商品网上交易系统-软件需求规格说明(SRS)

商品网上交易系统-软件需求规格说明(SRS)

1范围

1.1 标识

1.2 系统概述

1.3 文档概述

1.4 基线

2引用文件

3需求

3.1所需的状态和方式

3.2需求概述

3.2.1 目标

3.2.2运行环境

硬件环境

软件环境

3.2.3 用户的特点

3.2.4 关键点

3.2.5 约束条件

3.3需求规格

3.3.1软件系统总体功能/对象结构

3.3.2软件子系统功能/对象结构

用户登陆与注册

游客 / 用户商品查询与浏览流程图

用户购物车管理

用户个人信息管理

用户收货人管理

用户收藏夹管理

用户发起订单

用户管理订单

商品管理

查询订单

订单状态转换与可见性

3.3.3描述约定

3.4CSCI能力需求

3.4.1 商品浏览和搜索能力

3.4.2 购物车管理能力

3.5CSCI外部接口需求

3.5 CSCI外部接口需求

a. 用户接口

b. 硬件接口

c. 软件接口

d. 通信接口

3.5.1接口标识和接口图

E-R图

函数调用

3.6CSCI内部接口需求

3.7CSCI内部数据需求

3.8适应性需求

3.9保密性需求

3.10 保密性和私密性需求
保密性/私密性环境
提供的保密性/私密性的类型和程度
保密性/私密性的风险和安全措施
保密性/私密性政策
保密性/私密性审核和确证/认可准则
3.11 CSCI环境需求
3.12 计算机资源需求
3.12.1 计算机硬件需求
3.12.2 计算机硬件资源利用需求
3.12.3 计算机软件需求
3.12.4 计算机通信需求
3.13软件质量因素
产品运行
产品修改
产品转移
3.14设计和实现的约束
a. 特殊CSCI体系结构的使用或体系结构方面的需求
b. 特殊设计或实现标准的使用
c. 灵活性和可扩展性
3.15数据
3.16 操作
常规操作
特殊操作
初始化操作
恢复操作
3.17 故障处理
a. 识别软件系统问题
b. 错误信息
c. 补救措施
3.18算法说明
3.19 有关人员需求
人员数量
技能等级
责任期
培训需求
用户数量需求
内在帮助和培训能力需求
3.20有关培训需求
3.21有关后勤需求
3.22其他需求
3.23包装需求
3.24需求的优先次序和关键程度
4合格性规定
5需求可追踪性
5.1 从CSCI需求到系统（或子系统）需求的可追踪性
5.2 从系统（或子系统）需求到CSCI需求的可追踪性
6尚未解决的问题
地址管理
支付处理
推荐算法
商品评价

7注解
附录

1范围

1.1 标识

本条目描述了商品网上交易系统的相关标识信息，包括：

- **标识号**：SRS-001
- **标题**：商品网上交易系统-软件需求规格说明
- **缩略词语**：E-Commerce SRS
- **版本号**：1.0
- **发行号**：2024年4月

1.2 系统概述

本商品网上交易系统旨在为用户提供便捷、安全的在线购物体验。系统的主要特性包括商品浏览、搜索、购物车、订单管理、支付处理、用户账户管理和促销活动。系统的发展目标是提供高效、用户友好的平台，以满足客户和商家的需求。

项目由投资方XYZ公司出资，由ABC开发团队负责开发，用户为在线购物消费者。系统的运行和维护由XYZ公司的技术支持团队负责。该系统计划在XYZ公司现有的服务器基础设施上运行，并计划扩展到云端部署。

其他有关的文档包括：

- 项目计划书
- 系统架构设计
- 测试计划和报告

1.3 文档概述

本文档用于描述商品网上交易系统的需求规格，包括功能需求、性能需求和接口需求等。文档为项目团队、开发人员和用户提供一个详细的参考，以确保系统的开发和交付符合预期。

本说明书的预期读者为客户、业务或需求分析人员、测试人员、用户文档编写者、项目管理人员。

1.4 基线

本系统设计说明书编写依据如下设计基线：

- **业务需求文档**：描述了系统的主要业务需求。
- **技术规范文档**：列出了系统的技术要求和规范。
- **行业标准和法规**：系统设计参考了相关的行业标准和法规，包括网络安全、数据保护等。

2引用文件

中华人民共和国国家标准GB T-8567-2006。

《软件工程导论》，张海藩。

《实用软件工程》，郑人杰等。

《软件工程导论》，张海藩（第五版）；

《软件工程 第4版》，（美）弗里格等；

3需求

3.1所需的状态和方式

- **空闲状态**：在没有用户操作或系统不需要处理订单时，系统处于空闲状态。在这种状态下，系统应保持低资源占用，以确保高效率运行，同时随时准备接受用户的操作。
- **准备就绪状态**：系统已完成初始化并准备好接受用户操作。在这种状态下，系统应确保各项功能（如商品浏览、搜索、购物车等）都已正常启动并随时可用。
- **活动状态**：系统正处于处理用户请求或订单的过程中。在这种状态下，系统应确保响应时间快，确保用户体验良好。同时，应确保数据准确无误，操作过程不出错。
- **事后分析状态**：系统可以根据过去的订单和用户行为进行数据分析，生成报告或其他形式的统计数据。在这种状态下，系统应具备数据处理和分析能力，以提供有用的见解和决策支持。
- **降级状态**：在出现故障或资源受限的情况下，系统可以降级为部分功能可用或限制功能运行。在这种状态下，系统应继续保持核心功能的运行，以确保业务的基本正常运作。
- **紧急情况**：在出现紧急情况（如系统故障、网络中断等）时，系统应具备应急处理措施，如自动故障转移、数据备份和恢复等，确保业务连续性。
- **后备状态**：在系统的主要功能不可用时，系统应有备份机制或冗余设计，以确保业务不中断。在这种状态下，系统应能够快速切换到备用系统或组件，以确保正常运行。

3.2需求概述

3.2.1 目标

a. **开发意图、应用目标及作用范围**：本商品网上交易系统的开发意图是提供一个在线购物平台，使消费者能够方便地浏览、选择和购买商品。该系统旨在提高购物体验，提高用户满意度，并通过高效的订单和支付处理提高商家的运营效率。系统的作用范围包括产品展示、购物车、订单管理、支付处理、用户账户管理和评价等功能。

b. **主要功能、处理流程和数据流程**：系统的主要功能包括：

- **产品浏览和搜索**：用户可以浏览和搜索商品，并查看商品详细信息。
- **购物车**：用户可以将选定的商品添加到购物车，并进行管理。
- **订单管理**：用户可以创建订单、查看订单状态，并管理订单。
- **用户账户管理**：用户可以创建和管理个人账户，包括信息更新和订单历史查看。
- **评价和评论**：用户可以对商品进行评价和评论，以帮助其他用户做出决策。

处理流程：

- 用户浏览商品并选择要购买的商品。
- 将商品添加到购物车。

- 用户确认订单，输入支付信息并完成支付。
- 系统处理订单，更新库存，并通知用户订单状态。
- 用户完成订单后，可以对商品进行评价和评论。

c. **表示外部接口和数据流的系统高层次图**：系统与其他相关产品的关系包括：

- **支付网关**：与第三方支付网关接口，用于处理支付交易。
- **物流和配送服务**：与物流服务提供商接口，用于订单配送。
- **供应商系统**：与供应商系统接口，用于获取产品信息和库存数据。
- **社交媒体和市场营销**：与社交媒体和市场营销平台接口，用于广告和推广活动。

系统高层次图可以通过方框图来表示，展示系统的主要组件及其与外部系统之间的数据流和接口关系。例如，展示系统与支付网关、物流服务提供商和供应商系统之间的数据交互。

3.2.2运行环境

以下是在线购物平台的硬件和软件环境的表格，包括具体的型号和技术。

硬件环境

硬件组件	具体型号	说明
服务器	Aliyun	32 GB RAM，1 TB SSD存储

软件环境

软件组件	具体型号	说明
操作系统	Windows 11	Windows 11发行版，稳定且通用
数据库	MySQL	高性能、可扩展的关系型数据库
后端框架	Node.js	JavaScript后端框架，提供快速开发和高性能
前端框架	React	现代前端框架，提供高效的用户界面开发
调试浏览器	Edge/Chrome	测试和调试Web应用程序的主流浏览器
IDE	Visual Studio Code	流行的集成开发环境（IDE），支持多种编程语言

3.2.3 用户的特点

商品网上交易系统的用户类型主要包括以下两种：

- **消费者**：消费者是系统的主要用户。他们使用系统浏览和购买商品、管理订单、查看订单状态以及进行商品评价和评论。消费者期望系统易于使用，界面友好，并提供快速、可靠的支付方式。
- **商家**：商家是系统的合作伙伴，他们使用系统上传商品信息、管理库存、查看销售数据以及处理订单。他们期望系统能够为他们提供便捷的商品管理和销售数据分析功能。

3.2.4 关键点

本软件需求规格说明书中的关键点包括：

- **关键功能**：系统的关键功能包括商品浏览和搜索、购物车、订单管理、支付处理和用户账户管理。这些功能是系统的核心组成部分，决定了系统的用户体验和整体效率。
- **关键技术**：项目使用React作为前端框架，实现高效的用户界面设计和交互；使用MySQL作为数据库，确保数据的可靠存储和高效查询。前后端分离的架构也能提升开发效率和系统的可维护性。
- **关键算法**：可以考虑用于推荐商品的算法、商品搜索和排序的算法，这些算法将直接影响用户体验和系统性能。

3.2.5 约束条件

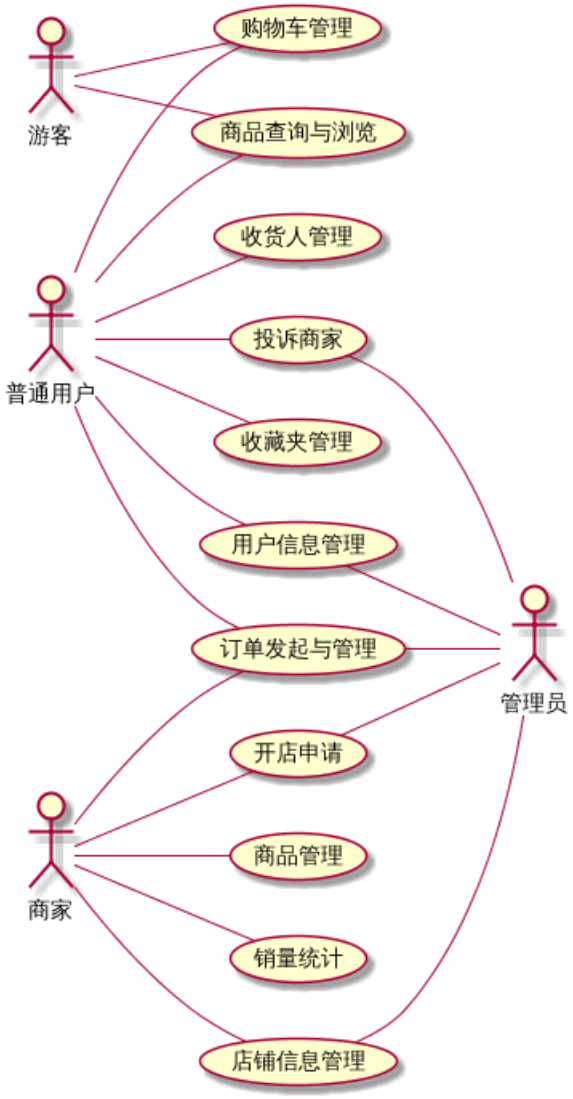
进行本系统开发工作的约束条件包括：

- **经费限制**：项目的预算有限，因此在选择技术和方法时要合理分配资源，确保成本效益最大化。
- **开发期限**：项目有明确的时间框架，要求在规定时间内完成开发和测试工作。因此，需要有效规划和管理开发流程，以确保项目按时交付。
- **采用的方法与技术**：项目采用React作为前端框架，MySQL作为数据库。
- **政治、社会、文化、法律等**：系统应符合所在地区的法律法规和文化规范，例如数据隐私和安全要求。此外，系统的设计和功能应尊重不同文化背景下的用户习惯和偏好。

3.3需求规格

3.3.1软件系统总体功能/对象结构

软件系统总体功能流程图。

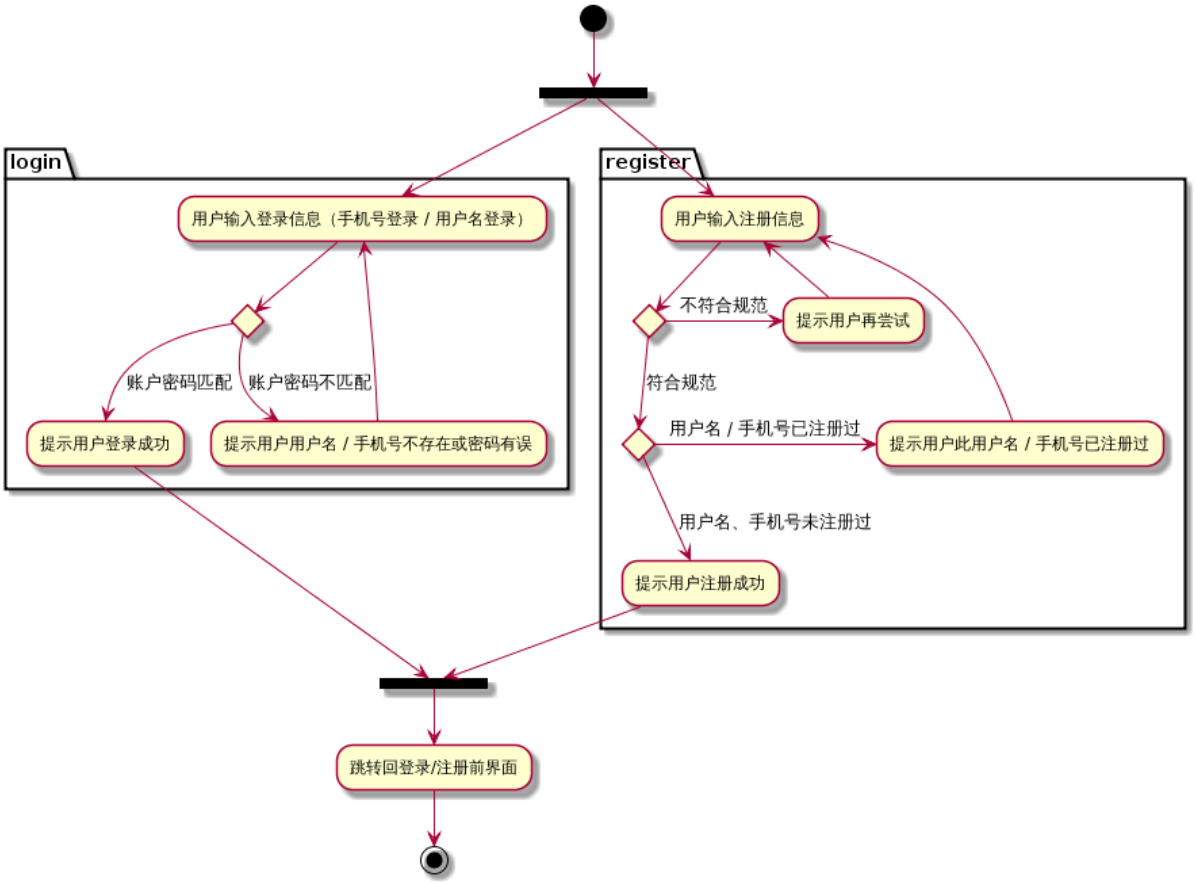


3.3.2软件子系统功能/对象结构

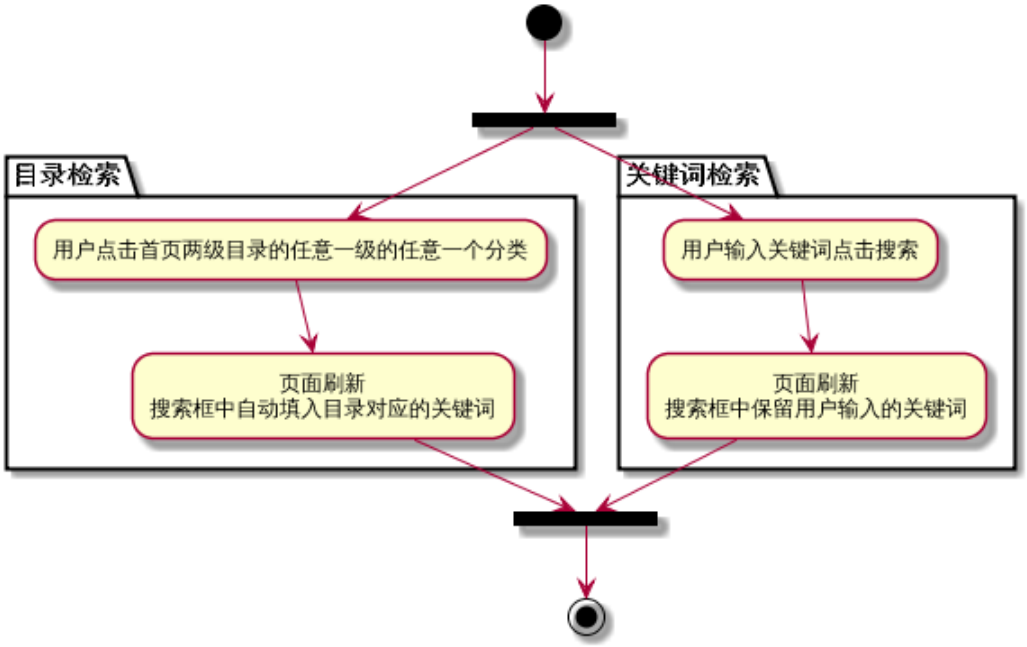
用户登陆与注册

游客可以通过注册来获取账户，注册时要提供自定义的用户名、可用的手机号、学号住址等信息，具体要求见数据字典。

游客 / 用户登录与注册用例活动图如下：



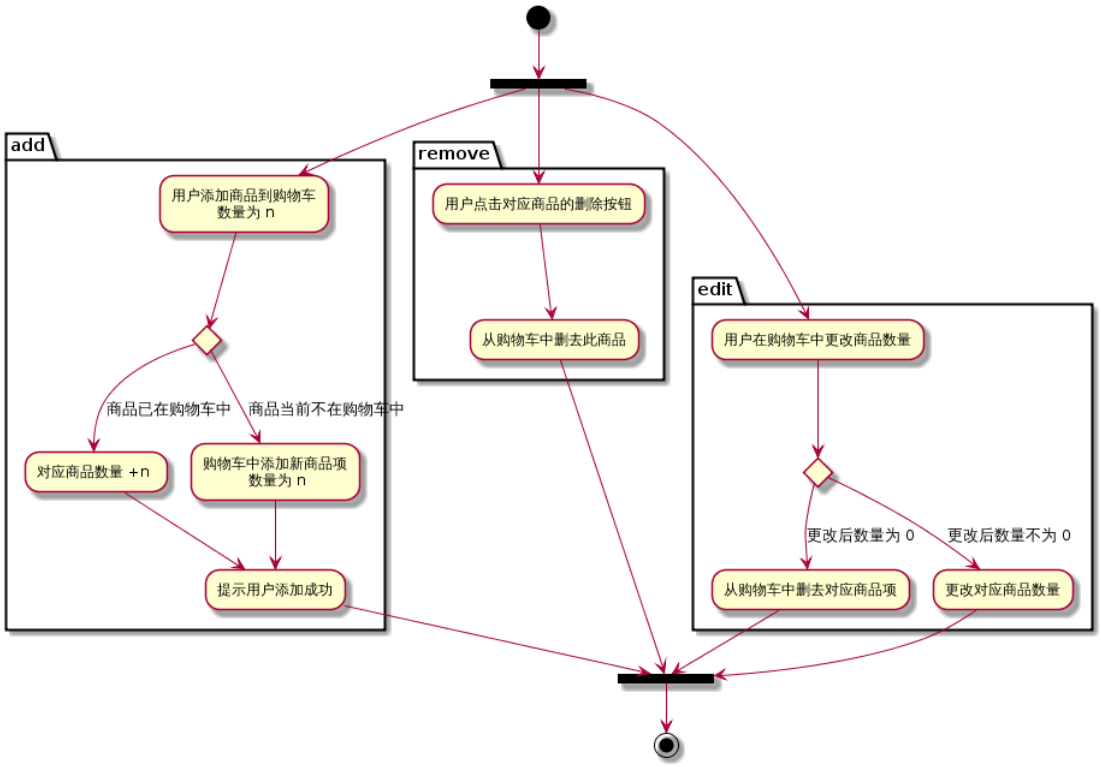
游客 / 用户商品查询与浏览流程图



用户购物车管理

游客 / 用户可以通过商品页面直接填加商品到购物车。在管理购物车时，可以修改商品数量，也可以直接删除商品。游客 / 用户购物车在关闭浏览器或者退出登录后仍保留。

游客 / 用户购物车管理用例活动图如下：



用户个人信息管理

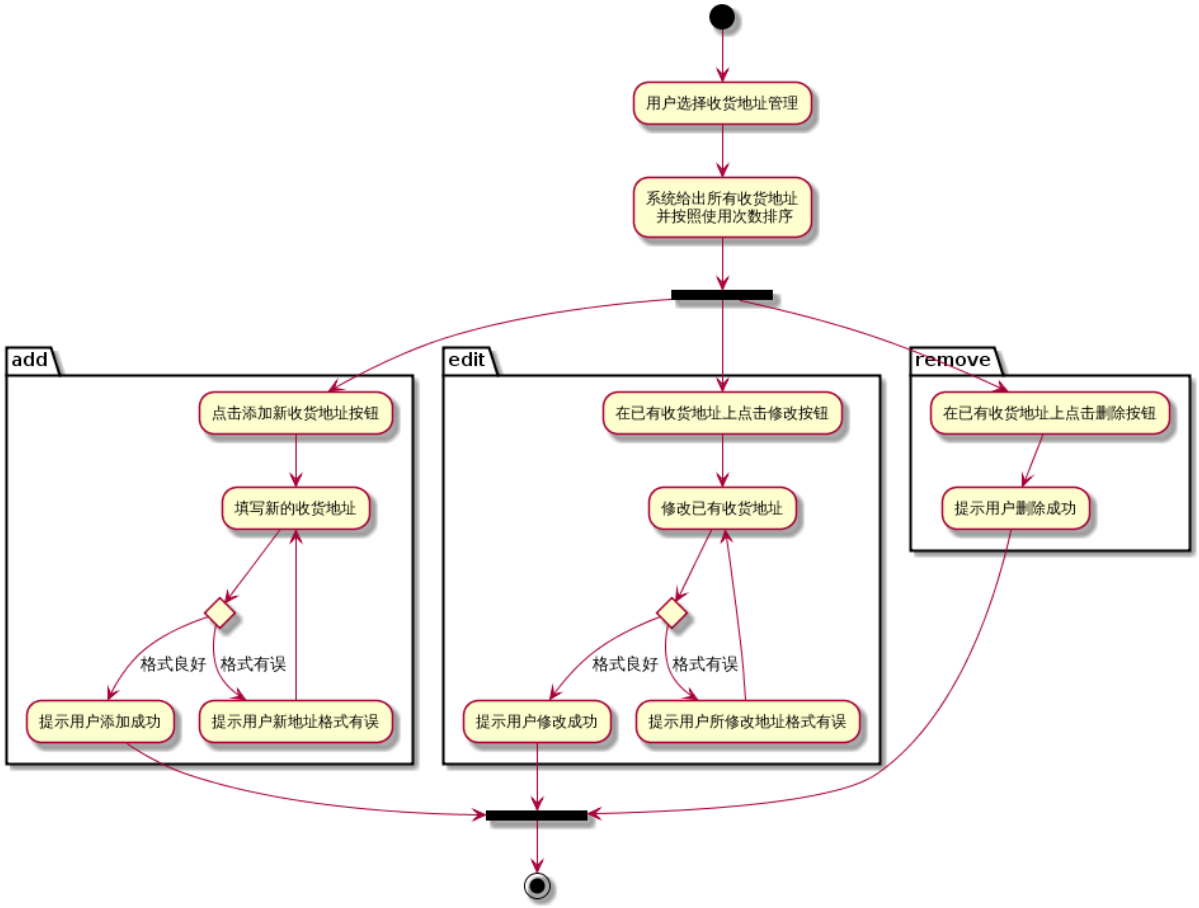
用户可以修改自己除用户名以外的任何个人信息，但不允许自主删除帐号。

用户个人信息管理过程过于简单，不提供用例活动图。

用户收货人管理

系统要记录用户收货地址使用次数，当用户管理或挑选地址使用时将使用次数最多的地址排在首位。当用户自主删除收货人时，系统要保留数据，以备后用。

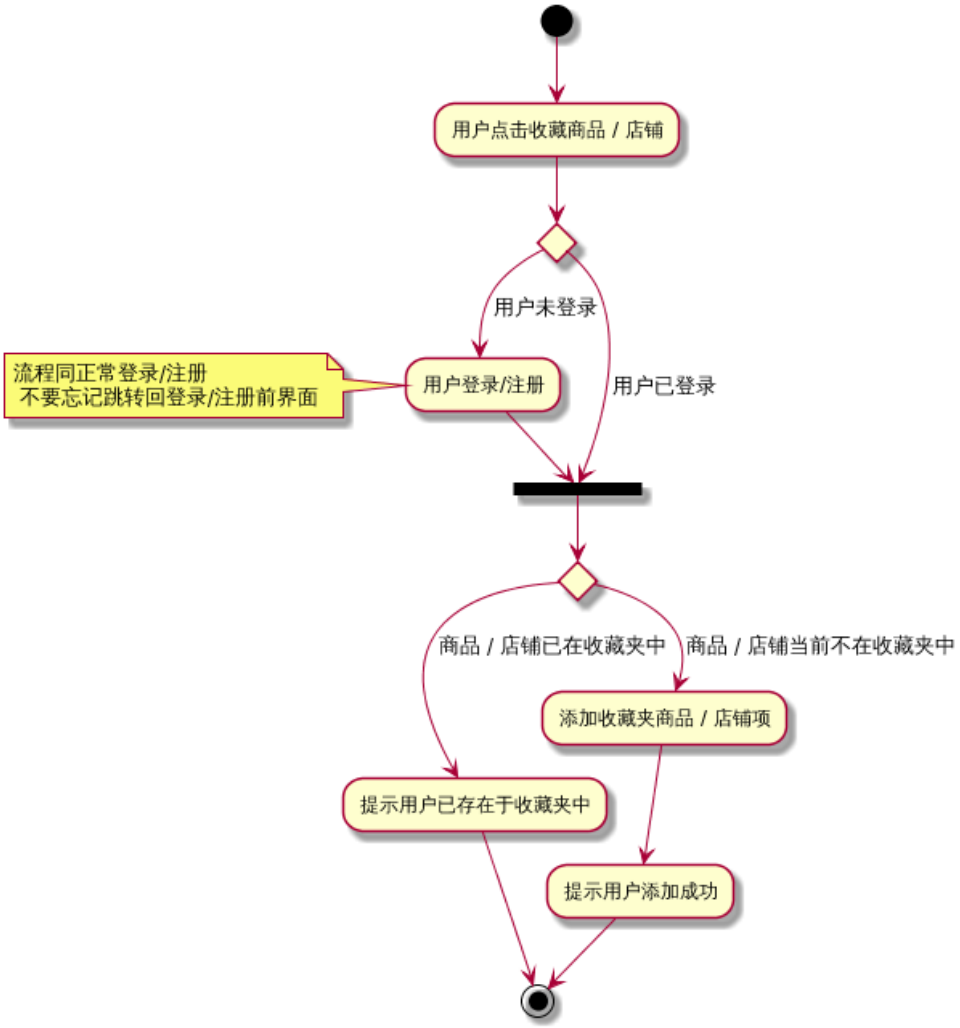
用户搜索用例活动图如下：



用户收藏夹管理

用户可以直接在商品页面点击收藏按钮收藏商品，在店铺页面点击收藏按钮收藏店铺。在管理收藏夹时，可以将任意的店铺和商品从收藏夹中删除。用户自主删除收藏后，系统要保留数据，以备后用。

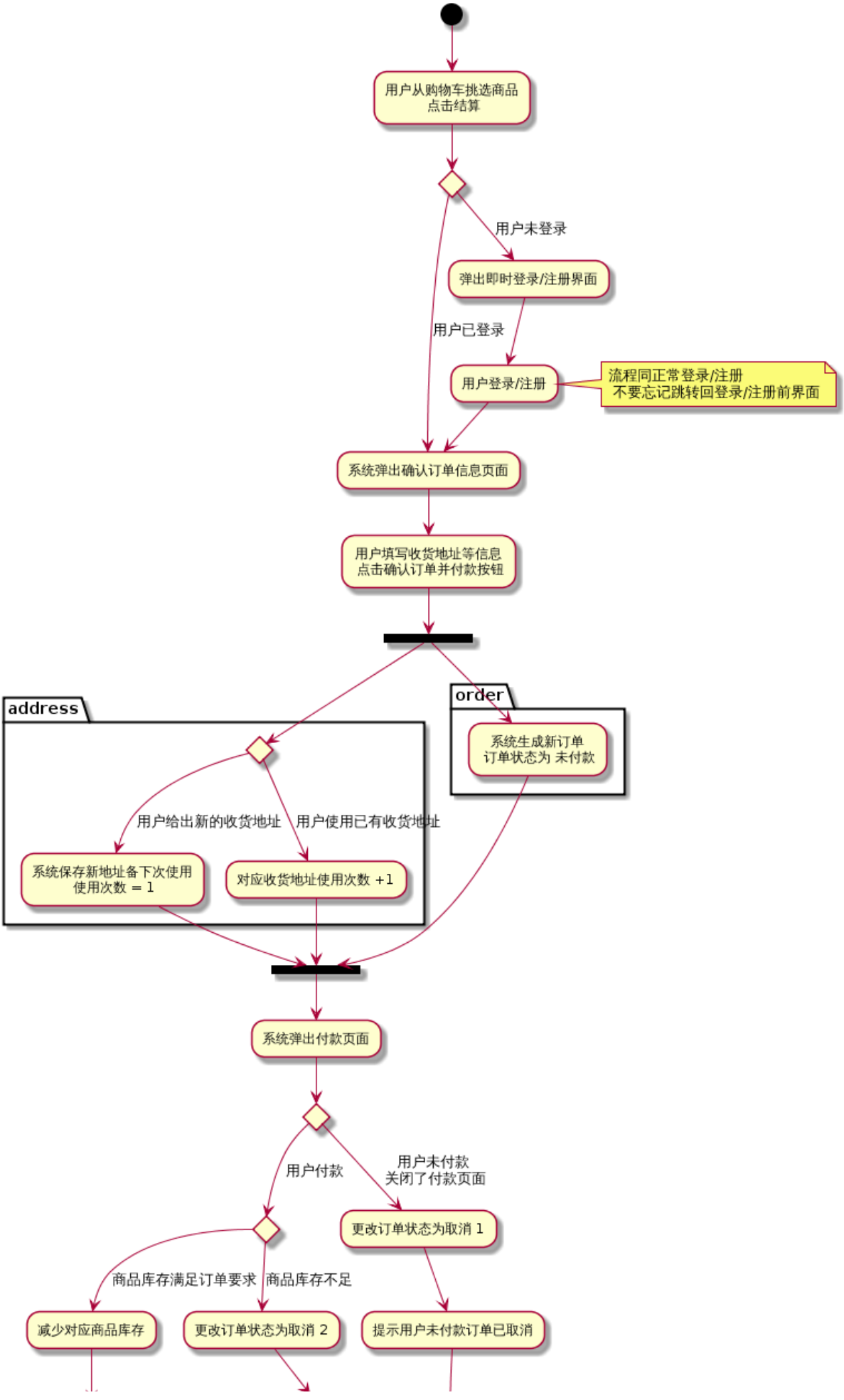
用户收藏夹管理用例活动图如下：

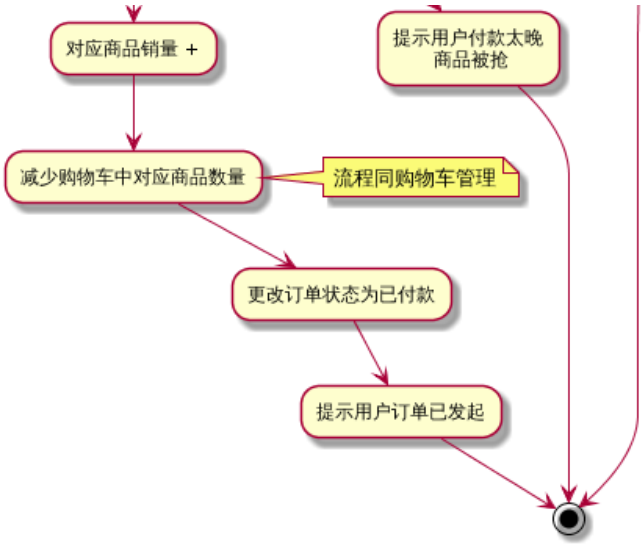


用户发起订单

用户通过在购物车中挑选商品发起订单。

用户发起订单用例活动图如下：

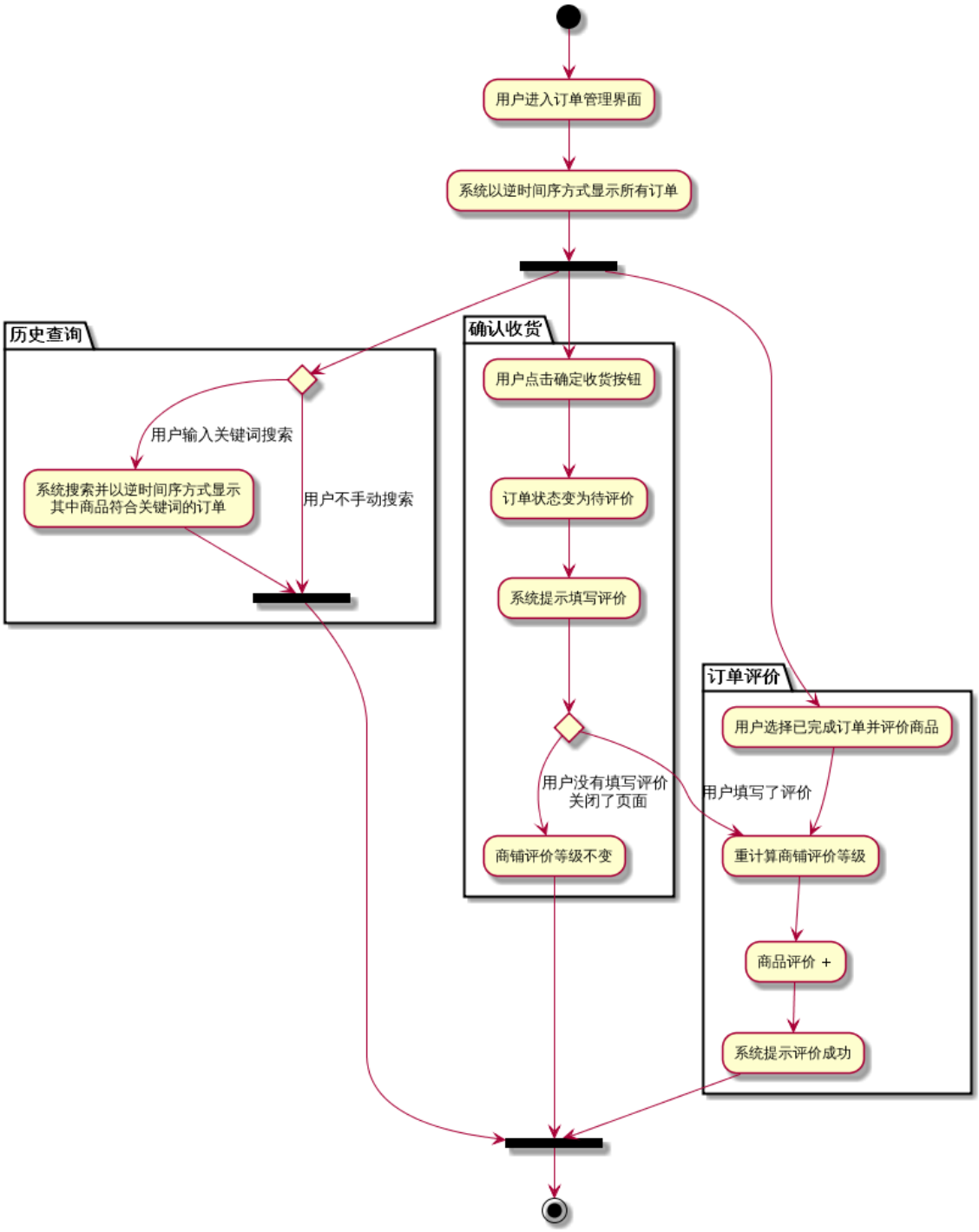

用户从购物车挑选商品
点击结算
用户未登录
弹出即时登录/注册界面
用户已登录
用户登录/注册
流程同正常登录/注册
不要忘记跳转回登录/注册前界面
系统弹出确认订单信息页面
用户填写收货地址等信息
点击确认订单并付款按钮
address
order
系统生成新订单
订单状态为 未付款
用户给出新的收货地址
用户使用已有收货地址
系统保存新地址备下次使用
使用次数 = 1
对应收货地址使用次数 +1
系统弹出付款页面
用户付款
用户未付款
关闭了付款页面
更改订单状态为取消 1
商品库存满足订单要求
商品库存不足
减少对应商品库存
更改订单状态为取消 2
提示用户未付款订单已取消



用户管理订单

用户在订单管理中心能看到自己所有状态的所有订单，但只能查看详细信息或对订单确认收货，不能删除订单。订单管理中心要提供用户按时间和按关键词两种查询方式。当用户确认收货时，不管当前订单状态是什么，标识着一个订单的结束。订单完成或后用户可以立即或稍后对商品进行评价，评价将影响店铺评价等级。

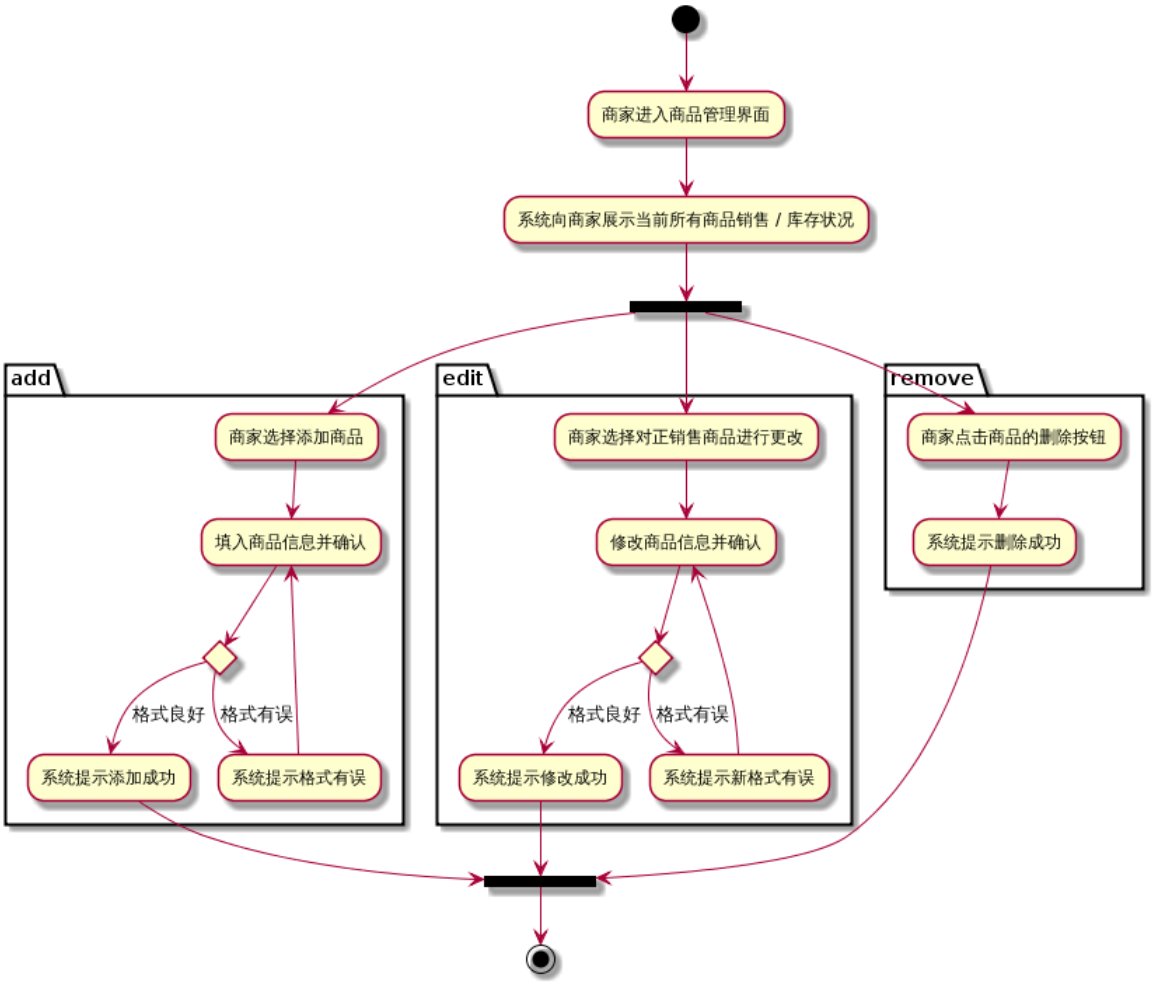
用户管理订单用例活动图如下：



商品管理

商家可以对自己店铺内的商品进行添加、修改、删除。商家可以对指定产品进行定时打折，可以指定打折后的金额或打折的比例。当商家 / 管理员删除商品时，系统要保留数据，以备后用。

商家商品管理用例活动图如下：



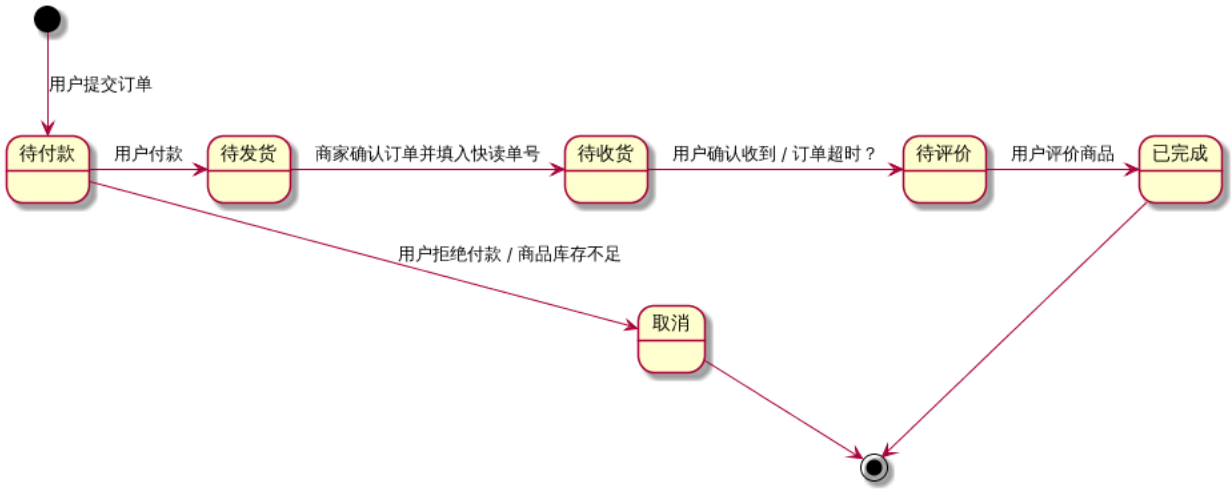
查询订单

所有订单的所有状态对管理员可见。管理员可以通过订单号查询订单、查看订单详细信息，但不可操作订单状态。

查询订单过程过于简单，不提供用例活动图。

订单状态转换与可见性

用户行为与订单状态转换图如下：



3.3.3描述约定

无。

3.4CSCI能力需求

3.4.1 商品浏览和搜索能力

a. **说明**：该功能旨在为用户提供便捷的商品浏览和搜索体验。通过搜索框和分类浏览功能，用户可以轻松找到所需商品。此功能使用现代前端技术，如Vue和React，提供直观、快速的用户界面。

b. **输入**：

- 用户输入的搜索关键字或过滤条件。
- 用户选择的商品分类、价格范围、品牌等过滤选项。
- 时间设定：用户可能在特定时间搜索商品。

c. **处理**：

- 系统根据用户输入的关键字或过滤条件查询数据库。
- 检查输入数据的有效性，防止非法或错误输入。
- 根据输入的条件进行查询，按照相关性或其他排序方式对结果进行排序。
- 响应异常情况，例如输入无效或数据库查询错误等。

d. **输出**：

- 返回搜索结果，包含商品名称、图片、价格、库存状态等信息。
- 输出数据的有效性检查，确保返回的数据准确、完整。
- 在发生错误时，返回友好的错误信息。

3.4.2 购物车管理能力

a. **说明**：该功能允许用户将商品添加到购物车，查看购物车内容，更新商品数量，以及删除商品。购物车管理功能为用户提供方便的购物体验。

b. **输入**：

- 用户选择添加到购物车的商品及其数量。
- 用户请求查看购物车内容或更新购物车信息。
- 用户请求删除购物车中的商品。

c. **处理**：

- 检查用户输入的商品信息和数量的有效性。
- 将商品添加到购物车或更新购物车中的商品数量。
- 处理购物车中的商品删除请求。
- 在异常情况（如商品库存不足）时，返回适当的错误信息。

d. **输出**：

- 返回购物车的当前内容，包括商品名称、数量、价格和总金额等信息。
- 输出数据的有效性检查，确保购物车中的商品信息准确无误。
- 提供错误信息（如商品库存不足、购物车更新失败）以便用户采取相应行动。

3.5CSCI外部接口需求

3.5 CSCI外部接口需求

a. 用户接口

商品网上交易系统的用户接口需求包括：

- **用户界面**：系统应提供直观、友好的用户界面，包括商品浏览、购物车管理、订单管理和支付等功能模块。界面应支持多语言和多货币，以适应不同地区的用户需求。
- **用户输入和输出**：用户可以通过输入框、按钮、下拉菜单等与系统交互。系统应及时、准确地反馈用户的操作，并提供适当的提示信息。
- **用户身份验证**：系统应支持用户注册和登录功能，确保用户身份的安全和唯一性。可以使用用户名/密码、短信验证、双因素认证等方式进行验证。

b. 硬件接口

商品网上交易系统的硬件接口需求包括：

- **服务器硬件接口**：系统需要与服务器硬件（如CPU、内存、存储）进行交互，以提供可靠的运行环境。
- **外设接口**：系统可能需要与打印机、扫码枪等外设进行接口，以支持订单打印、商品条码扫描等功能。

c. 软件接口

商品网上交易系统的主要软件接口需求包括：

- **支付网关**：系统需要与第三方支付网关集成，处理支付交易。接口要求包括安全的支付传输、支付确认和交易状态反馈等。
- **物流和配送服务**：系统需要与物流和配送服务提供商接口，支持订单配送、跟踪和确认。接口要求包括订单信息传递和物流状态反馈等。
- **供应商系统**：系统需要与供应商系统接口，获取商品信息和库存数据。接口要求包括数据同步和实时更新等。

d. 通信接口

商品网上交易系统的通信接口需求包括：

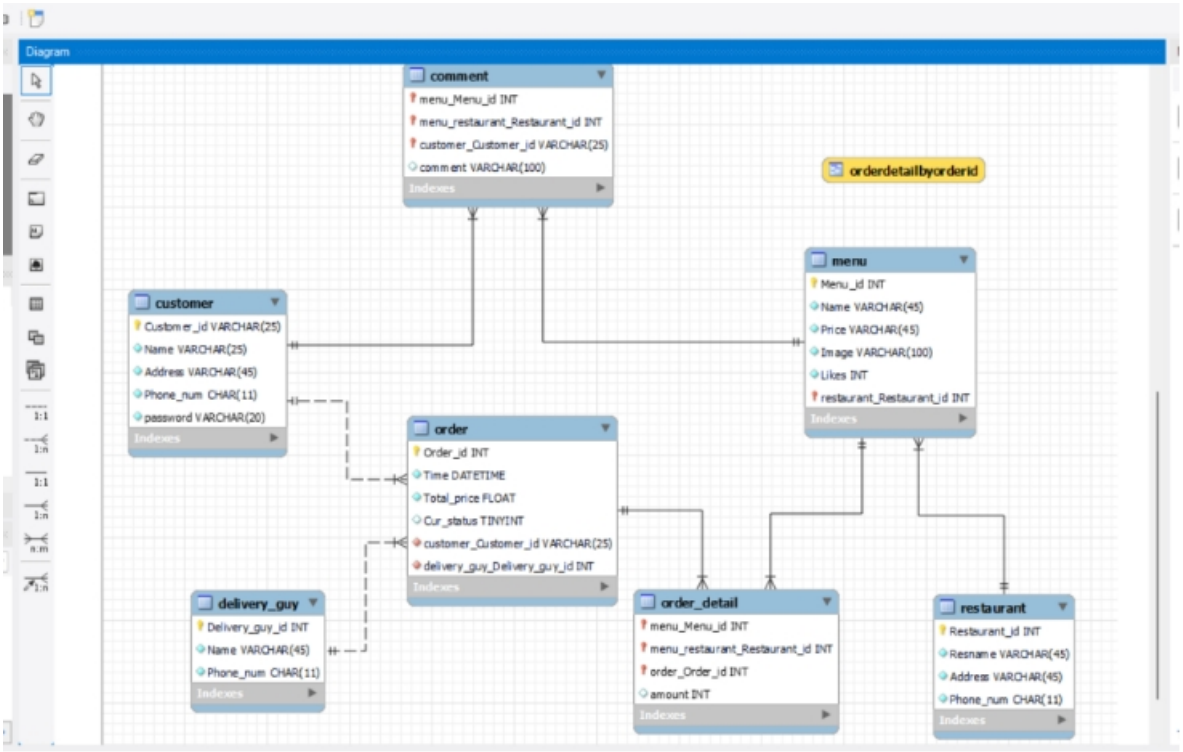
- **网络协议**：系统应支持HTTP/HTTPS协议进行数据传输，确保数据的安全性和完整性。
- **数据格式**：系统应支持JSON、XML等常见的数据格式进行数据交互，确保接口的兼容性和灵活性。
- **网络安全**：系统应确保通信过程中的数据加密和认证，防止数据泄露和篡改。

这些外部接口需求将确保商品网上交易系统能够与用户、硬件、软件和其他系统之间进行有效和安全的通信。

3.5.1接口标识和接口图

考虑到本项目相对简单，因此此处仅将调用数据库的接口函数关系图贴在此处。

E-R图



函数调用

此处仅列出关键或复杂的事务操作的设计，附相应SQL语句进行分析介绍。

1、 登录、注册

登录时，对于用户输入的账号和密码，调用函数申请数据库数据，若数据库返回对应数据，则登陆成功，网页内保存当前用户的ID；注册时，对于每一个输入都进行相对应的检查，若错误，则网页发出错误警告，并删除数据库对应数据，若成功，则数据插入成功。

```
,,
app.post("/addcustomer",(req,res)=>{
    const q = "INSERT INTO `customer` (`Customer_id`, `Name`, `Address`, `Phone_num`,`password`) VALUES (?)";
    const values = [
        req.body.id,
        req.body.name,
        req.body.address,
        req.body.phonenum,
        req.body.password
    ];
    db.query(q,[values],(err,data)=>{
        if (err) {
            return res.json(err);
        } else {
          return res.json(null) ;
        }
    });
```

2、 选择商品加入购物车

点单环节过程是在网页中进行，只有最后提交订单涉及数据库事务。

3、 结算购物车

由于结算购物车时要更改order和order_detail两张表，而且由于外码约束，需要先插入order再插入order_detail，需要事务处理以防止意外发生。

结算时，首先插入数据库数据，然后对事务进行检查，若无误，则commit，反之rollback。

```
app.post("/buy",(req,res)=>{
    const q="INSERT INTO `order` (`Time`,`Total_price`,`Cur_status`,`customer_Customer_id`,`delivery_guy_Delivery_guy_id`) VALUES (?);"
    const qq="INSERT INTO `order_detail` (`menu_Menu_id`, `menu_restaurant_Restaurant_id`, `order_Order_id`, `amount`) VALUES ?;"
    const e=new Date()
    const f=e.toLocaleString()
    const values = [
        f,
        req.body.price,
        0,
        req.body.id,
        1
    ];
    db.beginTransaction((err)=>{
        if(err)
        {return res.json(err);}
        db.query(q,[values],(error,data)=>{

            if(error)
            {
                db.rollback();
                return res.json(error);
            }
            const thevalue=req.body.buy.map((item)=>{
                return [item.id,item.restaurant_Restaurant_id,data.insertId,item.num]
            })
            db.query(qq,[thevalue],(err,data)=>{
```

```
                }
                const thevalue=req.body.buy.map((item)=>{
                    return [item.id,item.restaurant_Restaurant_id,data.insertId,item.num]
                })
                db.query(qq,[thevalue],(err,data)=>{
                    if(err)
                    {
                       db.rollback();
                       return res.json(err);
                    }
                    db.commit((err)=>{
                        if(err)
                        {
                            db.rollback();
                            return res.json(err);
                        }
                    return res.json(null);
                })


                })
            })
        })
```

4、 查看订单

查看订单是直接搜寻order_detail关系，返回对应用户的所有订单和订单详情。

```
app.get("/selectbyorderid",(req,res)=>{
    const q=`SELECT * FROM  takeoutapp.orderdetailbyorderid where order_Order_id = ${req.query.Id};`
    db.query(q,(err,data)=>{
        if(err){
            return res.json(err);
        } else {
            console.log(data)
            return res.json(data);
        }
       })
})
app.get("/selectfromorder",(req,res)=>{
    const q=`SELECT Order_id,DATE_FORMAT(Time, '%Y-%m-%d %H:%i:%S') AS Time,Total_price,Cur_status,delivery_guy_Delivery_guy_id FROM  takeoutapp.order where custome
    db.query(q,(err,data)=>{
        if(err){
            return res.json(err);
        } else {
            return res.json(data);
        }
       })
})
```

5、 用户确认收货

用户确认收货，在网页上进行相关操作后，返回数据至数据库。

```
app.post("/confirm",(req,res)=>{
    const q="UPDATE `takeoutapp`.`order` SET `Cur_status` = '1' WHERE (`Order_id` = (?))";
    const values =[
        req.body.id
    ]
    db.query(q,[values],(err,data)=>{
        if (err) {
            return res.json(err);
        } else {
          return res.json(null);
        }
    });
})
```

3.6CSCI内部接口需求

本项目工程量相对较小，无需内部接口，直接对外部接口进行测试即可，这样可以防止攻击者会绕过前端的验证直接对接口进行攻击。

3.7CSCI内部数据需求

项目中相关数据均已在3.15出列出，此处不再赘述。

3.8适应性需求

本系统要求具有高度的适应性，以满足各种不同的运行环境和用户需求。适应性需求具体包括以下方面：

1. **依赖于安装的数据**：
    - **地理位置信息**：系统应能够根据用户的地理位置信息（经纬度）提供本地化的商品推荐和配送服务。
    - **市场和用户习惯**：系统应根据不同地区的市场趋势和用户购物习惯，灵活调整商品类别和展示方式。
2. **运行参数**：
    - **动态定价**：根据市场需求和库存水平，系统应支持动态定价功能，调整商品价格以提升销售效率。

- **促销策略**：系统应根据不同季节、节假日和市场情况，支持各种促销策略的调整，如折扣、优惠券等。
- **库存管理**：系统应能够根据库存状态和供应链情况，动态调整商品的可见性和订购限制。

3. **配置文件**：
   - **环境配置**：系统应支持通过配置文件或环境变量来调整系统行为，如数据库连接、API密钥等。
   - **用户偏好**：系统应根据用户的历史购买记录、浏览历史和其他数据，调整推荐的商品类别和个性化服务。
4. **数据记录**：
   - **日志记录**：系统应详细记录运行过程中的各种数据和事件，包括用户行为、订单处理等，以供监控和调试。
   - **统计分析**：系统应支持数据统计和分析功能，以优化运营策略和用户体验。

系统应设计成易于扩展和定制，以适应不同的业务需求和运行环境。通过满足以上适应性需求，系统可以在不同场景下保持高效、稳定的运行状态，并提供优质的用户体验。

3.9保密性需求

商品交易系统应严格遵循保密性原则，确保用户数据和交易数据的安全。保密性需求具体包括以下方面：

1. **数据加密**：
   - **数据传输**：系统应在数据传输过程中使用SSL/TLS等加密协议，确保用户数据在网络中安全传输。
   - **数据存储**：敏感数据，如用户密码、支付信息等，应在数据库中使用强加密算法加密存储。
2. **身份验证**：
   - **多因素身份验证**：系统应提供多因素身份验证选项，增强用户账户的安全性。
   - **密码安全**：系统应强制要求用户设置强密码，并定期提醒用户更改密码。
3. **访问控制**：
   - **最小权限原则**：系统应根据最小权限原则，限制用户和管理员对数据和功能的访问权限。
   - **数据分层保护**：系统应对不同类型的数据和功能进行分层保护，确保只有授权用户才能访问敏感数据。
4. **活动监控**：
   - **日志记录**：系统应详细记录用户和管理员的活动，包括登录、数据访问和更改等，以便于审计和监控。
   - **异常检测**：系统应监控异常行为，如频繁登录失败、异常数据访问等，并及时发出警报。
5. **数据保护政策**：
   - **数据最小化**：系统应收集和存储最小必要的数据，并遵守相关数据保护法规。
   - **数据删除**：系统应提供数据删除功能，以满足用户删除个人数据的请求。
6. **安全措施**：
   - **权限分离**：确保用户和管理员权限分离，减少因权限过度集中导致的风险。
   - **备份和恢复**：定期备份数据，并制定数据恢复策略，以应对数据泄露或丢失。

通过满足以上保密性需求，系统可以有效地防止对用户、财产和环境产生潜在的危险，确保交易数据和用户数据的安全。

3.10 保密性和私密性需求

保密性/私密性环境

- **数据存储和传输加密**：系统应对存储和传输中的敏感数据（如用户密码、支付信息）进行加密，使用强加密算法（如AES或RSA）保护数据。
- **安全通信**：系统应使用HTTPS协议保护用户与服务器之间的通信，确保数据传输的机密性。

提供的保密性/私密性的类型和程度

- **用户数据保护**：系统应遵循数据保护法规（如GDPR、CCPA）确保用户数据的机密性、完整性和可用性。
- **权限控制**：系统应通过身份验证和授权机制，确保用户只能访问其权限范围内的功能和数据。

保密性/私密性的风险和安全措施

- **数据泄露预防**：系统应采取措施防止数据泄露，如数据加密、权限控制、日志记录等。
- **安全审计**：系统应定期进行安全审计和渗透测试，识别和修复潜在的安全漏洞。
- **异常检测**：系统应监控用户行为和系统活动，及时发现异常行为或潜在威胁。

保密性/私密性政策

- **数据保留政策**：系统应制定数据保留政策，确保仅在必要时保留用户数据，并在合适的时间删除不再需要的数据。
- **数据访问政策**：系统应限制对用户数据的访问，确保只有授权人员才能访问敏感数据。
- **用户知情权**：系统应向用户提供数据处理的透明度，包括收集、使用和共享的数据类型和目的。

保密性/私密性审核和确证/认可准则

- **安全认证**：系统应符合相关的安全认证标准（如ISO/IEC 27001）以确保数据安全。
- **审核日志**：系统应保留用户操作和管理员操作的详细日志，供审计和追踪之用。
- **隐私政策声明**：系统应提供清晰、透明的隐私政策声明，告知用户其数据的使用方式和保护措施。

3.11 CSCI环境需求

商品交易系统必须运行在指定的计算机硬件和操作系统环境中，以确保系统的稳定性和可靠性。具体要求如下：

- **操作系统**：系统应在Windows操作系统上运行。
- **计算机硬件**：系统需要运行在满足计算机资源需求的硬件配置上。
- **网络环境**：系统应在高速稳定的网络环境中运行，以确保数据传输的可靠性和速度。
- **其他环境要求**：根据系统需求可能需要其他支持软件，如数据库驱动、框架和库等。

3.12 计算机资源需求

3.12.1 计算机硬件需求

以下是商品交易系统软件需求规格说明中计算机硬件需求的表格形式描述：

硬件组件	具体要求
处理器	多核CPU，频率应为2.5 GHz或更高，确保处理能力
内存	至少16 GB RAM，确保高并发下的稳定运行
存储器	高速固态硬盘（SSD），存储容量通常为500 GB或以上
输入/输出设备	标准的输入/输出设备，如键盘、鼠标、显示器
辅助存储器	数据备份和存档用的外部存储设备（网络附加存储或云存储）
通信/网络设备	高速网络接口卡（如千兆以太网或更高），确保稳定快速连接
其他设备	可能需要其他外围设备，如打印机或扫描仪，视业务需求确定

3.12.2 计算机硬件资源利用需求

硬件资源	最大许可使用	备注
处理器能力	不超过80%	保留余量以应对突发流量
存储器容量	不超过75%	确保在高并发情况下的稳定性
输入/输出设备能力	合理范围内	避免过度使用导致性能下降
辅助存储器容量	持续监控	确保足够的可用容量
通信/网络设备能力	不超过80%	确保充足的带宽和低延迟连接

3.12.3 计算机软件需求

软件项	具体要求
操作系统	Windows
数据库管理系统	MySQL 8.0或更高版本
通信/网络软件	支持HTTPS（SSL/TLS）协议
输入和设备模拟器	可能需要设备模拟器来模拟用户操作以进行测试
生产用软件	在生产环境中部署高性能的Web服务器和应用服务器

3.12.4 计算机通信需求

通信需求	具体要求
连接的地理位置	全球互联网，支持全球用户访问
配置和网络拓扑结构	冗余网络架构和高可用性网络配置
传输技术	高速以太网和无线连接技术，确保数据传输的可靠性和速度
数据传输速率	支持千兆以太网或更高传输速率
网关	防火墙和安全网关，确保网络安全
系统使用时间	全天候（24/7）运行服务
传送/接收数据的类型和容量	支持传输HTTP请求/响应和WebSocket等数据类型
传送/接收/响应的时间限制	请求响应时间在毫秒级别，保证用户体验
诊断功能	支持网络诊断工具，如ping、traceroute等

3.13软件质量因素

（1）程序代码算法的可靠性

（2）功能需求的完成度

（3）使用合适的网络协议

（4）开发时间和开发环境

软件质量是指软件与明确和隐含定义的需求相一致的程度。管理角度对软件质量进行度量时，可将影响软件质量的主要因素划分为三组，分别反映用户在使用软件产品时的三种不同倾向或观点：产品运行、产品修改和产品转移。这些因素对软件质量的全面评估至关重要。

产品运行

- **正确性**：软件能够准确地执行明确描述的功能需求。
- **健壮性**：软件在异常或意外条件下的运行能力。
- **效率**：软件在执行任务时的资源利用率，包括时间、内存、处理器等。
- **完整性**：软件对数据和信息的保护能力，包括数据安全和访问控制。
- **可用性**：软件在正常和高负载情况下的可用时间。
- **风险**：软件在运行时对潜在问题或风险的抵御能力。

产品修改

- **可理解性**：软件代码和文档的清晰度和易读性。
- **可维修性**：软件在出现问题时的修复难度和效率。
- **灵活性**：软件对需求变化的适应能力。

- **可测试性**：软件进行测试和验证的难易程度。

产品转移

- **可移植性**：软件在不同环境或平台下运行的能力。
- **可再用性**：软件组件和模块在不同项目中重用的难易程度。
- **互操作性**：软件与其他系统或软件的集成和交互能力。

3.14设计和实现的约束

a. 特殊CSCI体系结构的使用或体系结构方面的需求

- **数据库和其他软件配置项**：系统应使用MySQL 8.0或更高版本作为主要数据库，或根据需求使用其他数据库系统，如PostgreSQL或NoSQL数据库。
- **标准部件、现有的部件的使用**：尽可能采用标准库、框架和现有组件，以提高开发效率和维护性。
- **需方提供的资源**：在设计和实现中，应充分利用需方提供的资源，包括现有数据、硬件、软件等。

b. 特殊设计或实现标准的使用

- **特殊数据标准的使用**：系统应遵循相关数据标准，如JSON或XML用于数据交换，UTF-8用于字符编码。
- **特殊编程语言的使用**：系统应主要使用Node.js作为后端语言，配合JavaScript、TypeScript等。
- **代码质量和风格**：应遵循代码质量和风格指南（例如ESLint规则或代码规范），保持代码的一致性和可读性。

c. 灵活性和可扩展性

- **可扩展性**：系统应具备横向和纵向扩展能力，包括支持添加更多服务器节点或增加数据库容量。
- **模块化设计**：系统应采用模块化设计，以便于将来升级和调整特定功能或组件。
- **可维护性**：系统应设计为易于维护，包括易于理解的代码结构、详细的注释和文档。
- **兼容性**：系统应与现有系统和服务保持兼容，以便于集成和数据交换。

通过满足以上设计和实现的约束，商品网上交易系统将具备稳定性、灵活性和可扩展性，为用户和商家提供高质量的服务。

3.15数据

数据字典内存储了用户表、订单表、快递员表、商品表、店铺表、相关索引、各个表的主码、外码、存储地址等信息。

	A	B	C	D	E	F	G	H	I
1	schema_nm	table_nm	obj_typ	ord	is_key	column_nm	data_typ	nullable	column_d
2	takeoutapp	comment	TBL	1	FK,PK	customer_Customer_id	int(10)	NOT NULL	
3	takeoutapp	comment	TBL	1	FK,PK	menu_Menu_id	int(10)	NOT NULL	
4	takeoutapp	comment	TBL	2	FK,PK	menu_Menu_id	int(10)	NOT NULL	
5	takeoutapp	comment	TBL	2	FK,PK	menu_restaurant_Restaurant_id	int(10)	NOT NULL	
6	takeoutapp	comment	TBL	3		content	varchar(100)	NOT NULL	NULL
7	takeoutapp	comment	TBL	3	FK,PK	customer_Customer_id	varchar(25)	NOT NULL	
8	takeoutapp	comment	TBL	4		comment	varchar(100)	NULL	
9	takeoutapp	customer	TBL	1	PK	customer_id	smallint(5)	NOT NULL	
10	takeoutapp	customer	TBL	1	PK	Customer_id	int(10)	NOT NULL	
11	takeoutapp	customer	TBL	1	PK	Customer_id	varchar(25)	NOT NULL	
12	takeoutapp	customer	TBL	2		store_id	tinyint(3)	NOT NULL	NULL
13	takeoutapp	customer	TBL	2		Name	varchar(25)	NOT NULL	
14	takeoutapp	customer	TBL	3		first_name	varchar(45)	NOT NULL	NULL
15	takeoutapp	customer	TBL	3		Address	varchar(45)	NOT NULL	
16	takeoutapp	customer	TBL	4		last_name	varchar(45)	NOT NULL	NULL
17	takeoutapp	customer	TBL	4		Phone_num	char(11)	NOT NULL	
18	takeoutapp	customer	TBL	5		email	varchar(50)	NULL	NULL
19	takeoutapp	customer	TBL	5		password	varchar(20)	NOT NULL	
20	takeoutapp	customer	TBL	6		address_id	smallint(5)	NOT NULL	NULL
21	takeoutapp	customer	TBL	7		active	tinyint(3)	NOT NULL	NULL
22	takeoutapp	customer	TBL	8		create_date	datetime(8)	NOT NULL	NULL
23	takeoutapp	customer	TBL	9		last_update	timestamp(4)	NULL	NULL
24	takeoutapp	delivery_guy	TBL	1	PK	Delivery_guy_id	int(10)	NOT NULL	
25	takeoutapp	delivery_guy	TBL	2		Name	varchar(45)	NOT NULL	
26	takeoutapp	delivery_guy	TBL	3		Phone_num	char(11)	NOT NULL	
27	takeoutapp	menu	TBL	1	PK	Menu_id	int(10)	NOT NULL	
28	takeoutapp	menu	TBL	2		Name	varchar(45)	NOT NULL	

	A	B	C	D	E	F	G	H	I
29	takeoutapp	menu	TBL	3		Price	varchar(45)	NOT NULL	
30	takeoutapp	menu	TBL	4		Image	varchar(100)	NOT NULL	
31	takeoutapp	menu	TBL	5		Likes	int(10)	NOT NULL	
32	takeoutapp	menu	TBL	6	FK,PK	restaurant_Restaurant_id	int(10)	NOT NULL	
33	takeoutapp	order	TBL	1	PK	Order_id	int(10)	NOT NULL	
34	takeoutapp	order	TBL	2		Time	datetime(8)	NOT NULL	
35	takeoutapp	order	TBL	3		Total_price	float(12)	NOT NULL	
36	takeoutapp	order	TBL	4		Cur_status	tinyint(3)	NULL	
37	takeoutapp	order	TBL	5	FK	customer_Customer_id	int(10)	NOT NULL	
38	takeoutapp	order	TBL	5	FK	customer_Customer_id	varchar(25)	NOT NULL	
39	takeoutapp	order	TBL	6	FK	delivery_guy_Delivery_guy_id	int(10)	NOT NULL	
40	takeoutapp	order_detail	TBL	1	FK,PK	order_Order_id	int(10)	NOT NULL	
41	takeoutapp	order_detail	TBL	1	FK,PK	menu_Menu_id	int(10)	NOT NULL	
42	takeoutapp	order_detail	TBL	2	FK,PK	menu_Menu_id	int(10)	NOT NULL	
43	takeoutapp	order_detail	TBL	2	FK,PK	menu_restaurant_Restaurant_id	int(10)	NOT NULL	
44	takeoutapp	order_detail	TBL	3		amount	int(10)	NOT NULL	
45	takeoutapp	order_detail	TBL	3	FK,PK	order_Order_id	int(10)	NOT NULL	
46	takeoutapp	order_detail	TBL	4		amount	int(10)	NULL	
47	takeoutapp	orderdetailb	VW	1		menu_Menu_id	int(10)	NOT NULL	
48	takeoutapp	orderdetailb	VW	2		menu_restaurant_Restaurant_id	int(10)	NOT NULL	
49	takeoutapp	orderdetailb	VW	3		order_Order_id	int(10)	NOT NULL	
50	takeoutapp	orderdetailb	VW	4		amount	int(10)	NULL	
51	takeoutapp	orderdetailb	VW	5		Menu_id	int(10)	NOT NULL	
52	takeoutapp	orderdetailb	VW	6		Name	varchar(45)	NOT NULL	
53	takeoutapp	orderdetailb	VW	7		Price	varchar(45)	NOT NULL	
54	takeoutapp	orderdetailb	VW	8		Image	varchar(100)	NOT NULL	
55	takeoutapp	orderdetailb	VW	9		Likes	int(10)	NOT NULL	
56	takeoutapp	orderdetailb	VW	10		restaurant_Restaurant_id	int(10)	NOT NULL	
57	takeoutapp	orderdetailb	VW	11		Restaurant_id	int(10)	NOT NULL	
58	takeoutapp	orderdetailb	VW	12		Resname	varchar(45)	NOT NULL	
59	takeoutapp	orderdetailb	VW	13		Address	varchar(45)	NOT NULL	

	A	B	C	D	E	F	G	H	I
55	takeoutapp	orderdetailb	VW	9		Likes	int(10)	NOT NULL	
56	takeoutapp	orderdetailb	VW	10		restaurant_Restaurant_id	int(10)	NOT NULL	
57	takeoutapp	orderdetailb	VW	11		Restaurant_id	int(10)	NOT NULL	
58	takeoutapp	orderdetailb	VW	12		Resname	varchar(45)	NOT NULL	
59	takeoutapp	orderdetailb	VW	13		Address	varchar(45)	NOT NULL	
60	takeoutapp	orderdetailb	VW	14		Phone_num	char(11)	NOT NULL	
61	takeoutapp	restaurant	TBL	1	PK	Restaurant_id	int(10)	NOT NULL	
62	takeoutapp	restaurant	TBL	2		Name	varchar(45)	NOT NULL	NULL
63	takeoutapp	restaurant	TBL	2		Resname	varchar(45)	NOT NULL	
64	takeoutapp	restaurant	TBL	3		Address	varchar(45)	NOT NULL	
65	takeoutapp	restaurant	TBL	4		Phone_num	char(11)	NOT NULL	
66									

3.16 操作

常规操作

- **用户认证和授权**：系统应支持用户登录和权限控制，确保用户只能访问其权限范围内的功能和数据。
- **商品浏览和搜索**：系统应支持用户浏览和搜索商品，并提供高级搜索和筛选功能。
- **购物车管理**：系统应允许用户将商品添加到购物车，并管理购物车中的商品（增、删、改）。
- **订单处理**：系统应支持订单的创建、确认和支付流程，并提供订单状态跟踪。
- **支付处理**：系统应集成安全可靠的支付网关，支持多种支付方式，并确保支付交易的安全性。
- **评价和反馈**：系统应支持用户对购买商品的评价和反馈，供其他用户参考。

特殊操作

- **促销和折扣**：系统应支持特定时间段或特定用户群体的促销和折扣活动，并在用户购物时自动应用。
- **库存管理**：系统应实时监控商品库存，确保库存信息的准确性，并在库存不足时发出警报。
- **活动和通知**：系统应支持商家发布活动、通知用户，并通过电子邮件或推送通知等方式告知用户。

初始化操作

- **环境配置**：系统应提供配置文件或环境变量来设置数据库连接、API密钥、日志级别等。
- **数据初始化**：在系统初始化时，应检查数据库中的数据，并在必要时进行数据迁移或初始化。
- **组件加载**: 系统初始化时，应确保React前端和Node.js后端组件正常加载。

恢复操作

- **数据备份和恢复**：系统应定期备份数据，并提供恢复操作，以便在数据丢失或损坏时恢复系统。
- **服务重启**：系统应支持服务的平稳重启，以便在代码更新或异常情况下快速恢复服务。
- **状态恢复**：系统应在重启后恢复服务的状态，包括会话、缓存等信息。
- **容错机制**：系统应具备容错机制，以在特定组件或服务故障时快速切换到备用系统。

3.17 故障处理

a. 识别软件系统问题

- **错误日志记录**：系统应实时记录所有软硬件错误，包括详细的错误信息和发生错误的时间和位置。
- **监控和报警**：系统应持续监控关键性能指标（如CPU、内存、网络使用率）以及业务指标（如订单处理时间、交易成功率），并在异常情况时发出警报。
- **自检和诊断**：系统应具备自检和诊断功能，识别系统问题或异常行为，以便采取适当措施。

b. 错误信息

- **明确错误信息**：系统应在发生错误时提供明确的错误信息，包括错误代码、错误描述和建议的补救措施。
- **用户友好的反馈**：对于用户可见的错误信息，系统应提供简洁、友好的反馈，以便用户了解错误情况和可能的解决方案。
- **管理员错误日志**：对于开发人员或管理员，系统应提供详细的错误日志，包括堆栈跟踪和上下文信息，以便快速定位问题。

c. 补救措施

- **重试机制**：系统应为关键操作（如订单处理、支付交易）提供重试机制，以在错误后自动尝试重新执行操作。
- **数据回滚**：对于数据库相关操作，系统应具备事务回滚功能，以确保在错误发生时数据的一致性和完整性。
- **故障隔离**：系统应设计为具有故障隔离能力，确保单个组件或服务的故障不会影响整个系统的运行。
- **备用系统**：系统应提供备用系统或冗余服务，在主系统故障时自动切换，确保业务连续性。
- **快速修复**：在确定错误原因后，开发人员应及时提供修复措施，并在必要时发布补丁或更新。
- **用户通知**：在必要情况下，系统应及时通知用户出现的问题以及可能的影响和解决时间。

3.18算法说明

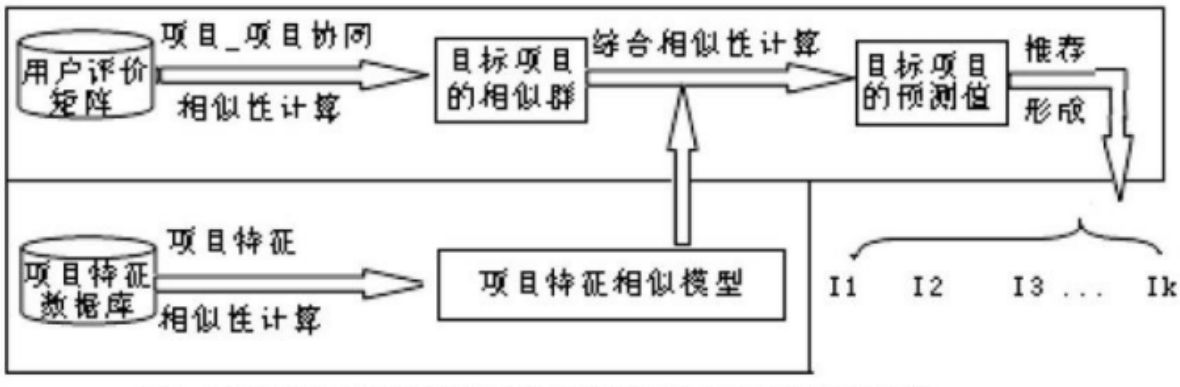
商品智能推荐算法

推荐算法的目的是为用户提供更有针对性、个性化、高效的资源，从而提高应用资源的利用率。可以采用基于ItemCF算法与CB算法的混合推荐技术，以期获得更高的推荐准确性和高效性，将ItemCF算法与CB算法的推荐结果进行加权混合，获得用户在推荐算法下的最佳匹配商品。

1. 基于物品的协同过滤算法（ItemCF）

协同过滤算法是目前大多数商用推荐系统中的主流推荐算法，该算法通过分析用户的行为习惯和兴趣喜好对目标用户的推荐对象进行筛选。其基本思想是：如果两个用户之间具有相近或相似的兴趣偏好，那么他们对于任意信息的评价结果也是相近的。由于ItemCF适用于物品个数远少于用户个数的场景（显然网上商城符合这样的场景特点），而UserCF则正好相反，所以我们选择采用ItemCF算法来进行推荐。

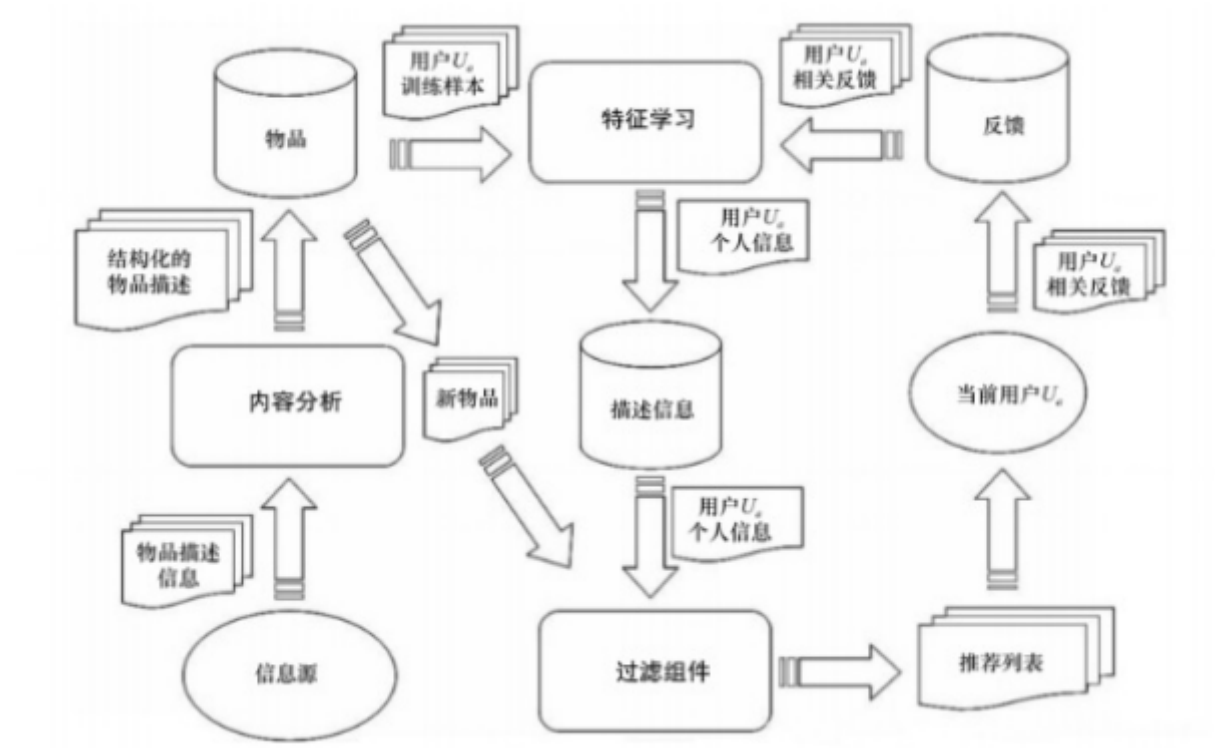
算法的流程可通过如下流程图来体现：



其中相似性计算采用余弦相似度的方法。最终按照物品相似度预测值对其进行降序排列，获取topN作为推荐的结果。

2. 基于内容的推荐算法（CB）

考虑到CF算法的局限性和CB算法与ItemCF算法在某些程度上的互补性，可以加入基于内容的推荐来使我们的推荐系统更准确高效。CB算法的流程如下：



该算法主要通过构建用户画像（user profile）和物品画像（item profile），并对两者进行相似度计算来对物品进行推荐。

3. 混合算法解决物品冷启动问题

ItemCF对物品冷启动敏感，对于新加入系统的物品，需要一段时间的用户评价积累才有可能对该物品进行推荐，换句话说，ItemCF无法很好地推荐新加入的物品。而CB算法因为直接考量的是物品的内在特征，所以并不会出现冷启动的问题，其对于新加入系统的物品和原有物品的推荐概率是相同的。我们通过对两种推荐算法的加权混合来解决冷启动的问题。

4. 数据的稀疏性问题

一般来说平台内都有着庞大的项目数量，可是被某一个用户进行评价过的项目所占比例却很低，这一现象必会造成用户-项目的评分矩阵中“0”元素占据绝大多数位置。这会对CF算法造成很大的影响。而CB算法收到的影响则相对小一些。我们的项目通过两种算法的融合来降低数据稀疏问题对于推荐准确性的影响。

3.19 有关人员需求

职责	小组成员
项目需求分析	所有人参与
数据库管理系统	霍泉如、石子超
前台系统实现	陈一苇
后台系统实现	鲍俊杰、夏楠翔

人员数量

- **开发团队**：五位学生负责项目的开发和实施，包括前端、后端和数据库等各个方面。

技能等级

- **前端开发**：至少一名学生熟悉React框架、JavaScript和前端开发技术。
- **后端开发**：至少一名学生熟悉Node.js框架和后端开发，包括API设计和数据处理。
- **数据库**：至少一名学生熟悉MySQL或其他数据库管理系统，包括数据设计、查询和优化。
- **其他技术**：团队成员应熟悉软件工程的基础知识，如需求分析、设计模式、代码质量和版本控制。

责任期

- **职责划分**：团队成员应根据各自的技能和知识划分职责，确保项目的各个方面都得到负责和推进。
- **沟通与协作**：团队成员应保持良好的沟通和协作，确保项目的进展和解决问题。

培训需求

- **新技术培训**：在项目开始前或进行中，团队可能需要对新的技术或框架进行培训。
- **工具和平台**：团队应熟悉项目所需的工具和平台，如版本控制系统（如Git）和项目管理工具。

用户数量需求

- **并发用户**：系统应设计为支持一定数量的并发用户，根据预期用户数量确定系统的并发处理能力。

内在帮助和培训能力需求

- **用户帮助文档**：系统应提供用户帮助文档或在线帮助，解释系统的功能和使用方法。
- **管理员培训**：如果需要管理员来管理系统，系统应提供管理员培训和指导。

3.20有关培训需求

一般的，为向项目接收方移交可交付产品需要执行以下活动：包括计划/协调会议，要交付给项目接收方的准备，产品支持环境的运输、安装和检测，运行产品的包装、安装和检测，以及接收人员的培训等。培训的地点暂定为学校的机房。培训的内容为：熟练掌握产品的调试及使用方法。交付产品所需人员为三人，要求十分了解此项目的相关功能，全程参与了此项目的开发制作，并可以对平台使用中可能出现的一些错误进行修正，并能对平台进行后期的调试以及维护。

3.21有关后勤需求

系统维护：在软件运行中可能会出现问题以及安卓系统版本迭代等，需要不断维护。

（1）改正性维护：在交付使用后，因开发时测试的不彻底、不完全，必然会有部分隐藏的错误遗留到运行阶段。这些隐藏下来的错误在某些特定的使用环境下就会暴露出来。为了识别和纠正错误、改正性能上的缺陷、排除实施中的误使用，应当进行的诊断和改正错误的过程就叫做改正性维护 。

（2）适应性维护：在使用过程中，外部环境（新的硬、软件配置）数据环境（数据库、数据格式、数据输入/输出方式、数据存储介质）可能发生变化 。

（3）完善性维护：在使用过程中，用户往往会提出新的功能与性能要求。为满求了足这些要求，需要修改或再开发，以扩充功能、增强性能、改进加工效率、提高可维护性。

（4）预防性维护：采用先进的软件工程方法对需要维护的软件或软件中的某一部分（重新）进行设计、编制和测试。

3.22其他需求

无。

3.23包装需求

1. 界面：界面统一标准，充分考虑用户的使用习惯，适量添加美工，以为用户第一原则进行设计。
2. 程序的结构：程序结构尽量做到功能模块化设计，便与升级、维护。
3. 软件文档：软件文档相当重要，如MIS等更重要，要规范化，排版漂亮、美观。

3.24需求的优先次序和关键程度

操作类型	优先级	操作说明
用户认证和授权	高	支持用户登录和权限控制，确保用户只能访问其权限范围内的功能和数据。
商品浏览和搜索	中	支持用户浏览和搜索商品，并提供高级搜索和筛选功能。
购物车管理	中	允许用户将商品添加到购物车，并管理购物车中的商品（增、删、改）。
订单处理	高	支持订单的创建、确认和支付流程，并提供订单状态跟踪。
支付处理	高	集成安全可靠的支付网关，支持多种支付方式，确保支付交易的安全性。
评价和反馈	低	支持用户对购买商品的评价和反馈，供其他用户参考。
促销和折扣	中	支持特定时间段或特定用户群体的促销和折扣活动，并在用户购物时自动应用。
库存管理	高	实时监控商品库存，确保库存信息的准确性，并在库存不足时发出警报。
活动和通知	中	支持商家发布活动、通知用户，通过电子邮件或推送通知告知用户。
状态恢复	高	重启后恢复服务的状态，包括会话、缓存等信息。
容错机制	高	具备容错机制，在特定组件或服务故障时快速切换到备用系统。
推荐算法	低	根据用户信息进行推荐，提升用户体验。

4合格性规定

需求类别	合格性方法	说明
用户认证和授权	测试	使用测试工具验证身份验证和授权机制，确保用户只能访问其权限范围内的功能和数据。
商品浏览和搜索	演示	通过演示功能操作，确保用户能够浏览和搜索商品，并提供高级搜索和筛选功能。
购物车管理	演示	通过演示购物车操作，验证用户能够添加、删除和修改购物车中的商品。
订单处理	测试	使用测试工具验证订单创建、确认和支付流程，并提供订单状态跟踪。
支付处理	测试	集成支付网关后，通过测试交易处理，确保支付过程的安全性和可靠性。
评价和反馈	演示	通过演示用户评价和反馈功能，验证用户能够提供对购买商品的评价和反馈。
促销和折扣	演示	通过演示促销和折扣功能，验证系统能够在特定时间段或用户群体中应用促销和折扣活动。
库存管理	测试、分析	使用测试工具验证库存监控功能，确保库存信息的准确性；通过分析库存数据，确保库存管理的正确性。
活动和通知	演示	通过演示活动和通知功能，验证系统能够支持商家发布活动并通知用户。
状态恢复	演示、测试	通过演示和测试状态恢复功能，确保系统在重启后恢复服务的状态，包括会话、缓存等信息。
容错机制	测试、分析	使用测试工具验证容错机制，确保系统能够在特定组件或服务故障时快速切换到备用系统；通过分析系统日志和数据，评估容错机制的有效性。
推荐算法	测试、分析	使用测试工具验证推荐算法的准确性和有效性；通过分析用户数据和推荐结果，确保推荐算法的性能。

5需求可追踪性

5.1 从CSCI需求到系统（或子系统）需求的可追踪性

本章包括从每个CSCI的需求到其所涉及的系统（或子系统）需求的可追踪性。我们可以通过以下方式来描述这种可追踪性：

- 在第3章中，每个需求后进行注释，明确指出该需求追踪到哪个系统或子系统需求。
- 使用需求映射表来列出每个CSCI需求及其所涉及的系统（或子系统）需求。例如：

CSCI需求	系统/子系统需求
用户认证和授权	系统需求：用户认证和权限控制
商品浏览和搜索	系统需求：商品展示、搜索和筛选功能
购物车管理	系统需求：购物车操作（增、删、改）
订单处理	系统需求：订单创建、确认和支付流程
支付处理	系统需求：支付网关集成与支付交易处理
评价和反馈	系统需求：商品评价与用户反馈
促销和折扣	系统需求：促销活动与折扣管理
库存管理	系统需求：库存监控与管理
活动和通知	系统需求：商家活动发布与用户通知
状态恢复	系统需求：服务状态恢复
容错机制	系统需求：容错处理与备份恢复
推荐算法	系统需求：个性化推荐与算法应用

5.2 从系统（或子系统）需求到CSCI需求的可追踪性

本部分描述了分配到CSCI的每个系统（或子系统）需求以及它们所涉及的CSCI需求的可追踪性。

- 在系统需求规格说明中注明需求与CSCI需求之间的映射关系，以确保需求的一致性和准确性。
- 使用需求追踪矩阵或工具来显示系统（或子系统）需求与CSCI需求之间的映射。例如：

系统/子系统需求	CSCI需求
用户认证和权限控制	用户认证和授权
商品展示、搜索和筛选功能	商品浏览和搜索
购物车操作（增、删、改）	购物车管理
订单创建、确认和支付流程	订单处理
支付网关集成与支付交易处理	支付处理
商品评价与用户反馈	评价和反馈
促销活动与折扣管理	促销和折扣
库存监控与管理	库存管理
商家活动发布与用户通知	活动和通知
服务状态恢复	状态恢复
容错处理与备份恢复	容错机制
个性化推荐与算法应用	推荐算法

6尚未解决的问题

地址管理

- **地址验证**：尚未实现有效的地址验证功能，无法确保用户提供的地址是有效且可配送的。
- **地址簿管理**：系统尚未提供用户管理多个地址的功能，例如添加、编辑和删除地址簿。

支付处理

- **多样化支付方式**：系统尚未完全支持支付方式。
- **支付安全**：目前的支付处理尚未完全实现强大的安全措施，例如反欺诈系统、二次身份验证等。

推荐算法

- **个性化推荐**：当前的推荐算法尚未实现个性化推荐功能，需要根据用户的行为数据进行精准推荐。
- **推荐模型**：系统尚未采用先进的推荐模型，如协同过滤或机器学习模型，以提高推荐的准确性。

商品评价

- **评价审核**：系统尚未提供对商品评价进行审核的功能，可能存在虚假或不合适的评价。
- **评价反馈**：尚未实现对用户评价的反馈和回应功能，例如商家回复用户评价。
- **评价排序和筛选**：系统尚未提供评价的排序和筛选功能，以便用户查看最相关的评价。

7注解

无。

附录

无。