优果汇生鲜农电商后台管理系统设计与实现

邱泽楷 赵楷迪 邓子林 刘向阳 张颖婷

(广州新华学院信息与智能工程学院 广东 东莞 523133)

摘 要 消费者在线购物推动了生鲜农产品电商市场的快速发展。为了满足生鲜农产品电商的需求，本文设计并实现了一个电商后台管理系统。该系统基于 SpringBoot 和 MyBatis 的后端架构，以及 Vue.js 和 ElementUI 的前端架构，具有订单管理、商品管理、促销管理、运营管理和用户管理等多个模块。实验测试证明，该系统具有良好的性能和用户友好的界面，可以有效地促进生鲜农产品电商的发展。

关键词 生鲜农产品；开发框架；管理系统

中图法分类号 TP31 DOI:10.16707/j.cnki.fjpc.2024.02.018

Design and Implementation of Youguohui Fresh Agriculture E Commerce Backend Management System

QIU Zekai, ZHAO Kaidi, DENG Zilin, LIU Xiangyang, ZHANG Yingting

(School of Information And Intelligent Engineering, Guangzhou Xinhua University, Dongguan, China, 523133)

Abstract Consumer online shopping has driven the rapid development of the e-commerce market for fresh agricultural products. In order to meet the needs of fresh agricultural product e-commerce, this article designs and implements an e-commerce backend management system. The system is based on the backend architecture of SpringBoot and MyBatis, as well as the front-end architecture of Vue.js and ElementUI. It has multiple modules such as order management, product management, promotion management, operations management, and user management. Through experimental testing, it has been proven that the system has good performance and user-friendly interface, which can effectively promote the development of fresh agricultural product e-commerce.

Keywords Fresh Agricultural Products; Development Framework; Management System

1 引言

随着全球疫情影响和人们对健康食品的需求不断增加，生鲜农产品的电商销售成为了农产品行业的一项重要趋势。传统的农产品销售模式面临着诸多挑战，包括供应链的不透明性、信息不对称、物流配送的不便等问题，限制了农产品的市场开拓和销售效率的提升[1]。

为了解决这些问题并推动农产品电商化的发展，设计和开发一套适合生鲜农产品的电商后台管理系统以提升农产品销售效率和优化供应链管理具有重要意义[2]。为此，本文对电商后台管理系统作设计和开发，主要的研究内容有：（1）系统架构设计；（2）前端界面设计；（3）商品管理与展示；（4）订单管理与支付；（5）物流配送与跟踪；（6）数据分析与营销策略。

2 技术选型

2.1 SpringBoot

SpringBoot 是基于 Spring 框架的轻量级开发框架，可以帮助开发者快速构建和部署 Spring 应用程序。它具有自动配置和约定优于配置等特点，简化

本文得到广东省大学生创新创业训练计划项目(No.202213902064)资助。邱泽楷(通信作者)，男，2002年生，主要研究领域为计算机科学与技术。E-mail: ben43@foxmail.com。

了 Spring 应用程序的配置和开发。同时 SpringBoot 通过各种注释或配置文件提供统一的方式来配置应用程序中的各个组件，例如数据库连接、日志记录和安全性等。这使得开发者可以快速入门，无需花费时间在复杂的配置上[3]。

2.2 MyBatis

MyBatis 是一款流行的持久化框架，可以帮助开发者将 Java 对象映射到关系数据库，并提供了一种灵活的方式来执行 SQL 查询。MyBatis 易于使用，提供了很多功能，如缓存、动态 SQL 生成和事务管理。同时 MyBatis 使用 XML 或注释来定义 SQL 查询和 Java 对象与数据库表之间的映射关系。这种关注点分离使得数据访问层的维护和演进更容易[4]。

2.3 Vue.js

Vue.js 是一款用于构建用户界面的渐进式 JavaScript 框架。它专注于视图层，并提供了一组工具来构建可重用的组件、管理应用程序的状态和处理事件。同时 Vue.js 提供了强大的响应式系统。这意味着当底层数据发生变化时，视图层将实时更新，在构建复杂和动态的用户界面时会更加容易。Vue.js 还拥有庞大而活跃的社区，提供了丰富的插件库，具有良好的扩展性[5]。

3 优果汇系统设计

系统需具备的功能主要有：（1）商品管理。管理员可以查看、搜索、排序商品信息，对商品进行编辑、添加、删除和上下架等操作。（2）商品分类。管理员可以设置商品分类，对商品进行分类商品，方便用户浏览和筛选商品。（3）订单管理。管理员可以查看、搜索、筛选、排序订单信息，了解订单状态和物流信息等，对订单进行处理、取消等操作。（4）退货申请处理。管理员处理用户提交的退货申请，包括审核、退款、退货等操作，保障用户权益。（5）用户管理。管理员可以搜索和查看用户信息，对用户进行编辑、添加和删除等操作，保障平台安全。（6）促销管理。管理员设置秒杀活动信息和编辑发布优惠卷信息，促进用户购物。同时管理员可以上下线首页轮播图广告列表信息，提升产品知名度。（7）待处理事务。管理员可以查看、处理系统待处理的事务，包括新缺货登记、待处理退货订单、待处理退货申请和广告位临期等。（8）订单统计。管理员进行订单统计分析，包括本周与本月的销售额与订单量等指标，了解运营情况。

3.1 系统模块设计

“优果汇”后台管理主要由商品管理、订单管理、促销管理、运营管理和用户管理模块组成。其软件结构图如图 1 所示。

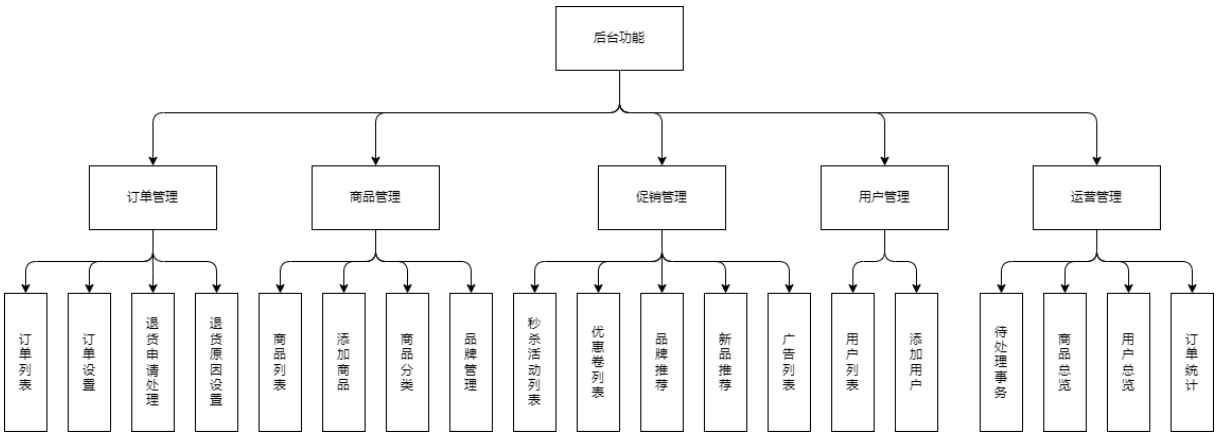

图 1 后台管理模块

3.2 数据库设计

“优果汇”后台管理系统的数据库按照模块设计规划，主要分为商品模块数据库表、订单模块数据库表和营销模块数据库表三大类。各类的数据表及相关字段见表 1-表 3。

4 系统的实现

4.1 商品管理模块

商品管理的具体运行主窗口如图 2 所示。

表 1 商品模块主要数据表及相关字段

序号	数据表	包含字段
1	商品	id， 品牌 id，运费模版 id，品牌属性分类 id，商品名称，图片，货号，删除状态，上架状态，新品状态，推荐状态，审核状态，排序，销量，价格，促销价格，赠送的积分，限制使用的积分数，副标题，商品描述，市场价，库存，库存预警值，单位，商品重量，是否为预告商品，关键字，备注，画册图片，详情标题，详情描述，产品详情网页内容，网页详情地址，促销开始时间，促销结束时间，活动限购数量，促销类型，产品分类名称，品牌名称
2	商品分类	id，上级分类的编号，名称，分类级别，商品数量，商品单位，是否显示在导航栏，显示状态，排序，图标，关键字，描述
3	商品品牌	id，名称，首字母，排序，是否为品牌制造商，是否显示，产品数量，产品评论数量，品牌 logo，专区大图，品牌故事
4	商品属性分类	id，名称，属性数量，参数数量
5	商品属性	id，商品属性分类 id，名称，属性选择类型，属性录入方式，可选值列表，排序，分类筛选样式，检索类型，相同属性产品是否关联，是否支持手动新增，属性的类型
6	商品 SKU	id，商品 id，sku 编码，价格，库存，预警库存，规格属性 1，规格属性 2，规格属性 3，展示图片，销量，单品促销价格，锁定库存
7	商品阶梯价格	id，商品 id，满足的商品数量，折扣，折后价格
8	商品满减	id，商品 id，商品满足金额，商品减少金额
9	商品评价	id，商品 id，用户昵称，商品名称，评价星数，评价的 ip，创建时间，是否显示，购买时的商品属性，收藏数，阅读数，内容，上传图片地址，评论用户头像，回复数
10	产品评价回复	id，评论 id，用户昵称，用户头像，内容，创建时间
11	商品审核记录	id，商品 id，创建时间，审核人，审核后的状态，反馈详情
12	商品操作记录	id，商品 id，改变前价格，改变后价格，改变前优惠价，改变后优惠价，改变前积分，改变后积分，改变前积分使用限制，改变后积分使用限制，操作人，创建时间

表 2 订单模块主要数据表及相关字段

序号	数据表	包含字段
1	订单	id，用户 id，优惠券 id，订单编号，提交时间，用户帐号，订单总金额，实际支付金额，运费金额，促销优化金额，积分抵扣金额，优惠券抵扣金额，管理员后台调整订单使用的折扣金额，支付方式，订单来源，订单状态，订单类型，物流公司，物流单号，自动确认时间，可以获得的积分，活动信息，发票类型，发票抬头，发票内容，收票人电话，收票人邮箱，收货人姓名，收货人电话，收货人邮编，省份/直辖市，城市，区，详细地址，订单备注，确认收货状态，删除状态，下单时使用的积分，支付时间，发货时间，确认收货时间，评价时间，修改时间
2	订单商品信息	id，订单 id，订单编号，商品 id，商品图片，商品名称，商品品牌，商品条码，销售价格，购买数量，商品 sku 编号，商品 sku 条码，商品分类 id，商品的销售属性 1，商品的销售属性 2，商品的销售属性 3，商品促销名称，商品促销分解金额，优惠券优惠分解金额，积分优惠分解金额，该商品经过优惠后的分解金额，商品赠送积分，商品销售属性
3	订单操作记录	id，订单 id，操作人，操作时间，订单状态，备注
4	订单设置	id，秒杀订单超时关闭时间(分)，正常订单超时时间(分)，发货后自动确认收货时间（天），自动完成交易时间，不能申请售后（天），订单完成后自动好评时间（天）
5	购物车	id，商品 id，商品 sku 的 id，用户 id，购买数量，添加到购物车的价格，销售属性 1，销售属性 2，销售属性 3，商品主图，商品名称，商品品牌，商品的条码，商品副标题，商品 sku 条码，用户昵称，创建时间，修改时间，是否删除，商品的分类，商品销售属性
6	订单退货申请	id，订单 id，收货地址表 id，退货商品 id，订单编号，申请时间，用户名，退款金额，退货人姓名，退货人电话，申请状态，处理时间，商品图片，商品名称，商品品牌，商品销售属性，退货数量，商品单价，商品实际支付单价，原因，描述，凭证图片，处理备注，处理人员，收货人，收货时间，收货备注
7	收货地址	id，地址名称，默认发货地址，是否默认收货地址，收货人姓名，收货人电话，省/直辖市，市，区，详细地址
8	订单退货原因	id，退货类型，排序，是否启用，添加时间

表 3 营销模块主要数据表及相关字段

序号	数据表	包含字段
1	限时购	id，标题，开始日期，结束日期，上下线状态，创建时间
2	限时购场次	id，场次名称，每日开始时间，每日结束时间，启用状态，创建时间
3	限时购与商品关系	id，限时购 id，编号，商品价格，限时购价格，限时购数量，每人限购数量，排序
4	限时购通知记录	id，用户 id，商品 id，用户电话，商品名称，用户订阅事件，发送时间
5	优惠券	id，优惠卷类型，名称，使用平台，数量，每人限领张数，使用门槛，开始使用时间，结束使用时间，使用类型，备注，发行数量，已使用数量，领取数量，可以领取的日期，优惠码
6	优惠券历史记录	id，优惠券 id，用户 id，订单 id，优惠券码，领取人昵称，获取类型，创建时间，使用状态，使用时间，订单号码
7	优惠券和商品的关系	id，优惠券 id，商品 id，商品名称，商品条码
8	优惠券和商品分类关系	id，优惠券 id，商品分类 id，商品分类名称，父分类名称
9	首页品牌推荐	id，商品品牌 id，商品品牌名称，推荐状态，排序
10	新品推荐商品	id，商品 id，商品名称，推荐状态，排序
11	首页轮播广告	id，名称，轮播位置，图片地址，开始时间，结束时间，上下线状态，点击数，下单数，链接地址，备注，排序

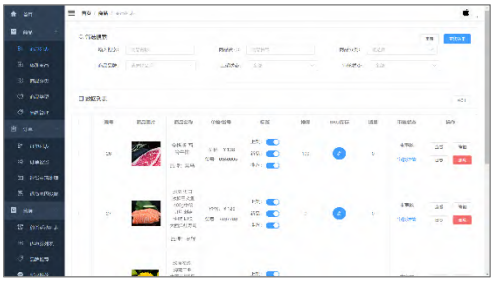

图 2 后台商品管理列表

相关业务代码如下：

```
getList() { // 获取商品列表
  this.listLoading = true;
  fetchList(this.listQuery).then(response => {
    this.listLoading = false;
    this.list = response.data.list;
    this.total = response.data.total;
  });
},
handleAddProduct() { // 添加商品
this.$router.push({path:'/pms/addProduct'});
},
handleSizeChange(val) { // 改变商品列表页面展示数量
  this.listQuery.pageNum = 1;
  this.listQuery.pageSize = val;
  this.getList();
},
handleCurrentChange(val) {
  this.listQuery.pageNum = val;
  this.getList();
},
handleSelectionChange(val) {
  this.multipleSelection = val;
},
handleDelete(index, row){ // 删除商品信息
  this.$confirm('是否要进行删除操作?', '提示', {
    confirmButtonText: '确定',
    cancelButtonText: '取消',
    type: 'warning'
  }).then(() => {
    let ids = [];
    ids.push(row.id);
    this.updateDeleteStatus(1,ids);
  });
},
handleUpdateProduct(index,row){
this.$router.push({path:'/pms/updateProduct',query:{id:row.id}
});
},
updateRecommendStatus(recommendStatus, ids) { // 更新商品推荐状态
  let params = new URLSearchParams();
  params.append('ids', ids);
  params.append('recommendStatus', recommendStatus);
  updateRecommendStatus(params).then(response => {
    this.$message({
      message: '修改成功',
      type: 'success',
      duration: 1000
    });
  });
},
```

4.2 运营模块

运营管理位于后台管理系统的首页，具体运行主窗口如图 3 所示。

图 3 后台运营首页

相关业务代码如下：

```
handleDateChange(){ // 日期变化时获取最新值
  this.getData();
},
initOrderCountDate(){ // 初始化订单图表时间
  let start = new Date();
  start.setFullYear(2023);
  start.setMonth(3);
  start.setDate(14);
  const end = new Date();
  end.setTime(start.getTime() + 1000 * 60 * 60 * 24 * 7);
```

```
    this.orderCountDate=[start,end];
  },
  getData(){ // 获取图表相关数据
    setTimeout(() => {
      this.chartData = {
        columns: ['date', 'orderCount','orderAmount'],
        rows: []
      };
      for(let
i=0;i<DATA_FROM_BACKEND.rows.length;i++){
        let item=DATA_FROM_BACKEND.rows[i];
        let currDate=str2Date(item.date);
        let start=this.orderCountDate[0];
        let end=this.orderCountDate[1];

if(currDate.getTime()>=start.getTime()&&currDate.getTime()<
=end.getTime()){
          this.chartData.rows.push(item);
        }
      }
      this.dataEmpty = false;
      this.loading = false
    }, 1000)
  }
```

4.3 订单管理模块

订单管理包括订单列表、订单设置、退货申请处理和退货原因设置。具体运行主窗口如图 4 所示。

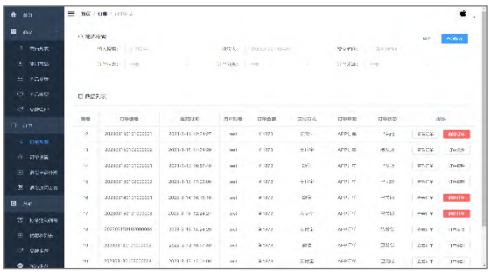

图 4 后台订单管理列表

相关业务代码如下：

```
      handleBatchOperate(){

if(this.multipleSelection==null||this.multipleSelection.length<1
){
          this.$message({
            message: '请选择要操作的订单',
            type: 'warning',
            duration: 1000
          });
          return;
        }
        if(this.operateType===1){
          //批量发货
          let list=[];
          for(let i=0;i<this.multipleSelection.length;i++){
            if(this.multipleSelection[i].status===1){

list.push(this.covertOrder(this.multipleSelection[i]));
            }
          }
          if(list.length===0){
            this.$message({
              message: '选中订单中没有可以发货的订单',
              type: 'warning',
              duration: 1000
            });
            return;
          }

this.$router.push({path:'/oms/deliverOrderList',query:{list:list}
})
        }else if(this.operateType===2){
          //关闭订单
          this.closeOrder.orderIds=[];
          for(let i=0;i<this.multipleSelection.length;i++){

this.closeOrder.orderIds.push(this.multipleSelection[i].id);
          }
          this.closeOrder.dialogVisible=true;
        }else if(this.operateType===3){
          //删除订单
          let ids=[];
          for(let i=0;i<this.multipleSelection.length;i++){
            ids.push(this.multipleSelection[i].id);
          }
          this.deleteOrder(ids);
        }
      },
      getList() {
        this.listLoading = true;
        fetchList(this.listQuery).then(response => {
```

```
            this.listLoading = false;
            this.list = response.data.list;
            this.total = response.data.total;
          });
        },
        deleteOrder(ids){
          this.$confirm('是否要进行该删除操作?', '提示', {
            confirmButtonText: '确定',
            cancelButtonText: '取消',
            type: 'warning'
          }).then(() => {
            let params = new URLSearchParams();
            params.append("ids",ids);
            deleteOrder(params).then(response=>{
              this.$message({
                message: '删除成功！',
                type: 'success',
                duration: 1000
              });
              this.getList();
            });
          })
        },
        covertOrder(order){
          let
address=order.receiverProvince+order.receiverCity+order.receiverRegion+order.receiverDetailAddress;
          let listItem={
            orderId:order.id,
            orderSn:order.orderSn,
            receiverName:order.receiverName,
            receiverPhone:order.receiverPhone,
            receiverPostCode:order.receiverPostCode,
            address:address,
            deliveryCompany:null,
            deliverySn:null
          };
          return listItem;
        }
```

4.4 促销管理模块

促销管理包括秒杀活动列表、优惠卷列表、品牌推荐、新品推荐和广告列表。具体运行主窗口如图 5 所示。

图 5 后台促销管理优惠券列表

相关业务代码如下：

```
      handleResetSearch() {
        this.listQuery = Object.assign({}, defaultListQuery);
      },
      handleSearchList() {
        this.listQuery.pageNum = 1;
        this.getList();
      },
      handleSelectionChange(val){
        this.multipleSelection = val;
      },
      handleSizeChange(val) {
        this.listQuery.pageNum = 1;
        this.listQuery.pageSize = val;
        this.getList();
      },
      handleCurrentChange(val) {
        this.listQuery.pageNum = val;
        this.getList();
      },
      handleAdd(){
        this.$router.push({path: '/sms/addCoupon'})
      },
      handleView(index, row) {
        this.$router.push({path: '/sms/couponHistory', query:
{id: row.id}})
      },
      handleUpdate(index, row) {
        this.$router.push({path: '/sms/updateCoupon', query:
{id: row.id}})
      },
      handleDelete(index, row) {
        this.$confirm('是否进行删除操作?', '提示', {
          confirmButtonText: '确定',
          cancelButtonText: '取消',
          type: 'warning'
```

```
    }).then(() => {
      deleteCoupon(row.id).then(response=>{
        this.$message({
          type: 'success',
          message: '删除成功!'
        });
        this.getList();
      });
    })
  },
  getList(){
    this.listLoading=true;
    fetchList(this.listQuery).then(response=>{
      this.listLoading = false;
      this.list = response.data.list;
      this.total = response.data.total;
    });
  }
```

4.5 用户模块

在用户列表中，展示用户的详细信息，管理员可以选择添加用户、更新用户信息或删除用户。用户列表同时配置搜索功能，能帮助快速定位到某一个用户。具体运行主窗口如图 6 所示。

图 6　后台用户管理列表

相关业务代码如下：

```
  handleAdd() {
    this.dialogVisible = true;
    this.isEdit = false;
    this.admin = Object.assign({},defaultAdmin);
  },
  handleStatusChange(index, row) {
    this.$confirm('是否要修改该状态?', '提示', {
      confirmButtonText: '确定',
      cancelButtonText: '取消',
      type: 'warning'
    }).then(() => {
      updateStatus(row.id, {status:
row.status}).then(response => {
        this.$message({
          type: 'success',
          message: '修改成功!'
        });
      });
    }).catch(() => {
      this.$message({
        type: 'info',
        message: '取消修改'
      });
      this.getList();
    });
  },
  handleDelete(index, row) {
    this.$confirm('是否要删除该用户?', '提示', {
      confirmButtonText: '确定',
      cancelButtonText: '取消',
      type: 'warning'
    }).then(() => {
      deleteAdmin(row.id).then(response => {
        this.$message({
          type: 'success',
          message: '删除成功!'
        });
        this.getList();
      });
    });
  },
  handleUpdate(index, row) {
    this.dialogVisible = true;
    this.isEdit = true;
    this.admin = Object.assign({},row);
  },
  handleDialogConfirm() {
    this.$confirm('是否要确认?', '提示', {
      confirmButtonText: '确定',
      cancelButtonText: '取消',
      type: 'warning'
    }).then(() => {
      if (this.isEdit) {
    updateAdmin(this.admin.id,this.admin).then(response =>
{
```

```
            this.$message({
              message: '修改成功！',
              type: 'success'
            });
            this.dialogVisible =false;
            this.getList();
          })
        } else {
          createAdmin(this.admin).then(response => {
            this.$message({
              message: '添加成功！',
              type: 'success'
            });
            this.dialogVisible =false;
            this.getList();
          })
        }
      })
    },
    handleAllocDialogConfirm(){
      this.$confirm('是否要确认?', '提示', {
        confirmButtonText: '确定',
        cancelButtonText: '取消',
        type: 'warning'
      }).then(() => {
        let params = new URLSearchParams();
        params.append("adminId", this.allocAdminId);
        params.append("roleIds", this.allocRoleIds);
        allocRole(params).then(response => {
          this.$message({
            message: '分配成功！',
            type: 'success'
          });
          this.allocDialogVisible = false;
        })
      })
    },
    handleSelectRole(index,row){
      this.allocAdminId = row.id;
      this.allocDialogVisible = true;
      this.getRoleListByAdmin(row.id);
    },
    getList() {
      this.listLoading = true;
      fetchList(this.listQuery).then(response => {
        this.listLoading = false;
        this.list = response.data.list;
        this.total = response.data.total;
      });
    },
    getAllRoleList() {
      fetchAllRoleList().then(response => {
        this.allRoleList = response.data;
      });
    },
    getRoleListByAdmin(adminId) {
      getRoleByAdmin(adminId).then(response => {
        let allocRoleList = response.data;
        this.allocRoleIds=[];
        if(allocRoleList!=null&&allocRoleList.length>0){
          for(let i=0;i<allocRoleList.length;i++){
            this.allocRoleIds.push(allocRoleList[i].id);
          }
        }
      });
    }
```

5 结语

通过对生鲜农产品电商业务的需求分析，“优果汇”实现了用户管理、商品管理、订单管理、促销管理和运营管理等功能。在技术选型上采用 SpringBoot 框架和 MyBatis 持久层框架实现后端服务，采用 Vue.js 和 ElementUI 实现前端界面和交互，使得整个系统具有良好的扩展性、可维护性和稳定性。

参 考 文 献

[1] 张怡.互联网+背景下农业电子商务发展初探.农业经济,2019(5):126-128

[2] 胡莹瑾.“互联网+”背景下农业电子商务发展对策探析——评《实战农业电子商务》.中国瓜菜,2020,33(01):9

[3] 陈小燕,朱映辉,余晓春.基于SpringBoot+Vue的好农物商城的设计与实现.电脑知识与技术,2022,18(22):37-39

[4] 耿庆阳.基于Spring Boot与Vue的电子商城设计与实现[硕士学位论文].西安石油大学,西安,2020

[5] 刘亚茹,张军.Vue.js 框架在网站前端开发中的研究.电脑编程技巧与维,2022(1):18-19,39