三、需求规格说明书

[1．引言 2](#_Toc506358922)

[1.1编写目的 2](#_Toc506358923)

[1.2项目背景 2](#_Toc506358924)

[1.3定义 2](#_Toc506358925)

[1.4参考资料 2](#_Toc506358926)

[2．任务概述 3](#_Toc506358927)

[2.1目标 3](#_Toc506358928)

[2.2运行环境 3](#_Toc506358929)

[2.3条件与限制 3](#_Toc506358930)

[3．数据描述 4](#_Toc506358931)

[3.1静态数据 4](#_Toc506358932)

[3.2动态数据 4](#_Toc506358933)

[3.3数据库介绍 4](#_Toc506358934)

[3.4数据词典 6](#_Toc506358935)

[3.5数据采集 8](#_Toc506358936)

[4．功能需求 9](#_Toc506358937)

[4.1功能划分 9](#_Toc506358938)

[4.2功能描述 11](#_Toc506358939)

[5．性能需求 13](#_Toc506358940)

[5.1数据精确度 13](#_Toc506358941)

[5.2时间特性 13](#_Toc506358942)

[5.3适应性 14](#_Toc506358943)

[6．运行需求 14](#_Toc506358944)

[6.1用户界面 14](#_Toc506358945)

[6.2硬件接口 14](#_Toc506358946)

[6.3软件接口 15](#_Toc506358947)

[6.4故障处理 15](#_Toc506358948)

[7．其它需求 15](#_Toc506358949)

# 1．引言

## 1.1编写目的

本需求规格说明书的目的是为了全面而清晰地描述软件产品商城的功能和需求，指导项目的各个阶段，包括需求分析、系统设计、开发和测试。本文档旨在为项目经理、开发人员、测试人员及其他相关利益方提供明确的参考，以确保各方对项目目标和实施细节的共同理解，并减少项目开发过程中的误解和偏差。

## 1.2项目背景

1. 项目的委托单位: 软件工程课程团队
2. 开发单位: 软件开发团队
3. 主管部门: 软件开发部门
4. 软件系统关系: 该商城将与现有的用户管理系统、支付网关及物流管理系统进行集成，以实现完整的用户购物体验，包括浏览、购买、支付及售后服务。

## 1.3定义

**商城**: 指在线软件产品销售平台，用户可以通过该平台浏览、购买及下载软件。

**用户**：指在商城上注册并进行交易的终端客户，包括普通消费者和企业用户。

**API**：应用程序接口，允许商城与其他系统（如支付系统、物流系统等）进行数据交互与功能集成。

**管理员**：负责商城运营、产品管理和用户服务的后台管理人员，确保商城正常运行。

**前端与后端**：前端指用户界面及交互部分，后端指数据处理、存储及业务逻辑实现部分。

## 1.4参考资料

1. **项目任务书**：项目经核准的计划任务书（编号：XYZ-2024-001），明确了项目的总体目标与实施方案。
2. **行业标准与规范**：相关的行业标准和软件开发规范（如ISO 25010）将作为项目实施的参考。
3. **市场调研报告**：针对目标市场的调研报告，分析了竞争对手、用户需求及市场趋势，为商城的功能规划提供了依据。
4. **技术文档**：包括与数据库、框架及开发工具相关的技术文档，指导开发团队在技术实施时的选择。

# 2．任务概述

## 2.1目标

开发一个功能完善、用户友好的软件产品商城，目标包括：

1. 提供多种软件产品的在线销售与下载服务，满足用户对不同软件需求的购买和使用体验。
2. 实现高效的用户管理功能，包括注册、登录、个人信息维护和密码重置。
3. 建立完善的订单管理系统，支持用户的在线支付、订单查询和历史订单管理。
4. 提供用户反馈和评价功能，让用户可以对所购买的软件进行评分和评论，提升社区互动。
5. 设计直观易用的用户界面，确保不同设备（PC、手机）上的良好体验。
6. 确保系统的安全性，保护用户信息及交易安全。

## 2.2运行环境

**操作系统**：使用Windows 11作为服务器和客户端操作系统。

**支持环境**：采用IIS 10.0作为Web服务器，确保高效的页面响应和数据处理能力。

**数据库**：采用MySQL作为数据库系统，进行用户数据、产品信息和订单管理的存储。客户**客户端支持**：确保系统在主流浏览器（Chrome、Firefox、Edge、Safari）中的兼容性，适配各类设备和屏幕尺寸。

## 2.3条件与限制

**法律限制**：系统必须遵循国家相关法律法规，包括软件著作权、隐私保护及电子商务法。

**预算限制**：整个项目的预算上限为50万元人民币，包括开发、测试及运营成本。

**时间限制**：项目计划在6个月内完成，从需求分析到最终上线。分阶段进行开发与测试，确保各个模块的功能完整与稳定。

**技术限制**：开发团队需具备MySQL、前端技术（如HTML、CSS、JavaScript）、前端框架(如Vue3)及后端开发语言（如Java）、后端开发框架(如SpringBoot)的技能，确保能够高效实施项目。

# 3．数据描述

## 3.1静态数据

· 系统应能够适应多种操作系统和浏览器环境，确保在主流浏览器（Chrome、Firefox、Safari、IE）和移动设备上均能良好运行。

· 若未来需要集成新的支付接口或支付方式，系统架构应支持灵活扩展，能够快速实现与新支付系统的对接。

· 系统应具备良好的模块化设计，以便在未来添加新功能或修改现有功能时，不影响其他模块的正常运行。

· 针对用户需求变化，系统应能快速响应并进行相应的功能调整，以保证用户体验的持续提升。

## 3.2动态数据

**输入数据**

输入数据是由用户通过前端界面或外部系统输入的数据，这些数据将被系统处理，并存储在数据库或用于触发某些功能。

1. **用户登录数据**：
   * 用户名/邮箱：用户在登录界面输入的登录凭证。
   * 密码：用户输入的密码，用于验证身份。
2. **注册信息**：
   * 用户名：新用户注册时输入的用户名。
   * 密码：用户设置的登录密码。
   * 联系方式：用户输入的邮箱或手机号码，用于账户验证。
3. **商品查询条件**：
   * 搜索关键词：用户在搜索框中输入的商品名称或关键词。
   * 分类筛选条件：用户选择的商品类别、价格范围、品牌等。
4. **购物车数据**：
   * 商品ID：用户选择添加到购物车的商品ID。
   * 商品数量：用户选择购买的商品数量。
5. **订单信息**：
   * 收货地址：用户填写的收货地址信息。
   * 支付方式：用户选择的支付方式（如信用卡、支付宝、微信等）。
   * 备注信息：用户在订单提交时输入的备注内容。
6. **支付信息**：
   * 支付金额：用户确认支付的订单总金额。
   * 支付凭证：通过支付网关返回的支付交易ID或凭证。
7. **反馈和评价**：
   * 用户评论：用户对商品的评价和评分。
   * 意见反馈：用户对平台或服务的反馈意见。

**输出数据**

输出数据是系统经过处理后生成并展示给用户的数据，这些数据可能是由用户操作产生，或者系统后台自动生成的数据。

1. **登录状态**：
   * 登录成功或失败的提示信息。
   * 用户身份认证后的欢迎信息或错误提示（如用户名或密码错误）。
2. **商品查询结果**：
   * 商品列表：根据用户输入的搜索条件，系统返回的符合条件的商品列表，包括商品名称、价格、库存等信息。
   * 商品详情：用户点击商品后，系统展示商品的详细信息，如商品描述、规格参数、用户评价等。
3. **购物车信息**：
   * 已添加商品列表：系统返回购物车中当前商品的详细信息，如商品名称、数量、单价、小计金额等。
   * 库存提醒：系统根据库存情况提示商品是否可购买。
4. **订单信息**：
   * 订单生成确认信息：订单成功提交后的确认信息，包含订单编号、商品明细、总金额、收货地址等。
   * 订单状态更新：系统在后台根据支付和物流情况，实时更新订单状态（如待发货、已发货、已签收等）。
5. **支付结果**：
   * 支付成功或失败信息：系统根据支付网关的反馈，返回支付成功或失败的结果提示。
   * 交易凭证：支付成功后的交易凭证或发票。
6. **下载信息**：
   * 下载记录：系统返回与订单相关的下载信息，包括下载时间、下载次数、商品ID、用户ID等。
7. **用户反馈处理结果**：
   * 反馈处理结果：系统将处理用户反馈的结果返回给用户，如问题解决方案或售后服务进度。

动态数据是随着用户的操作或系统运行状态而发生变化的数据，主要包括系统运行过程中产生的临时数据或状态数据。

* **用户登录状态**：用户在系统中的会话状态、登录时间、登录IP等。
* **购物车数据**：用户未完成交易的商品列表、数量等。
* **订单状态**：订单的实时处理状态，如未支付、已支付、待发货、已发货、已完成、已取消等。
* **支付记录**：每笔订单的支付方式、支付时间、支付金额等。

## 3.3数据库介绍

**用户信息表（UserInfo）：**存储平台用户的注册信息、登录凭证和个人资料。包括用户名、邮箱、密码（加密存储）、联系方式、地址等。

**商品信息表（ProductInfo）：**存储商城中所有商品的详细信息，如商品名称、描述、价格、库存、分类、品牌等。

**订单表（OrderInfo）：**存储用户的订单数据，包括订单编号、用户ID、商品ID、购买数量、订单状态、支付方式、支付时间、物流信息等。

**购物车表（CartInfo）：**存储用户临时添加到购物车中的商品信息，包括用户ID、商品ID、数量等，便于用户在提交订单前对购物车内容进行操作。

**支付记录表（PaymentInfo）：**存储每笔订单的支付信息，包括支付ID、订单ID、支付金额、支付时间、支付状态、支付方式等。

**评价与反馈表（FeedbackInfo）：**存储用户对商品或服务的评价与反馈信息，包括用户ID、商品ID、评价内容、评分、反馈时间等。

**下载记录表（DownloadInfo）（可选）：**如果需要记录用户对软件的下载情况，可以添加一个新的表，存储下载ID、用户ID、产品ID、下载时间等信息。

## 3.4数据词典

**用户信息表（UserInfo）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **长度** | **必填** | **默认值** | **描述** |
| **user\_id** | **INT** | **11** | **是** | **无** | **用户ID，主键，自增** |
| **username** | **VARCHAR** | **50** | **是** | **无** | **用户名，唯一** |
| **email** | **VARCHAR** | **100** | **是** | **无** | **用户电子邮箱，唯一** |
| **password** | **VARCHAR** | **255** | **是** | **无** | **加密存储的用户密码** |
| **phone\_number** | **VARCHAR** | **20** | **否** | **无** | **用户联系方式** |
| **address** | **VARCHAR** | **255** | **否** | **无** | **用户收货地址** |
| **registration\_date** | **DATETIME** | **无** | **是** | **当前时间** | **用户注册时间** |
| **last\_login** | **DATETIME** | **无** | **否** | **无** | **上次登录时间** |

**商品信息表（ProductInfo）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **长度** | **必填** | **默认值** | **描述** |
| **product\_id** | **INT** | **11** | **是** | **无** | **商品ID，主键，自增** |
| **product\_name** | **VARCHAR** | **100** | **是** | **无** | **商品名称** |
| **description** | **TEXT** | **无** | **否** | **无** | **商品描述** |
| **price** | **DECIMAL(10,2)** | **无** | **是** | **无** | **商品价格** |
| **stock** | **INT** | **11** | **是** | **0** | **库存数量** |
| **category** | **VARCHAR** | **50** | **是** | **无** | **商品分类** |
| **brand** | **VARCHAR** | **50** | **否** | **无** | **商品品牌** |

**订单表（OrderInfo）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **长度** | **必填** | **默认值** | **描述** |
| **order\_id** | **INT** | **11** | **是** | **无** | **订单编号，主键，自增** |
| **user\_id** | **INT** | **11** | **是** | **无** | **用户ID，外键，关联用户信息表** |
| **product\_id** | **INT** | **11** | **是** | **无** | **商品ID，外键，关联商品信息表** |
| **quantity** | **INT** | **11** | **是** | **无** | **购买数量** |
| **order\_status** | **VARCHAR** | **20** | **是** | **"Pending"** | **订单状态（Pending, Paid, Shipped, Delivered）** |
| **payment\_method** | **VARCHAR** | **30** | **否** | **无** | **支付方式** |
| **payment\_time** | **DATETIME** | **否** | **无** | **支付时间** |  |
| **logistics\_info** | **VARCHAR** | **255** | **否** | **无** | **物流信息** |

**购物车表（CartInfo）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **长度** | **必填** | **默认值** | **描述** |
| **cart\_id** | **INT** | **11** | **是** | **无** | **购物车ID，主键，自增** |
| **user\_id** | **INT** | **11** | **是** | **无** | **用户ID，外键，关联用户信息表** |
| **product\_id** | **INT** | **11** | **是** | **无** | **商品ID，外键，关联商品信息表** |
| **quantity** | **INT** | **11** | **是** | **1** | **购买数量** |
| **added\_date** | **DATETIME** | **无** | **是** | **当前时间** | **商品加入购物车时间** |

**支付记录表（PaymentInfo）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **长度** | **必填** | **默认值** | **描述** |
| **payment\_id** | **INT** | **11** | **是** | **无** | **支付ID，主键，自增** |
| **order\_id** | **INT** | **11** | **是** | **无** | **订单ID，外键，关联订单表** |
| **payment\_amount** | **DECIMAL(10,2)** | **无** | **是** | **无** | **支付金额** |
| **payment\_time** | **DATETIME** | **是** | **无** | **支付时间** |  |
| **payment\_status** | **VARCHAR** | **20** | **是** | **"Success"** | **支付状态（Success, Failed）** |
| **payment\_method** | **VARCHAR** | **30** | **是** | **无** | **支付方式** |

**评价与反馈表（FeedbackInfo）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **长度** | **必填** | **默认值** | **描述** |
| **feedback\_id** | **INT** | **11** | **是** | **无** | **反馈ID，主键，自增** |
| **user\_id** | **INT** | **11** | **是** | **无** | **用户ID，外键，关联用户信息表** |
| **product\_id** | **INT** | **11** | **是** | **无** | **商品ID，外键，关联商品信息表** |
| **feedback\_content** | **TEXT** | **无** | **是** | **无** | **反馈内容** |
| **rating** | **INT** | **5** | **是** | **无** | **评分（1-5）** |
| **feedback\_time** | **DATETIME** | **是** | **无** | **反馈时间** |  |

**下载记录表（DownloadInfo）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **长度** | **必填** | **默认值** | **描述** |
| **download\_id** | **INT** | **11** | **是** | **无** | **下载记录ID，主键，自增** |
| **user\_id** | **INT** | **11** | **是** | **无** | **用户ID，外键，关联用户信息表** |
| **product\_id** | **INT** | **11** | **是** | **无** | **商品ID，外键，关联商品信息表** |
| **download\_time** | **DATETIME** |  | **是** | **无** | **下载时间** |
| **file\_size** | **DECIMAL** | **10,2** | **否** | **0.00** | **下载文件大小（以MB为单位）** |
| **download\_status** | **VARCHAR** | **20** | **是** | **'Success'** | **下载状态（Success, Failed）** |
| **ip\_address** | **VARCHAR** | **45** | **否** | **无** | **用户下载时的IP地址** |

## 3.5数据采集

**3.5.1 获取用户和商品相关信息**

1. **用户输入**
   * **注册信息**：用户在注册时通过前端界面输入的个人信息，包括用户名、密码、联系方式和邮箱。
   * **登录信息**：用户在登录时输入的凭证，包括用户名或邮箱及密码。
   * **购物车操作**：用户通过点击添加到购物车按钮输入的商品信息及数量。
   * **订单信息**：用户在结账时填写的收货地址、支付方式及备注信息。
   * **反馈和评价**：用户在商品详情页或订单完成后填写的评价和反馈意见。
2. **系统自动生成**
   * **订单生成**：用户提交订单后，系统自动生成订单ID、生成时间、总金额等信息。
   * **支付记录**：用户支付后，系统记录支付ID、支付时间、支付状态等信息。
3. **外部数据接口**
   * **支付接口**：与支付网关的集成，自动获取支付结果和交易凭证。
   * **物流接口**：与物流服务提供商的数据接口，获取实时物流状态和预计送达时间。
4. **后台管理操作**
   * **商品管理**：管理员通过后台管理系统录入或更新商品信息，包括商品名称、描述、价格、库存等。
   * **用户管理**：管理员对用户信息进行维护，包括用户账号的启用/禁用、信息修改等。

**3.5.2 数据采集工具**

为确保数据采集的准确性和高效性，我们将采用以下工具和技术：

1. **前端开发框架**
   * 使用 React 或 Vue.js 等现代前端框架构建用户界面，以便用户方便地输入和管理数据。
2. **后端开发框架**
   * 使用 Spring Boot 等后端框架来处理数据采集、业务逻辑和数据库交互，确保数据的安全性和完整性。
3. **数据库管理系统**
   * 使用 MySQL 或 PostgreSQL 等关系型数据库管理系统，便于数据的存储、查询和维护。
4. **API接口**
   * 通过 RESTful API 与外部服务（如支付和物流系统）进行数据交互，确保数据的实时更新。
5. **数据验证工具**
   * 在数据输入环节，使用数据验证工具（如 JavaScript 表单验证、后端验证）确保用户输入数据的合法性和完整性。

**3.5.3 数据采集流程**

数据采集的整体流程如下：

1. **用户注册/登录**：
   * 用户通过注册页面输入信息，系统验证信息合法性，成功后将数据存入用户信息表（UserInfo）。
2. **商品浏览与选择**：
   * 用户在商品页面浏览商品，系统记录用户的操作并将选择的商品添加到购物车（CartInfo）。
3. **订单生成**：
   * 用户在购物车页面确认商品后，提交订单，系统生成订单信息，并保存至订单表（OrderInfo）。
4. **支付处理**：
   * 用户选择支付方式并提交支付请求，系统与支付网关进行交互，获取支付结果并记录到支付记录表（PaymentInfo）。
5. **下载记录采集**：
   * 用户在支付成功后下载商品，系统记录下载信息并保存至下载记录表（DownloadInfo）。
6. **用户反馈收集**：
   * 用户在商品详情页或订单完成后，填写评价与反馈，系统将反馈数据保存至评价与反馈表（FeedbackInfo）。

# 4．功能需求

## 4.1功能划分

**(一)商家模块**

1.商家注册与登录

查看和编辑资料，修改密码

2.商品管理

商品上架/下架，商品信息编辑（名称、描述、价格、库存、图片），商品分类管理

3.订单管理

查看订单状态，订单发货，处理退款请求

4.数据统计与分析

销售报表，用户反馈与评价管理

**（二）用户模块**

1. 用户注册与登录

2. 个人信息管理

查看和编辑个人资料，修改密码

3. 商品浏览与搜索

商品分类浏览（热门、推荐、新品、特价），商品搜索功能（关键词、分类、价格范围评分，品牌，销量），商品详细展示

4. 购物车管理

添加/删除到购物车，修改商品数量，查看购物车，购物车总价计算

5. 订单处理

下单流程（选择支付方式），订单支付，订单查询（待付款、待发货、已完成、已取消），订单取消（特定情况），订单跟踪（实时发布订单状态）

6. 商品评价与评论

评价商品，查看商品评论

7. 客服模块

在线客服聊天，常见问题解答（FAQ）页面，用户反馈提交功能

**（三）平台管理员模块**

1. 管理员登录

2. 商家管理

审核商家注册请求，管理商家信息（禁用/启用商家账户）

3. 商品审核

审核商品上架请求，管理商品信息

4. 用户管理

查看用户信息，管理用户账户（禁用/启用用户）

5. 数据统计与分析

平台销售数据统计，用户行为分析

6. 系统设置

配置支付方式，管理平台通知

## 4.2功能描述

**（一）商家模块**

1.商家注册与登录：

商家需提供营业执照、商家类型和联系方式进行注册。

登录界面需支持记住密码功能，并提供找回密码的选项。

2.商品管理：

商家可创建新商品，填写基本信息（名称、描述、价格、库存等）。

商品信息中应支持富文本格式，允许商家上传多张商品图片，并可添加视频介绍。

商家可对已上架商品进行编辑，更新库存和价格。

商品可按类别、品牌、价格等属性进行分类管理。

3.订单管理：

商家可查看所有订单的详细信息，包括下单用户、商品清单、订单状态（待发货、已发 货、已完成、已取消）。

发货时，系统将自动更新订单状态。

商家可查看并处理退款请求，提供退款原因，并更新订单状态。

4.数据统计与分析：

商家可生成月度或季度的销售报表，包含销售额、订单数量、退货率等数据。

商家可查看用户反馈和评价，便于改进商品质量和服务。

**（二）用户模块**

1.用户注册与登录：

用户需提供邮箱或手机号进行注册，系统会发送验证码以验证身份。

登录后，用户可选择设置二次验证，提高账户安全性。

2.个人信息管理：

用户可查看和编辑个人资料（昵称、头像、联系方式）。

提供修改密码和邮箱验证功能，确保账户安全。

3.商品浏览与搜索：

用户可通过首页的推荐、分类或搜索框快速找到商品。

支持多条件筛选（价格、销量、评分，关键词），并可按热销、上新、价格等排序。

商品详情页应包含详细信息、使用指南、保修政策等，提升用户购物体验。

4.购物车管理：

用户可将多个商品添加到购物车，购物车应支持实时更新。

用户可修改购物车商品的数量或删除不需要的商品，系统应实时计算总价。

5.订单处理：

用户下单时，系统需提供多种支付方式供选择（如信用卡、支付宝、微信支付）。

用户可在个人中心查看历史订单，跟踪物流状态，并可申请取消订单。

6.商品评价与评论：

用户购买商品后可在订单详情页进行评价，包括打分和文字评论。

支持用户上传使用商品的图片，评价将经过审核后展示在商品详情页。

**（三）平台管理员模块**

1.管理员登录：

管理员使用用户名和密码登录后台，支持权限管理。

2.商家管理：

管理员可审核新注册的商家，检查其提供的资料和营业执照的真实性。

管理员可随时禁用或启用商家账户，并能查看商家的历史违规记录。

2.商品审核：

管理员可审核商家上架的商品信息，确保合规和真实性。

管理员可删除不符合标准的商品，并向商家发送通知。

4.用户管理：

管理员可查看注册用户的信息，监控用户行为（如投诉、评价）。

对于不良用户，管理员可进行禁用账户或删除的操作。

数据统计与分析：

管理员可查看整体平台的销售数据，包括用户活跃度、商品销售趋势等。

提供分析报告，帮助优化平台运营策略。

6.系统设置：

管理员可设置第三方支付接口及相关参数，确保支付功能正常。

管理员可管理平台消息通知，包括系统公告、促销活动等。

# 5．性能需求

## 5.1数据精确度

· 系统需确保用户和商家输入的数据（如商品信息、订单信息）经过严格的格式和有效性校验，防止无效数据录入。

· 商品库存信息应实时更新，确保用户在下单时获取准确的库存情况。

## 5.2时间特性

· **响应时间**：

用户发起的操作（如商品搜索、购物车更新）应在1-2秒内得到响应。

系统应在用户下单后5秒内完成订单生成，并返回确认信息。

· **更新处理时间**：

商家提交商品信息的审核处理时间应不超过24小时。

订单支付处理完成后，系统需在3秒内更新订单状态。

· **数据转换与传输时间**：

数据在各模块间传输（如用户信息、订单信息）应在1秒内完成。

系统在进行数据库查询时，响应时间应控制在2秒内。

## 5.3适应性

· 系统应能够适应多种操作系统和浏览器环境，确保在主流浏览器（Chrome、Firefox、Safari、IE）和移动设备上均能良好运行。

· 若未来需要集成新的支付接口或支付方式，系统架构应支持灵活扩展，能够快速实现与新支付系统的对接。

· 系统应具备良好的模块化设计，以便在未来添加新功能或修改现有功能时，不影响其他模块的正常运行。

· 针对用户需求变化，系统应能快速响应并进行相应的功能调整，以保证用户体验的持续提升。

# 6．运行需求

## 6.1用户界面

6.1.1屏幕格式:

**整体布局**：采用响应式设计，确保在不同设备上都能良好展示，主要分为头部、主体和底部。

**头部**：包含商城Logo、搜索框和用户登录状态（如用户名、头像）。

**主体**：以网格布局展示产品，产品卡片包含图片、名称、价格和购买按钮。

**底部**：提供关于我们、联系方式、隐私政策等信息链接。

6.1.2 报表格式

**需求**：展示用户交易记录、收益报表等，以表格形式呈现。

**实现**：使用显著的样式和图表生成库（如EChart）生成可视化报表。

6.1.3 菜单格式

主菜单应包含“首页”、“交易”、“账户信息”、“帮助”等模块。

每个模块下应有子菜单，便于用户快速导航到具体功能。

## 6.2硬件接口

**服务器配置**：

* 需配置高性能服务器（至少16GB内存，4核CPU），以支持并发用户访问。
* 存储设备应具备RAID配置，确保数据安全性。

**用户终端**：

* 支持用户通过PC（Windows、macOS）和移动设备（iOS、Android）访问平台。

## 6.3软件接口

**第三方支付接口**：

* 与主流支付平台（如支付宝、微信支付、PayPal等）进行集成，确保交易的安全迅速。

**API接口**：

* 提供RESTful API，方便外部系统进行数据交互。
* 定义清晰的API文档，确保第三方开发者能够顺利接入。

**数据库接口**：

* 数据库应选择支持高并发的关系型数据库（如MySQL、PostgreSQL）。

## 6.4故障处理

**错误日志**

* 系统需集成日志记录工具（如Log4j或Winston），将所有操作及错误详细记录。

**故障恢复**

* **数据备份**：使用数据库备份工具（如mysqldump），设置自动定期备份机制（每日备份和实时增量备份）。
  + 实现过程：配置Cron作业来自动化备份任务。
* **用户提示**：在用户界面集成友好的错误提示框，使用Toast通知（如noty.js或SweetAlert）对错误进行提示。

# 7．其它需求

**可用性**

* **用户文档和帮助**：提供详细的用户手册和FAQ，使用Markdown或Wiki工具（如GitHub Wiki）。
* **用户反馈**：设计反馈机制，让用户能够方便地提交反馈和问题。

**安全保密**

* **数据加密**：使用SSL/TLS加密用户传输数据。对用户密码采用bcrypt等加密算法存储。
* **身份验证**：实现双因素认证（2FA），使用第三方Authenticator应用进行动态验证码生成。

**可维护性**

* **代码管理**：使用Git及GitHub进行版本控制，确保代码可追溯和管理。
* **文档维护**：每个功能模块均需编写详细文档，使用Apifox等工具生成API文档。

**可移植性**

* **技术选择**：采用跨平台技术栈（如Node.js和React），确保在主流操作系统上可运行。
* **容器化**：使用Docker将应用程序及其所有依赖打包，保证其在不同环境中的一致性。