【在线支付系统】

——付款交易处理子系统

总体设计说明书

组长：应承峻 3170103456@zju.edu.cn

组员：贺婷婷　张佳瑶　杨佳妮　韩汶东

日期：2019/05/24

版本：Version 1.0

目录

1 引言

1.1 编写目的

从本阶段开始，项目进入正式开发阶段。这份总体设计报告的编写目的，是以本项目的需求分析说明书为依据，从总体设计的角度，明确系统管理子系统的总体架构、流程、数据结构、数据库设计。

目的在于：

- 为开发人员提供依据
- 为修改、测试、维护提供条件
- 明确各模块外部接口，内部接口，用户接口
- 项目负责人将按计划说明书的要求布置和控制开发工作全过程

本说明书的预期读者包括：

- 软件客户
- 项目经理
- 项目开发人员
- 软件质量分析人员
- 系统维护人员

1.2 项目背景

软件系统名称

在线支付系统

任务提出者

浙江大学软件工程基础任课老师-王新宇

开发者

浙江大学 2018~2019 学年夏学期软件工程基础课程学生项目组

用户

卖家、买家、管理员

实现该软件的计算机网络

由若干台 PC 机组成的局域网

相关背景介绍

浙江大学软件工程基础课程分为理论课与实践课两个部分。在理论课中，教师有选择地介绍了与软件工程基础相关的理论；强调并确定了适用于整个软件生命期的基本原则，全面而深入地介绍了这些基本原则在软件设计、规范、验证、软件生产过程和管理 活动中的运用。而实验课采取分组形式完成，每 5 个学生为一组，分别设有组长、主程序员、程序员、测试员、文档员等角色。本次课程，教师选取教学服务系统作为综合性实验题目 。

1.3 相关定义

(1) MySQL

一个小型关系型数据库管理系统。

(2) Apache

世界使用排名第一的 Web 服务器软件，由于其跨平台和安全性被广泛使用，是最流行的 Web 服务器端软件之一。

(3) JavaScript

Javascript 是一种面向对象的动态类型的区分大小写的客户端脚本语言。

(4) AJAX

即“Asynchronous JavaScript and XML”（异步 JavaScript 和 XML)，是指一种创建交互式网页应用的网页开发技术。

(5) SQL 注入

通过把 SQL 命令插入到 Web 表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的 SQL 命令。

(6) 数据库连接池

数据库连接池负责分配、管理和释放数据库连接，它允许应用程序重复使用一个现有的数据库连接，这项技术能明显提高对数据库操作的性能。

(7) UTF-8

UTF-8 是 UNICODE 的一种变长字符编码又称万国码安全证书: 安全证书是在进行网上交易时的身份证，或者说是私人钥匙，安全证书是唯一的，与任何其他人的证书都不相同。

(8) MD5 加密算法

Message DigestAlgorithm MD5（中文名为消息摘要算法第五版）为计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护。

(9) Bootstrap

Bootstrap 是 Twitter 推出的一个用于前端开发的开源工具包。它由 Twitter 的设计师 Mark Otto 和 Jacob Thornton 合作开发，是一个 CSS/HTML 框架。

1.4 系统概述

在线支付系统是第三方担保的交易系统。任何买家都可以先向账户提交付款，然后系统会通知卖家托运货物。一旦买方收到并确认收到货物，系统就会把钱转移到卖方的账户，从而完成他们的网上交易。 一个完整的在线支付系统需要的模块有个人账户管理、付款交易处理、在线预定、账户对账和审核、系统管理。具体模块的功能要求如下:

（1）个人账户管理

基本用户（即买方或卖方）可以通过主界面使用他/她的基本个人信息（如真实姓名，身份证号码和电子邮件地址）注册账户，并随时修改此信息；处理基本的资金账户操作，包括设置和更改支付密码，计费账户以及查询账户余额；检查月度和年度的付款记录（清单）。

同时该模块为其他模块提供链接。

（2）付款交易处理

用户可以查看当前订单信息，包括订单总数，订购商品清单，交易金额，买方和卖方信息以及订单状态。买家能够执行支付和退款等操作，而卖家可以确认货物已经发货。卖家和卖家可以进行交易历史查询和投诉。

在交易前进行交易安全认证。根据交易状态（例如处理中，未付款，等待发货，等待确认，退款，完成和失败）和时间（例如今天，上周，上个月，过去三年月，去年，一年前）进行分类及展示，记录交易流程，提供该模块的接口，以用于账户核对和审计。

（3）在线预订

酒店和航班在线预订是在线支付系统的增值服务。用户可以使用此服务根据需要查找酒店和航班信息，预订房间和机票，并使用他们的账户进行付款。基本操作包括酒店和航班搜索，信息显示，预订和支付，评论和评分以及预订历史查询。

对于信息显示，折扣机票和特价房的信息有特别显示，将被列在右侧。搜索结果可以根据用户的需求进行智能排序。例如，酒店搜索结果可以按价格水平，酒店星级，热门水平和客户反馈评分进行排序，航班搜索结果可以按价格，航班时间，直飞航空公司和航空公司进行分类。

（4）账户对账和审核

审核员在完成交易时使用数据库中记录的事务流运行账户对账和审核。系统将在每天的固定时间生成

对账数据列表。该清单应包括每笔交易的订单号，买方和卖方 ID，总金额，订单状态和交易时间。如果出现错误或需要仔细检查的事项，应立即记录，并应警告审核员。

（5）系统管理

管理员可以通过后台界面执行以下任务：添加新管理员（包括系统和在线预订），维护管理员信息，管理用户（常规用户，VIP 用户和审核员），验证真实姓名以及管理仲裁和黑名单。构建和维护用于自动化实名验证的 ID 数据库。

1.5 项目概述

付款交易处理模块是在线支付系统的子系统，与其他子系统均有着密切的联系。付款交易处理子系统涉及功能较多，可以说是在线支付系统的核心模块所在。

本模块设置两种身份——买方和卖方，并为其提供不同的功能。买家能够执行支付、退款、确认收货、投诉等操作，而卖方可以执行确认发货、退款审核等操作。

为便于与账户核对和审计对接，该模块将记录交易流程并将该信息通过接口暴露给其他子系统。

此外，还应进行一定程度上的交易安全认证。金融账户攻击可能会遭到钓鱼攻击、中间人攻击、恶意软件攻击等，本模块将使用 SSL 加密技术保护敏感数据在传送过程中的安全，防范中间人攻击。

2 总体设计

2.1 需求规定

2.1.1 系统功能

根据需求，本项目需要提供账户信息服务器、API 和配套前端。服务器后端，本项目需设计一个储存账户信息和管理员信息的数据库。API 方面，本项目主要提供身份验证、订单状态修改、账户资金修改、历史查询、投诉 5 个 API。与 API 相对应，前端界面，本项目需要订单取消、订单支付、确认发货、确认收货、退款退货、投诉等功能。

2.1.2 系统性能

- 界面设计应简洁直观，布局合理，清晰地呈现信息，突出重点内容。 操作方便，用户容易上手。
- 系统具有良好的反应速度，给用户良好的使用体验。我们要求在良好的网络情况下，系统应具有以下时间特性要求：

  (1) 单个用户在线时 Web 响应用户动作时间小于 1 秒。信息搜索操作响应用户动作时间小于 2 秒。

(2) 500 个用户同时在线时 Web 响应用户动作时间小于 2 秒。信息搜索操作响应用户动作时间小于 5 秒。

- 访问容量：该系统至少在同一时间内支持 500 个用户并发访问。
- 服务器配置最低要求：CPU2.6G，内存 2.0G，硬盘 7200 转。
- 数据处理能力：至少支持 10000 笔交易记录。
- 可用性：该子系统应实现在大多数流行的 Web 浏览器中正确显示和执行，包括 Firefox、Chrome、Edge、IE 等。

2.1.3 输入输出要求

客户端通过网页展现给用户一个友好的界面，用户可以通过提交表单或者点击超链接向 服务器提供数据与命令。服务器后台处理后将结果显示到用户的网页界面上。

API 则为其他子系统和前端提供清晰、简洁的接口，子系统通过 API 向服务器发送请求，服务器后台处理后返回格式化的结果；若子系统进行非法操作，服务器能够进行判断并返回错误信息，避免发送的请求影响后端稳定性。

2.1.4 数据管理能力要求

2.1.4.1 安全

- 保密性
  1. 用于身份验证的用户名和密码应防止未经授权的用户访问系统。应构建访问控制以防止合法用户非法使用系统资源。
  2. 某些敏感数据（如用户名，密码和资本金额）在交换时应加密。密码在存储之前应加密。
  3. 在用户登录期间，应该防止 SQL 注入，密码强制破解和伪造会话入侵。
- 完整性
  1. 防止非法用户对数据进行无意或恶意的修改、插入、删除，防止数据丢失。
  2. 防止内部用户对数据进行无意或恶意的修改、插入、删除，防止数据丢失。
  3. 为数据库加上一定的约束，对关键性操作如删除、修改进行限制，并对用户进行警示。
  4. 定期备份数据。

2.1.4.2 性能

对于频繁访问数据库的操作，后台需要建立持久的数据库连接，以避免重复连接数据库耗费资源。

2.2 运行环境规定

2.2.1 服务器端

由于实验条件有限，我们并不能提供专门的服务器运行系统，故将利用配置较高的 PC 作为服务器，保证服务器以及客户端间网络畅通即可。

设备要求：

- CPU: 不小于 2.0GHz
- 内存：不小于 2.0GB

软件依赖：

- 操作系统：Windows Vista/7/8/8.1/10, Mac OS，Linux
- 数据库平台：MySQL
- Web 服务器：Apache
- MySQL 管理软件：PHPMyAdmin 或 MySQL WorkBench 等
- 开发工具：能支持网页开发的工具均可（如 IDEA）
- 测试工具：能支持测试的工具均可（如 JEST）
- 浏览器：Chrome、Edge 浏览器
- 前端框架：BootStrap
- 后端框架：Express

2.2.2 客户端

外围设备

- 键盘鼠标：可正常使用
- 显示器：可正常使用
- 硬盘：不小于 100GB
- 硬盘转速：不小于 7200rpm
- 通讯设备
- 网线：正常联通且数据传输能力良好
- 网卡：100M

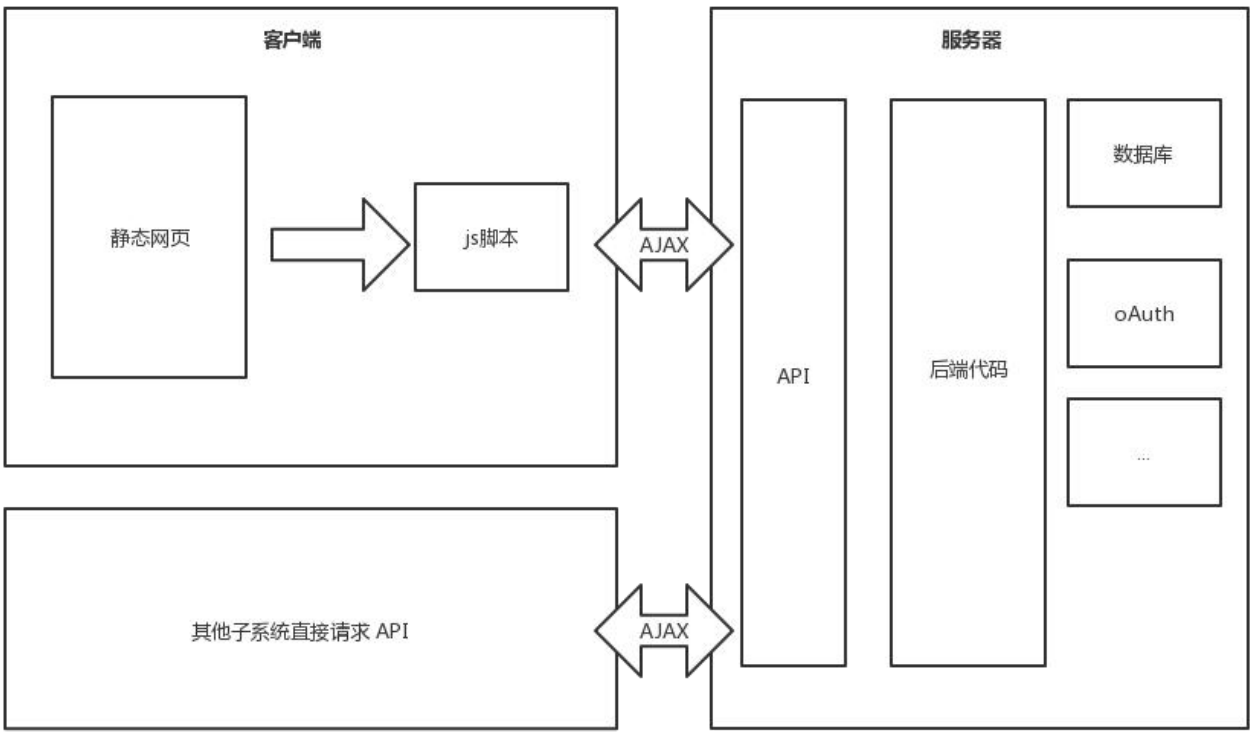
2.3 基本设计概念和处理流程

本子系统是一个横跨前后端的大模块，主要负责安全身份验证、订单信息修改、交易流程记录、投诉等。

- 服务器端：基于 node.js 的 express 框架

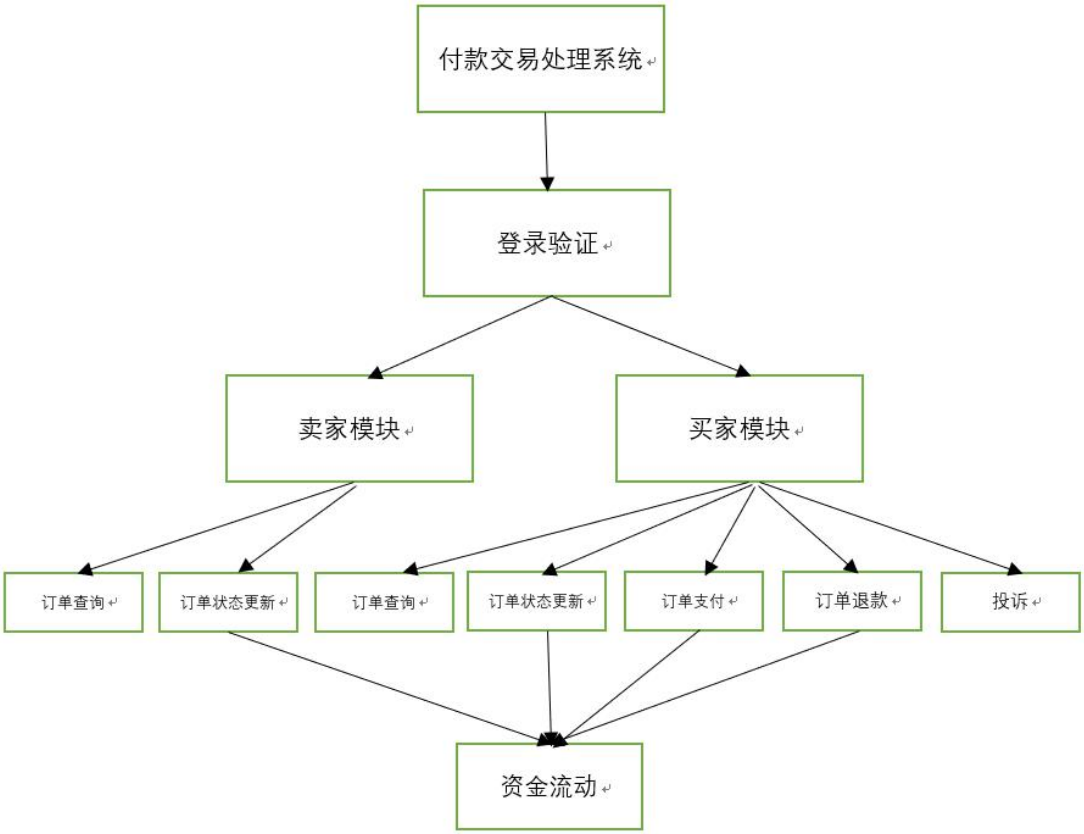
- 数据库：数据库采用 MySQL，通过 promise 来实现异步查询。
- 身份验证框架：采用 OAuth。
- 客户端：考虑到本系统所使用的框架较新，推荐使用 Chrome、Firefox、Safari 或 Microsoft Edge 浏览器。不推荐使用任何一个版本的 IE 浏览器，即使使用，版本须在 11 及以上。

处理流程图如下：



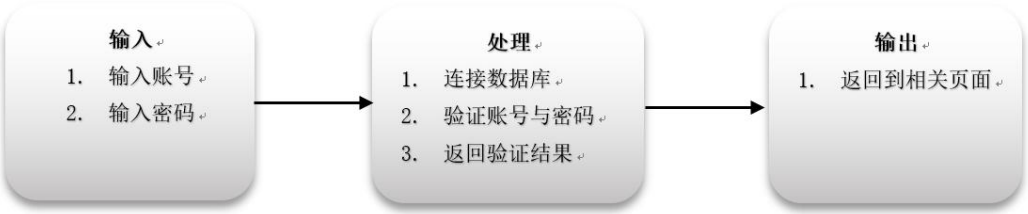
2.4 结构

2.4.1 付款交易处理系统层次图

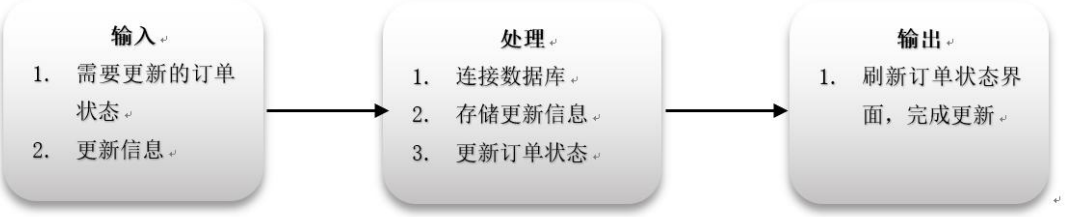


2.4.2 功能 IPO 图

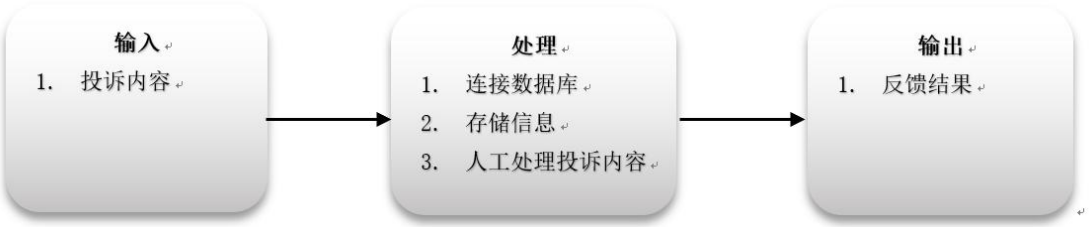
2.4.2.1 登录验证



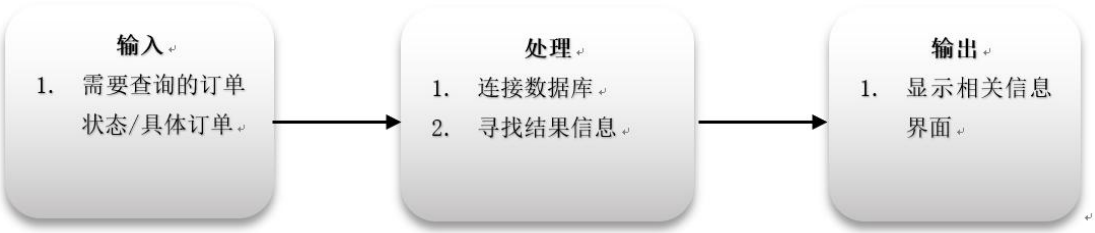
2.4.2.2 订单状态更新



2.4.2.3 投诉



2.4.2.4 订单查询



2.4.2.5 订单支付

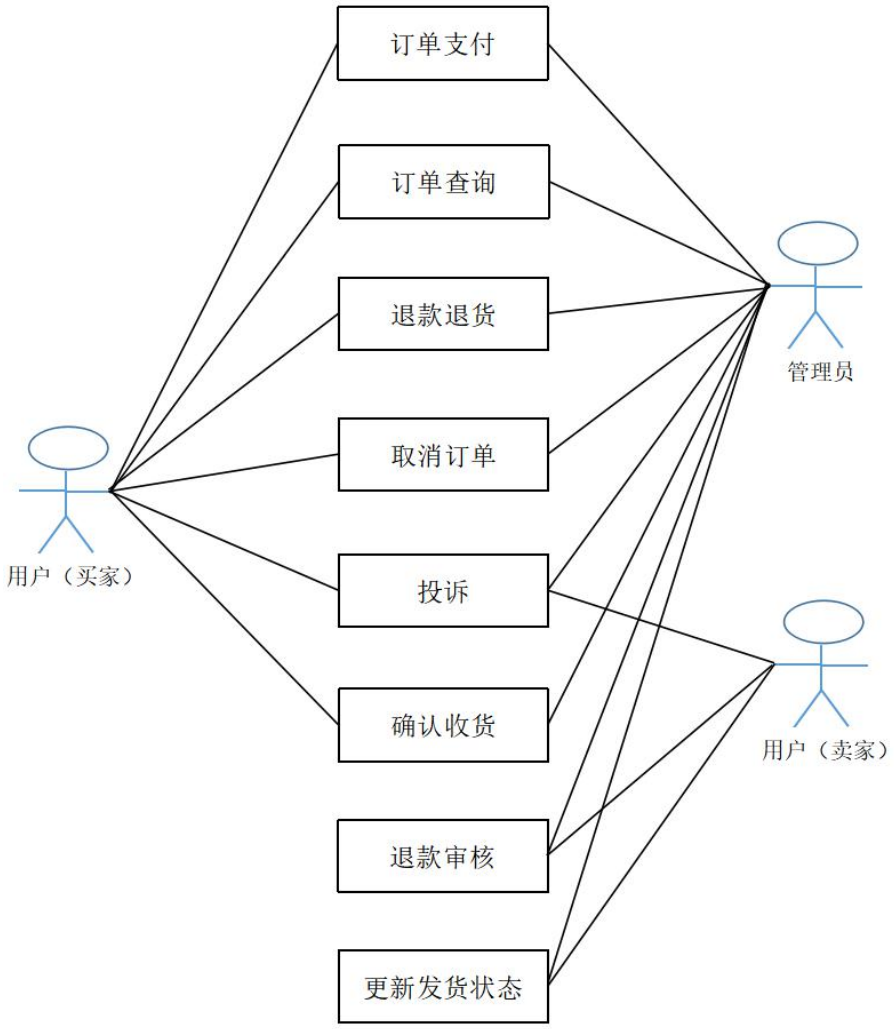
2.4.2.6 订单退款

2.4.2.7 资金流动

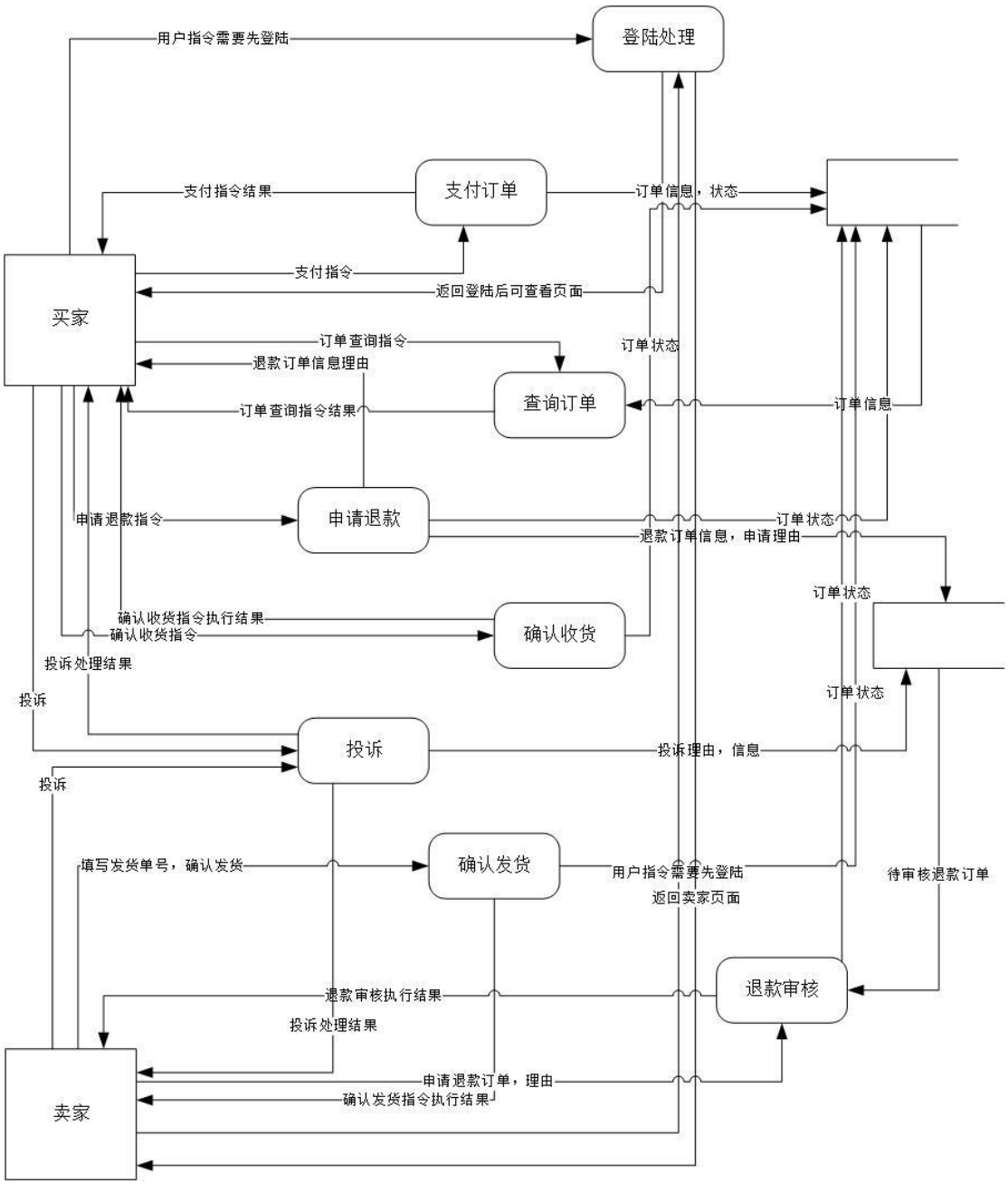
3 系统结构

3.1 系统功能结构

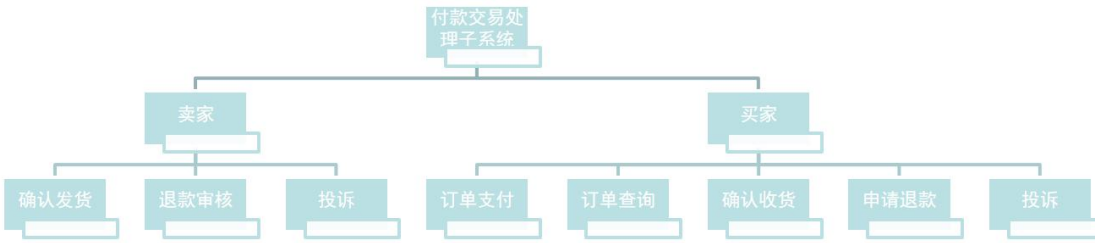
通过需求分析，已经完成对系统的用例分析，具体如下所示：



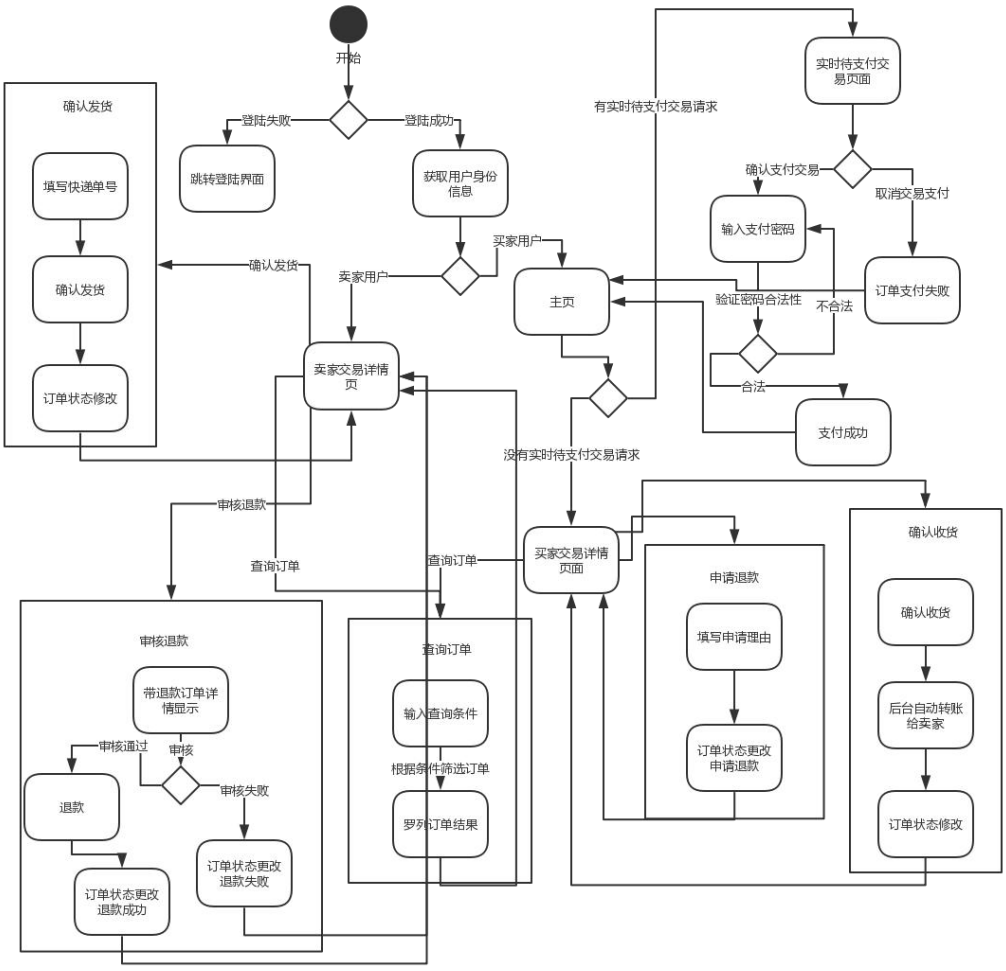
需求分析中，已经完成了对数据流相应的分析，具体如下图所示：



通过系统需求的分析，系统的总体结构设计变得明晰了。系统的系统层次图如下所示：



需求分析中完成了系统的状态图设计，这里引用状态图作为系统运行的整体流程参考。具体如下图所示:



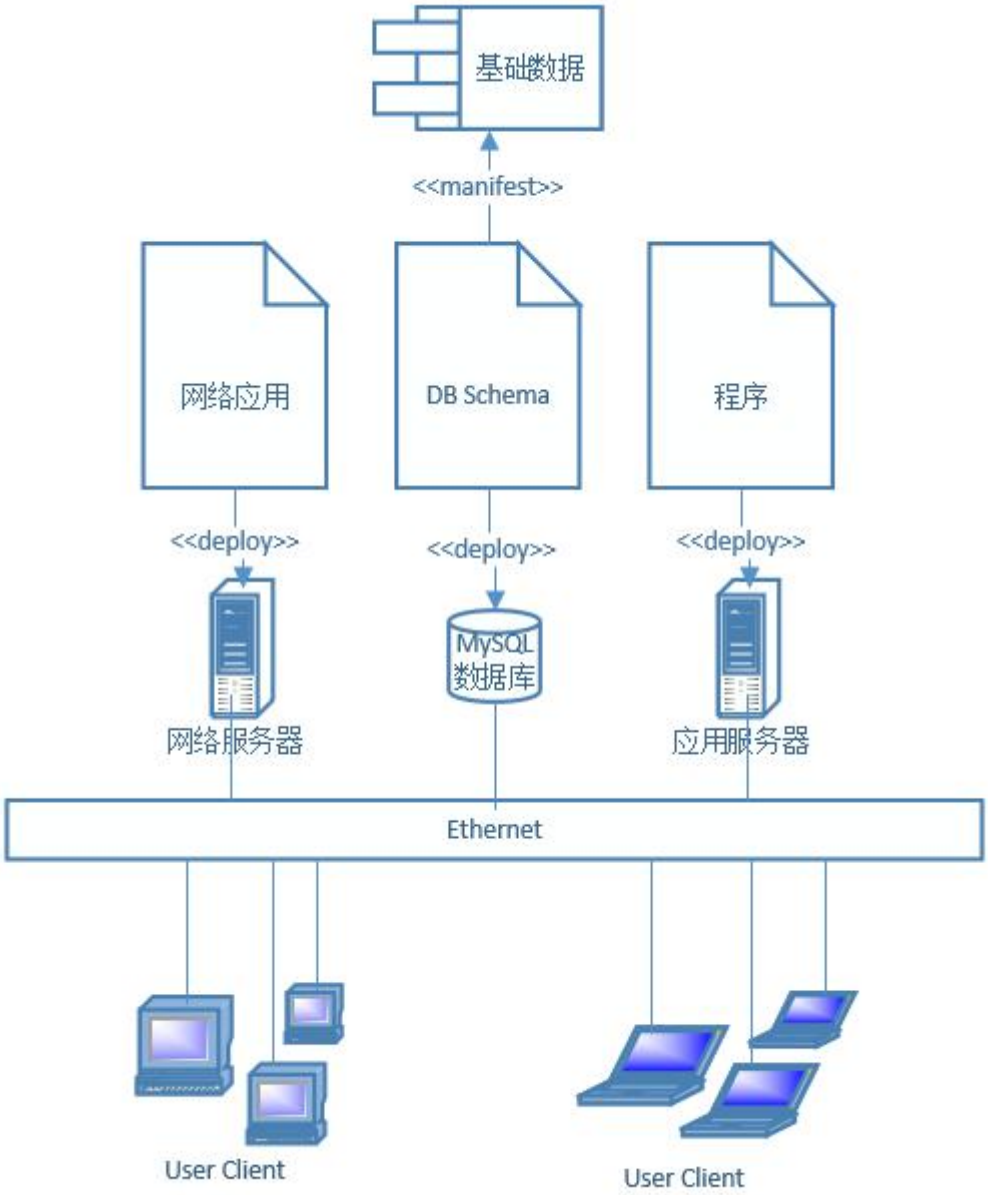
3.2 技术简介

在系统设计中，我们将采用基于 Node.js 的 Express 框架来进行 Web 页面的构建。Express 是一个简洁而灵活的 Node.js Web 应用框架，提供一系列强大特性来创建各种 Web 应用。Express 不对 node.js 已有的特性进行二次抽象，我们只是在它之上扩展了 Web 应用所需的功能。丰富的 HTTP 工具以及来自 Connect 框架的中间件随取随用，创建强健、友好的 API 变得快速又简单。

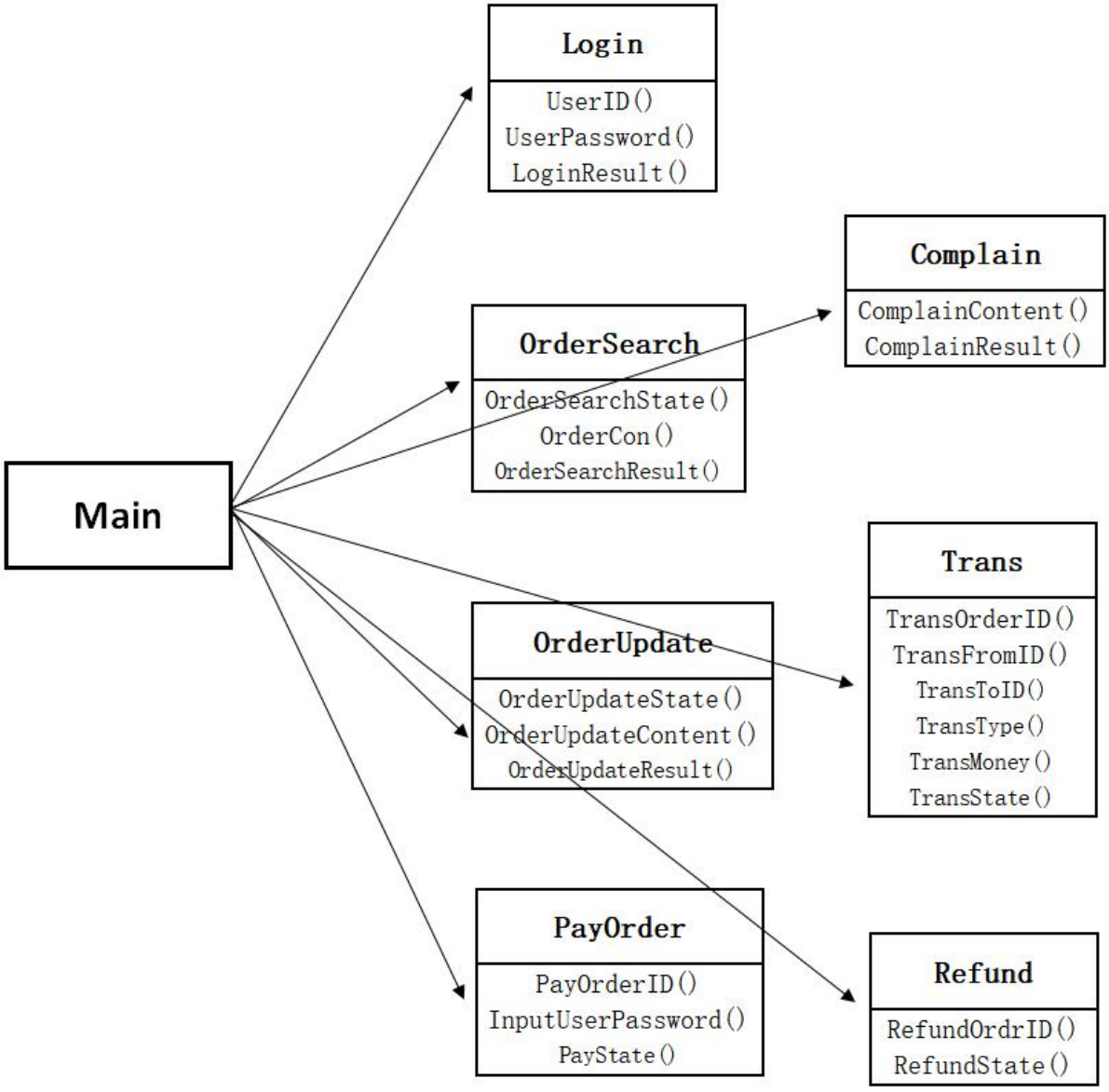
此外，我们将采用 MVC 模型对整个项目进行设计和实现。“MVC 模式（三层架构模式）(Model-View-Controller)是软件工程中的一种软件架构模式，把软件系统分为三个基本部分：模型(Model)、视图（View）和控制器（Controller）。” 在这种模式下，程序的开发人员能够将模型、视图、控制器这三个模块独立开来，从而能够对各个模块进行更为方便的修改和完善。在这个模式中，“模型” 不依赖“视

图”和“控制器”，它用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法，能够对数据库中的数据进行直接访问。 模型中数据的变化一般会通过一种刷新机制被公布，而为了实现该机制， 相应的视图必须事先注册，以了解相应的数据改变。“视图”的任务则是实现数据有目的的显示。 它需要访问相应的数据模型从而实现上面我们提到过的刷新功能。“控制器”则负责处理事件并作出响应。

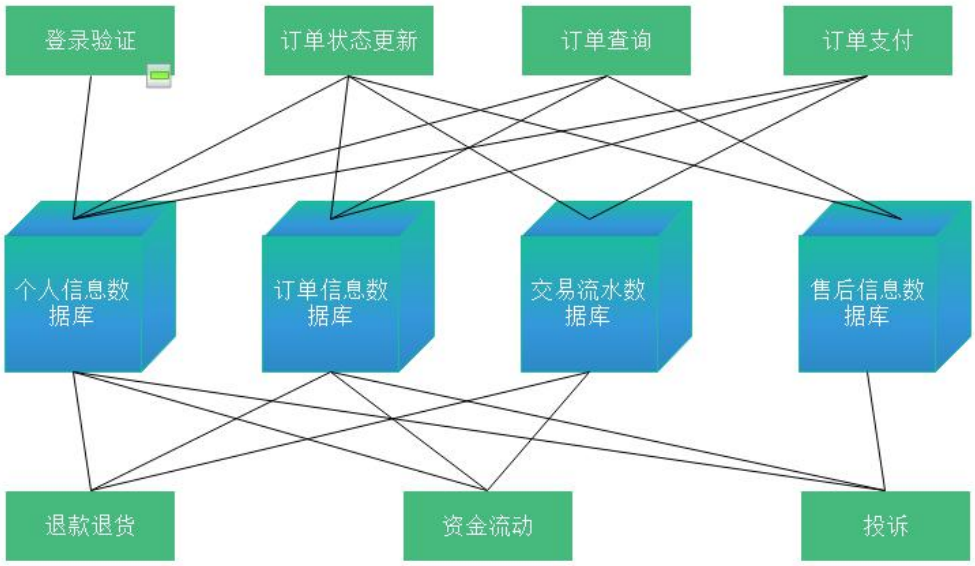
3.3 部署图



3.4 类图

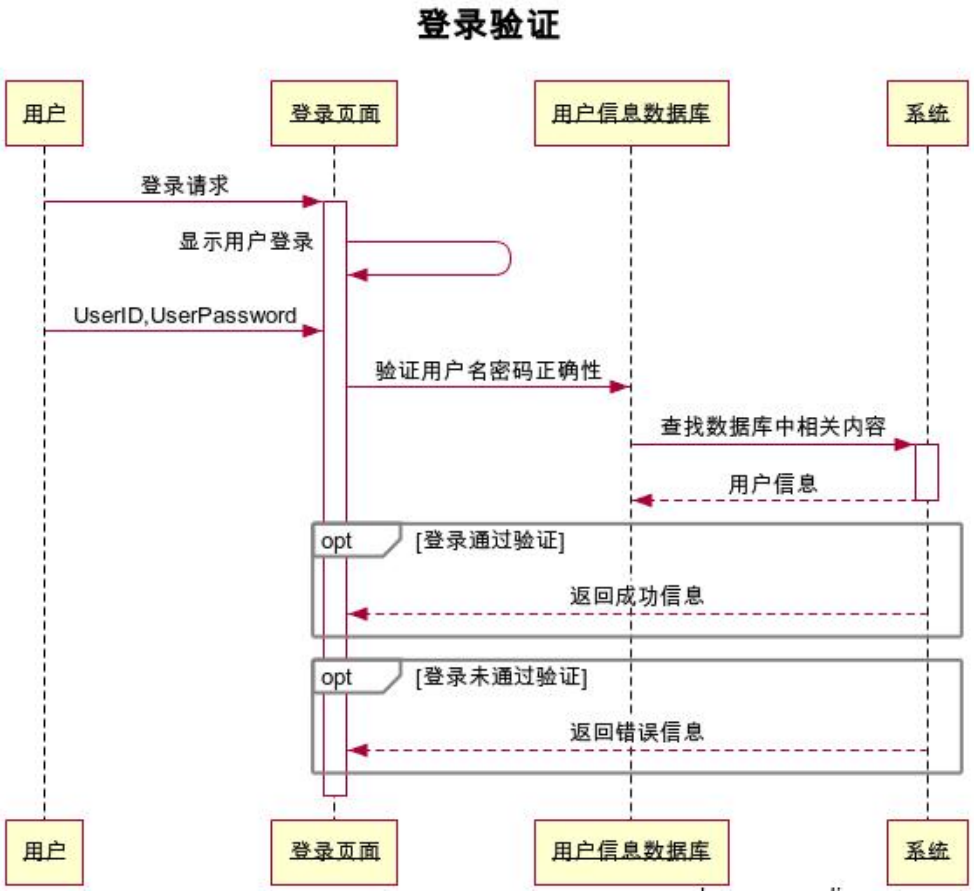


3.5 内部接口图

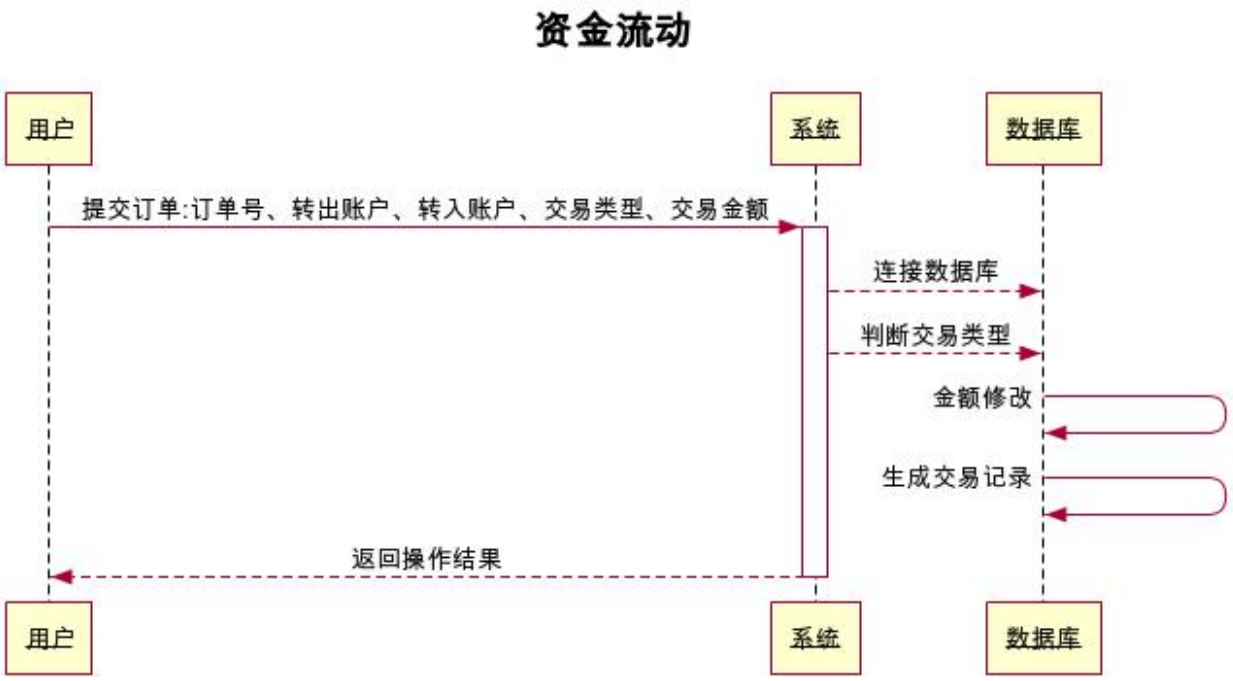


3.6 顺序图

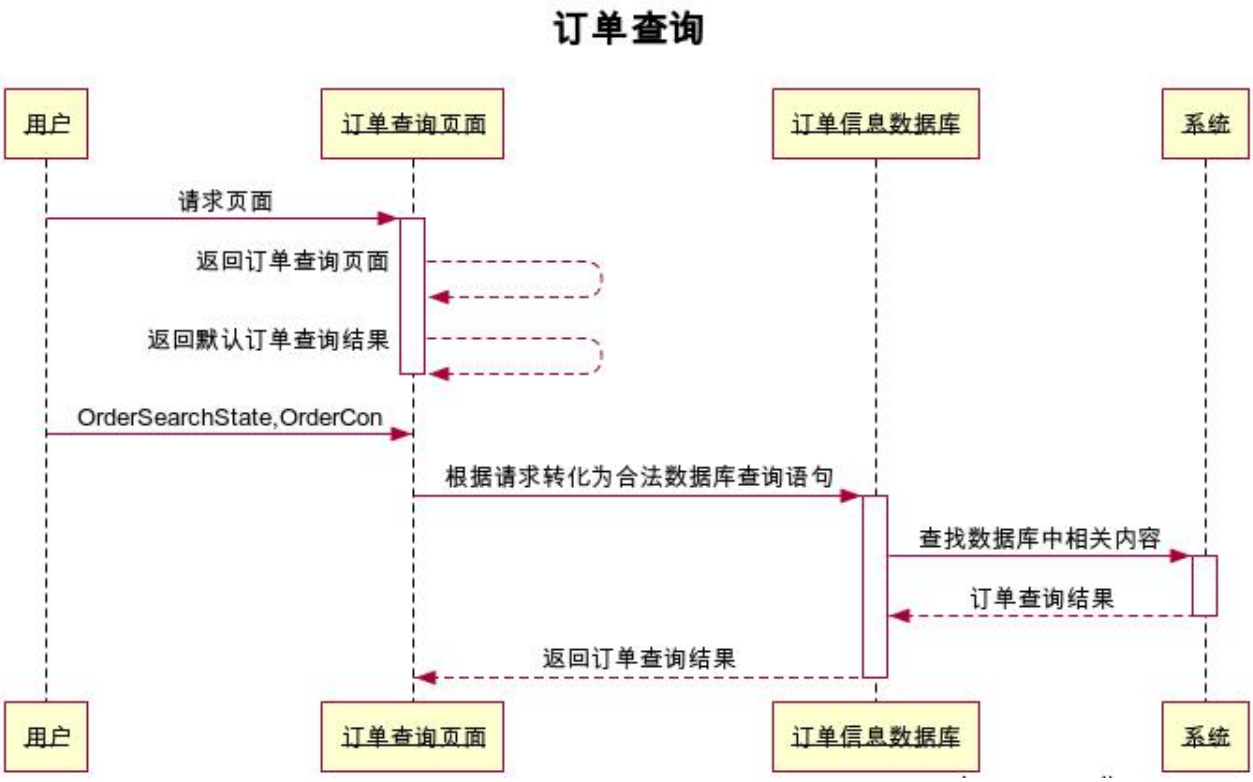
3.6.1 登录验证



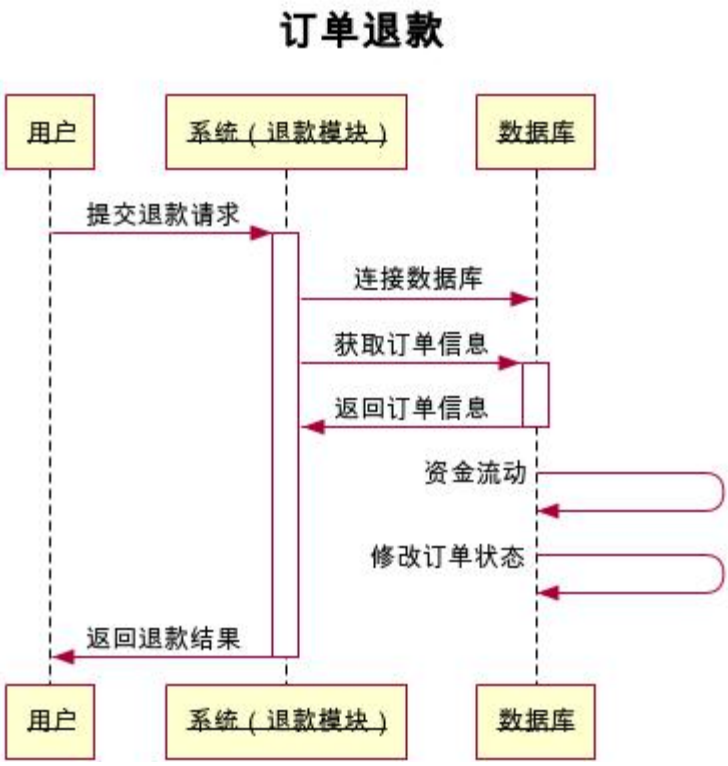
3.6.2 资金流动



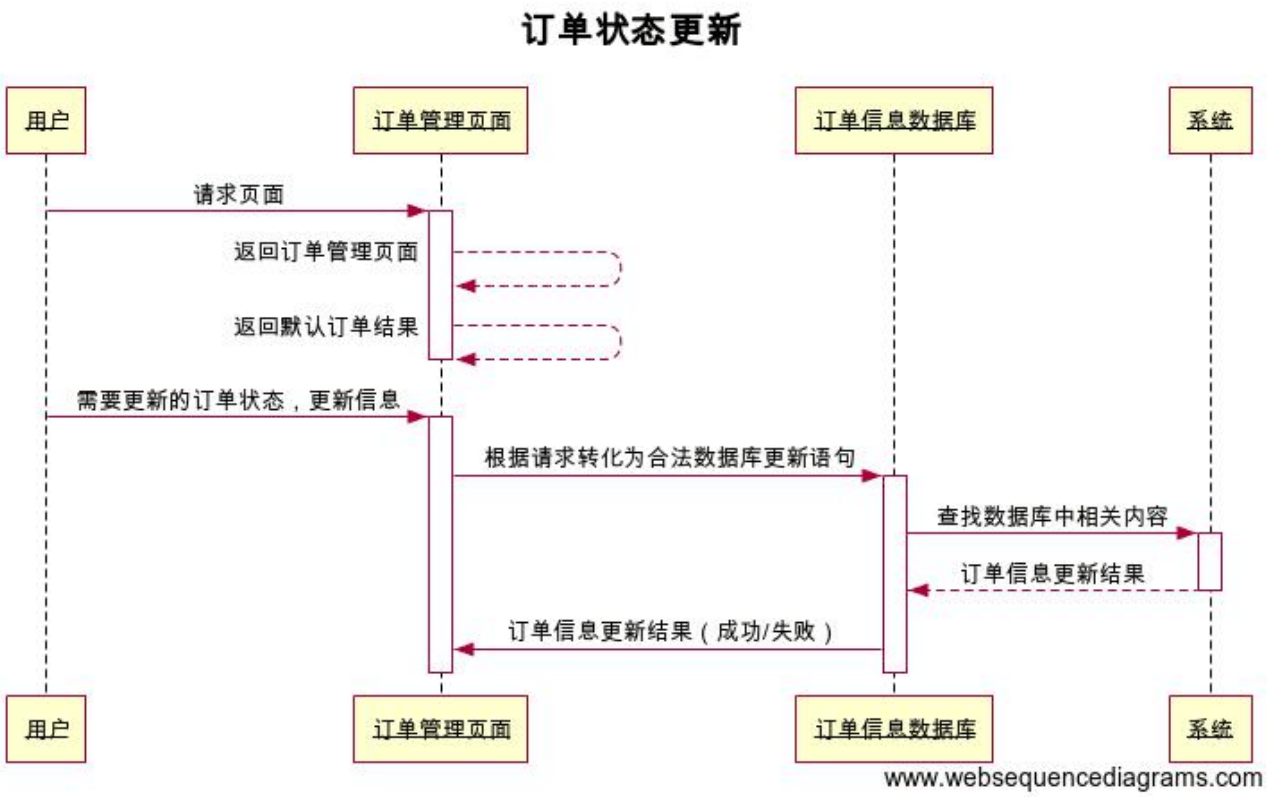
3.6.3 订单查询



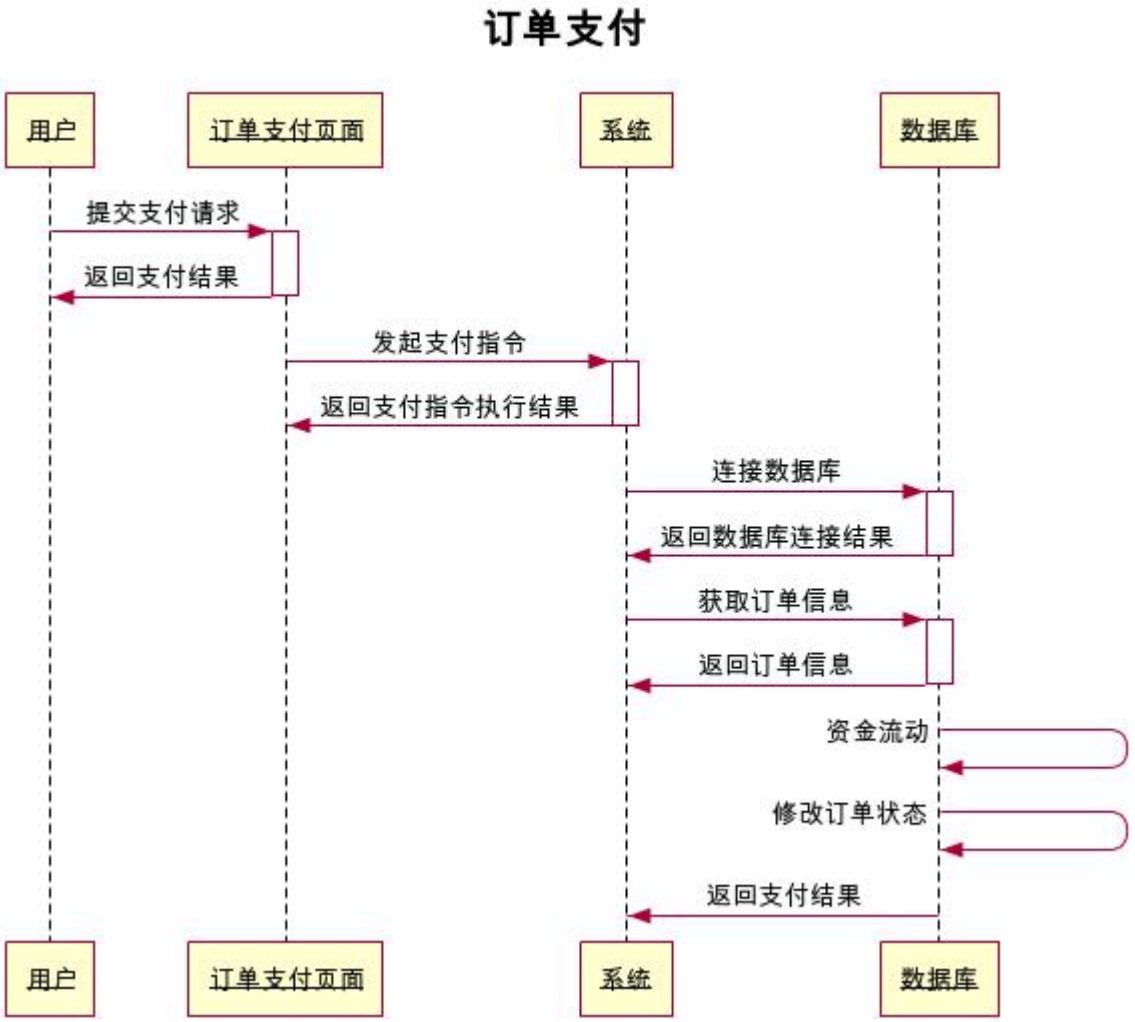
3.6.4 订单退款



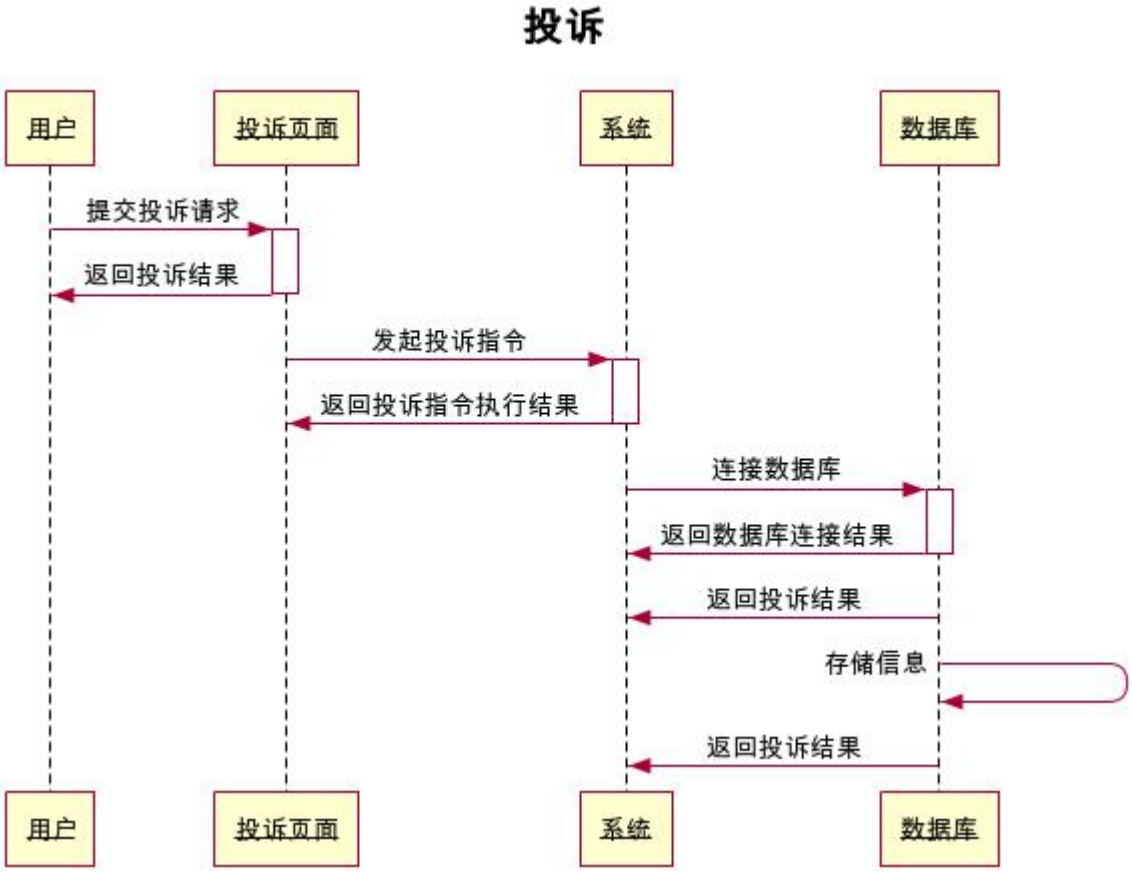
3.6.5 订单状态更新



3.6.6 订单支付



3.6.7 投诉



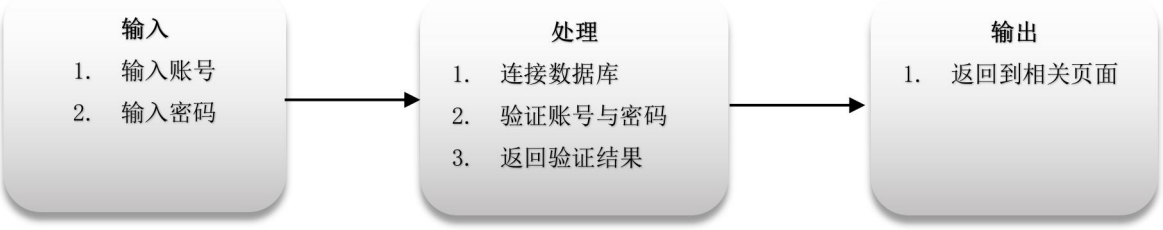
4 执行概念

4.1 登录验证

4.1.1 模块概述

本模块为用户（买方/卖方）在使用支付系统时的第一个界面，在此界面进行身份认证成功后才可使用其他功能。存在的目的是确认用户（买方/卖方）的身份，从而判断用户是否有权限使用其他功能，保证系统安全性。

4.1.2 IPO 图



4.1.3 功能

输入账号密码后登陆，继而进行身份认证。

4.1.4 输入项

名称	标识	类型和格式	输入方式
用户账号	UserID	Varchar(10)	外部输入
用户密码	UserPassword	Varchar(20)	外部输入

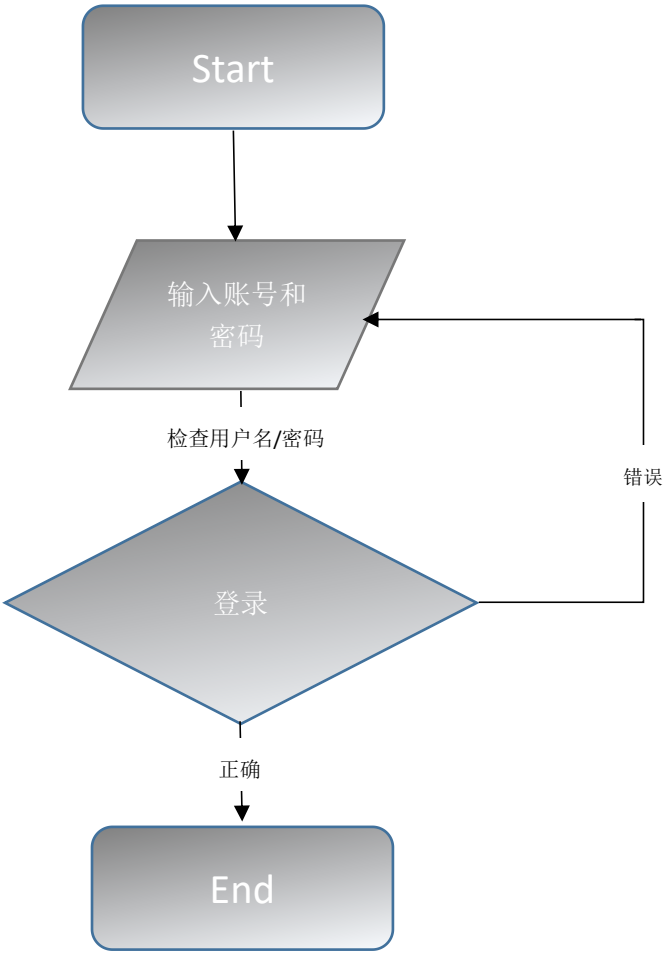
4.1.5 输出项

名称	标识	类型和格式	输出方式
登录结果	LoginResult	Bool	由脚本输出

4.1.6 设计方法（算法）

```
if 用户账号未登录：
    检验环境安全性
    创建数据库连接
    检测输入信息是否合法（如信息长度是否足够等）
    if 信息合法：
        存储信息
        查询数据库中相关信息
        进行比对验证
        if 比对成功：
            返回成功信息（登陆成功）
        else
            返回失败信息（密码错误）
    else
        返回失败信息（信息非法）
else 返回用户详细信息界面
```

4.1.7 流程图



4.1.8 测试计划

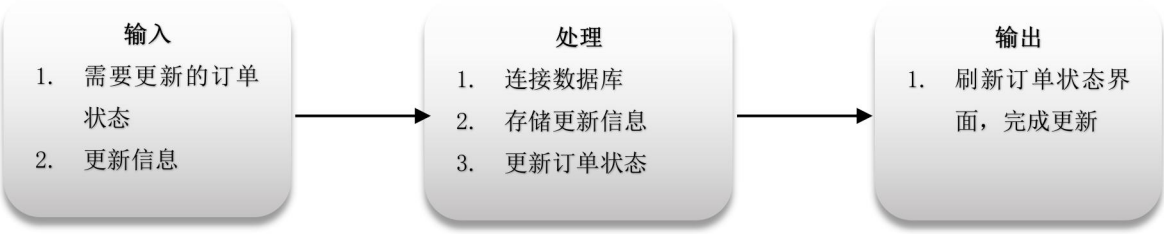
输入数据	预期结果
输入不合法的信息	返回错误信息，输入信息不合法
输入不匹配的账号和密码	返回错误信息，密码错误
输入匹配的账号和密码	返回成功信息，登陆成功

4.2 订单状态更新

4.2.1 模块概述

在这个模块中，用户（买方/卖方）可以根据实际情况更新订单状态。其中买方可更新的状态为：待付款、已完成、退款中，买方也可以在一定条件下取消订单；卖方可更新的状态为：待发货。用户（买方/卖方）可更具交易流程及时更新订单状态。

4.2.2 IPO 图



4.2.3 功能

根据线下情况手动更新订单状态。

4.2.4 输入项

名称	标识	类型和格式	输入方式
需要更新的订单状态	OrderUpdateState	Int	外部选择 —>脚本转换
更新信息	OrderUpdateContent	Varchar(100)	外部输入

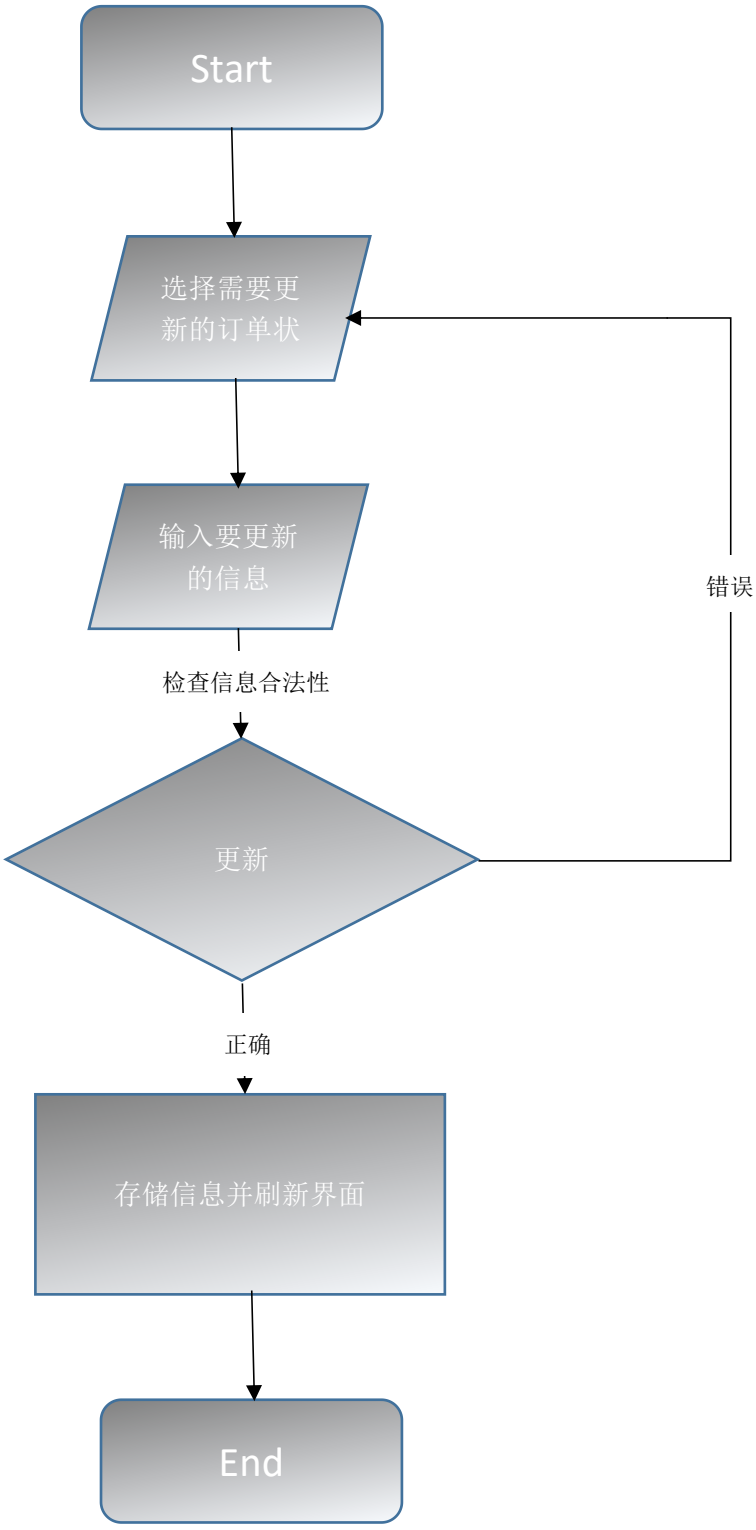
4.2.5 输出项

名称	标识	类型和格式	输出方式
更新结果	OrderUpdateResult	Bool	由脚本输出

4.2.6 设计方法（算法）

```
1. if 用户已登录
2.     创建数据库连接
3.     记录用户选择项
4.     存储输入的更新信息
5.     if 信息验证合法
6.         更新数据库订单状态信息
7.         返回成功信息（刷新界面）
8.     else
9.         返回失败信息（信息不合法）
10. else 返回失败信息（跳转到登录界面）
```

4.2.7 流程图



4.2.8 测试计划

输入数据	预期结果
未登录，输入信息	返回错误信息，跳转到登录界面

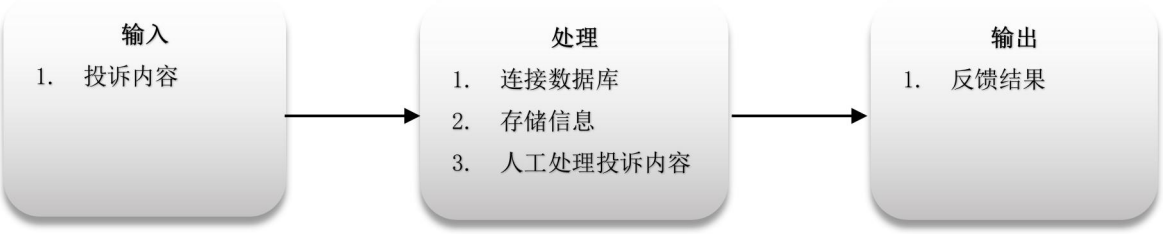
输入信息不合法	返回错误信息，输入信息不合法
已登录，输入信息合法	返回成功信息，更新成功

4.3 投诉

4.3.1 模块概述

在整个交易过程中用户（买方/卖方）如果对交易过程有不满意之处或者有建议，可以在此模块进行投诉。投诉内容将由平台收集，待平台处理后，将会进行反馈。

4.3.2 IPO 图



4.3.3 功能

提交用户的投诉内容，平台进行收集。

4.3.4 输入项

名称	标识	类型和格式	输入方式
投诉内容	ComplainContent	Varchar(200)	外部输入

4.3.5 输出项

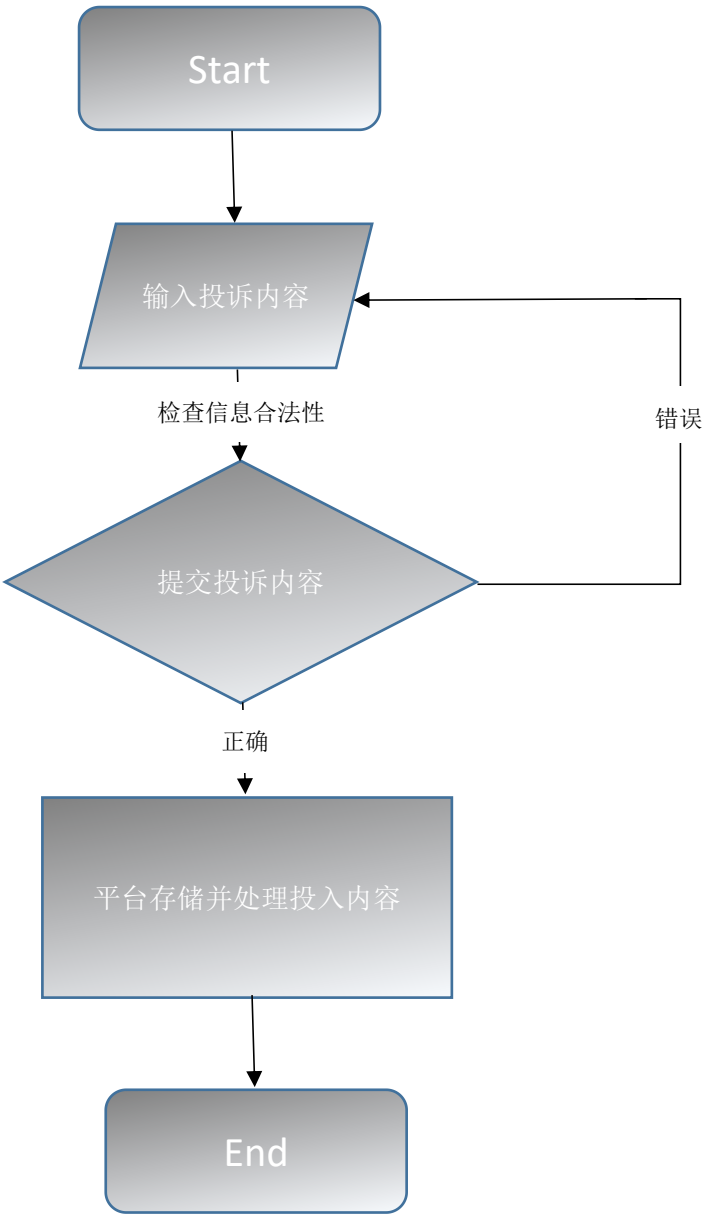
名称	标识	类型和格式	输出方式
投诉结果	ComplainResult	Varchar(50)	由脚本输出

4.3.6 设计方法（算法）

```
1. if 用户已登录
2.     创建数据库连接
3.     记录用户输入信息
```

```
4.      if 信息验证合法
5.          平台记录投诉信息
6.          返回成功信息（投诉成功）
7.      else
8.          返回失败信息（信息不合法）
9.  else 返回失败信息（跳转到登录界面）
```

4.3.7 流程图



4.3.8 测试计划

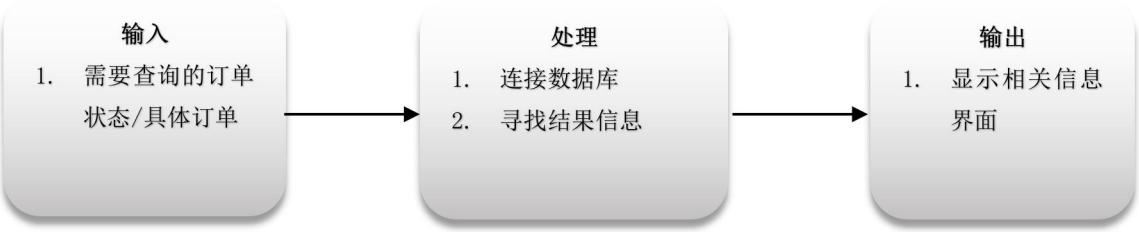
输入数据	预期结果
未登录，输入信息	返回错误信息，跳转到登录界面
输入信息不合法	返回错误信息，输入信息不合法
已登录，输入信息合法	返回成功信息，投诉成功

4.4 订单查询

4.4.1 模块概述

在这个模块中，用户（买方/卖方）可以对交易订单的状态或具体订单详情进行查询。买方可查询的订单状态为：待付款、待发货、待收货、已完成、退款中和已关闭；卖方可查询的订单状态为：待收款、待发货、已发货、退款中和已关闭。用户（买方/卖方）可随时对订单状态进行查询，掌握交易进程。

4.4.2 IPO 图



4.4.3 功能

用户查询订单状态和具体订单的详细信息。

4.4.4 输入项

名称	标识	类型和格式	输入方式
需要查询的订单状态	OrderSearchState	Int	外部选择—>脚本转换
(选填)具体订单	OrderCon	Int	外部选择—>脚本转换

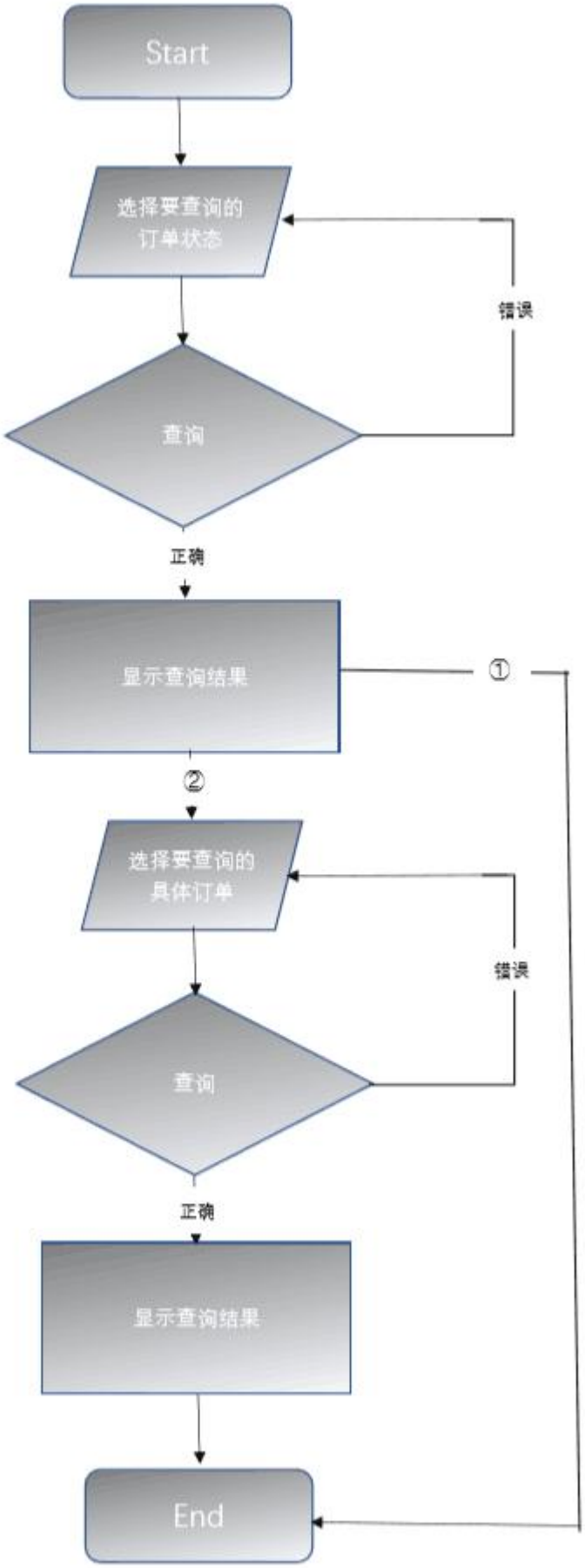
4.4.5 输出项

名称	标识	类型和格式	输出方式
查询结果	OrderSearchResult	Bool	由脚本输出

4.4.6 设计方法（算法）

```
if 用户已登录
    创建数据库连接
    记录用户选择信息
    根据选择读取数据库信息
    返回成功信息(刷新界面)
    if 用户选择具体订单信息
        记录用户选择信息
        根据选择信息读取数据库信息
        返回成功信息（刷新界面）
    else
        在此界面停留
   else 返回失败信息，跳转到登录界面
```

4.4.7 流程图


Start
选择要查询的
订单状态
错误
查询
正确
显示查询结果
①
②
选择要查询的
具体订单
错误
查询
正确
显示查询结果
End

4.4.8 测试计划

输入数据	预期结果
未登录，输入选择信息	返回错误信息，跳转到登录界面
已登录，输入订单状态选择信息	返回成功信息，刷新界面显示相关状态订单
已登录，输入具体订单选择信息	返回成功信息，刷新界面显示该订单详细信息

4.5 订单支付

4.5.1 模块概述

此模块为本子系统“付款交易处理”的核心模块，负责完成“付款”步骤。

该模块接收订单信息与用户信息，要求用户在确认信息后输入交易认证密码，在通过交易认证后完成资金的流动、订单信息的更改，最后将本次交易相关的信息传送给资金流动模块生成交易记录，返回支付结果。

该模块目的在于根据订单交易信息修改账户余额以实现现实意义上的交易。

4.5.2 IPO 图

4.5.3 功能

用户完成支付时的身份验证功能，系统对支付前的状态进行审核。

4.5.4 输入项

名称	标识	类型和格式	输入方式
订单 ID	PayOrderID	Int	外部选择->脚本传输
用户输入密码	InputUserPassword	Varchar(20)	外部输入

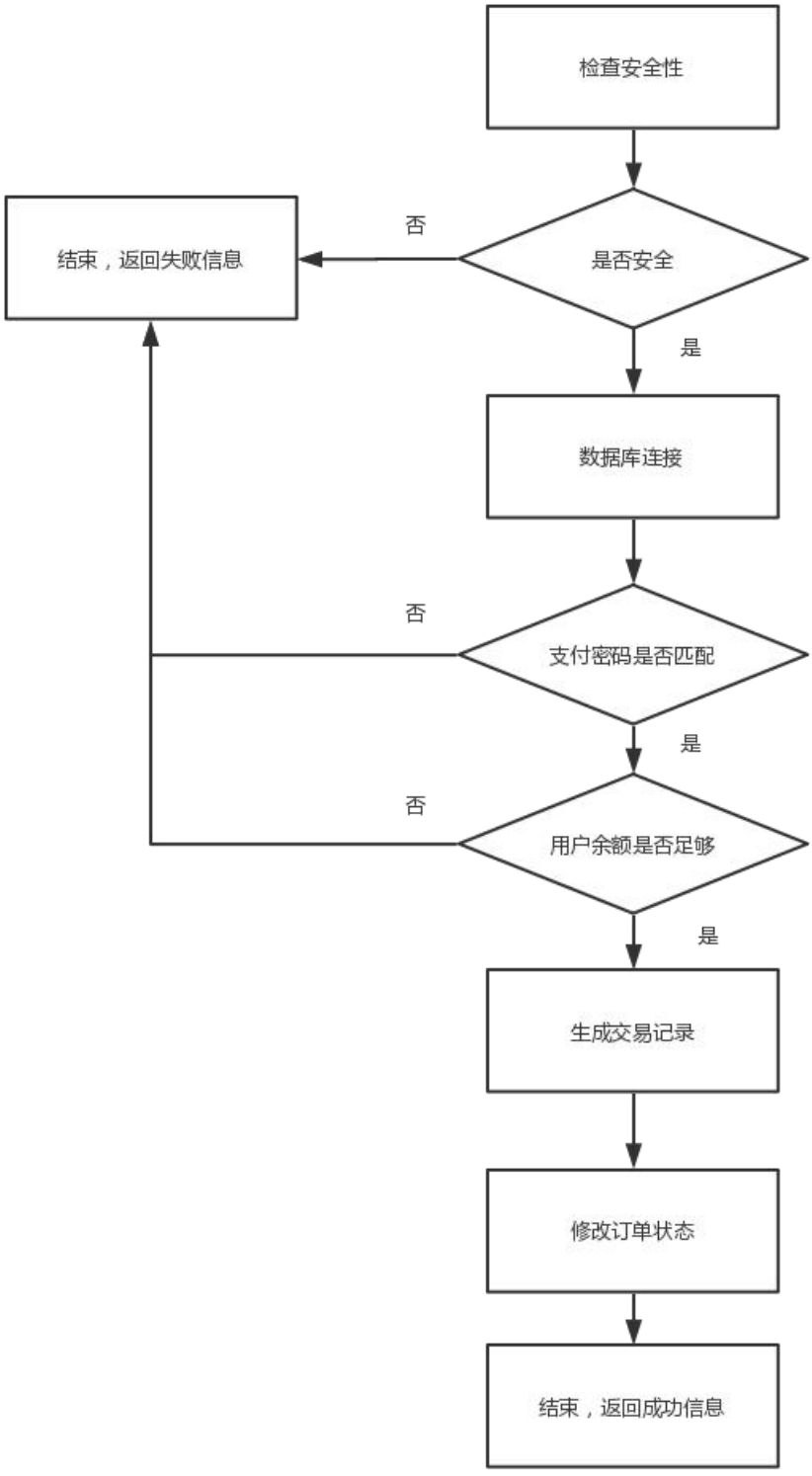
4.5.5 输出项

名称	标识	类型和格式	输出方式
操作状态	PayState	String	由脚本输出

4.5.6 设计方法（算法）

```
if 用户已登录：
    检验安全性
    创建数据库连接
    if 用户信息匹配：
        通过 ID 查询数据库中用户相关信息（密码、余额）
        支付密码认证
        if 认证通过：
            if 用户余额足够：
                发送信息至资金流动模块生成交易记录
                if 交易成功：
                    修改订单状态
                    返回成功信息
                else：
                    返回失败信息（交易失败）
            else：
                返回失败信息（用户资金不足）
        else：
            返回失败信息（用户认证不通过）
    else：
    返回失败信息（用户信息不匹配）
else：
    返回失败信息（用户未登录）
```

4.5.7 流程图


检查安全性
是否安全
否
结束，返回失败信息
是
数据库连接
支付密码是否匹配
否
是
用户余额是否足够
否
是
生成交易记录
修改订单状态
结束，返回成功信息

4.5.8 测试计划

集成测试:

向该模块传输测试数据，检查输出结果及数据库前后变化

输入数据	预期结果
未登录时输入数据	返回未登录信息，数据库未修改
输入用户 ID 与买方 ID 不匹配的数据	返回错误信息，数据库未修改
输入订单金额大于账户资金的数据	返回错误信息，数据库未修改
输入用户输入密码错误的数据	返回错误信息，数据库未修改
输入合理正常的支付数据	返回成功信息，数据库做出相应修改

4.6 订单退款

4.6.1 模块概述

此模块为本子系统“付款交易处理”中涉及交易处理的另一层核心模块，负责完成退款操作。

该模块接收订单信息与用户信息，通过交易认证后完成资金的流动、订单信息的更改，最后将本次交易相关的信息传送给资金流动模块生成交易记录，返回退款结果。

该模块目的在于根据订单信息修改账户余额以实现现实意义上的退款。

4.6.2 IPO 图

4.6.3 功能

用户完成退款时的身份验证功能，系统对退款支付前的状态进行审核。

4.6.4 输入项

名称	标识	类型和格式	输入方式

订单 ID	RefundOrderID	Int	外部选择->脚本传输

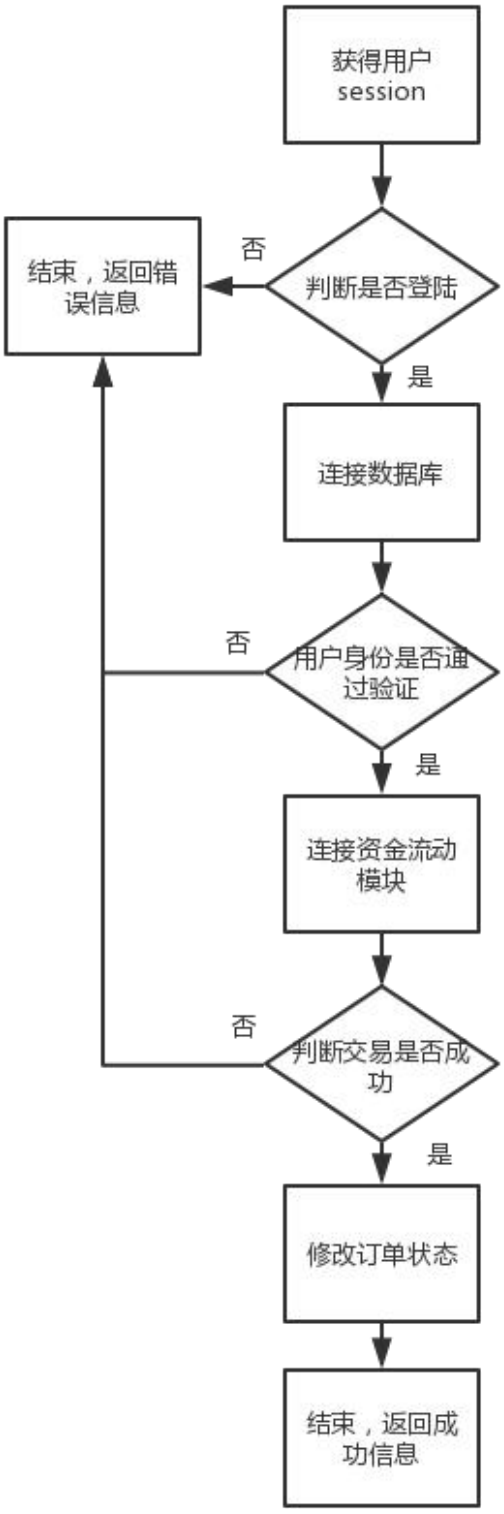
4.6.5 输出项

名称	标识	类型和格式	输出方式
操作状态	RefundState	String	由脚本输出

4.6.6 设计方法（算法）

```
if 用户已登录：
    检验安全性
    创建数据库连接
    if 用户信息匹配，通过认证：
        根据订单 ID 获取订单信息
        发送信息至资金流动模块生成交易记录
        if 交易成功：
            修改订单状态
            返回成功信息
        else：
            返回失败信息（交易失败）
    else：
        返回失败信息（用户信息不匹配）
else：
    返回失败信息（用户未登录）
```

4.6.7 流程图


获得用户session
判断是否登陆
否
结束，返回错误信息
是
连接数据库
用户身份是否通过验证
否
是
连接资金流动模块
判断交易是否成功
否
是
修改订单状态
结束，返回成功信息

4.6.8 测试计划

集成测试:

向该模块传输测试数据，检查输出结果及数据库前后变化

输入数据	预期结果
未登录时输入数据	返回未登录信息，数据库未修改
输入用户 ID 与订单卖方 ID 不匹配的数据	返回错误信息，数据库未修改
输入合理正常的退款数据	返回成功信息，数据库做出相应修改

4.7 资金流动

4.7.1 模块概述

该模块目的在于记录每次交易产生的出入账记录并完成资金的流动，接收交易涉及信息如转出账户、转入账户、涉及订单、交易类型等，返回交易结果。

由于入口有多个模块（支付、收货、退款），故将其单独抽象出来进行处理。

4.7.2 IPO 图

4.7.3 功能

用户在完成身份验证后，进行实际余额的扣除与平台余额的变化，随后生成交易信息（入账、出账）以便结算和对账，最后返回操作结果。

4.7.4 输入项

名称	标识	类型和格式	输入方式
订单号	TransOrderID	Int	脚本传输
转出账户	TransFromID	Int	脚本传输

转入账户	TransToID	Int	脚本传输
交易类型	TransType	Int	脚本传输
交易金额	TransMoney	Int	脚本传输

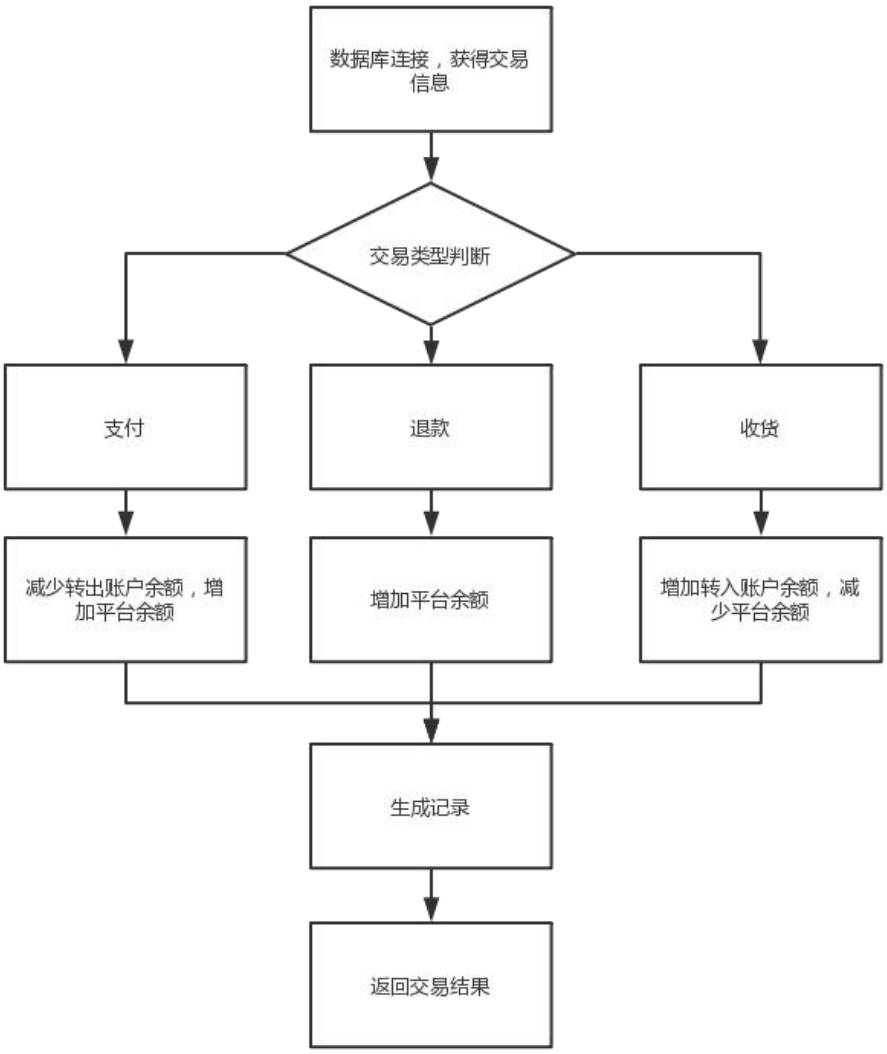
4.7.5 输出项

名称	标识	类型和格式	输出方式
操作状态	TransState	String	由脚本输出

4.7.6 设计方法（算法）

```
1. 连接数据库
2. switch（交易类型）：
3.     case（支付）：
4.         减少转出账户余额
5.         增加平台余额
6.         生成记录（转出账户，转出，交易金额，支付，订单号）
7.     case（退款）：
8.         增加平台余额
9.         生成记录（转出账户，转出，交易金额，退款支出，订单号）
10.        减少平台余额
11.        生成记录（转入账户，转入，交易金额，退款收入，订单号）
12.    case（收货）：
13.        增加转入账户余额
14.        减少平台余额
15.        生成记录（转出账户，转出，交易金额，支付，订单号）
16. 返回操作结果（交易成功/失败）
```

4.7.7 流程图



4.7.8 测试计划

集成测试：

向该模块传输测试数据，检查输出结果及数据库前后变化

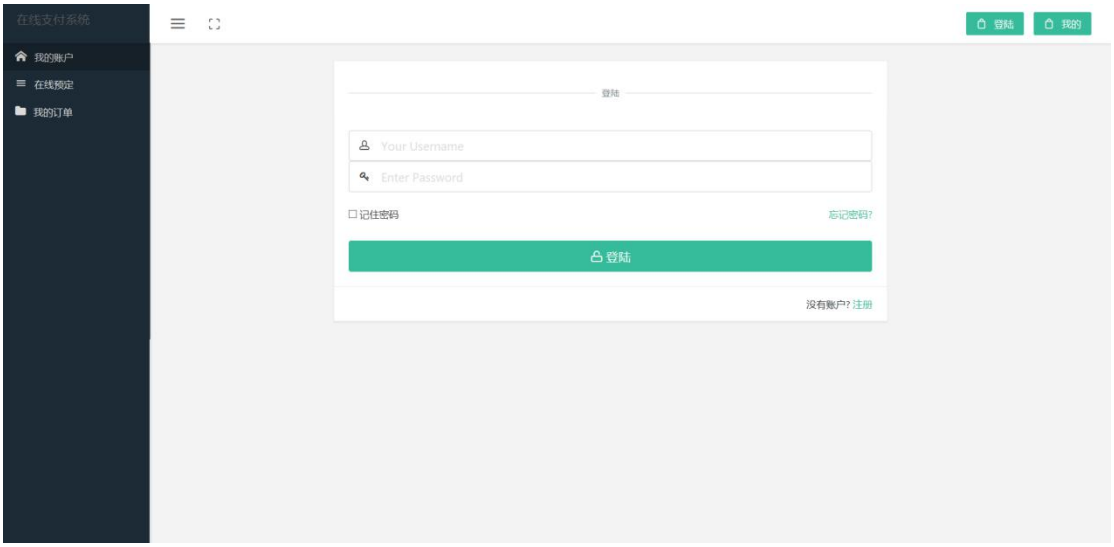
输入数据	预期结果
输入合法数据	返回成功信息，数据库做出相应修改
中断操作	返回错误信息，数据库未修改

5 接口设计

5.1 登录界面

点击右上角登陆按钮可进入，如果点击我的订单时，用户未登录，会自动跳转到登陆界面。

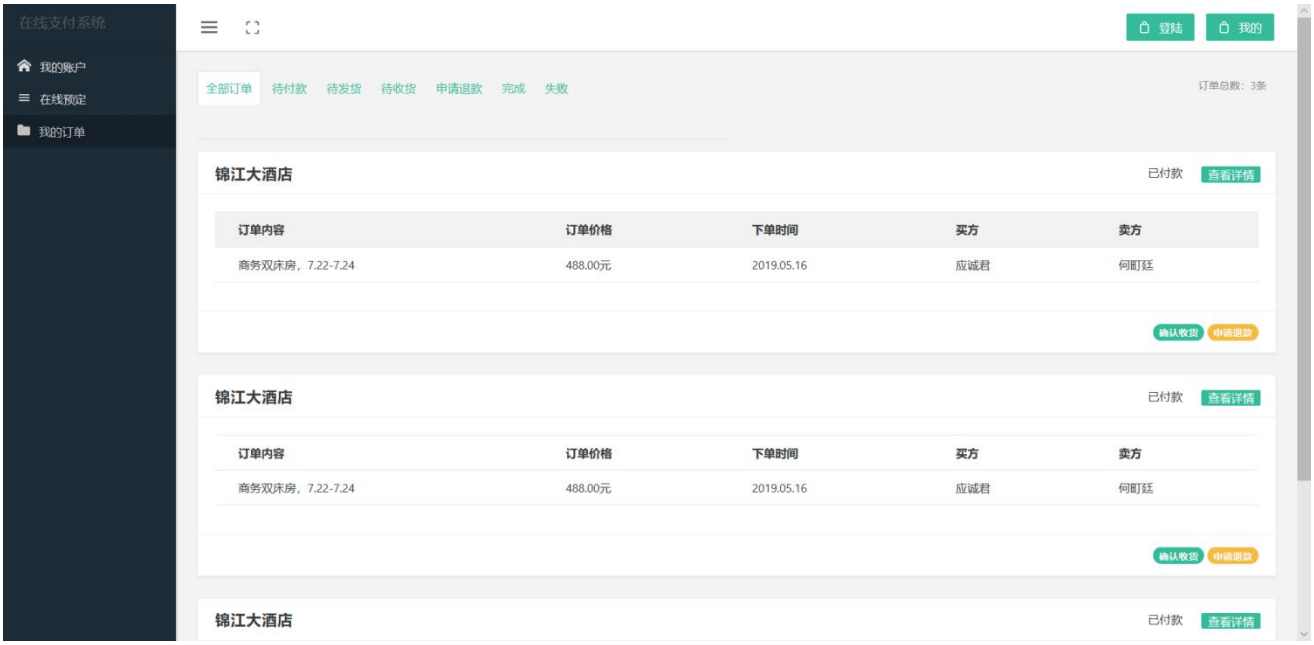
输入用户名、密码后点击“登陆”按钮登录，忘记密码时可以进行找回密码操作，如未注册，也可以跳转至注册界面。



5.2 我的订单界面

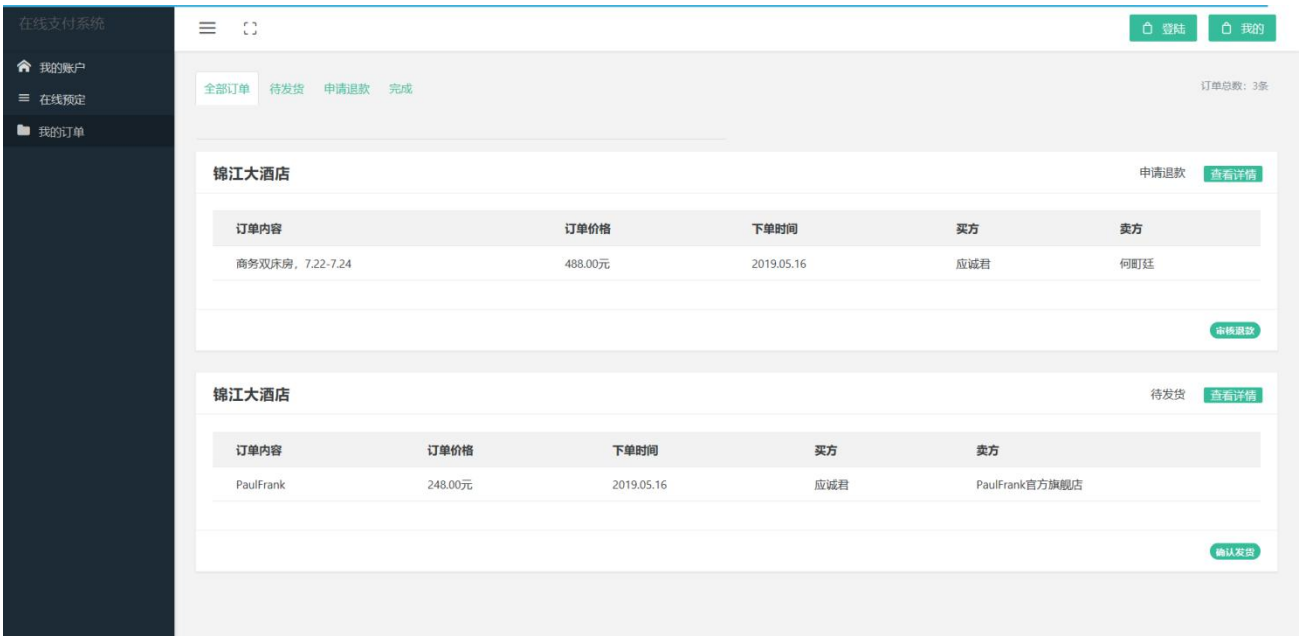
- **买方**

可切换头部导航栏，筛选特定的订单，右边显示当前选项的订单总数。可查看每个订单的详情页面。



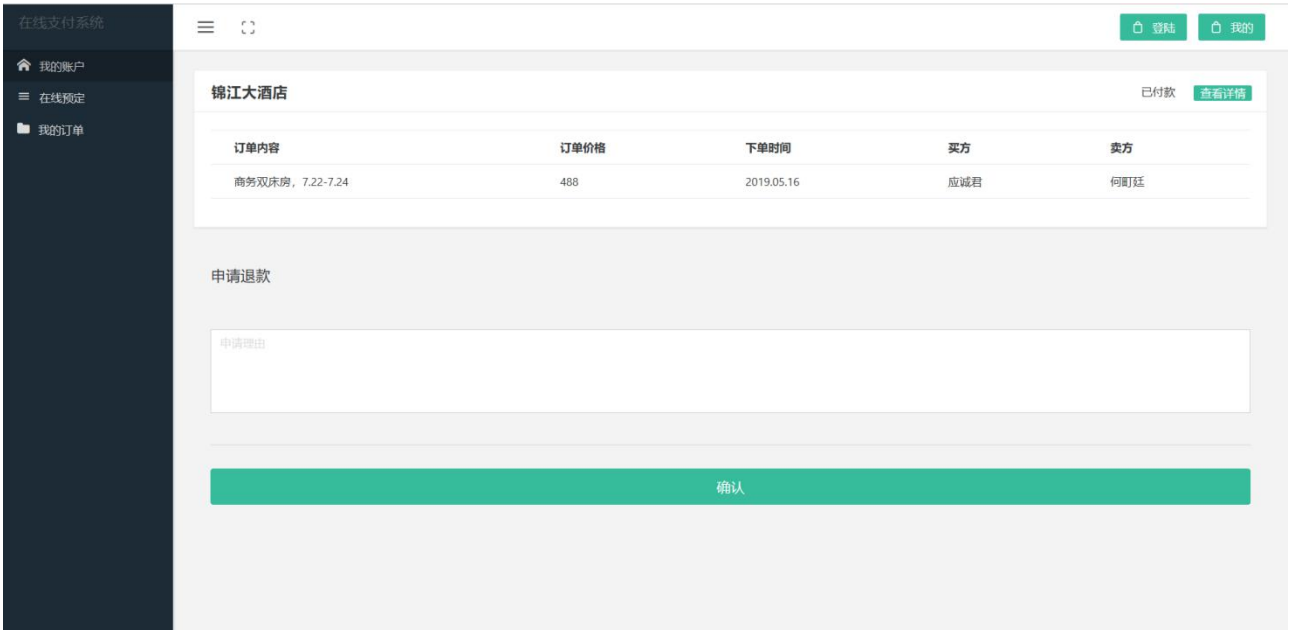
- **卖方**

可对退款申请的订单进行退款审核，对待发货的订单可进行确认发货。



5.3 申请退款

可填写申请退款的理由，确认订单转至申请退款状态。



5.4 外部接口

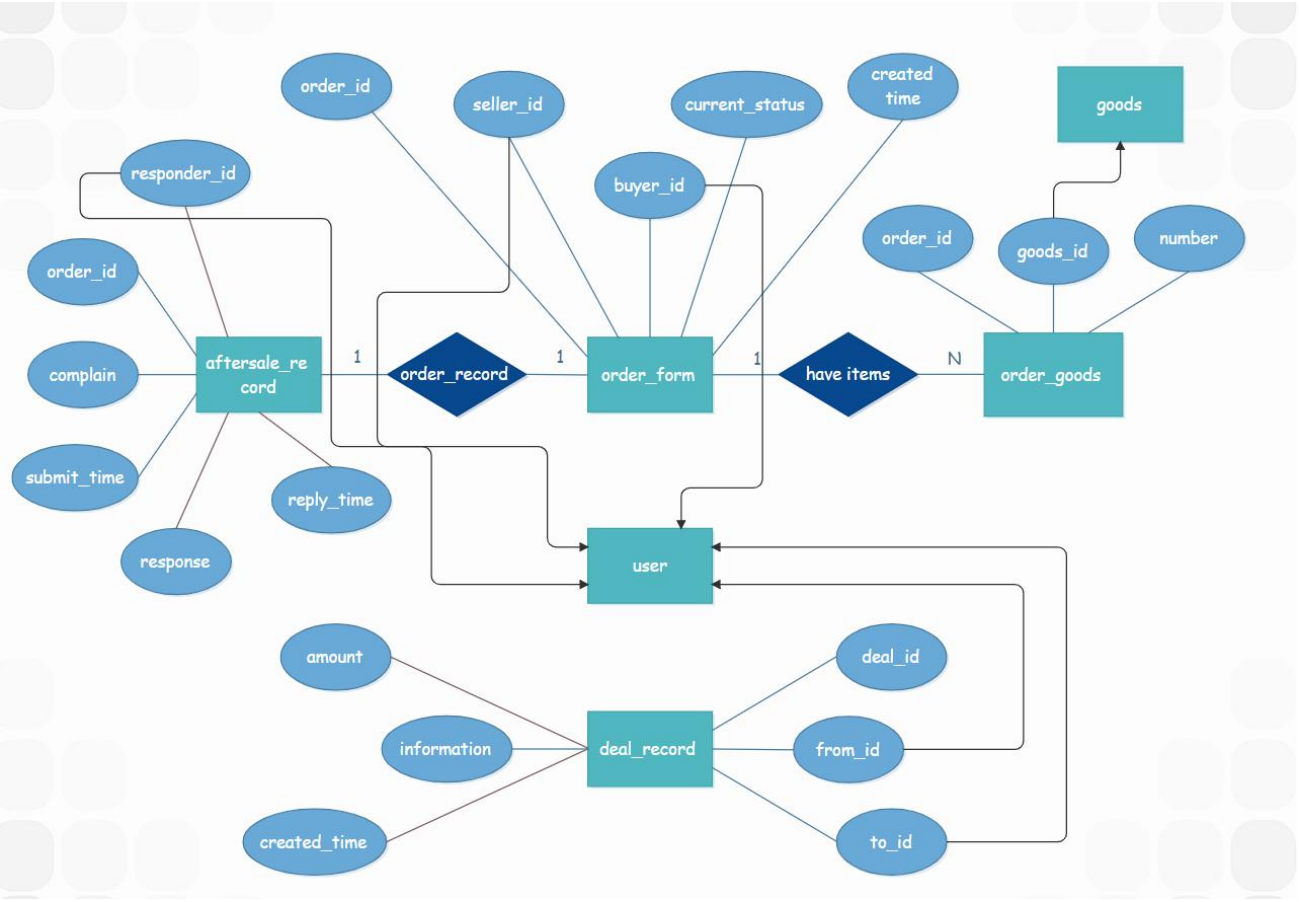
通过 nodejs 创建连接池 connect_pool，通过连接池中的 creatPool 方法，与 mysql 数据库建立连接。

5.5 内部接口

	个人账户管理	在线预定	账户对账和审核	系统管理
付款交易处理	用户个人信息和登陆信息 登陆信息数据库 付款记录数据	订单信息 支付交易	系统一天处理的订单清单信息	用户权限信息

6 数据库设计

6.1 ER 图



6.2 逻辑结构设计

- 订单信息：

```
order_form(order_id, seller_id, buyer_id, current_status, created_time)
```

- 售后记录：记录用户对于订单的投诉理由，系统管理员/商家的反馈结果

```
aftersale_record(order_id, complain, submit_time, response, reponder_id, reply_time)
```

- 订单商品信息:

```
order_goods(order_id, goods_id,number)
```

- 交易流水：

```
deal_record(deal_id, from_id, to_id, amount, information ,created_time)
```

6.3 物理结构设计

6.3.1 订单信息

字段	类型	是否为空	是否主键	备注
order_id	varchar(20)	否	是	订单号，全部由数字字符组成
seller_id	varchar(15)	否	否	卖家唯一标识符

				外键：个人账户管理子系统中的用户表
buyer_id	varchar(15)	否	否	买家唯一标识符 外键：个人账户管理子系统中的用户表
current_status	int(1)	否	否	状态码： [0]待付款 [1]待发货 [2]待收货 [3]已完成 [4]退款中 [5]已关闭
created_time	datetime	否	否	订单创建时间

6.3.2 售后记录

字段	类型	是否为空	是否主键	备注
order_id	varchar(20)	否	是	订单号，全部由数字字符组成 外键：order_form 表中 order_id 字段
complain	varchar(500)	否	否	用户投诉内容
submit_time	datetime	否	否	买家唯一标识符 外键：个人账户管理子系统中的用户表
response	varchar(500)	是	否	管理员或卖家回复内容
reponder_id	varchar(15)	是	否	回复者唯一标识符 外键：个人账户管理子系统中的用户表
reply_time	datetime	是	否	最后更新的回复时间

6.3.3 订单商品信息

字段	类型	是否为空	是否主键	备注
order_id	varchar(20)	否	是	订单号，全部由数字字符组成 外键：order_form 表中 order_id 字段
goods_id	varchar(20)	否	是	商品编号 外键：在线预订子系统中的商品表
number	int	否	否	购买数量

6.3.4 交易流水

字段	类型	是否为空	是否主键	备注
deal_id	varchar(20)	否	是	交易流水号，为交易信息的哈希值，全部由数字字符组成
from_id	varchar(20)	否	否	资金源唯一标识号
to_id	varchar(20)	否	否	资金流向者唯一标识号
amount	int	否	否	交易金额
information	varchar(200)	是	否	交易内容
created_time	datetime	否	否	交易创建时间

7 运行设计

7.1 运行模块组合

本子系统按照功能划分模块，每个模块又按流程划分为客户端界面，客户端脚本，服务 器后台程序。功能模块之间会共享部分界面（导航栏等）。

7.2 运行控制

用户通过处于主界面的导航栏选择功能： 用户登录模块（将在后续调试中使用第一组的模块）：用户在表单中输入正确的账户密码，从而登录系统。根据“卖家”和“买家”两种不同的身份，将拥有不同的模块功能。

- **若登陆者为买家**

查看订单详细信息

查看订单信息，详细信息将包括订单号、商品名称、交易金额、买方的账号名称以及真实姓名、订单状态、操作。订单状态与操作如下：

订单状态	买家操作
待付款	支付、取消订单
待发货	申请退款
待收货	确认收货
已完成	投诉、退款
待退款	无（等待卖家审核）
已关闭	无（退款完成/取消订单后的订单状态）

查询历史交易记录

通过设置筛选与排序条件，根据交易状态（待付款、待发货、待收货、已完成、待退款、已关闭）和时间（今天，上周，上个月，过去三年月，去年，一年前）来筛选历史交易记录进行历史交易查询。

支付

对于“待付款”状态的订单操作，可以单击支付按钮页面跳转至支付页面，支付页面通过订单传递过来的参数，显示订单号、当前账户、目标账户、支付金额等信息。

可以选择支付手段（余额、预付卡），单击支付按钮后，将先判断所选支付方式余额是否满足当前支付金额，若满足，则输入支付密码进行认证。

认证成功将生成交易记录（流水号，关联订单，金额，转出账户，转入账户，交易状态），并将订单信息状态修改为“待发货”，随后页面跳转至订单查询页面。认证失败时，允许用户三次错误输入的机会，三次均失败后将冻结该账户。

取消订单

对于“待付款”状态的订单操作，可以单击取消订单按钮，并在跳出的弹窗进行确认，随后订单信息状态将修改为“已关闭”，页面刷新更新记录。

退款

订单在“待发货“阶段和”已完成“阶段，可以单击操作中的退款按钮，页面会跳转至退款页面，页面会显示当前订单信息、订单状态、对应操作（退款或退款退货），买家输入退款理由，单击确认后，订单状态会被修改为”待退款“。

确认收货

买家点击“待收货”状态的订单的确认收货操作按钮后，订单状态修改为“已完成”。

投诉

买家点击“已完成”状态的订单的投诉操作按钮后，将弹出弹窗，用户输入投诉理由后点击提交，即投诉完成。

● 若登陆者为卖家

查看订单详细信息

查看订单信息，详细信息将包括订单号、商品名称、交易金额、卖方的账号名称以及真实姓名、订单状态、操作。订单状态与操作如下：

订单状态	卖家操作
待付款	无
待发货	发货
待收货	无
已完成	无
待退款	审核
已关闭	无

查询历史交易记录

通过设置筛选与排序条件，根据交易状态（待付款、待发货、待收货、已完成、待退款、已关闭）和时间（今天，上周，上个月，过去三年月，去年，一年前）来筛选历史交易记录进行历史交易查询。

退款审核

卖家点击“待退款”状态的订单的审核操作按钮后，将弹窗显示订单信息、操作、退款理由等，点击同意，将扣除卖家余额，订单状态修改为“已关闭”，并生成交易记录（流水号，关联订单，金额，转出账户，转入账户，交易状态，交易时间）；点击拒绝，退款申请将被拒绝，订单状态回归“已完成”，刷新页面显示更新后订单信息。

发货

卖家点击“待发货”状态的订单的发货操作按钮后，输入订单物流号，随后订单状态修改为“待收货”。

8 系统出错设计

8.1 出错信息

系统输出信息的形式	含义	处理方法
数据库无法连接	①数据库配置出错 ②数据库连接数超过上限	①修改数据库配置 ②限制并发访问量
用户名或密码错误	①用户忘记密码 ②被他人恶意篡改密码	通过登录界面的“找回密码”进行密码修改
服务器暂时无法访问	①服务器正在维护中 ②短时间内有大量流量导致服务器瘫痪	对于②的情况，联系系统管理员进行紧急处理
无法读取磁盘内容	磁盘受损	对磁盘与数据库进行周期性的备份
非法访问	部分用户企图访问管理员界面或后台程序，窃取网站	限制普通用户越权访问，通过各种手段保护后台数据
数据库执行出错	部分用户企图恶意实施 SQL 注入，导致正常查询出错	使用参数绑定的方法进行 SQL 语句构建。对用户填入的信息进行字符串过滤

8.2 补救措施

8.2.1 系统恢复

系统崩溃后，根据系统运行日志恢复系统和数据，并重新启动系统。

8.2.2 定时备份

（1）周期性地备份数据库中的数据，将其存储在更稳定的介质上，并及时进行校对、更新，以避免数据损失之后难以找回。

（2）将工程代码通过 GitHub 进行完善的版本管理。

8.2.3 人工操作

当出现紧急情况时，数据库管理员人工地对数据库中数据进行修改，并做相应的记录。

8.3 系统维护设计

（1）用户在该系统执行操作时应该留下痕迹，以方便检查系统是否被恶意篡改。同时系统管理员定时查看系统日志，统计非法攻击来源和次数，并针对相应攻击加强安全防范措施。

（2）系统管理员的登录不仅需要账户密码，还需要识别 IP，禁止非白名单 IP 的任何访问。

（3）系统维护人员及时更新技术漏洞，通过各种手段防止各种对系统的攻击，增强代码的可靠性。

（4）定期维护数据库，涉及到检查数据库表、检查日志文件等，确保数据库内数据的正确性。

（5）在可能出错的地方使用 try-catch 语句捕获异常，并输出相应的出错信息和可能的处理方法提示。

9 需求回溯

9.1 功能性需求回溯

需求 ID	需求简述	对应的设计模块	实现方式
A01	用户身份有效性核验	身份验证模块	页面加载时，服务器验证 Session 是否过期，如果过期则重定向到所有子系统统一的登录界面。否则用户将根据 Session 中存储的身份信息，拥有不同的模块功能。
A02	为买家提供能够显示所有订单（待付款、待发货、待收货、已完成、退款中、已关闭）的界面	订单查询模块	服务端验证 Session 后确认用户身份为买家时，调用订单查询模块中的函数，通过用户唯一对应的标识码查询用户所有的订单信息。并根据筛选条件进行筛选，同时按照时间先后进行排序，越新的排在越前面。
A03	为卖家提供所有显示卖出订单（待发货、已完成、待处理）的界面	订单查询模块	服务端验证 Session 后确认用户身份为卖家时，调用订单查询模块中的函数，通过用户唯一对应的标识码查询用户所有的订单信息。并根据筛选条件进行筛选，同时按照时间先后进行排序，越新的排在越前面。
A04	为买家提供支付订单和取消订单的操作	订单支付模块	①当用户选择支付订单时，系统先通过支付密码对用户身份进行再次核验，核验完成后将订单流水号提交到后台，由后台进行余额结算、订单状态更新和交易流水创建的操作。 ②当用户选择取消订单时，系统弹出对对话框进行再次确认，用户点击确定取消后，后台对订单状态进行更新。
A05	为买家提供退货退款和投诉的操作	售后处理模块	①用户选择退货时，系统对订单状态修改为待退货，此时卖家可以对退货理由进行审核，审核通过后由后台进行余额结算（将买家账户的资金划到第三方账户中）、订单状态更新和交易流水创建的操作。 ②用户投诉时，系统将投诉理由发送给后台，管理员可以通过后台查询到用户的投诉信息。

A06	为买家对某一订单提供确认收货的操作	订单处理模块	买家在点击确认收货后，系统将弹出对话框进行再次确认，买家确认后，系统会进行余额结算（将第三方账户的资金划到卖家账户中）、订单状态更新和交易流水创建的操作。
A07	为卖家在发货后提供执行确认发货的操作	订单处理模块	卖家在点击确认发货后，系统将弹出对话框进行再次确认，卖家确认后，系统会对订单状态进行更新。

9.2 性能及安全需求

需求 ID	需求简述	实现方式
B01	【安全性】用户账户安全	使用 OAuth2.0 标准协议
B02	【安全性】数据库数据保护，防止用户恶意访问数据库或者破坏数据库。	①使用占位符进行查询参数绑定 ②数据库根据不同的用户赋予不同的最小需要的操作权限。
B03	【性能】能够应对数据库访问量过大的情况以及通过大量访问量来实现攻击的手段	对于频繁访问数据库的操作，需要建立索引，使用 redis 缓存来进行优化
B04	【性能】客户端通过网页展现给用户一个友好、易于操作的界面	通过 BootStrap 和定义外部 CSS 样式表实现扁平化样式与栅格排版，使用 JavaScript 和 JQuery 实现数据的动态显示和交互。