

软件(结构)设计说明(SDD)

软件(结构)设计说明(SDD)

1 引言

- 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 CSCI级设计决策

- 3.1 输入和输出的设计决策
- 3.2 CSCI行为的设计决策
- 3.3 数据库/数据文件的设计决策
- 3.4 安全性、保密性、私密性设计决策
- 3.5 其他设计决策

4 CSCI体系结构设计

4.1 体系结构

- 4.1.1 程序（模块）划分
- 4.1.2 程序(模块)层次结构关系
 - 4.1.2.1 软件系统总体功能流程图
 - 4.1.2.2 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

4.2 全局数据结构说明

- 4.2.1 常量
- 4.2.2 变量
 - 变量表
- 4.2.3 数据结构
 - 数据库建表
 - 数据字典

4.3 CSCI部件

- 4.3.1 软件配置项标识
- 4.3.2 软件配置项的静态关系
- 4.3.3 软件配置项的用途和需求分配

4.4 执行概念

- 4.4.1 动态关系概述
- 4.4.2 控制流和数据流
- 4.4.3 动态控制序列
- 4.4.4 状态转换过程
- 4.4.5 配置项之间的优先关系
- 4.4.6 并发执行和动态分配

- 4.4.7 异常处理
- 4.4.8 中断处理
- 4.4.9 时间/序列关系
- 4.4.10 对象/进程/任务的动态创建与删除
- 4.5接口设计
 - 4.5.1接口标识与接口图
 - 接口标识与特性表
 - E-R图

5 CSCI详细设计

- 5.1 用户管理模块 (UMM-001)
 - 5.1.1 项目唯一标识符
 - 5.1.2 配置项设计决策
 - 5.1.3 设计约束和特征
 - 5.1.4 输入和输出数据元素
 - 5.1.5 逻辑
- 5.2 商品管理模块 (PMM-002)
 - 5.2.1 项目唯一标识符
 - 5.2.2 配置项设计决策
 - 5.2.3 设计约束和特征
 - 5.2.4 输入和输出数据元素
 - 5.2.5 逻辑
- 5.3 购物车模块 (SCM-003)
 - 5.3.1 项目唯一标识符
 - 5.3.2 配置项设计决策
 - 5.3.3 设计约束和特征
 - 5.3.4 输入和输出数据元素
 - 5.3.5 逻辑
- 5.4 订单管理模块 (OMM-004)
 - 5.4.1 项目唯一标识符
 - 5.4.2 配置项设计决策
 - 5.4.3 设计约束和特征
 - 5.4.4 输入和输出数据元素
 - 5.4.5 逻辑
- 5.5 收藏夹模块 (FMM-005)
 - 5.5.1 项目唯一标识符
 - 5.5.2 配置项设计决策
 - 5.5.3 设计约束和特征
 - 5.5.4 输入和输出数据元素
 - 5.5.5 逻辑
- 5.6 个人信息管理模块 (PIMM-006)
 - 5.6.1 项目唯一标识符
 - 5.6.2 配置项设计决策
 - 5.6.3 设计约束和特征
 - 5.6.4 输入和输出数据元素
 - 5.6.5 逻辑
- 5.7 数据库管理模块 (DBMM-007)
 - 5.7.1 项目唯一标识符
 - 5.7.2 配置项设计决策
 - 5.7.3 设计约束和特征
 - 5.7.4 输入和输出数据元素
 - 5.7.5 逻辑
- 5.8 用户界面模块 (UIM-008)

- 5.8.1 项目唯一标识符
 - 5.8.2 配置项设计决策
 - 5.8.3 设计约束和特征
 - 5.8.4 输入和输出数据元素
 - 5.8.5 逻辑
- 5.9 支付处理模块 (PPM-009)
 - 5.9.1 项目唯一标识符
 - 5.9.2 配置项设计决策
 - 5.9.3 设计约束和特征
 - 5.9.4 输入和输出数据元素
 - 5.9.5 逻辑
- 5.10 通知管理模块 (NMM-010)
 - 5.10.1 项目唯一标识符
 - 5.10.2 配置项设计决策
 - 5.10.3 设计约束和特征
 - 5.10.4 输入和输出数据元素
 - 5.10.5 逻辑

6 需求的可追踪性

- a. 软件配置项到CSCI需求的可追踪性
- b. CSCI需求到软件配置项的可追踪性

7注解

地址管理
支付处理
推荐算法
商品评价
商品管理

附录

1 引言

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5月

1.2 系统概述

本系统是一个面向广大消费者的在线交易平台，旨在提供一个安全、便捷、功能丰富的购物环境。其一般性质包括：

- 用途：**为用户提供从商品浏览、选择、下单到支付的一站式购物服务。
- 系统开发历史：**该系统的开发始于2024年，基于对市场需求的分析和用户调研，采用敏捷开发模式，分阶段实现和发布主要功能。
- 运行和维护历史：**初始版本预计在2024年第一季度上线，计划每季度进行一次主要功能更新和维护。
- 用户：**普通消费者、商家用户
- 开发方：**山东大学-wangwang_team

1.3 文档概述

本文档旨在详细描述商品网上交易系统的结构设计，以确保系统设计的清晰、完整和可实施性。本文件将用于指导开发团队的设计和实现工作，并提供项目干系人评审的基础。文档的使用需遵守公司内部的保密性规定，确保未授权人员无法访问敏感信息。

1.4 基线

- 需求规格说明书：**描述了系统的功能需求和非功能需求。
- 系统架构设计文档：**提供了系统的总体架构设计方案。
- 技术规范和标准：**包括编码规范、接口标准、安全规范等。
- 项目计划书：**包含项目的时间表、资源分配、风险管理等内容。

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 CSCI级设计决策

本章描述了网上交易系统的CSCI级设计决策。这些决策包括对CSCI行为的设计决策，以及其他影响CSCI组成的软件配置项的选择与设计决策。以下内容涵盖了系统设计的各个方面，包括输入和输出设计、行为响应设计、数据呈现设计、安全性设计，以及其他关键设计决策。

3.1 输入和输出的设计决策

网上交易系统必须与其他系统、硬件配置项（HWCI）、其他软件配置项（CSCI）以及用户进行交互。以下是对这些交互的具体设计决策：

- a. **用户输入和输出**：用户通过图形用户界面（GUI）输入交易指令、查询账户信息和查看交易记录。系统将生成确认信息、错误信息和交易结果反馈给用户。GUI设计需确保易用性和响应速度，以满足用户体验要求。
- b. **与支付网关的接口**：系统将与第三方支付网关进行交互，以处理支付和结算。接口设计需遵循支付网关提供的API标准，并确保数据传输的安全性和可靠性。

3.2 CSCI行为的设计决策

网上交易系统需要响应各种用户输入和系统条件，执行相应的操作。以下是有关CSCI行为的具体设计决策：

- a. **交易指令处理**：系统需快速响应用户的交易指令，包括买入、卖出和撤单操作。交易处理模块需保证高效性和准确性，避免因延迟导致的用户损失。
- b. **账户管理**：系统需提供用户账户的查询、修改和管理功能。需确保账户数据的一致性和完整性，并在响应时间上满足用户需求。
- c. **异常处理**：系统需能够处理非法输入和异常情况，如网络故障、支付失败等。需设计有效的错误处理机制，向用户提供明确的错误信息和解决建议。

3.3 数据库/数据文件的设计决策

网上交易系统涉及大量数据的存储和处理，以下是有关数据库和数据文件的具体设计决策：

- a. **数据存储**：用户账户信息、交易记录和系统日志等数据需存储在高可用性和高性能的数据库中。需设计合理的数据模型，确保数据的完整性和一致性。
- b. **数据展示**：用户界面需提供友好的数据展示功能，支持多种查询和筛选条件。需优化数据查询和展示的性能，确保响应速度。

3.4 安全性、保密性、私密性设计决策

为满足系统的安全性、保密性和私密性需求，需采取以下设计决策：

- a. **数据加密**：所有敏感数据在存储和传输过程中均需加密。采用强大的加密算法，确保数据不被非法窃取或篡改。

b. **用户认证**：系统需采用多因素认证机制，确保用户身份的唯一性和真实性。包括密码、短信验证码、指纹识别等多种认证方式。

c. **访问控制**：根据用户角色和权限，设计严格的访问控制机制，确保用户只能访问和操作其权限范围内的数据。

3.5 其他设计决策

为提供系统的灵活性、可用性和可维护性，需采取以下设计决策：

a. **模块化设计**：系统需采用模块化设计，将不同功能模块独立开发和维护，增强系统的扩展性和维护性。

b. **性能优化**：针对系统的关键性能指标，如交易处理速度、数据查询响应时间等，需进行专项优化，确保系统在高负载情况下的稳定性和性能。

c. **备份与恢复**：设计完善的数据备份和恢复机制，确保在系统故障或数据丢失情况下，能够快速恢复正常运行。

以上设计决策为网上交易系统的CSCI级设计提供了明确的指导，确保系统能够满足用户需求和关键性需求，具备高安全性、高可靠性和高可用性。

4 CSCI体系结构设计

本章描述网上交易系统的CSCI体系结构设计。若设计的部分或全部依赖于系统状态或方式，将指出这种依赖性。设计信息如在多条中出现，将描述一次并在其他条目引用。将提供理解这些设计所需的设计约定。

4.1 体系结构

4.1.1 程序（模块）划分

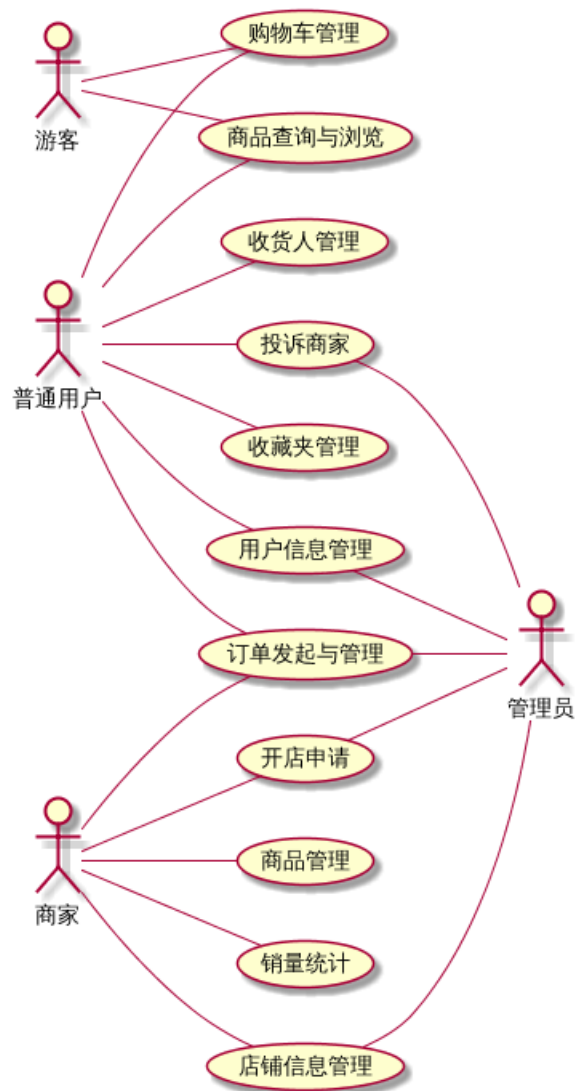
以下图表列出了网上交易系统内每个程序模块的名称、标识符、功能及其所包含的源代码文件。

模块名称	功能描述	源代码文件
用户界面模块	提供用户与系统交互的界面，包括登录、查询、交易等	/src/components/UserInterface.js
用户认证与授权模块	处理用户登录、权限控制，确保用户只能访问其权限范围内的数据	/src/auth/AuthController.js
商品浏览和搜索模块	提供商品的浏览和搜索功能，包括高级搜索和筛选	/src/components/ProductSearch.js
购物车管理模块	允许用户管理购物车中的商品（添加、删除、修改）	/src/components/CartManager.js
订单处理模块	支持订单的创建、确认和支付流程，并提供订单状态跟踪	/src/controllers/OrderController.js
支付处理模块	集成支付网关，支持多种支付方式，确保支付交易安全	/src/controllers/PaymentController.js
评价和反馈模块	支持用户对购买商品的评价和反馈	/src/components/ReviewFeedback.js
数据库管理模块	管理系统数据库，提供数据存储和查询服务	/src/db/DatabaseManager.js
安全管理模块	负责系统的安全性，包括认证、授权和加密	/src/security/SecurityManager.js
错误处理模块	处理系统运行中的各种错误和异常	/src/middleware/ErrorHandler.js
日志管理模块	记录系统运行日志，供运维和审计使用	/src/middleware/Logger.js

以上模块分工明确，确保系统各项功能的实现和高效运行。

4.1.2 程序(模块)层次结构关系

4.1.2.1 软件系统总体功能流程图

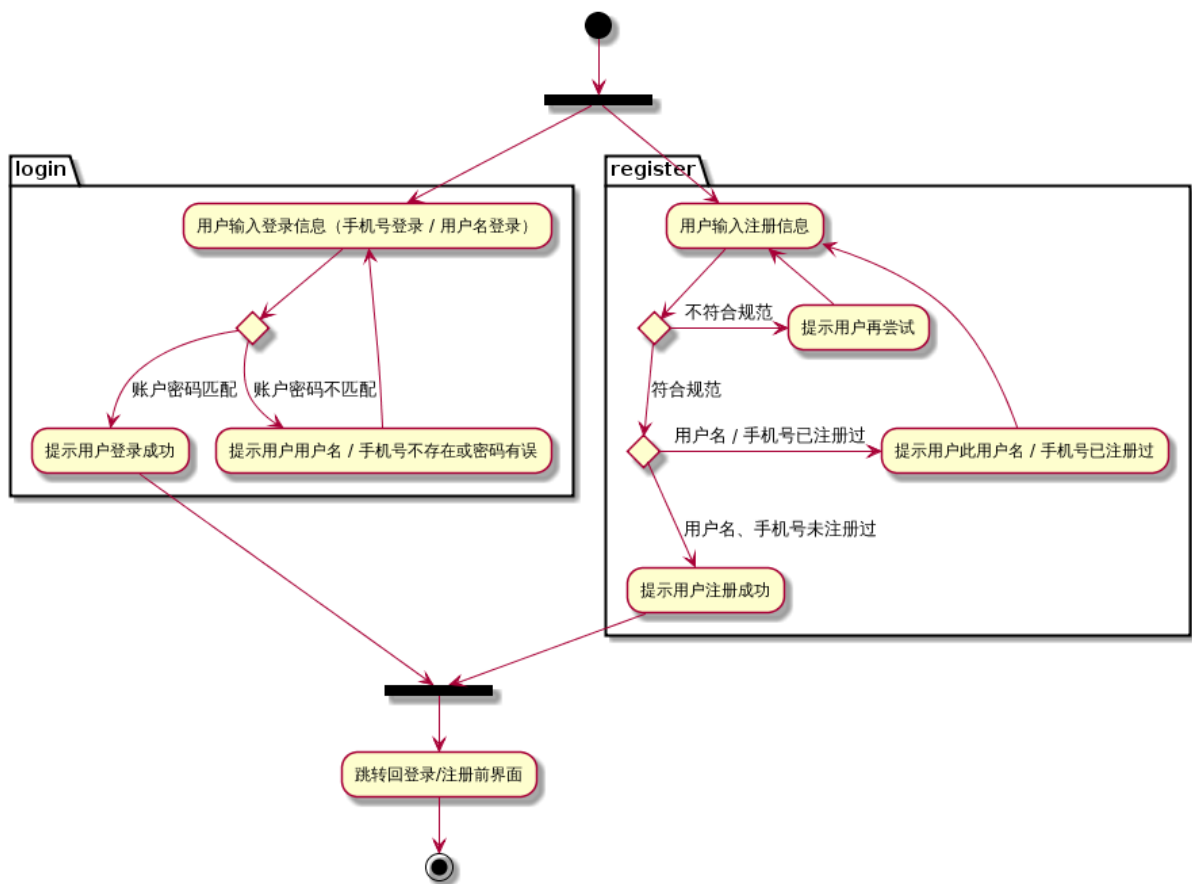


4.1.2.2 软件子系统功能/对象结构

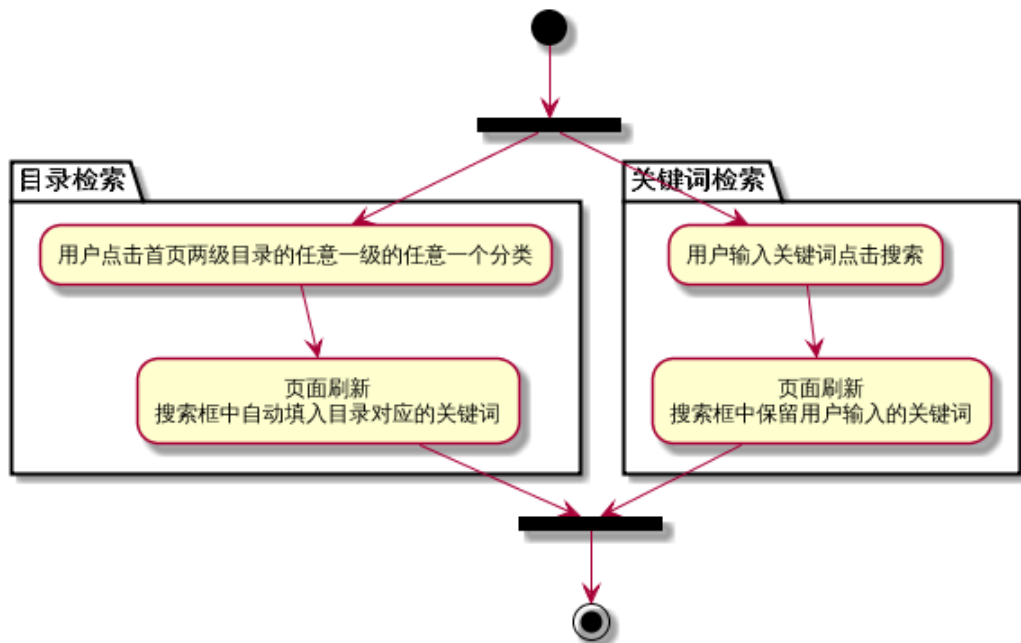
用户登陆与注册

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：



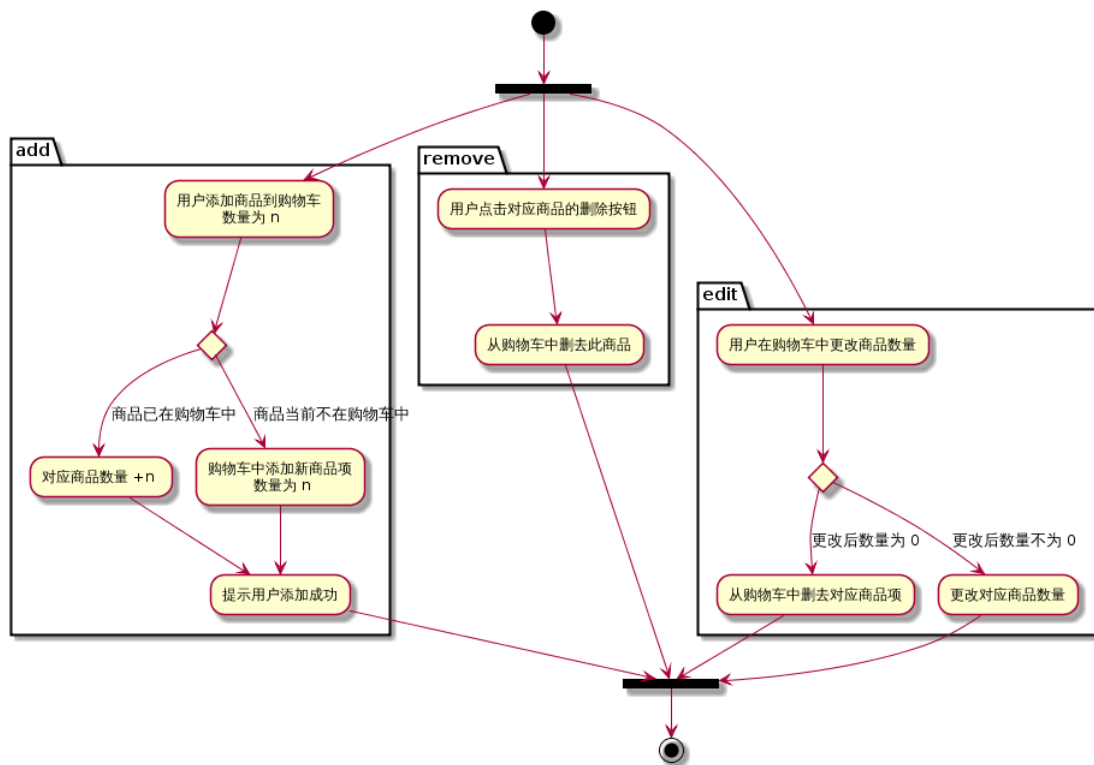
游客 / 用户商品查询与浏览流程图



用户购物车管理

游客 / 用户可以通过商品页面直接添加商品到购物车。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

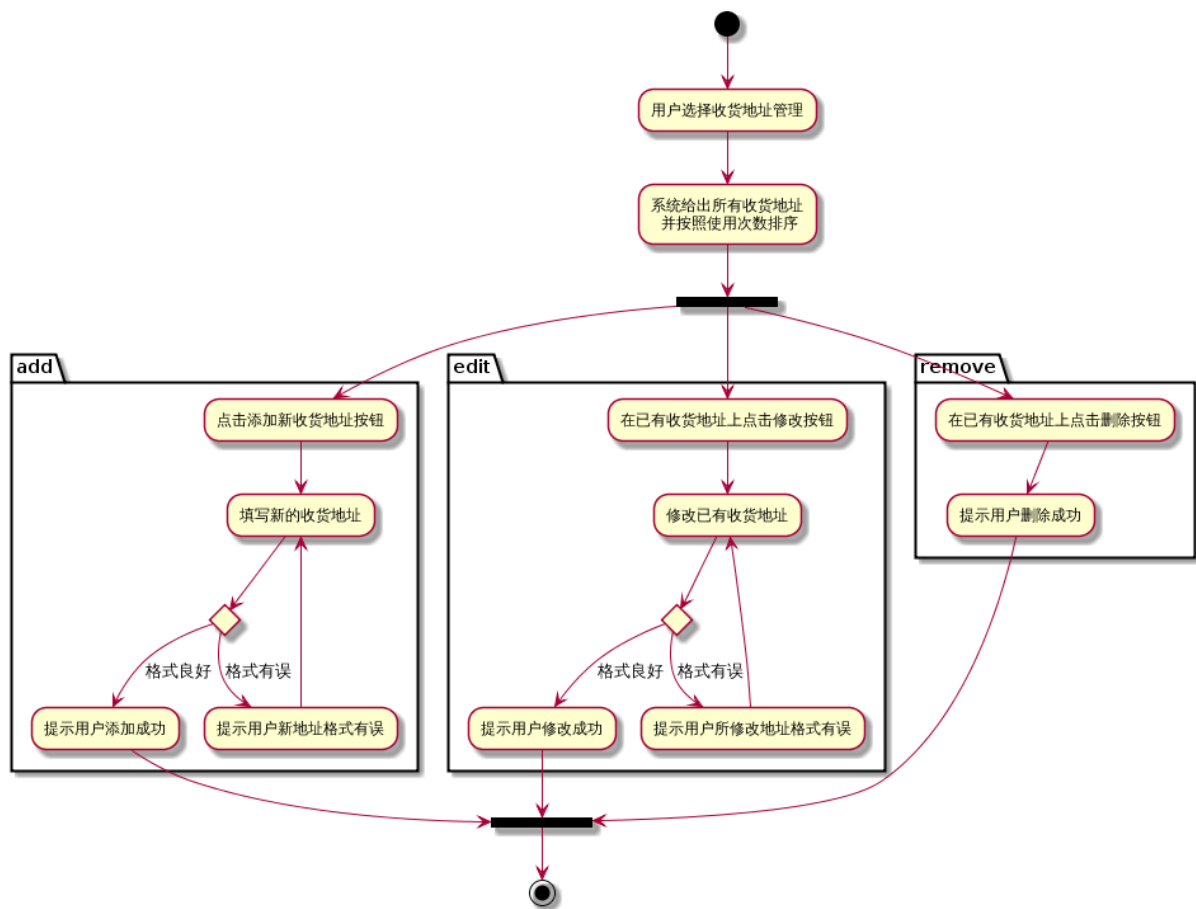
用户可以修改自己除用户名以外的任何个人信息，但不允许自主删除帐号。

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。

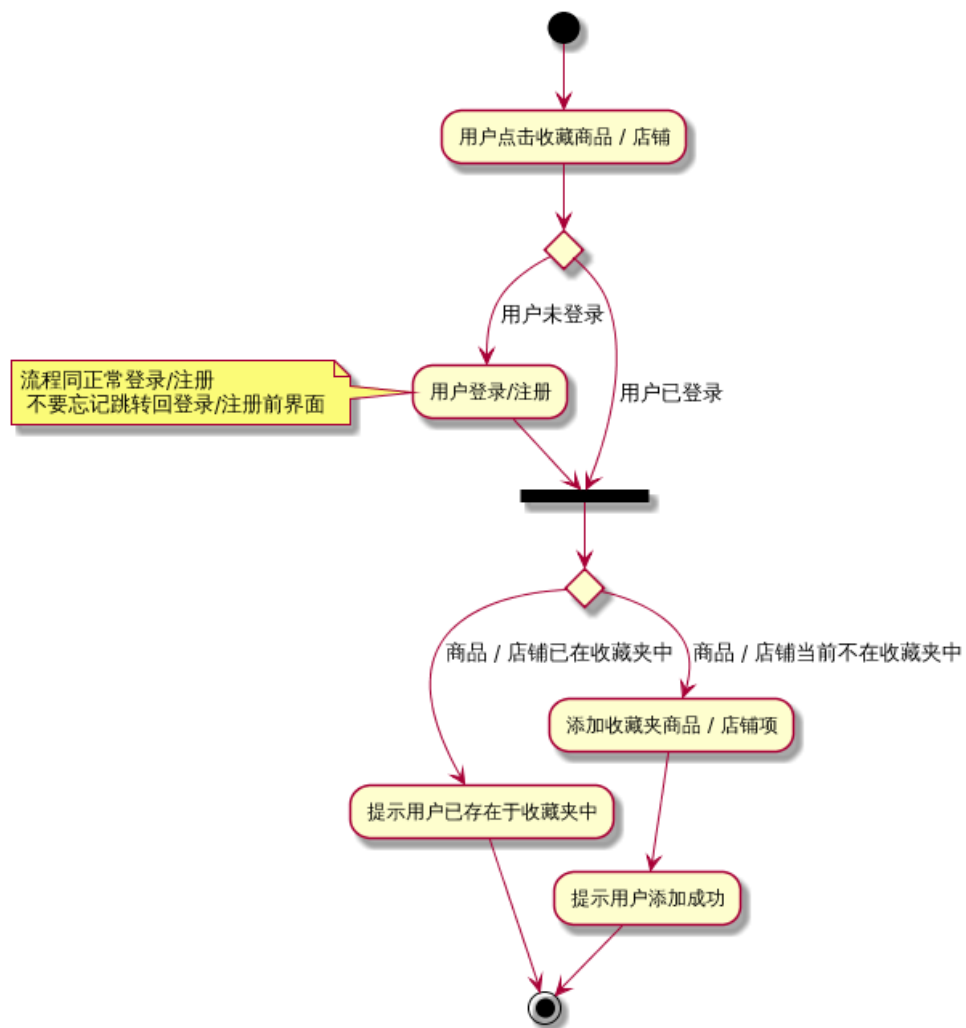
用户搜索用例活动图如下：



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。

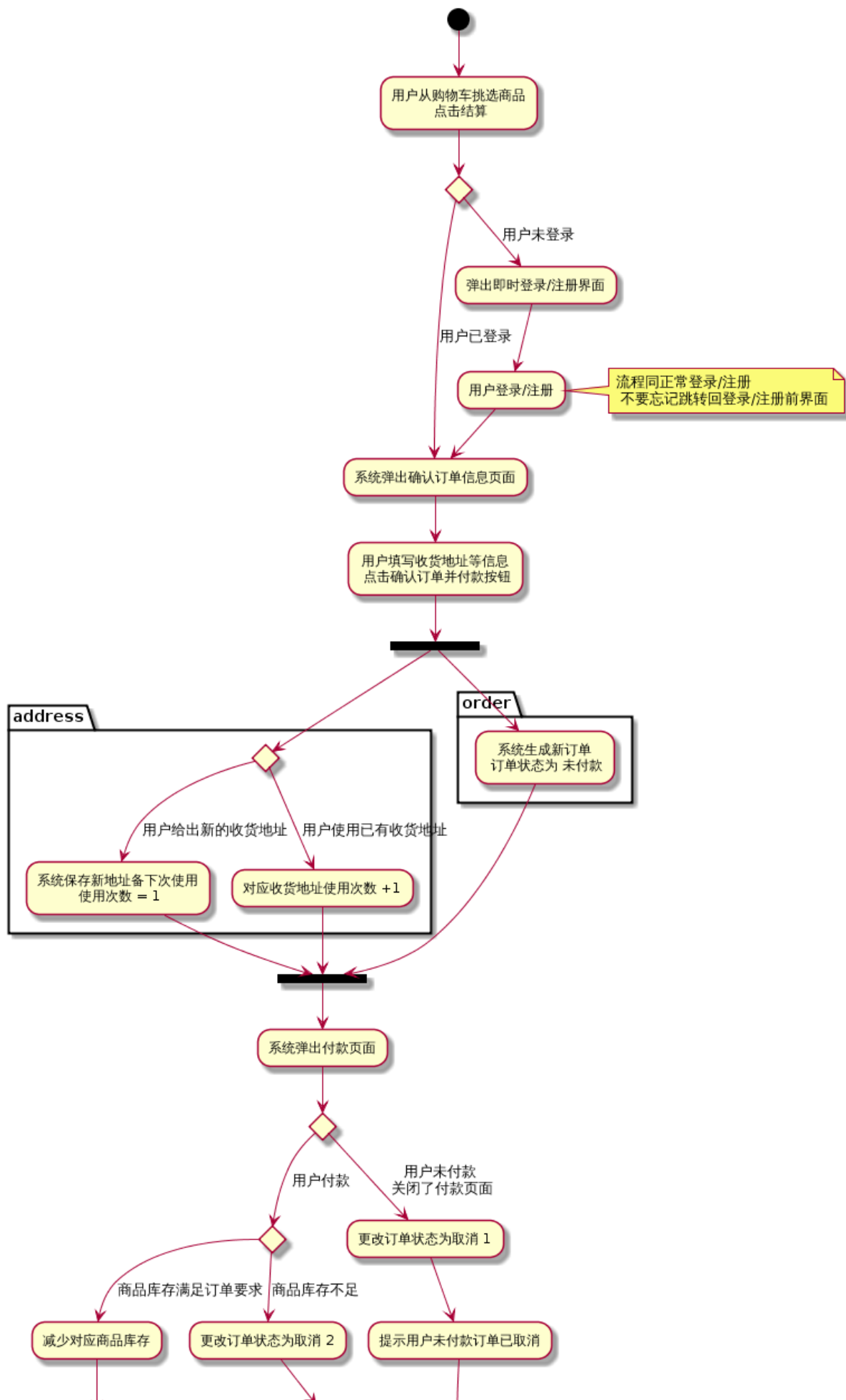
用户收藏夹管理用例活动图如下：

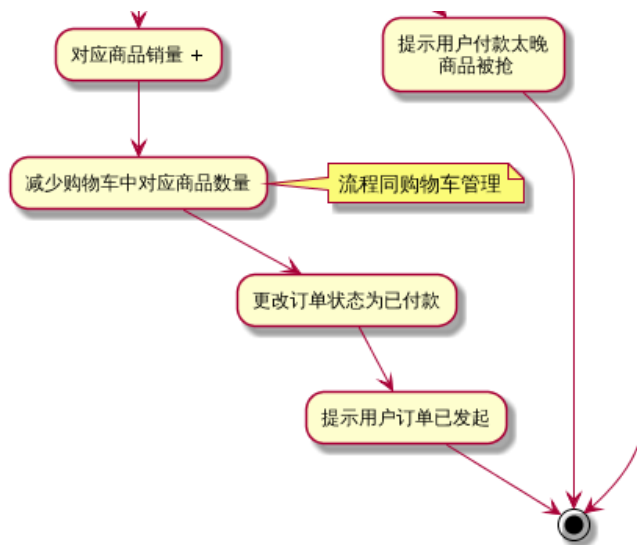


用户发起订单

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：

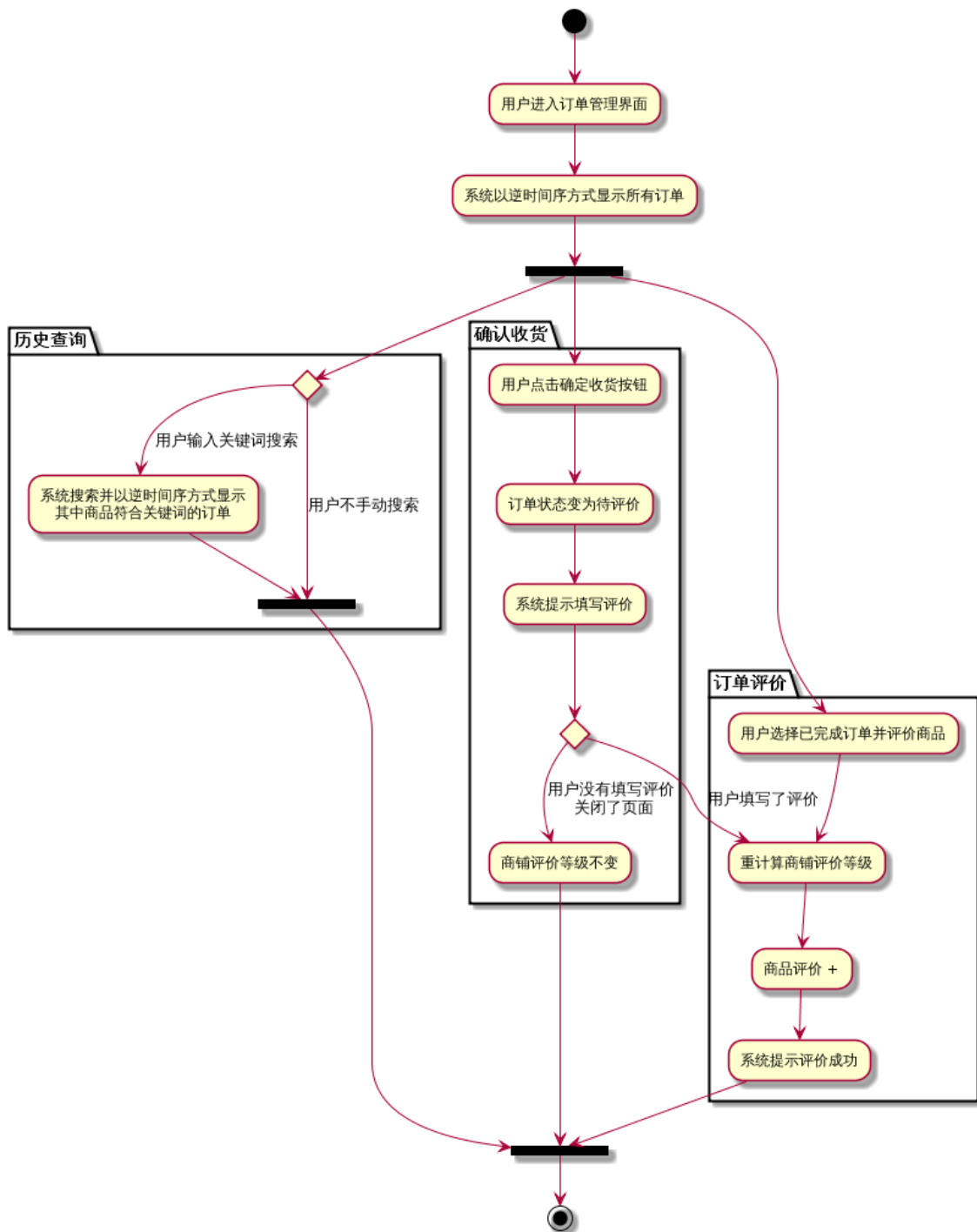




用户管理订单

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。

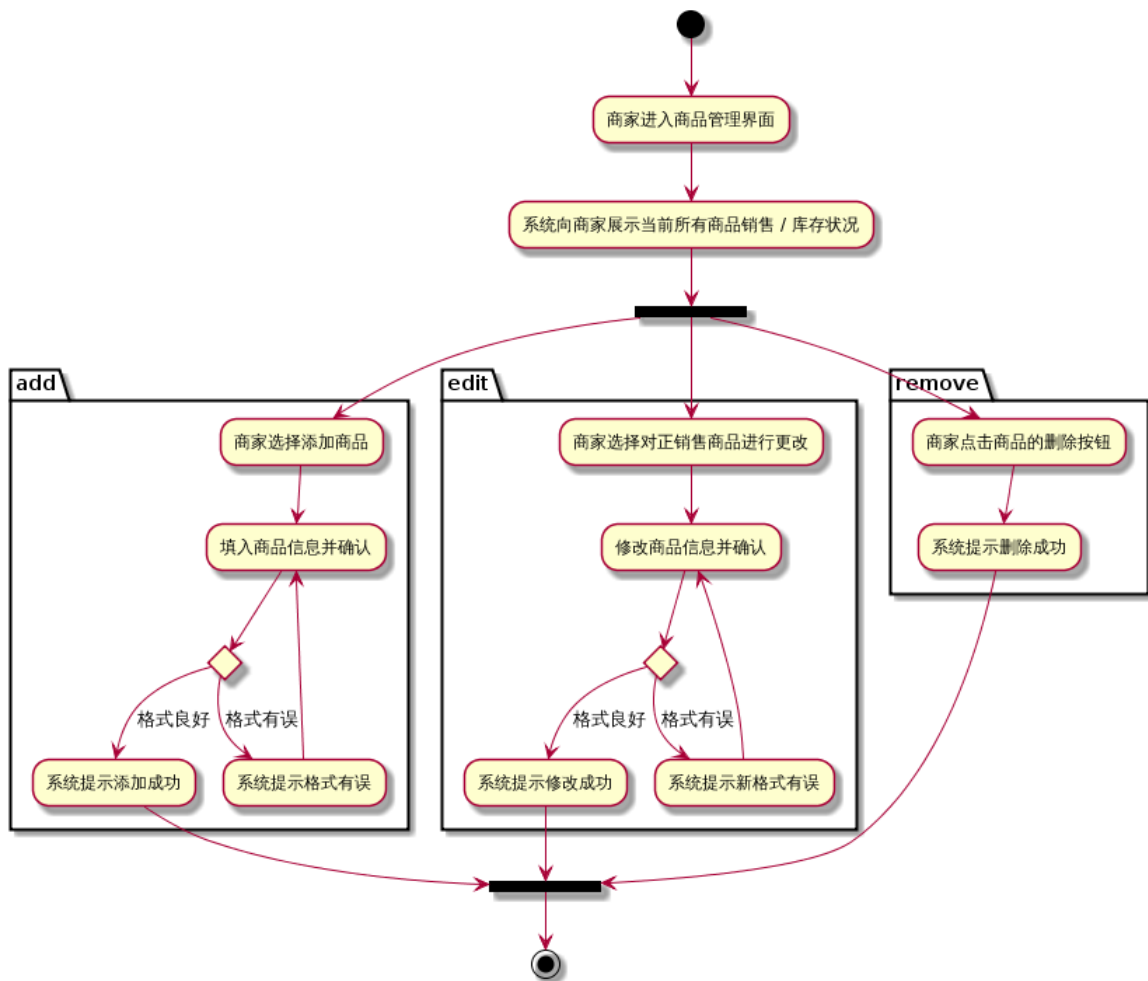
用户管理订单用例活动图如下：



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：



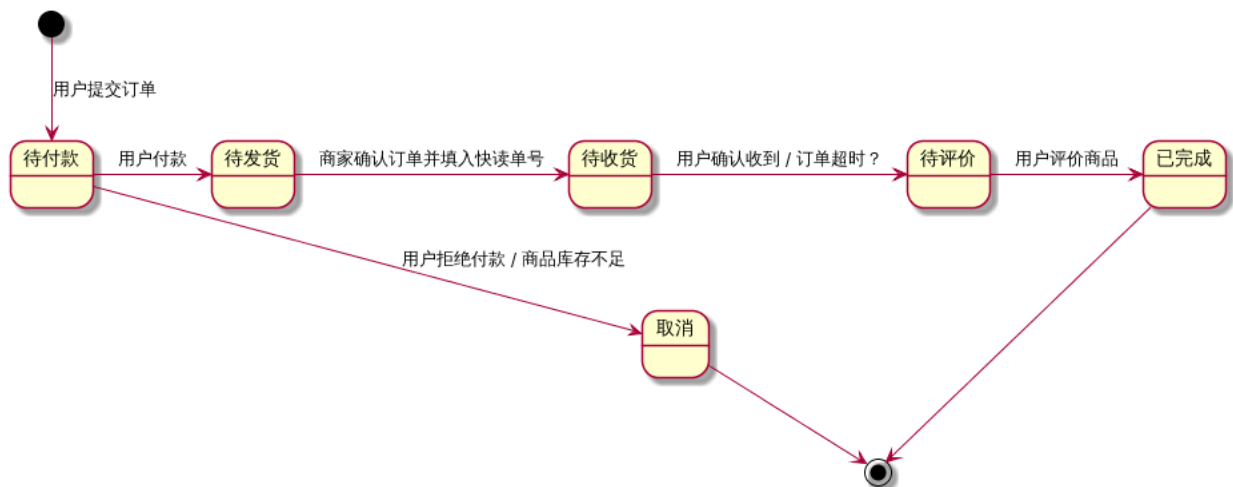
查询订单

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：



4.2 全局数据结构说明

4.2.1 常量

常量用于存储系统中不变的数据，如文件路径、默认值和系统参数等。以下表格详细描述了系统中的常量。

常量表

常量名称	数据类型	功能说明	所在目录
DB_CONNECTION_STRING	String	数据库连接字符串，用于连接系统数据库	config/database.js
MAX_LOGIN_ATTEMPTS	int	用户最大登录尝试次数，超过该次数将锁定账户	config/security.js
SESSION_TIMEOUT	int	用户会话超时时间（以分钟为单位），用户超过该时间需重新登录	config/security.js
PRODUCT_IMAGE_PATH	String	商品图片存储路径，用户上传的商品图片将存储在该目录下	config/app.js
SUPPORT_EMAIL	String	系统支持邮件地址，用于用户反馈和系统通知	config/app.js
DEFAULT_CURRENCY	String	系统默认货币类型	config/app.js
PAYMENT_GATEWAY_URL	String	支付网关接口地址，用于支付处理模块调用支付服务	config/payment.js
ORDER_STATUS_PENDING	String	订单状态：待处理	constants/orderStatus.js
ORDER_STATUS_COMPLETED	String	订单状态：已完成	constants/orderStatus.js
ORDER_STATUS_CANCELLED	String	订单状态：已取消	constants/orderStatus.js

服务器和服务的相关配置信息：

配置名称	配置内容	功能说明	数据类型
coreServer	{ 'url': 'http://:5432', 'flask-ip': "127.0.0.1", 'port': 5000 }	核心服务器相关配置信息	dict
mysql	{ 'host': "host", 'port': 3306, 'user': "ets", 'password': "xnx021128", 'database': "book" }	中心数据库地址	dict

4.2.2 变量

变量用于存储系统运行过程中动态变化的数据，包括用户输入、系统配置和临时数据等。以下表格详细描述了系统中的变量。

变量表

变量名称	数据类型	功能说明	所在目录
currentUser	Object	当前登录用户对象，包含用户的基本信息和权限	context/UserContext.js
shoppingCart	Object	当前用户的购物车对象，包含购物车中商品的列表	context/CartContext.js
productList	Array	当前查询结果中的商品列表	context/ProductContext.js
orderDetails	Object	当前用户正在处理的订单对象	context/OrderContext.js
searchCriteria	Object	用户商品搜索的条件	context/SearchContext.js
paymentInfo	Object	当前订单的支付信息，包括支付方式和支付状态	context/PaymentContext.js
feedbackList	Array	当前商品的用户评价列表	context/FeedbackContext.js
authToken	String	用户认证令牌，用于验证用户身份	context/AuthContext.js
connection	Object	数据库连接对象，用于数据库操作	database/connection.js
logEntries	Array	系统运行日志列表，记录系统运行过程中的重要事件	context/LogContext.js

4.2.3 数据结构

数据库建表

```
CREATE TABLE `comment` (  
  `menu_Menu_id` int NOT NULL,  
  `menu_restaurant_Restaurant_id` int NOT NULL,  
  `customer_Customer_id` varchar(25) NOT NULL,  
  `comment` varchar(100) DEFAULT NULL,  
  PRIMARY KEY  
  (`menu_Menu_id`,`menu_restaurant_Restaurant_id`,`customer_Customer_id`),  
  KEY `fk_menu_has_customer_customer1_idx` (`customer_Customer_id`),  
  KEY `fk_menu_has_customer_menu1_idx`  
  (`menu_Menu_id`,`menu_restaurant_Restaurant_id`),  
  CONSTRAINT `fk_menu_has_customer_customer1` FOREIGN KEY (`customer_Customer_id`)  
  REFERENCES `customer` (`Customer_id`),  
  CONSTRAINT `fk_menu_has_customer_menu1` FOREIGN KEY (`menu_Menu_id`,  
  `menu_restaurant_Restaurant_id`) REFERENCES `menu` (`Menu_id`,  
  `restaurant_Restaurant_id`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3
```

```
CREATE TABLE `customer` (  
  `Customer_id` varchar(25) NOT NULL,  
  `Name` varchar(25) NOT NULL,  
  `Address` varchar(45) NOT NULL,  
  `Phone_num` char(11) NOT NULL,  
  `password` varchar(20) NOT NULL,  
  PRIMARY KEY (`Customer_id`),  
  KEY `customerindex` (`Customer_id`,`password`) /*!80000 INVISIBLE */  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3
```

```
CREATE TABLE `delivery_guy` (  
  `Delivery_guy_id` int NOT NULL,  
  `Name` varchar(45) NOT NULL,  
  `Phone_num` char(11) NOT NULL,  
  PRIMARY KEY (`Delivery_guy_id`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3
```

```

CREATE TABLE `menu` (
  `Menu_id` int NOT NULL AUTO_INCREMENT,
  `Name` varchar(45) NOT NULL,
  `Price` varchar(45) NOT NULL,
  `Image` varchar(100) NOT NULL,
  `Likes` int NOT NULL,
  `restaurant_Restaurant_id` int NOT NULL,
  PRIMARY KEY (`Menu_id`,`restaurant_Restaurant_id`),
  KEY `fk_menu_restaurant1_idx` (`restaurant_Restaurant_id`) /*!80000 INVISIBLE */,
  KEY `nameindex` (`Name`),
  CONSTRAINT `fk_menu_restaurant1` FOREIGN KEY (`restaurant_Restaurant_id`)
REFERENCES `restaurant` (`Restaurant_id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb3

```

```

CREATE TABLE `order` (
  `Order_id` int NOT NULL AUTO_INCREMENT,
  `Time` datetime NOT NULL,
  `Total_price` float NOT NULL,
  `Cur_status` tinyint DEFAULT NULL,
  `customer_Customer_id` varchar(25) NOT NULL,
  `delivery_guy_Delivery_guy_id` int NOT NULL,
  PRIMARY KEY (`Order_id`),
  KEY `fk_order_customer_idx` (`customer_Customer_id`),
  KEY `fk_order_delivery_guy1_idx` (`delivery_guy_Delivery_guy_id`),
  CONSTRAINT `fk_order_customer` FOREIGN KEY (`customer_Customer_id`) REFERENCES
`customer` (`Customer_id`),
  CONSTRAINT `fk_order_delivery_guy1` FOREIGN KEY (`delivery_guy_Delivery_guy_id`)
REFERENCES `delivery_guy` (`Delivery_guy_id`)
) ENGINE=InnoDB AUTO_INCREMENT=99 DEFAULT CHARSET=utf8mb3

```

```

CREATE TABLE `order_detail` (
  `menu_Menu_id` int NOT NULL,
  `menu_restaurant_Restaurant_id` int NOT NULL,
  `order_Order_id` int NOT NULL,
  `amount` int DEFAULT NULL,
  PRIMARY KEY (`menu_Menu_id`,`menu_restaurant_Restaurant_id`,`order_Order_id`),
  KEY `fk_menu_has_order_order1_idx` (`order_Order_id`),
  KEY `fk_menu_has_order_menu1_idx`
(`menu_Menu_id`,`menu_restaurant_Restaurant_id`),
  CONSTRAINT `fk_menu_has_order_menu1` FOREIGN KEY (`menu_Menu_id`,
`menu_restaurant_Restaurant_id`) REFERENCES `menu` (`Menu_id`,
`restaurant_Restaurant_id`),
  CONSTRAINT `fk_menu_has_order_order1` FOREIGN KEY (`order_Order_id`) REFERENCES
`order` (`Order_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3

```

```
CREATE TABLE `restaurant` (
  `Restaurant_id` int NOT NULL,
  `Resname` varchar(45) NOT NULL,
  `Address` varchar(45) NOT NULL,
  `Phone_num` char(11) NOT NULL,
  PRIMARY KEY (`Restaurant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3
```

数据字典

	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	
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	
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									

4.3 CSCI部件

4.3.1 软件配置项标识

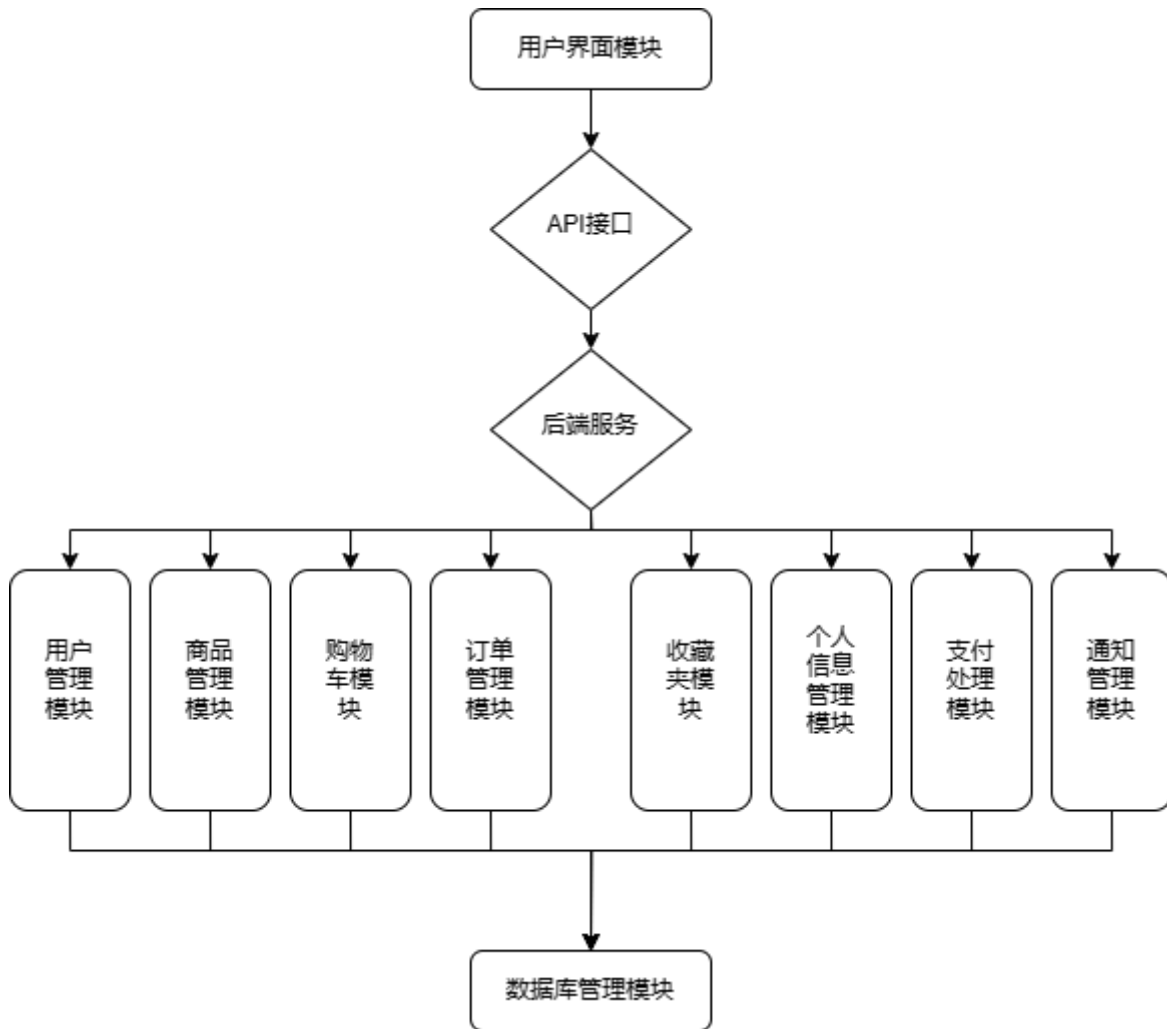
本系统由以下软件配置项（Software Configuration Items, SCIs）构成，每个配置项都有唯一标识符：

1. 用户管理模块（UMM-001）
2. 商品管理模块（PMM-002）
3. 购物车模块（SCM-003）
4. 订单管理模块（OMM-004）
5. 收藏夹模块（FMM-005）
6. 个人信息管理模块（PIMM-006）
7. 数据库管理模块（DBMM-007）

- 8. 用户界面模块 (UIM-008)
- 9. 支付处理模块 (PPM-009)
- 10. 通知管理模块 (NMM-010)

4.3.2 软件配置项的静态关系

- **组成关系：**各个模块之间的组成关系如下：
 - 用户界面模块 (UIM-008) 作为前端，与所有其他后端模块 (UMM-001, PMM-002, SCM-003, OMM-004, FMM-005, PIMM-006, PPM-009, NMM-010) 通过API接口进行交互。
 - 数据库管理模块 (DBMM-007) 作为后端数据支持，与所有功能模块进行数据交互。
- **关系图：**



4.3.3 软件配置项的用途和需求分配

1. 用户管理模块 (UMM-001)
 - **用途：**负责用户的注册、登录和权限管理。
 - **分配需求：**用户身份验证，用户权限分配。
 - **设计决策：**采用JWT进行用户身份验证。
2. 商品管理模块 (PMM-002)

- **用途：**管理商品的添加、删除、修改及查询。
- **分配需求：**商品信息的CRUD操作。
- **设计决策：**采用RESTful API进行商品操作。

3. 购物车模块 (SCM-003)

- **用途：**管理用户的购物车，包括商品的添加、删除和数量调整。
- **分配需求：**购物车项管理。
- **设计决策：**购物车信息存储在数据库中，采用缓存机制优化性能。

4. 订单管理模块 (OMM-004)

- **用途：**处理用户订单的创建、支付和状态管理。
- **分配需求：**订单创建、订单状态管理、订单查询。
- **设计决策：**订单信息与支付处理模块对接。

5. 收藏夹模块 (FMM-005)

- **用途：**管理用户的收藏夹，包括商品的收藏和取消收藏。
- **分配需求：**收藏夹项管理。
- **设计决策：**收藏夹信息存储在数据库中。

6. 个人信息管理模块 (PIMM-006)

- **用途：**管理用户的个人信息，包括查看和更新。
- **分配需求：**用户信息的CRUD操作。
- **设计决策：**采用加密技术保护用户敏感信息。

7. 数据库管理模块 (DBMM-007)

- **用途：**负责所有数据的存储、查询和管理。
- **分配需求：**数据存储、数据备份和恢复。
- **设计决策：**采用关系型数据库（如MySQL）存储数据。

8. 用户界面模块 (UIM-008)

- **用途：**提供系统的用户交互界面。
- **分配需求：**用户界面设计，用户体验优化。
- **设计决策：**采用响应式设计，支持多设备访问。

9. 支付处理模块 (PPM-009)

- **用途：**处理订单的支付事务。
- **分配需求：**支付接口集成，支付状态管理。
- **设计决策：**集成第三方支付服务（如支付宝、微信支付）。

10. 通知管理模块 (NMM-010)

- **用途：**处理系统的通知和消息推送。
- **分配需求：**消息通知，推送服务。
- **设计决策：**采用消息队列服务（如RabbitMQ）实现异步通知。

4.4执行概念

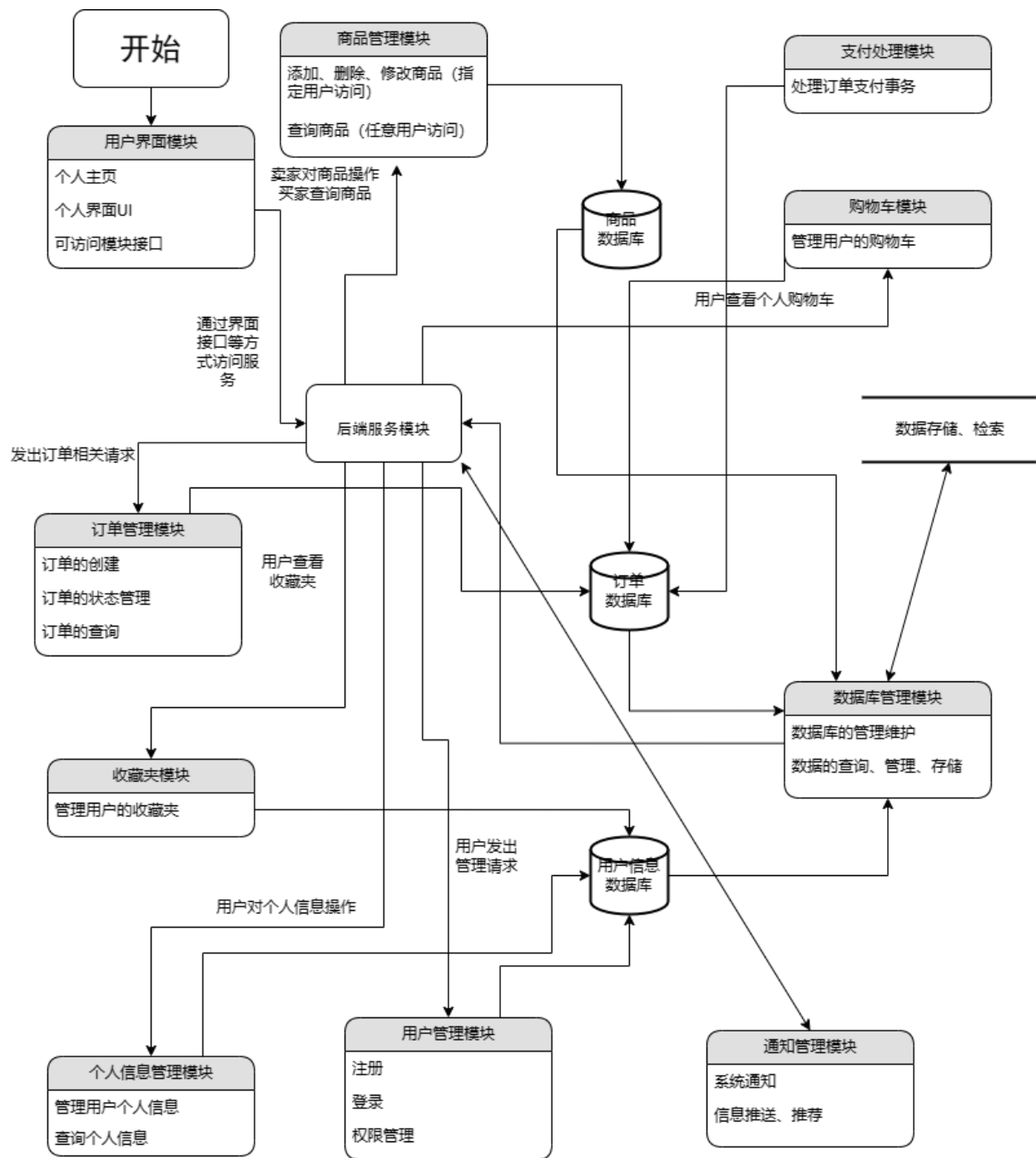
本条应描述软件配置项间的执行概念。为表示软件配置项之间的动态关系，即CSCI运行期间它们如何交互的，本条应包含图示和说明，(若适用)包括执行控制流、数据流、动态控制序列、状态转换图、时序图、配置项之间的优先关系、中断处理、时间/序列关系、异常处理、并发执行、动态分配与去分配、对象/进程/任务的动态创建与删除和其他的动态行为。

4.4.1 动态关系概述

系统运行时，各软件配置项的主要动态关系包括：

- 用户通过用户界面模块（UIM-008）发起请求。
- 用户界面模块将请求转发给相应的后端模块（如UMM-001、PMM-002等）。
- 后端模块处理请求，并与数据库管理模块（DBMM-007）进行数据交互。
- 数据库管理模块将结果返回给相应的后端模块。
- 后端模块将处理结果通过用户界面模块反馈给用户。

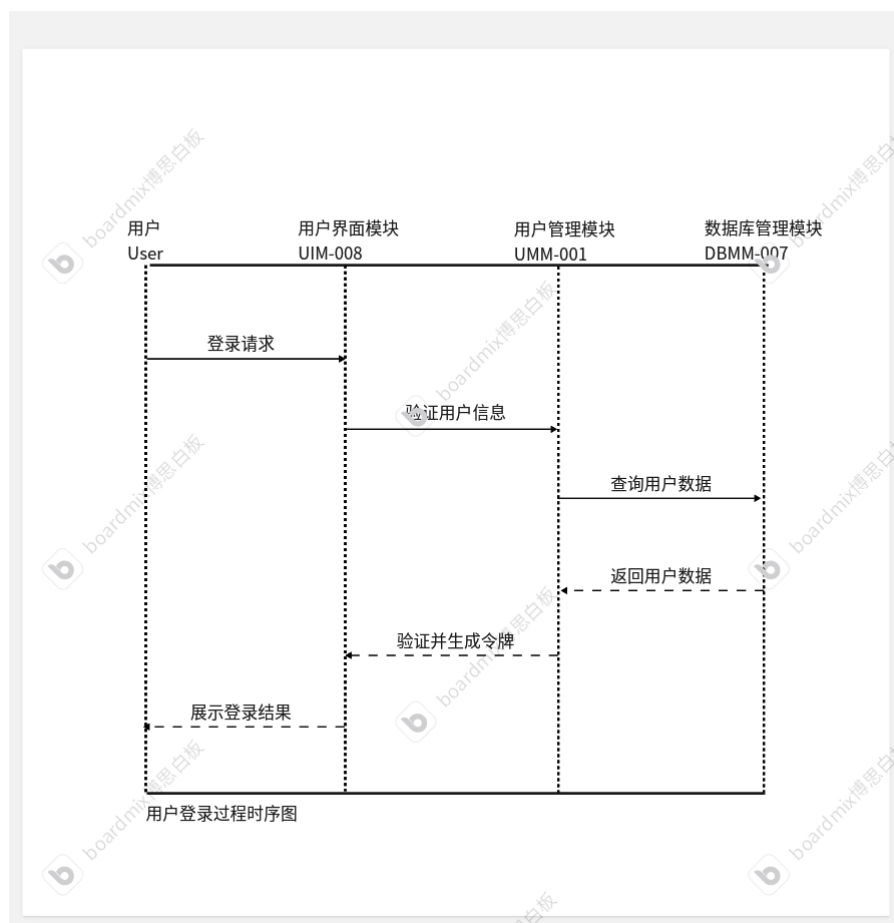
4.4.2 控制流和数据流



4.4.3 动态控制序列

以下是用户登录过程的动态控制序列图：

1. 用户发起登录请求
2. 用户界面模块 (UIM-008) 接收到登录请求，并将其转发至用户管理模块 (UMM-001)。
3. 用户管理模块 (UMM-001) 验证用户信息，并向数据库管理模块 (DBMM-007) 查询用户数据。
4. 数据库管理模块 (DBMM-007) 返回查询结果给用户管理模块 (UMM-001)。
5. 用户管理模块 (UMM-001) 验证用户信息并生成认证令牌，将结果返回给用户界面模块 (UIM-008)。
6. 用户界面模块 (UIM-008) 向用户展示登录结果



4.4.4 状态转换过程

1. **未登录**：用户未进行登录操作。
2. **登录中**：用户提交登录信息，系统正在验证。
3. **已登录**：用户信息验证通过，用户成功登录。
4. **登录失败**：用户信息验证失败，用户登录失败。

4.4.5 配置项之间的优先关系

- **用户界面模块（UIM-008）** 对所有用户操作具有最高优先级。
- **后端服务模块（如UMM-001、PMM-002等）** 对具体业务逻辑处理具有次高优先级。
- **数据库管理模块（DBMM-007）** 对数据存储和检索具有次优先级。

4.4.6 并发执行和动态分配

- **并发执行**：用户界面模块（UIM-008）支持多个用户的并发操作。后端服务模块（UMM-001、PMM-002等）采用多线程技术处理并发请求。数据库管理模块（DBMM-007）支持并发读写操作。
- **动态分配与去分配**：系统根据请求的实时情况动态分配和释放资源。用户登录时动态创建用户会话，退出时删除会话。

4.4.7 异常处理

- **用户界面模块 (UIM-008)**：显示友好的错误提示信息。
- **后端服务模块 (UMM-001、PMM-002等)**：捕获并记录异常，返回统一的错误响应。
- **数据库管理模块 (DBMM-007)**：处理数据库连接异常，提供自动重连机制。

4.4.8 中断处理

系统采用异步处理机制，确保在高并发情况下能够有效处理用户请求。对于时间敏感的操作（如支付处理），系统设置了中断优先级，确保关键任务优先执行。

4.4.9 时间/序列关系

- **用户登录**：从用户输入到验证成功不超过1秒。
- **商品查询**：从用户输入查询条件到返回结果不超过2秒。
- **订单处理**：从用户提交订单到生成订单确认不超过3秒。

4.4.10 对象/进程/任务的动态创建与删除

- **用户会话**：登录时创建会话，退出时删除会话。
- **订单**：用户提交订单时动态创建订单对象，订单完成或取消后删除对象。
- **通知**：生成通知任务，发送完毕后删除任务。

4.5接口设计

本条应分条描述软件配置项的接口特性，既包括软件配置项之间的接口，也包括与外部实体，如系统、配置项及用户之间的接口。如果这些信息的部分或全部已在接口设计说明(IDD)、本文的第5章或其他地方说明的话，可在此处引用。

功能模块	接口URL	接口用途
支付模块	/order/create	创建一个订单
	/payment/initial	请求支付
	/payment/confirm	确认支付
商品模块	/goods/history	获取搜索历史
	/goods/hot	热搜榜单
	/goods/newgoods	获取商品榜单
	/goods/start	商品评分
	/goods/like_info	获取用户收藏
	/goods/buy_info	获取购买信息
	/goods/buy	购买商品
	/goods/put	商品放入购物车
	/goods/like	收藏商品
	/goods/search	按关键字搜索商品
	/goods/info	获取商品信息
	/goods/share	将商品分享
	/goods/category	根据分类搜索商品
管理员模块	/goods/remove	下架商品
	/goods/insert	上架商品
	/goods/changeinfo	修改商品信息
	/goods/changeprice	修改商品价格
	/goods/new	添加新商品信息
推荐模块	/recommend/user	对一位用户的推荐（用户个性化推荐）
	/recommend/goods	对商品的推荐（相关商品推荐）
用户模块	/user/change	修改用户信息
	/profile/id	获取用户信息
	/user/forget	忘记密码
	/signup	用户注册

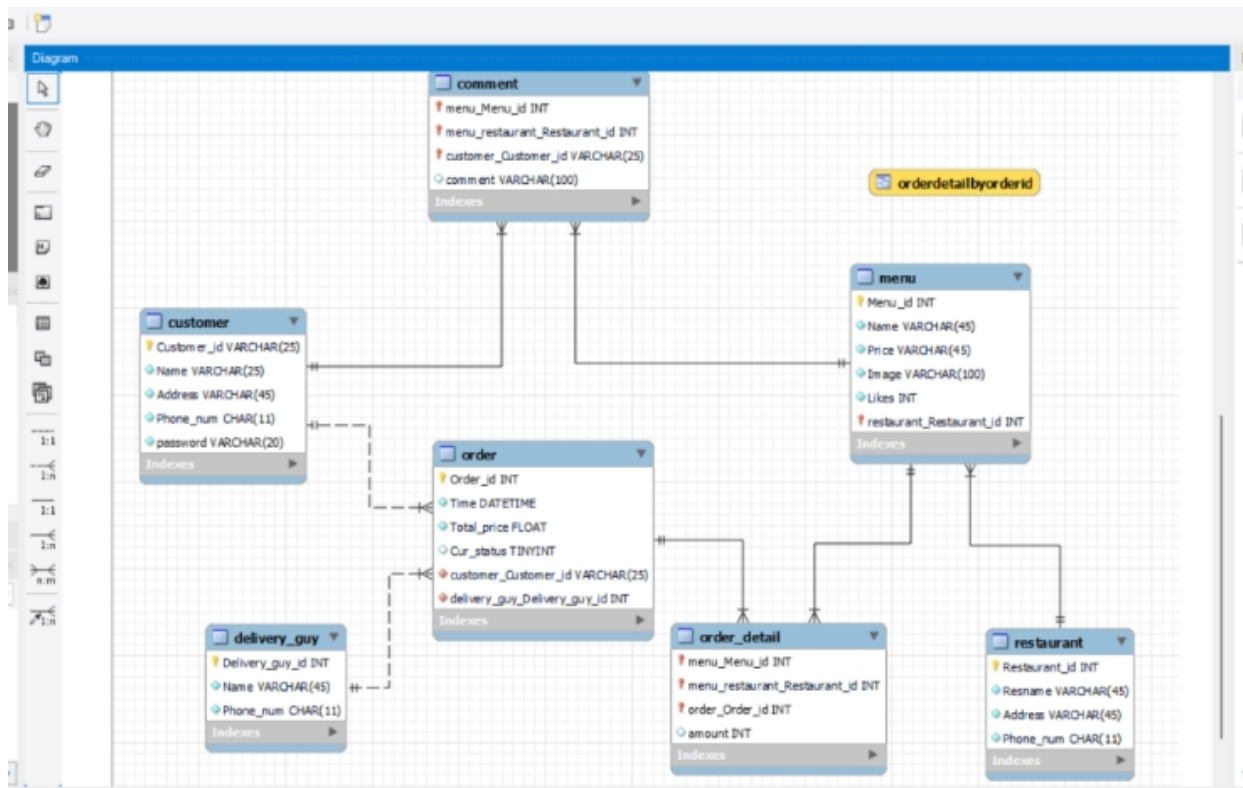
功能模块	接口URL	接口用途
	/user/verify	短信验证码
	/signin	用户登录

4.5.1接口标识与接口图

接口标识与特性表

接口标识符	接口名称	优先级	接口类型	数据元素特性
UMMI-001	用户管理模块接口	高	数据存储与检索	提供用户注册、登录、权限管理等功能
PMMI-002	商品管理模块接口	高	数据存储与检索	提供商品的添加、删除、修改和查询功能
SCMI-003	购物车模块接口	中	数据存储与检索	提供用户购物车的管理功能，包括商品的添加、删除和数量修改
OMMI-004	订单管理模块接口	高	数据存储与检索	提供订单的创建、修改、删除和查询功能
FMMI-005	收藏夹模块接口	中	数据存储与检索	提供用户收藏商品的管理功能
PIMMI-006	个人信息管理模块接口	中	数据存储与检索	提供用户个人信息的查看和修改功能
DBMMI-007	数据库管理模块接口	高	数据存储与检索	提供系统所有数据的存储和检索功能
UIMI-008	用户界面模块接口	高	用户界面	提供系统的前端展示和交互功能
PPMI-009	支付处理模块接口	高	实时数据传输	提供订单支付功能，处理支付请求和响应
NMMI-010	通知管理模块接口	中	数据存储与检索	提供系统通知的管理功能，包括通知的发送和接收

E-R图



5 CSCI详细设计

5.1 用户管理模块（UMM-001）

5.1.1 项目唯一标识符

UMM-001

5.1.2 配置项设计决策

- 使用JWT（JSON Web Tokens）进行用户身份验证，以实现无状态的认证机制。
- 实现用户密码加密存储，采用哈希算法（如bcrypt）进行密码加密存储，以增强安全性。
- 采用分布式数据库架构，以支持用户信息的水平扩展和高可用性。

5.1.3 设计约束和特征

- 采用Node.js作为开发语言，以实现快速的异步处理和高并发性能。
- 使用Express.js框架来构建RESTful API，以实现简洁的路由和中间件功能。

5.1.4 输入和输出数据元素

- 输入数据元素：用户登录信息（用户名和密码）、用户注册信息（用户名、密码、电子邮件等）。
- 输出数据元素：用户认证令牌、用户信息（用户名、电子邮件、权限等）。

5.1.5 逻辑

- **启动条件：**当收到用户登录或注册请求时，用户管理模块开始工作。
- **控制交给其他配置项的条件：**当用户登录请求验证通过时，控制交给用户界面模块返回认证令牌。
- **输入的响应：**
 - 用户登录请求：验证用户信息，如果验证通过，生成认证令牌并返回给用户界面模块；否则返回错误信息。
 - 用户注册请求：将用户信息存储到数据库中，并返回注册成功信息。
- **操作序列：**
 1. 接收用户登录或注册请求。
 2. 验证用户信息。
 3. 生成认证令牌（登录请求）或存储用户信息（注册请求）。
 4. 返回认证令牌或注册成功信息给用户界面模块。
- **异常与错误处理：**
 - 如果用户提供的登录信息不正确，返回错误信息提示用户名或密码错误。
 - 如果用户注册时用户名已存在，返回错误信息提示用户名已被使用。

5.2 商品管理模块（PMM-002）

5.2.1 项目唯一标识符

PMM-002

5.2.2 配置项设计决策

- 使用RESTful API设计，以实现统一的接口规范和易于扩展的架构。
- 实现商品信息的CRUD（创建、读取、更新、删除）功能，以满足用户对商品的管理需求。
- 采用异步消息队列技术，处理商品的批量上传和异步处理任务，提高系统的并发处理能力。

5.2.3 设计约束和特征

- 使用Python语言编写，以实现快速的开发和易于维护的代码。
- 基于Django框架搭建，利用Django ORM实现与数据库的交互，简化数据库操作。
- 依赖数据库模型设计，需要与数据库管理模块（DBMM-007）紧密配合。

5.2.4 输入和输出数据元素

- 输入数据元素：商品信息（名称、描述、价格等）。
- 输出数据元素：商品信息（名称、描述、价格等）。

5.2.5 逻辑

- **启动条件：**收到商品管理请求时，商品管理模块开始工作。
- **控制交给其他配置项的条件：**处理完商品管理请求后，控制交给用户界面模块返回处理结果。
- **输入的响应：**
 - 创建商品请求：接收商品信息并保存到数据库中，返回创建成功信息。
 - 读取商品请求：根据商品ID或查询条件从数据库中读取商品信息，返回查询结果。
 - 更新商品请求：接收更新后的商品信息，更新数据库中对应的商品记录，返回更新成功信息。
 - 删除商品请求：根据商品ID删除数据库中对应的商品记录，返回删除成功信息。
- **操作序列：**
 1. 接收商品管理请求。
 2. 根据请求类型执行相应的操作（创建、读取、更新、删除）。
 3. 将操作结果返回给用户界面模块。
- **异常与错误处理：**
 - 如果商品信息不完整或格式不正确，返回错误信息提示商品信息错误。
 - 如果商品不存在或已被删除，返回错误信息提示商品不存在。

5.3 购物车模块（SCM-003）

5.3.1 项目唯一标识符

SCM-003

5.3.2 配置项设计决策

- 使用数据库存储购物车信息，以实现购物车数据的持久化存储。
- 提供用户添加、删除、更新购物车商品数量的功能，以满足用户的购物需求。
- 实现购物车数据的缓存机制，提高购物车操作的响应速度。

5.3.3 设计约束和特征

- 使用Java语言编写，以实现跨平台的可移植性。
- 基于Spring框架开发，利用Spring Data实现与数据库的交互，简化数据访问层代码。
- 依赖于用户管理模块（UMM-001）和商品管理模块（PMM-002）提供的服务。

5.3.4 输入和输出数据元素

- 输入数据元素：用户ID、商品ID、购物车操作类型（添加、删除、更新）等。
- 输出数据元素：购物车商品信息、购物车操作结果。

5.3.5 逻辑

- **启动条件：**收到购物车管理请求时，购物车模块开始工作。
- **控制交给其他配置项的条件：**处理完购物车管理请求后，控制交给用户界面模块返回处理结果。
- **输入的响应：**
 - 添加商品到购物车请求：根据用户ID和商品ID将商品添加到购物车，并返回添加成功信息。
 - 删除购物车中的商品请求：根据用户ID和商品ID从购物车中删除商品，并返回删除成功信息。
 - 更新购物车中商品数量请求：根据用户ID和商品ID更新购物车中对应商品的数量，并返回更新成功信息。
- **操作序列：**
 1. 接收购物车管理请求。
 2. 根据请求类型执行相应的操作（添加、删除、更新）

5.4 订单管理模块（OMM-004）

5.4.1 项目唯一标识符

OMM-004

5.4.2 配置项设计决策

- 实现订单创建、支付和状态管理功能。
- 与支付处理模块对接，确保订单支付流程的顺利进行。
- 使用状态机模式管理订单状态，实现订单状态的可控性和可扩展性。

5.4.3 设计约束和特征

- 订单信息存储在数据库中，以便于持久化存储和跟踪订单状态。
- 提供订单查询功能，以使用户查询订单状态和历史订单信息。

5.4.4 输入和输出数据元素

- 输入数据元素：用户ID、商品信息、支付信息。
- 输出数据元素：订单号、订单状态、支付状态。

5.4.5 逻辑

- **启动条件：**接收到用户的订单创建请求。
- **控制交给其他配置项的条件：**在订单创建后，将控制交给支付处理模块进行支付处理。
- **输入的响应：**
 - 订单创建请求：接收用户的订单信息并创建订单，返回订单号。
 - 订单支付请求：接收用户的支付信息并处理支付，返回支付状态。
- **操作序列：**
 1. 接收订单创建请求。

2. 创建订单并保存到数据库中。
 3. 将订单状态设置为待支付。
 4. 将订单号返回给用户，并等待支付处理。
 5. 收到支付结果后更新订单状态。
- 异常与错误处理：
 - 如果订单创建失败或支付失败，返回错误信息提示用户订单创建或支付失败。
 - 如果用户查询订单状态时发生错误，返回错误信息提示用户查询失败。

5.5 收藏夹模块 (FMM-005)

5.5.1 项目唯一标识符

FMM-005

5.5.2 配置项设计决策

- 实现管理用户的收藏夹功能，包括商品的收藏和取消收藏。
- 提供收藏夹项的管理功能，包括添加商品到收藏夹和从收藏夹中移除商品。

5.5.3 设计约束和特征

- 收藏夹信息存储在数据库中，以便于持久化存储和跨设备同步。
- 用户可以通过用户界面进行收藏夹的管理操作。

5.5.4 输入和输出数据元素

- 输入数据元素：用户ID、商品信息。
- 输出数据元素：收藏夹列表、收藏夹项操作结果。

5.5.5 逻辑

- 启动条件：接收到用户的收藏夹管理请求。
- 控制交给其他配置项的条件：在收到用户的收藏夹管理请求后，将控制交给数据库管理模块进行数据操作。
- 输入的响应：
 - 添加商品到收藏夹请求：接收到用户的收藏请求并将商品添加到收藏夹中。
 - 从收藏夹中移除商品请求：接收到用户的取消收藏请求并从收藏夹中移除商品。
- 操作序列：
 1. 接收收藏夹管理请求。
 2. 根据请求类型执行相应的操作（添加、移除）。
 3. 更新数据库中的收藏夹信息。
 4. 返回操作结果给用户界面模块。
- 异常与错误处理：
 - 如果添加商品到收藏夹失败或移除商品失败，返回错误信息提示用户操作失败。

- 如果用户查询收藏夹信息时发生错误，返回错误信息提示用户查询失败。

5.6 个人信息管理模块 (PIMM-006)

5.6.1 项目唯一标识符

PIMM-006

5.6.2 配置项设计决策

- 实现管理用户的个人信息功能，包括查看和更新用户的个人信息。
- 提供用户信息的CRUD（创建、读取、更新、删除）操作，以满足用户对个人信息的管理需求。

5.6.3 设计约束和特征

- 采用加密技术保护用户敏感信息，确保用户数据的安全性和隐私性。
- 用户可以通过用户界面进行个人信息的查看和更新操作。

5.6.4 输入和输出数据元素

- 输入数据元素：用户ID、个人信息更新内容。
- 输出数据元素：用户个人信息、个人信息更新结果。

5.6.5 逻辑

- 启动条件：接收到用户的个人信息管理请求。
- 控制交给其他配置项的条件：在接收到用户的个人信息管理请求后，将控制交给数据库管理模块进行数据操作。
- 输入的响应：
 - 查看个人信息请求：从数据库中读取用户个人信息并返回给用户。
 - 更新个人信息请求：接收用户提供的个人信息更新内容并更新数据库中的用户信息。
- 操作序列：
 1. 接收个人信息管理请求。
 2. 根据请求类型执行相应的操作（查看、更新）。
 3. 更新数据库中的用户信息。
 4. 返回操作结果给用户界面模块。
- 异常与错误处理：
 - 如果更新个人信息失败，返回错误信息提示用户操作失败。
 - 如果用户查询个人信息时发生错误，返回错误信息提示用户查询失败。

5.7 数据库管理模块 (DBMM-007)

5.7.1 项目唯一标识符

DBMM-007

5.7.2 配置项设计决策

- 负责管理系统的数据库，包括数据库的设计、创建、维护和备份。
- 使用分布式数据库架构，以支持系统的水平扩展和高可用性。
- 选择合适的数据库管理系统（如MySQL、PostgreSQL等）来存储系统数据。

5.7.3 设计约束和特征

- 实现数据库的数据模型设计，确保数据库结构的合理性和灵活性。
- 提供数据库的备份和恢复机制，以保障系统数据的安全性和完整性。

5.7.4 输入和输出数据元素

- 输入数据元素：数据库设计需求、数据库备份策略。
- 输出数据元素：数据库设计文档、数据库备份文件。

5.7.5 逻辑

- 启动条件：系统初始化或数据库管理操作触发。
- 控制交给其他配置项的条件：在收到数据库管理请求后，将控制交给数据库管理模块进行数据库操作。
- 输入的响应：
 - 数据库设计请求：根据系统需求进行数据库设计，并生成数据库设计文档。
 - 数据库备份请求：执行数据库备份操作，并生成备份文件。
- 操作序列：
 1. 接收数据库管理请求。
 2. 根据请求类型执行相应的操作（设计、备份）。
 3. 更新数据库或生成备份文件。
 4. 返回操作结果给请求方。

5.8 用户界面模块（UIM-008）

5.8.1 项目唯一标识符

UIM-008

5.8.2 配置项设计决策

- 负责系统的用户界面设计和实现，提供用户友好的界面交互。
- 使用现代化的前端技术（如HTML、CSS、JavaScript框架）开发用户界面。
- 实现响应式设计，以适配不同设备上的用户访问。

5.8.3 设计约束和特征

- 设计简洁清晰的用户界面，以提高用户体验和易用性。
- 实现与后端系统的接口调用，以获取和展示系统数据。

5.8.4 输入和输出数据元素

- 输入数据元素：界面设计需求、系统数据。
- 输出数据元素：用户界面、用户交互数据。

5.8.5 逻辑

- 启动条件：系统初始化或用户界面访问触发。
- 控制交给其他配置项的条件：在接收到用户界面请求后，将控制交给用户界面模块进行界面展示。
- 输入的响应：
 - 用户界面访问请求：根据用户请求展示相应的界面。
 - 用户交互请求：接收用户交互操作并处理相应的业务逻辑。
- 操作序列：
 1. 接收用户界面请求。
 2. 根据请求类型展示相应的界面或处理用户交互操作。
 3. 与后端系统进行数据交互。
 4. 返回操作结果给用户或请求方。

5.9 支付处理模块（PPM-009）

5.9.1 项目唯一标识符

PPM-009

5.9.2 配置项设计决策

- 负责处理系统中的支付业务，包括接收用户支付请求、处理支付流程和更新订单状态。
- 与第三方支付服务（如支付宝、微信支付）对接，实现支付功能。

5.9.3 设计约束和特征

- 实现安全可靠的支付处理流程，确保用户支付操作的安全性和可靠性。
- 采用加密技术保护用户支付信息，防止信息泄露和篡改。

5.9.4 输入和输出数据元素

- 输入数据元素：用户支付请求、订单信息。
- 输出数据元素：支付结果、订单状态更新结果。

5.9.5 逻辑

- 启动条件：接收到用户的支付请求。
- 控制交给其他配置项的条件：

5.10 通知管理模块（NMM-010）

5.10.1 项目唯一标识符

NMM-010

5.10.2 配置项设计决策

- 负责系统中的通知管理，包括向用户发送通知和管理通知模板。
- 实现多种通知方式，如邮件、短信、站内信等。
- 支持用户自定义通知设置，以满足不同用户的需求。

5.10.3 设计约束和特征

- 设计通知模板，以便于灵活地定制通知内容和格式。
- 实现通知发送队列，以提高通知发送效率和稳定性。

5.10.4 输入和输出数据元素

- 输入数据元素：通知内容、接收用户信息。
- 输出数据元素：通知发送结果。

5.10.5 逻辑

- 启动条件：接收到系统生成通知的请求或用户操作触发。
- 控制交给其他配置项的条件：在收到通知发送请求后，将控制交给通知管理模块进行通知发送操作。
- 输入的响应：
 - 通知发送请求：接收到通知内容和接收用户信息后，将通知加入发送队列。
- 操作序列：
 1. 接收通知发送请求。
 2. 将通知加入发送队列。
 3. 从发送队列中取出通知，并根据通知方式进行发送。
 4. 更新通知发送状态。
- 异常与错误处理：
 - 如果通知发送失败，记录错误信息并进行重试或报警处理。
 - 如果通知模板设置错误，返回错误信息提示用户操作失败。

6 需求的可追踪性

a. 软件配置项到CSCI需求的可追踪性

- 订单管理模块（OMM-004）：
 - 分配需求: 订单创建、订单状态管理、订单查询。
- 收藏夹模块（FMM-005）：
 - 分配需求: 收藏夹项管理。
- 个人信息管理模块（PIMM-006）：
 - 分配需求: 用户信息的CRUD操作。
- 数据库管理模块（DBMM-007）：
 - 分配需求: 数据库设计、数据库备份。
- 用户界面模块（UIM-008）：
 - 分配需求: 用户界面设计和用户交互操作。
- 支付处理模块（PPM-009）：
 - 分配需求: 支付业务处理和订单状态更新。
- 通知管理模块（NMM-010）：
 - 分配需求: 发送通知和管理通知模板。

b. CSCI需求到软件配置项的可追踪性

- 订单创建:
 - 分配给: 订单管理模块（OMM-004）。
- 订单状态管理:
 - 分配给: 订单管理模块（OMM-004）。
- 订单查询:
 - 分配给: 订单管理模块（OMM-004）。
- 收藏夹项管理:
 - 分配给: 收藏夹模块（FMM-005）。
- 用户信息的CRUD操作:
 - 分配给: 个人信息管理模块（PIMM-006）。
- 数据库设计:
 - 分配给: 数据库管理模块（DBMM-007）。
- 数据库备份:
 - 分配给: 数据库管理模块（DBMM-007）。
- 用户界面设计和用户交互操作:
 - 分配给: 用户界面模块（UIM-008）。
- 支付业务处理和订单状态更新:
 - 分配给: 支付处理模块（PPM-009）。
- 发送通知和管理通知模板:

- 分配给: 通知管理模块 (NMM-010) 。

7注解

由于开发时间限制，上述部分功能尚未实现或为完全实现。

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，以使用户查看最相关的评价。

商品管理

- **管理功能**：商家客户端暂未完成对商品的有效管理。

附录

无。