



重庆对外经贸学院  
CHONGQING COLLEGE OF INTERNATIONAL BUSINESS AND ECONOMICS

## 2024 届本科毕业论文（设计）

题 目      基于 JAVA 的网上水果商城的设计与实现

---

学      院      数学与计算机学院

---

年      级      2020 级

---

专      业      计算机科学与技术

---

学生姓名      魏文来      学号      20200551089

---

指导老师      杨巧梅      职称      讲师

---

日      期      2024 年 5 月

---

# 重庆对外经贸学院

## 本科毕业论文（设计）诚信声明

本人郑重声明：所呈交的本科毕业论文（设计）是本人在指导教师的指导下，独立进行研究工作所取得的成果。除了文中特别加以注明的地方外，论文（设计）中没有抄袭他人研究成果和伪造数据等行为。本人完全意识到本声明的法律结果由本人承担。

本科毕业论文（设计）作者签名：

年 月 日

## 致 谢

值此论文完成之际，谨向所有关心和支持我的人们致以诚挚的谢意！在整个研究过程中，你们的支持和鼓励都是我前进的动力。无论是在学术上的建议还是在情感上的支持，都让我感到非常幸运和温暖。这篇论文的完成不仅是我个人努力的成果，更是大家共同努力的结晶。愿我们未来继续携手并肩，共同前行！

作者：魏文来

日期：2024.4.2

## 摘 要

本项目旨在开发一款高性能、可扩展的网上水果商城系统，采用前后端分离架构，基于 Vue、SpringBoot、MyBatis 和 Redis 技术栈，以 B2B2C 设计为基础，并使用 SpringBoot 和 MyBatis 框架。为了提升系统安全，采用防范 XSS 攻击的措施，并引入分布式锁机制以确保在多实例环境下的数据一致性。系统实现了多实例部署，可实现浏览商品、加入购物车、下单购买、商品管理、用户管理等功能。提供用户友好的购物体验 and 高效的系统运行，为生产环境提供了安全可靠的解决方案。

**关键词：** 前后端分离；网上水果商城系统；Vue；SpringBoot；MyBatis；

## **Abstract**

This project aims to develop a high-performance and scalable online fruit mall system, using a front-end and back-end separation architecture, based on Vue, SpringBoot, MyBatis, and Redis technology stacks, with a B2B2C design as the foundation, and using SpringBoot and MyBatis frameworks. In order to improve system security, measures are taken to prevent XSS attacks and a distributed locking mechanism is introduced to ensure data consistency in a multi instance environment. The system has implemented multi instance deployment, which can achieve functions such as browsing products, adding to shopping carts, placing orders for purchases, product management, and user management. Providing a user-friendly shopping experience and efficient system operation, providing a safe and reliable solution for the production environment.

**Key words:** Front-end and back-end separation; Online fruit mall system; Vue; SpringBoot; MyBatis;

# 目 录

1 绪论 .....	1
1.1 背景与意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究目的.....	1
1.2 研究内容.....	2
2 相关技术介绍 .....	3
2.1 开发框架.....	3
2.1.1 SpringBoot 框架 .....	3
2.1.2 Vue 框架 .....	3
2.2 开发语言.....	3
2.2.1 JAVA 语言 .....	3
2.2.2 JSON 语言 .....	3
2.3 开发工具.....	3
2.3.1 MySQL 数据库 .....	3
2.3.2 Vscode.....	3
2.3.3 IntelliJIDEA.....	3
2.3.4 Hbuilder.....	4
3 系统需求 .....	5
3.1 需求分析与概要设计.....	5
3.1.1 需求分析.....	5
3.1.2 概要设计.....	6
4 系统的详细设计与实现 .....	7
4.1 系统开发环境与实现框架.....	7

4.1.1 系统开发环境.....	7
4.1.2 数据库设计.....	7
4.1.3 系统实现架构.....	12
4.2 系统详细设计.....	12
4.2.1 前台购物系统.....	15
4.2.2 后台管理系统.....	24
5 系统测试.....	31
5.1 前台购物系统测试.....	31
5.2 后台管理系统测试.....	33
6 总结与展望.....	35

# 1 绪论

## 1.1 背景与意义

### 1.1.1 研究背景

研究背景：随着互联网的发展，电子商务已经渗透到人们生活的方方面面，消费者在网络上购买各种商品的趋势日益增长。这种趋势也逐渐延伸到了食品行业。水果作为人们日常生活必需品之一，也渐渐成为了网上购物的热门商品之一。然而，当前市场上的大部分水果电商平台在水果的质量、配送服务等方面存在一些问题。一些电商平台不提供水果品质的保证，消费者很难确保购买到的水果是新鲜的和高质量的。另外，由于水果易腐的特性，顺利配送对时间和温度都有较高要求，但是许多平台在这方面没有很好的解决方案。因此，设计并实现一个高质量、高效的网上水果商城是非常有必要的。

### 1.1.2 研究目的

（1）网上水果商城为消费者提供了便捷的购物体验，消费者可以随时浏览商品，选择水果并下单购买，极大地提高了消费者的购物体验，节省了他们的时间和精力。网上水果商城全天候开放的在线浏览和购买功能，消费者可以随时随地进行购物，不受时间和地点的限制。

（2）网上水果商城的出现提高了整个水果行业的效率。它减少了中间环节，直接链接了供应商和消费者，降低了成本，提高了销售效率。供应商可以直接通过网上平台销售新鲜的水果。网上水果商城提供了直接链接供应商和消费者的平台。

（3）网上水果商城还提升了服务质量。网上水果商城拥有消费者反馈评价系统，供应商可以更加方便地了解消费者的需求和偏好，从而调整产品的供应和销售策略。商家可以通过物流配送服务将商品送达全国甚至全球各地，从而拓展市场，增加销售额。



（4）设计基于 JAVA 的网上水果商城可以进一步推动电商行业的发展，提升整个行业的效率和竞争力。该商城设计采用当下最流行的 SpringBoot 与 Vue 框架实现，极大的提高了开发效率，并保证安全性和可读性。

## 1.2 研究内容

本文旨在探讨利用 SpringBoot 与 Vue 框架来设计与实现网上水果商城。具体研究内容包括以下方面：

系统设计：研究包括前端、后端、系统模块、数据流和交互式规划。

界面设计：研究将关注用户界面的设计，确保用户友好性和易用性。这包括注册、登录、浏览商品、管理购物车、下单支付等功能的界面设计和交互体验。

后端数据管理与交互：研究将聚焦于后端系统的设计和实现，包括数据库的设计与管理、后端逻辑，前后端交互。

系统核心功能：注册、登录、商品浏览与购买；商品管理、会员管理、系统管理。

## 2 相关技术介绍

### 2.1 开发框架

#### 2.1.1 SpringBoot 框架

本系统采用 Spring Boot 框架，它简单、高效和可靠。

#### 2.1.2 Vue 框架

前端采用 Vue 框架，它支持模板语法、组件化开发、响应式数据绑定，并拥有丰富的生态系统和逐步应用的特性。

### 2.2 开发语言

#### 2.2.1 JAVA 语言

Java 是面向对象的高级编程语言，其跨平台特性使得开发者能够编写一次代码，在任何支持 Java 虚拟机（JVM）上运行，能够满足多样化的部署需求<sup>[1]</sup>。

#### 2.2.2 JSON 语言

JSON 是一种轻量级的数据交换格式，它采用键值对的方式表示数据，易于理解和编写，同时具有很好的跨平台性。

### 2.3 开发工具

#### 2.3.1 MySQL 数据库

MySQL 是一个开放源码、体积小、速度快、成本低、易用且灵活的关系型数据库管理系统，在中小型 web 平台开发中广泛应用。

#### 2.3.2 Vscode

Vscode 用于该系统的前端开发，同时它适用于各种编程任务，拥有活跃社区和丰富扩展生态系统，是许多开发者的首选工具。

#### 2.3.3 IntelliJIDEA

IDEA 用于该系统的后端开发，它具有智能代码编辑、强大的功能和丰富的插件生态系统，是开发 Java 等多种语言项目的首选工具。

#### 2.3.4 Hbuilder

HBuilder 是由中国软件公司迅捷通信（Egret Technologies）开发的集成开发环境（IDE），专注于 Web 开发、移动应用开发和跨平台应用程序开发，提供丰富的功能和工具，让开发过程更加便捷高效。

### 3 系统需求

本章主要进行网上水果商城系统的需求分析和概要设计。需求分析包括对功能、性能和数据进行详细分析，概要设计则涵盖总体架构、模块设计和数据库设计<sup>[2]</sup>。

#### 3.1 需求分析与概要设计

##### 3.1.1 需求分析

初步分析表明，系统的用户主要分为普通消费者和水果出售商家。

消费者需求：

- （1）注册与登录：提供给普通消费者注册和登录功能，并确保普通消费者的账户安全。
- （2）浏览：用户可以查看商品，并根据不同的分类筛选。
- （3）搜索：为用户提供搜索功能，使用户能够简单快捷的找到自己当前所需的水果商品，节省大量时间与精力。
- （4）购物车：用户可以将产品添加到购物车中并管理购物车。
- （5）下单和支付：提供简便的下单流程和安全的支付选项。
- （6）订单历史：用户能够查看以前的订单记录，追踪订单状态。
- （7）界面友好：设计简洁、直观的用户界面，提供良好的用户体验。

商家需求：

- （1）商品管理：商家可以对商品进行添加、编辑和删除操作，管理商品的库存数量和价格设置<sup>[3]</sup>。
- （2）订单管理：查看和处理订单。
- （3）库存管理：查看与更改库存信息。
- （4）促销活动：商家可以设置促销活动，如打折、满减等。
- （5）数据统计：提供销售数据统计和报表，帮助商家了解销售状况。

### 3.1.2 概要设计

该论文旨在建立和推出便利的在线水果交易平台，服务于消费者和商家。消费者可以在手机或电脑等设备上快速访问该交易平台，挑选心仪的商品进行购买。商家可以登录后台管理系统，对商品信息进行管理，以便于商品顺利销售。系统的用户角色包括游客、会员、商家、以及系统管理员，每个角色都具有特定的特点和权限<sup>[4]</sup>，详见表 3.1。

表 3.1 角色的特点和权限

角色	特点和权限
游客	可以浏览商品信息，但无法进行购买或管理个人信息
会员	可以浏览商品信息，进行购买，并管理个人订单信息
商家	可以管理商品信息，包括新增、修改、删除商品，管理订单和供应情况
系统管理员	拥有全部权限

4 系统的详细设计与实现

4.1 系统开发环境与实现框架

4.1.1 系统开发环境

系统开发环境详见表 4.1:

表 4.1 工具版本

工具	版本
jdk	17
mysql	5.7+
redis	4.0+
Node.js	14-16

4.1.2 数据库设计

数据库整体 E-R 图如图 4.2:

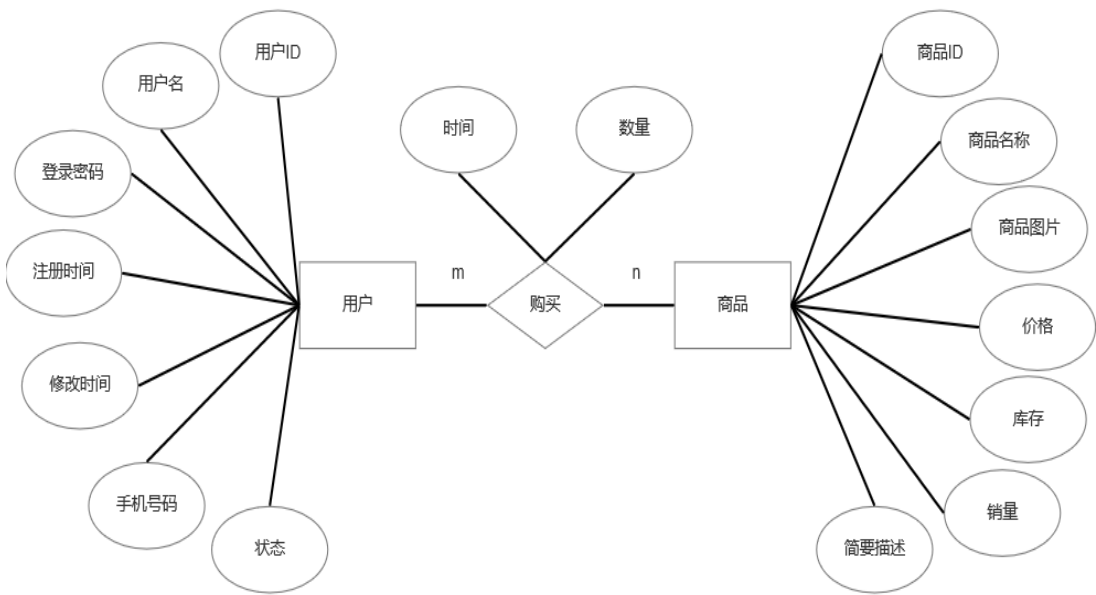


图 4.2 数据库整体 E-R 图

管理员实体属性如图 4.3:

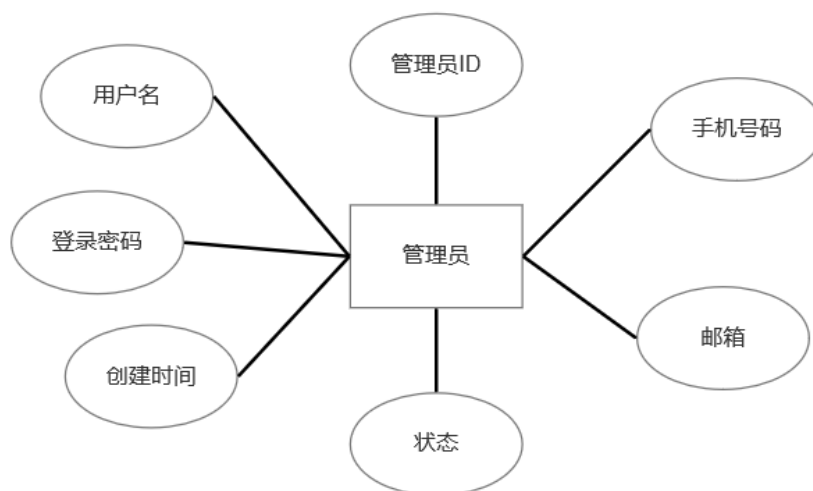


图 4.3 管理员实体图

数据库表有四张：用户表、管理员表、商品表和订单表。

#### (1) 用户表

用户表记录用户 ID、用户名、登录密码、手机号等信息

表 4.4 用户表

名称	类型	长度	非空	主键	说明
user_id	varchar	50	是	是	ID
nick_name	varchar	50	否	否	用户名
login_password	varchar	50	否	否	登录密码
user_regtime	varchar	50	是	否	注册时间
modify_time	varvher	50	是	否	修改时间
user_mobile	varvher	50	否	否	手机号码
status	int	1	是	否	状态

#### (2) 管理员表

管理员表记录管理员 ID、用户名、登录密码、邮箱等信息。

表 4.5 管理员表

名称	类型	长度	非空	主键	说明
user_id	varchar	50	是	是	ID
user_name	varchar	50	否	否	用户名
password	varchar	50	否	否	登录密码
email	varchar	50	是	否	邮箱
create_time	varvher	50	是	否	创建时间
mobile	varvher	50	否	否	手机号码
status	int	1	是	否	状态

## (3) 商品表

商品表记录商品 ID、商品名称、店铺 ID、价格等信息。

表 4.6 商品表

名称	类型	长度	非空	主键	说明
pord_id	varchar	50	是	是	商品 ID
pord_name	varchar	50	否	否	商品名称
price	varchar	50	否	否	价格
brief	varvher	50	否	否	简要描述
imgs	varchar	50	否	否	商品图片
status	varchar	50	否	否	状态
sold_num	varchar	50	否	否	销量
total_stocks	varchar	50	否	否	总库存
create_time	varvher	50	否	否	录入时间
category_id	varvher	50	否	否	商品分类

## (4) 订单表

订单表记录订单 ID、店铺 ID、订购用户 ID、产品名称等信息。



表 4.7 订单表

名称	类型	长度	非空	主键	说明
order_id	varchar	50	是	是	订单 ID
user_id	varchar	50	是	否	用户 ID
prod_id	varchar	50	是	否	商品 ID
order_number	varchar	50	否	否	订购流水号
total	varchar	50	否	否	总值
remarks	varchar	50	否	否	订单备注
status	varchar	50	否	否	状态
product_nums	varchar	50	否	否	订单商品总数
create_time	varchar	50	否	否	创建时间
status	varchar	50	否	否	状态
dvy_flow_id	varchar	50	否	否	物流单号
freight_amount	varchar	50	否	否	订单运费
addr_order_id	varchar	50	是	否	用户订单地址 ID

数据库设计代码示例：

```

@Data
@TableName("tz_prod")
public class Product implements Serializable {
    /*商品 ID*/
    @TableId
    private Long prodId;
    /* 店铺 id*/
    private Long shopId;
    /* 商品名称*/
    private String prodName;
    /* 简要描述,卖点等*/
    private String brief;
    /* 商品主图*/
    private String pic;
    /*商品图片*/
    private String imgs;
    /* 默认是 1, 表示正常状态, -1 表示删除, 0 下架*/
    private Integer status;
    /* 商品分类*/
    private Long categoryId;
    /*已经销售数量*/
    private Integer soldNum;
    /* 录入时间*/
    private Date createTime;
    /* 修改时间*/
    private Date updateTime;
    /* 详细描述*/
    private String content;
    /* 上架时间*/
    private Date putawayTime;
    @Data
    public static class DeliveryModeVO {
        /* 用户自提*/
        private Boolean hasUserPickUp;
        /* 店铺配送*/
        private Boolean hasShopDelivery;
    }
}

```

### 4.1.3 系统实现架构

在构建这个系统时，我们采用了 SpringBoot 和 vue 框架，并使用 Java 语言进行开发<sup>[5]</sup>。后台实现架构可分为四层：拦截器过滤器、控制层、服务层和 Dao 层。如图 4.8 所示：

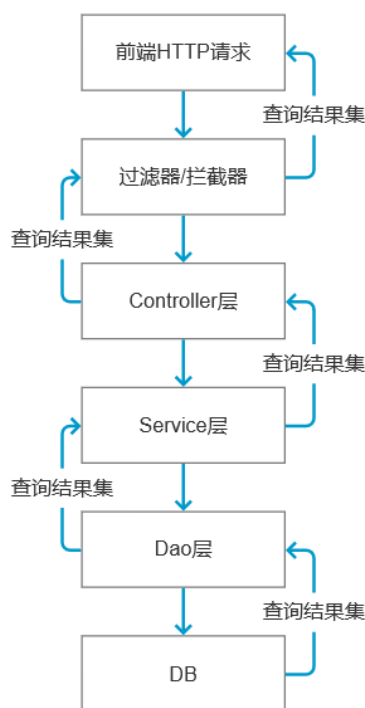


图 4.8 系统架构图

用户访问页面，请求经过过滤器层进行安全检查，再传输给 Controller 层进行业务逻辑处理,Service 层对业务逻辑进行封装，Dao 层对数据库的运行进行处理。

## 4.2 系统详细设计

系统详细设计图如 4.9 所示

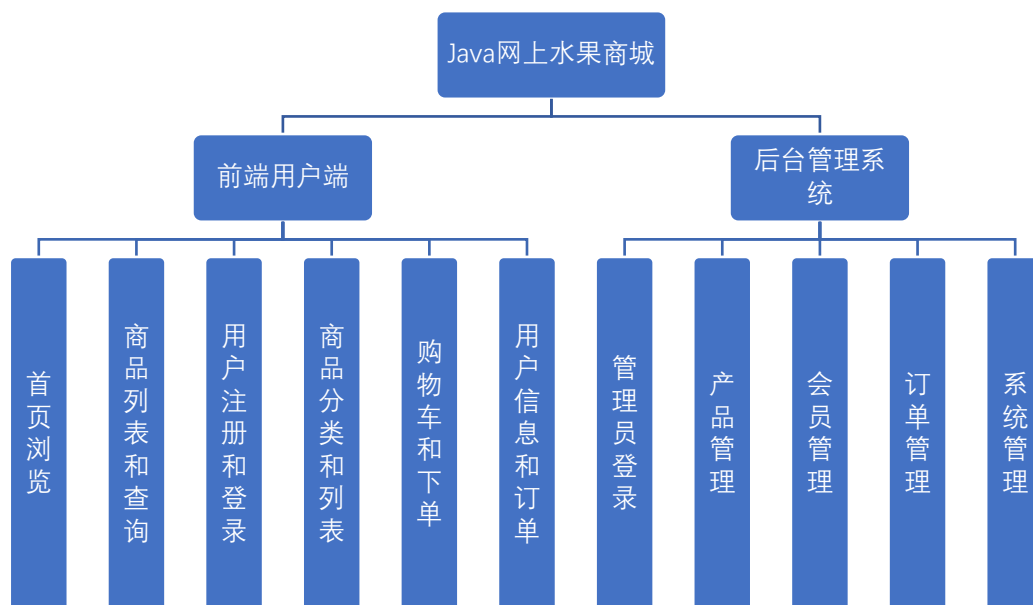


图 4.9 系统详细设计图

经过需求分析，对水果购物系统的用户特性和功能需求进行了详细分析，确定了适合 B/S 架构的优点：

（1）硬件：用户只需要一个浏览器访问，无需高系统硬件，降低了用户门槛，增加了系统的可访问性。

（2）系统：B/S 架构稳定可靠，成本低廉，适用于中小型网站，确保系统稳定性。

（3）界面：界面更美观、动态，具有更好的用户体验。

水果商城系统总体架构分前端与后端。前端方面由用户层与表示层组成，后端由控制器层，业务逻辑层以及数据层组成<sup>[6]</sup>。

水果商城系统的各个功能单元可以根据用户需求和整体结构进行分析，分为前台购物系统和后台管理系统。

（1）前台功能如下表 4.10：

表 4.10 前台功能

模块	功能
个人中心模块	用户个人信息管理、订单管理、收货地址管理等。
商品浏览模块	用户可以浏览各类商品信息，包括商品详情、价格、库存等。
购物车管理模块	用户可以将感兴趣的商品加入购物车，并进行管理。
商品分类模块	按照商品类型进行分类展示，方便用户查找和选择商品。

(2)后台功能如下表 4.11:

表 4.11 后台功能

模块	功能
商品管理模块	管理员可以对商品进行添加、修改、删除、管理库存，价格等操作。
商城管理模块	设置商城的基本信息、页面布局管理等。
订单管理模块	处理用户下单、支付、发货等订单相关操作。
物流管理模块	管理物流信息，包括物流公司合作管理、订单配送跟踪等。
用户管理模块	管理用户信息、权限控制等。

以上为水果商城功能的整体设计，能够满足使用者和管理员的需求，提供较

好的使用体验和管理功能，符合系统架构要求，确保水果商城系统的顺畅运行。

#### 4.2.1 前台购物系统

前台购物系统可实现商品信息展现、商品分类展示、根据商品名称搜索商品、添加商品至购物车、选择商品下单付款等功能。用户可在首页点击商品，进入详情页，查看商品描述、价格、图片等内容。用户如果有青睐的商品，还可以将其添加至购物车，以方便后续购买等操作。商品分类将大量商品进行有序的组织 and 分类，以便于用户进行挑选。此外，用户还可以进行搜索。界面如图 4.12：

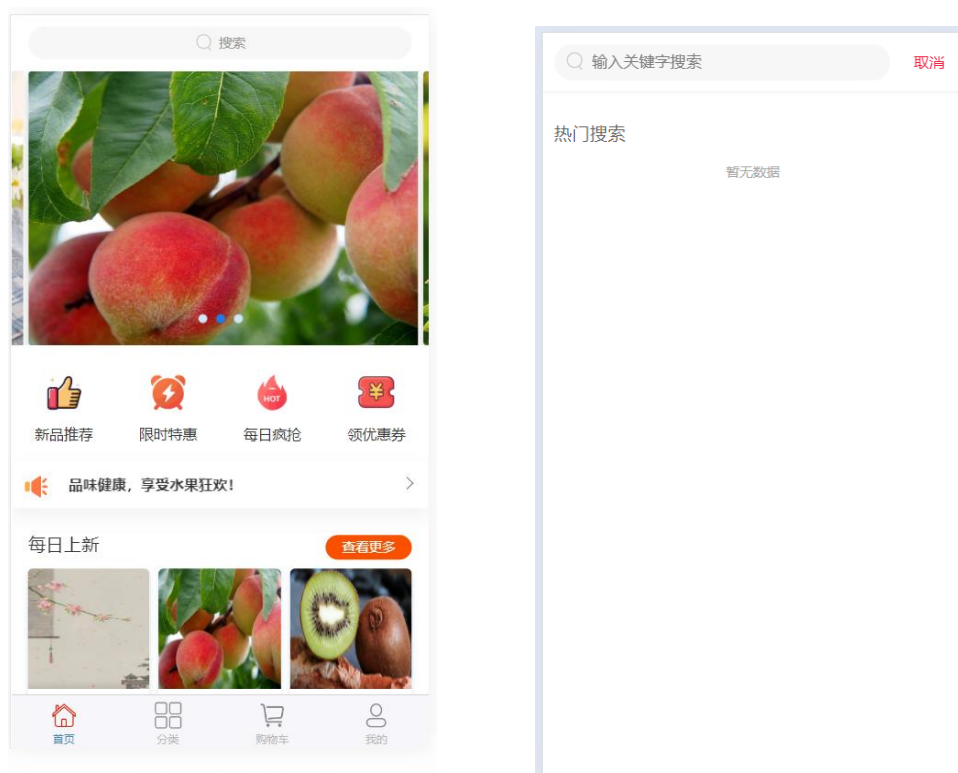


图 4.12 前台系统首页

通过轮播图、列表渲染等方式将页面展示出来，首页代码示例如下：

```

<swiper :autoplay="autoplay" :indicator-color="indicatorColor"
:interval="interval" :duration="duration"
:indicator-active-color="indicatorActiveColor + ' '" circular="true"
class="pic-swiper" indicator-dots previous-margin="20rpx"
next-margin="20rpx">
  <block v-for="(item, index) in indexImgs" :key="index">
    <swiper-item class="banner-item">
      <view class="img-box">
        <image :src="item.imgUrl" :data-prodid="item.relation" @tap="toProdPage"
class="banner"></image> </view>
      </swiper-item> </block>
</swiper>
<!-- end swiper -->
<view class="cat-item">
  <view class="item" @tap="toClassifyPage" data-sts="1">
    <image src="/static/images/icon/newProd.png"></image>
    <text>新品推荐</text>
  </view>
  <view class="item" @tap="toClassifyPage" data-sts="1">
    <image src="/static/images/icon/timePrice.png"></image>
    <text>限时特惠</text>
  </view>
  <view class="item" @tap="toClassifyPage" data-sts="3">
    <image src="/static/images/icon/neweveryday.png"></image>
    <text>每日疯抢</text>
  </view>
  <view class="item" @tap="toCouponCenter">
    <image src="/static/images/icon/newprods.png"></image>
    <text>领优惠券</text>
  </view>
</view>

```

商品浏览模块是水果商城系统的重要组成部分之一，该模块为用户提供了浏览和选择心仪商品的功能。用户能通过该模块浏览商品详细信息，包括商品图片、文字描述、规格参数，价格信息等<sup>[7]</sup>。通过展示商品的图片和文字描述，用户能够充分了解商品的外观和特点，帮助用户对商品进行价格比较和选择。

该系统会根据商品的不同类别进行分类展示，例如热带水果、桃李杏梨、菩提浆果等。界面如图 4.13：



图 4.13 商品分类

通过分类展示，用户可以快速找到感兴趣的商品类别。通过 `onMenuTab()` 方法加载分类和子分类列表，当用户点击子分类后调用 `toCatePage()` 方法实现将当前页面跳转至该子分类页面，从而提高浏览效率。核心代码如下：



```

onMenuTab: function (e) {
    var id = e.currentTarget.dataset.id;
    var index = e.currentTarget.dataset.index; // this.getProdList(id);
    this.getProdList(this.categoryList[index].categoryId);
    this.parentId = this.categoryList[index].categoryId
    this.setData({
        categoryImg: this.categoryList[index].pic,
        sellIndex: index }); }
.getProdList(categoryId) {
    //加载分类列表
    var params = {
        url: "/category/categoryInfo",
        method: "GET",
        data: {
            parentId: categoryId },
        callBack: res => {
            this.setData({
                subCategoryList: res }); }
    };
    http.request(params); },
    // 跳转子分类商品页面
toCatePage: function(e) {
    const {
        parentid,
        categoryid,
    } = e.currentTarget.dataset;
    uni.navigateTo({
        url: ` /pages/sub-category/sub-category?parentId
        =${this.parentId}&categoryId=${categoryid}`
    })
}
}

```

用户能便捷的浏览和选择商品，还能利用该模块将心仪商品添加至购物车，并对购物车中的商品进行管理，包括增加数量、删除、全选等操作<sup>[8]</sup>。使用户能够便利的整理和管理自己的购物清单，确保购物流程的顺利进行。购物车功能为用户提供了更灵活和个性化的购物体验。如图 4.14:

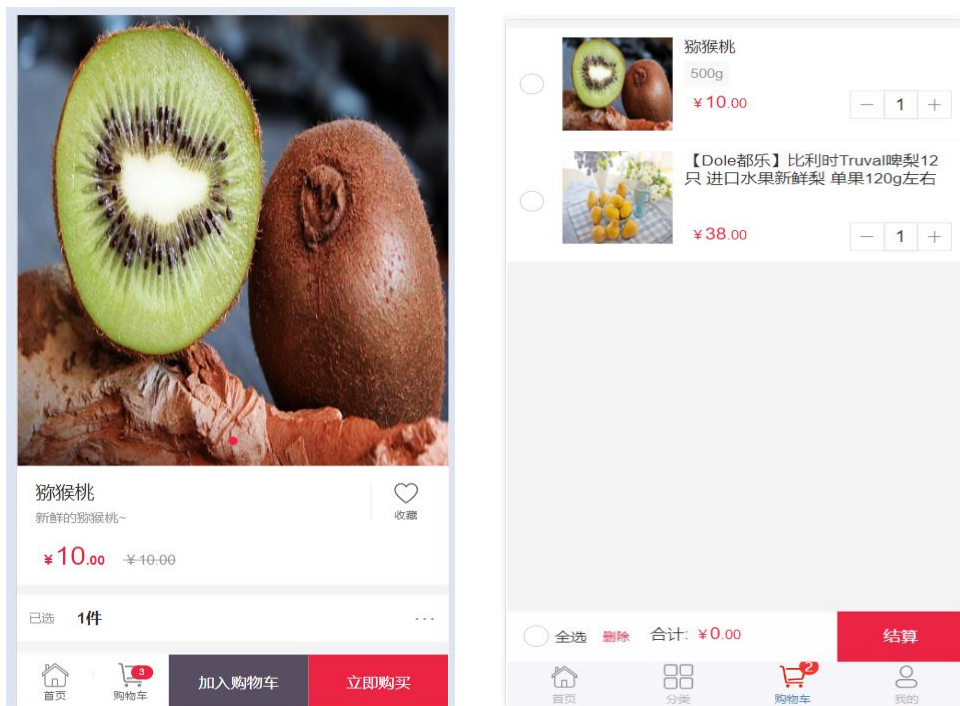


图 4. 14 购物车

通过调用 `this.loadBasketData()`和 `http.getCartCount()`等方法,可对购物车中的商品进行添加数量、选中、删除、全选、结算等操作。核心代码如下:

```
onShow: function () {  
  this.loadBasketData(); http.getCartCount();  
  methods: {  
    loadBasketData(){ uni.showLoading(); //加载购物车  
    var params = {  
      url: "/p/shopCart/info", method: "POST", data: {},  
      callBack: res => {  
        if (res.length > 0) { // 默认不选中  
          var shopCartItemDiscounts = res[0].shopCartItemDiscounts;  
          shopCartItemDiscounts.forEach(shopCartItemDiscount => {  
            shopCartItemDiscount.shopCartItems.forEach(shopCartItem => {  
              shopCartItem.checked = false; });  
            });  
          this.setData({  
            shopCartItemDiscounts: shopCartItemDiscounts, allChecked: false });  
        } else { this.setData({  
          shopCartItemDiscounts: [] });  
          this.calTotalPrice(); //计算总价 uni.hideLoading();  
        } };http.request(params);  
      }  
    }  
  }  
}
```

个人中心模块是用户进行订单和收货地址管理的功能区域。如图 4.15:

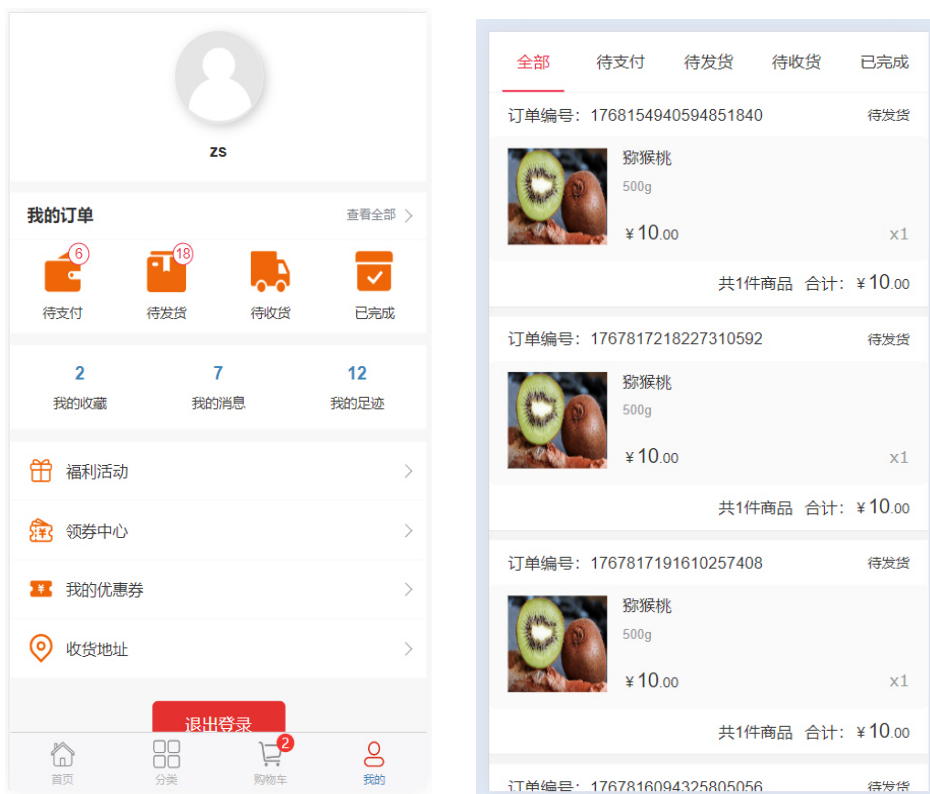


图 4.15 个人中心

通过调用订单查询接口:

```
toOrderListPage: function(e) {  
    var sts = e.currentTarget.dataset.sts;  
    uni.navigateTo({  
        url: '/pages/orderList/orderList?sts=' + sts  
    });  
},
```

再调用接口获取到的数据进行渲染，可以将订单状态、历史订单、订单详情等信息展示在页面上。

```

<view class="total-order">
  <view class="order-tit">
    <text style="font-weight:bold">我的订单</text>
    <view class="checkmore" @tap="toOrderListPage" data-sts="0">
      <text>查看全部 </text>
      <text class="arrowhead"></text>
    </view>
  </view>
</view>
<view class="procedure">
  <view class="items" @tap="toOrderListPage" data-sts="1">
    <image src="/static/images/icon/toPay.png"></image>
    <text>待支付</text>
    <text class="num-badge" v-
      if="orderAmount.unPay>0">{{orderAmount.unPay}}</text>
  </view>
  <view class="items" @tap="toOrderListPage" data-sts="2">
    <image src="/static/images/icon/toDelivery.png"></image>
    <text>待发货</text>
    <text class="num-badge" v-
      if="orderAmount.payed>0">{{orderAmount.payed}}</text>
  </view>
  <view class="items" @tap="toOrderListPage" data-sts="3">
    <image src="/static/images/icon/toTake.png"></image>
    <text>待收货</text>
    <text class="num-badge" v-
      if="orderAmount.consignment>0">{{orderAmount.consignment}}</text>
  </view>
  <view class="items" @tap="toOrderListPage" data-sts="5">
    <image src="/static/images/icon/toComment.png"></image>
    <text>已完成</text>
  </view>
</view>
</view>

```

用户在收货地址中进行收货地址管理，如图 4.16：

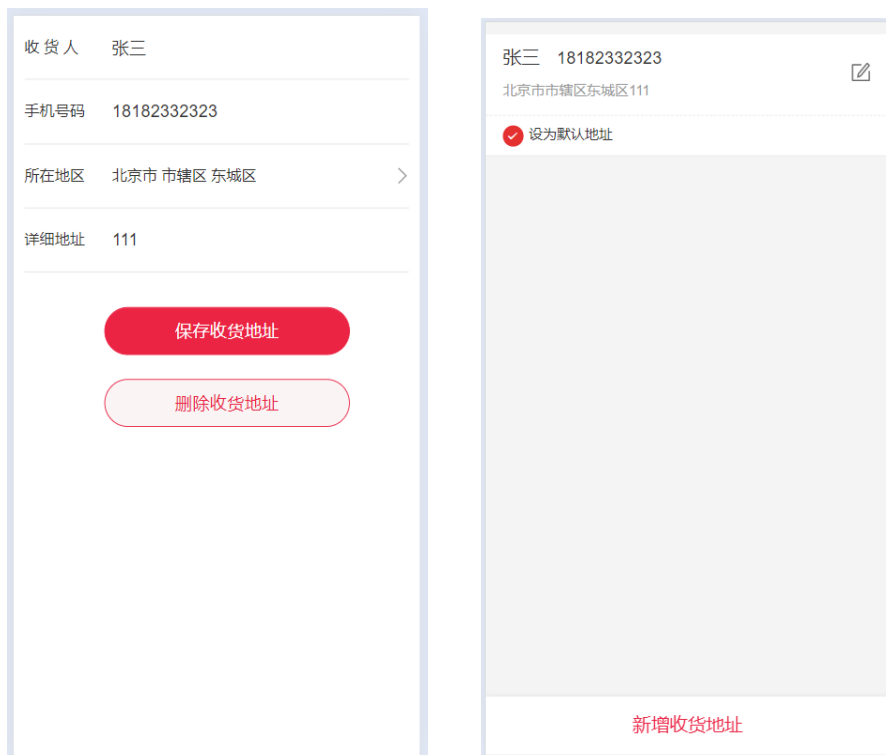


图 4.16 收货地址

通过调用新增和修改地址的接口，可以使用户在收货地址管理中添加、修改收货地址，方便进行订单配送，核心代码如下：

```
//新增收货地址
onAddAddr: function (e) {
  uni.navigateTo({
    url: '/pages/editAddress/editAddress'
  });
},
// 修改地址
toEditAddress: function (e) {
  var addrId = e.currentTarget.dataset.addrId;
  uni.navigateTo({
    url: '/pages/editAddress/editAddress?addrId=' + addrId
  });
},
```

4.2.2 后台管理系统

后台管理系统截图如图 4.17：

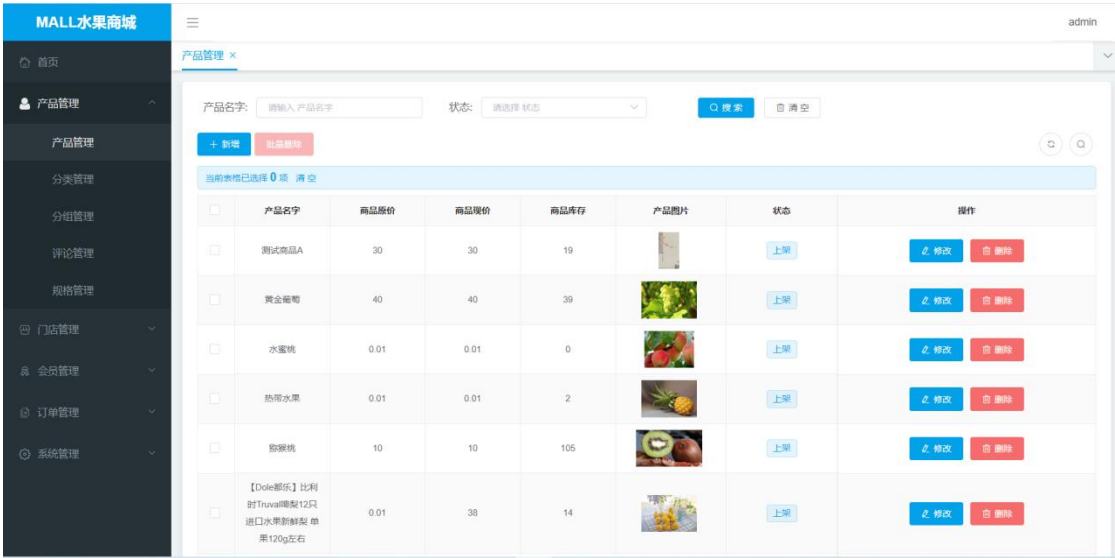


图 4.17 后台管理系统

在产品管理模块中，管理员拥有对产品的全面管理权限，能够对产品进行产品详细信息管理，包括产品名称、描述、价格、库存、类别等信息的管理，还能对产品进行分类、分组和规格的管理<sup>[9]</sup>，以便更好地组织和展示产品信息。管理员可通过添加、编辑和删除操作来更新产品列表和相关信息。管理员还能根据当前需要对产品进行搜索，从而便捷快速的找到所需的产品。通过产品管理模块，管理员将会实现对产品数据的集中管理和有效控制，提高工作效率并确保产品信息的准确性和完整性。用户也会更清晰地了解产品的不同属性，从而提升用户体验。产品管理界面代码示例：

```

<!-- 产品管理页面 -->
<template>
  <div class="mod-prod">
    <avue-crud ref="crud"
      :page="page" :data="dataList" :table-loading="dataListLoading"
      :permission="permission" :option="tableOption" @search-
      change="searchChange" @selection-change="selectionChange"
      @on-load="getDataList">
      <template slot="menuLeft">
        <el-button type="primary" icon="el-icon-plus" size="small"
          v-if="isAuth('shop:pickAddr:save')"
          @click.stop="addOrUpdateHandle()">新增</el-button>
        <el-button type="danger" @click="deleteHandle()" size="small"
          v-if="isAuth('shop:pickAddr:delete')"
          disabled="dataListSelections.length <= 0">批量删除</el-button>
      </template>
      <template slot-scope="scope"
        slot="status">
        <el-tag v-if="scope.row.status === 1"
          size="small">上架</el-tag>
        <el-tag v-else
          size="small">未上架</el-tag>
      </template>
      <template slot-scope="scope" slot="menu">
        <el-button type="primary" icon="el-icon-edit"
          size="small" v-if="isAuth('prod:prod:update')"
          @click="addOrUpdateHandle(scope.row.prodId)" 修 改</el-button>
        <el-button type="danger" icon="el-icon-delete" size="small"
          v-if="isAuth('prod:prod:delete')"
          @click="deleteHandle(scope.row.prodId)">删 除 </el-button>
      </template></avue-crud> </div>
    </template>
  </template>

```

管理员可以利用会员管理模块来完成查看会员信息、编辑和删除操作。此外，该模块还提供搜索功能，管理员可以依据会员昵称检索出所需的会员数据。如图



4.18:

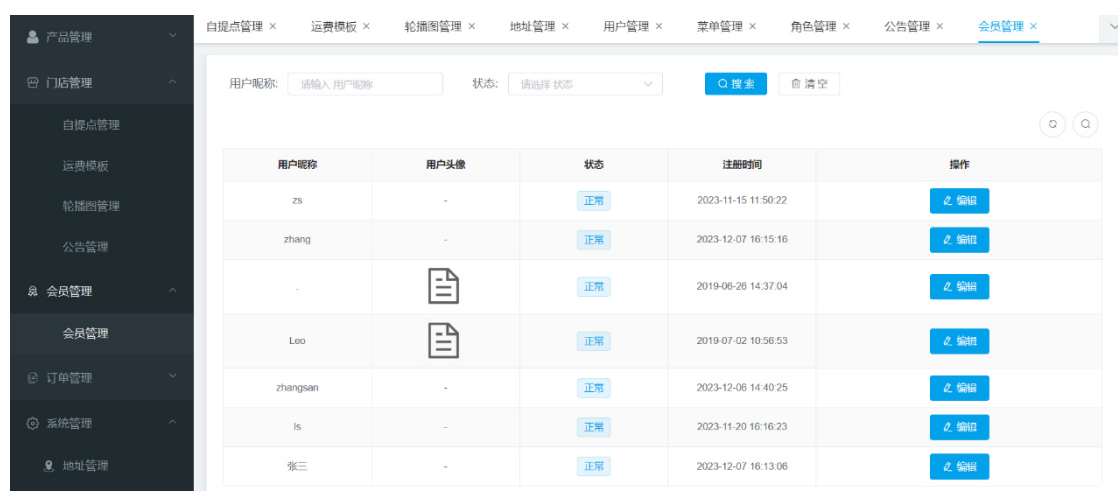


图 4.18 会员管理

会员信息的列表展示能让管理员一目了然，轻松地浏览和管理会员信息。若用户存在不安全的行为，管理员还能禁用会员，及时阻止其危险行为，其核心逻辑如下：

```

dataFormSubmit: Debounce(function () {
  this.$refs['dataForm'].validate(valid => {
    if (valid) {
      this.$http({
        url: this.$http.adornUrl(`/admin/user`),
        method: this.dataForm.userId ? 'put' : 'post',
        data: this.$http.adornData({
          userId: this.dataForm.userId || undefined,
          nickName: this.dataForm.nickName,
          status: this.dataForm.status
        })
      }).then(({ data }) => {
        this.$message({
          message: '操作成功', type: 'success', duration: 1500,
          onClose: () => {
            this.visible = false this.$emit('refreshDataList', this.page)
          }
        })
      })
    }
  })
})

```

订单管理模块可以让管理员便捷的看到当前全部订单的状态、配送信息和支付状态等主要信息。如图 4.19:



图 4.19 订单管理界面

通过该界面，管理员很容易就能看到订单当前的状态并知道其支付与配送情况，另外管理员可将快递公司、快递单号填入订单管理模块实现货物发货操作。发货界面如下 4.20：



图 4.20 发货界面

通过这个模块，管理员可以轻松地管理和追踪订单的状态，并及时处理发货操作，为顾客提供良好的购物体验。核心代码如下：

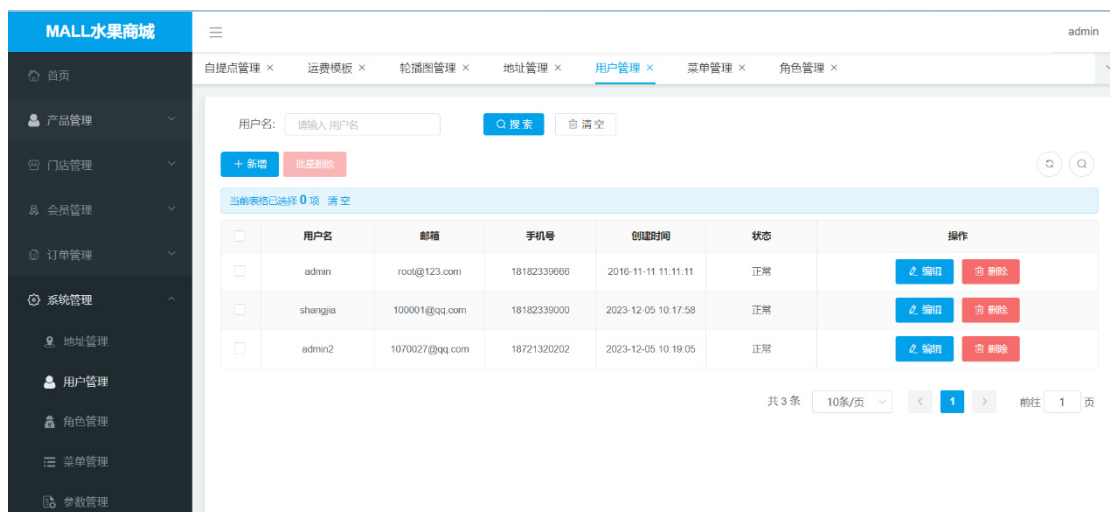
```

dataFormSubmit () {
  this.$refs['dataForm'].validate((valid) => {
    if (valid) {
      this.$http({
        url: this.$http.adornUrl('/order/order/delivery'),
        method: 'put',
        data: this.$http.adornData({
          'orderNumber': this.dataForm.orderNumber,
          'dvyld': this.dataForm.dvyld,
          'dvyFlowId': this.dataForm.dvyFlowId
        })
      }).then(({data}) => {
        this.$message({
          message: '操作成功', type: 'success', duration: 1500,
          onClose: () => {
            this.visible = false this.$emit('refreshDataList')
          }) })
      })
    }
  })
}

```

系统设置模块为用户提供用户管理、角色管理等功能。如图 4.21:

图 4. 21 用户管理



管理员可通过用户管理功能实现对用户信息的完全管理，如添加新用户、编

辑用户信息、删除用户，以维护用户信息的完整性和安全性。核心代码示例：

```
// 新增 / 修改
addOrUpdateHandle (id) {
  this.addOrUpdateVisible = true
  this.$nextTick(() => {
    this.$refs.addOrUpdate.init(id)
  })
},
```

通过角色管理功能，管理员可以分配不同的角色给用户，并管理用户的角色权限，以确保权限分配符合系统规定和安全要求。此外，参数管理功能允许管理员管理系统参数信息，包括系统配置和邮件配置等，以及时调整系统参数信息，确保系统稳定和安全运行。系统日志功能记录管理员的操作日志，帮助追踪管理操作记录，有助于提高系统的安全性和管理人员操作的审计。这些功能的存在提供了便捷的系统设置和管理，确保系统的正常运行和安全性。

## 5 系统测试

软件测试的步骤通常包括需求分析、测试计划制定、测试用例设计、测试环境搭建、测试执行、缺陷跟踪和修复、测试报告编写和总结等<sup>[10]</sup>。

在进行软件测试时，首先，基于需求文档和设计文档进行深入的需求解析，以便准确理解软件的预期功能和性能。接着，根据需求分析制定详细的测试策略，并基于此编写详细的测试用例，确保覆盖所有功能点和边界条件。搭建稳定且贴近实际环境的测试环境是至关重要的一步，它为执行测试用例提供了坚实的基础。执行测试用例时，需要仔细记录每个步骤的执行情况与结果，尤其注意出现的缺陷或异常。一旦发现缺陷，应立即进行跟踪和管理，直到成功修复。完成测试后，编写详尽准确的测试报告。

经过各方面考虑，决定对此系统采用黑盒测试方法，同时，还对其进行功能测试和性能测试。通过综合运用这些方法，可以确保软件的功能完整性和性能稳定性。

### 5.1 前台购物系统测试

测试对于一个软件来说是必不可少的一部分。本系统遵照测试原则，即尽早测试、IPO 以及独立测试原则对本系统进行测试。测试时采用了黑盒测试法对平台的功能进行测试，用以检验系统的功能是否可以正常使用，在登录界面，如图 5.1:



图 5.1 登录界面

进行登录测试，测试详情如下表 5.2：

表 5.2 登录测试表

用户名	密码	预期输出	实际输出
zs	123456	登录成功	登录成功
1	1	登录失败	登录失败
ls	111111	登录成功	登录成功

账号和密码输入正确后可正常登录跳转到首页，如图 5.3：

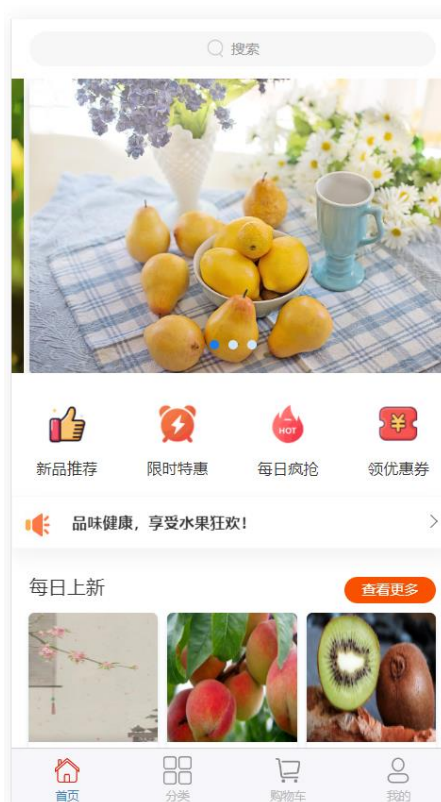


图 5.3 首页界面

## 5.2 后台管理系统测试

商品信息界面如图 5.4:



图 5.4 商品管理页面



对其进行商品操作测试，如下表 5.5：

表 5.5 商品修改测试表

商品 ID	操作	预期输出	实际输出
Sp0001	删除	删除成功	删除成功
Sp0002	修改名称为空	修改失败	修改失败
Sp0003	修改分类	修改成功	修改成功

经以上测试，该系统无程序漏洞，且功能完善，运行流畅。

## 6 总结与展望

该网上水果商城系统采用了 Vue、SpringBoot、MyBatis 和 Redis 作为开发技术栈，实现了前后端分离的架构，具备完整下单流程。该系统具备良好的扩展性和性能优势，可适用于生产环境的多实例部署。未来可继续优化系统的用户体验和交互设计，引入更多的商城功能和营销策略，提供用户更友好的购物体验。同时，为了提升系统的安全性和稳定性，我们可以根据用户的反馈和建议，以及市场的需求，持续地优化和升级系统的功能，满足不断变化的电子商务平台的需求。

## 参考文献

- [1] 王建雄.基于云计算的终端管理系统研究及其实现[D].电子科技大学,2013.
- [2] 董宁.法人客户信贷资产分类管理系统的设计与实现[D].山东大学,2015.
- [3] 张慧勇,郭小燕.基于个性化服务的农产品商城设计与实现[J].自动化与仪器仪表,2016,(05):157-159.
- [4] 徐喆.基于 Web 的大剧院演出剧目管理系统的设计与实现[D].电子科技大学,2013.
- [5] 韩露.基于深度学习的企业关系抽取系统的设计与实现[D].北京邮电大学,2020.DOI:10.26969/d.cnki.gbydu.2020.002370.
- [6] 耿庆阳.基于 Spring Boot 与 Vue 的电子商城设计与实现[D].西安石油大学,2020.DOI:10.27400/d.cnki.gxasc.2020.000569.
- [7] 樊涛.基于数据挖掘的网上书店的电子商务网站设计与实现[D].江西农业大学,2014.
- [8] 梁君权.基于第三方支付平台的电子商务系统的研究与实现[D].南昌大学,2015.
- [9] 王斌.大连三林木业网站的设计与实现[D].吉林大学,2009.
- [10] 陈苏红,魏银珍,孙萍.基于项目驱动的软件测试课程教学研究与实践[J].福建电脑,2016,32(08):165-166.