按钮后列表显示所有用户	
3	**目的：** 测试“用户管理”页面的“刷新用户列表”操作是否正确	**用例 1：** 在登录页面注册一个新用户“test”后，点击“刷新用户列表”按钮	列表显示新注册的用户	
4	**目的：** 测试“用户管理”页面的“删除选择用户”操作是否正确	**用例 1：** 在列表中选中“用户名”为“test”的用户，点击“删除选择用户”按钮	列表中用户“test”被删除	
5	**目的：** 测试“用户管理”页面的	**用例 1：** 点击列表中“用户名”为	显示用户“zxc123”的详情	

	“详情”操作是否正确	“zxc123”的用户右侧的“详情”		

9.1.4.1.6 订单管理测试

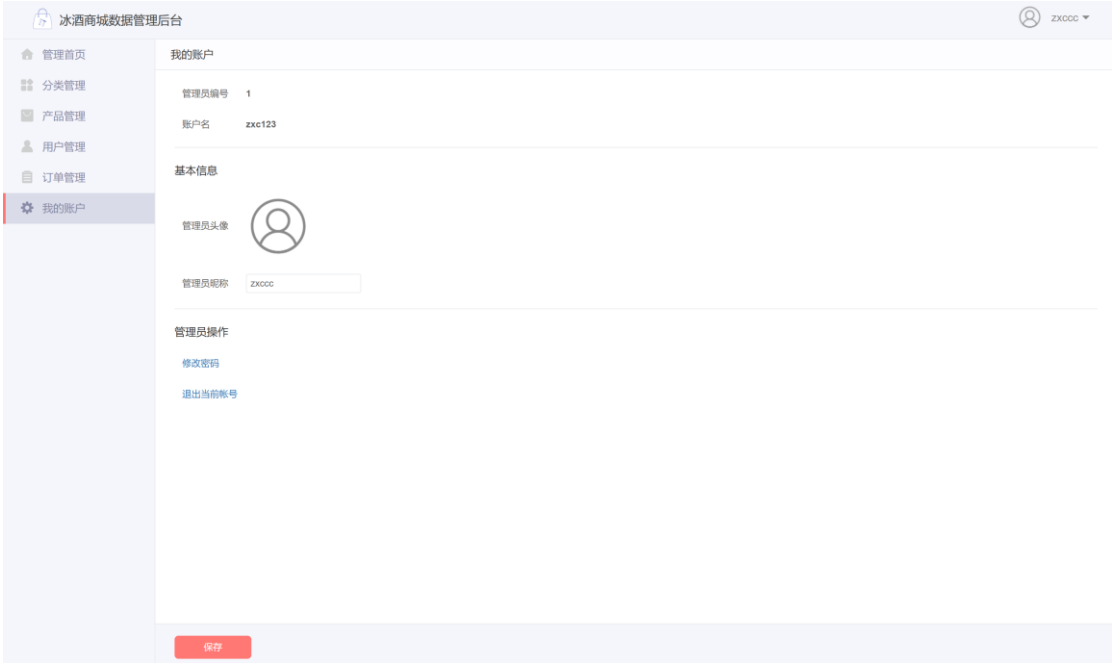


测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试“订单管理”页面的	**用例 1：** 在“订单号”的文本框输	列表显示“订单号”包含关键词“2021”的订单	

	“查询”操作是否正确	入“2021”，点击“查询”按钮		
		用例 2： 在“邮政编码”的文本框中输入“110000”，点击“查询”按钮	列表显示“邮政编码”包含关键词“110000”的订单	
		用例 3： 在“订单状态”的复选框中选择“待发货”，点击“查询”按钮	列表显示“订单状态”为“待发货”的订单	
2	**目的：** 测试“订单管理”页面的“详情”操作是否正确	**用例 1：** 点击列表中订单号为“2021102209242001”的订单右侧的“详情”	显示订单号为“2021102209242001”的订单右侧的详情	
3	**目的：** 测试“订单管理”页面的“刷新订单列表”操作是否正确	**用例 1：** 用命令行添加一个“订单号”为“2023102509242001”、“邮政编码”为“110000”、“收货人”为“陈婷”、“订单状态”为“已付款”的订单后，点击“刷新订单列表”按钮	列表显示新加入的“订单号”为“2023102509242001”、“邮政编码”为“110000”、“收货人”为“陈婷”、“订单状态”为“已付款”的订单	
4	**目的：** 测试“订单管理”页面的“删除选中订单”操作是否正确	**用例 1：** 选中列表中订单号为“2021102209242001”的订单，点击“删除选中订单”	列表中订单号为“2021102209242001”的订单被删除	

9.1.4.1.7 **我的账户测试**



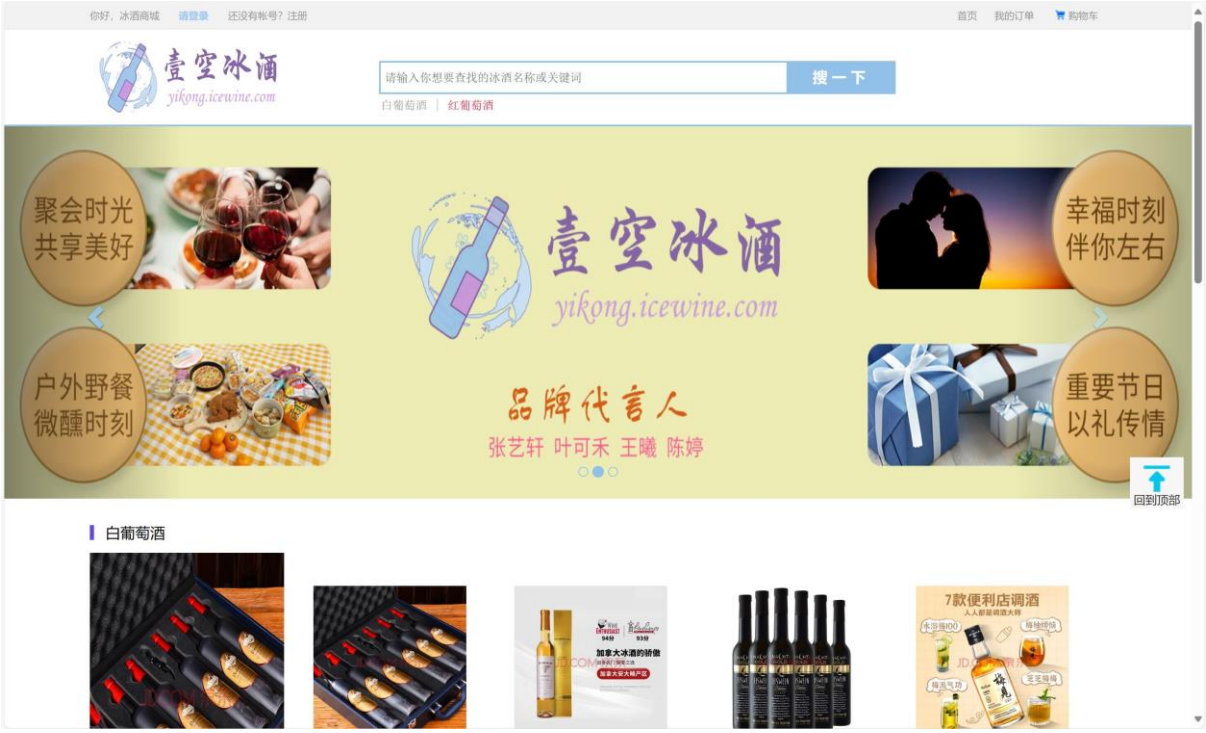
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试“我的账户”页面中修改“管理员昵称”是否正确 **先决条件：** 已登录账号	**用例 1 ：** 将“管理员昵称”的文本框中的内容改为“ct”，点击下方的“保存”	刷新页面后，右上角的用户昵称为“ct”	
2	**目的：** 测试“我的账户”页面中“修改密码”操作是否正确 **先决条件：** 已登录账号	**用例 1 ：** 在“修改密码”中“当前密码”的文本框中输入当前账号的密码，在“新密码”的文本框中输入要设定的新密码，在“确认密码”的文本框中再次输入要设定的新密码，点击下方的“保存”按钮	当前账号的密码更新为新密码，在后台的登陆页面使用旧密码无法登录，使用新密码可以登录	
		用例 2 ： 在“修改密码”中“当前密码”的文本框中输入错误的密码，在“新密码”的文本框中输入要设定的新密码，在“确认密码”的文本框中再次输入要设定的新密码，点击下方的“保存”按钮	输出提示信息：“当前密码”错误	

		用例 3 ： 在“修改密码”中“当前密码”的文本框中当前账户的密码，在“新密码”的文本框中输入当前账户的密码，在“确认密码”的文本框中再次输入当前账户的密码，点击下方的“保存”按钮	输出提示信息：“新密码”与“当前密码”相同	
		用例 4 ： 在“修改密码”中“当前密码”的文本框中输入当前账号的密码，在“新密码”的文本框中输入要设定的新密码，在“确认密码”的文本框中输入错误的要设定的新密码，点击下方的“保存”按钮	输出提示信息：“新密码”与“确认密码”不一致	
3	**目的：** 测试“我的账户”页面中“退出当前账号”操作是否正确 **先决条件：** 已登录账号	**用例 1 ：** 点击“退出当前账号”按钮	当前账号被退出，回到后台的登录页面	

9.1.4.2 前台功能测试

9.1.4.2.1 首页测试

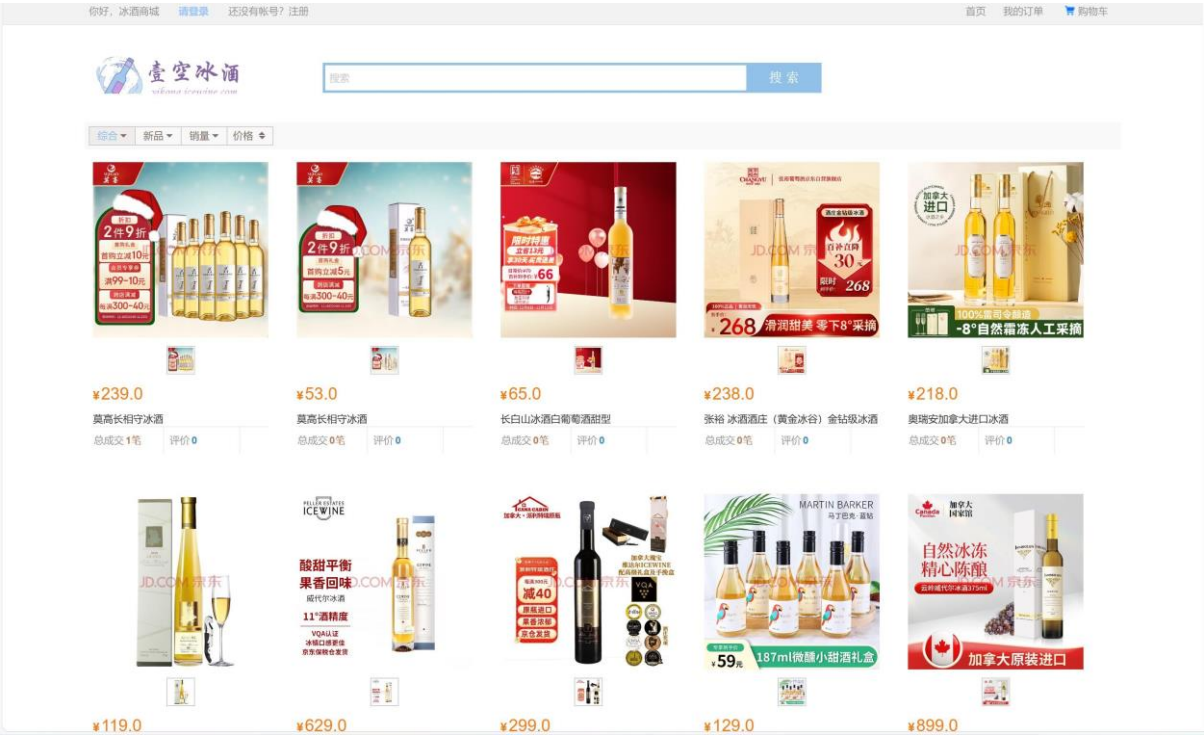


测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 首页 的“搜索框”操作是否正确	**用例 1 ：** 在 搜索 文本框输入“ 梅见【mini 版】青梅酒 ”，点击“搜一下”按钮	显示搜索结果中有且仅有“ 梅见【mini 版】青梅酒 ”这一商品	精确全名搜索
		用例 2 ： 在 搜索 文本框输入“ 葡萄 ”，点击“搜一下”按钮	显示搜索结果中有所有包含“ 葡萄 ”这一关键字的商品	关键词搜索
		用例 3 ： 在 搜索 文本框输入“ 玫瑰 ”，点击“搜一下”按钮	显示没有找到相关商品	搜索不存在的商品
2	**目的：** 测试 首页 的 导航菜单是否正常跳转	**用例 1 ：** 点击搜索框下方导航栏的 “ 白葡萄酒”	跳转到“白葡萄酒”商品展示界面，显示该种类下所有商品	查看筛选是否准确
		用例 2 ： 点击搜索框下方导航栏的 “ 红葡萄酒”	跳转到“红葡萄酒”商品展示界面，显示该种类下所有商品	

3	**目的：** 测试 首页 的宣传栏是否正常工作，能否切换不同图片	**用例 1 ：** 多次点击 “下一页” 按键，看图片能否正常切换	三张图片按照正常顺序切换	
		用例 2： 多次点击 “上一页” 按键，看图片能否正常切换		
4	**目的：** 测试 首页的 “回到顶部” 按键是否正常工作	**用例 1 ：** 下拉页面，多次滑动到不同位置，点击“回到顶部”，看页面能否回到顶部	点击 ”回到顶部“ 后页面成功回到顶部	
5	**目的：** 测试 首页的 某一 商品能否跳转到对应详情页面	**用例 1 ：** 点击首页的“VQA 冰葡萄酒(ADANNE)甜白甜红葡萄酒”，看能否正常跳转	成功跳转到“VQA 冰葡萄酒(ADANNE)甜白甜红葡萄酒”页面	
		用例 2 ： 点击首页的“CANA CABIN 加拿大原瓶进口冰酒”，看能否正常跳转	成功跳转到“CANA CABIN 加拿大原瓶进口冰酒”页面	
		用例 3 ： 点击首页的“露颂法国进口晚收女生晚安热红葡萄酒”，看能否正常跳转	成功跳转到“露颂法国进口晚收女生晚安热红葡萄酒”页面	

9.1.4.2.2 **商品筛选排序测试**



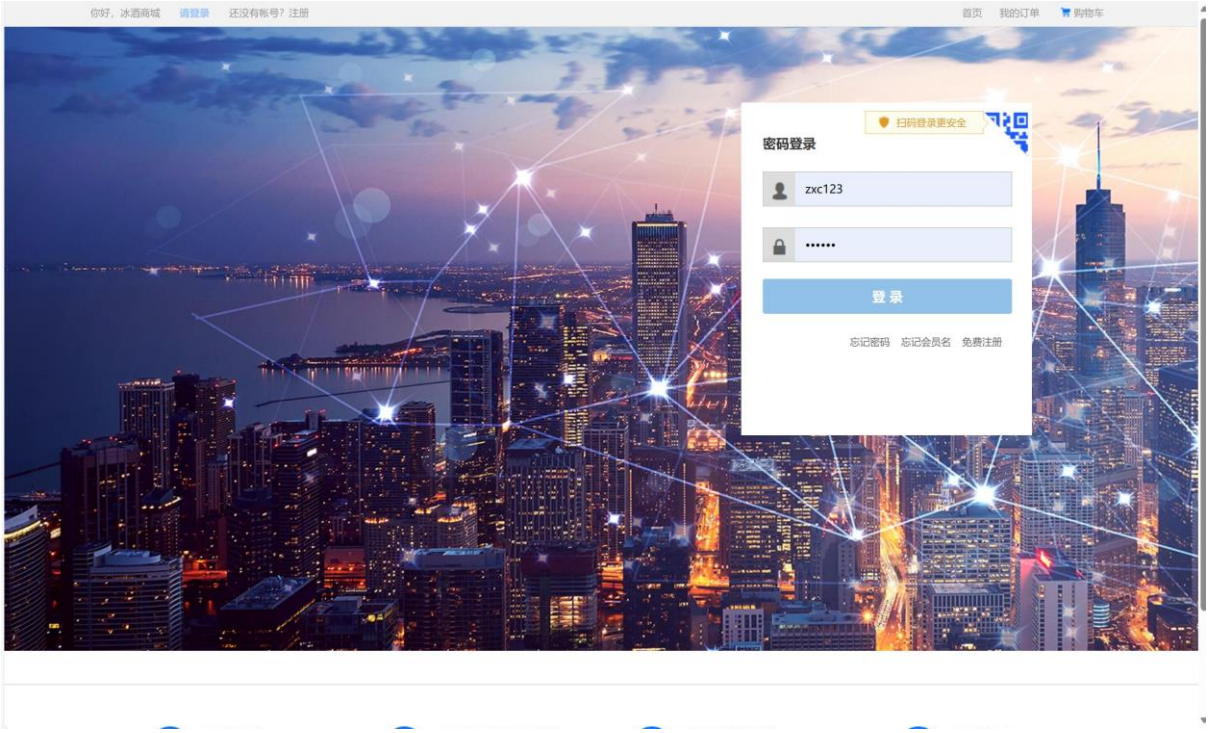
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 按照 “综合” 关键字的降序排序是否正确	**用例 1 ：** 点击“综合”，看按照“综合” 关键字的降序排序是否正确	该界面下所有商品均按照“综合“关键字降序排序	
2	**目的：** 测试 按照 “新品” 关键字的降序排序是否正确	**用例 1 ：** 点击“新品”，看按照“新品” 关键字的降序排序是否正确	该界面下所有商品均按照“新品“关键字降序排序	
3	**目的：** 测试 按照 “价格” 关键字的排序是否正确	**用例 1 ：** 点击“价格”，看按照“价格” 关键字的降序排序是否正确	该界面下所有商品均按照“价格“关键字降序排序	
		用例 2： 再次点击“价格”，看按照“价格”关键字的升序排序是否正确	该界面下所有商品均按照“价格“关键字升序排序	
4	**目的：** 测试 按照 “销量” 关键字的降序排序是否正确	**用例 1 ：** 点击“销量”，看按照“销量” 关键字的降序排序是否正确	该界面下所有商品均按照“销量“关键字降序排序	

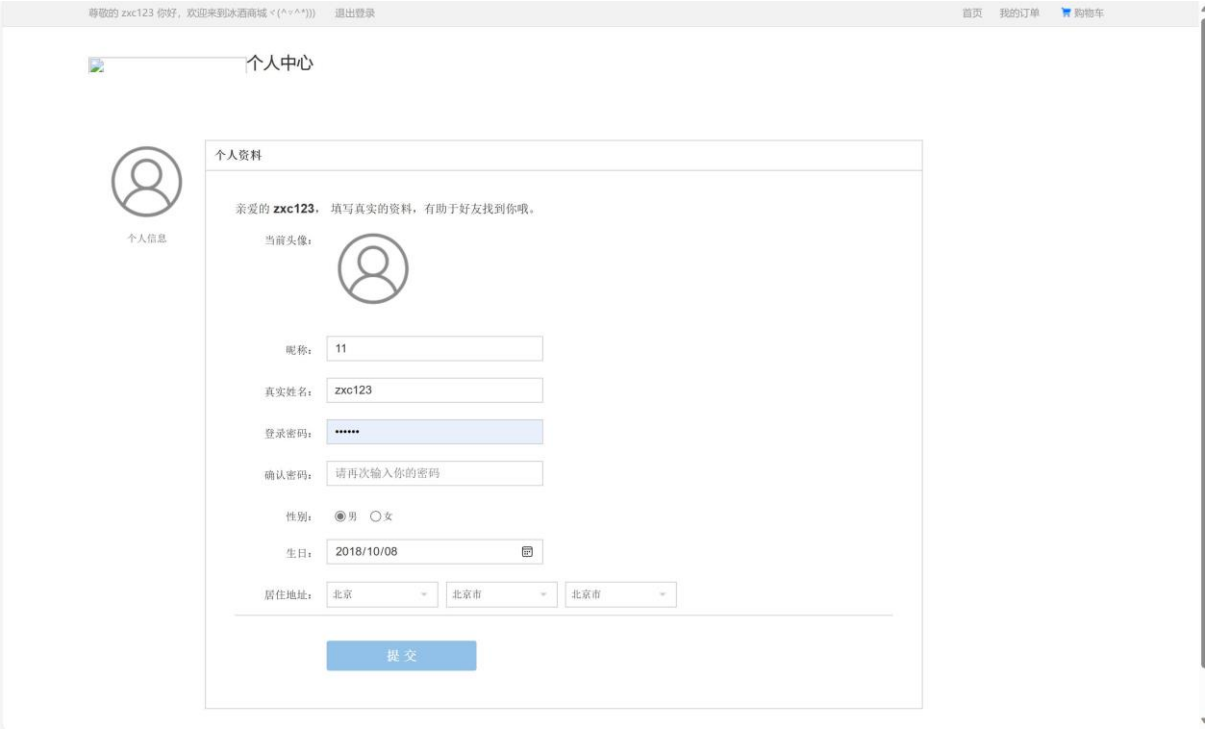
9.1.4.2.3 用户账户管理测试

注册

登录



管理

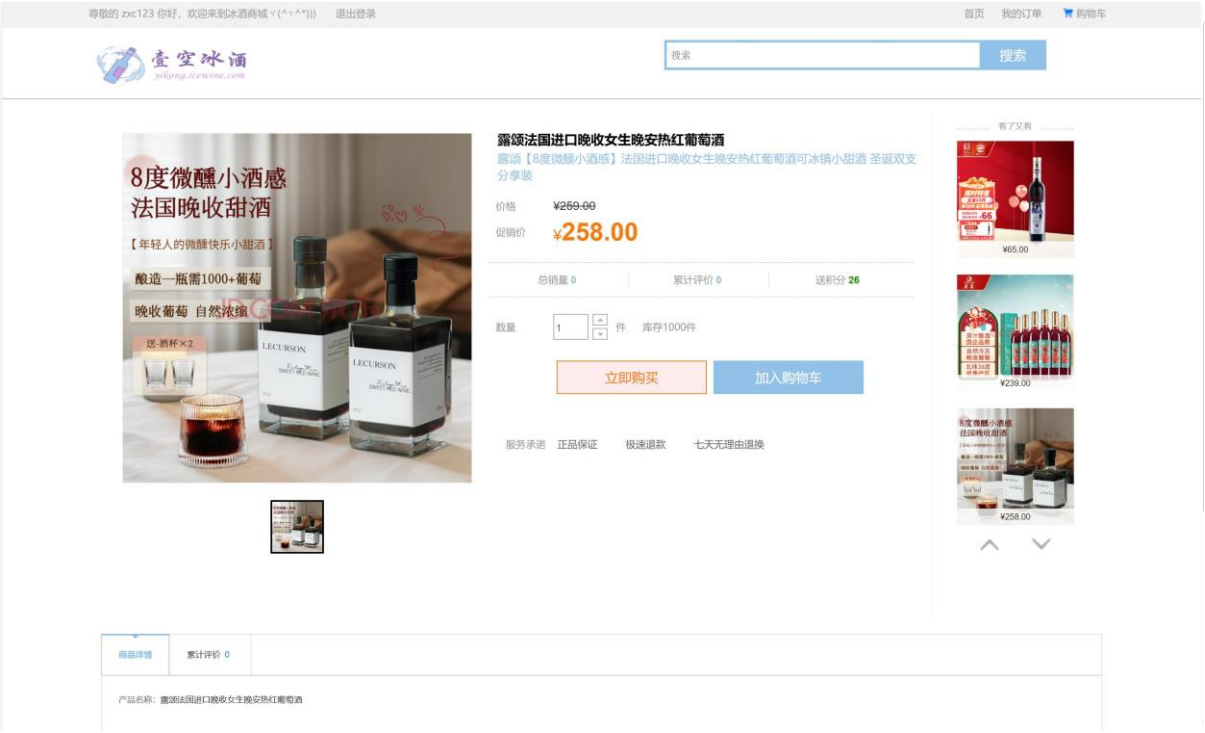


测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 ”注册“ 页面 能否正常注册，包括校验登陆密码的合理性，确认密码的检验等	**用例 1 ：** 在 ”登录密码“ 文本框输入 “root”（不合法密码），填写完其他信息后，点击 “注册”	“登陆密码”那一栏后面显示文字提醒“密码格式必须包含数字和字母”	密码合理性校验码
		用例 2 ： 在 ”登录密码“ 文本框输入 “root123”（不合法密码），填写完其他信息后，点击 “注册”	无报错提醒	
		用例 3 ： 在 ”登录密码“ 文本框输入 “root”（和登陆密码不同），填写完其他信息后，点击 “注册”	“确认密码”那一栏后面显示文字提醒“两次输入密码不相同”	密码确认合理性校验
		用例 4 ： 在 ”登录密码“ 文本框输入 “root”（和登陆密码不同），填写完其他信息后，点击 “注册”	无报错提醒	

		用例 5 ： 所有信息均合法且填写完毕后，点击“注册”	无报错提醒	测试能否正常注册
2	**目的：** 测试 ”登录“ 页面 能否正常登录	**用例 1 ：** 用户文本框输入“zxc123”，密码输入“zxc1233”(错误密码)，点击登录	提示“用户名和密码错误”	查看登录校验功能是否准确
		用例 2 ： 用户文本框输入“zxc123”，密码输入“zxc123”(正确密码)，点击登录	成功登录，跳转到商品首页	
3	**目的：** 测试 个人中心的账号信息管理功能	**用例 1 ：** 修改真实姓名为“zxc1233”，查看是否修改成功	真实姓名成功修改为“zxc123”	注意每次修改个人信息都要重新输入登陆密码

9.1.4.2.4 商品详情页测试

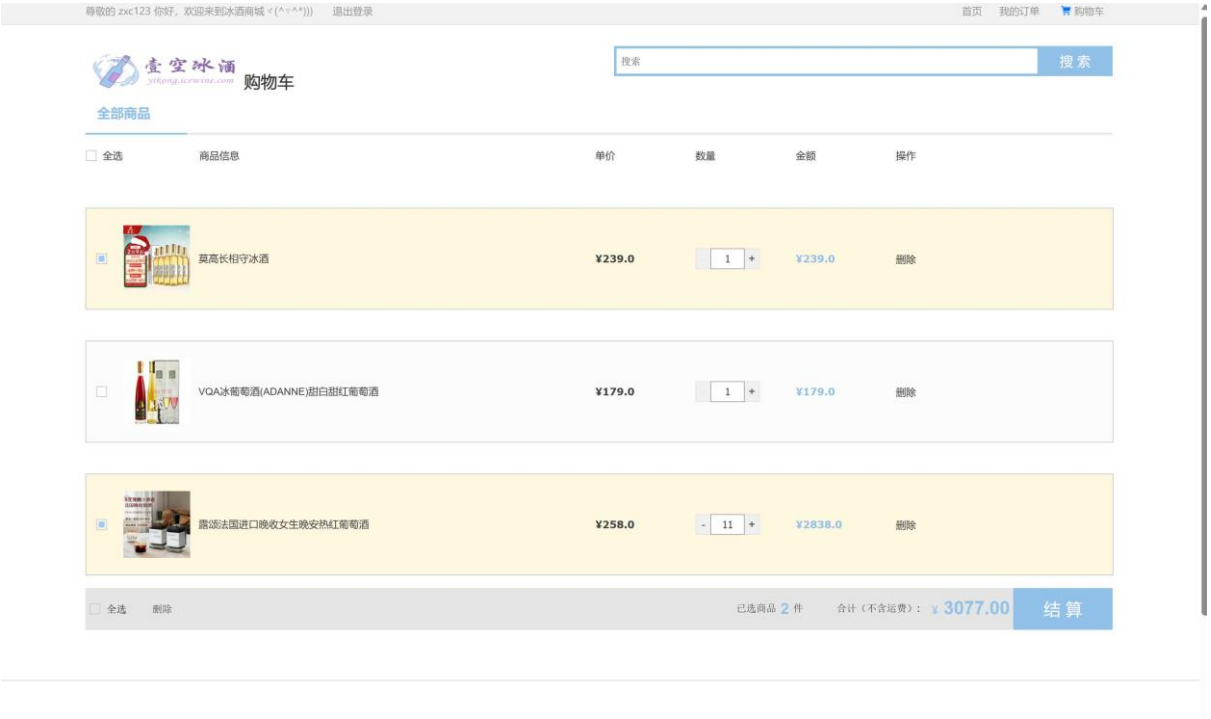


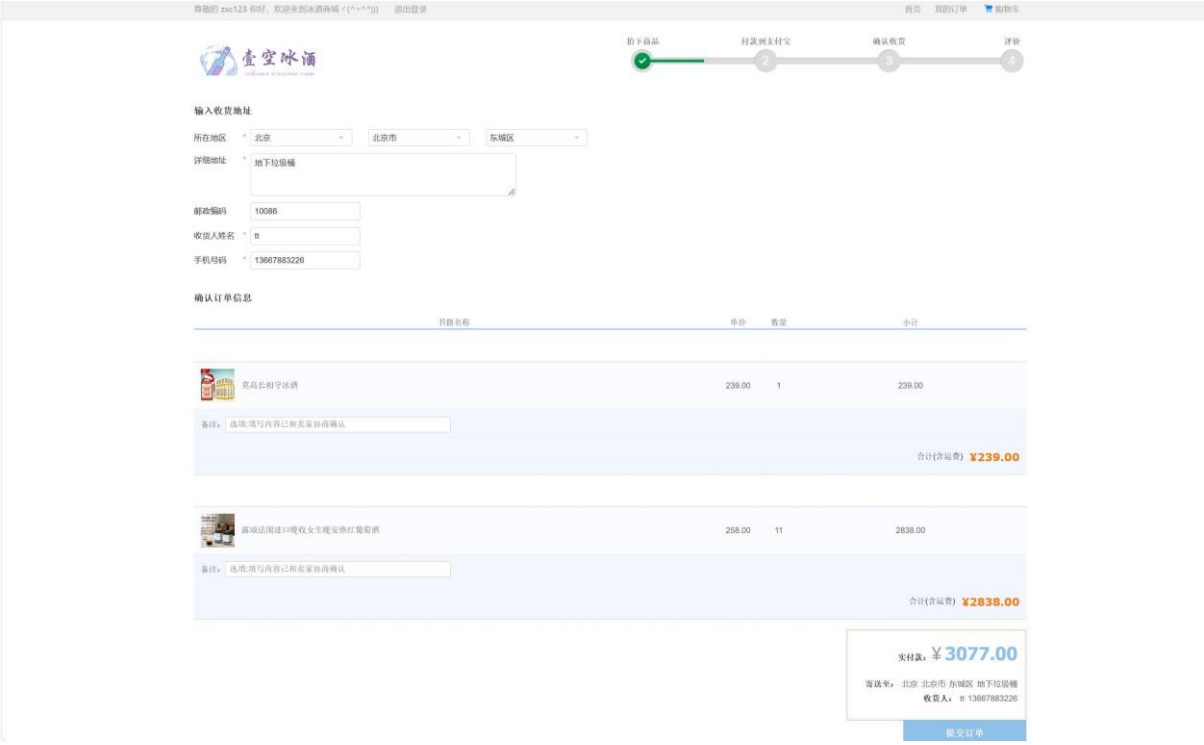
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 商品详情页 的“加	**用例 1 ：** 将商品数量设置为 10	在购物车界面查看到该商品有 10 件	首次加入

	入购物车”操作是否正确	件，然后点击“加入购物车”，去购物车界面查看是否加入成功		
		用例 2 ： 将商品数量设置为 10 件，然再次点击“加入购物车”，去购物车界面查看是否加入成功	在购物车界面查看到该商品有 20 件	再次加入
2	**目的：** 测试 商品详情页 的“立即购买”操作是否正确	**用例 1 ：** 点击“立即购买”界面，跳转到该商品的支付界面	跳转到该商品的支付界面	是否能直接购买
3	**目的：** 测试 商品详情页 的“看了又看”能否链接到其他商品页面	**用例 1：** 点击右侧 “看了又看” 按键的图片，看能否跳转到对应画面	跳转到对应的商品对应的页面	

9.1.4.2.5 **购物车测试**

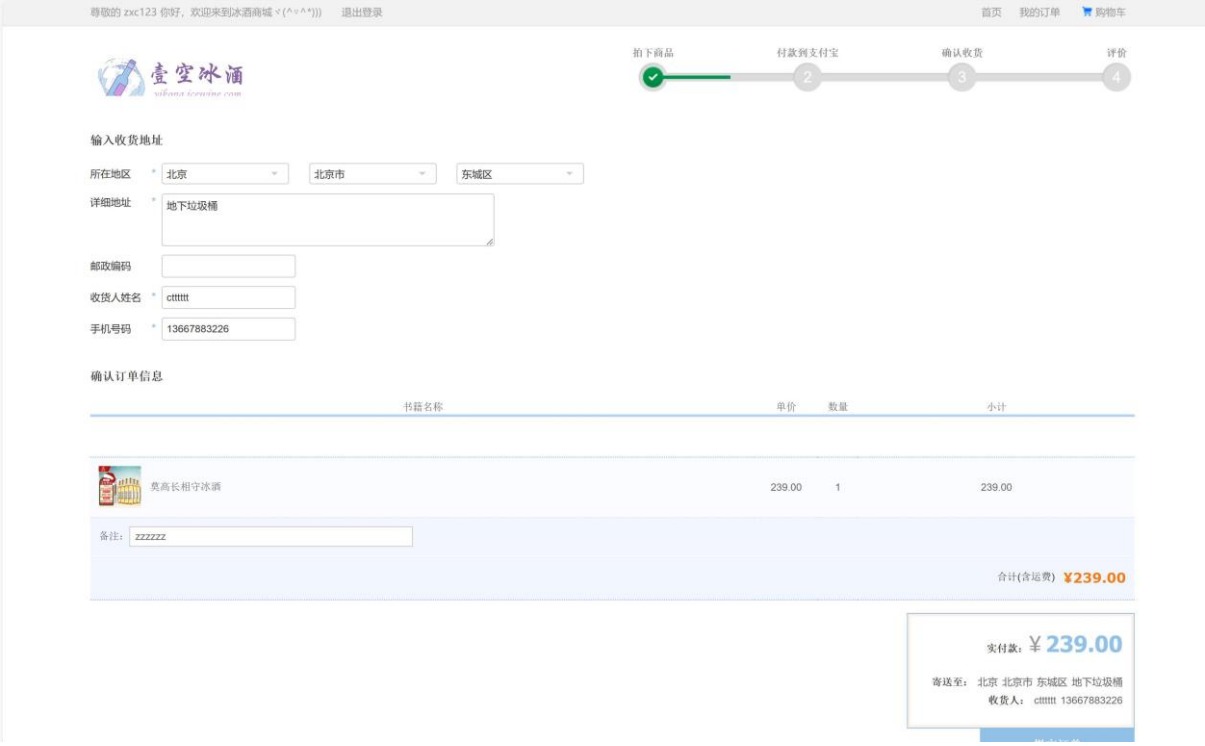




测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 购物车 的 商品增删改 等功能是否有效	**用例 1 ：** 新增商品“长白山冰酒”到购物车	购物车页面出现新增商品“长白山冰酒”	增
		用例 2 ： 将购物车中，“莫高长相守冰酒”数量设置为 1， “VQA 冰酒”数量设置为 1， “露颂冰酒”数量设置为 7	根据对应的单价计算出的总价为 2224 元	改
		用例 3 ： 将购物车中，删除“VQA 冰酒”	购物车页面中“长白山冰酒”被删除	删
2	**目的：** 测试 购物车 的 结算 能否正常跳转	**用例 1 ：** 选中“莫高”和“露颂”，点击结算	跳转到结算页面，显示选中的商品	

9.1.4.2.6 地址填写测试



测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试能否正常填写地址，如校验必填项，和生成对应消息供用户确认	**用例 1 ：** 所在地区选择“北京-北京市-东城区”，收货人姓名“ctttttt”，电话号码填写“13667883226”，点击“提交订单”	详细地址一栏被标红	必选项校验
		用例 2 ： 所在地区选择“北京-北京市-东城区”，详细地址“地下垃圾桶”，电话号码填写“13667883226”，点击“提交订单”	收货人姓名一栏被标红	
		用例 3 ： 所在地区选择“北京-北京市-东城区”，收货人姓名“ctttttt”，详细地址“地下垃圾桶”，点击“提交订单”	电话号码一栏被标红	
2	**目的：** 测试能否正常提交订单	**用例 1 ：** 点击提交订单，看能否到达支付界面	跳转到支付界面	

9.1.4.3 系统测试

9.1.4.3.1 压力测试

（一）目的

压力测试旨在评估冰酒销售管理系统在正常和峰值负载条件下的性能表现，以确保系统能够处理预期范围内的用户并发和数据量。

（二）测试场景

（1）正常负载测试：

1）模拟系统在正常使用情况下的并发用户数和交易量。
2）验证系统在正常工作负载下的性能表现。

（2）峰值负载测试：

1）模拟系统在最大并发用户数和最大交易量的情况下。
2）评估系统在负载峰值情况下的性能稳定性和响应时间。

（3）渐进增加负载测试：

1）逐步增加并发用户数和交易量，以确定系统的性能阈值和瓶颈。
2）识别系统在逐步增加负载时的性能退化点。

（三）测试环境

（1）使用性能测试工具，例如 Apache JMeter。
（2）使用真实业务场景的测试数据集。
（3）监测系统资源使用情况，包括 CPU 使用率、内存使用率、磁盘 I/O 等。

（四）测试指标

（1）响应时间：

1）记录系统对用户请求的响应时间，包括平均响应时间和最大响应时间。
2）确保在正常和峰值负载下，响应时间仍然在可接受范围内。

（2）吞吐量：

1）评估系统在单位时间内能够处理的事务数量。
2）确保系统在正常和峰值负载下的吞吐量满足业务需求。

（3）错误率：

1）评估系统在单位时间内能够处理的事务数量。
2）确保系统在正常和峰值负载下的吞吐量满足业务需求。

（4）资源利用率：

1）监测系统在不同负载下的硬件资源利用率。
2）确保系统不会因资源不足而导致性能下降。

（五）测试执行计划

（1）测试准备：

1）准备测试环境，包括性能测试工具的安装和配置。
2）加载测试数据集到测试数据库。
3）定义并配置性能测试场景。

（2）测试执行：

1）执行正常负载测试，记录响应时间、吞吐量和错误率。
2）执行峰值负载测试，评估系统在最大负载下的性能表现。
3）执行渐进增加负载测试，识别性能瓶颈和退化点。

（3）结果分析：

1）分析测试结果，确定系统的性能指标是否满足预期。
2）识别性能瓶颈，并记录可能的优化建议。

（4）报告与总结：

1）撰写性能测试报告，包括测试执行概要、结果摘要、问题识别和建议。
2）将报告分享给相关团队，包括开发团队和系统架构师。

9.1.4.3.2 系统反应速度测试

（一）目的

系统反应速度测试旨在评估冰酒销售管理系统对用户请求的实时响应性能。该测试侧重于测量系统处理单个用户请求的时间，以确保用户在系统中执行操作时能够获得及时的反馈。

（二）测试场景

（1）用户登录：

1）模拟用户登录操作，记录登录过程中系统的响应时间。
2）评估用户从输入凭证到成功登录所需的时间。

（2）商品浏览：

1）模拟用户浏览商品的操作，记录系统对商品查询的响应时间。
2）评估用户浏览商品列表和详细信息的速度。

（3）订单创建：

1）模拟用户创建订单的操作，记录系统处理订单创建请求的时间。
2）评估用户从选择商品到成功创建订单所需的时间。

（三）测试环境

（1）使用性能测试工具，例如 Apache JMeter 和 LoadRunner。
（2）使用真实业务场景的测试数据集，包括常见和边界值的数据。
（3）监测系统资源使用情况，包括 CPU 使用率、内存使用率、磁盘 I/O 等。

（四）测试指标

（1）页面加载时间：

记录系统加载每个页面所需的时间，包括但不限于登录页面、商品列表页面和订单创建页面。

（2）交互响应时间：

1）评估用户在系统中进行交互时，系统对用户输入的实时响应速度。
2）包括按钮点击、下拉菜单选择等用户操作。

（3）事务处理时间：

记录系统处理关键业务事务的时间，例如订单创建和提交。

（五）测试执行计划

（1）测试准备：

1）准备测试环境，包括性能测试工具的安装和配置。
2）加载测试数据集到测试数据库。
3）定义并配置系统反应速度测试场景。

（2）测试执行：

1）执行登录操作，记录登录页面加载时间和登录成功的响应时间。
2）执行商品浏览操作，记录商品列表页面加载时间和商品详细信息的响应时间。
3）执行订单创建操作，记录订单创建页面加载时间和订单提交的响应时间。

（3）结果分析：

1）分析测试结果，确定系统在各个场景下的响应速度是否满足用户期望。
2）识别性能瓶颈，并记录可能的优化建议。

（4）报告与总结：

1）撰写性能测试报告，包括测试执行概要、结果摘要、问题识别和建议。
2）将报告分享给相关团队，包括开发团队和系统架构师。

9.1.4.4 单元测试

9.1.4.4.1 测试说明

（一）JUnit Jupiter

在单元测试代码中，使用了 JUnit Jupiter（JUnit 5）作为 Java 测试框架进行单元测试。JUnit Jupiter 是 JUnit 5 的一部分，它提供了一组强大的注解和断言方法，用于编写和执行单元测试。

JUnit Jupiter 的优势：

（1）成熟的生态系统：JUnit Jupiter 是广泛使用的 Java 单元测试框架之一，具有广泛的生态系统和社区支持。它具有丰富的文档和资源，可以方便地获得帮助和支持。
（2）注解风格的测试编写：JUnit Jupiter 采用了基于注解的测试编写风格，使得编写测试用例更加简洁和易读。通过使用@Test 注解标记测试方法，可以明确标识出要执行的测试方法。
（3）断言丰富：JUnit Jupiter 提供了一组丰富的断言方法，例如 assertEquals()、assertTrue()、assertFalse()等，可以方便地进行结果验证和断言判断。这些断言方法使得测试代码更加清晰和易于理解。
（4）扩展性和灵活性：JUnit Jupiter 提供了许多扩展点和自定义选项，使得可以根据需要进行自定义扩展和配置。例如，可以编写自定义的扩展来实现额外的测试功能或集成其他工具。

（二）Arrange-Act-Assert (AAA)

Arrange-Act-Assert (AAA) 是一种常用的单元测试方法论，它有助于组织和结构化测试代码，使测试更加清晰和易于理解。

AAA 方法的步骤：

（1）准备阶段 (Arrange)：

在这个阶段，我们准备测试所需的数据和对象，设置测试的前置条件。这包括创建对象实例、设置输入数据、模拟环境等。准备阶段的目标是为测试方法创建一个独立的、可重复的环境。

（2）执行阶段 (Act)：

在这个阶段，我们执行要测试的代码或调用要测试的方法。这是对待测试代码进行实际调用的一步。执行阶段的目标是触发代码的执行，以便在下一阶段进行验证。

（3）断言阶段 (Assert)：

在这个阶段，我们对执行阶段得到的结果进行验证和断言。我们使用断言方法来比较实际输出与预期输出是否一致。如果断言失败，则测试失败。断言阶段的目标是验证代码的行为是否符合预期。

AAA 方法的优势：

（1）可读性强：

AAA 方法使测试代码的组织结构更清晰，每个阶段的目标清晰可见。

（2）可维护性高：

通过明确定义准备、执行和断言的阶段，使得测试代码易于维护和修改。

（3）可重复性好：

由于准备阶段的独立性，测试可以在任何环境中重复运行，结果应该保持一致。

9.1.4.4.2 测试计划

（一）测试文件 src/main/java/com/uzykj/mall/util/DateUtil.java

（1）测试方法 1 ：testGetOrderIdByTime()
测试目标：验证`DateUtil.getOrderIdByTime()`方法是否返回非空的订单 ID。

（2）测试方法 2 ：testGetCurrTime()
测试目标：验证`DateUtil.getCurrTime()`方法是否返回非空的当前时间字符串，并且该时间字符串的长度为 14，并且以"2023"开头。

（3）测试代码：

```java
import com.uzykj.mall.util.*;
import org.junit.jupiter.api.Test;

import java.util.Date;

import static org.junit.jupiter.api.Assertions.*;

no usages
class DateUtilTest {
    no usages
    @Test
    void testGetOrderIdByTime() {
        // Arrange
        String orderId = DateUtil.getOrderIdByTime();

        // Assert
        assertNotNull(orderId);
        assertEquals(17, orderId.length());
    }

    no usages
    @Test
    void testGetCurrTime() {
        // Arrange
        String currTime = DateUtil.getCurrTime();

        // Assert
        assertNotNull(currTime);
        assertEquals(14, currTime.length());
        assertTrue(currTime.startsWith("2023"));
    }
}
```

（二）测试文件 src/main/java/com/uzykj/mall/util/FileUtil.java

（1）测试说明：

1）在此测试中，我们使用@Mock 注解创建了一个模拟的 HttpSession 对象和一个模拟的 ServletContext 对象。

2）我们使用模拟对象的目的是为了隔离被测试方法中的外部依赖，例如在 FileUtil.judeFileExists()方法中可能会使用 HttpSession 或 ServletContext 对象。

3）通过模拟这些外部依赖，我们可以控制它们的行为，使测试更加独立和可控。

（2）测试方法 1 ：testJudeFileExists()

测试目标：验证 FileUtil.judeFileExists(File file)方法是否正确判断给定文件是否存在。

（3）测试方法 2 ：testJudeDirExists()

测试目标：验证 FileUtil.judeDirExists(File file)方法是否正确判断给定目录是否存在。

（4）测试方法 3 ：testCreateDirectory()

测试目标：验证 FileUtil.createDirectory(String path)方法是否能成功创建目录。

（5）测试方法 4 ：testGenerFileUrl()

测试目标：验证 FileUtil.generateFileUrl(String fileName)方法是否能正确生成文件的 URL 路径。

（6）测试方法 5 ：testGenerNewFileName()

测试目标：验证 FileUtil.generateNewFileName(String originalFileName)方法是否能正确生成新的文件名。

（7）测试方法 6：testGenerLocalFilePath()

测试目标：验证 FileUtil.generateLocalFilePath(String directory, String fileName)方法是否能正确生成本地文件路径。

（8）测试代码：

```java
package com.uzykj.mall.util;

import org.junit.jupiter.api.Test;
import org.mockito.Mock;
import org.mockito.Mockito;

import javax.servlet.ServletContext;
import javax.servlet.http.HttpSession;
import java.io.File;

import static org.junit.jupiter.api.Assertions.*;

no usages
class FileUtilTest {

    2 usages
    @Mock
    private HttpSession session;

    2 usages
    @Mock
    private ServletContext context;
```

```java
no usages
@Test
void testJudeFileExists() {
    // Arrange
    File file = new File("test.txt");
    // Act
    boolean result = FileUtil.judeFileExists(file);
    // Assert
    assertFalse(result);
}

no usages
@Test
void testJudeDirExists() {
    // Arrange
    File file = new File("test");
    // Act
    boolean result = FileUtil.judeDirExists(file);
    // Assert
    assertFalse(result);
}

no usages
@Test
void testCreateDirectory() {
    // Arrange
    String dirName = "test";
    // Act
    boolean result = FileUtil.createDirectory(dirName);
    // Assert
    assertTrue(result);
}

no usages
@Test
void testGenerFileUrl() {
    // Arrange
    String fileSubfix = "images";
    String imageType = "png";
    String name = "logo";
    // Act
    String result = FileUtil.generFileUrl(fileSubfix, imageType, name);
    // Assert
    assertEquals("images/png/logo", result);
}
```

```java
    no usages
    @Test
    void testGenerNewFileName() {
        // Arrange
        String originalFileName = "test.txt";
        // Act
        String result = FileUtil.generNewFileName(originalFileName);
        // Assert
        assertNotNull(result);
        assertTrue(result.endsWith(".txt"));
    }

    no usages
    @Test
    void testGenerLocalFilePath() {
        // Arrange
        String path = "images";
        String imageType = "png";
        Mockito.when(session.getServletContext()).thenReturn(context);
        Mockito.when(context.getRealPath("/")).thenReturn("/home/user/");
        // Act
        String result = FileUtil.generLocalFilePath(session, path, imageType);
        // Assert
        assertEquals("/home/user/images/png", result);
    }
}
```

（三）测试文件 src/main/java/com/uzykj/mall/util/Md5Util.java

（1）测试方法 1 ：testByteArrayToHexString()

测试目标：验证 Md5Util.byteArrayToHexString(byte[] bytes)方法是否能正确将字节数组转换为十六进制字符串。

（2）测试方法 2 ：testByteToHexString()

测试目标：验证 Md5Util.byteToHexString(byte b)方法是否能正确将字节转换为十六进制字符串。

（3）测试方法 3 ：testMD5Encode()

测试目标：验证 Md5Util.MD5Encode(String origin, String charsetname)方法是否能正确对字符串进行 MD5 编码。

（4）测试代码：

```java
package com.uzykj.mall.util;

import org.junit.jupiter.api.Test;

import java.security.MessageDigest;

import static org.junit.jupiter.api.Assertions.*;

no usages
class Md5UtilTest {

    no usages
    @Test
    void testByteArrayToHexString() {
        // Arrange
        byte[] bytes = {0, 1, 2, 15, 16, 255};
        // Act
        String result = Md5Util.byteArrayToHexString(bytes);
        // Assert
        assertEquals("0001020f10ff", result);
    }

    no usages
    @Test
    void testByteToHexString() {
        // Arrange
        byte b = -1;
        // Act
        String result = Md5Util.byteToHexString(b);
        // Assert
        assertEquals("ff", result);
    }

    no usages
    @Test
    void testMD5Encode() {
        // Arrange
        String origin = "Hello World";
        String charsetname = "UTF-8";
        // Act
        String result = Md5Util.MD5Encode(origin, charsetname);
        // Assert
        assertEquals("b10a8db164e0754105b7a99be72e3fe5", result);
    }
}
```

（四）src/main/java/com/uzykj/mall/util/PageUtil.java

（1）测试方法 1 ：testIsHasPrev()
测试目标：验证 PageUtil.isHasPrev()方法是否正确判断当前页是否有前一页。

（2）测试方法 2 ：testIsHasNext()
测试目标：验证 PageUtil.isHasNext()方法是否正确判断当前页是否有后一页。

（3）测试方法 3 ：testGetTotalPage()
测试目标：验证 PageUtil.getTotalPage()方法是否能正确计算总页数。

（4）测试方法 4 ：testGetPageStart()

测试目标：验证 PageUtil.getPageStart()方法是否能正确计算当前页的起始记录索引。

（5）测试代码：

```java
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

no usages
class PageUtilTest {
    no usages
    @Test
    void testIsHasPrev() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        // Act
        boolean result = pageUtil.isHasPrev();
        // Assert
        assertFalse(result);
    }
    no usages
    @Test
    void testIsHasNext() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        pageUtil.setTotal(100);
        // Act
        boolean result = pageUtil.isHasNext();
        // Assert
        assertTrue(result);
    }
    no usages
    @Test
    void testGetTotalPage() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        pageUtil.setTotal(95);
        // Act
        int result = pageUtil.getTotalPage();
        // Assert
        assertEquals(10, result);
    }
    no usages
    @Test
    void testGetPageStart() {
        // Arrange
        PageUtil pageUtil = new PageUtil(1, 10);
        // Act
        int result = pageUtil.getPageStart();
        // Assert
        assertEquals(10, result);
    }
```

```java
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

no usages
class PageUtilTest {
    no usages
    @Test
    void testIsHasPrev() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        // Act
        boolean result = pageUtil.isHasPrev();
        // Assert
        assertFalse(result);
    }
    no usages
    @Test
    void testIsHasNext() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        pageUtil.setTotal(100);
        // Act
        boolean result = pageUtil.isHasNext();
        // Assert
        assertTrue(result);
    }
    no usages
    @Test
    void testGetTotalPage() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        pageUtil.setTotal(95);
        // Act
        int result = pageUtil.getTotalPage();
        // Assert
        assertEquals(10, result);
    }
    no usages
    @Test
    void testGetPageStart() {
        // Arrange
        PageUtil pageUtil = new PageUtil(1, 10);
        // Act
        int result = pageUtil.getPageStart();
        // Assert
        assertEquals(10, result);
    }
```

9.1.4.5 集成测试

9.1.4.5.1 测试说明

（一）集成测试简介

集成测试是软件开发过程中的一种测试策略，用于验证多个组件或模块在集成在一起后的正确性和功能。它旨在检查系统的各个组件之间的交互和协作是否正常，以确保整个系统在集成环境中正常运行。

（二）测试目的

（1）验证组件之间的接口和通信：集成测试通过测试组件之间的接口和通信路径，确保不同组件之间的数据传递和交互正常工作。这有助于发现和解决由于接口不匹配、数据丢失或通信问题而导致的错误。

（2）检测功能完整性：集成测试确保系统的各个组件正确地协同工作，以实现系统的预期功能。它可以验证各个组件在集成环境中的行为是否符合预期，并捕获可能存在的功能缺陷。

（3）发现集成问题：集成测试有助于发现仅在组件集成后才会出现的问题。这些问题可能包括数据冲突、资源竞争、性能问题、依赖关系错误等。

（4）提高系统的稳定性和可靠性：通过在集成环境中进行全面的测试，集成测试有助于提高系统的稳定性和可靠性。它可以发现和解决在组件集成时可能出现的错误和缺陷，减少在生产环境中遇到问题的风险。

9.1.4.5.2 测试计划

（一）框架

Spring Boot + Mybatis Plus + MySQL 8.0 + JQuery + JSP

（二）测试场景

（1）Spring Boot 的启动
（2）Mybatis Plus 的启动
（3）数据库的连接
（4）数据库的修改和查询

9.2 系统测试报告

9.2.1 范围

9.2.1.1 系统概述

冰酒订购系统旨在简化冰酒供应商接受和履行冰酒订单的流程。它旨在取代将订单记录在卡片上并存储在数据库中的手动方法。有了这个系统，顾客可以通过电话或直接访问商店下单。收到订单后，冰酒供应商输入客户的详细信息、冰酒要求，并使用系统计算成本。这确保了准确的定价和高效的订单管理。该系统中描述的功能旨在在初始版本 1.0 中完全实现。然而，该系统被设计为适应未来的增强功能和后续版本中可能需要的附加功能。这种灵活性确保了冰酒订购系统能够适应不断变化的业务需求和客户要求。

9.2.1.2 文档描述

此文档记录了依据《冰酒销售管理系统测试计划》进行的系统测试结果。

9.2.2 参考

本文档参考如下文档进行编写：

（一） 《冰酒销售管理系统需求分析报告》
（二） 《冰酒销售管理系统系统原型报告》
（三） 《冰酒销售管理系统系统测试计划》
（四） 《计算机软件文档编制规范》

9.2.3 测试结果概述

系统测试的结果显示，冰酒销售管理系统在大多数情况下表现良好，符合预期的功能和性能要求。以下是对测试结果的概述：

（一）基本功能测试

系统的基本功能，包括用户管理、产品管理、销售订单、库存管理和报表生成等方面，经过测试都能够正常工作。用户能够顺利注册、登录，并执行各种用户操作。产品信息的管理和订单的处理也在预期范围内。

（二） 用户权限验证

用户权限验证功能有效，系统能够正确地限制用户对敏感信息和操作的访问。不同用户角

色的权限设置得以有效执行，确保系统的安全性和隐私保护。

（三） 报表生成

报表生成功能正常，生成的报表准确地反映了销售和库存的状态。用户能够方便地获取系统关键数据，有助于业务决策和分析。

（四） 缺陷报告

在测试过程中发现的缺陷主要集中在订单状态更新和跨浏览器兼容性方面。这些缺陷虽然存在，但并不影响系统的整体稳定性和基本功能的使用。

（五） 建议和改进

为了进一步提升系统的质量和用户体验，建议进行以下改进：

（1）订单状态更新优化： 优化订单状态更新的流程，以确保在所有情况下都能够及时和准确地更新订单状态。

（2）跨浏览器兼容性测试： 进行更全面的跨浏览器兼容性测试，确保用户界面在不同浏览器上的一致性，提高用户体验。

（六） 总体评估

系统测试的结果整体上是积极的，系统在功能和性能方面符合预期。通过及时解决发现的缺陷和不断进行改进，系统将更好地满足用户需求，保持良好的稳定性和可用性。

9.2.4 详细测试结果

在系统测试阶段，我们对冰酒销售管理系统的各个方面进行了详细的测试，涵盖了用户管理、产品管理、销售订单、库存管理和报表生成等核心功能。以下是对每个方面的详细测试结果的简要介绍：

（一） 用户管理

用户管理模块经过测试，包括用户注册、登录、密码修改等功能。测试覆盖了不同用户角色的权限验证，确保系统能够正确地限制用户对敏感信息和操作的访问。

（二） 产品管理

产品管理模块测试主要集中在添加、编辑和删除产品信息的功能上。此外，我们验证了产品分类和标签的管理，确保系统在产品信息方面的准确性和完整性。

（三） 销售订单

销售订单模块经过测试，包括订单的创建、编辑和删除。我们特别关注了订单状态的更新，以确保系统能够及时和正确地反映销售订单的实际状态。

（四） 库存管理

库存管理模块测试覆盖了库存数量的准确性，以及入库和出库操作。我们验证了系统对库存变化的追踪和管理，确保库存数据的一致性。

（五） 报表生成

报表生成模块的测试主要包括销售报表和库存报表的生成。我们验证了报表的准确性和及时性，确保用户能够方便地获取系统关键数据进行分析和决策。

（六） 缺陷报告

在测试过程中，我们发现了一些缺陷，主要涉及订单状态更新的延迟和跨浏览器兼容性。这些缺陷已被详细记录，并将在后续版本中进行修复和改进。

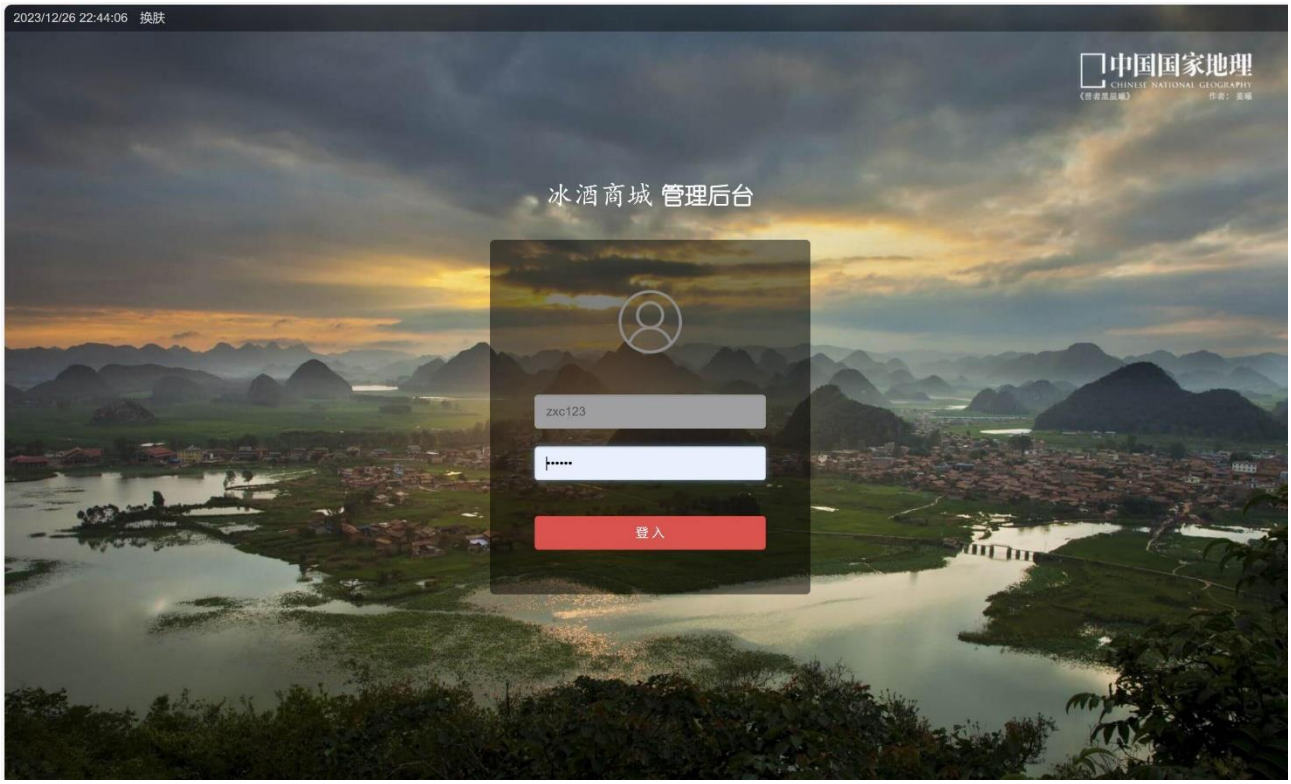
（七） 建议和改进

基于测试结果，我们提出了改进建议，包括优化订单状态更新流程和进行跨浏览器兼容性测试。这些建议旨在提升系统的质量和用户体验。

详细测试结果的分析将为开发团队提供有力的反馈，以便他们能够更好地理解系统的强项和改进的空间，进而提高系统的整体质量。

9.2.4.1 后台功能测试

9.2.4.1.1 系统进入测试



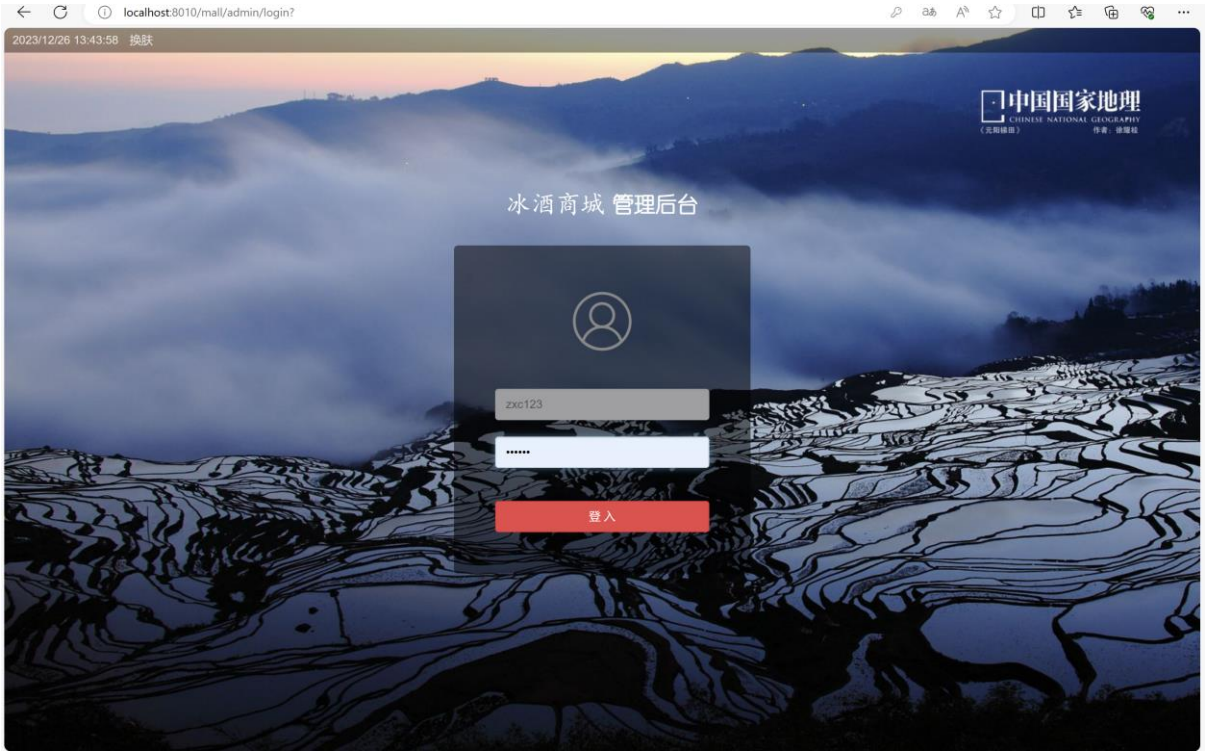
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	目的：检测系统登录操作成功	用例 1 （正确）： 用户名：zxc123 密码：zxc123	登录进入后台	
2	目的：检测系统登录操作不成功及其反馈	用例 2 （不正确）： 用户名：zxc 密码：1233	提示出错信息	

测试过程

登陆操作成功

登陆操作不成功



测试结果

序号	测试目的 & 先决条件	测试用例输入	测试结果	备注

1	目的：检测系统登录操作成功	用例 1 （正确）： 用户名：zxc123 密码：zxc123	成功登录，进入后台	
2	目的：检测系统登录操作不成功及其反馈	用例 2 （不正确）： 用户名：zxc 密码：1233	提示出错信息	

9.2.4.1.2 管理首页测试

测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	目的：检测一段时间内对上架产品数量、注册用户数量、成交订单数量、商城订单交易额的统计是否正确	用例 1 ： 统计时间区间选择 2023/12/19 到 2023/12/25	上架商品数量：21 注册用户数量：2 成交订单数量：0 商城订单交易额：0	
		用例 2 ： 上架一件商品后，统计时间区间选择 2023/12/25 到 2023/12/25	上架商品数量：22 注册用户数量：2 成交订单数量：0 商城订单交易额：0	
		用例 3 ： 注册一个新用户后，统计时间区间选择 2023/12/25 到 2023/12/25	上架商品数量：22 注册用户数量：3 成交订单数量：0 商城订单交易额：0	

测试过程

统计时间区间选择 2023/12/19 到 2023/12/25

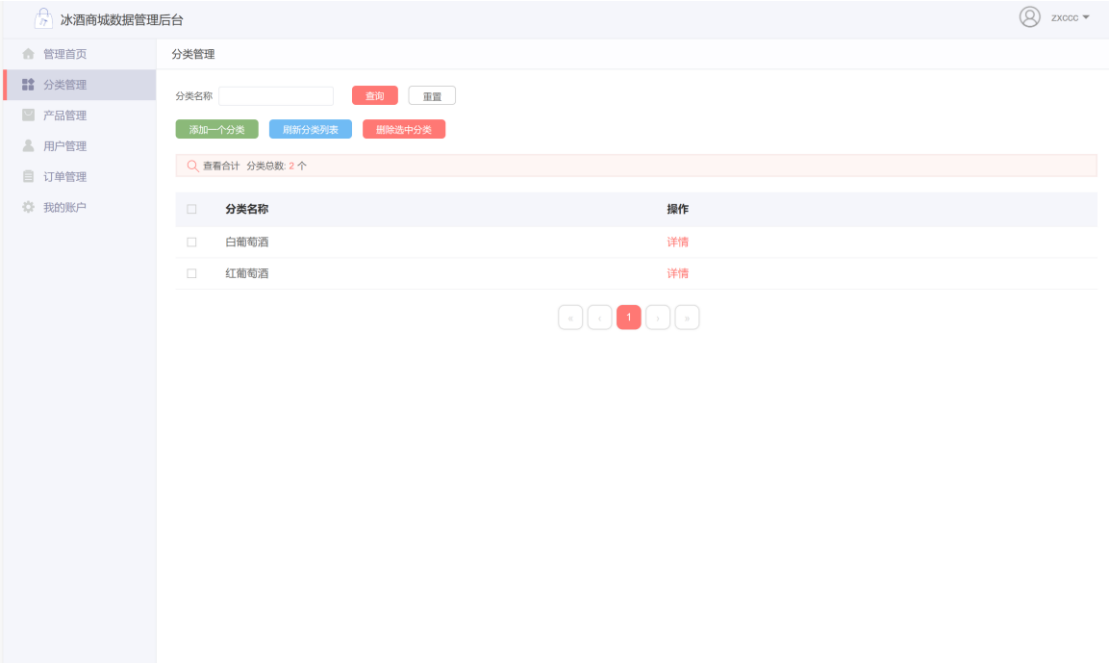
上架一件商品后，统计时间区间选择 2023/12/25 到 2023/12/25

注册一个新用户后，统计时间区间选择 2023/12/25 到 2023/12/25

测试结果

序号	测试目的 & 先决条件	测试用例输入	测试结果	备注
1	目的：检测一段时间内对上架产品数量、注册用户数量、成交订单数量、商城订单交易额的统计是否正确	用例 1： 统计时间区间选择 2023/12/19 到 2023/12/25	上架商品数量：21 注册用户数量：2 成交订单数量：0 商城订单交易额：0	
		用例 2： 上架一件商品后，统计时间区间选择 2023/12/25 到 2023/12/25	上架商品数量：22 注册用户数量：2 成交订单数量：0 商城订单交易额：0	
		用例 3： 注册一个新用户后，统计时间区间选择 2023/12/25 到 2023/12/25	上架商品数量：22 注册用户数量：3 成交订单数量：0 商城订单交易额：0	

9.2.4.1.3 分类管理测试



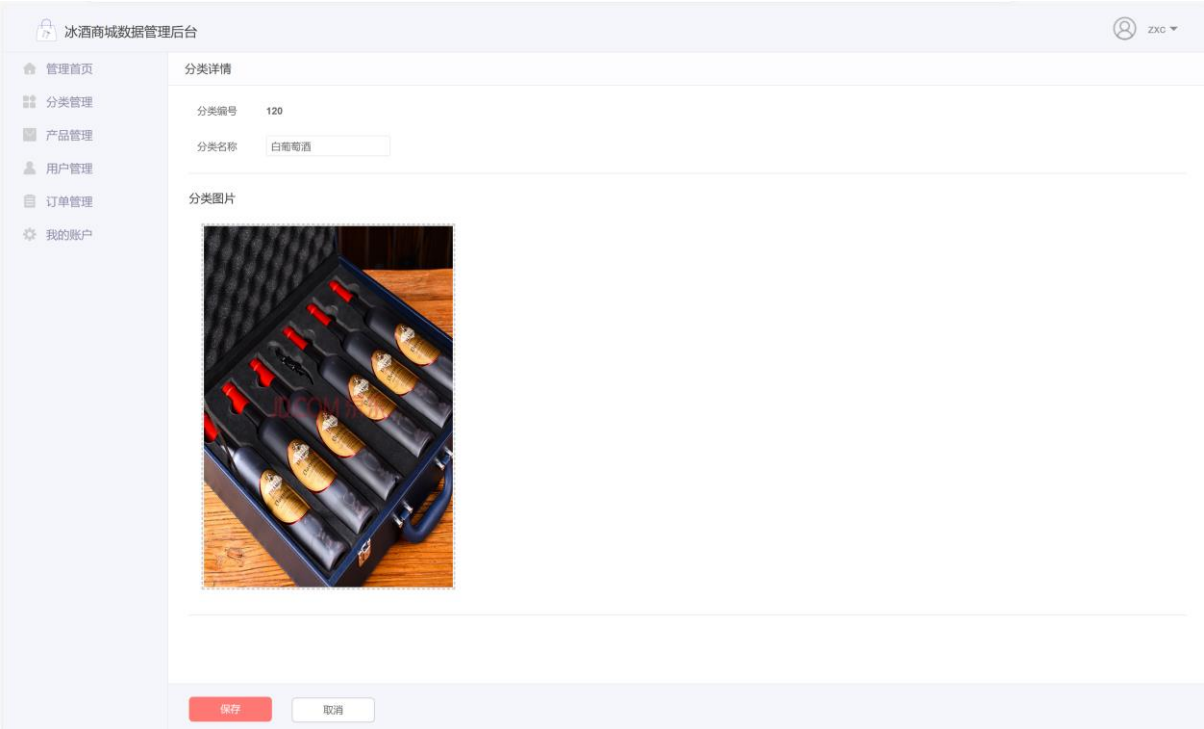
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 检测分类管理页面的“详情 ”操作是否正确	**用例 1 ：** 点击分类“白葡萄酒”右侧的“详情”	显示分类“白葡萄酒”的详情	
2	**目的：** 检测分类管理页面的“查询 ”操作是否正确	**用例 2 ：** 在“分类名称”的文本框中输入“葡萄”后，点击右侧的“查询”按钮	列表显示名称有关键词“葡萄”的分类	
		用例 3 ： 在“分类名称”的文本框中输入空值后，点击右侧的“查询”按钮	列表显示所有分类	
3	**目的：** 测试分类管理页面的“重置 ”操作是否正确 先决条件：先前进行过查询	**用例 4 ：** 在“分类名称”的文本框中输入“葡萄”后，点击右侧的“查询”按钮，再点击“重置”按钮	点击“查询”按钮后，列表显示名称有关键词“葡萄”的分类；点击“重置”按钮后，“分类名称”的文本框清空，列表显示所有分类	
4	**目的：** 测试分类管理页面的“添加一个分类 ”操作是否正确	**用例 5 ：** 点击“添加一个分类”，在“添加分类”页面的“分类名称”中输入“测试分类”，并上传“分类图片”后，点击下方的“确定”按钮	成功创建“测试分类”，列表中显示该分类，在详情中可看到该分类的图片为上传的图片	
		用例 6 ：	显示提示信息：“分类名称”	

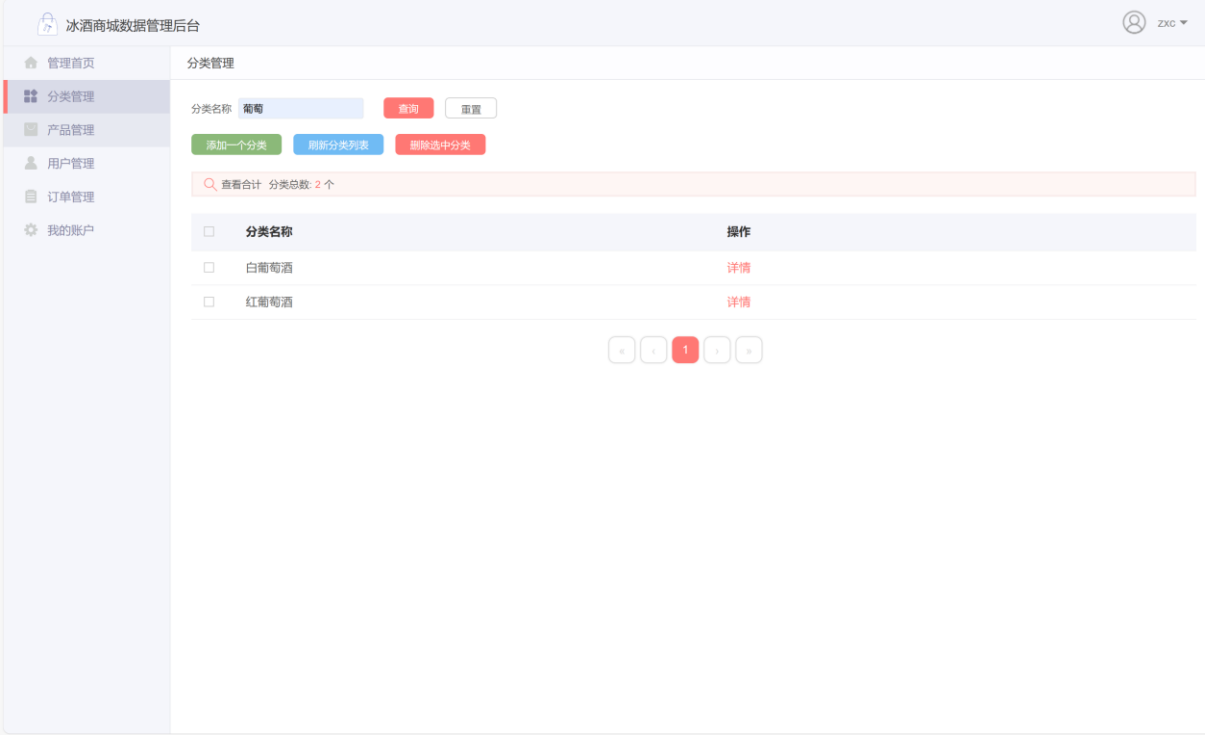
		点击“添加一个分类”，在“添加分类”页面的“分类名称”中输入空值，并上传“分类图片”后，点击下方的“确定”按钮	不能为空	
5	**目的：** 测试分类管理页面的“刷新分类列表 ”操作是否正确	**用例 7 ：** 用命令行添加分类“测试分类”后，点击“刷新分类列表”按钮	列表刷新后显示用命令行添加的分类	
6	**目的：** 测试分类管理页面的“删除选中分类”操作是否正确	**用例 8 ：** 在列表中选择分类“测试分类”后，点击“删除选中分类”按钮，点击“刷新分类列表”按钮	列表刷新后不再显示分类“测试分类”	

测试过程

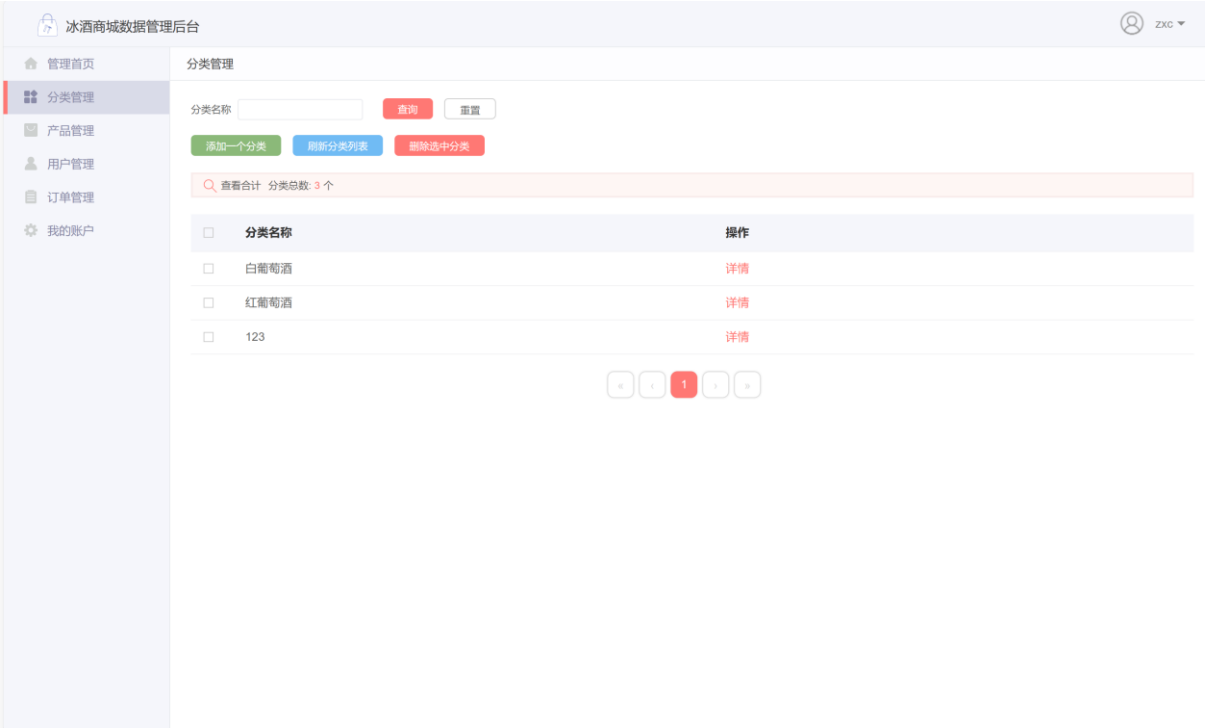
“白葡萄酒”分类详情：



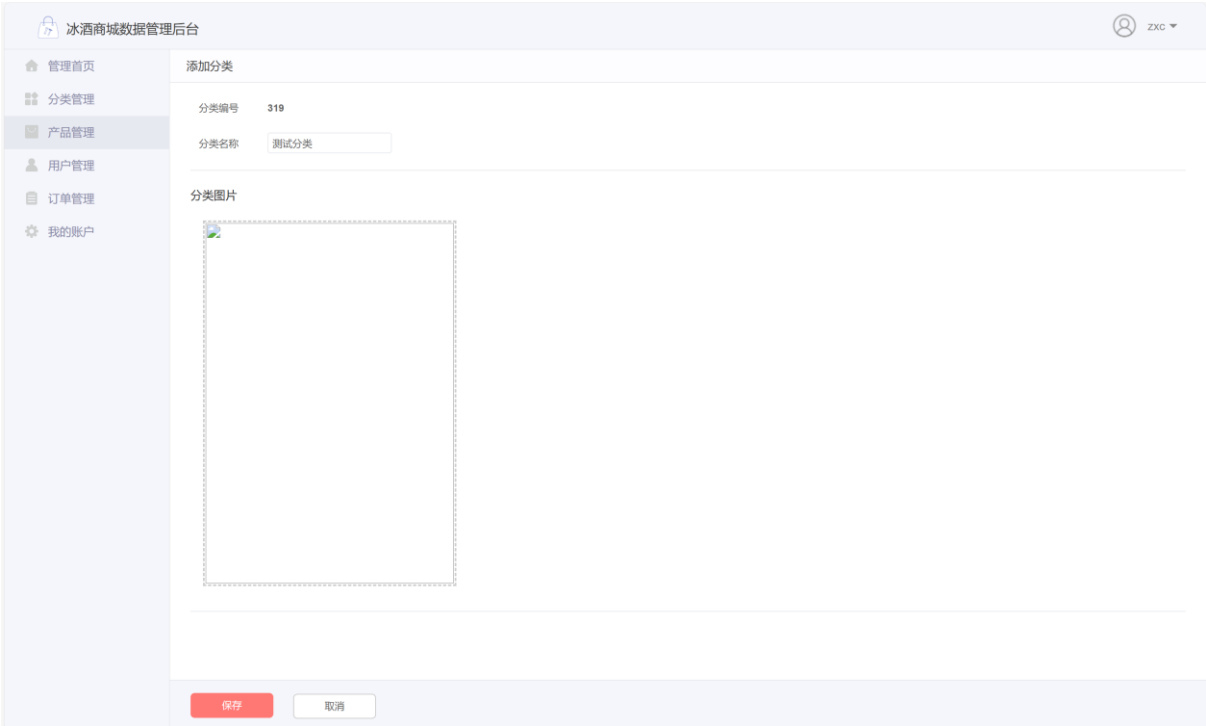
查询“葡萄”



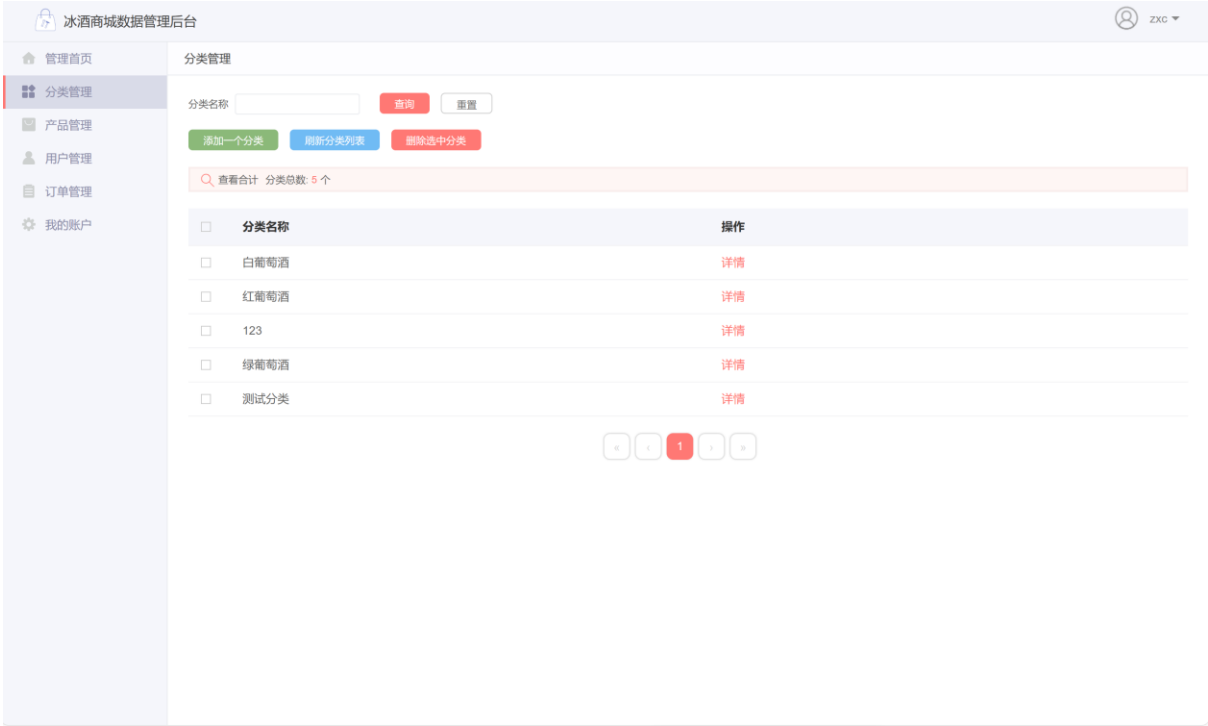
查询空值和查询后重置：



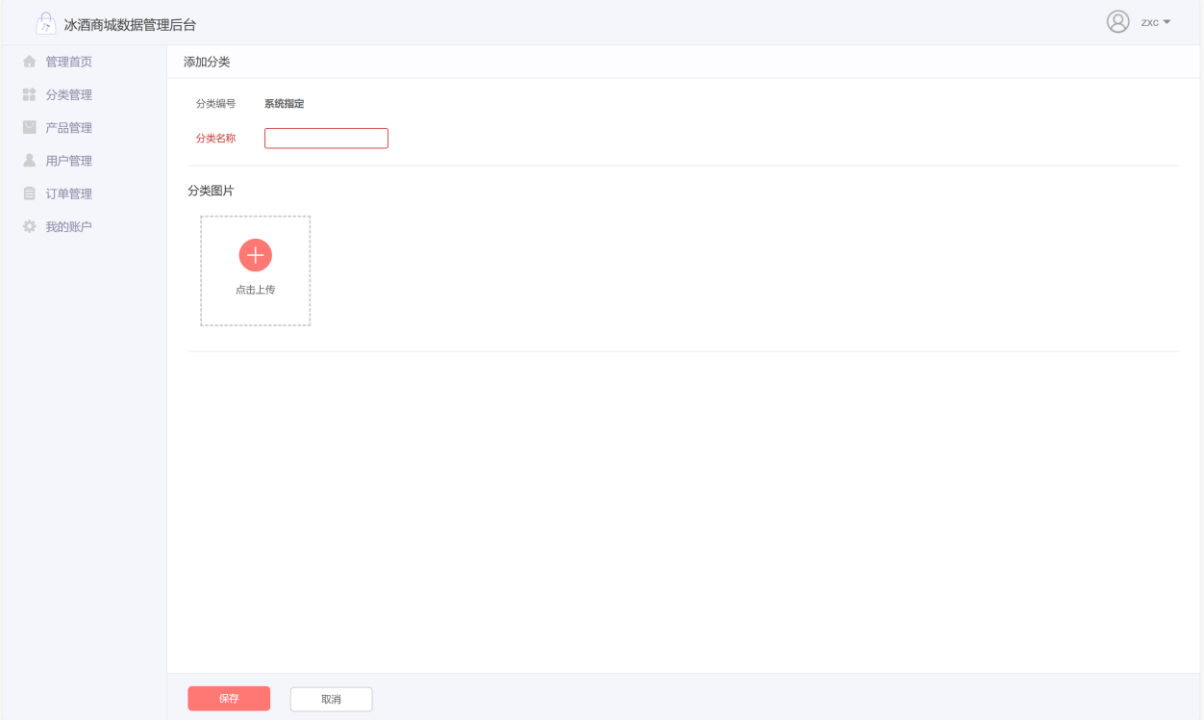
新增“测试分类”



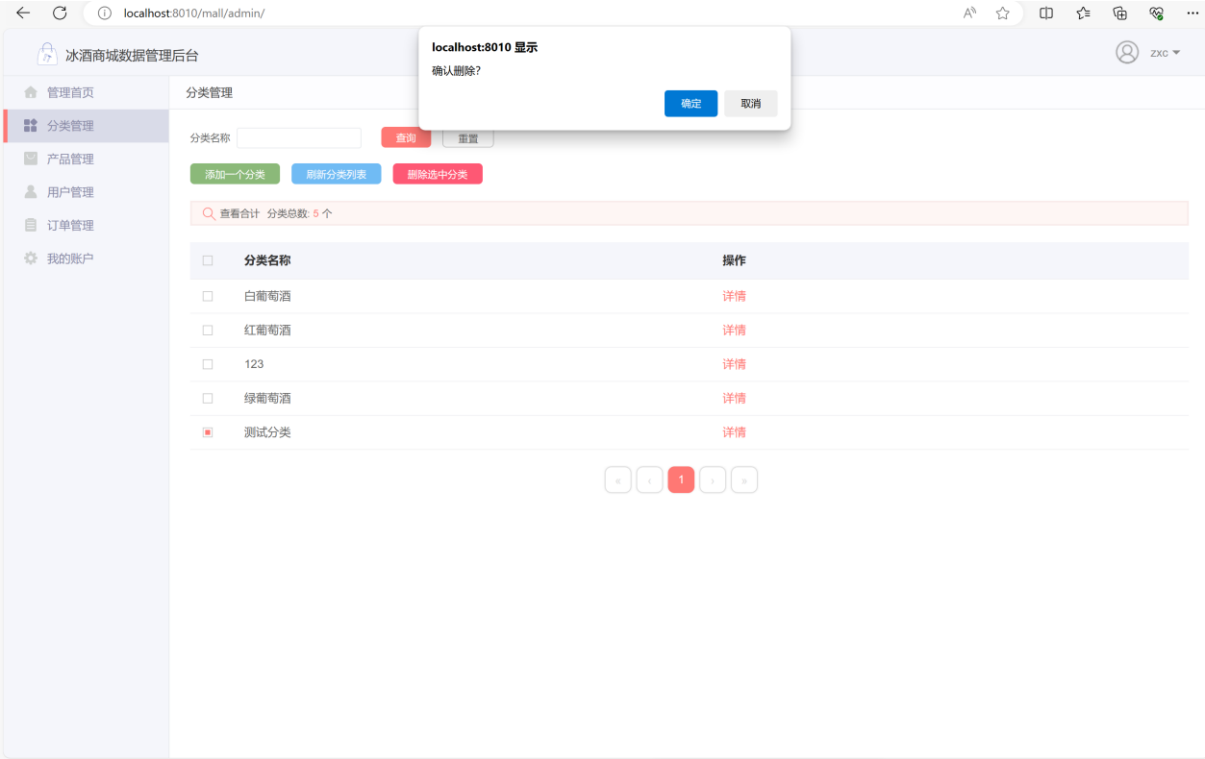
刷新列表查看

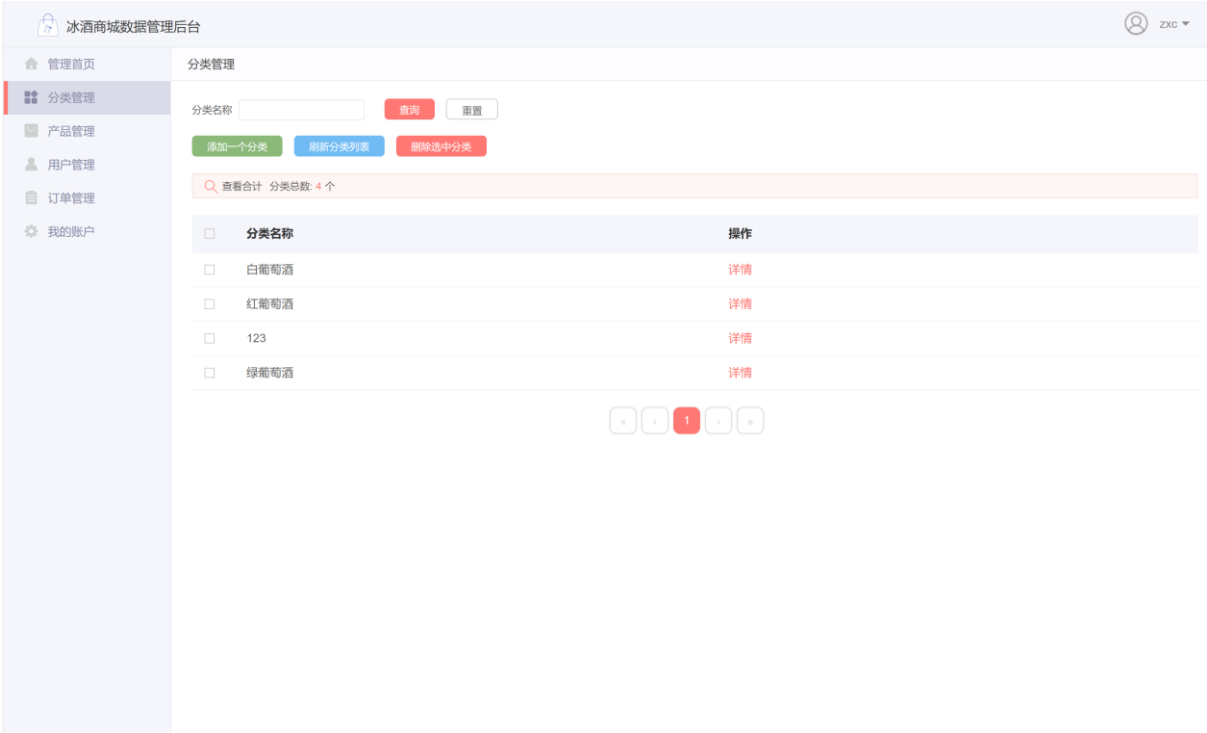


新增分类时名称不为空


冰酒商城数据管理后台
zxc
管理首页
分类管理
产品管理
用户管理
订单管理
我的账户
添加分类
分类编号
系统指定
分类名称
分类图片
点击上传
保存
取消

删除分类


localhost:8010/mall/admin/
冰酒商城数据管理后台
zxc
localhost:8010 显示
确认删除?
确定
取消
管理首页
分类管理
产品管理
用户管理
订单管理
我的账户
分类管理
分类名称
查询
重置
添加一个分类
刷新分类列表
删除选中分类
查看合计 分类总数: 5 个
分类名称
操作
白葡萄酒
详情
红葡萄酒
详情
123
详情
绿葡萄酒
详情
测试分类
详情
1

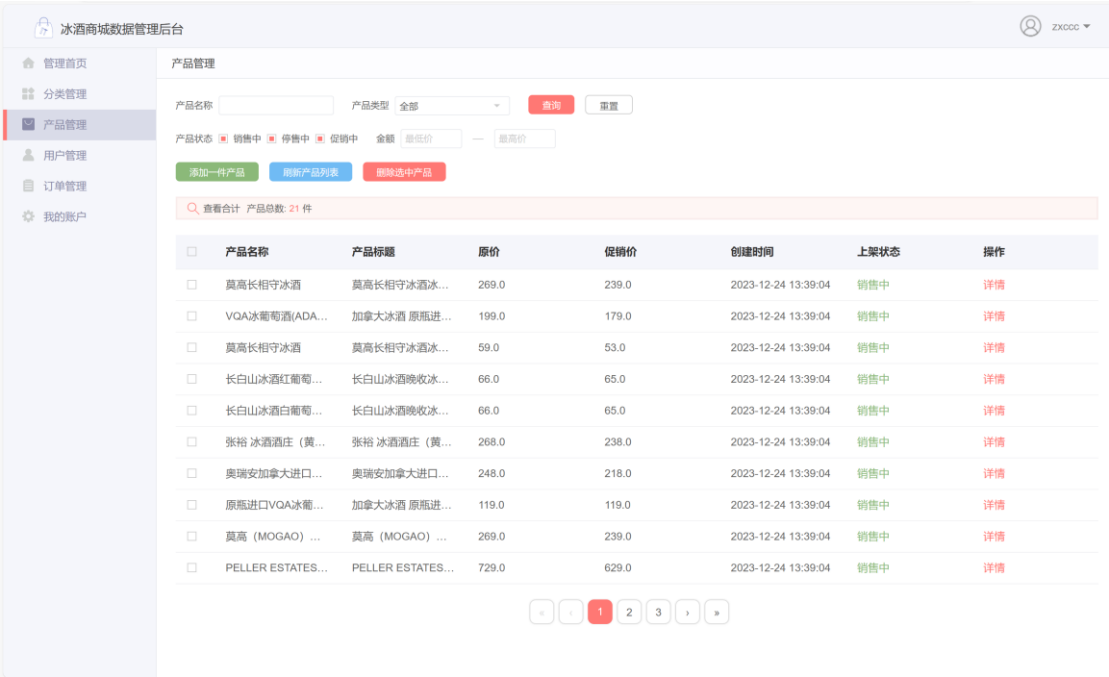


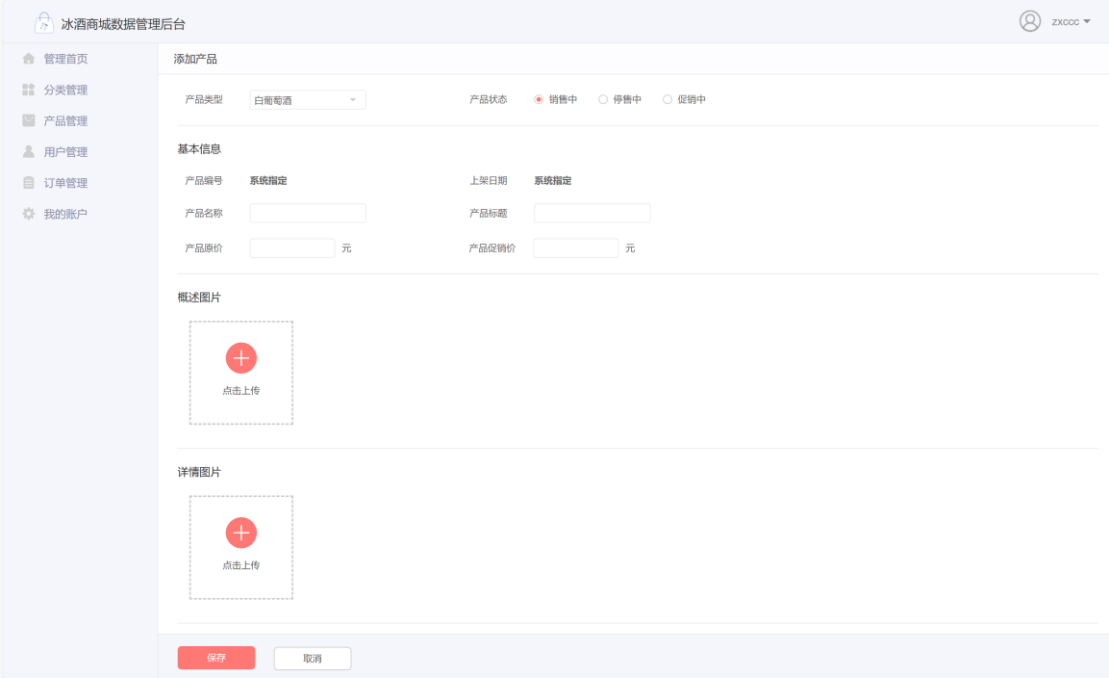
测试结果

序号	测试目的 & 先决条件	测试用例输入	测试结果	备注
1	**目的：** 检测分类管理页面的“详情 ”操作是否正确	**用例 1 ：** 点击分类“白葡萄酒”右侧的“详情”	显示分类“白葡萄酒”的详情	
2	**目的：** 检测分类管理页面的“查询 ”操作是否正确	**用例 2 ：** 在“分类名称”的文本框中输入“葡萄”后，点击右侧的“查询”按钮	列表显示名称有关键词“葡萄”的分类	
		用例 3 ： 在“分类名称”的文本框中输入空值后，点击右侧的“查询”按钮	列表显示所有分类	
3	**目的：** 测试分类管理页面的“重置 ”操作是否正确 先决条件：先前进行过查询	**用例 4 ：** 在“分类名称”的文本框中输入“葡萄”后，点击右侧的“查询”按钮，再点击“重置”按钮	点击“查询”按钮后，列表显示名称有关键词“葡萄”的分类；点击“重置”按钮后，“分类名称”的文本框清空，列表显示所有分类	
4	**目的：** 测试分类管理页面的“添加一个分类 ”操作是否正确	**用例 5 ：** 点击“添加一个分类”，在“添加分类”页面的“分类名称”中输入“测试分类”，并上传“分类图片”后，点击下方的“确定”按	成功创建“测试分类”，列表中显示该分类，在详情中可看到该分类的图片为上传的图片	

		钮		
		用例 6 ： 点击“添加一个分类”，在“添加分类”页面的“分类名称”中输入空值，并上传“分类图片”后，点击下方的“确定”按钮	显示提示信息：“分类名称”不能为空	
5	**目的：** 测试分类管理页面的“刷新分类列表 ”操作是否正确	**用例 7 ：** 用命令行添加分类“测试分类”后，点击“刷新分类列表”按钮	列表刷新后显示用命令行添加的分类	
6	**目的：** 测试分类管理页面的“删除选中分类”操作是否正确	**用例 8 ：** 在列表中选择分类“测试分类”后，点击“删除选中分类”按钮，点击“刷新分类列表”按钮	列表刷新后不再显示分类“测试分类”	

9.2.4.1.4 产品管理测试





测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试“产品管理”页面中“查询”功能是否正确	**用例 1：** 在“产品名称”的文本框中输入“长白山”，点击“查询”按钮	列表显示“产品名称”含关键字“长白山”的产品	
		用例 2： 在“产品类型”的选择框中选择“白葡萄酒”，点击“查询”按钮	列表显示类别为“白葡萄酒”的产品	
		用例 3： 在“产品状态”的复选框中选择“销售中”，点击“查询”按钮	列表显示“产品状态”为“销售中”的产品	
		用例 4： 在“金额”的“最低价”和“最高价”中分别输入“300”和“500”，点击“查询”按钮	列表显示“销售价”介于 300 到 500 的产品	
		用例 5： 在“产品名称”的文本框中输入“长白山”，点击“查询”按钮，再点击“重置”按钮	点击“查询”按钮后列表显示“产品名称”含关键字“长白山”的产品；点击“重置”按钮后列表显示所有产品	
2	**目的：**	**用例 1：**	显示产品“莫高长相守冰	

	测试“产品管理”页面中“详情”功能是否正确	点击 产品“莫高长相守冰酒”右侧的“详情”	酒”的详情	
3	**目的：** 测试“产品管理”页面的“添加一件产品”操作是否正确	**用例 1 ：** 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选择“停售中”，在“产品名称”的文本框中输入“测试白葡萄酒名称”，在“产品标题”的文本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮	列表中出现“产品名称”为“测试白葡萄酒名称”、“产品标题”为“测试白葡萄酒标题”、“原价”为400、“促销价”为300、创建时间为测试时刻、上架状态为“停售中”的产品	
		用例 2 ： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择 框中选择“白葡萄酒”，在 “产品状态”的单选框中选 择“停售中”，在“产品名 称”的文本框中输入“空值”，在“产品标题”的文 本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促 销价”中输入“300”，上传 “概述图片”和“详情图 片”，点击“保存”按钮	显示提示信息：“产品名称”不能为空	
		用例 3 ： 点击“添加一件产品”按	显示提示信息：“产品标题”不能为空	

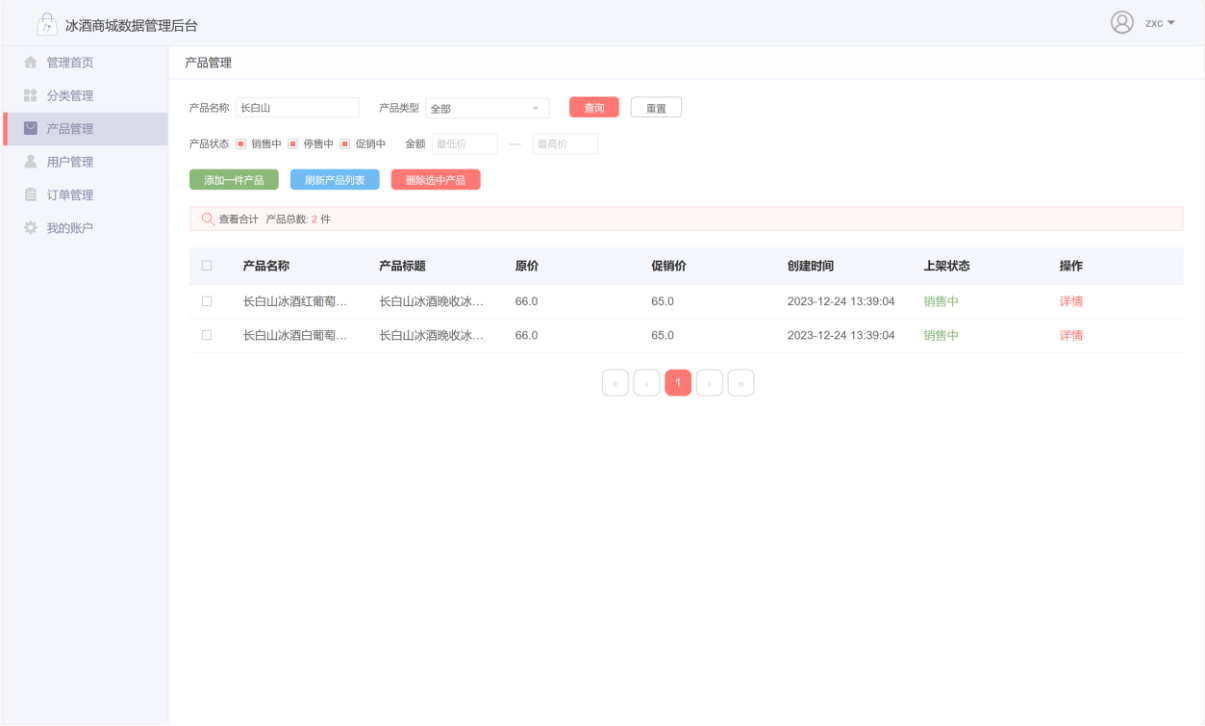
		钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“空 值”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮		
		用例 4 ： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“测 试白葡萄酒标题”，在“产品原价”中输入“空值”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮	显示提示信息：“产品原价”不能为空	
		用例 5 ： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试	显示提示信息：“促销价”不能为空	

		白葡萄酒名称”，在“产品标题”的文本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“空值”，上传“概述图片”和“详情图片”，点击“保存”按钮		
		用例 6： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选择“停售中”，在“产品名称”的文本框中输入“测试白葡萄酒名称”，在“产品标题”的文本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，不上传“概述图片”，上传“详情图片”，点击“保存”按钮	显示提示信息：“概述图片” 不能为空	
		用例 7： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选择“停售中”，在“产品名称”的文本框中输入“测试白葡萄酒名称”，在“产品标题”的文本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入	显示提示信息：“详情图片” 不能为空	

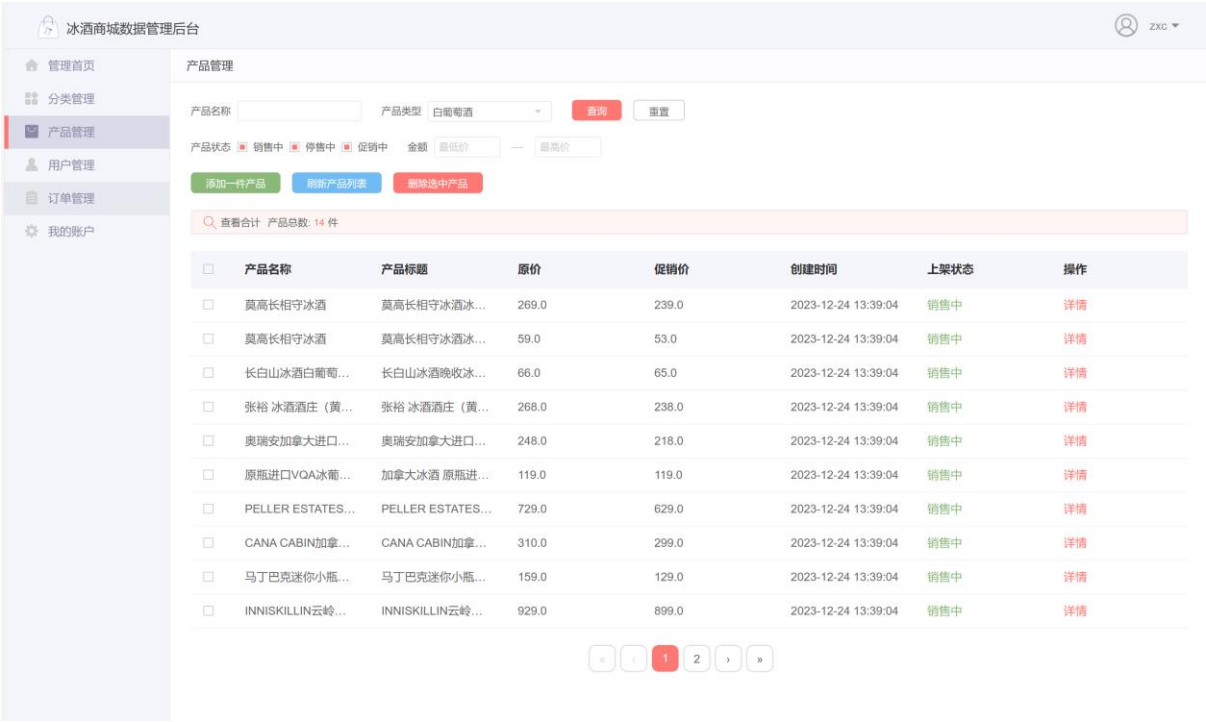
		“300”，上传“概述图片”，不上传“详情图片”，点击“保存”按钮		
4	**目的：** 测试“产品管理”页面的“刷新产品列表”操作是否正确	**用例 1：** 用命令行添加一个“产品类型”为“红葡萄酒”、“产品状态”为“促销中”、“产品名称”为“测试红葡萄酒名称”、“产品标题”为“测试红葡萄酒标题”、“产品原价”为“500”、“促销价”为“400”、“概述贴图”和“详情贴图”任意设 置的产品后，点击“刷新产品列表”按钮	列表刷新后显示新加入的产品	
5	**目的：** 测试“产品管理”页面的“删除选中产品”操作是否正确	**用例 1：** 在列表中选中产品“测试红葡萄酒名称”，点击“删除选中产品”按钮	列表中产品“测试红葡萄酒名称”被删除	

测试过程

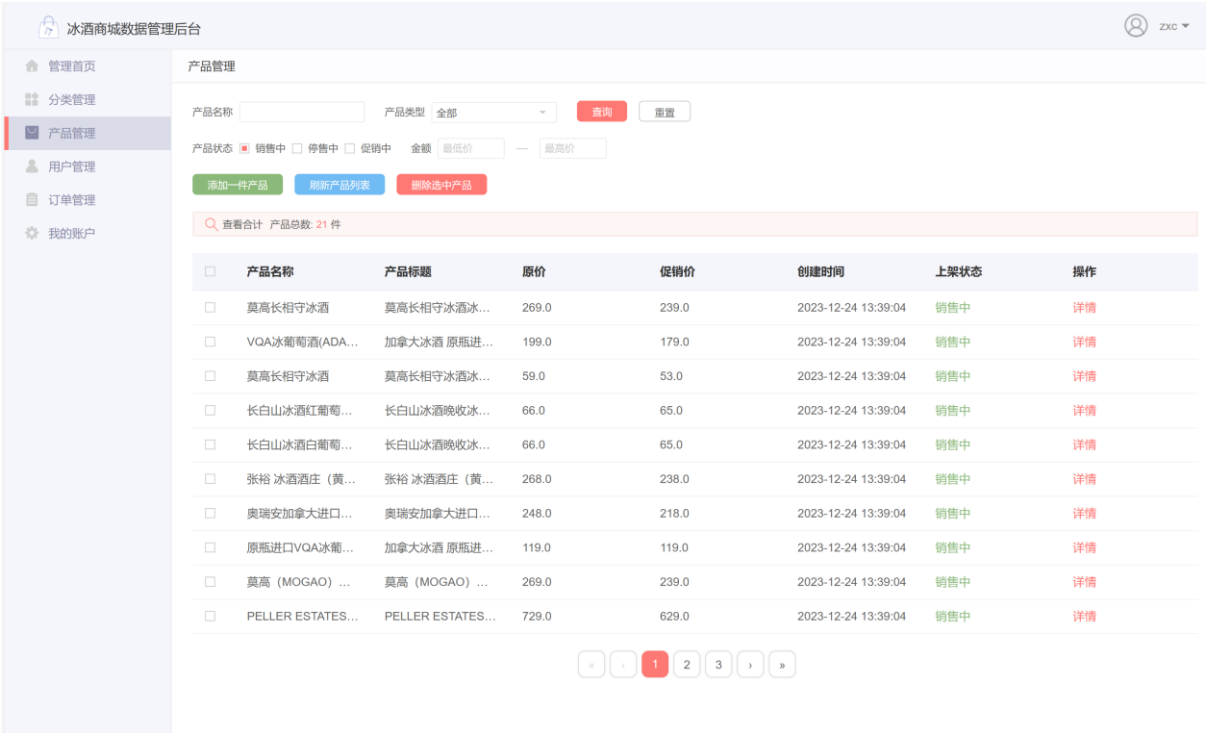
查询“长白山”相关商品



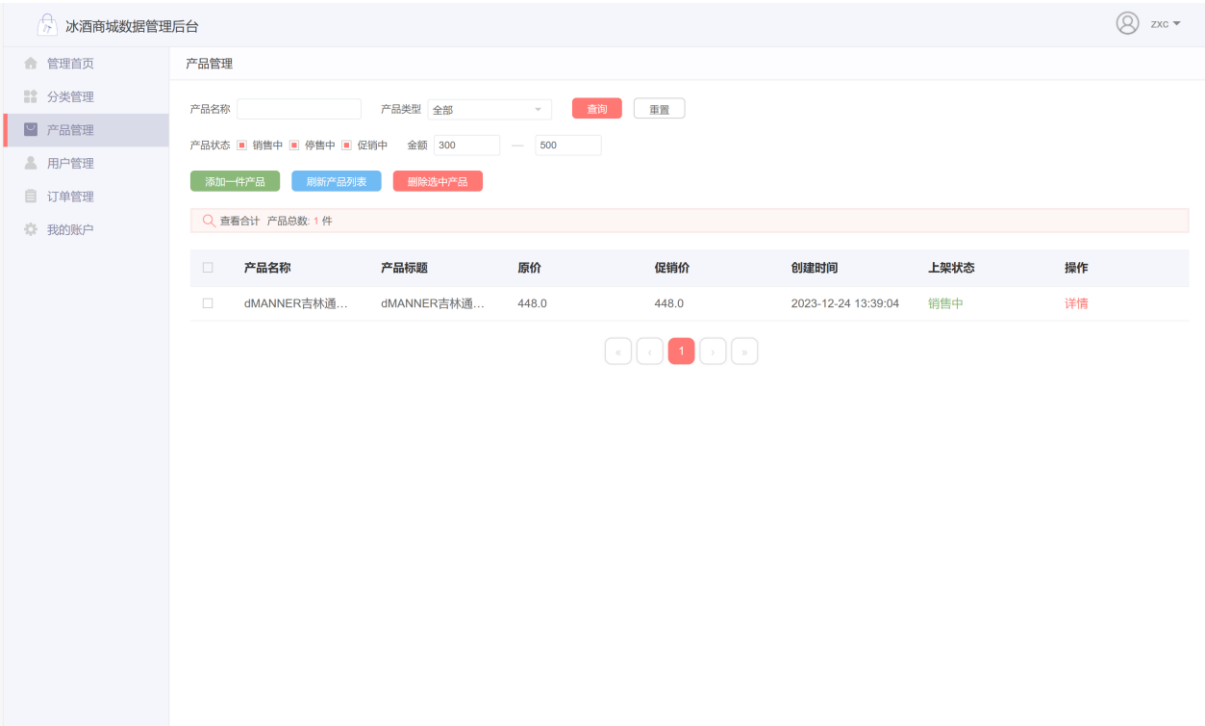
查询“白葡萄酒”类商品



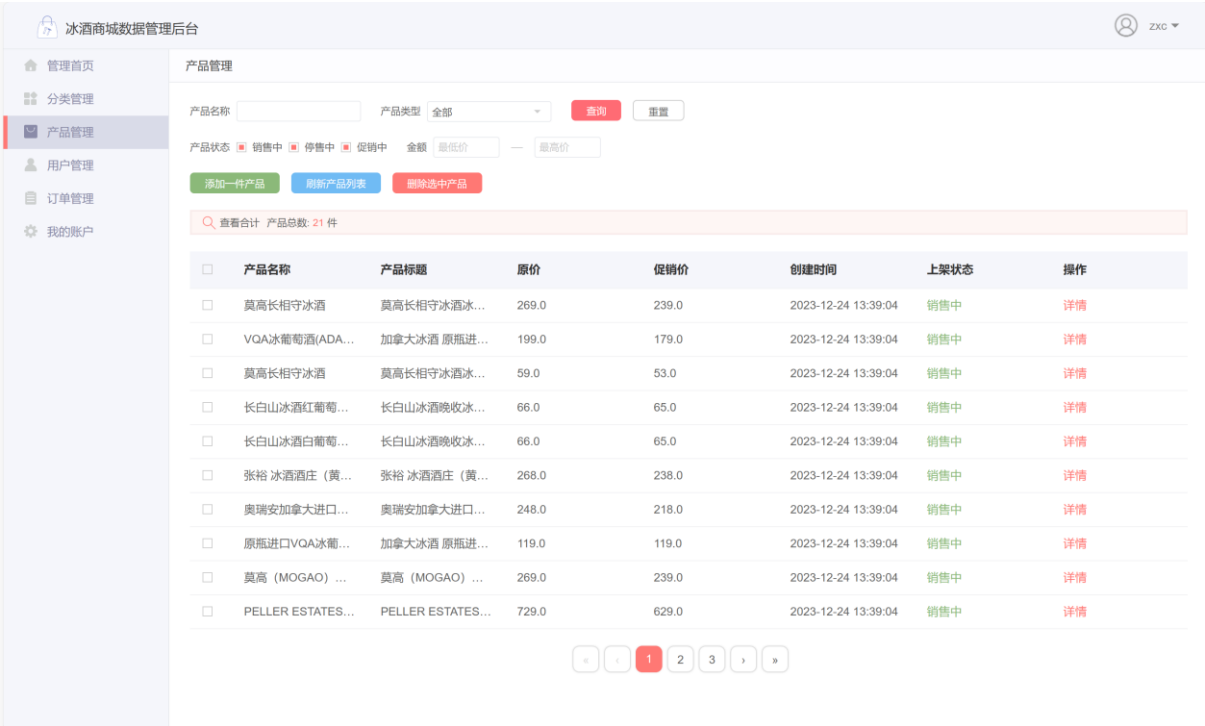
查询“销售中”商品



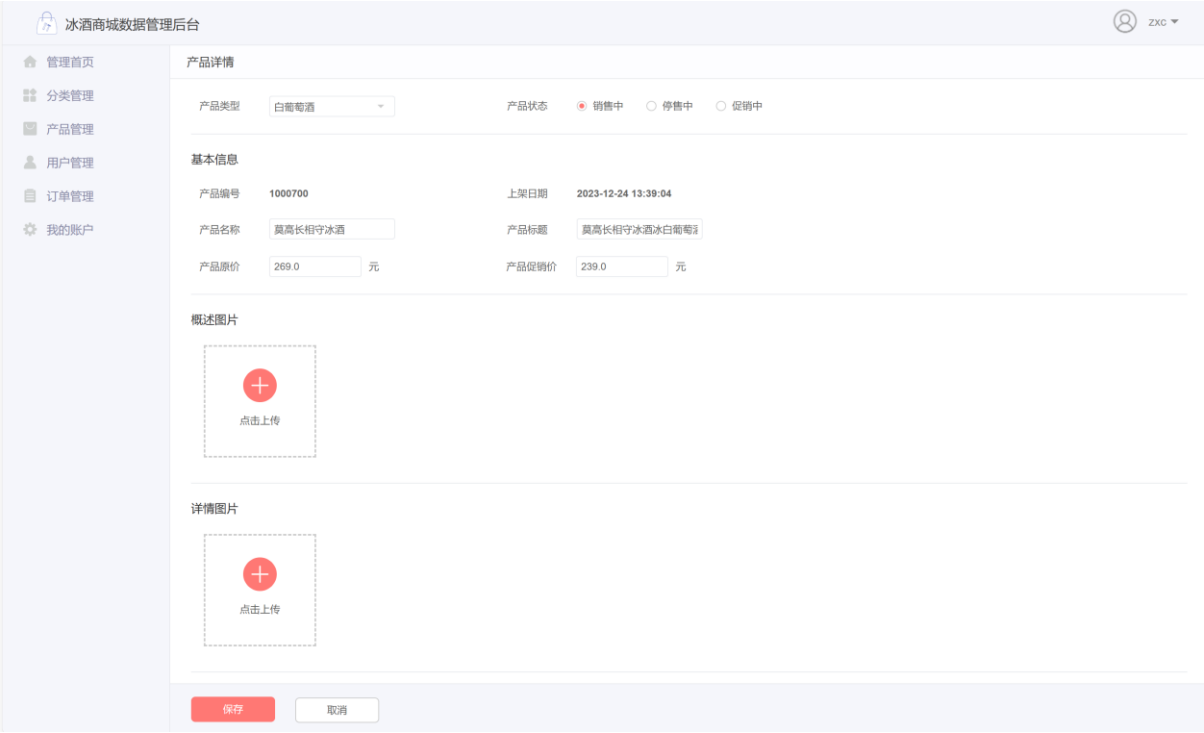
查询价格区间在 “300” 到 “500” 的商品



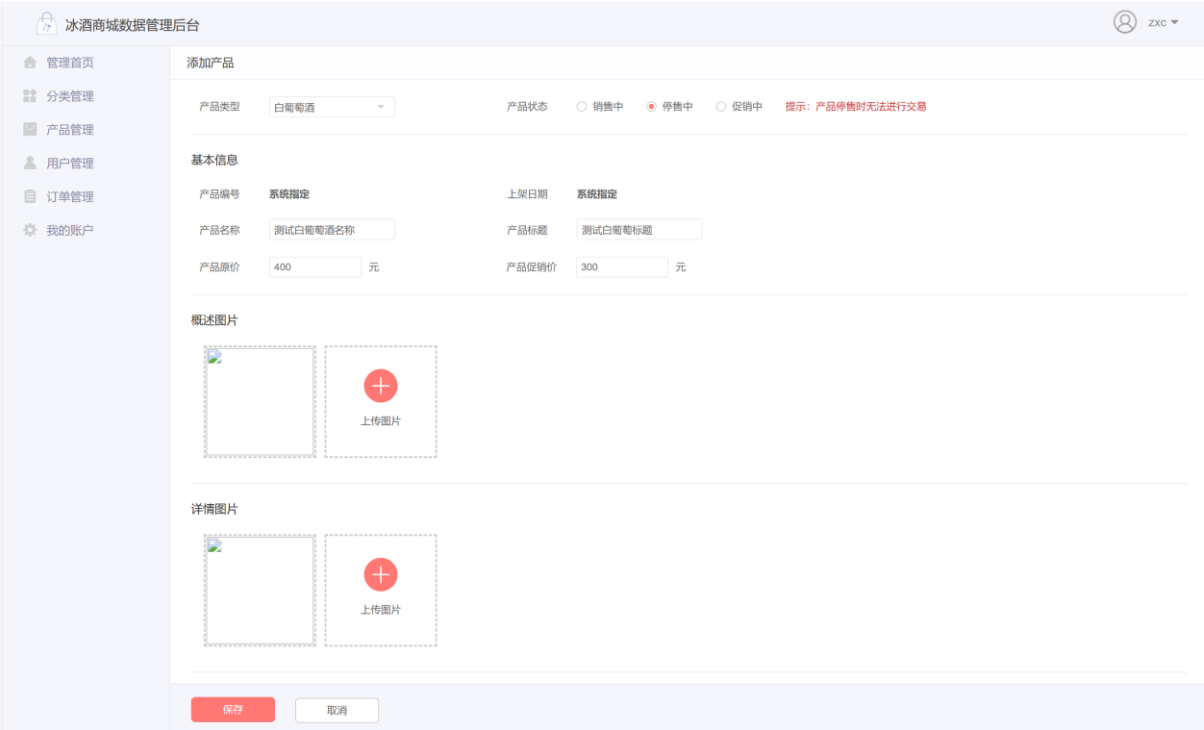
查询“长白山”然后重置



详情界面



产品管理的产品添加测试



测试结果

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：**	**用例 1 ：**	列表显示“产品名称”含	

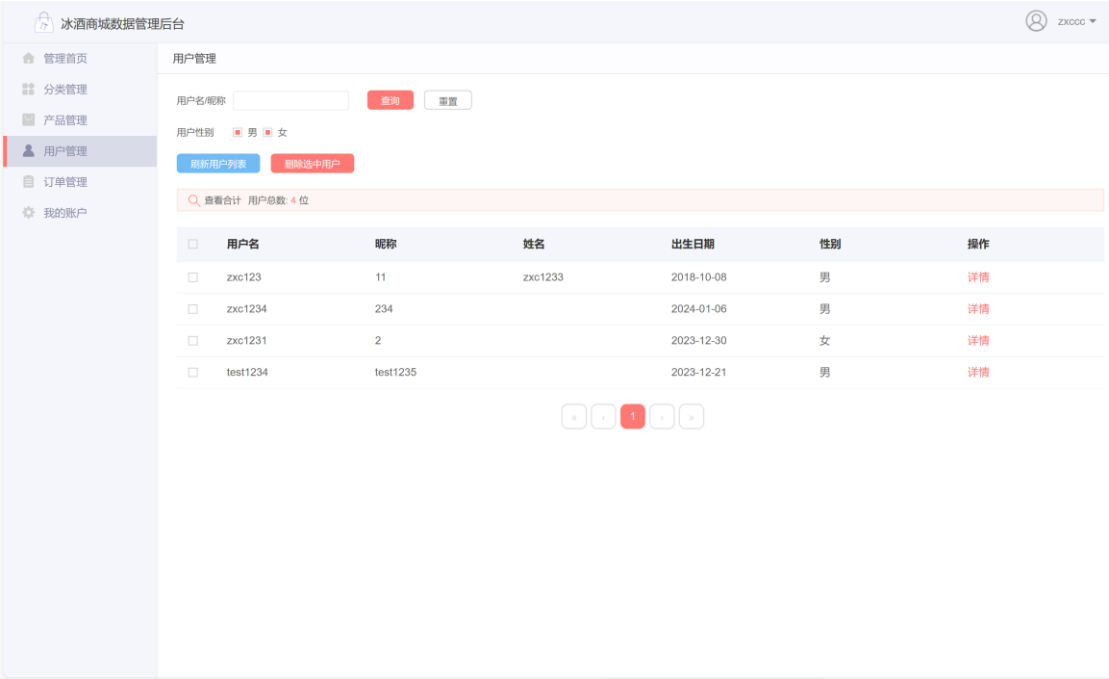
	测试“产品管理”页面中“查询”功能是否正确	在“产品名称”的文本框中输入“长白山”，点击“查询”按钮	关键字“长白山”的产品	
		用例 2： 在“产品类型”的选择框中选择“白葡萄酒”，点击“查询”按钮	列表显示类别为“白葡萄酒”的产品	
		用例 3： 在“产品状态”的复选框中选择“销售中”，点击“查询”按钮	列表显示“产品状态”为“销售中”的产品	
		用例 4： 在“金额”的“最低价”和“最高价”中分别输入“300”和“500”，点击“查询”按钮	列表显示“销售价”介于 300 到 500 的产品	
		用例 5： 在“产品名称”的文本框中输入“长白山”，点击“查询”按钮，再点击“重置”按钮	点击“查询”按钮后列表显示“产品名称”含关键字“长白山”的产品；点击“重置”按钮后列表显示所有产品	
2	**目的：** 测试“产品管理”页面中“详情”功能是否正确	**用例 1：** 点击 产品“莫高长相守冰酒”右侧的“详情”	显示产品“莫高长相守冰酒”的详情	
3	**目的：** 测试“产品管理”页面的“添加一件产品”操作是否正确	**用例 1：** 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选择“停售中”，在“产品名称”的文本框中输入“测试白葡萄酒名称”，在“产品标题”的文本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮	列表中出现“产品名称”为“测试白葡萄酒名称”、“产品标题”为“测试白葡萄酒标题”、“原价”为 400、“促销价”为 300、创建时间为测试时刻、上架状态为“停售中”的产品	bug
		用例 2： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选	显示提示信息：“产品名称”不能为空	

		择 框中选择“白葡萄酒”，在 “产品状态”的单选框中选 择“停售中”，在“产品名 称”的文本框中输入“空值”，在“产品标题”的文 本框中输入“测试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促 销价”中输入“300”，上传 “概述图片”和“详情图 片”，点击“保存”按钮		
		用例 3： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“空 值”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图 片”，点击“保存”按钮	显示提示信息：“产品标题”不能为空	
		用例 4： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选	显示提示信息：“产品原价”不能为空	

		择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“测 试白葡萄酒标题”，在“产品原价”中输入“空值”，在“产品促销价”中输入“300”，上传“概述图片”和“详情图片”，点击“保存”按钮		
		用例 5： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“测 试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“空值”，上传“概述图片”和“详情图片”，点击“保存”按钮	显示提示信息：“促销价”不能为空	
		用例 6： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“测	显示提示信息：“概述图片” 不能为空	

		试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，不上传“概述图片”，上传“详情图片”，点击“保存”按钮		
		用例 7 ： 点击“添加一件产品”按钮，在“添加产品”页面中，在“产品类型”的选择框中选择“白葡萄酒”，在“产品状态”的单选框中选 择“停售中”，在“产品名称”的文本框中输入“测试 白葡萄酒名称”，在“产品标题”的文本框中输入“测 试白葡萄酒标题”，在“产品原价”中输入“400”，在“产品促销价”中输入“300”，上传“概述图 片”，不上传“详情图 片”，点击“保存”按钮	显示提示信息：“详情图片” 不能为空	
4	**目的：** 测试“产品管理”页面的“刷新产品列表”操作是否正确	**用例 1 ：** 用命令行添加一个“产品类型”为“红葡萄酒”、“产品状态”为“促销中”、“产品名称”为“测试红葡萄酒名称”、“产品标题”为“测试红葡萄酒标题”、“产品原价”为“500”、“促销价”为“400”、“概述贴图”和“详情贴图”任意设 置的产品后，点击“刷新产品列表”按钮	列表刷新后显示新加入的产品	
5	**目的：** 测试“产品管理”页面的“删除选中产品”操作是否正确	**用例 1 ：** 在列表中选中产品“测试红葡萄酒名称”，点击“删除选中产品”按钮	列表中产品“测试红葡萄酒名称”被删除	

9.2.4.1.5 用户管理测试



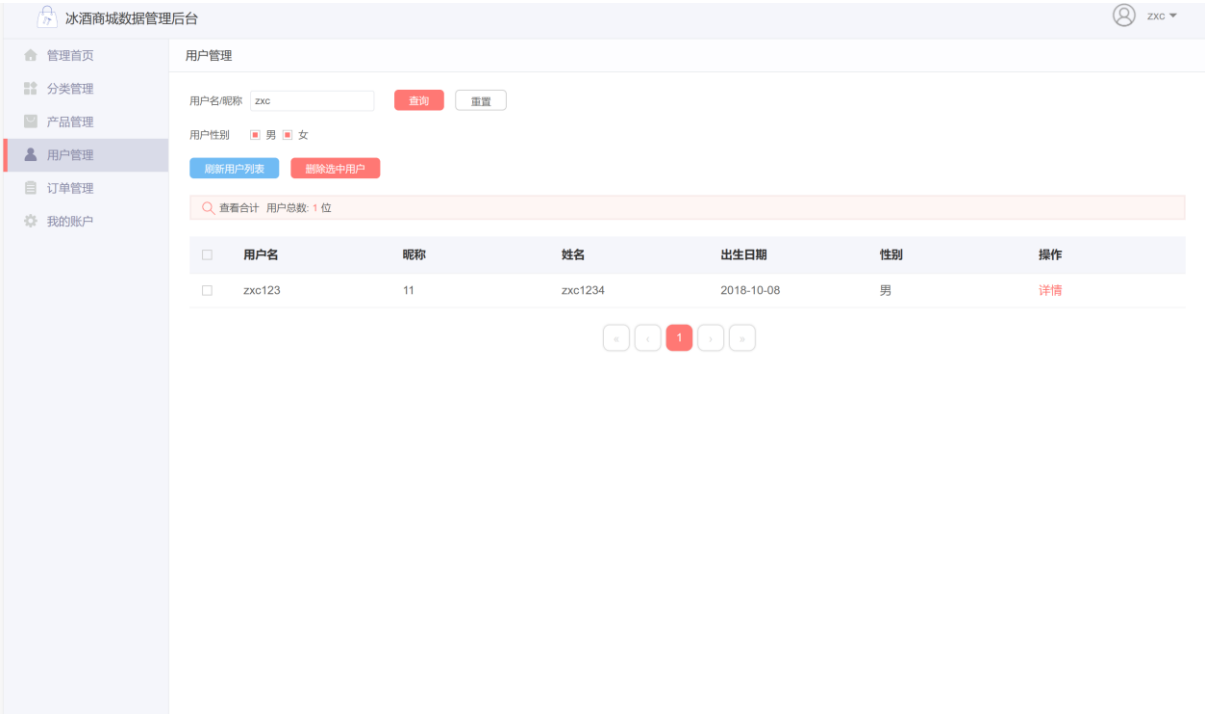
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试“用户管理”页面的“查询”操作是否正确	**用例 1：** 在“用户名/昵称”的文本框中输入“zxc”，点击“查询”按钮	列表显示“用户名/昵称”含关键词“zxc”的用户	
		用例 2： 在“用户性别”的复选框中选择“男”，点击“查询”按钮	列表显示“用户性别”为“男”的用户	
2	**目的：** 测试“用户管理”页面的“重置”操作是否正确	**用例 1：** 在“用户名/昵称”的文本框中输入“zxc”，点击“查询”按钮，再点击“重置”按钮	点击“查询”按钮后列表显示“用户名/昵称”含关键词“zxc”的用户；点击“重置”按钮后列表显示所有用户	
3	**目的：** 测试“用户管理”页面的“刷新用户列表”操作是否正确	**用例 1：** 在登录页面注册一个新用户“test123”后，点击“刷新用户列表”按钮	列表显示新注册的用户	
4	**目的：** 测试“用户管理”页面的“删除选择用户”操作是否正确	**用例 1：** 在列表中选中“用户名”为“test123”的用户，点击“删除选择用户”按钮	列表中用户“test123”被删除	
5	**目的：**	**用例 1：**	显示用户“zxc123”的详	

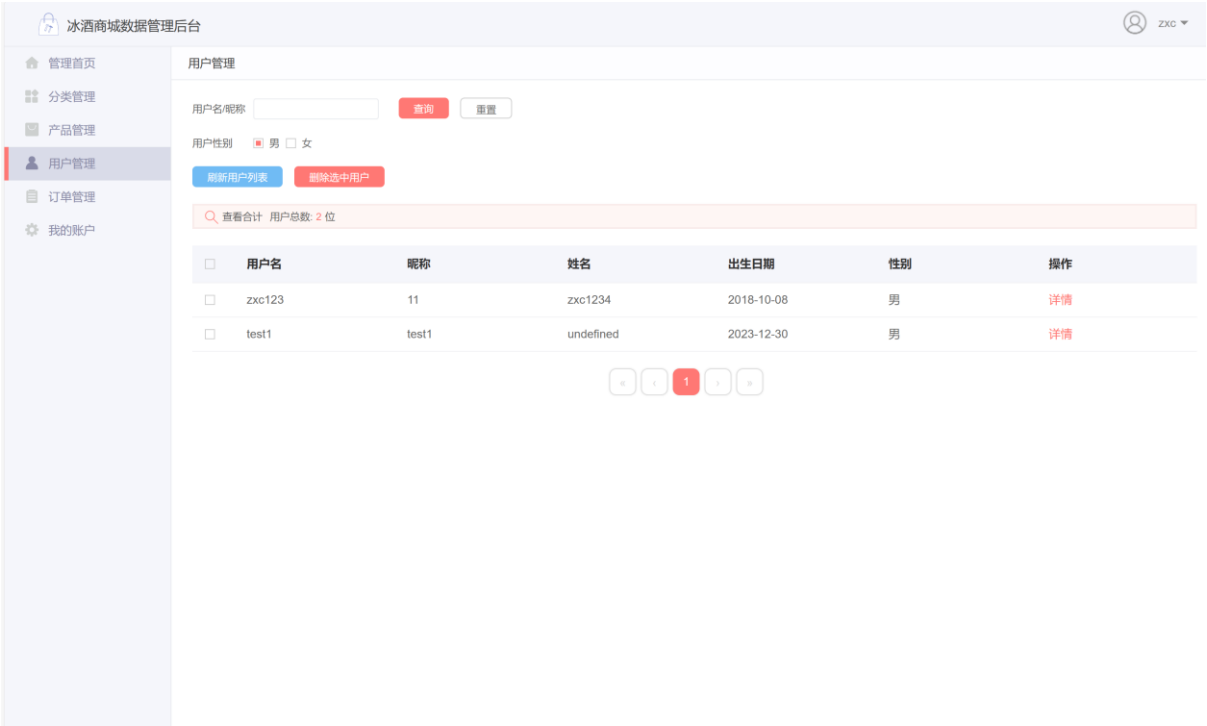
	测试“用户管理”页面的“详情”操作是否正确	点击列表中“用户名”为“zxc123”的用户右侧的“详情”	情	

测试过程

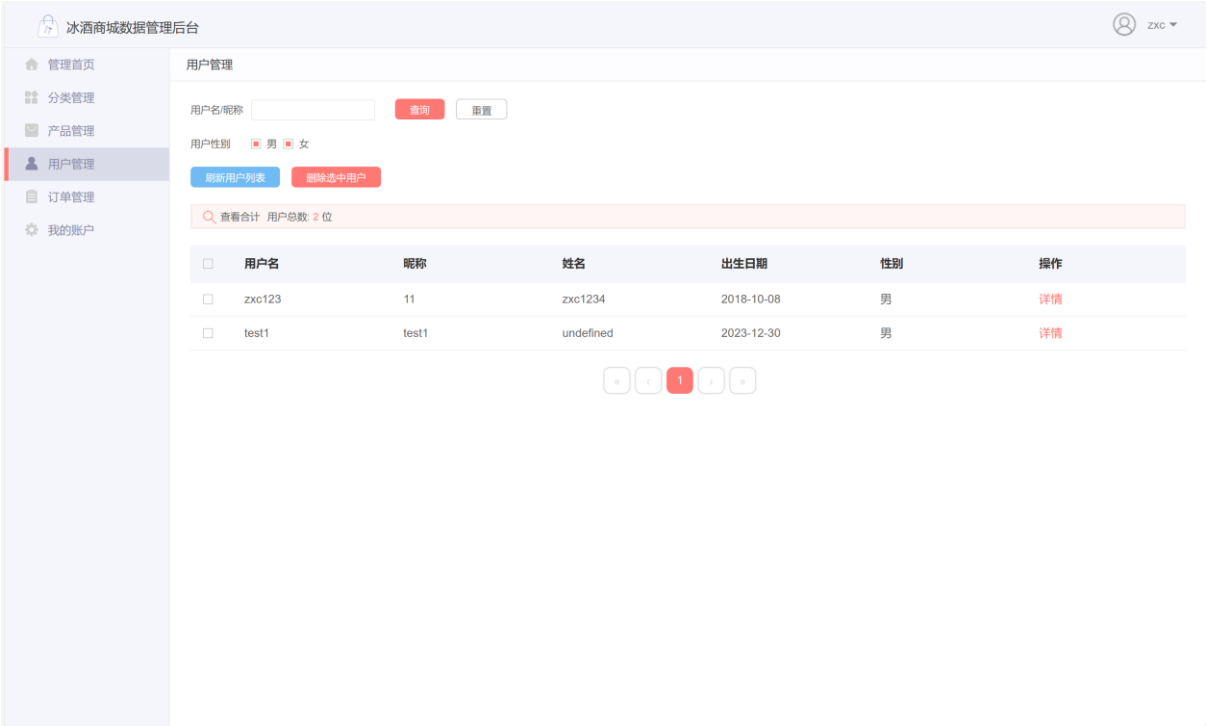
查询用户"zxc"



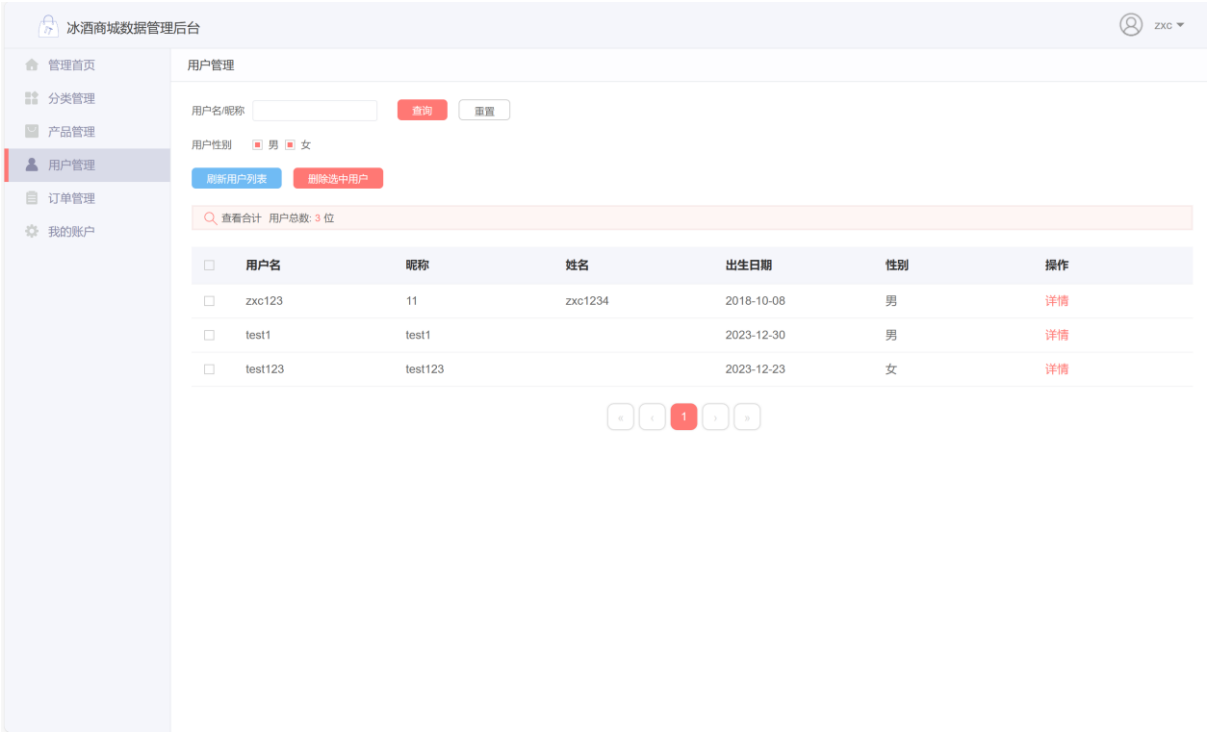
查询“男”用户



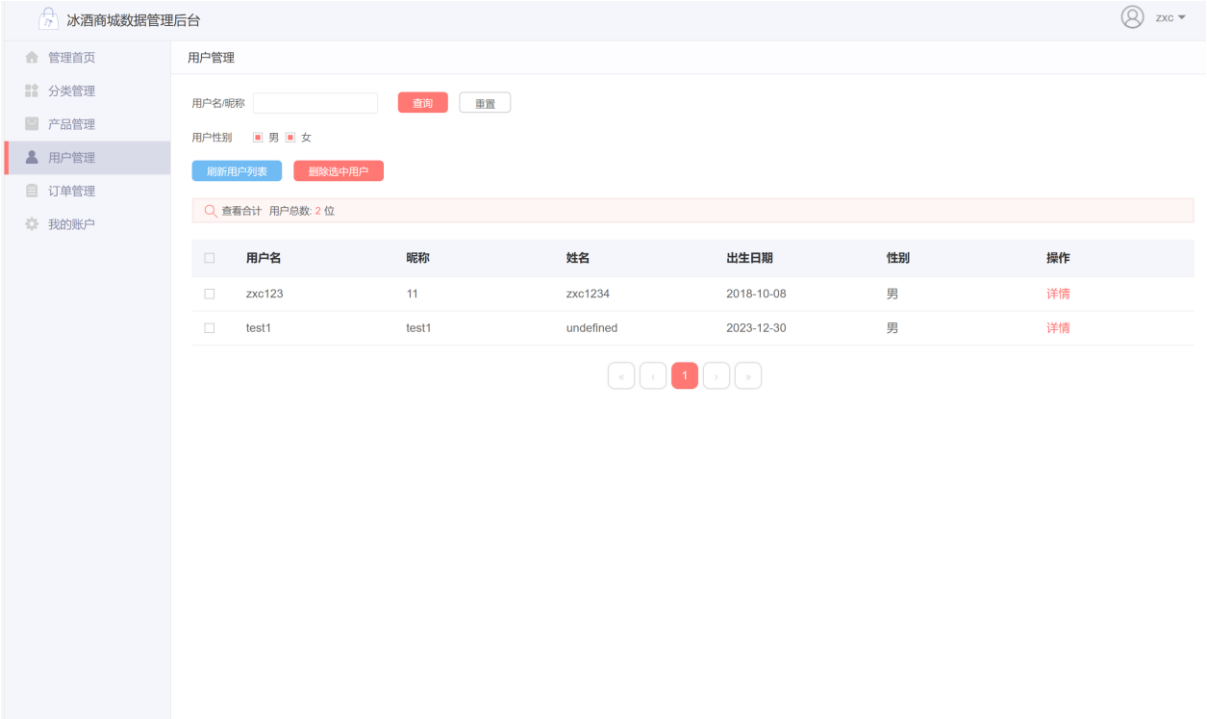
查询后重置



新增用户“test123”



删除用户“test123”



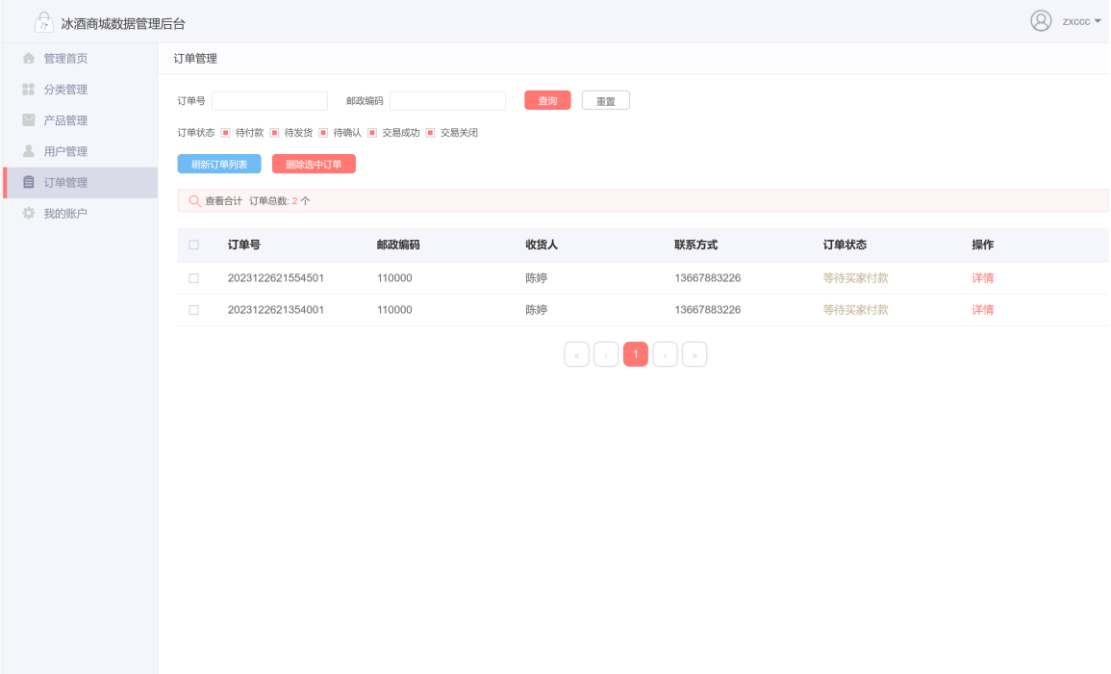
查看用户“zxc123” 的详情

测试结果

序号	测试目的 & 先决条件	测试用例输入	测试结果	备注
1	**目的：** 测试“用户管理”页面的“查询”操作是否正确	**用例 1：** 在“用户名/昵称”的文本框中输入“zxc”，点击“查询”按钮	列表显示“用户名/昵称”含关键词“zxc”的用户	
		用例 2： 在“用户性别”的复选框中选择“男”，点击“查询”按钮	列表显示“用户性别”为“男”的用户	
2	**目的：** 测试“用户管理”页面的“重置”操作是否正确	**用例 1：** 在“用户名/昵称”的文本框中输入“zxc”，点击“查询”按钮，再点击“重置”按钮	点击“查询”按钮后列表显示“用户名/昵称”含关键词“zxc”的用户；点击“重置”按钮后列表显示所有用户	
3	**目的：** 测试“用户管理”页面的“刷新用户列表”操作是否正确	**用例 1：** 在登录页面注册一个新用户“test123”后，点击“刷新用户列表”按钮	列表显示新注册的用户	
4	**目的：** 测试“用户管理”页面的“删除选择用户”操作是否正确	**用例 1：** 在列表中选中“用户名”为“test123”的用户，点击“删除选择用户”按钮	列表中用户“test123”被删除	
5	**目的：**	**用例 1：**	显示用户“zxc123”的详	

	测试“用户管理”页面的“详情”操作是否正确	点击列表中“用户名”为“zxc123”的用户右侧的“详情”	情	

9.2.4.1.6 订单管理测试



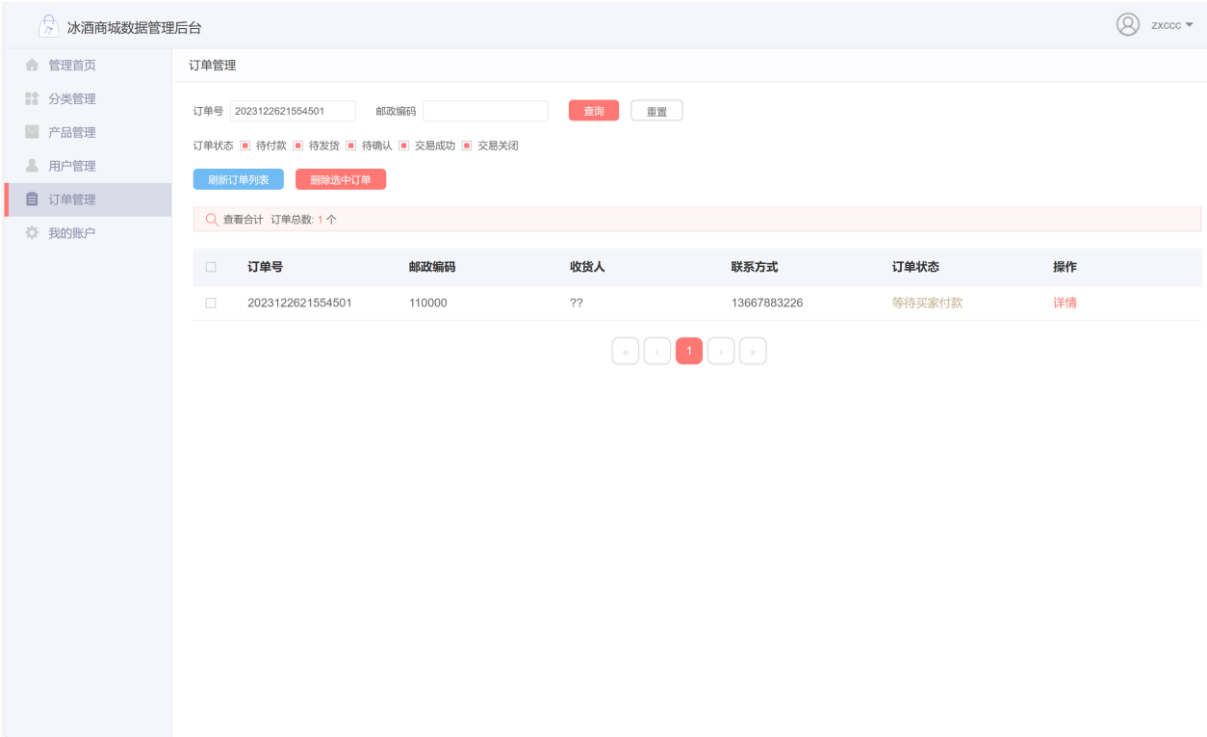
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试“订单管理”页面的“查询”操作是否正确	**用例 1：** 在“订单号”的文本框输入“2021”，点击“查询”按钮	列表显示“订单号”包含关键词“2021”的订单	
		用例 2： 在“邮政编码”的文本框中输入“110000”，点击“查询”按钮	列表显示“邮政编码”包含关键词“110000”的订单	
		用例 3： 在“订单状态”的复选框中选择“待发货”，点击“查询”按钮	列表显示“订单状态”为“待发货”的订单	
2	**目的：** 测试“订单管理”页面的“详情”操作是否正确	**用例 1：** 点击列表中订单号为“2021102209242001”的订单右侧的“详情”	显示订单号为“2021102209242001”的订单右侧的详情	
3	**目的：**	**用例 1：**	列表显示新加入的“订单	

	测试“订单管理”页面的“刷新订单列表”操作是否正确	用命令行添加一个“订单号”为“2023122620415801”、“邮政编码”为“110000”、“收货人”为“陈婷”、“订单状态”为“待付款”的订单后，点击“刷新订单列表”按钮	号”为“2023122620415801”、“邮政编码”为“110000”、“收货人”为“陈婷”、“订单状态”为“待付款”的订单	
4	**目的：** 测试“订单管理”页面的“删除选中订单”操作是否正确	**用例 1：** 选中列表中订单号为“2023122620415801”的订单，点击“删除选中订单”	列表中订单号为“2023122620415801”的订单被删除	

测试过程

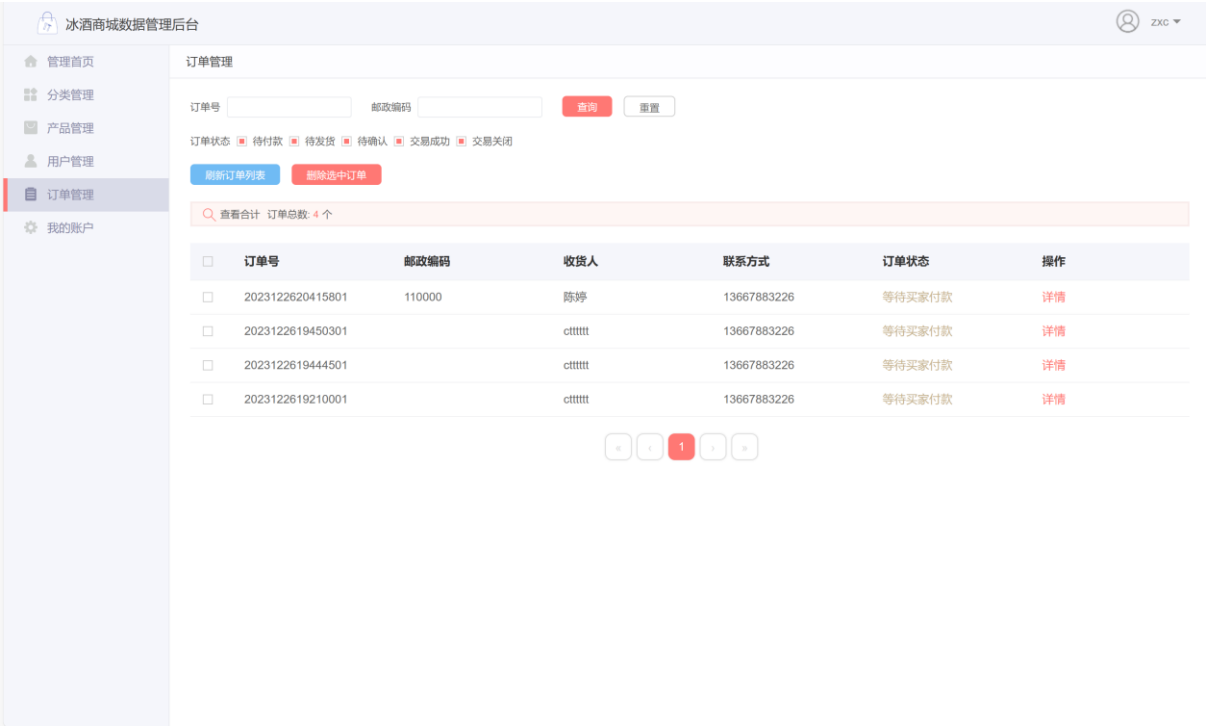
订单查询



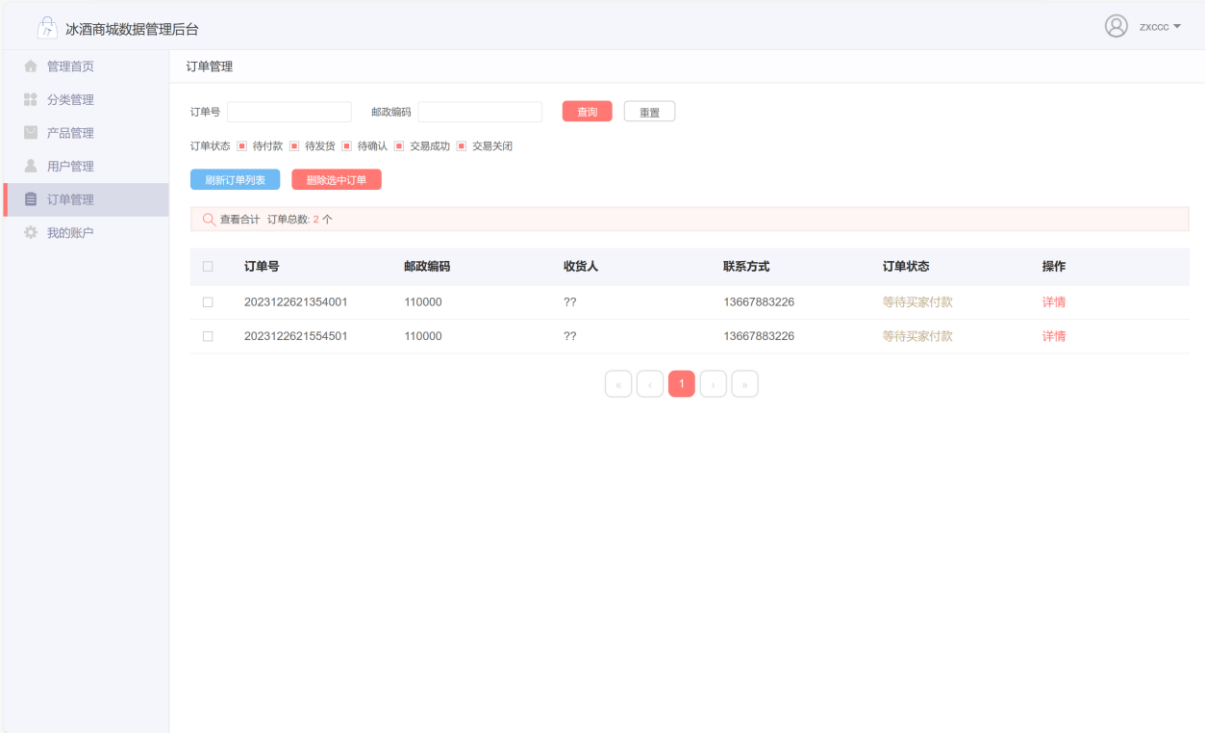
订单详情页面

冰酒商城数据管理后台
zxc
管理首页
分类管理
产品管理
用户管理
订单管理
我的账户
订单详情
订单号 2023122619450301
所属用户
基本信息
收货人姓名 ctttttt
收货地址 北京 北京市 东城区 地下垃圾桶
邮政编码
联系电话 13667883226
订单状态
等待买家付款
等待卖家发货
等待买家确认
订单交易完成
订单项信息
产品图片 产品名称 单价 数量 价格 备注 操作
VQA冰葡萄酒(ADA... 179.0 1 179.0 详情
取消

新增订单


冰酒商城数据管理后台
zxc
管理首页
分类管理
产品管理
用户管理
订单管理
我的账户
订单管理
订单号
邮政编码
查询
重置
订单状态 待付款 待发货 待确认 交易成功 交易关闭
刷新订单列表
删除选中订单
查看合计 订单总数: 4 个
订单号 邮政编码 收货人 联系方式 订单状态 操作
2023122620415801 110000 陈婷 13667883226 等待买家付款 详情
2023122619450301 ctttttt 13667883226 等待买家付款 详情
2023122619444501 ctttttt 13667883226 等待买家付款 详情
2023122619210001 ctttttt 13667883226 等待买家付款 详情
1

删除订单

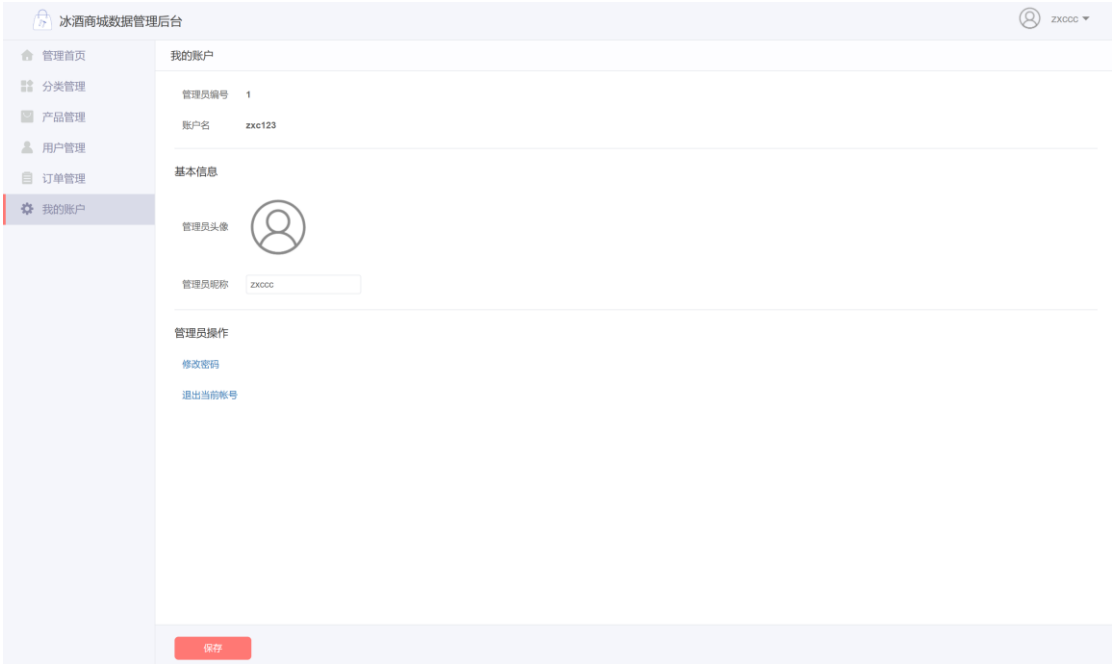


测试结果

序号	测试目的 & 先决条件	测试用例输入	测试结果	备注
1	**目的：** 测试“订单管理”页面的“查询”操作是否正确	**用例 1：** 在“订单号”的文本框输入“2021”，点击“查询”按钮	列表显示“订单号”包含关键词“2021”的订单	bug
		用例 2： 在“邮政编码”的文本框中输入“110000”，点击“查询”按钮	列表显示“邮政编码”包含关键词“110000”的订单	
		用例 3： 在“订单状态”的复选框中选择“待发货”，点击“查询”按钮	列表显示“订单状态”为“待发货”的订单	
2	**目的：** 测试“订单管理”页面的“详情”操作是否正确	**用例 1：** 点击列表中订单号为“2021102209242001”的订单右侧的“详情”	显示订单号为“2021102209242001”的订单右侧的详情	
3	**目的：** 测试“订单管理”页面的“刷新订单列表”操作是否正确	**用例 1：** 用命令行添加一个“订单号”为“2023122620415801”、“邮政编码”为“110000”、“收货人”为	列表显示新加入的“订单号”为“2023122620415801”、“邮政编码”为“110000”、“收货人”为“陈婷”、“订单状态”为	

		“陈婷”、“订单状态”为“待付款”的订单后，点击“刷新订单列表”按钮	“待付款”的订单	
4	**目的：** 测试“订单管理”页面的“删除选中订单”操作是否正确	**用例 1：** 选中列表中订单号为“2023122620415801”的订单，点击“删除选中订单”	列表中订单号为“2023122620415801”的订单被删除	bug

9.2.4.1.7 我的账户测试



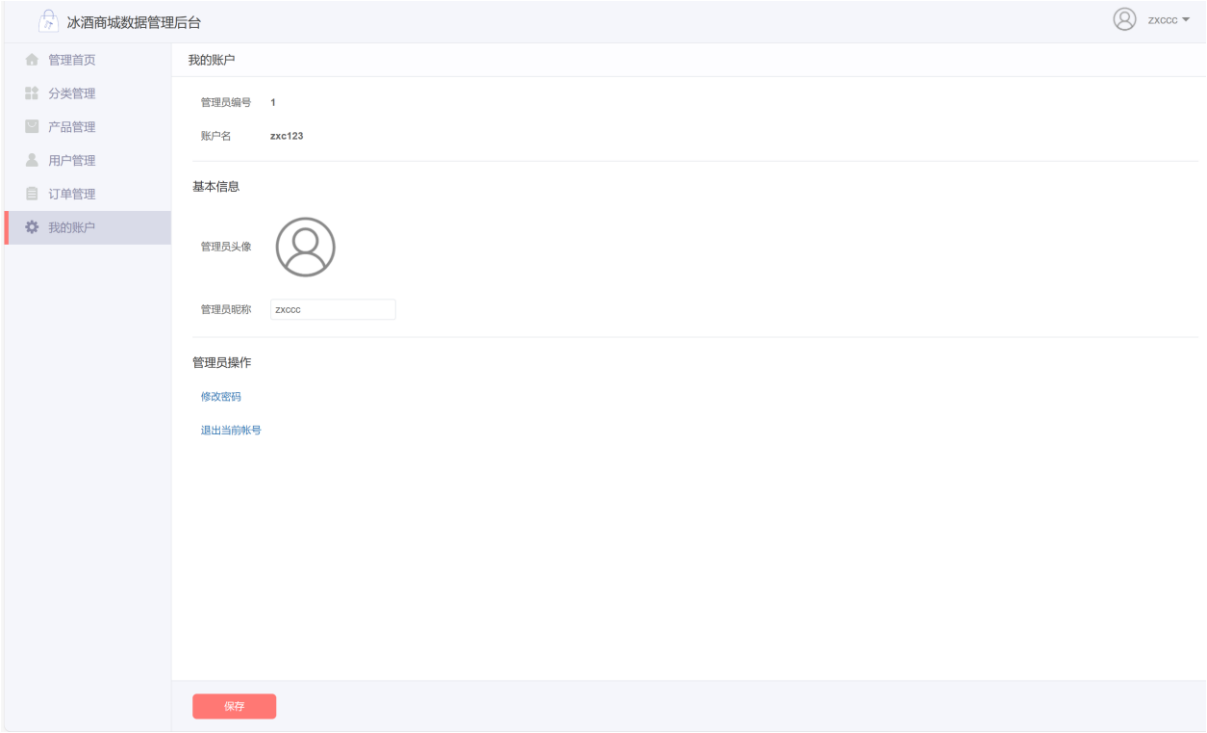
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试“我的账户”页面中修改“管理员昵称”是否正确 **先决条件：** 已登录账号	**用例 1：** 将“管理员昵称”的文本框中的内容改为“zxccc”，点击下方的“保存”	刷新页面后，右上角的用户昵称为“zxccc”	
2	**目的：** 测试“我的账户”页面中“修改密码”操作是否正确 **先决条件：** 已登录账号	**用例 1：** 在“修改密码”中“当前密码”的文本框中输入当前账号的密码，在“新密码”的文本框中输入要设定的新密码，在“确认密码”的文本框中再次输入	当前账号的密码更新为新密码，在后台的登陆页面使用旧密码无法登录，使用新密码可以登录	

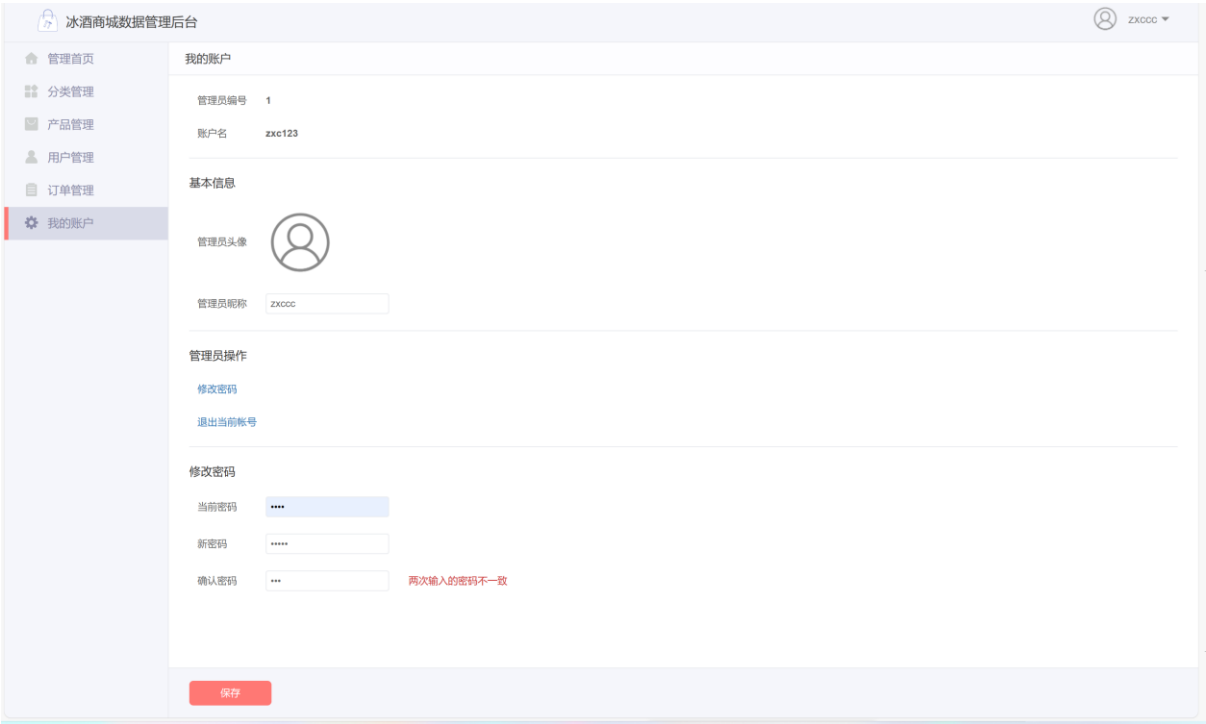
		要设定的新密码，点击下方的“保存”按钮		
		用例 2 ： 在“修改密码”中“当前密码”的文本框中输入当前账号的密码，在“新密码”的文本框中输入要设定的新密码，在“确认密码”的文本框中输入错误的要设定的新密码，点击下方的“保存”按钮	输出提示信息：“新密码”与“确认密码”不一致	
3	**目的：** 测试“我的账户”页面中“退出当前账号”操作是否正确 **先决条件：** 已登录账号	**用例 1 ：** 点击“退出当前账号”按钮	当前账号被退出，回到后台的登录页面	

测试过程

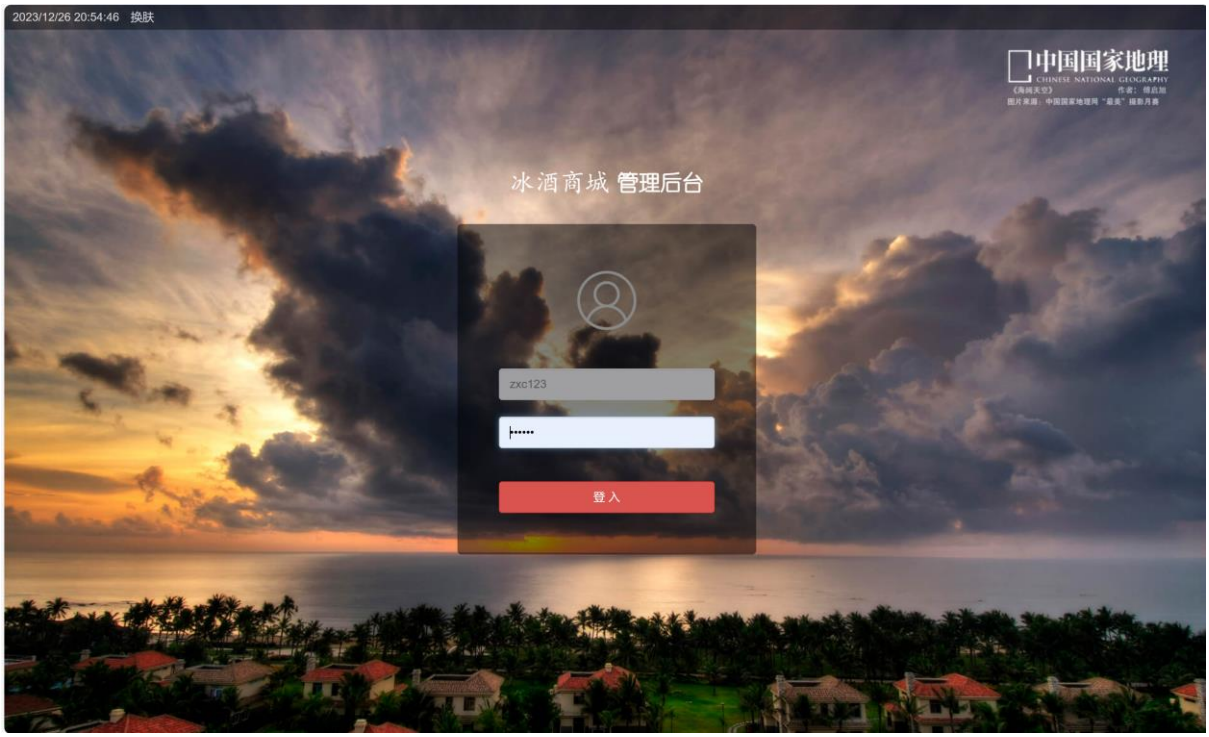
账号信息修改



两次密码不一致



退出当前帐号



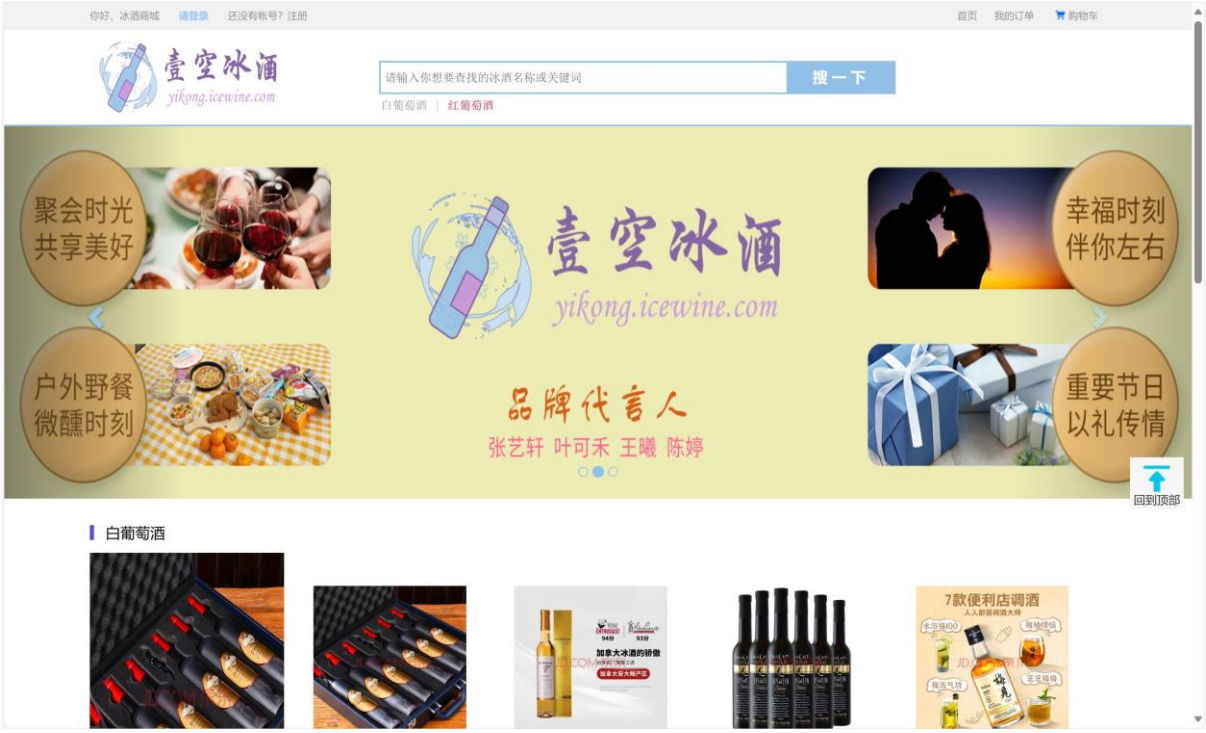
测试结果

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注

1	**目的：** 测试“我的账户”页面中修改“管理员昵称”是否正确 **先决条件：** 已登录账号	**用例 1 ：** 将“管理员昵称”的文本框中的内容改为“zxccc”，点击下方的“保存”	刷新页面后，右上角的用户昵称为“zxccc”	
2	**目的：** 测试“我的账户”页面中“修改密码”操作是否正确 **先决条件：** 已登录账号	**用例 1 ：** 在“修改密码”中“当前密码”的文本框中输入当前账号的密码，在“新密码”的文本框中输入要设定的新密码，在“确认密码”的文本框中再次输入要设定的新密码，点击下方的“保存”按钮	当前账号的密码更新为新密码，在后台的登陆页面使用旧密码无法登录，使用新密码可以登录	
		用例 2 ： 在“修改密码”中“当前密码”的文本框中输入当前账号的密码，在“新密码”的文本框中输入要设定的新密码，在“确认密码”的文本框中输入错误的要设定的新密码，点击下方的“保存”按钮	输出提示信息：“新密码”与“确认密码”不一致	
3	**目的：** 测试“我的账户”页面中“退出当前账号”操作是否正确 **先决条件：** 已登录账号	**用例 1 ：** 点击“退出当前账号”按钮	当前账号被退出，回到后台的登录页面	

9.2.4.2 前台功能测试

9.2.4.2.1 首页测试



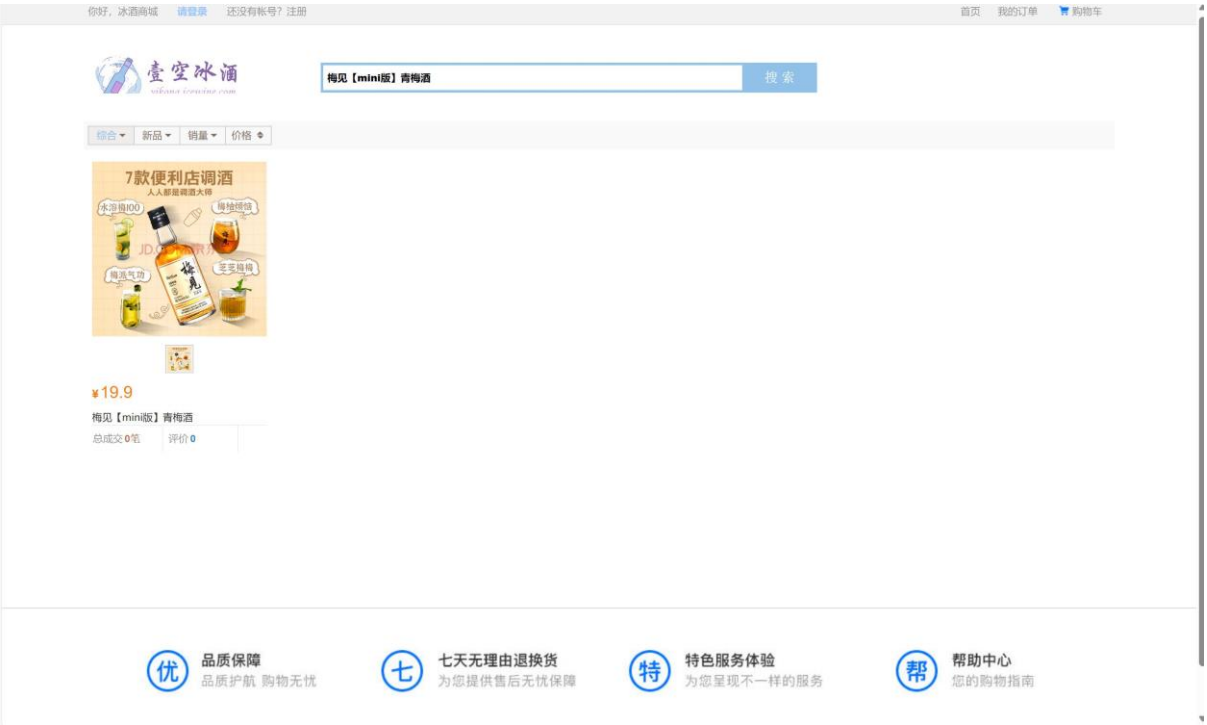
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 首页 的“搜索框”操作是否正确	**用例 1 ：** 在 搜索 文本框输入“ 梅见【mini 版】青梅酒 ”，点击“搜一下”按钮	显示搜索结果中有且仅有“ 梅见【mini 版】青梅酒 ”这一商品	精确全名搜索
		用例 2 ： 在 搜索 文本框输入“ 葡萄 ”，点击“搜一下”按钮	显示搜索结果中有所有包含“ 葡萄 ”这一关键字的商品	关键词搜索
		用例 3 ： 在 搜索 文本框输入“ 玫瑰 ”，点击“搜一下”按钮	显示没有找到相关商品	搜索不存在的商品
2	**目的：** 测试 首页 的 导航菜单是否正常跳转	**用例 1 ：** 点击搜索框下方导航栏的 “ 白葡萄酒”	跳转到“白葡萄酒”商品展示界面，显示该种类下所有商品	查看筛选是否准确
		用例 2 ： 点击搜索框下方导航栏的 “ 红葡萄酒”	跳转到“红葡萄酒”商品展示界面，显示该种类下所有商品	

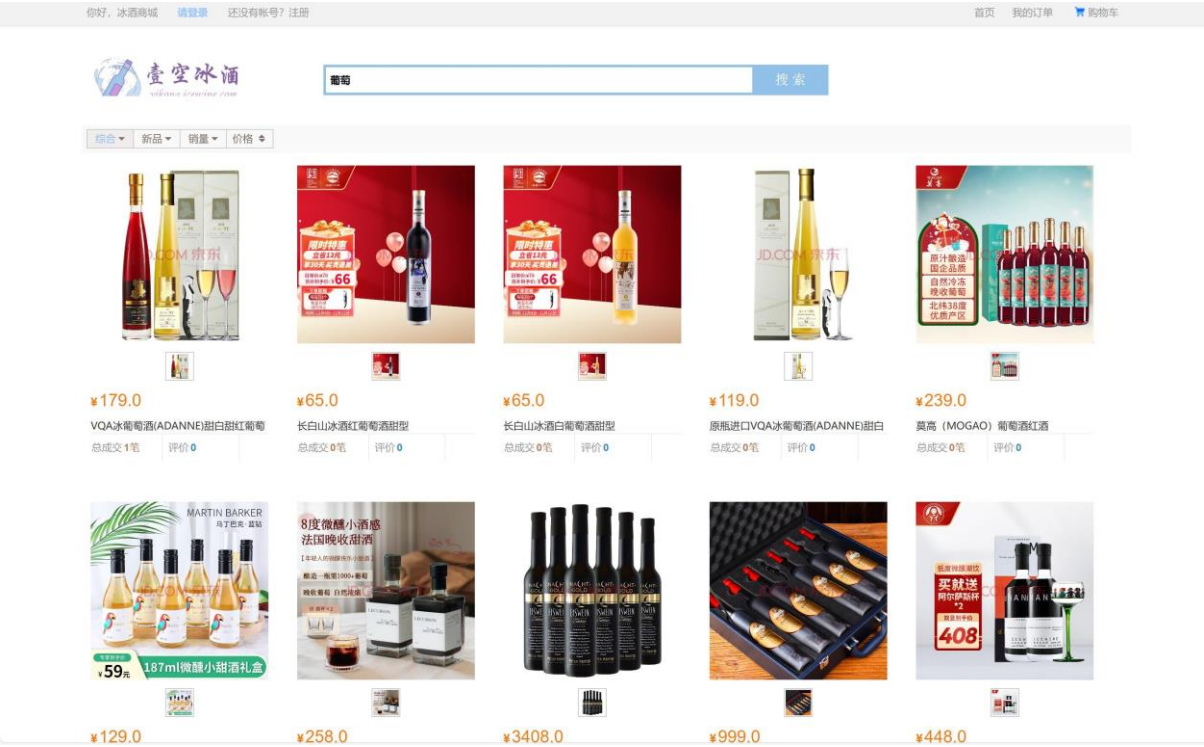
3	**目的：** 测试 首页 的宣传栏是否正常工作，能否切换不同图片	**用例 1 ：** 多次点击 “下一页” 按键，看图片能否正常切换	三张图片按照正常顺序切换	
		用例 2： 多次点击 “上一页” 按键，看图片能否正常切换		
4	**目的：** 测试 首页的 “回到顶部” 按键是否正常工作	**用例 1 ：** 下拉页面，多次滑动到不同位置，点击“回到顶部”，看页面能否回到顶部	点击 ”回到顶部“ 后页面成功回到顶部	
5	**目的：** 测试 首页的 某一 商品能否跳转到对应详情页面	**用例 1 ：** 点击首页的“VQA 冰葡萄酒(ADANNE)甜白甜红葡萄酒”，看能否正常跳转	成功跳转到“VQA 冰葡萄酒(ADANNE)甜白甜红葡萄酒” 页面	
		用例 2 ： 点击首页的“CANA CABIN 加拿大原瓶进口冰酒”，看能否正常跳转	成功跳转到“CANA CABIN 加拿大原瓶进口冰酒” 页面	
		用例 3 ： 点击首页的“露颂法国进口晚收女生晚安热红葡萄酒”，看能否正常跳转	成功跳转到“露颂法国进口晚收女生晚安热红葡萄酒” 页面	

测试过程

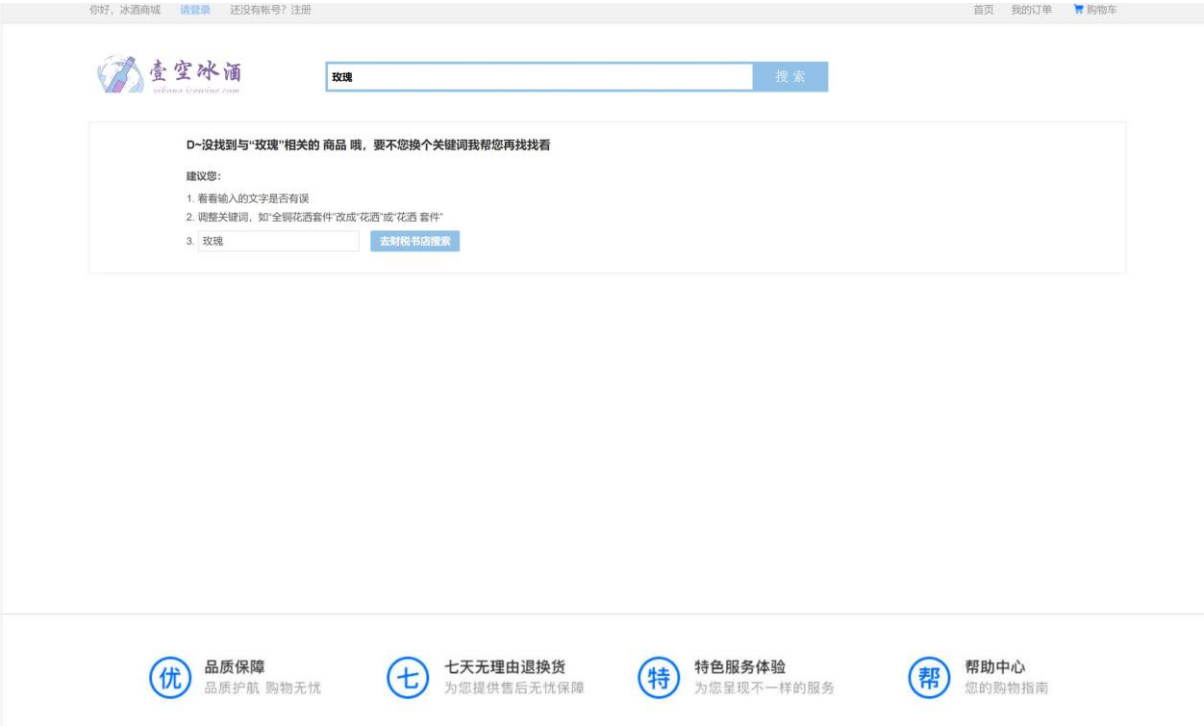
在 搜索 文本框输入“ 梅见【mini 版】青梅酒 ”，点击“搜一下”按钮：



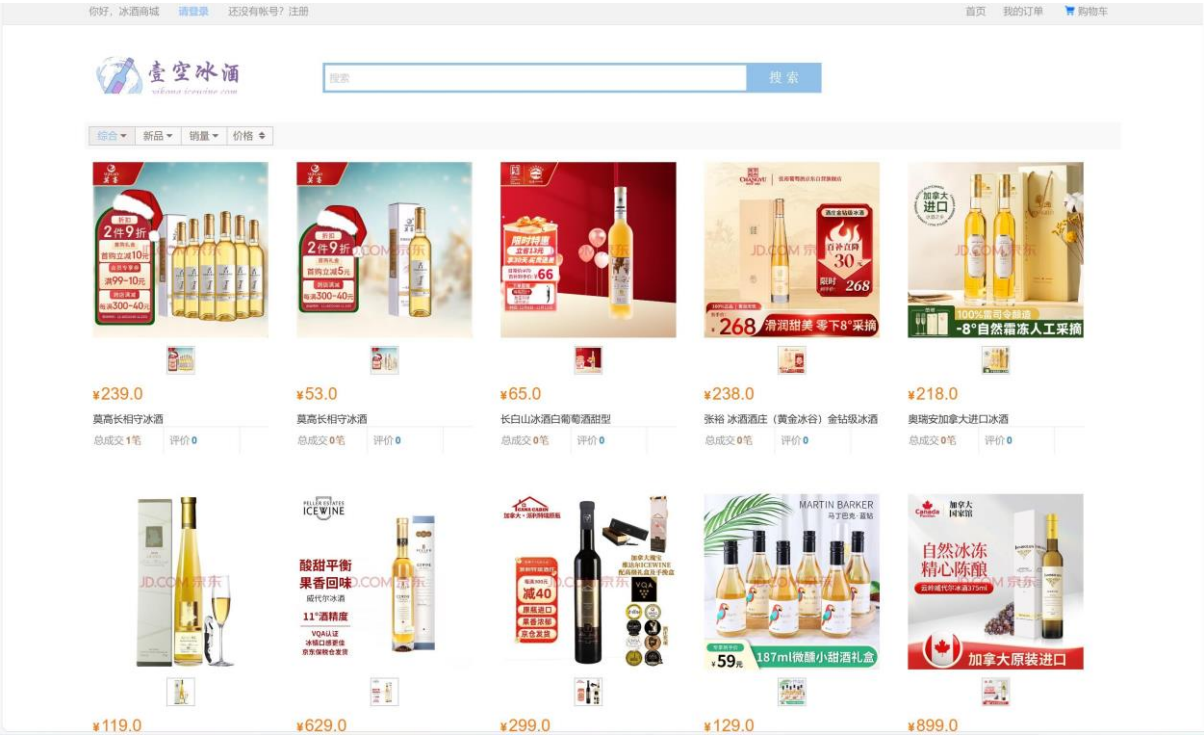
在 搜索 文本框输入“ 葡萄 ”，点击“搜一下”按钮：



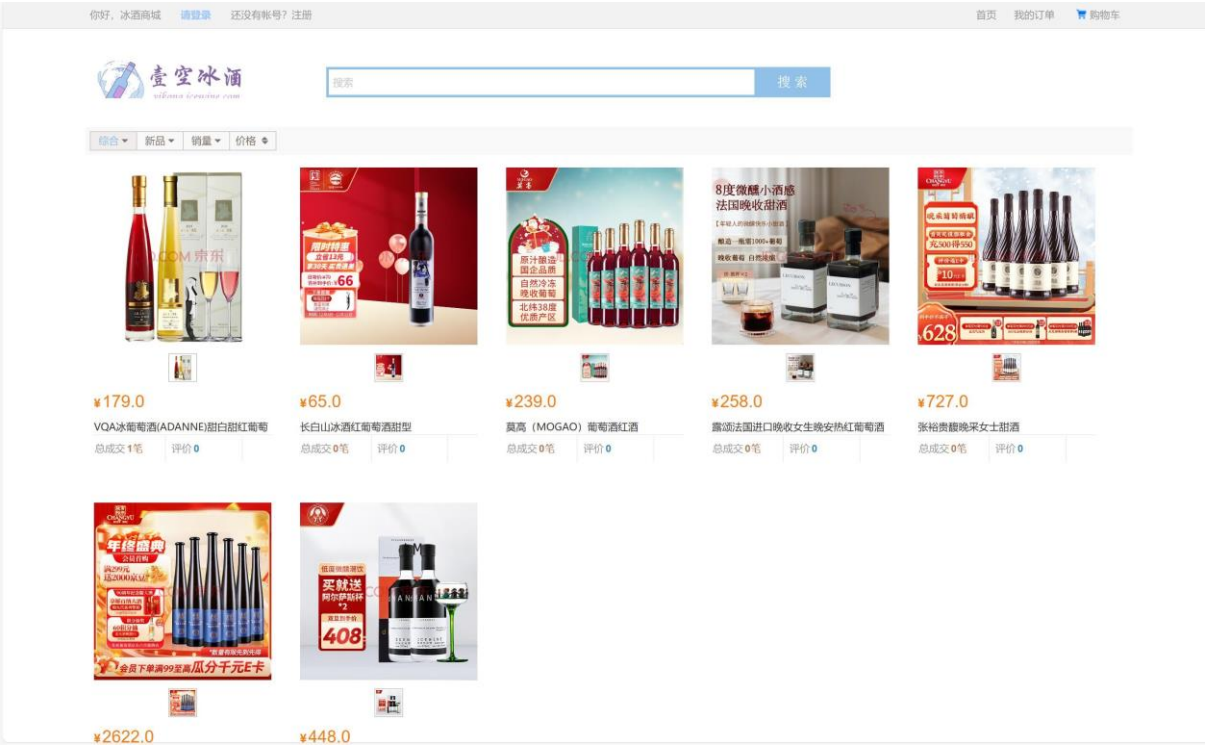
在 搜索 文本框输入“ 玫瑰 ”，点击“搜一下”按钮：



点击搜索框下方导航栏的 “ 白葡萄酒”：

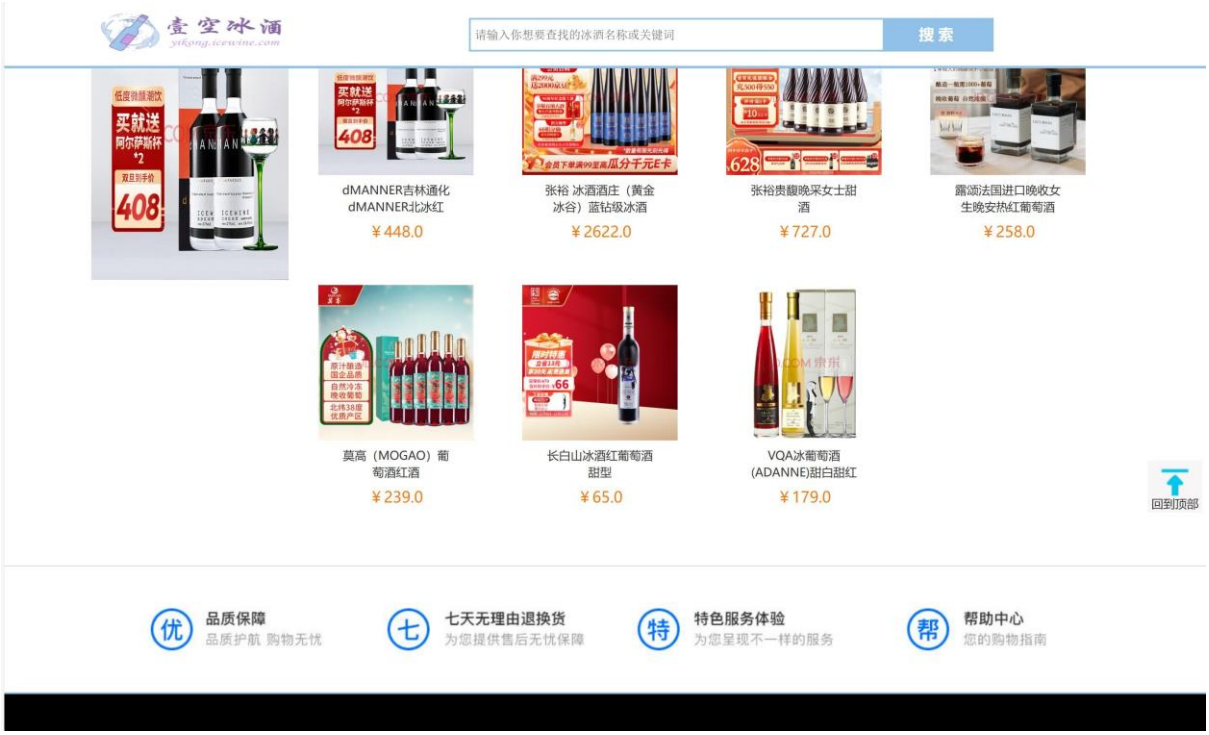


点击搜索框下方导航栏的 “ 红葡萄酒”：

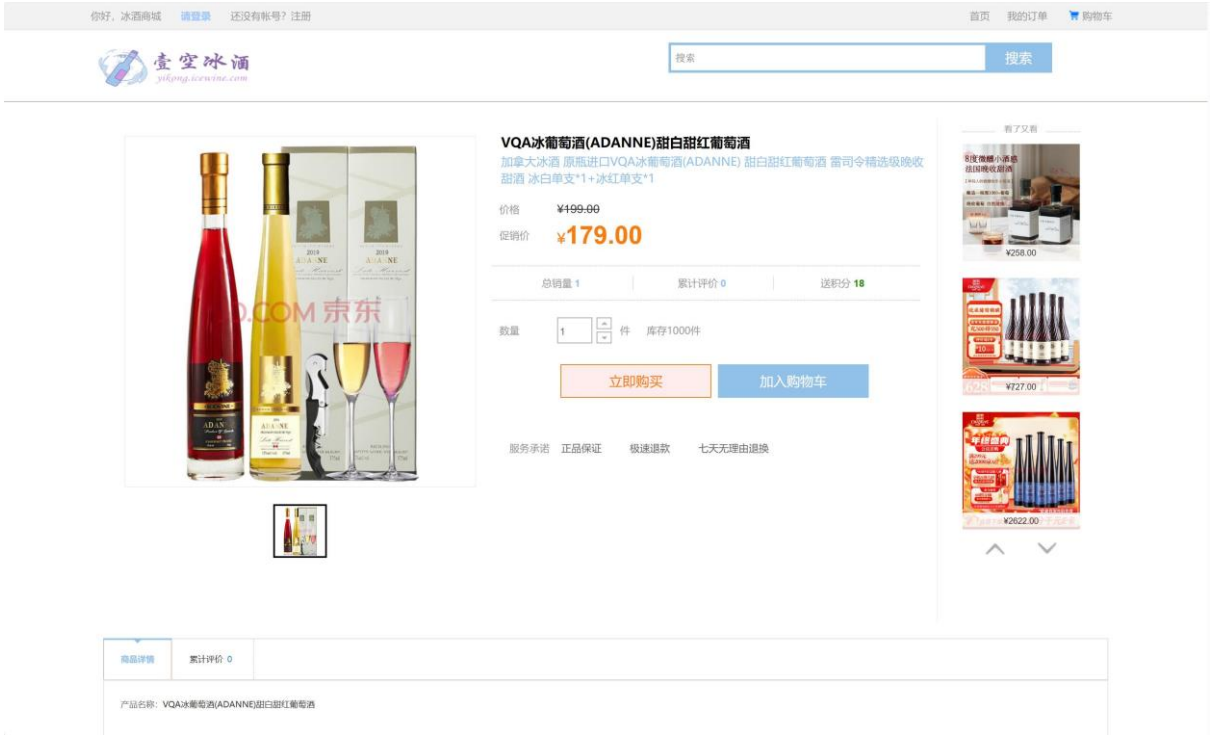


广告栏的图片切换测试：

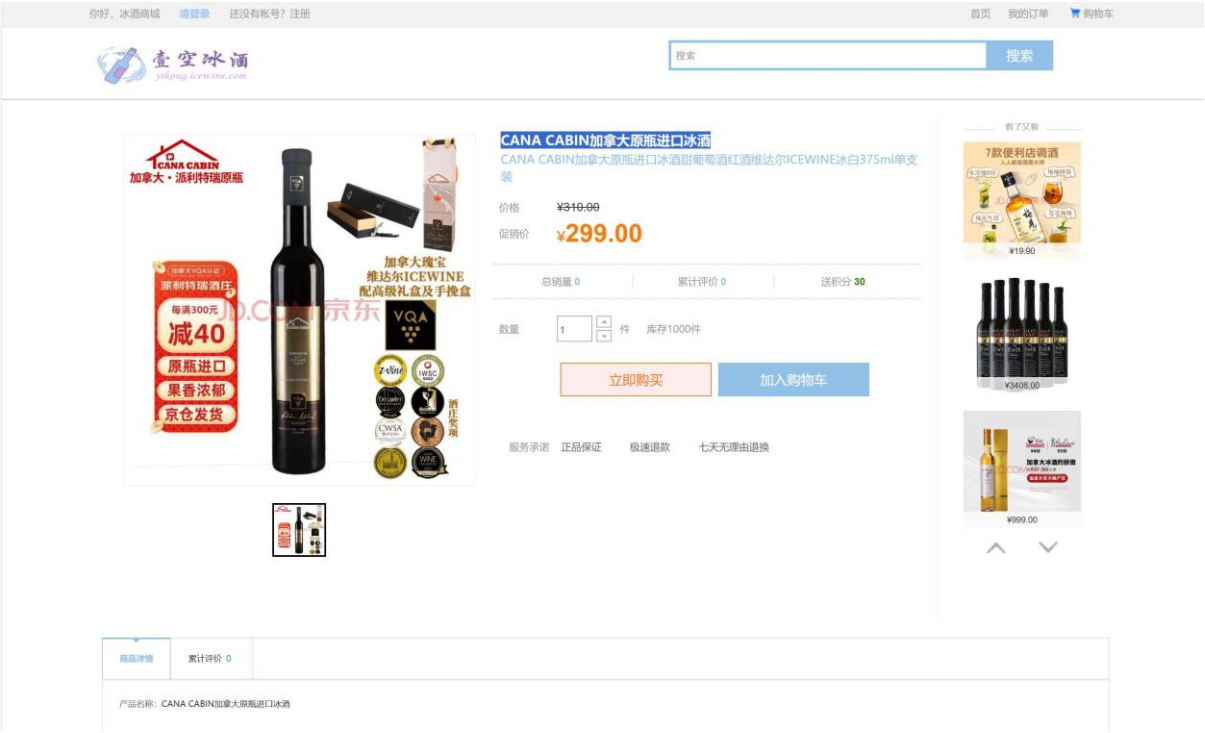
测试“回到顶部”：



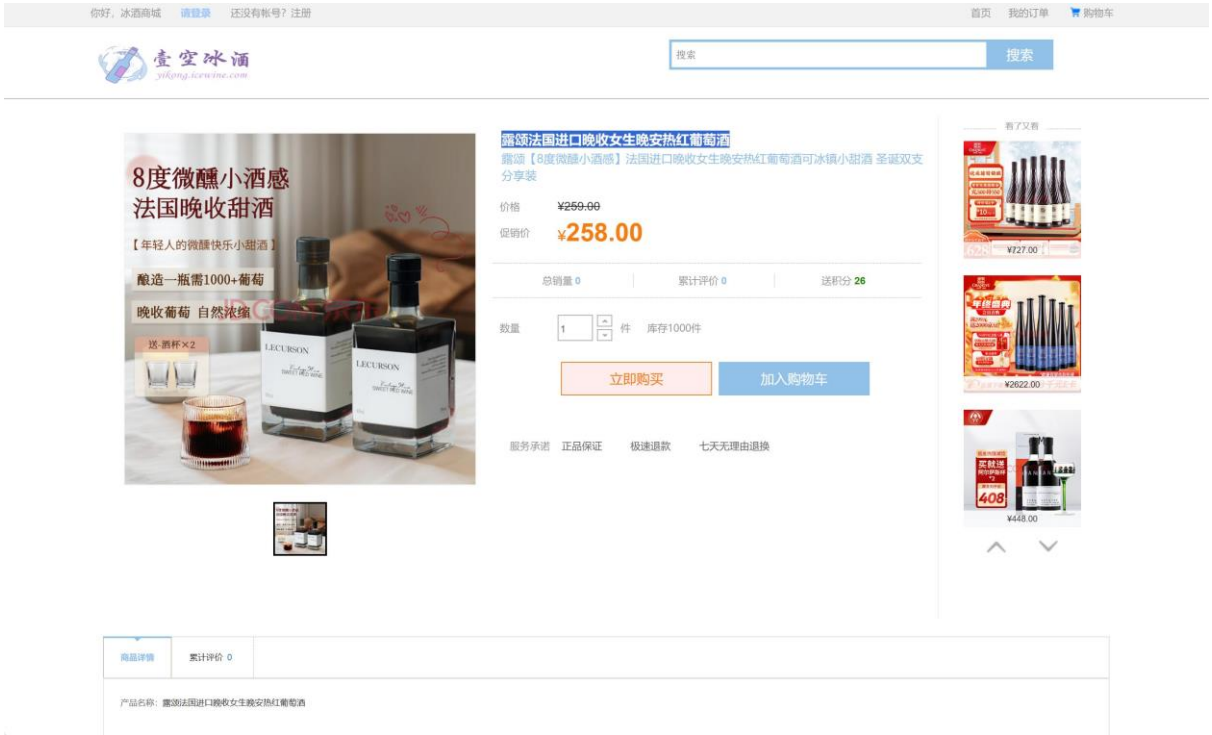
点击首页的“VQA 冰葡萄酒(ADANNE)甜白甜红葡萄酒”，看能否正常跳转



点击首页的“CANA CABIN 加拿大原瓶进口冰酒”，看能否正常跳转



点击首页的“露颂法国进口晚收女生晚安热红葡萄酒”，看能否正常跳转



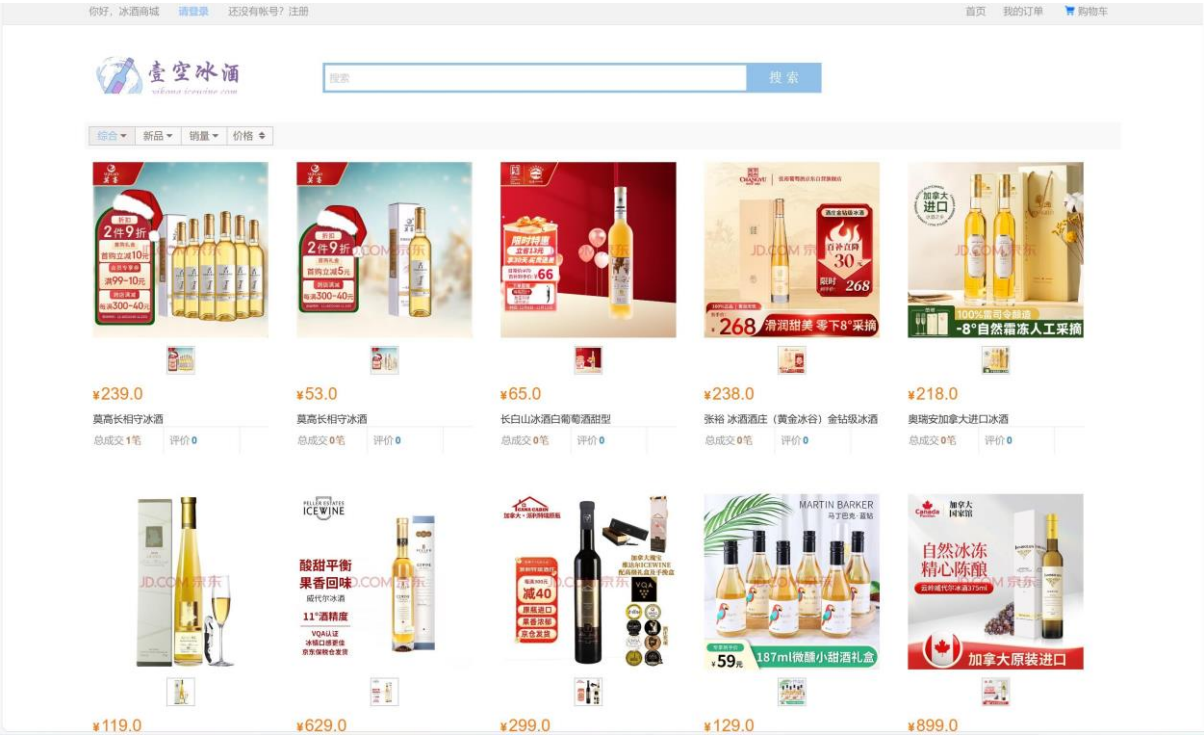
测试结果

序号	测试目的 & 先决条件	测试用例输入	测试结果	备注
1	**目的：**	**用例 1 ：**	显示搜索结果中有且	精确全名搜索

	测试 首页 的“搜索框”操作是否正确	在 搜索 文本框输入“ 梅见【mini 版】青梅酒 ”，点击“搜一下”按钮	仅有“ 梅见【mini 版】青梅酒 ”这一商品	
		用例 2 ： 在 搜索 文本框输入“ 葡萄 ”，点击“搜一下”按钮	显示搜索结果中有所有包含“ 葡萄 ”这一关键字的商品	关键词搜索
		用例 3 ： 在 搜索 文本框输入“ 玫瑰 ”，点击“搜一下”按钮	显示没有找到相关商品	搜索不存在的商品
2	**目的：** 测试 首页 的 导航菜单是否正常跳转	**用例 1 ：** 点击搜索框下方导航栏的 “ 白葡萄酒”	跳转到“白葡萄酒”商品展示界面，显示该种类下所有商品	查看筛选是否准确
		用例 2 ： 点击搜索框下方导航栏的 “ 红葡萄酒”	跳转到“红葡萄酒”商品展示界面，显示该种类下所有商品	
3	**目的：** 测试 首页 的宣传栏是否正常工作，能否切换不同图片	**用例 1 ：** 多次点击 “下一页”按键，看图片能否正常切换	三张图片按照正常顺序切换	
		用例 2： 多次点击 “上一页”按键，看图片能否正常切换		
4	**目的：** 测试 首页的 “回到顶部”按键是否正常工作	**用例 1 ：** 下拉页面，多次滑动到不同位置，点击“回到顶部”，看页面能否回到顶部	点击 ”回到顶部“ 后页面成功回到顶部	
5	**目的：** 测试 首页的 某一 商品能否跳转到对应详情页面	**用例 1 ：** 点击首页的“VQA 冰葡萄酒(ADANNE)甜白甜红葡萄酒”，看能否正常跳转	成功跳转到“VQA 冰葡萄酒(ADANNE)甜白甜红葡萄酒”页面	
		用例 2 ： 点击首页的“CANA CABIN 加拿大原瓶进口冰酒”，看能否正常跳转	成功跳转到“CANA CABIN 加拿大原瓶进口冰酒”页面	

		用例 3 ： 点击首页的“露颂法国进口晚收女生晚安热红葡萄酒”，看能否正常跳转	成功跳转到“露颂法国进口晚收女生晚安热红葡萄酒”页面	

9.2.4.2.2 商品筛选排序测试



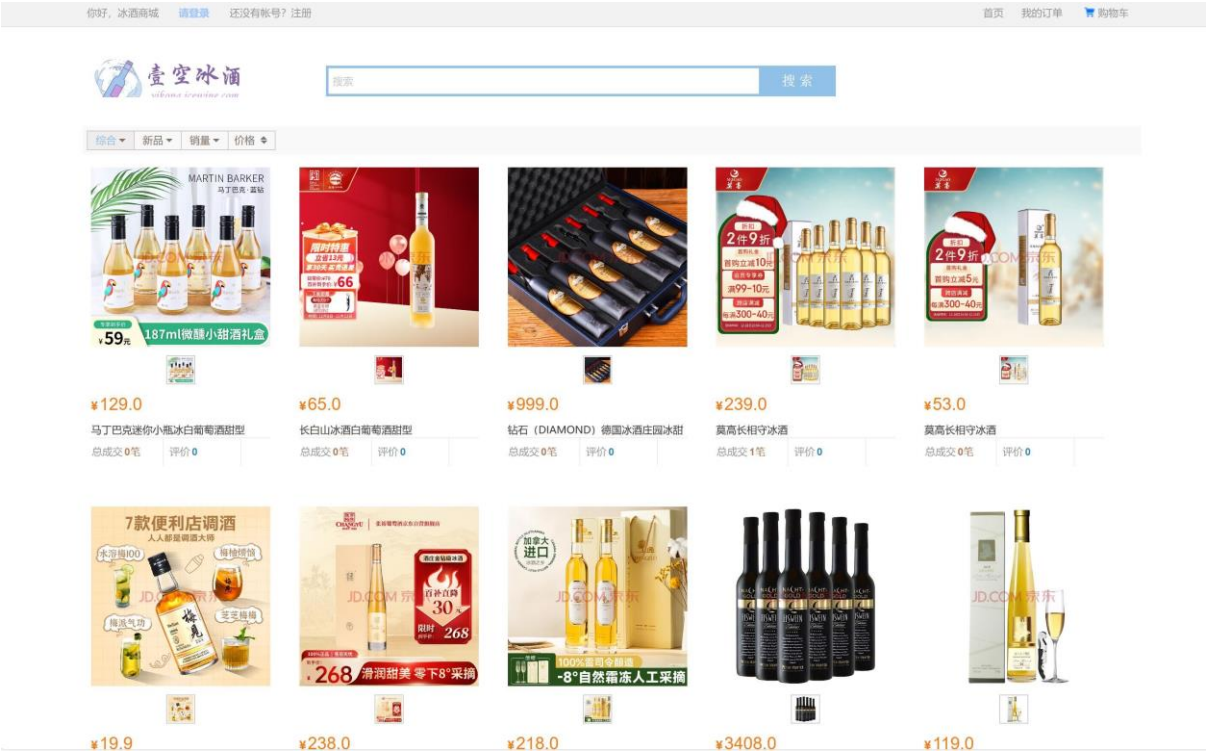
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 按照 “综合” 关键字的降序排序是否正确	**用例 1 ：** 点击“综合”，看按照“综合” 关键字的降序排序是否正确	该界面下所有商品均按照“综合“关键字降序排序	
2	**目的：** 测试 按照 “新品” 关键字的降序排序是否正确	**用例 1 ：** 点击“新品”，看按照“新品” 关键字的降序排序是否正确	该界面下所有商品均按照“新品“关键字降序排序	
3	**目的：** 测试 按照 “价格” 关键字的排序是否正确	**用例 1 ：** 点击“价格”，看按照“价格” 关键字的降序排序是否正确	该界面下所有商品均按照“价格“关键字降序排序	

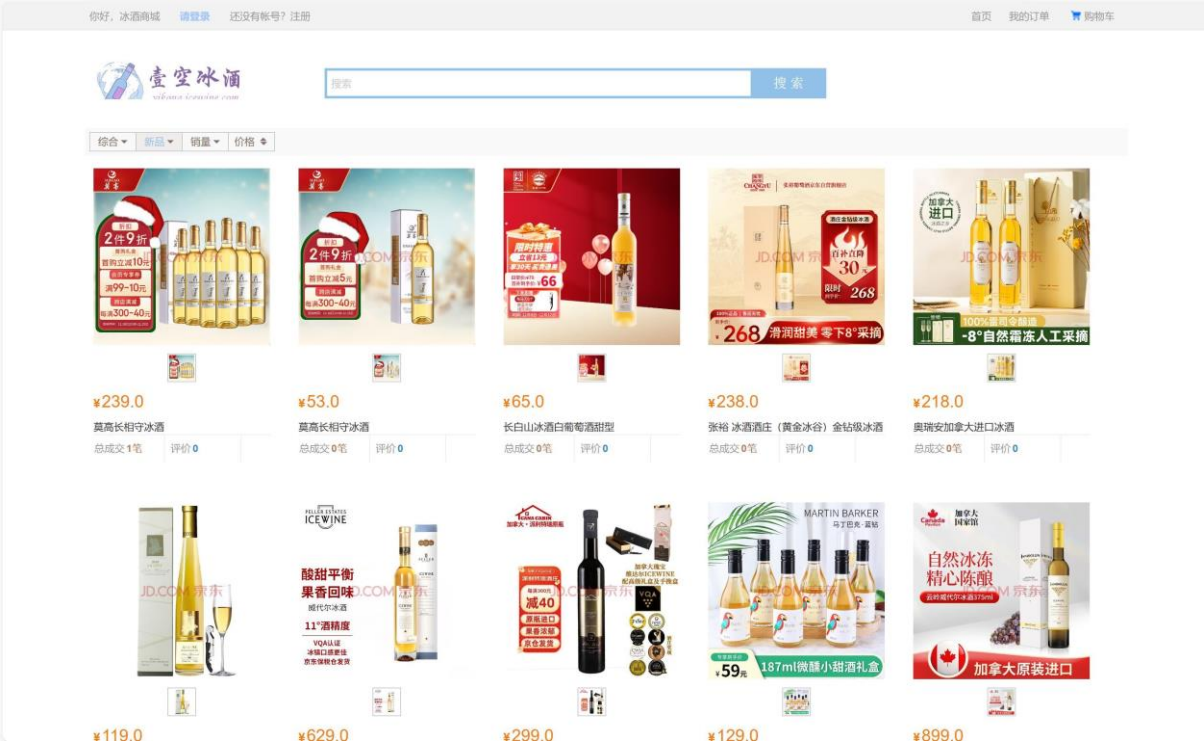
		用例 2： 再次点击“价格”，看按照“价格”关键字的升序排序是否正确	该界面下所有商品均按照“价格“关键字升序排序	
4	**目的：** 测试 按照 “销量” 关键字的降序排序是否正确	**用例 1 ：** 点击“销量”，看按照“销量” 关键字的降序排序是否正确	该界面下所有商品均按照“销量“关键字降序排序	

测试过程

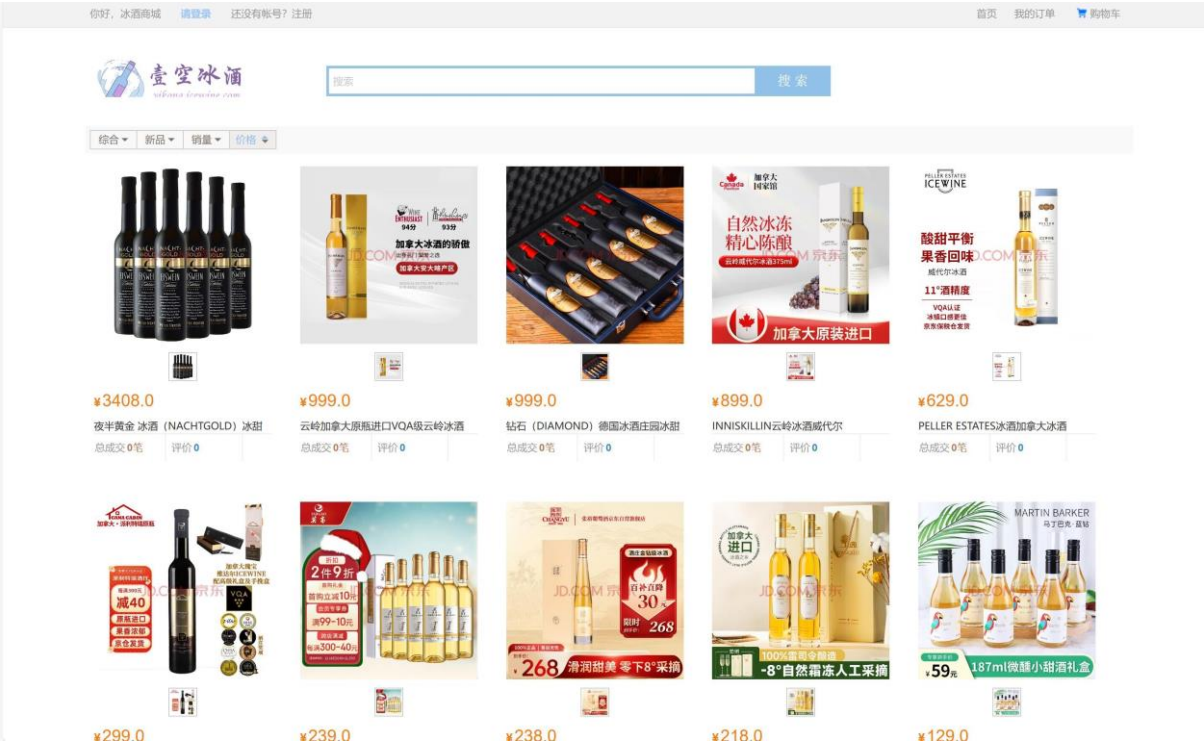
测试 按照 “综合” 关键字的降序排序是否正确



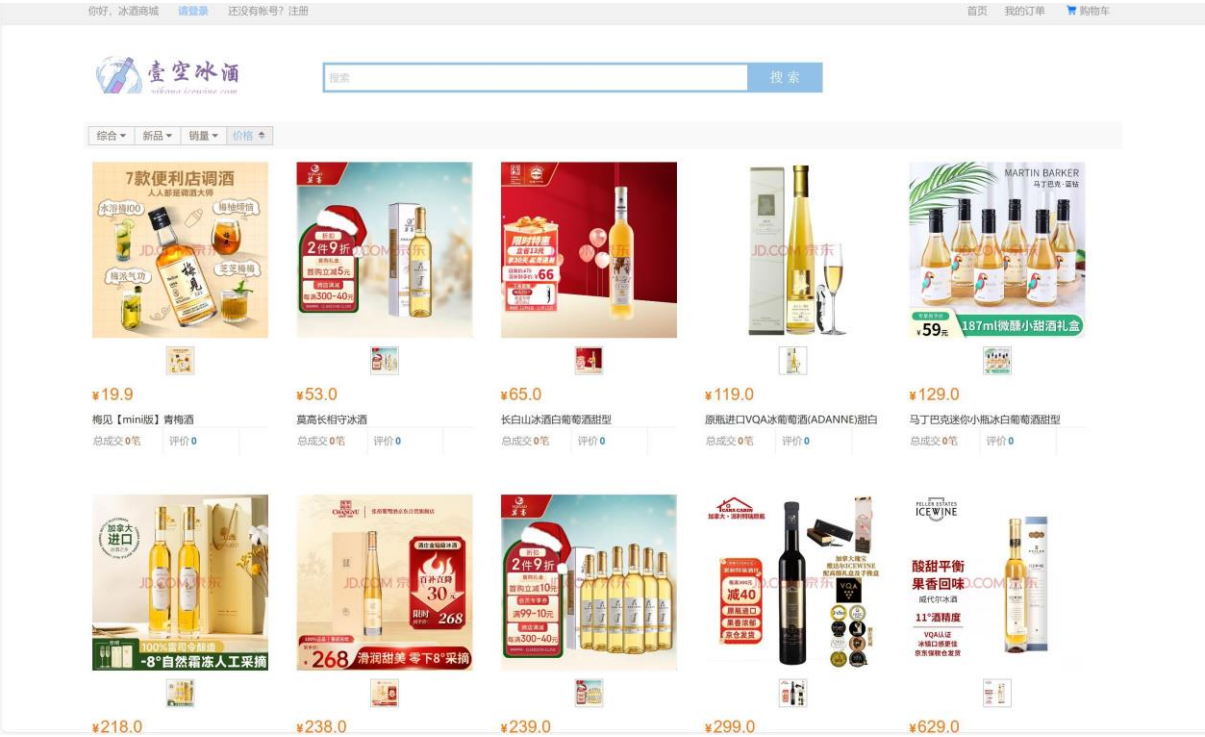
测试 按照 “新品” 关键字的降序排序是否正确



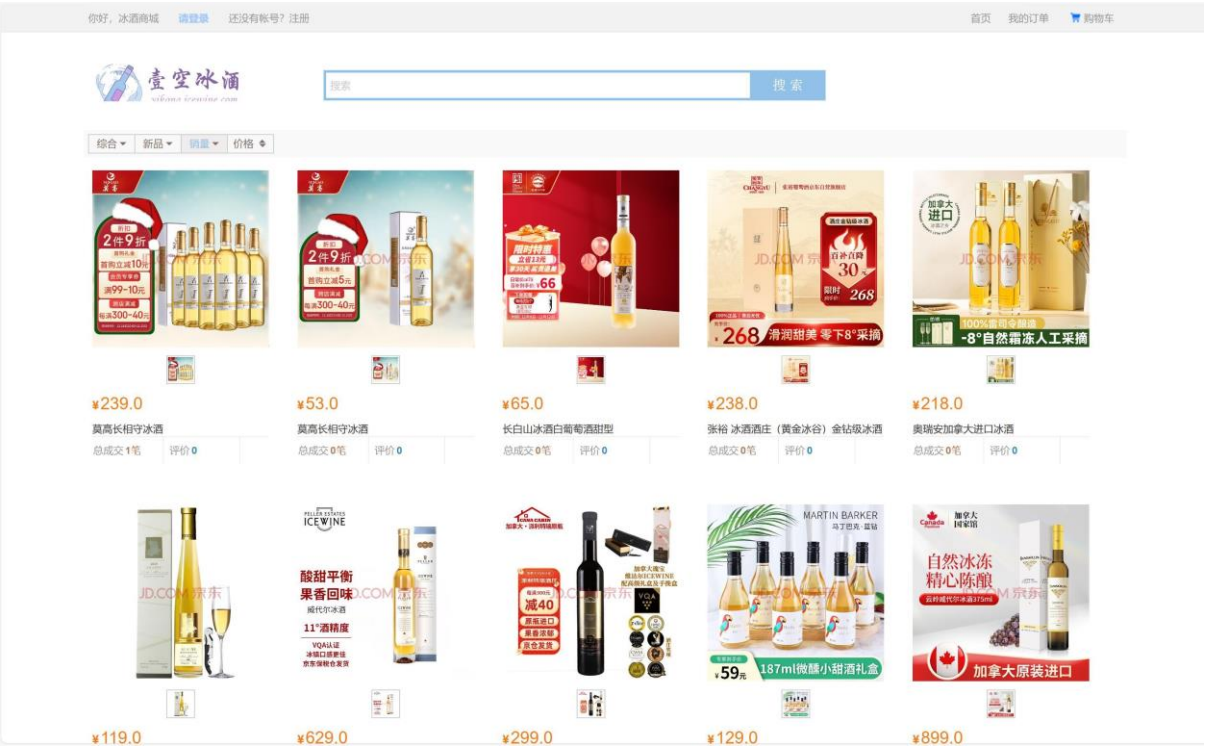
测试 按照 “价格” 关键字的降序排序是否正确



测试 按照 “价格” 关键字的升序排序是否正确



测试 按照 “销量” 关键字的降序排序是否正确



测试结果

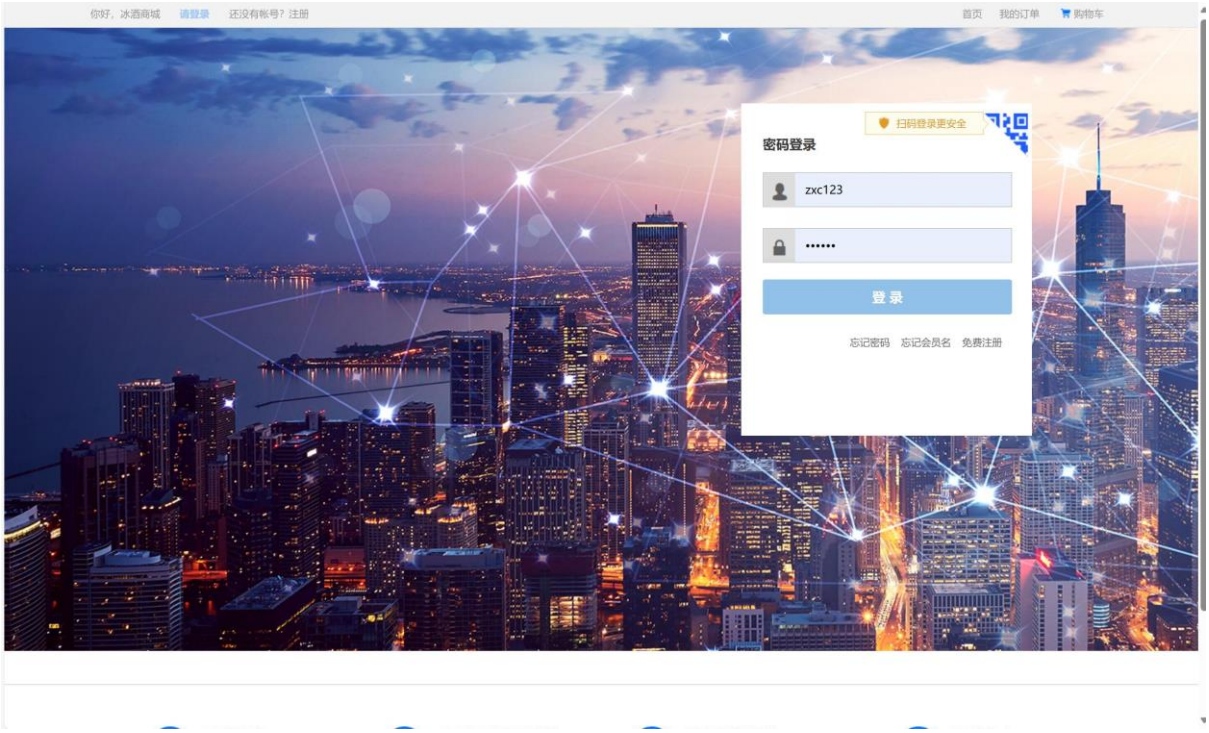
序号	测试目的 & 先决条件	测试用例输入	测试结果	备注

1	**目的：** 测试 按照 “综合” 关键字的降序排序是否正确	**用例 1 ：** 点击“综合”，看按照“综合” 关键字的降序排序是否正确	该界面下所有商品均按照“综合“关键字降序排序	
2	**目的：** 测试 按照 “新品” 关键字的降序排序是否正确	**用例 1 ：** 点击“新品”，看按照“新品” 关键字的降序排序是否正确	该界面下所有商品均按照“新品“关键字降序排序	
3	**目的：** 测试 按照 “价格” 关键字的排序是否正确	**用例 1 ：** 点击“价格”，看按照“价格” 关键字的降序排序是否正确	该界面下所有商品均按照“价格“关键字降序排序	
		用例 2： 再次点击“价格”，看按照“价格”关键字的升序排序是否正确	该界面下所有商品均按照“价格“关键字升序排序	
4	**目的：** 测试 按照 “销量” 关键字的降序排序是否正确	**用例 1 ：** 点击“销量”，看按照“销量” 关键字的降序排序是否正确	该界面下所有商品均按照“销量“关键字降序排序	

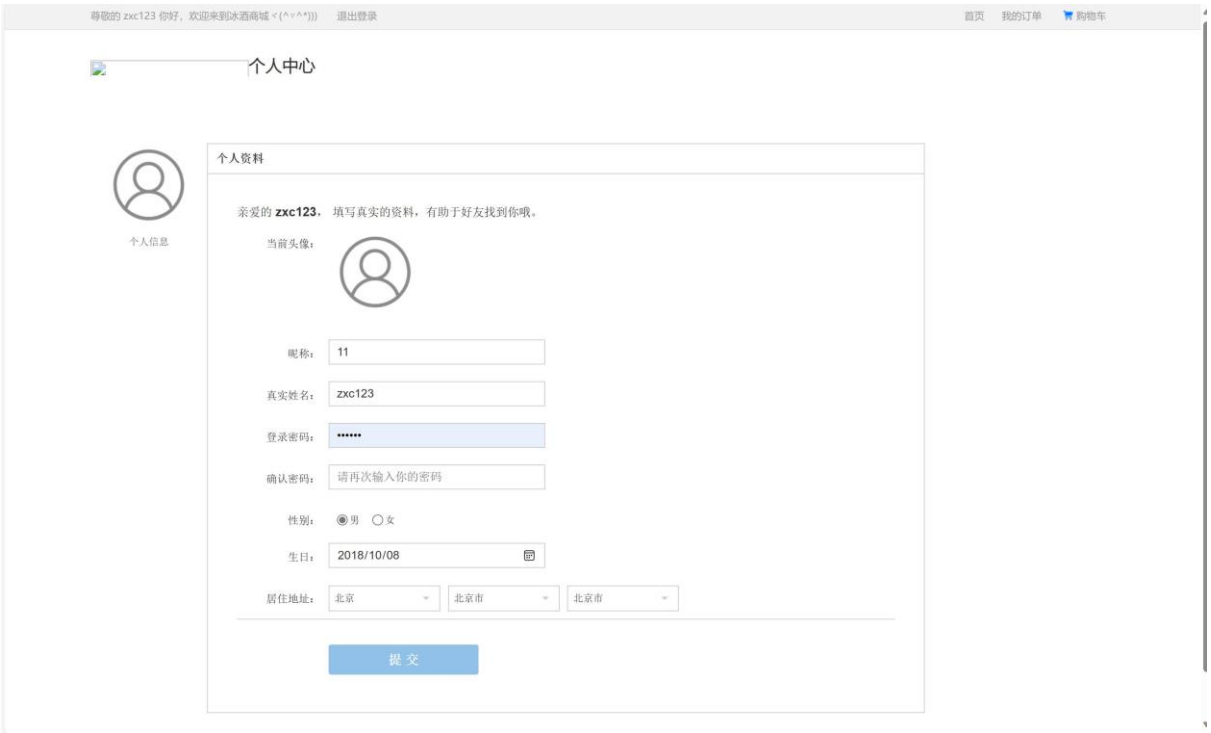
9.2.4.2.3 用户账户管理测试

注册

登录


你好，冰酒商城
请登录
还没有帐号？注册
首页
我的订单
购物车
扫码登录更安全
密码登录
zxc123
登 录
忘记密码
忘记会员名
免费注册

管理


尊敬的 zxc123 你好，欢迎来到冰酒商城 <(^▽^*)))
退出登录
首页
我的订单
购物车
个人中心
个人信息
个人资料
亲爱的 zxc123，填写真实的资料，有助于好友找到你哦。
当前头像：
昵称：
11
真实姓名：
zxc123
登录密码：
确认密码：
请再次输入你的密码
性别：
男
女
生日：
2018/10/08
居住地址：
北京
北京市
北京市
提 交

测试用例清单

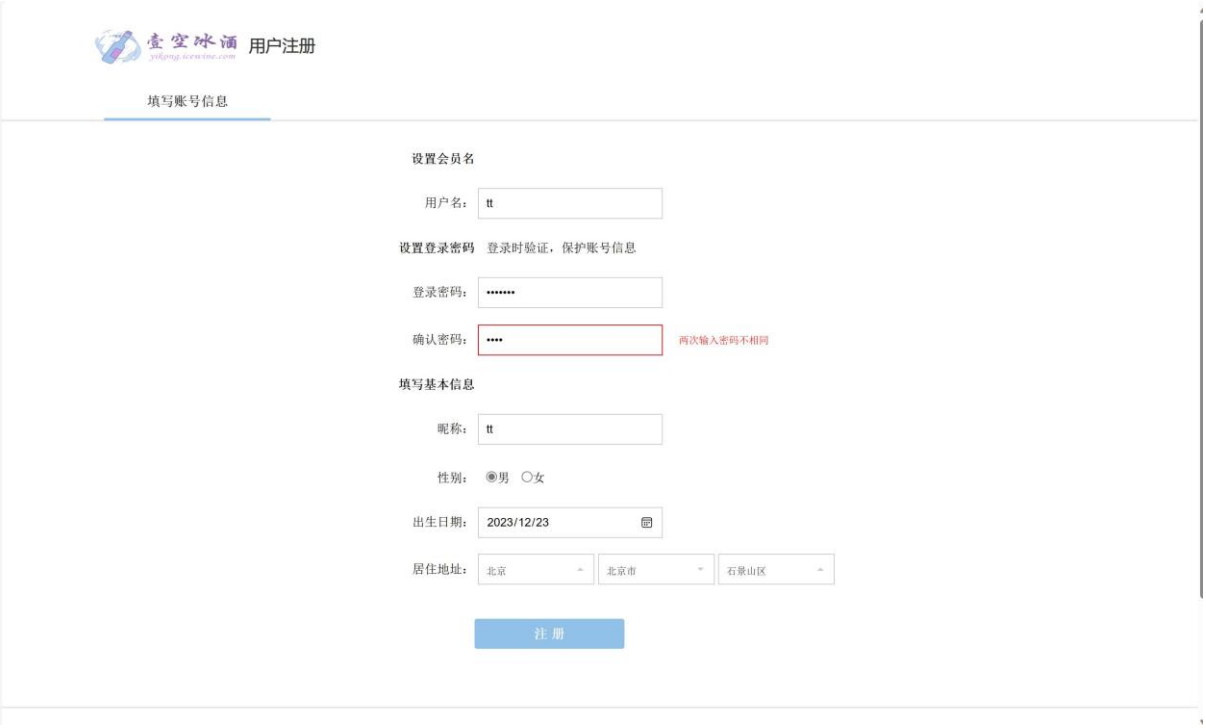
序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 ”注册“ 页面 能否正常注册，包括校验登陆密码的合理性，确认密码的检验等	**用例 1 ：** 在 ”登录密码“ 文本框输入 “root”（不合法密码），填写完其他信息后，点击 “注册”	“登陆密码”那一栏后面显示文字提醒“密码格式必须包含数字和字母”	密码合理性校验码
		用例 2 ： 在 ”登录密码“ 文本框输入 “root123”（不合法密码），填写完其他信息后，点击 “注册”	无报错提醒	
		用例 3 ： 在 ”登录密码“ 文本框输入 “root”（和登陆密码不同），填写完其他信息后，点击 “注册”	“确认密码”那一栏后面显示文字提醒“两次输入密码不相同”	密码确认合理性校验
		用例 4 ： 在 ”登录密码“ 文本框输入 “root”（和登陆密码不同），填写完其他信息后，点击 “注册”	无报错提醒	
		用例 5 ： 所有信息均合法且填写完毕后，点击“注册”	无报错提醒	测试能否正常注册
2	**目的：** 测试 ”登录“ 页面 能否正常登录	**用例 1 ：** 用户文本框输入 “zxc123”，密码输入 “zxc1233”(错误密码)，点击登录	提示“用户名和密码错误”	查看登录校验功能是否准确
		用例 2 ： 用户文本框输入 “zxc123”，密码输入 “zxc123”(正确密码)，点击登录	成功登录，跳转到商品首页	
3	**目的：** 测试 个人中心的账号信息管理功能	**用例 1 ：** 修改真实姓名为 “zxc1233”，查看是否修改成功	真实姓名成功修改为“zxc123”	注意每次修改个人信息都要重新输入登陆密码

测试过程

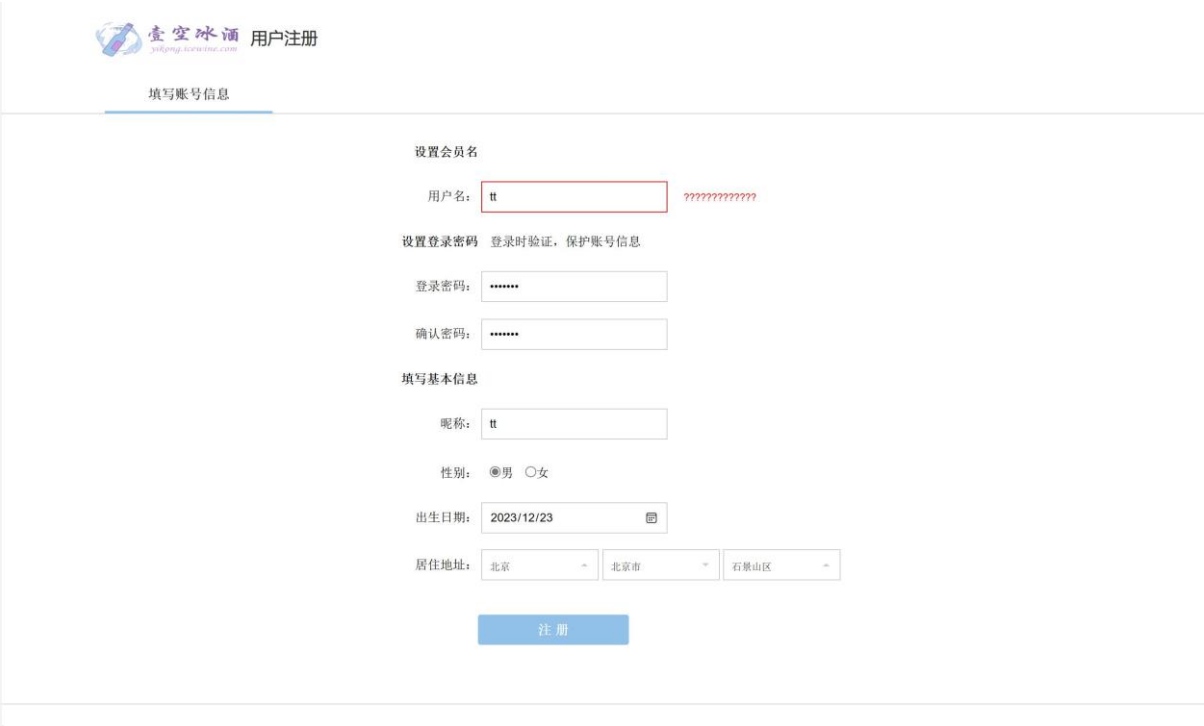
密码合理性校验

壹空冰酒 用户注册
填写账号信息
设置会员名
用户名：
tt
设置登录密码
登录时验证，保护账号信息
登录密码：
密码格式必须包含数字和字母
确认密码：
填写基本信息
昵称：
tt
性别：
男
女
出生日期：
2023/12/23
居住地址：
北京
北京市
石景山区
注 册

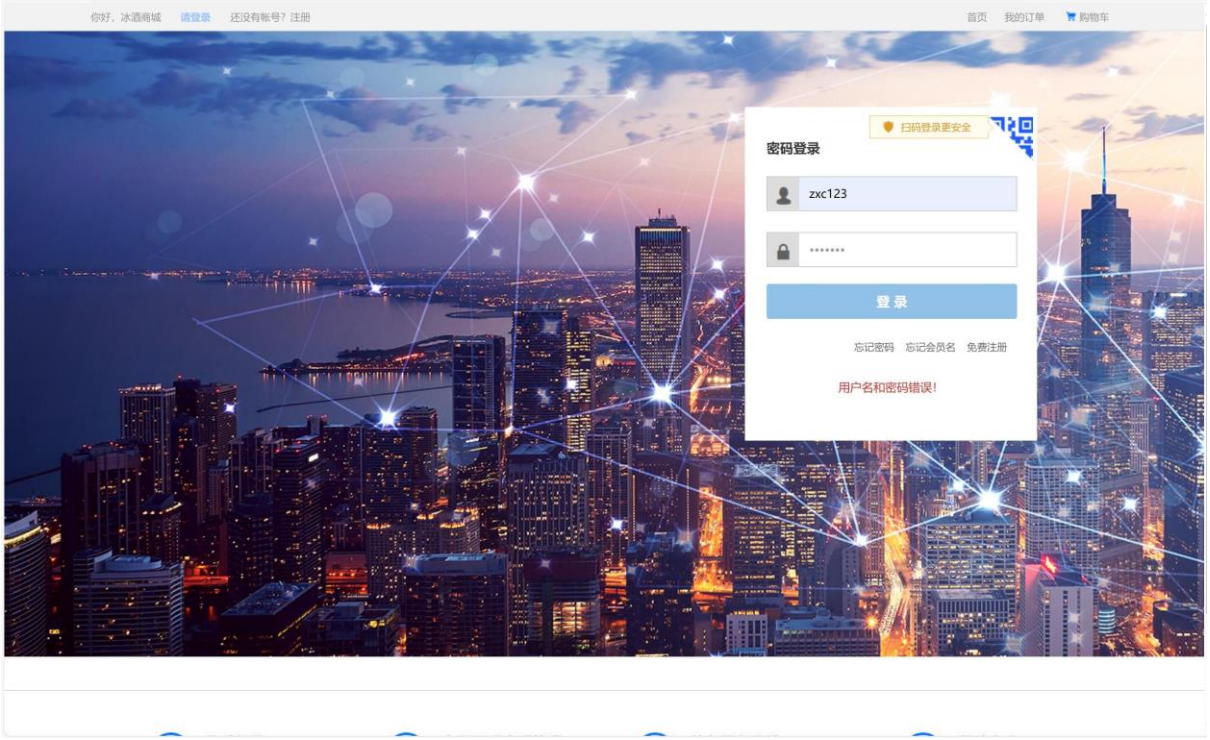
密码确认校验

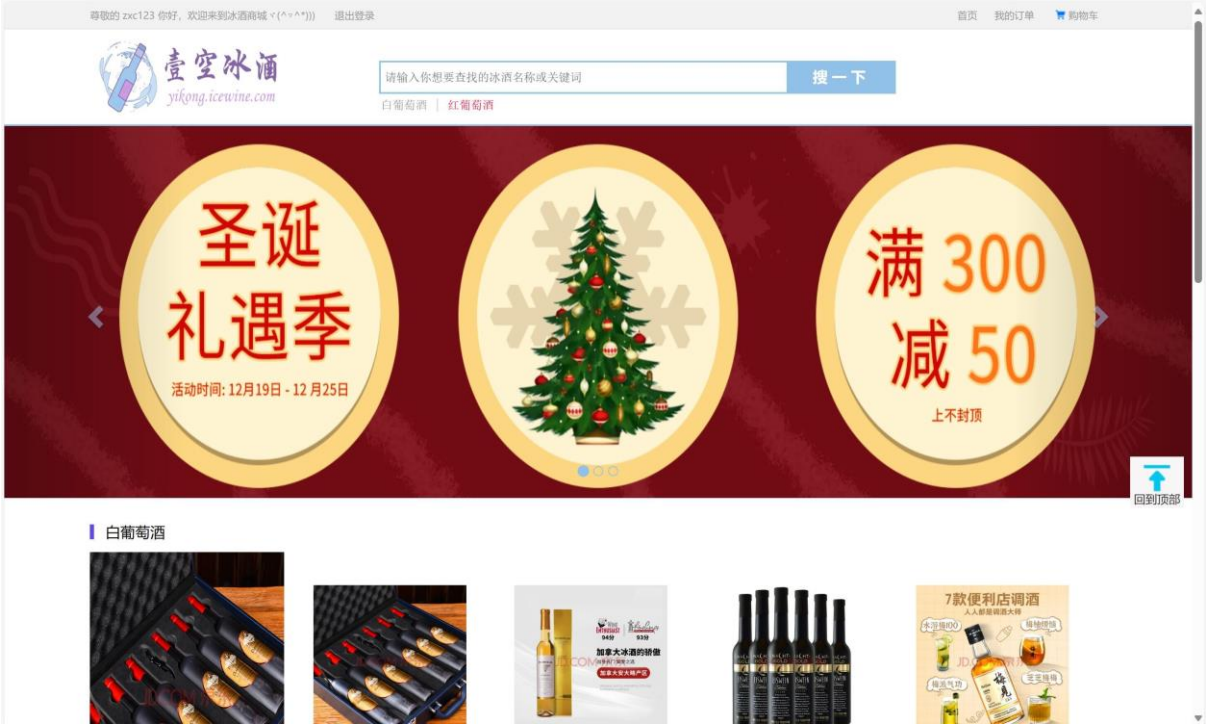

壹空冰酒 用户注册
填写账号信息
设置会员名
用户名：
tt
设置登录密码
登录时验证，保护账号信息
登录密码：
确认密码：
两次输入密码不相同
填写基本信息
昵称：
tt
性别：
男
女
出生日期：
2023/12/23
居住地址：
北京
北京市
石景山区
注 册

注册成功：


壹空冰酒 用户注册
填写账号信息
设置会员名
用户名：
tt
????????????
设置登录密码
登录时验证，保护账号信息
登录密码：
确认密码：
填写基本信息
昵称：
tt
性别：
男
女
出生日期：
2023/12/23
居住地址：
北京
北京市
石景山区
注 册

登录校验


你好，冰酒商城
请登录
还没有帐号？注册
首页
我的订单
购物车
扫码登录更安全
密码登录
zxc123
登 录
忘记密码
忘记会员名
免费注册
用户名和密码错误！



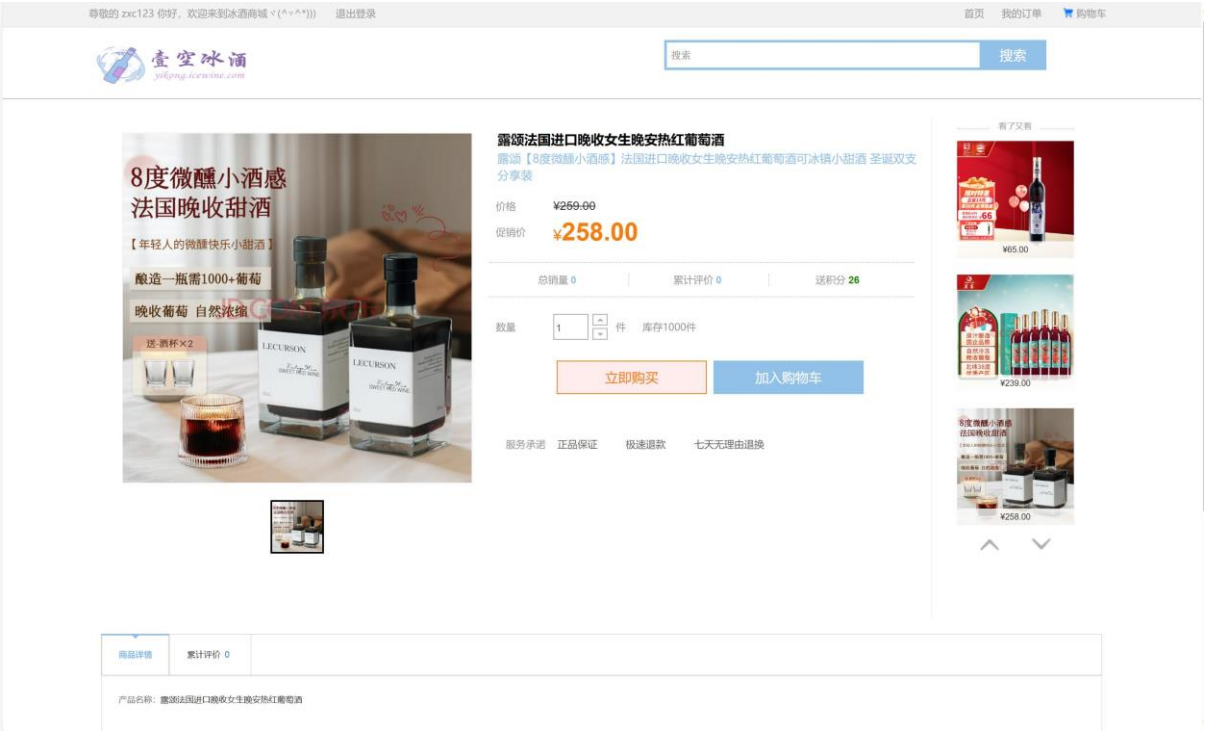
信息修改

测试结果

序号	测试目的 & 先决条件	测试用例输入	测试结果	备注
1	**目的：**	**用例 1 ：**	“登陆密码”那一栏	密码合理性校验

	测试 ”注册“ 页面 能否正常注册，包括校验登陆密码的合理性，确认密码的检验等	在 ”登录密码“ 文本框输入 “root”（不合法密码），填写完其他信息后，点击 “注册”	后面显示文字提醒“密码格式必须包含数字和字母”	码
		用例 2 ： 在 ”登录密码“ 文本框输入 “root123”（不合法密码），填写完其他信息后，点击 “注册”	无报错提醒	
		用例 3 ： 在 ”登录密码“ 文本框输入 “root”（和登陆密码不同），填写完其他信息后，点击 “注册”	“确认密码”那一栏后面显示文字提醒“两次输入密码不相同”	密码确认合理性校验
		用例 4 ： 在 ”登录密码“ 文本框输入 “root”（和登陆密码不同），填写完其他信息后，点击 “注册”	无报错提醒	
		用例 5 ： 所有信息均合法且填写完毕后，点击“注册”	注册成功没有提示	bug 已修复
2	**目的：** 测试 ”登录“ 页面 能否正常登录	**用例 1 ：** 用户文本框输入“zxc123”，密码输入“zxc1233”(错误密码)，点击登录	提示“用户名和密码错误”	查看登录校验功能是否准确
		用例 2 ： 用户文本框输入“zxc123”，密码输入“zxc123”(正确密码)，点击登录	成功登录，跳转到商品首页	
3	**目的：** 测试 个人中心的账号信息管理功能	**用例 1 ：** 修改真实姓名为“zxc1233”，查看是否修改成功	没法修改	bug 已修复

9.2.4.2.4 **商品详情页测试**

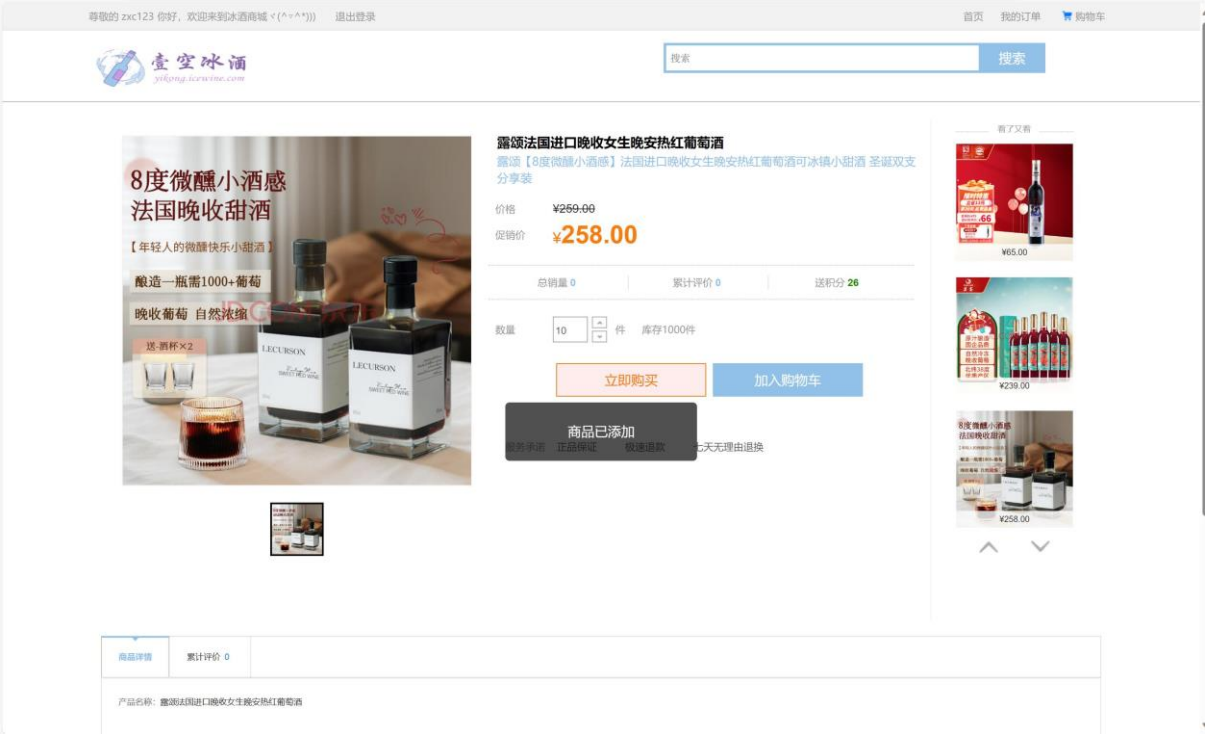


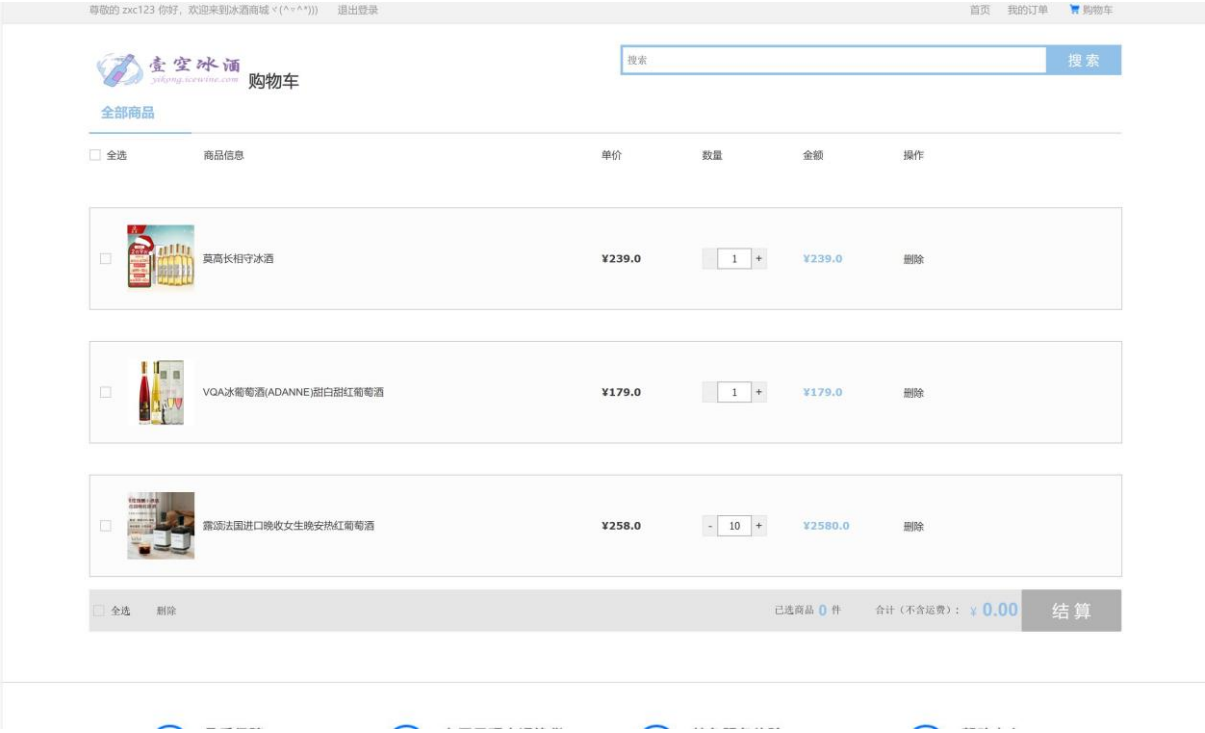
测试用例清单

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 商品详情页 的“加入购物车”操作是否正确	**用例 1 ：** 将商品数量设置为 10 件，然后点击“加入购物车”，去购物车界面查看是否加入成功	在购物车界面查看到该商品有 10 件	首次加入
		用例 2 ： 将商品数量设置为 10 件，然再次点击“加入购物车”，去购物车界面查看是否加入成功	在购物车界面查看到该商品有 20 件	再次加入
2	**目的：** 测试 商品详情页 的“立即购买”操作是否正确	**用例 1 ：** 点击“立即购买”界面，跳转到该商品的支付界面	跳转到该商品的支付界面	是否能直接购买
3	**目的：** 测试 商品详情页 的“看了又看”能否链接到其他商品页面	**用例 1：** 点击右侧 “看了又看” 按键的图片，看能否跳转到对应画面	跳转到对应的商品对应的页面	

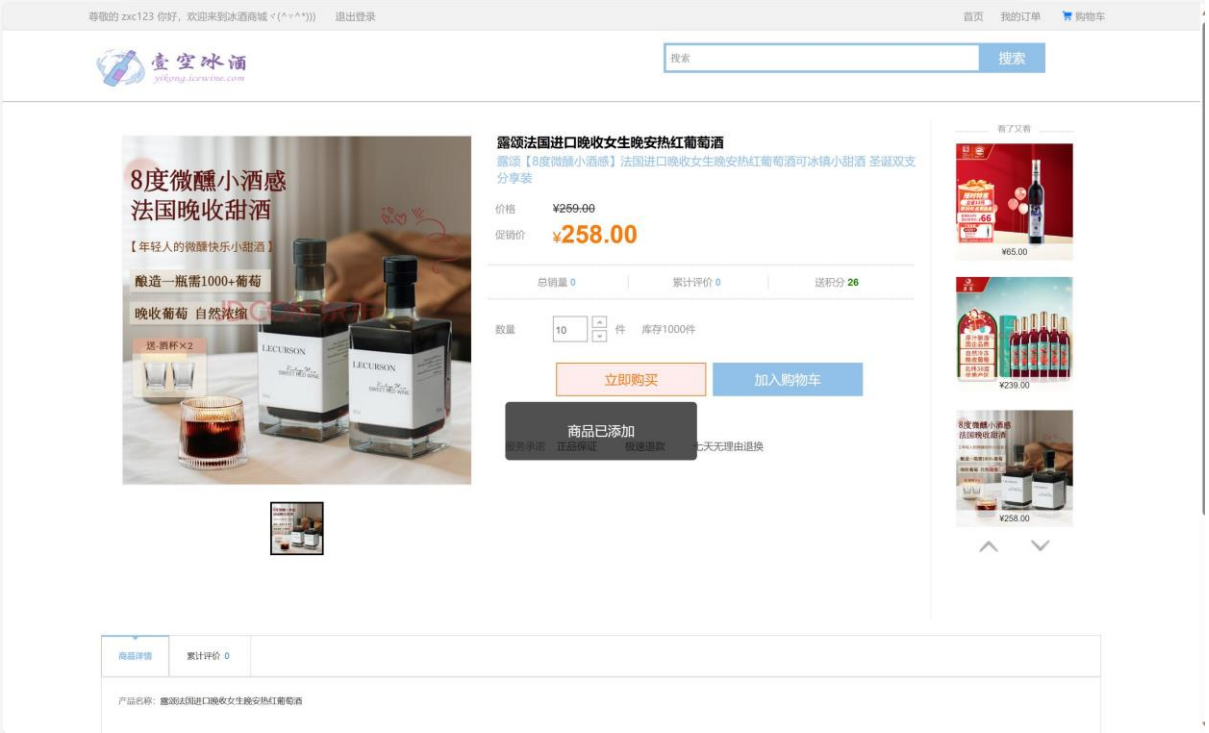
测试过程

首次加入 10 件

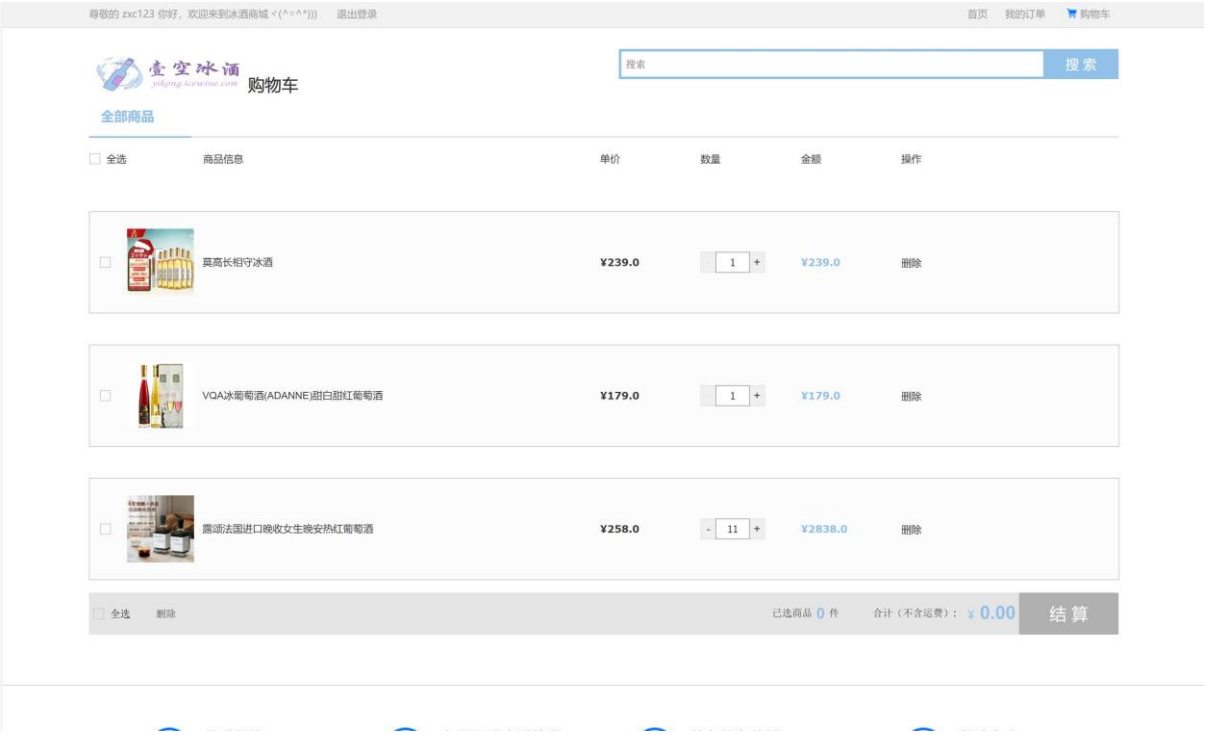




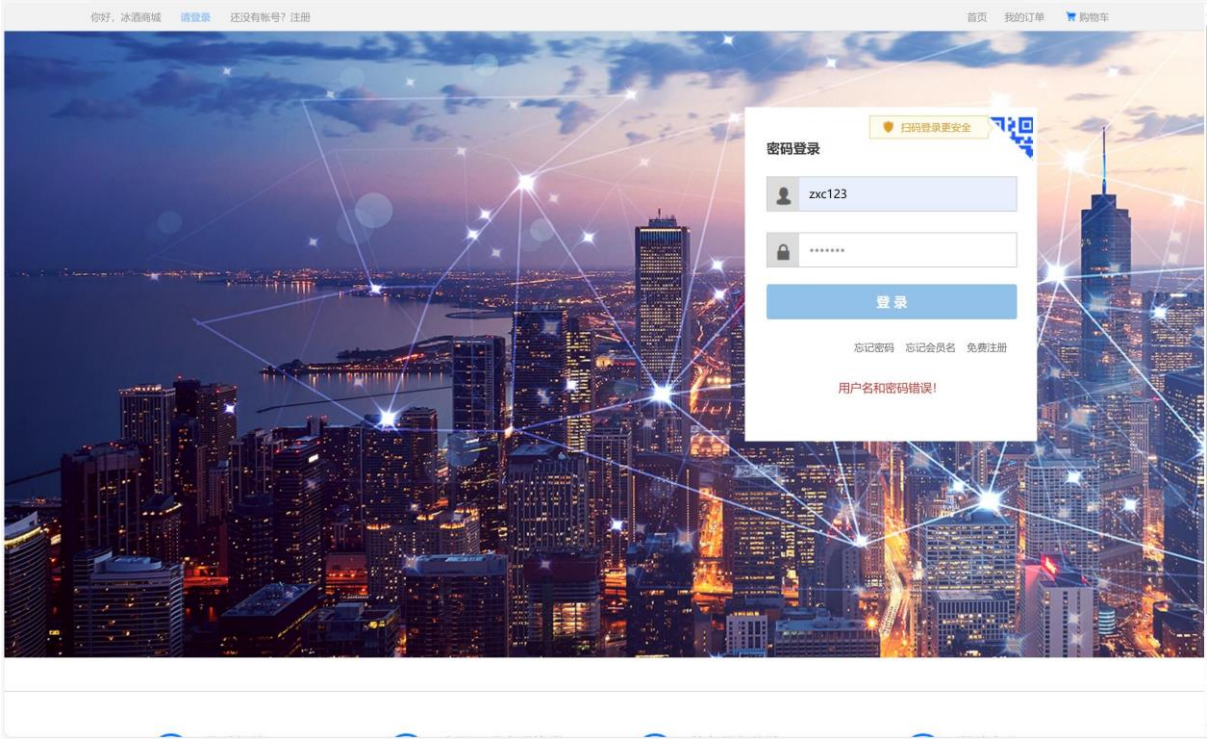
再次加入 10 件



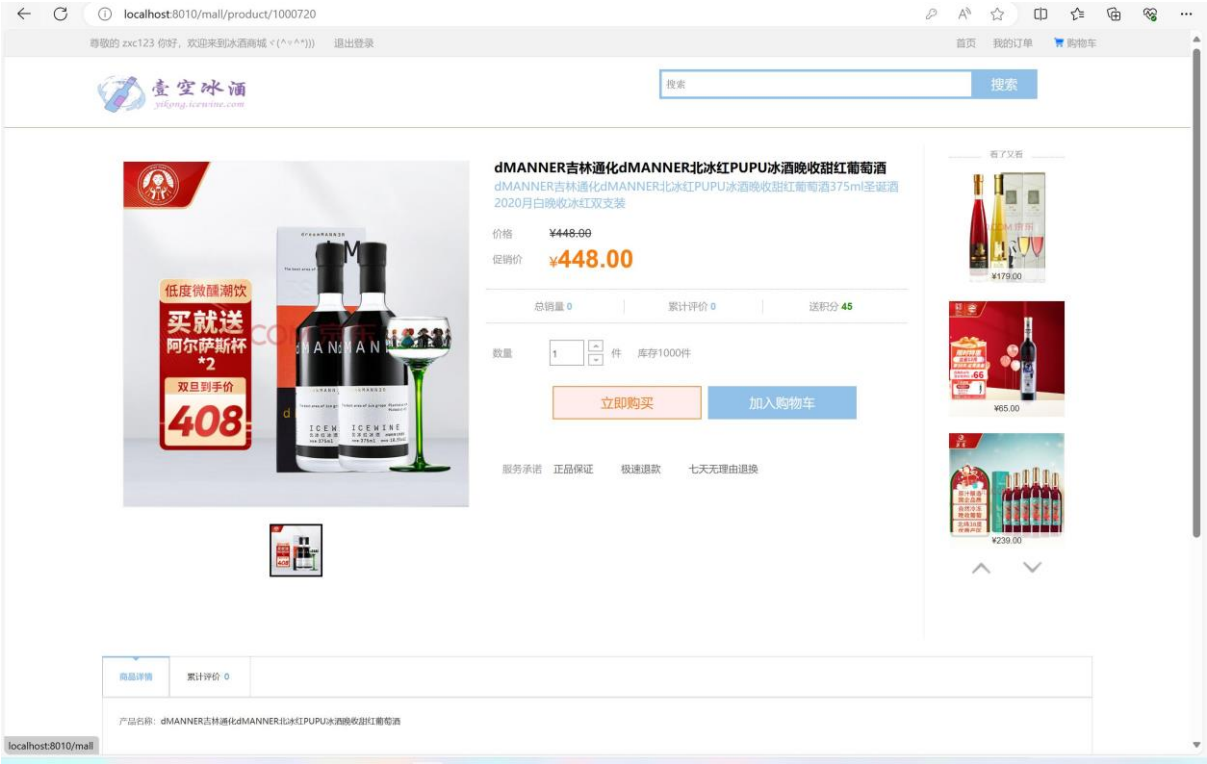
但是只显示多加了 1 件



点击立即购买会跳转到这里



“看了又看”的跳转

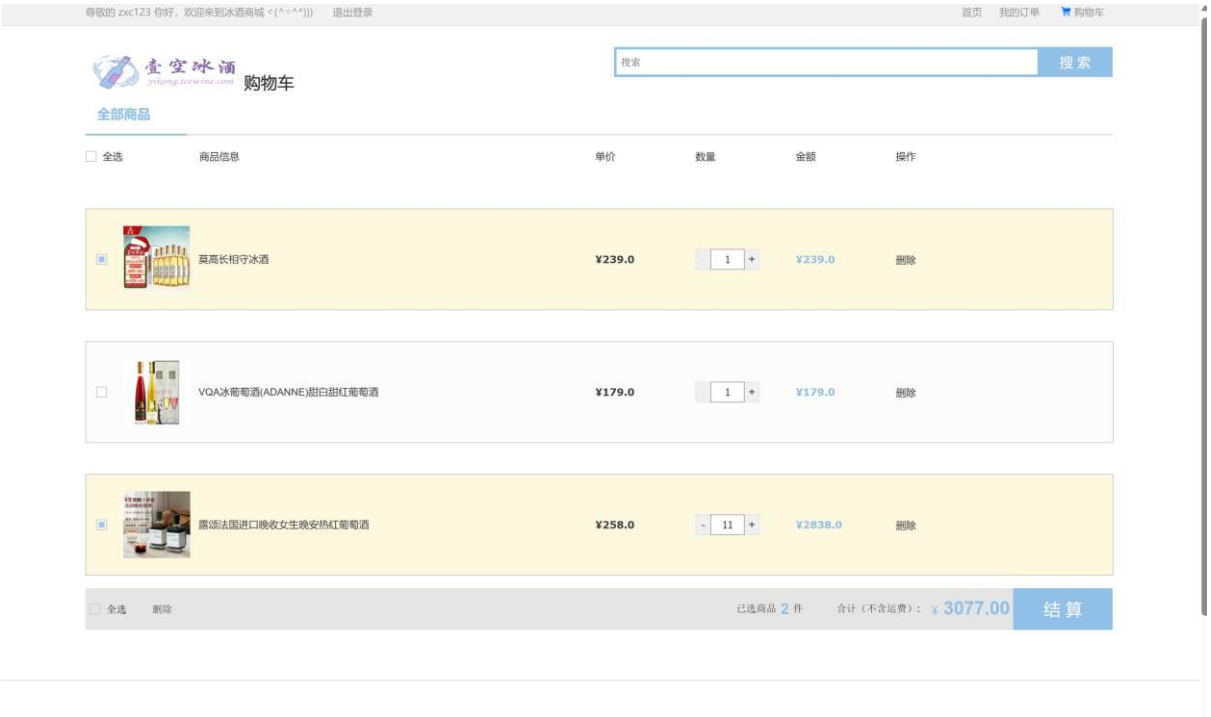


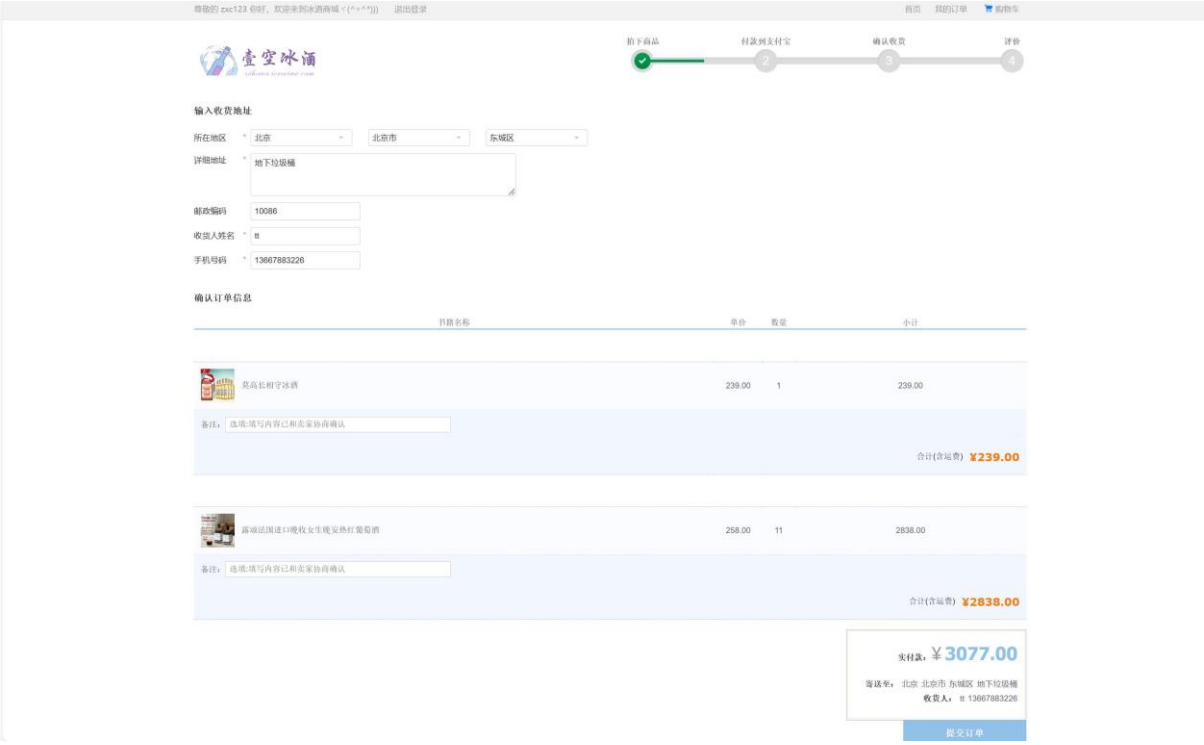
测试结果

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注

1	**目的：** 测试 商品详情页 的“加入购物车”操作是否正确	**用例 1 ：** 将商品数量设置为 10 件，然后点击“加入购物车”，去购物车界面查看是否加入成功	在购物车界面查看到该商品有 10 件	
		用例 2 ： 将商品数量设置为 10 件，然再次点击“加入购物车”，去购物车界面查看是否加入成功	在购物车界面查看到该商品只有 11 件	bug 已修复
2	**目的：** 测试 商品详情页 的“立即购买”操作是否正确	**用例 1 ：** 点击“立即购买”界面，跳转到该商品的支付界面	跳转到了登陆界面	bug 已修复
3	**目的：** 测试 商品详情页 的“看了又看”能否链接到其他商品页面	**用例 1：** 点击右侧 “看了又看” 按键的图片，看能否跳转到对应画面	跳转到对应的商品对应的页面	

9.2.4.2.5 **购物车测试**



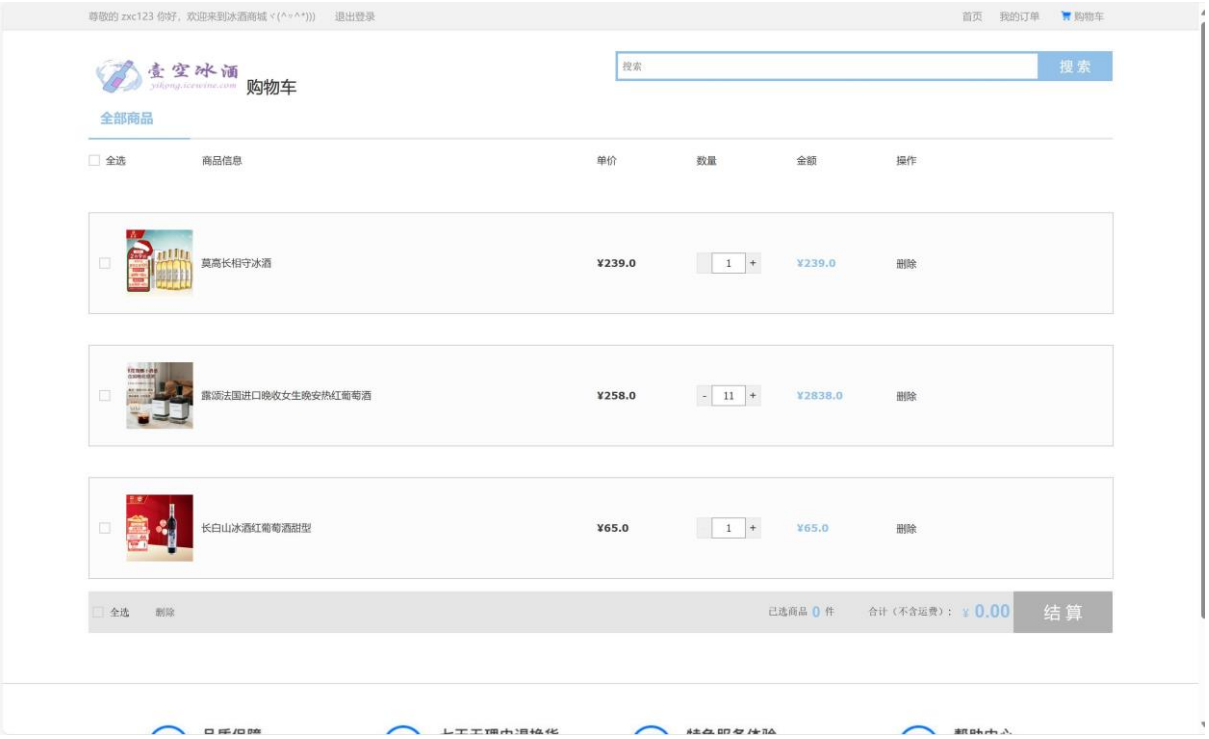


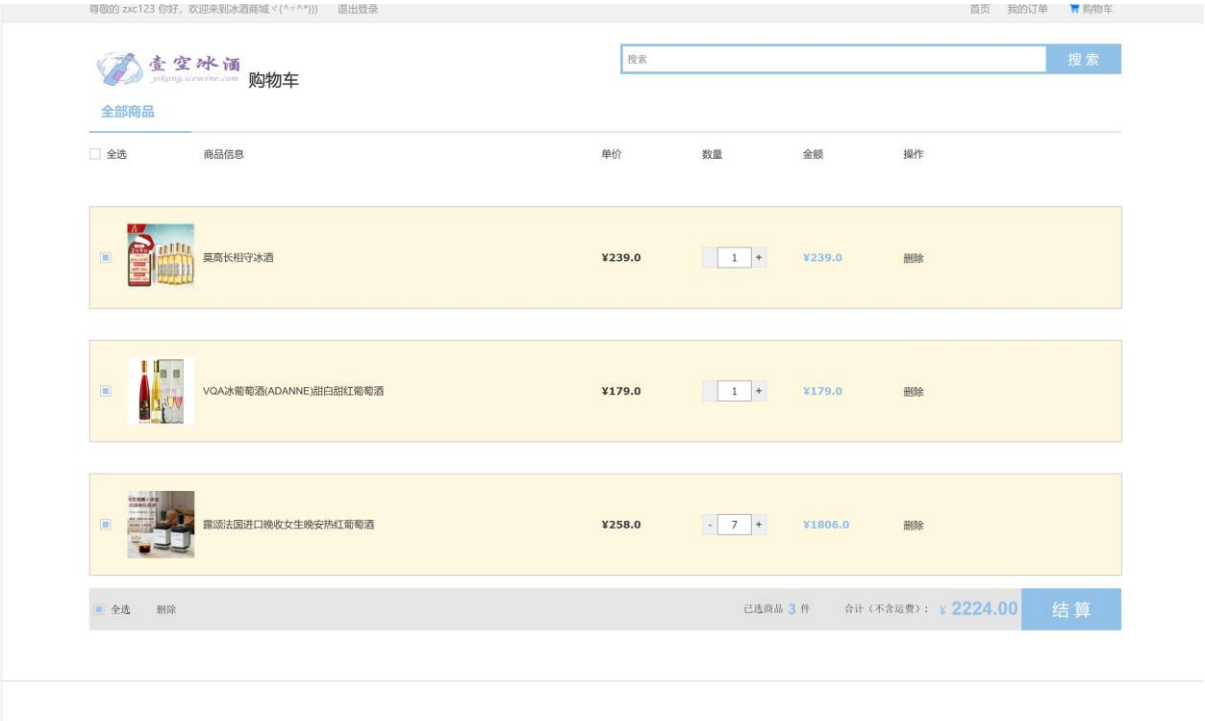
测试用例清单

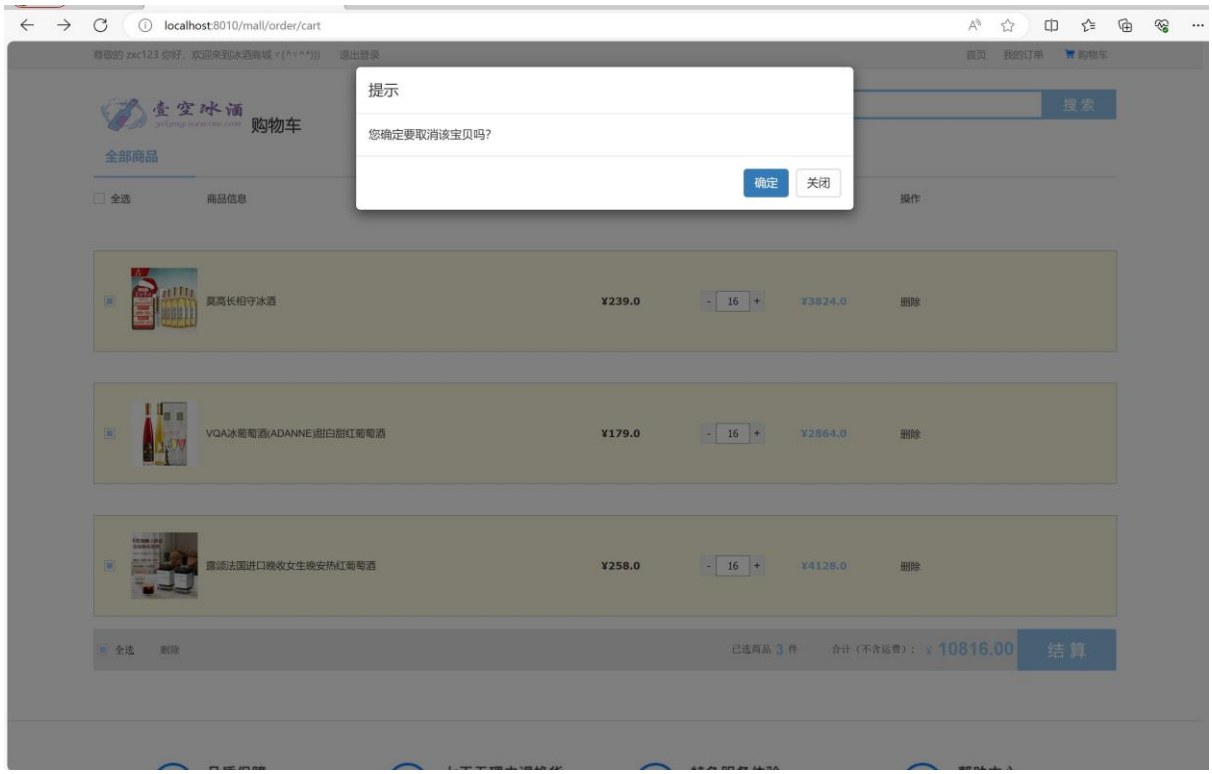
序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试 购物车 的 商品增删改 等功能是否有效	**用例 1 ：** 新增商品“长白山冰酒”到购物车	购物车页面出现新增商品“长白山冰酒”	增
		用例 2 ： 将购物车中，“莫高长相守冰酒”数量设置为 1， “VQA 冰酒”数量设置为 1， “露颂冰酒”数量设置为 7	根据对应的单价计算出的总价为 2224 元	改
		用例 3 ： 将购物车中，删除“VQA 冰酒”	购物车页面中“长白山冰酒”被删除	删
2	**目的：** 测试 购物车 的 结算 能否正常跳转	**用例 1 ：** 选中“莫高”和“露颂”，点击结算	跳转到结算页面，显示选中的商品	

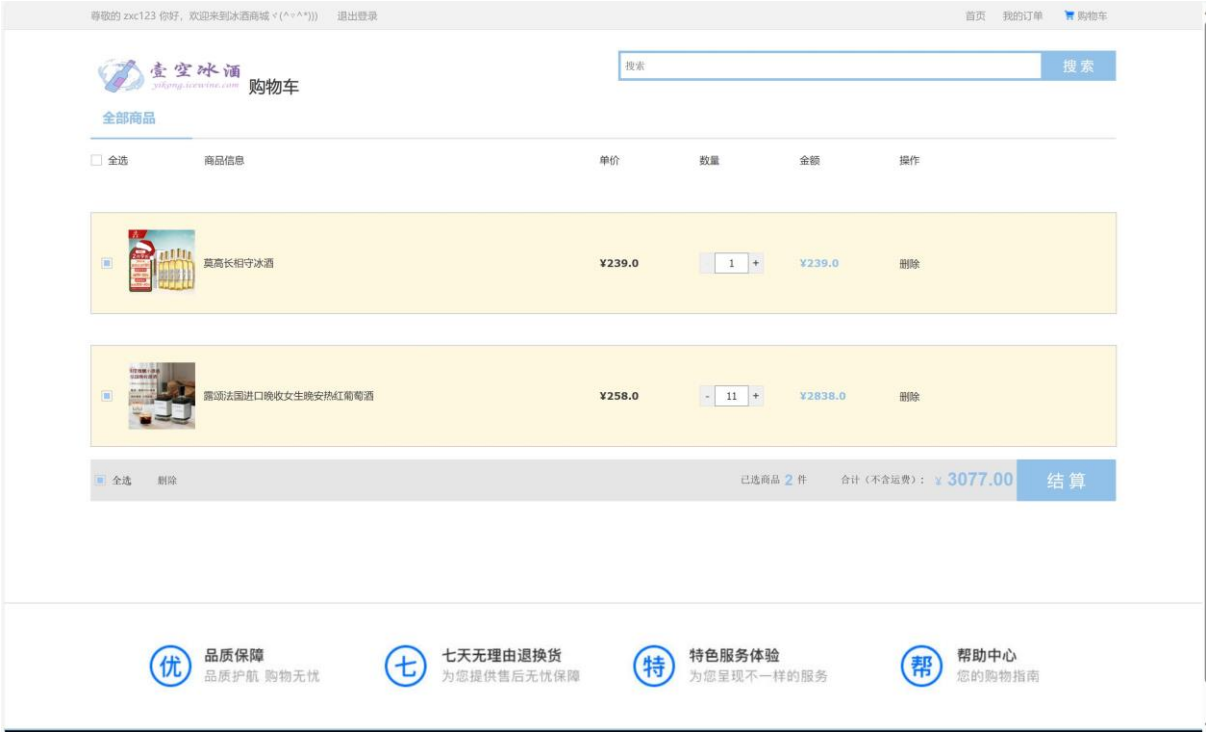
测试过程

增删改

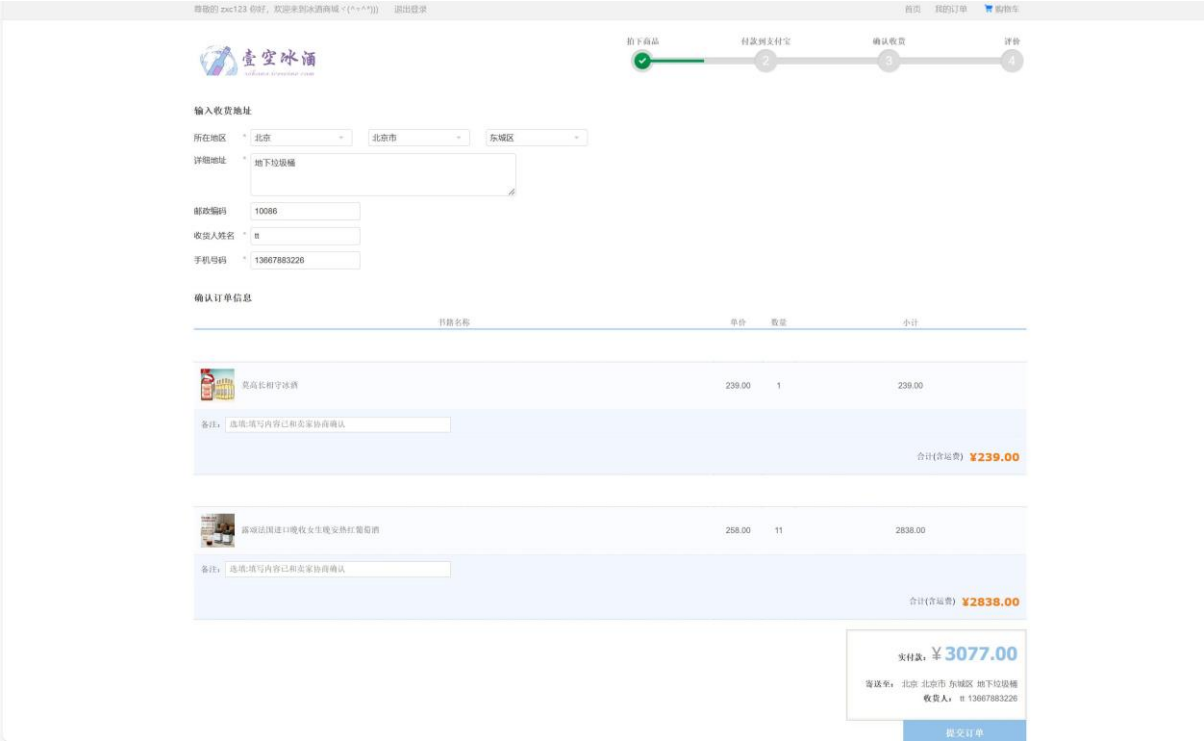

尊敬的 zxc123 你好，欢迎来到冰酒商城 ~(^~^*))) 退出登录
首页 我的订单 购物车
壹空冰酒
yikong.icewine.com
购物车
搜索
搜 索
全部商品
全选
商品信息
单价
数量
金额
操作
莫高长相守冰酒
¥239.0
1
¥239.0
删除
露颂法国进口晚收女生晚安热红葡萄酒
¥258.0
11
¥2838.0
删除
长白山冰酒红葡萄酒甜型
¥65.0
1
¥65.0
删除
全选
删除
已选商品 0 件
合计（不含运费）：¥ 0.00
结 算


尊敬的 zxc123 你好，欢迎来到冰酒商城 ~(^~^*))) 退出登录
首页 我的订单 购物车
壹空冰酒
yikong.icewine.com
购物车
搜索
搜 索
全部商品
全选
商品信息
单价
数量
金额
操作
莫高长相守冰酒
¥239.0
1
¥239.0
删除
VQA冰葡萄酒(ADANNE)甜白甜红葡萄酒
¥179.0
1
¥179.0
删除
露颂法国进口晚收女生晚安热红葡萄酒
¥258.0
7
¥1806.0
删除
全选
删除
已选商品 3 件
合计（不含运费）：¥ 2224.00
结 算





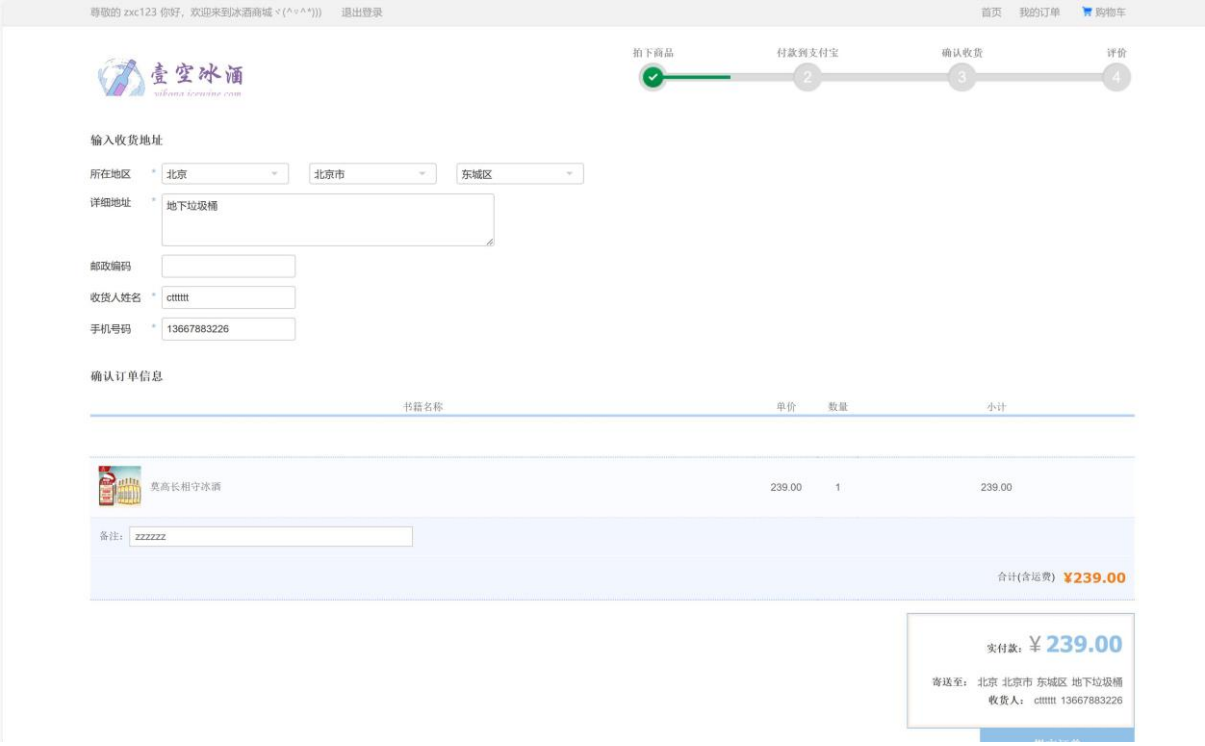
成功结算对应商品，跳转到支付界面



测试结果

序号	测试目的 & 先决条件	测试用例输入	测试结果	备注
1	**目的：** 测试 购物车 的 商品增删改 等功能是否有效	**用例 1 ：** 新增商品“长白山冰酒”到购物车	购物车页面出现新增商品“长白山冰酒”	增
		用例 2 ： 将购物车中，“莫高长相守冰酒”数量设置为 1， “VQA 冰酒”数量设置为 1， “露颂冰酒”数量设置为 7	根据对应的单价计算出的总价为 2224 元	改
		用例 3 ： 将购物车中，删除“VQA 冰酒”	购物车页面中“长白山冰酒”被删除	删
2	**目的：** 测试 购物车 的 结算 能否正常跳转	**用例 1 ：** 选中“莫高”和“露颂”，点击结算	跳转到结算页面，显示选中的商品	

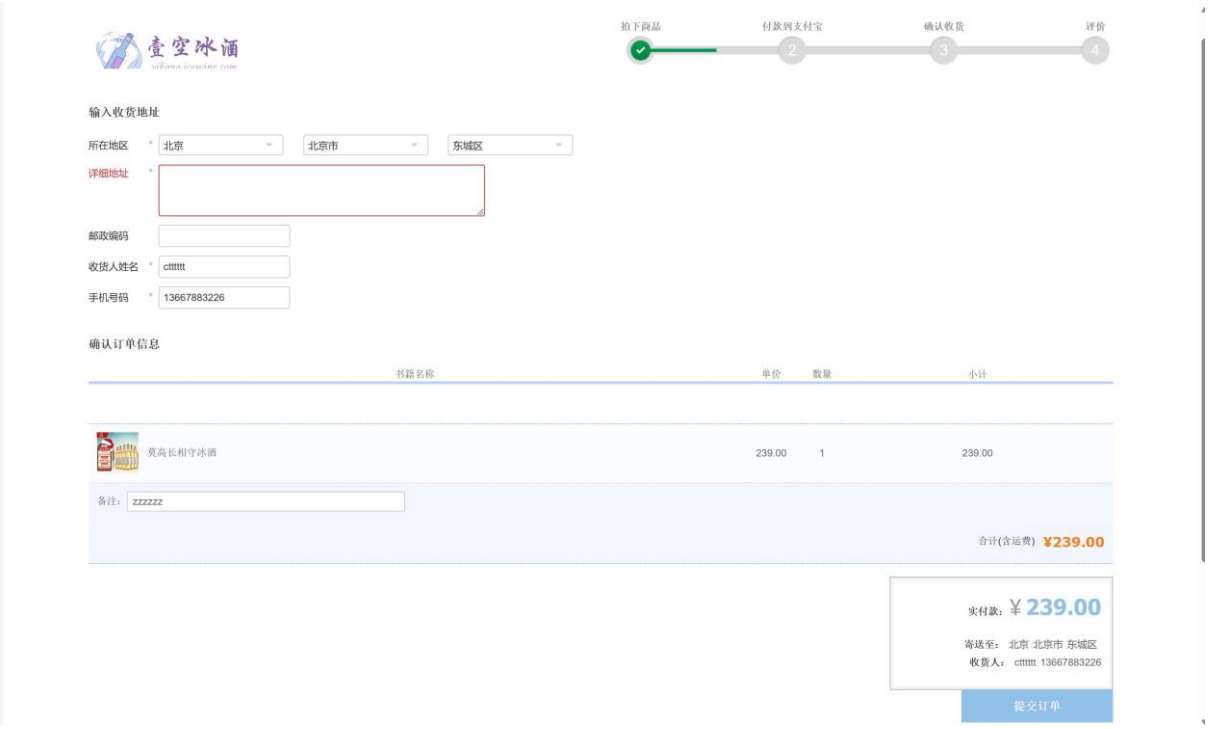
9.2.4.2.6 **地址填写测试**

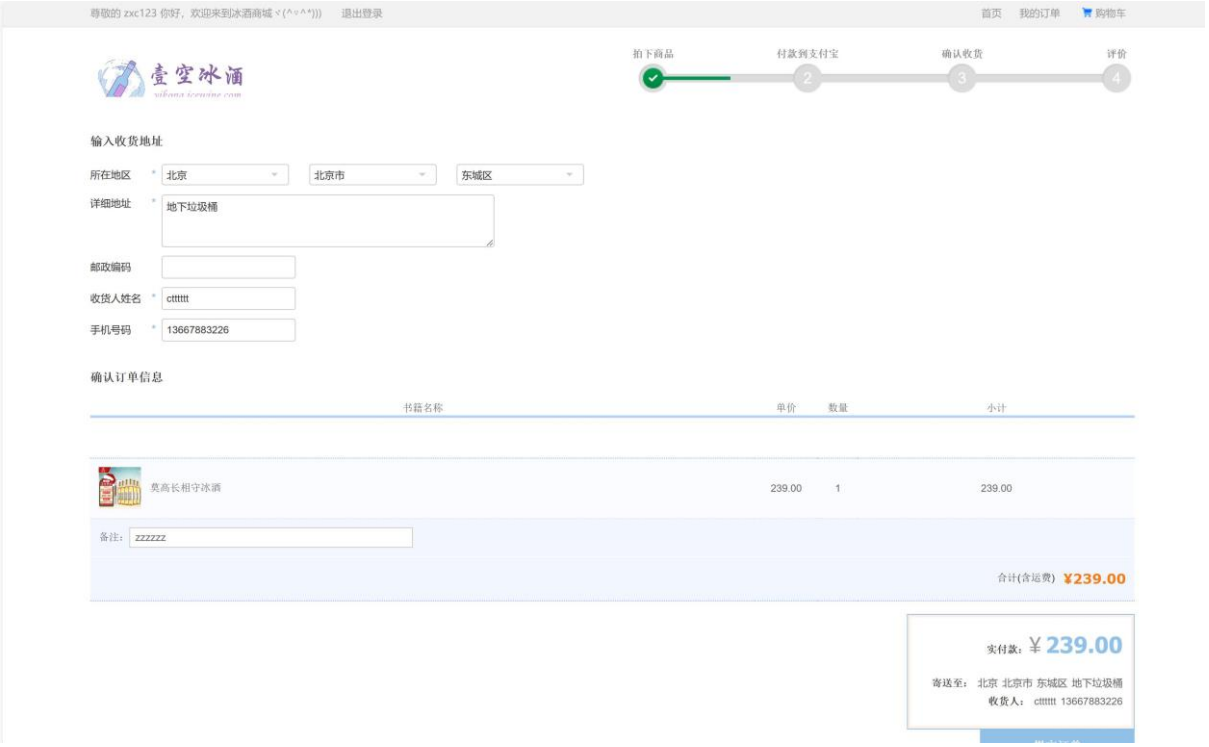


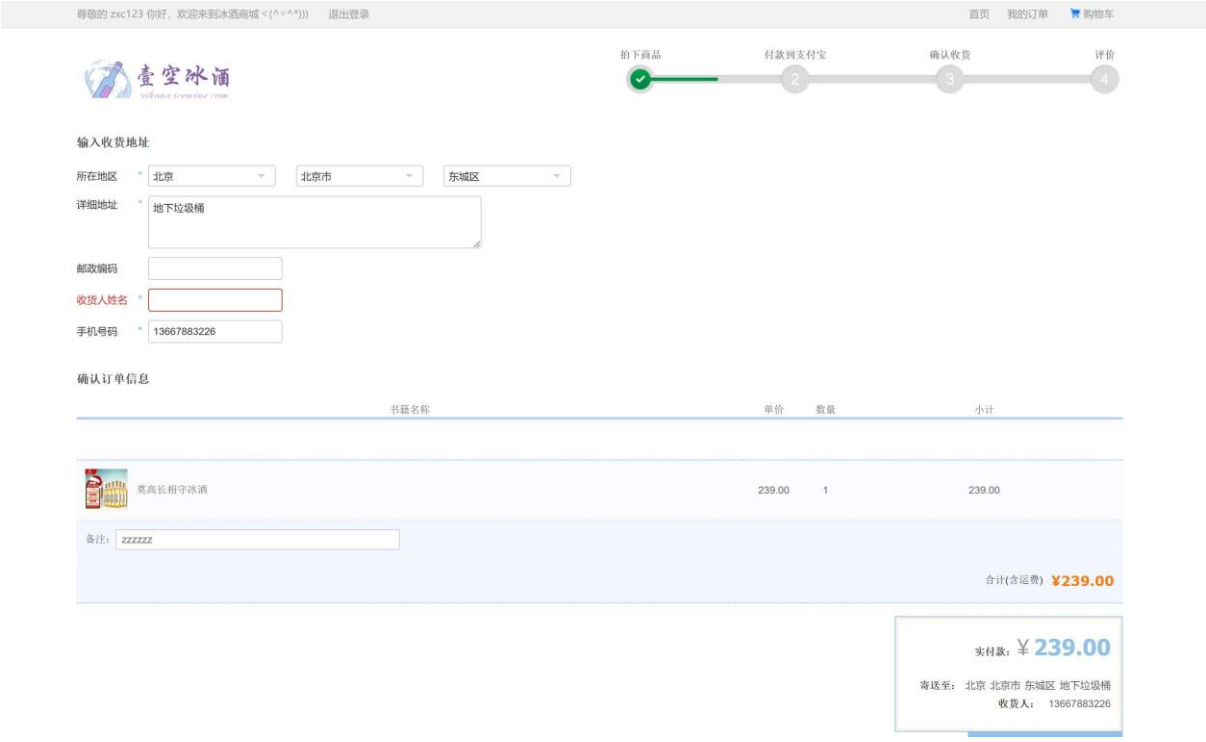
测试用例清单

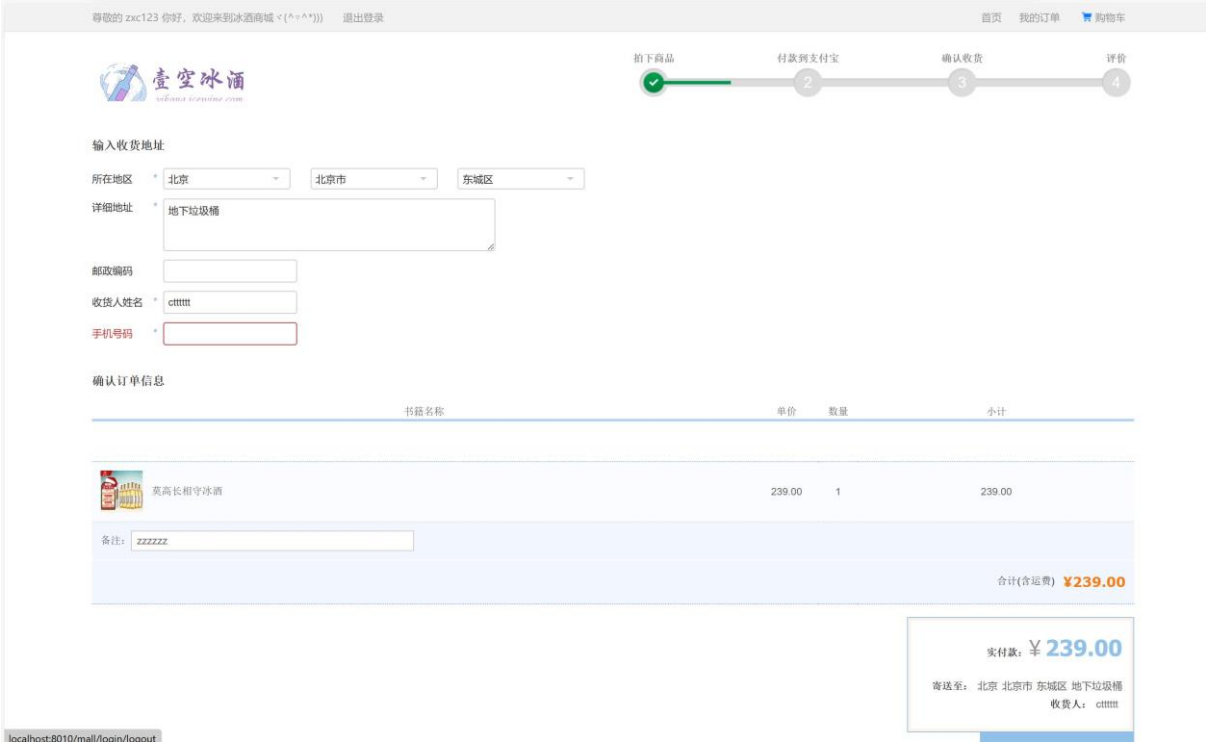
序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试能否正常填写地址，如校验必填项，和生成对应消息供用户确认	**用例 1 ：** 所在地区选择“北京-北京市-东城区”，收货人姓名“ctttttt”，电话号码填写“13667883226”，点击“提交订单”	详细地址一栏被标红	必选项校验
		用例 2 ： 所在地区选择“北京-北京市-东城区”，详细地址“地下垃圾桶”，电话号码填写“13667883226”，点击“提交订单”	收货人姓名一栏被标红	
		用例 3 ： 所在地区选择“北京-北京市-东城区”，收货人姓名“ctttttt”，详细地址“地下垃圾桶”，点击“提交订单”	电话号码一栏被标红	
2	**目的：** 测试能否正常提交订单	**用例 1 ：** 点击提交订单，看能否到达支付界面	跳转到支付界面	

测试过程


壹空冰酒
拍下商品
付款到支付宝
确认收货
评价
输入收货地址
所在地区
北京
北京市
东城区
详细地址
邮政编码
收货人姓名
ctttttt
手机号码
13667883226
确认订单信息
书籍名称
单价
数量
小计
莫高长相守冰酒
239.00
1
239.00
备注：
zzzzzz
合计(含运费) ¥239.00
实付款：¥ 239.00
寄送至： 北京 北京市 东城区
收货人： ctttttt 13667883226
提交订单


尊敬的 zxc123 你好，欢迎来到冰酒商城 ^(^v^^)))
退出登录
首页
我的订单
购物车
壹空冰酒
拍下商品
付款到支付宝
确认收货
评价
输入收货地址
所在地区
北京
北京市
东城区
详细地址
地下垃圾桶
邮政编码
收货人姓名
ctttttt
手机号码
13667883226
确认订单信息
书籍名称
单价
数量
小计
莫高长相守冰酒
239.00
1
239.00
备注：
zzzzzz
合计(含运费) ¥239.00
实付款：¥ 239.00
寄送至： 北京 北京市 东城区 地下垃圾桶
收货人： ctttttt 13667883226





测试结果

序号	测试目的 & 先决条件	测试用例输入	预期测试结果	备注
1	**目的：** 测试能否正常填写地址，如校验必填项，和生成对应消	**用例 1 ：** 所在地区选择“北京-北京市-东城区”，收货人	详细地址一栏被标红	必选项校验

	息供用户确认	姓名“ctttttt”，电话号码填写“13667883226”，点击“提交订单”		
		用例 2 ： 所在地区选择“北京-北京市-东城区”，详细地址“地下垃圾桶”，电话号码填写“13667883226”，点击“提交订单”	收货人姓名一栏被标红	
		用例 3 ： 所在地区选择“北京-北京市-东城区”，收货人姓名“ctttttt”，详细地址“地下垃圾桶”，点击“提交订单”	电话号码一栏被标红	
2	**目的：** 测试能否正常提交订单	**用例 1 ：** 点击提交订单，看能否到达支付界面	订单创建失败	bug 已修复

9.2.4.3 性能测试

9.2.4.3.1 压力测试

（一）测试场景

（1）正常负载测试：

1）模拟系统在正常使用情况下的并发用户数和交易量。
2）验证系统在正常工作负载下的性能表现。

（2）峰值负载测试：

1）模拟系统在最大并发用户数和最大交易量的情况下。
2）评估系统在负载峰值情况下的性能稳定性和响应时间。

（3）渐进增加负载测试：

1）逐步增加并发用户数和交易量，以确定系统的性能阈值和瓶颈。
2）识别系统在逐步增加负载时的性能退化点。

（二）测试指标

（1）响应时间：

1）记录系统对用户请求的响应时间，包括平均响应时间和最大响应时间。
2）确保在正常和峰值负载下，响应时间仍然在可接受范围内。

（2）吞吐量：

1）评估系统在单位时间内能够处理的事务数量。
2）确保系统在正常和峰值负载下的吞吐量满足业务需求。

（3）错误率：

1）评估系统在单位时间内能够处理的事务数量。
2）确保系统在正常和峰值负载下的吞吐量满足业务需求。

（4）资源利用率：

1）监测系统在不同负载下的硬件资源利用率。
2）确保系统不会因资源不足而导致性能下降。

（三）测试执行计划

（1）测试准备：

1）准备测试环境，包括性能测试工具的安装和配置。
2）加载测试数据集到测试数据库。
3）定义并配置性能测试场景。

（2）测试执行：

1）执行正常负载测试，记录响应时间、吞吐量和错误率。
2）执行峰值负载测试，评估系统在最大负载下的性能表现。
3）执行渐进增加负载测试，识别性能瓶颈和退化点。

（3）结果分析：

1）分析测试结果，确定系统的性能指标是否满足预期。
2）识别性能瓶颈，并记录可能的优化建议。

（4）报告与总结：

1）撰写性能测试报告，包括测试执行概要、结果摘要、问题识别和建议。
2）将报告分享给相关团队，包括开发团队和系统架构师。

（四）测试结果

（1）吞吐量：

1）平均吞吐量：650 请求/秒
2）峰值吞吐量：900 请求/秒

（2）错误率：

1）平均错误率：3.2 %
2）错误类型：500 Internal Server Error - 2.5%，503 Service Unavailable - 0.7%

（3）响应时间：

1）平均响应时间：1600 毫秒
2）最大响应时间：3200 毫秒

9.2.4.3.2 系统反应速度测试

（一）测试场景

（1）用户登录：

1）模拟用户登录操作，记录登录过程中系统的响应时间。
2）评估用户从输入凭证到成功登录所需的时间。

（2）商品浏览：

1）模拟用户浏览商品的操作，记录系统对商品查询的响应时间。
2）评估用户浏览商品列表和详细信息的速度。

（3）订单创建：

1）模拟用户创建订单的操作，记录系统处理订单创建请求的时间。
2）评估用户从选择商品到成功创建订单所需的时间。

（二）测试指标

（1）页面加载时间：

记录系统加载每个页面所需的时间，包括但不限于登录页面、商品列表页面和订单创建页面。

（2）交互响应时间：

1）评估用户在系统中进行交互时，系统对用户输入的实时响应速度。
2）包括按钮点击、下拉菜单选择等用户操作。

（3）事务处理时间：

记录系统处理关键业务事务的时间，例如订单创建和提交。

（三）测试执行计划

（1）测试准备：

1）准备测试环境，包括性能测试工具的安装和配置。
2）加载测试数据集到测试数据库。
3）定义并配置系统反应速度测试场景。

（2）测试执行：

1）执行登录操作，记录登录页面加载时间和登录成功的响应时间。
2）执行商品浏览操作，记录商品列表页面加载时间和商品详细信息的响应时间。
3）执行订单创建操作，记录订单创建页面加载时间和订单提交的响应时间。

（3）结果分析：

1）分析测试结果，确定系统在各个场景下的响应速度是否满足用户期望。
2）识别性能瓶颈，并记录可能的优化建议。

（4）报告与总结：

1）撰写性能测试报告，包括测试执行概要、结果摘要、问题识别和建议。
2）将报告分享给相关团队，包括开发团队和系统架构师。

（四）测试结果

（1）页面加载时间：

1）登录页面加载时间：800 毫秒
2）商品列表页面加载时间：1200 毫秒
3）订单创建页面加载时间：1000 毫秒

（2）交互响应时间：

1）按钮点击后的响应时间：300 毫秒
2）下拉菜单选择后的响应时间：400 毫秒

（3）事务处理时间：

订单创建到提交的总时间：2500 毫秒

9.2.4.4 单元测试

（一）测试文件 src/main/java/com/uzykj/mall/util/DateUtil.java

（1）测试方法 1 ：testGetOrderIdByTime()
测试目标：验证`DateUtil.getOrderIdByTime()`方法是否返回非空的订单 ID。

（2）测试方法 2 ：testGetCurrTime()
测试目标：验证`DateUtil.getCurrTime()`方法是否返回非空的当前时间字符串，并且该时间字符串的长度为 14，并且以"2023"开头。

（3）测试代码：

```java
import com.uzykj.mall.util.*;
import org.junit.jupiter.api.Test;

import java.util.Date;

import static org.junit.jupiter.api.Assertions.*;

no usages
class DateUtilTest {
    no usages
    @Test
    void testGetOrderIdByTime() {
        // Arrange
        String orderId = DateUtil.getOrderIdByTime();

        // Assert
        assertNotNull(orderId);
        assertEquals(17, orderId.length());
    }

    no usages
    @Test
    void testGetCurrTime() {
        // Arrange
        String currTime = DateUtil.getCurrTime();

        // Assert
        assertNotNull(currTime);
        assertEquals(14, currTime.length());
        assertTrue(currTime.startsWith("2023"));
    }
}
```

（二）测试文件 src/main/java/com/uzykj/mall/util/FileUtil.java

（1）测试说明：

1）在此测试中，我们使用@Mock 注解创建了一个模拟的 HttpSession 对象和一个模拟的 ServletContext 对象。

2）我们使用模拟对象的目的是为了隔离被测试方法中的外部依赖，例如在 FileUtil.judeFileExists()方法中可能会使用 HttpSession 或 ServletContext 对象。

3）通过模拟这些外部依赖，我们可以控制它们的行为，使测试更加独立和可控。

（2）测试方法 1 ：testJudeFileExists()

测试目标：验证 FileUtil.judeFileExists(File file)方法是否正确判断给定文件是否存在。

（3）测试方法 2 ：testJudeDirExists()

测试目标：验证 FileUtil.judeDirExists(File file)方法是否正确判断给定目录是否存在。

（4）测试方法 3 ：testCreateDirectory()

测试目标：验证 FileUtil.createDirectory(String path)方法是否能成功创建目录。

（5）测试方法 4 ：testGenerFileUrl()

测试目标：验证 FileUtil.generateFileUrl(String fileName)方法是否能正确生成文件的 URL 路径。

（6）测试方法 5 ：testGenerNewFileName()

测试目标：验证 FileUtil.generateNewFileName(String originalFileName)方法是否能正确生成新的文件名。

（7）测试方法 6：testGenerLocalFilePath()

测试目标：验证 FileUtil.generateLocalFilePath(String directory, String fileName)方法是否能正确生成本地文件路径。

（8）测试代码：

```
package com.uzykj.mall.util;

import org.junit.jupiter.api.Test;
import org.mockito.Mock;
import org.mockito.Mockito;

import javax.servlet.ServletContext;
import javax.servlet.http.HttpSession;
import java.io.File;

import static org.junit.jupiter.api.Assertions.*;

no usages
class FileUtilTest {

    2 usages
    @Mock
    private HttpSession session;

    2 usages
    @Mock
    private ServletContext context;
```

```
no usages
@Test
void testJudeFileExists() {
    // Arrange
    File file = new File("test.txt");
    // Act
    boolean result = FileUtil.judeFileExists(file);
    // Assert
    assertFalse(result);
}

no usages
@Test
void testJudeDirExists() {
    // Arrange
    File file = new File("test");
    // Act
    boolean result = FileUtil.judeDirExists(file);
    // Assert
    assertFalse(result);
}

no usages
@Test
void testCreateDirectory() {
    // Arrange
    String dirName = "test";
    // Act
    boolean result = FileUtil.createDirectory(dirName);
    // Assert
    assertTrue(result);
}

no usages
@Test
void testGenerFileUrl() {
    // Arrange
    String fileSubfix = "images";
    String imageType = "png";
    String name = "logo";
    // Act
    String result = FileUtil.generFileUrl(fileSubfix, imageType, name);
    // Assert
    assertEquals("images/png/logo", result);
}
```

```java
    no usages
    @Test
    void testGenerNewFileName() {
        // Arrange
        String originalFileName = "test.txt";
        // Act
        String result = FileUtil.generNewFileName(originalFileName);
        // Assert
        assertNotNull(result);
        assertTrue(result.endsWith(".txt"));
    }

    no usages
    @Test
    void testGenerLocalFilePath() {
        // Arrange
        String path = "images";
        String imageType = "png";
        Mockito.when(session.getServletContext()).thenReturn(context);
        Mockito.when(context.getRealPath("/")).thenReturn("/home/user/");
        // Act
        String result = FileUtil.generLocalFilePath(session, path, imageType);
        // Assert
        assertEquals("/home/user/images/png", result);
    }
}
```

（三）测试文件 src/main/java/com/uzykj/mall/util/Md5Util.java

（1）测试方法 1 ：testByteArrayToHexString()

测试目标：验证 Md5Util.byteArrayToHexString(byte[] bytes)方法是否能正确将字节数组转换为十六进制字符串。

（2）测试方法 2 ：testByteToHexString()

测试目标：验证 Md5Util.byteToHexString(byte b)方法是否能正确将字节转换为十六进制字符串。

（3）测试方法 3 ：testMD5Encode()

测试目标：验证 Md5Util.MD5Encode(String origin, String charsetname)方法是否能正确对字符串进行 MD5 编码。

（4）测试代码：

```java
package com.uzykj.mall.util;

import org.junit.jupiter.api.Test;

import java.security.MessageDigest;

import static org.junit.jupiter.api.Assertions.*;

no usages
class Md5UtilTest {

    no usages
    @Test
    void testByteArrayToHexString() {
        // Arrange
        byte[] bytes = {0, 1, 2, 15, 16, 255};
        // Act
        String result = Md5Util.byteArrayToHexString(bytes);
        // Assert
        assertEquals("0001020f10ff", result);
    }

    no usages
    @Test
    void testByteToHexString() {
        // Arrange
        byte b = -1;
        // Act
        String result = Md5Util.byteToHexString(b);
        // Assert
        assertEquals("ff", result);
    }

    no usages
    @Test
    void testMD5Encode() {
        // Arrange
        String origin = "Hello World";
        String charsetname = "UTF-8";
        // Act
        String result = Md5Util.MD5Encode(origin, charsetname);
        // Assert
        assertEquals("b10a8db164e0754105b7a99be72e3fe5", result);
    }
}
```

（四）src/main/java/com/uzykj/mall/util/PageUtil.java

（1）测试方法 1 ：testIsHasPrev()

测试目标：验证 PageUtil.isHasPrev()方法是否正确判断当前页是否有前一页。

（2）测试方法 2 ：testIsHasNext()

测试目标：验证 PageUtil.isHasNext()方法是否正确判断当前页是否有后一页。

（3）测试方法 3 ：testGetTotalPage()

测试目标：验证 PageUtil.getTotalPage()方法是否能正确计算总页数。

（4）测试方法 4 ：testGetPageStart()

测试目标：验证 PageUtil.getPageStart()方法是否能正确计算当前页的起始记录索引。

（5）测试代码：

```java
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

no usages
class PageUtilTest {
    no usages
    @Test
    void testIsHasPrev() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        // Act
        boolean result = pageUtil.isHasPrev();
        // Assert
        assertFalse(result);
    }
    no usages
    @Test
    void testIsHasNext() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        pageUtil.setTotal(100);
        // Act
        boolean result = pageUtil.isHasNext();
        // Assert
        assertTrue(result);
    }
    no usages
    @Test
    void testGetTotalPage() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        pageUtil.setTotal(95);
        // Act
        int result = pageUtil.getTotalPage();
        // Assert
        assertEquals(10, result);
    }
    no usages
    @Test
    void testGetPageStart() {
        // Arrange
        PageUtil pageUtil = new PageUtil(1, 10);
        // Act
        int result = pageUtil.getPageStart();
        // Assert
        assertEquals(10, result);
    }
```

```java
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

no usages
class PageUtilTest {
    no usages
    @Test
    void testIsHasPrev() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        // Act
        boolean result = pageUtil.isHasPrev();
        // Assert
        assertFalse(result);
    }
    no usages
    @Test
    void testIsHasNext() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        pageUtil.setTotal(100);
        // Act
        boolean result = pageUtil.isHasNext();
        // Assert
        assertTrue(result);
    }
    no usages
    @Test
    void testGetTotalPage() {
        // Arrange
        PageUtil pageUtil = new PageUtil(0, 10);
        pageUtil.setTotal(95);
        // Act
        int result = pageUtil.getTotalPage();
        // Assert
        assertEquals(10, result);
    }
    no usages
    @Test
    void testGetPageStart() {
        // Arrange
        PageUtil pageUtil = new PageUtil(1, 10);
        // Act
        int result = pageUtil.getPageStart();
        // Assert
        assertEquals(10, result);
    }
```

9.2.4.5 集成测试

9.2.4.5.1 测试说明

（一）测试目的

（1）验证组件之间的接口和通信：集成测试通过测试组件之间的接口和通信路径，确保不同组件之间的数据传递和交互正常工作。这有助于发现和解决由于接口不匹配、数据丢失或通信问题而导致的错误。

（2）检测功能完整性：集成测试确保系统的各个组件正确地协同工作，以实现系统的预期功能。它可以验证各个组件在集成环境中的行为是否符合预期，并捕获可能存在的功能缺陷。

（3）发现集成问题：集成测试有助于发现仅在组件集成后才会出现的问题。这些问题可能包括数据冲突、资源竞争、性能问题、依赖关系错误等。

（4）提高系统的稳定性和可靠性：通过在集成环境中进行全面的测试，集成测试有助于提高系统的稳定性和可靠性。它可以发现和解决在组件集成时可能出现的错误和缺陷，减少在生产环境中遇到问题的风险。

4.5.2 测试计划

（一）框架

Spring Boot + Mybatis Plus + MySQL 8.0 + JQuery + JSP

（二）测试场景

（1）Spring Boot 的启动
（2）Mybatis Plus 的启动
（3）数据库的连接
（4）数据库的修改和查询

（三）测试结果

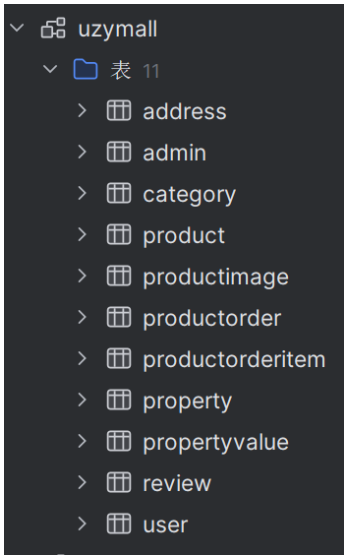
（1）Spring Boot 启动成功

```
"C:\Program Files\Java\jdk-19\bin\java.exe" ...
21:10:14.934 [main] INFO org.springframework.core.KotlinDetector - Kotlin reflection implementation not found at runtime, related features won't be available.

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.5.RELEASE)

2023-12-26 21:10:15.317  INFO 20736 --- [           main] com.uzykj.mall.MallApplication           : Starting MallApplication on LAPTOP-NFLM89GS with PID 20736 (C:\Users\SwordFlame\Desktop\uzy-ssm-m
2023-12-26 21:10:15.319  INFO 20736 --- [           main] com.uzykj.mall.MallApplication           : No active profile set, falling back to default profiles: default
2023-12-26 21:10:15.418  WARN 20736 --- [kground-preinit] o.s.h.c.j.Jackson2ObjectMapperBuilder    : For Jackson Kotlin classes support please add "com.fasterxml.jackson.module:jackson-module-kotlin
2023-12-26 21:10:16.290  INFO 20736 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8010 (http)
2023-12-26 21:10:16.306  INFO 20736 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-12-26 21:10:16.306  INFO 20736 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.19]
```

（2）Mybatis Plus 启动成功

```
 _ _   |_  _ _|_. ___ _ |    _
| | |\/|_)(_| | |_\  |_)||_|_\
     /               |         |
                        3.4.3.4
2023-12-26 21:10:17.310  INFO 20736 --- [           main] o.s.s.c.ThreadPoolTaskScheduler          : Initializing ExecutorService 'taskScheduler'
2023-12-26 21:10:17.389  INFO 20736 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8010 (http) with context path '/mall'
2023-12-26 21:10:17.391  INFO 20736 --- [           main] com.uzykj.mall.MallApplication           : Started MallApplication in 2.426 seconds (JVM running for 2.9)
2023-12-26 21:10:17.392  INFO 20736 --- [           main] com.uzykj.mall.MallApplication           : This project is started! http://localhost:8010/mall
```

(3）数据库连接成功



（4）测试数据库修改和查询

1）修改前

WHERE ORDER BY

	product_id	product_name	product_title	product_price	product_sale_price
1	1000700	莫高长相守冰酒	莫高长相守冰酒冰白葡萄酒甜酒500ml*6整箱装	269.00	239.0
2	1000701	VQA冰葡萄酒(ADANNE)甜白甜红葡萄酒	加拿大冰酒 原瓶进口VQA冰葡萄酒(ADANNE) 甜	199.00	179.0
3	1000702	莫高长相守冰酒	莫高长相守冰酒冰白葡萄酒甜酒500ml单盒装	59.00	53.0
4	1000703	长白山冰酒红葡萄酒甜型	长白山冰酒晚收冰红葡萄酒甜型红酒国产高档女士酒	66.00	65.0
5	1000704	长白山冰酒白葡萄酒甜型	长白山冰酒晚收冰白葡萄酒甜型红酒女士微醺晚安酒	66.00	65.0
6	1000705	张裕 冰酒酒庄（黄金冰谷）金钻级冰酒	张裕 冰酒酒庄（黄金冰谷）金钻级冰酒375ml礼盒	268.00	238.0
7	1000706	奥瑞安加拿大进口冰酒	奥瑞安加拿大进口冰酒晚收冰白葡萄酒红酒礼盒配2	248.00	218.0
8	1000707	原瓶进口VQA冰葡萄酒(ADANNE)甜白甜红葡萄酒	加拿大冰酒 原瓶进口VQA冰葡萄酒(ADANNE) 甜	119.00	119.0
9	1000708	莫高（MOGAO）葡萄酒红酒	莫高（MOGAO）葡萄酒红酒 甜果冰酒冰红 锦妍 5	269.00	239.0
10	1000709	PELLER ESTATES冰酒加拿大冰酒	PELLER ESTATES冰酒加拿大冰酒冰葡萄酒冰白果	729.00	629.0
11	1000710	CANA CABIN加拿大原瓶进口冰酒	CANA CABIN加拿大原瓶进口冰酒甜葡萄酒红酒维	310.00	299.0
12	1000711	马丁巴克迷你小瓶冰白葡萄酒甜型	马丁巴克迷你小瓶冰白葡萄酒甜型白雷司令女士酒1	159.00	129.0
13	1000712	INNISKILLIN云岭冰酒威代尔	INNISKILLIN云岭冰酒威代尔375ml vidal i	929.00	899.0
14	1000713	露颂法国进口晚收女生晚安热红葡萄酒	露颂【8度微醺小酒感】法国进口晚收女生晚安热红	259.00	258.0
15	1000714	张裕贵馥晚采女士甜酒	张裕（CHANGYU）烟台 张裕贵馥晚采女士甜酒 冰	797.00	727.0
16	1000715	张裕 冰酒酒庄（黄金冰谷）蓝钻级冰酒	张裕 冰酒酒庄（黄金冰谷）蓝钻级冰酒375ml*6	2989.00	2622.0
17	1000716	梅见【mini版】青梅酒	梅见【mini版】青梅酒12度150ml尝鲜装1瓶果酒	29.90	19.9
18	1000717	夜半黄金 冰酒（NACHTGOLD）冰甜葡萄酒	夜半黄金 冰酒（NACHTGOLD）冰甜葡萄酒QMP特	3608.00	3408.0
19	1000718	云岭加拿大原瓶进口VQA级云岭冰酒	云岭加拿大原瓶进口VQA级云岭冰酒Inniskillin	1049.00	999.0
20	1000719	钻石（DIAMOND）德国冰酒庄园冰甜葡萄酒	钻石（DIAMOND）德国冰酒庄园原瓶原装进口钻石	1059.00	999.0
21	1000720	dMANNER吉林通化dMANNER北冰红PUPU冰酒晚收	dMANNER吉林通化dMANNER北冰红PUPU冰酒晚收	448.00	448.0

2）添加商品

3）修改后：数据库已更新

WHERE ORDER BY

	product_id	product_name	product_title	product_price	product_sale_price
1	1000700	莫高长相守冰酒	莫高长相守冰酒冰白葡萄酒甜酒500ml*6整箱装	269.00	239.0
2	1000701	VQA冰葡萄酒(ADANNE)甜白甜红葡萄酒	加拿大冰酒 原瓶进口VQA冰葡萄酒(ADANNE) 甜	199.00	179.0
3	1000702	莫高长相守冰酒	莫高长相守冰酒冰白葡萄酒甜酒500ml单盒装	59.00	53.0
4	1000703	长白山冰酒红葡萄酒甜型	长白山冰酒晚收冰红葡萄酒甜型红酒国产高档女士酒	66.00	65.0
5	1000704	长白山冰酒白葡萄酒甜型	长白山冰酒晚收冰白葡萄酒甜型红酒女士微醺晚安酒	66.00	65.0
6	1000705	张裕 冰酒酒庄（黄金冰谷）金钻级冰酒	张裕 冰酒酒庄（黄金冰谷）金钻级冰酒375ml礼盒	268.00	238.0
7	1000706	奥瑞安加拿大进口冰酒	奥瑞安加拿大进口冰酒晚收冰白葡萄酒红酒礼盒配2	248.00	218.0
8	1000707	原瓶进口VQA冰葡萄酒(ADANNE)甜白甜红葡萄酒	加拿大冰酒 原瓶进口VQA冰葡萄酒(ADANNE) 甜	119.00	119.0
9	1000708	莫高（MOGAO）葡萄酒红酒	莫高（MOGAO）葡萄酒红酒 甜果冰酒冰红 锦妍 5	269.00	239.0
10	1000709	PELLER ESTATES冰酒加拿大冰酒	PELLER ESTATES冰酒加拿大冰酒冰葡萄酒冰白果	729.00	629.0
11	1000710	CANA CABIN加拿大原瓶进口冰酒	CANA CABIN加拿大原瓶进口冰酒甜葡萄酒红酒维	310.00	299.0
12	1000711	马丁巴克迷你小瓶冰白葡萄酒甜型	马丁巴克迷你小瓶冰白葡萄酒甜型白雷司令女士酒1	159.00	129.0
13	1000712	INNISKILLIN云岭冰酒威代尔	INNISKILLIN云岭冰酒威代尔375ml vidal i	929.00	899.0
14	1000713	露颂法国进口晚收女生晚安热红葡萄酒	露颂【8度微醺小酒感】法国进口晚收女生晚安热红	259.00	258.0
15	1000714	张裕贵馥晚采女士甜酒	张裕（CHANGYU）烟台 张裕贵馥晚采女士甜酒 冰	797.00	727.0
16	1000715	张裕 冰酒酒庄（黄金冰谷）蓝钻级冰酒	张裕 冰酒酒庄（黄金冰谷）蓝钻级冰酒375ml*6	2989.00	2622.0
17	1000716	梅见【mini版】青梅酒	梅见【mini版】青梅酒12度150ml尝鲜装1瓶果酒	29.90	19.9
18	1000717	夜半黄金 冰酒（NACHTGOLD）冰甜葡萄酒	夜半黄金 冰酒（NACHTGOLD）冰甜葡萄酒QMP特	3608.00	3408.0
19	1000718	云岭加拿大原瓶进口VQA级云岭冰酒	云岭加拿大原瓶进口VQA级云岭冰酒Inniskillin	1049.00	999.0
20	1000719	钻石（DIAMOND）德国冰酒庄园冰甜葡萄酒	钻石（DIAMOND）德国冰酒庄园原瓶原装进口钻石	1059.00	999.0
21	1000720	dMANNER吉林通化dMANNER北冰红PUPU冰酒晚收	dMANNER吉林通化dMANNER北冰红PUPU冰酒晚收	448.00	448.0
22	1000844	ABC	ABCD	123.00	123.0

9.2.5 评价

冰酒销售管理系统经过系统测试，在大多数功能和性能方面表现良好，但也存在一些改进的空间。以下是对系统测试过程的评价：

（一）成功之处

（1）核心功能稳定性： 系统的核心功能如用户管理、产品管理、销售订单处理等在测试中表现稳定可靠。

（2）报表生成准确性： 报表生成功能能够准确反映销售和库存状态，为用户提供了重要的业务数据分析依据。

（3）用户权限验证有效性： 用户权限验证机制能够有效地限制不同用户角色对敏感信息和功能的访问，保障系统安全性。

（二）改进空间

（1）订单状态更新优化： 在某些情况下存在订单状态更新延迟的问题，需要优化更新流程以确保及时性和准确性。

（2）跨浏览器兼容性： 用户界面在不同浏览器上的样式可能存在不一致性，建议进行更全面的跨浏览器兼容性测试。

（三）建议

（1）持续改进迭代： 尽管系统在大多数方面表现良好，但测试过程中发现的问题和改进点应成为后续版本迭代改进的重点。

（2）用户体验优化： 除了功能性改进外，注重用户界面友好性和易用性的优化，将提升系统整体用户体验。

（四）总结

冰酒销售管理系统在测试中展现了其稳定性和基本功能的可靠性。然而，对于发现的缺陷和改进点，开发团队应采取积极的措施，在后续版本中持续改进，以提高系统的质量和用户满意度。系统测试是一个循序渐进的过程，持续的改进和迭代将推动系统向更高水平发展。

10. 系统使用说明书

10.1 前台

10.1.1 注册

填写用户名，登陆密码，确认密码，昵称。选择性别，出生日期，居住地址。点击注册。

图 10.1.1 ：注册

10.1.2 登录

输入用户名和密码，点击登录，跳转到商城首页。

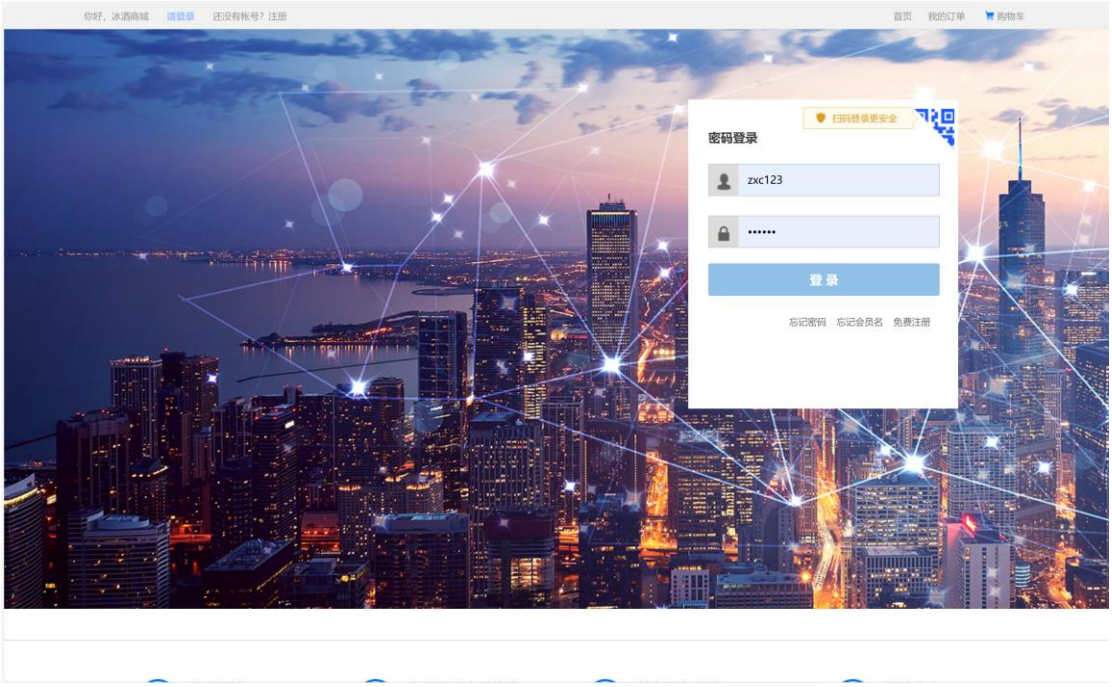


图 10.1.2 ：登录

10.1.3 首页

点击商品：跳转到对应的商品详情页。
点击分类：筛选出指定种类下的商品。
在搜索框中输入关键字：搜索出符合条件的相关商品。

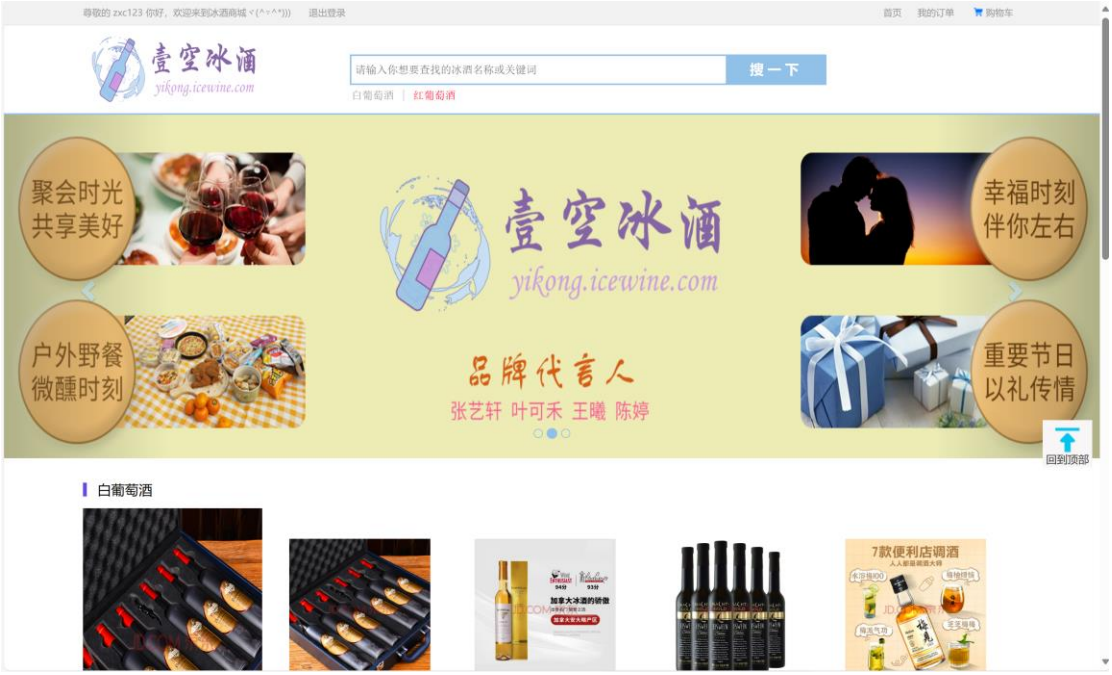


图 10.1.3 ：首页

10.1.4 商品详情页

点击立即购买：创建购买该商品的订单，跳转到地址填写页面。
点击加入购物车：将对应数量的商品加入购物车。

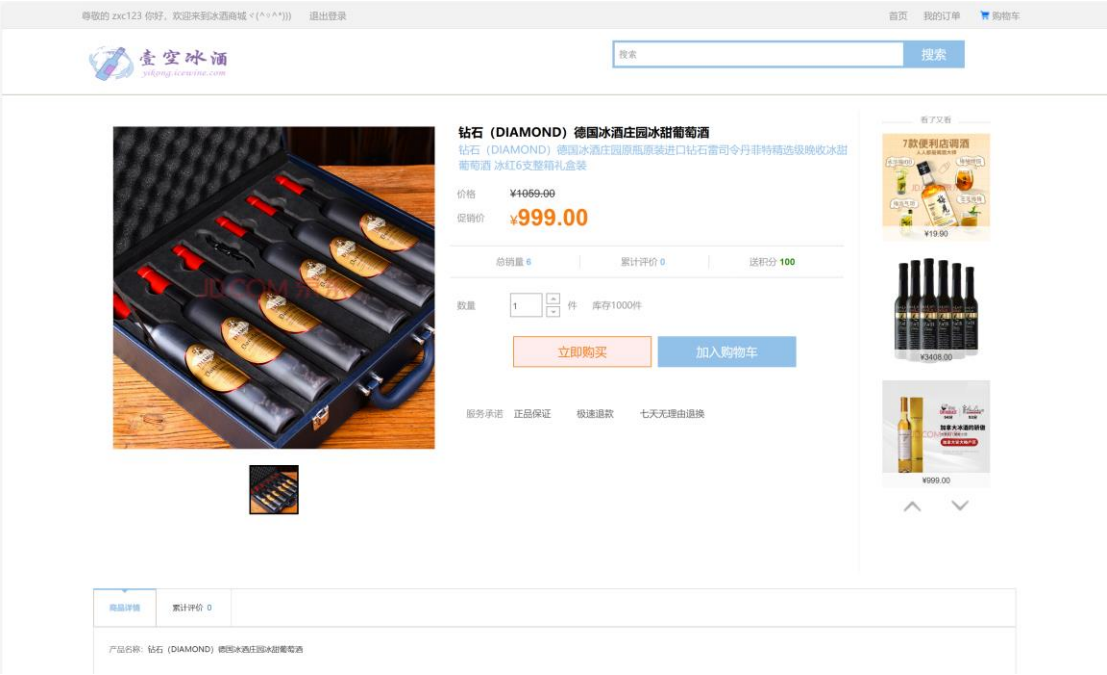


图 10.1.4 ：商品详情页

10.1.5 购物车

可以对购物车内商品进行数量的修改以及删除操作。
点击数量对应的增减按钮以及输入可以改变商品数量。
点击删除，该商品从购物车中被删除。
选择购物车内商品，点击结算，可以生成相应订单并跳转到支付界面。

图 10.1.5 ：购物车

10.1.6 我的订单

已创建的订单会在这里显示，可以在这里对订单进行管理。
点击“立即付款”：跳转到支付界面。
点击“取消订单”：出现删除订单按键。
点击“删除订单“：该订单被删除。
点击“提醒发货”：后台接收到发货提醒，可以选择发货。

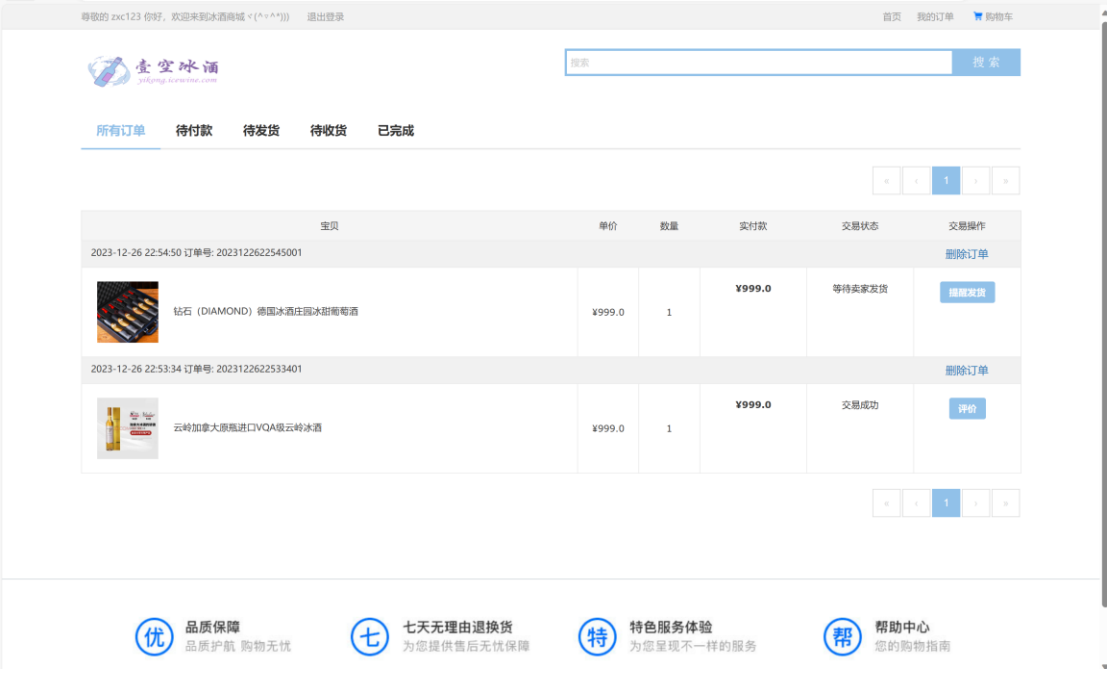


图 10.1.6.1 ：我的订单 1

等待后台发货后，界面上该订单的交易操作会变为“确认收货”。

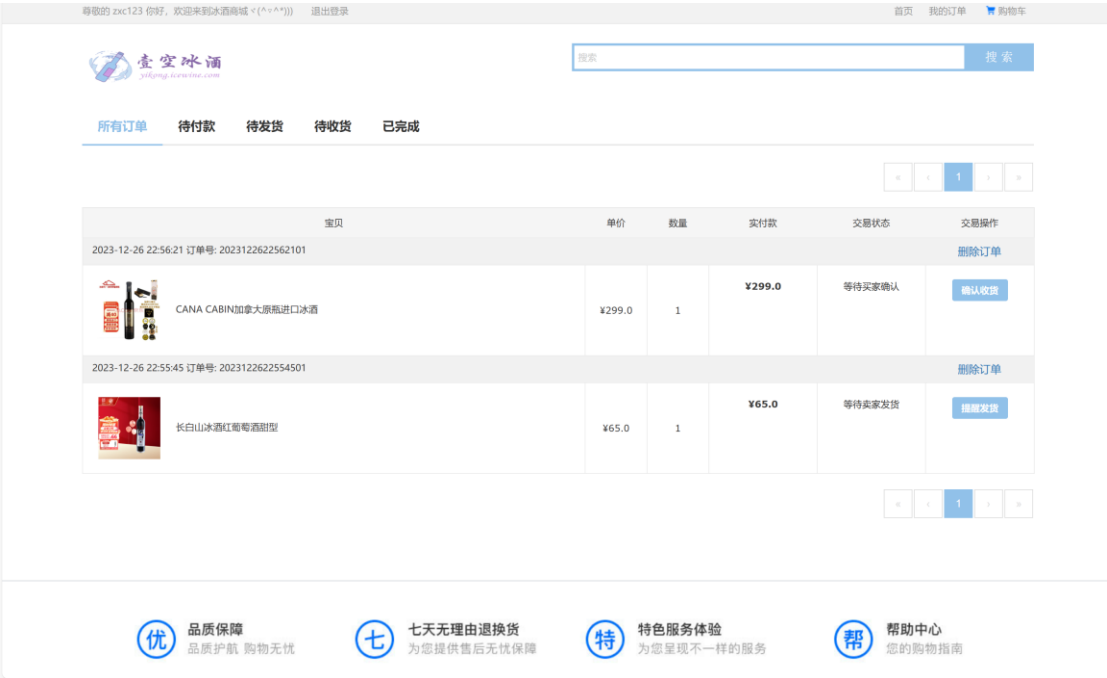


图 10.1.6.2 ：我的订单 2

点击“确认收货”，进入到下一流程，点击“确定”，即可完成收货。

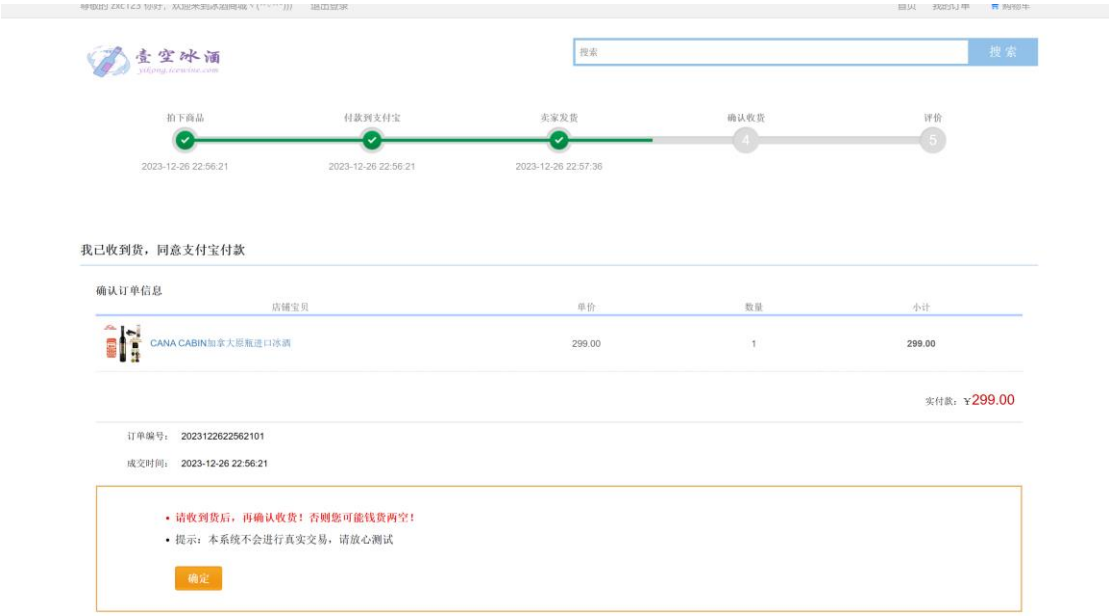


图 10.1.6.3 ：我的订单 3

收货完成：

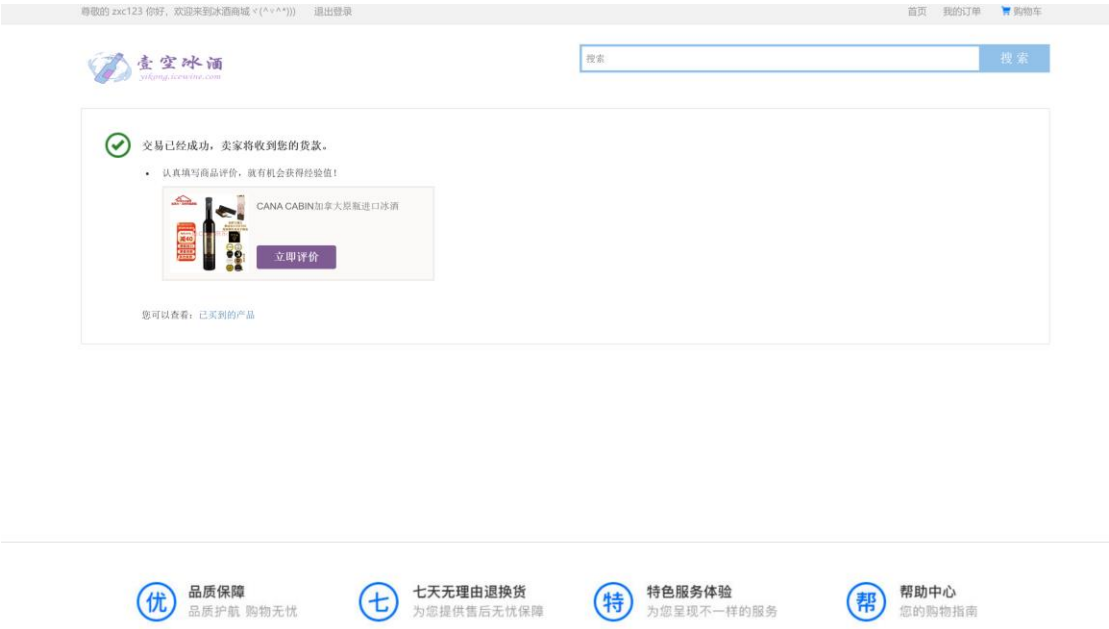


图 10.1.6.4 ：我的订单 4

10.1.7 商品筛选页

点击“综合”：商品按照综合条件降序排列。
点击“新品”：商品按照新品条件降序排列。
点击“销量”：商品按照销量条件降序排列。

点击“价格”：商品按照价格条件降序/升序排列。

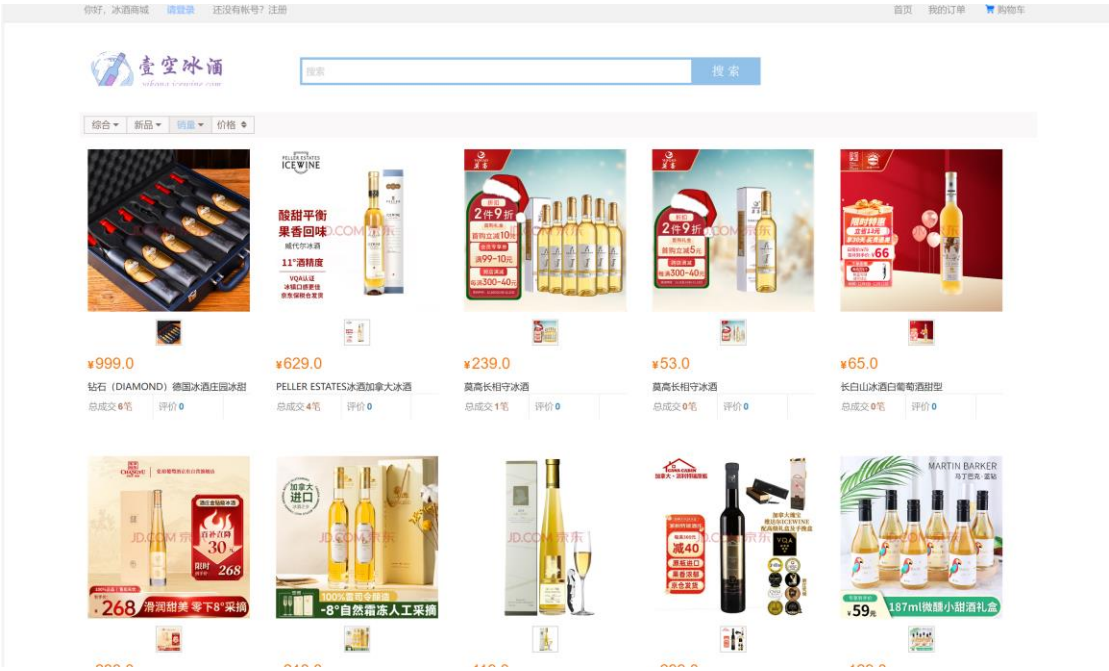


图 10.1.7 ：商品筛选页

10.1.8 支付界面

点击支付订单，跳转到支付界面。

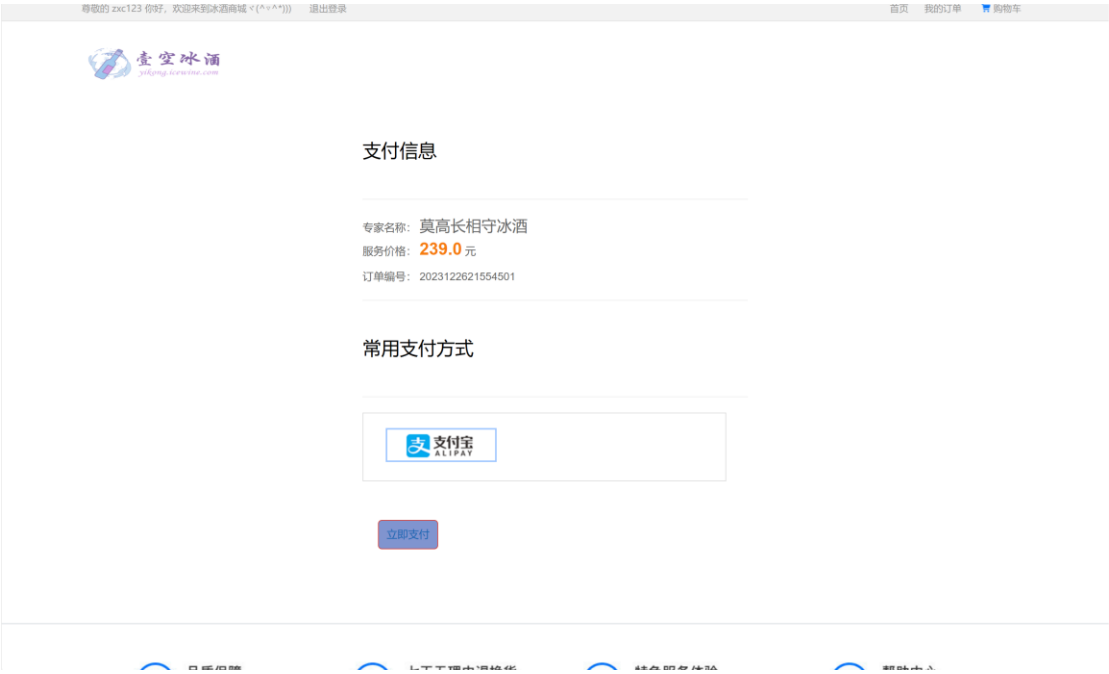


图 10.1.8 ：支付页面

10.2 后台

10.2.1 管理首页

管理首页显示当前上架产品数量，注册用户数量，成交单数量和订单交易额。
选择对应日期可以筛选出该时间范围内的相关信息。

图 10.2.1 ：管理首页

10.2.2 分类管理

在分类名称中输入相应关键词，点击查询可以搜索出对应结果。
点击重置会把输入框清空然后显示所有结果。

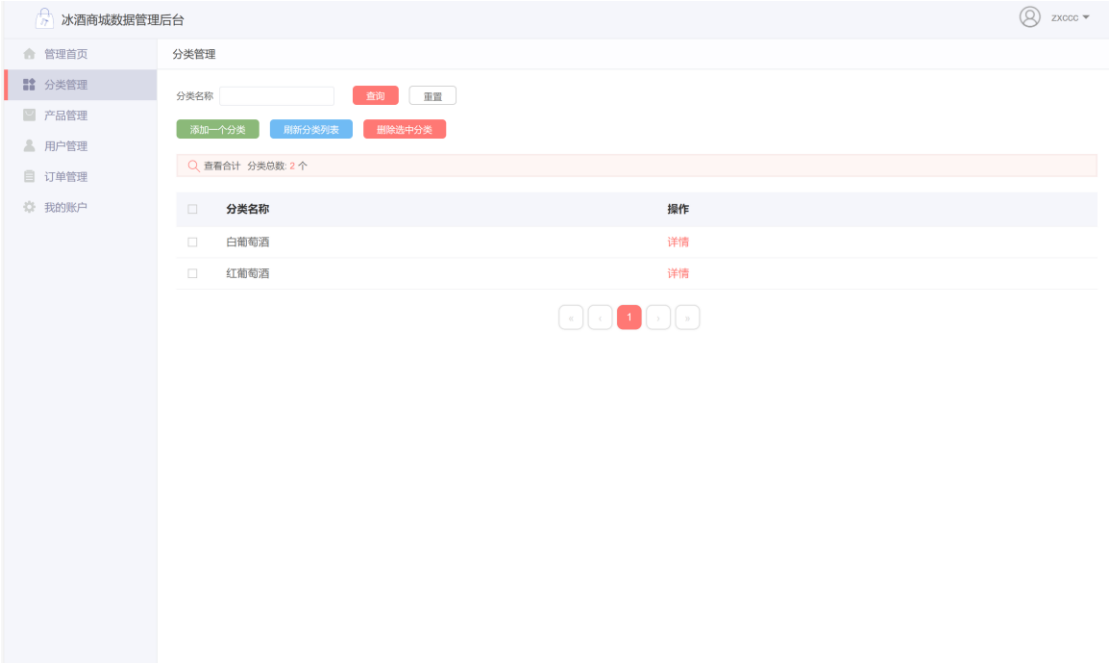


图 10.2.2.1 ：分类管理 1

点击 添加一个分类，输入分类名称，上传图片，点击提交即可生成新分类。

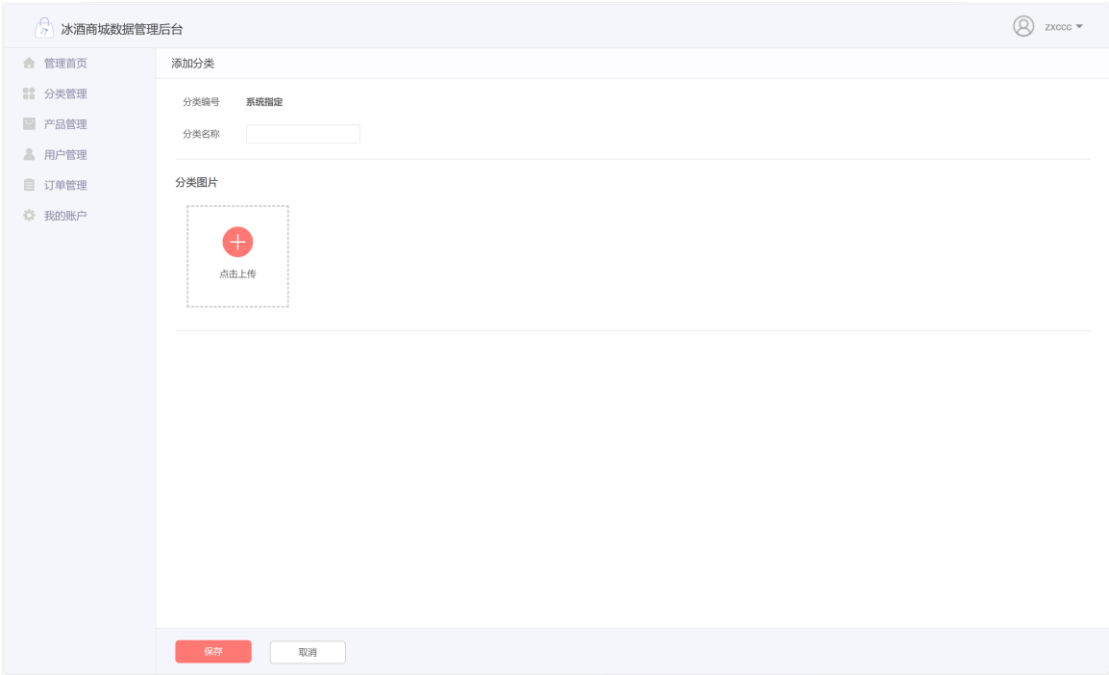


图 10.2.2.2 ：分类管理 2

点击 刷新分类列表，更新当前列表状态。
选中分类，点击 删除选中分类 即可删除对应分类。

10.2.3 产品管理

在产品名称中输入相应关键词/选择产品状态/输入对应金额范围/选择产品类型，点击查询可以搜索出对应结果。

点击重置会把输入框清空然后显示所有结果。

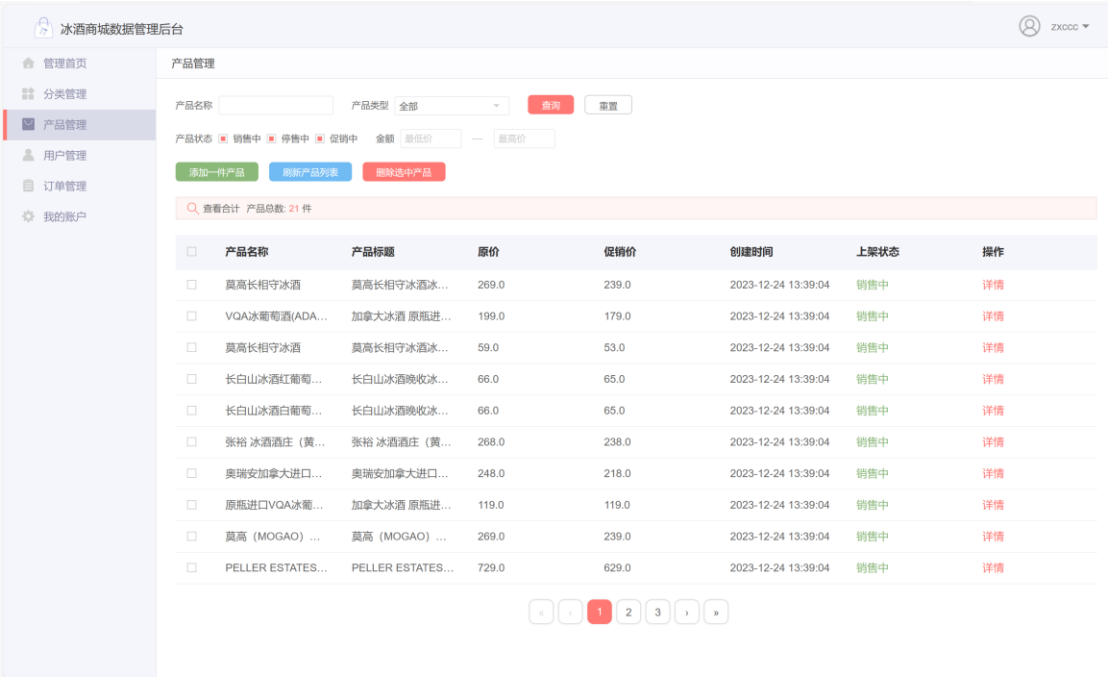


图 10.2.3.1 ：产品管理 1

点击 添加一件产品，输入分类名称，上传图片，点击提交即可生成新分类。

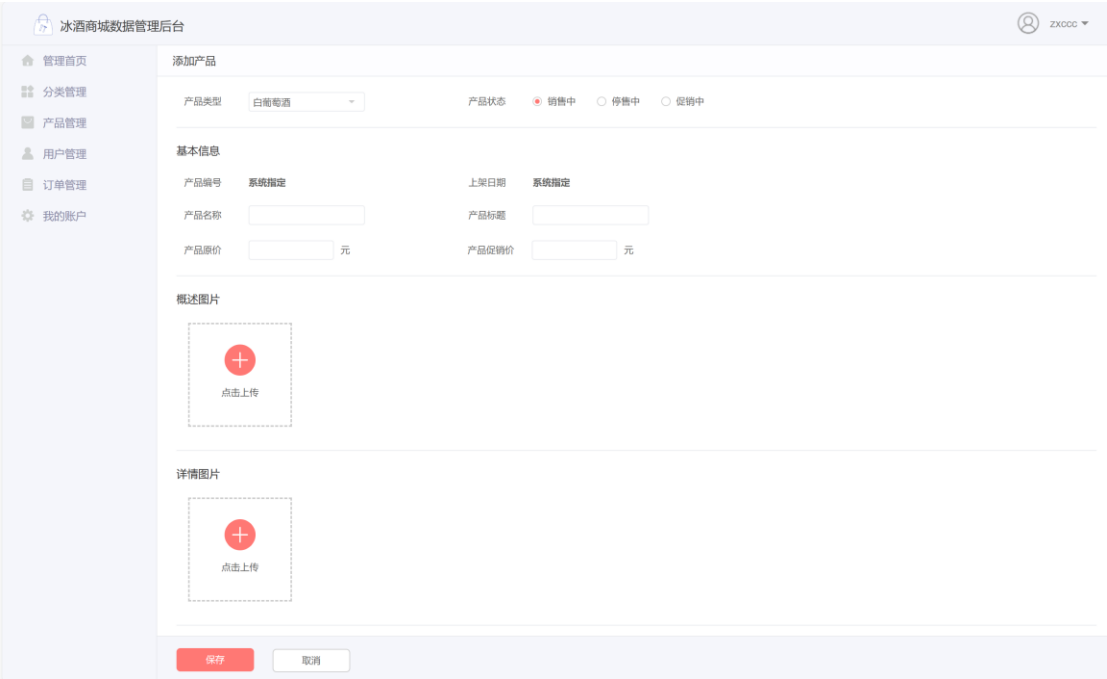


图 10.2.3.2 ：产品管理 2

点击 刷新产品列表，更新当前列表状态。
选中产品，点击 删除选中产品 即可删除对应产品。

10.2.4 用户管理

在用户名/昵称中输入相应关键词，点击查询可以搜索出对应结果。
点击重置会把输入框清空然后显示所有结果。

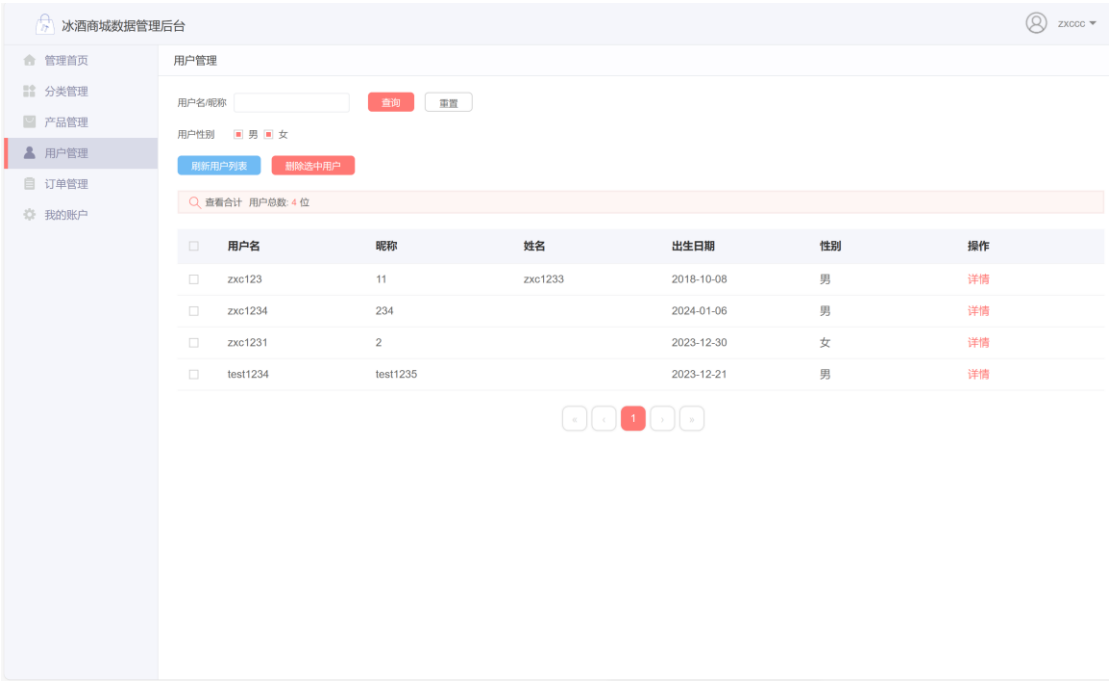


图 10.2.4 ：用户管理

点击 刷新用户列表，更新当前列表状态。
选中用户，点击 删除选中用户 即可删除对应用户。

10.2.5 订单管理

在分类名称中输入相应关键词，点击查询可以搜索出对应结果。
点击重置会把输入框清空然后显示所有结果。
点击 刷新订单列表，更新当前订单状态。
选中订单，点击 删除选中订单 即可删除对应订单。

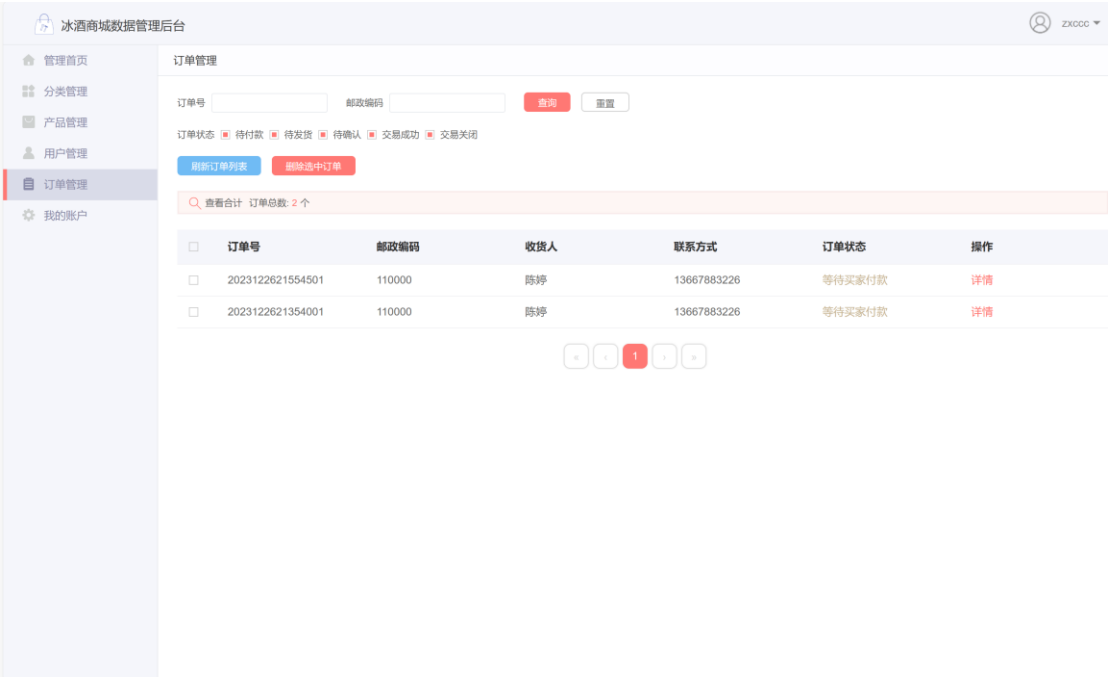


图 10.2.5.1 ：订单管理 1

订单详情：

图 10.2.5.2 ：订单管理 2

后台可以管理发货，点击发货之后会更新订单状态。

10.2.6 我的账户

该界面可以修改昵称，头像和密码，以及退出当前帐号。

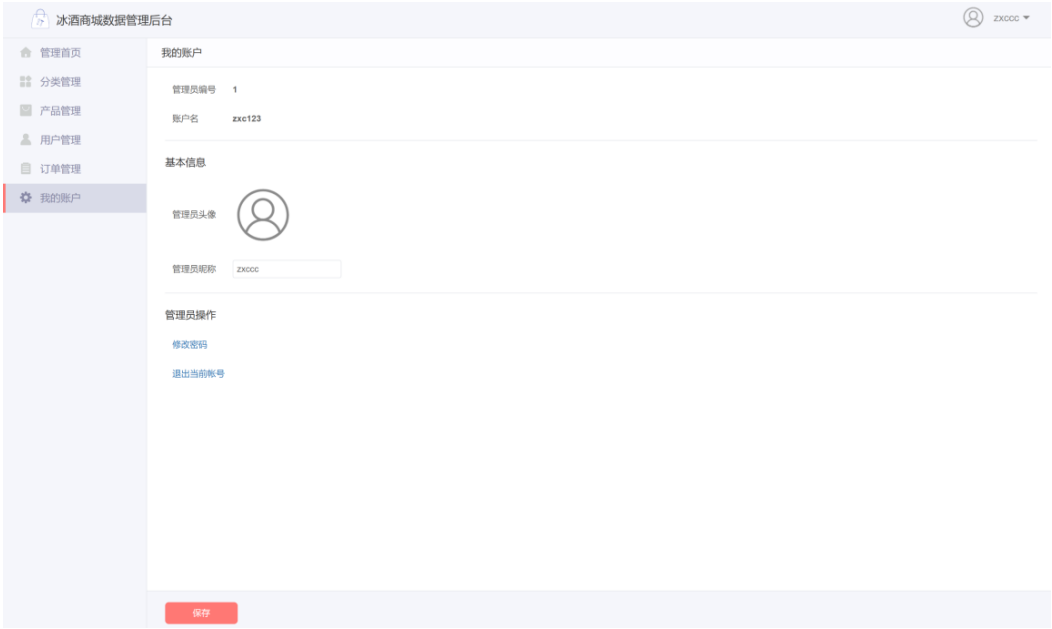


图 10.2.6 ：我的账户

10.2.7 登录

输入用户名和密码，点击登录，跳转到后台管理首页。

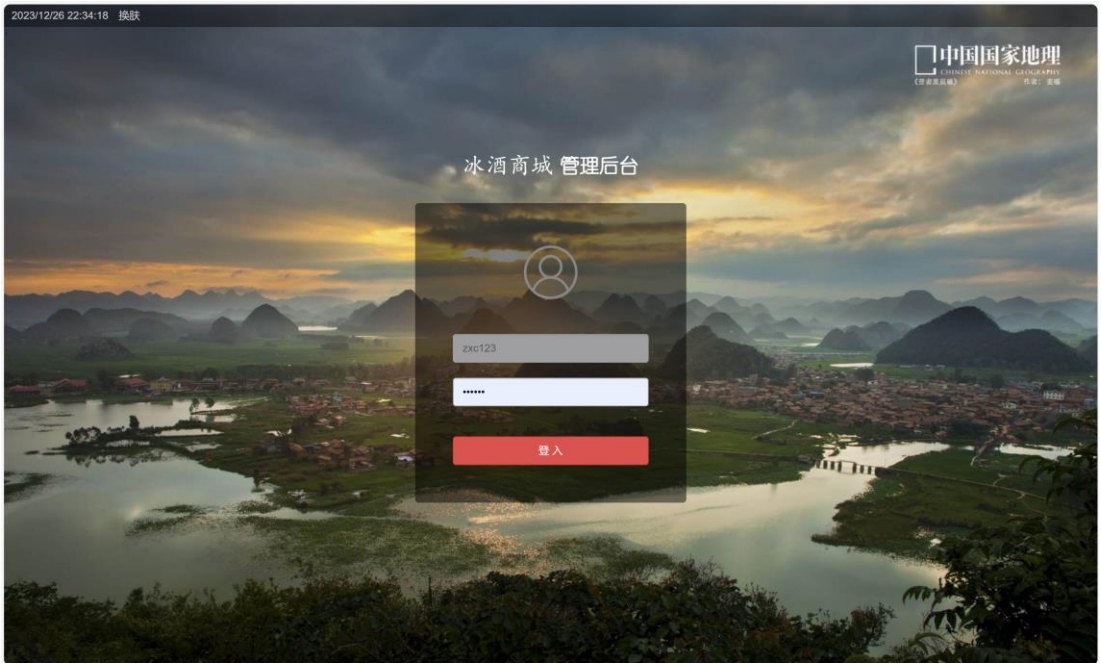


图 10.2.7 ：登录

11. 参考

11.1 参考资料

① Kleppmann, Martin. "Designing Data-Intensive Applications". O'Reilly Media, 2017. 提供对于构建可靠的数据持久化系统的深入理解。

② Eich, Brendan et al, "ECMAScript 2022 Language Specification". ECMA International, 2022. 提供 JavaScript 的标准和规范。

③ Pivotal Software, "Spring Boot Reference Guide". Pivotal Software, 2023. 提供了对于 Spring Boot 的详细指南和使用方法。

④ Facebook Inc., "React Documentation". Facebook, 2023. 提供对于使用 React 构建用户界面的指南和说明。

11.2 技术文档

① MySQL Official Documentation, Available at: https://dev.mysql.com/doc/. 提供了对于使用 MySQL 数据库的全方位指南。

② Vue.js Guide, Available at: https://v3.vuejs.org/guide/introduction.html. 提供了对于 Vue.js 的详细指南和使用方法。

③ AWS Documentation, Available at: https://docs.aws.amazon.com/. 提供关于使用 AWS 云服务的详细指南。

11.3 研究和报告

International Organisation of Vine and Wine, "Global economic vitiviniculture data 2023". 提供了全球葡萄酒和冰酒销售和生产的详细数据。

11.4 相关法规

① Food and Drug Regulations (C.R.C., c. 870) Canada, Available at: https://laws-lois.justice.gc.ca/eng/regulations/C.R.C.,_c._870/. 提供了关于在加拿大销售和生产冰酒的法

规。

② The Alcohol and Gaming Commission of Ontario, Available at: https://www.agco.ca/alcohol. 提供了关于在安大略省销售和生产冰酒的法规。

附录

附录 1 小组成员及贡献比例

第 5 组 壹零空

学号	姓名	贡献比例
2021192010	王曦	31
2021150024	张艺轩	23
2021280462	叶可禾	23
2021040074	陈婷	23