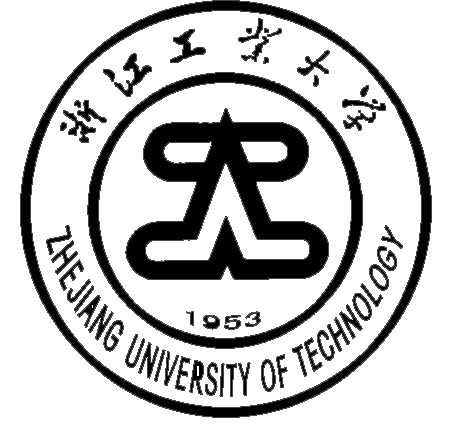


《软件工程》课程作业

模拟期货交易系统项目技术文档

**2023-2024(1)**



项目名称 模拟期货交易系统

组长姓名 A

小组成员 Mcrivers B C D E

指导教师 王婷

所在学院 计算机科学与技术学院

提交日期 2024.1.28

**项目评分表**

|  |  |  |  |
| --- | --- | --- | --- |
| **项目组成员** | **任务分工** | **小组自评** | **教师评分** |
| A（组长） | 后端代码编写，注册和登录逻辑实现，期货数据调整收集，实验报告撰写 | 80 |  |
| Mcrivers | 1. 前端：用户主页、行情页、交易页、订单管理页的设计与实现，登录页面和注册页面的重新设计与实现，所有页面操作逻辑的设计与实现 2. 后端：实体类的设计，获取期货信息控制类、创建委托订单控制类、获取订单信息控制类、获取用户持仓控制类、头像处理控制类的设计与实现，注册控制类、登录控制类的修改，系统整体逻辑的设计 3. 报告撰写：绪论、项目管理计划说明书、用例图的绘制、用例描述的撰写、数据流图的绘制、数据字典的撰写、类图的绘制、顺序图的绘制、协作图的绘制、系统结构图的绘制、测试技术说明书的撰写、用户使用说明住的撰写 4. 其他：前后端接口的对接和测试，alpha测试，beta测试。 | 100 |  |
| C | 后端代码编写，完成了用户资产的增减逻辑和获取用户资产信息，平仓操作，撤销委托订单操作，开仓和更新期货操作的部分实现 | 95 |  |
| D | 完成报告的撰写合并 | 60 |  |
| E | 前端页面设计，做了登陆、注册、订单委托界面 完成ppt 协助完成报告 | 80 |  |
| F | 报告设计与撰写 | 60 |  |
| 项目总分 | | 90 |  |

目录

项目评分表 - 2 -

1. 绪论 - 10 -

1.1. 项目背景 - 10 -

1.2. 项目基本情况 - 11 -

1.2.1. 项目的重要性 - 11 -

1.2.2. 需求和机会 - 12 -

1.2.3. 项目目标 - 12 -

1.2.4. 项目实施 - 13 -

1.2.5. 项目成果 - 13 -

1.3. 同类产品的分析与比较 - 14 -

1.3.1. 分析 - 14 -

1.3.2. 比较 - 16 -

1.4. 项目的应用价值和社会效益 - 16 -

1.4.1. 应用价值 - 16 -

1.4.2. 社会效益 - 17 -

2. 项目管理计划说明书 - 20 -

2.1. 项目概述 - 20 -

2.1.1. 意图、范围和目标 - 20 -

2.1.2. 设想和限制 - 20 -

2.1.3. 可交付项 - 20 -

2.2. 项目组织 - 20 -

2.2.1. 外部接口 - 20 -

2.2.2. 内部结构 - 20 -

2.2.3. 角色和责任 - 21 -

2.3管理过程计划 - 21 -

2.3.1. 需求控制计划 - 21 -

2.3.2. 时间表控制计划 - 21 -

2.3.3. 预算控制计划 - 21 -

2.3.4. 质量控制计划 - 22 -

2.3.5. 风险管理计划 - 22 -

2.4技术过程计划 - 22 -

2.4.1 过程模型 - 22 -

2.4.2 方法、工具和技术 - 22 -

2.4.3 基础设施计划 - 23 -

2.5支持过程计划 - 23 -

2.5.1. 配置管理计划 - 23 -

2.5.2. 测试计划 - 23 -

2.5.3. 归档计划 - 23 -

2.5.4. 质量保证计划 - 24 -

2.5.5. 评审和审计计划 - 24 -

2.5.6. 问题解决计划 - 24 -

2.6. 制定项目计划 - 25 -

2.6.1. 任务列表 - 25 -

2.6.2. 任务依赖关系 - 25 -

2.6.3. 里程碑 - 25 -

2.7. 资源分配 - 26 -

2.7.1. 人员和团队 - 26 -

2.7.2. 资源分配 - 26 -

2.8. 进度管理 - 26 -

2.9. 风险管理 - 27 -

2.9.1. 技术难题 - 27 -

2.9.2. 人员风险 - 27 -

2.9.3. 进度风险 - 27 -

2.9.4. 生成报告 - 28 -

3. 系统需求说明书 - 28 -

3.1. 可行性分析 - 28 -

3.1.1. 经济可行性 - 28 -

3.1.2. 技术可行性 - 28 -

3.1.3. 社会可行性 - 30 -

3.1.4. 风险因素及对策 - 30 -

3.1.5. 可行性分析结论 - 30 -

3.2. 系统需求 - 31 -

3.2.1. 需求分析 - 31 -

3.2.2. 用例图 - 32 -

3.2.3. 用例描述 - 33 -

4. 传统的分析说明书 - 46 -

4.1. 数据流图 - 46 -

4.1.1. 顶层数据流图 - 46 -

4.1.2. 0层数据流图 - 47 -

4.1.3. 1层数据流图 - 48 -

4.1.4. 2层数据流图 - 50 -

4.2. 数据字典 - 52 -

4.2.1. 数据流 - 52 -

4.2.2. 数据存储 - 61 -

4.2.3. 处理过程 - 63 -

5. 面向对象的分析说明书 - 66 -

5.1. 类图 - 66 -

5.2. 顺序图 - 67 -

5.3. 协作图 - 70 -

6. 面向对象的设计说明书 - 73 -

6.1. 系统结构图 - 73 -

6.2. 主要类信息 - 74 -

6.2.1. 实体类 - 74 -

6.2.2. 控制类 - 78 -

6.2.3. 边界类 - 81 -

7. 测试技术说明书 - 83 -

7.1. 等价类划分法 - 83 -

7.1.1. 保证金充值 - 83 -

7.1.2. 平仓交易 - 85 -

7.1.3. 开仓交易 - 87 -

7.2. 边界值分析法 - 90 -

7.2.1. 保证金充值 - 90 -

7.2.2. 平仓交易 - 90 -

7.2.3. 开仓交易 - 92 -

8. 用户使用说明书 - 94 -

8.1. 系统功能概述 - 94 -

8.2. 系统使用说明 - 94 -

8.2.1. 登录 - 94 -

8.2.2. 注册 - 96 -

8.2.3. 上传头像 - 98 -

8.2.4. 查看个人信息 - 99 -

8.2.5. 开仓 - 99 -

8.2.6. 平仓 - 102 -

8.2.7. 查看订单 - 104 -

8.2.8. 撤单 - 105 -

9. 小组项目的体会与心得 - 106 -

9.1. 王睿昊 - 106 -

9.2. 齐昊霖 - 107 -

9.3. 万瑞凯 - 108 -

9.4. 郑庚灿 - 109 -

9.5. 郭增端 - 110 -

参考文献 - 111 -

附录（电子版本材料） - 111 -

附录A 源代码.zip - 111 -

1. 绪论
   1. 项目背景

随着全球金融市场的日益复杂和竞争的加剧，投资者对于期货市场的兴趣不断高涨。期货交易作为金融市场中的一项关键活动，扮演着风险管理、价格发现和投资多样化等重要角色。期货合约允许交易双方在未来的某个时间点以约定价格买入或卖出一定数量的商品或金融工具，这种交易方式涵盖了多个领域，从农产品和原材料到外汇和股指等。期货市场的特点是高风险高回报，因此对于投资者来说，了解期货交易的基本原理、学习市场分析技巧以及掌握交易策略都是至关重要的。

在这个充满机遇和挑战的市场背景下，我们的开发团队决定开发一款创新性的期货交易模拟软件，旨在为广大投资者提供一个安全、实时且教育性的平台。这个决策的背后有着深远的动机和意义。

首先，我们意识到期货市场对于初学者来说可能是一片未知的领域，充满了复杂的术语和交易策略。因此，我们希望打破这一学习壁垒，为那些希望探索期货交易的人提供一个友好和易于理解的入门途径。这就是教育性成为我们项目的第一个目标的原因。

其次，实际参与期货交易可能涉及高额的风险，不适合所有人。我们的团队深知这一点，因此决定提供一种模拟交易的环境，让用户能够在不冒险的情况下体验期货市场的激情和挑战。这个模拟交易平台将使用户能够用虚拟资金进行交易，观察市场波动，制定和优化自己的交易策略，而无需担心资金损失。这一目标的实现将为新手投资者提供一个安全的试验场，有助于培养他们的交易技能，提高自信心。

此外，我们意识到实时市场数据对于投资者做出明智的决策至关重要。因此，我们将提供最新的市场数据，包括价格、成交量、K线图等，以便用户可以在真实市场条件下模拟交易，更好地了解市场动态。

最后，用户友好性和多平台支持也是我们关注的重点。我们希望用户无论使用哪种设备或操作系统，都能轻松访问和使用我们的软件。用户友好的界面将使交易变得简单直观，而多平台支持将确保尽可能多的人能够享受到我们的服务。

总之，期货交易模拟软件项目旨在为投资者提供一种全面的学习和实践期货交易的机会，无论他们是初学者还是有经验的交易者。这个项目的意义不仅在于教育和培训，还在于为用户提供一个安全的环境来积累交易经验，帮助他们在期货市场中取得成功。通过这个软件，我们希望能够为更多的人打开期货市场的大门，助力他们在这个激动人心的领域中取得卓越成就。

* 1. 项目基本情况

### 项目的重要性

期货市场是全球金融体系的重要组成部分，它允许各种类型的交易，包括商品期货、金融期货和外汇期货。这些市场为生产者、消费者和投资者提供了一个有效的风险管理工具，帮助他们锁定未来价格，降低风险，同时也为投机者提供了投资机会。

期货市场的重要性体现在以下几个方面：

1. 价格发现机制： 期货市场允许参与者根据供需关系和市场预期来决定商品或金融工具的未来价格。这有助于确保价格合理和公平，有利于市场的稳定运行。
2. 风险管理： 生产商可以使用期货合约锁定将来的销售价格，从而降低原材料价格波动带来的风险。同样，投资者可以使用期货来对冲其投资组合，减轻市场波动的冲击。
3. 投机机会： 期货市场吸引了投机者，他们试图通过短期交易获得利润。这种活动为市场提供了流动性，增加了市场的效率。
4. 国际贸易支持： 外汇期货市场支持全球贸易，允许跨国公司管理货币风险，促进国际贸易的发展。
5. 投资多样性： 期货市场提供了各种不同资产类别的期货合约，使投资者能够多样化他们的投资组合。

考虑到期货市场的重要性，我们认为提供一个期货交易模拟软件将有助于更多的人了解这个市场、学习交易技能并参与其中。

### 需求和机会

在开始开发期货交易模拟软件之前，我们的团队进行了广泛的市场研究和需求分析。以下是我们发现的一些需求和机会：

1. 教育需求： 很多人对期货交易知之甚少，他们需要一个可靠的平台来学习这个领域的基础知识。教育是成为成功期货交易者的第一步。
2. 模拟交易需求： 新手投资者需要一个安全的环境来进行模拟交易，以便了解市场的运作方式，熟悉交易平台，并制定自己的交易策略。
3. 实时市场数据： 投资者需要实时的市场数据来做出明智的决策。这包括价格、成交量、交易深度等信息。
4. 用户友好性： 软件需要易于使用，用户界面友好，以吸引更多的用户。
5. 多平台支持： 考虑到用户的多样性，支持多种操作系统和设备是至关重要的。

### 项目目标

基于上述需求和机会，我们为期货交易模拟软件项目设定了以下目标：

1. 教育性： 提供丰富的教育资源，包括教程、视频和文章，帮助用户学习期货交易的基本概念和高级技巧。
2. 实时模拟交易： 允许用户在虚拟资金环境中进行实时模拟交易，以实际市场数据为依据，锻炼他们的交易技能。
3. 市场数据： 提供实时市场数据，包括价格、成交量、K线图等，帮助用户进行市场分析和决策。
4. 用户友好性： 开发用户友好的界面，确保用户能够轻松导航和使用软件，即使是没有交易经验的人也能上手。

### 项目实施

为了实现这些目标，我们的开发团队经过精心规划和开发，建立了期货交易模拟软件。以下是我们的一些关键实施步骤：

1. 技术栈选择： 我们选择了先进的技术栈，包括前端和后端技术，以确保软件的性能和稳定性。
2. 实时数据源： 与市场数据提供商合作，获取实时的市场数据，以提供准确的交易环境。
3. 用户界面设计： 我们聘请了经验丰富的UI/UX设计师，设计了直观且吸引人的用户界面，使用户能够轻松进行模拟交易。
4. 教育内容： 开发了大量的教育内容，包括文章、视频和交互式教程，帮助用户学习交易技巧。
5. 安全性和隐私保护： 我们重视用户数据的安全和隐私保护，采取了严格的安全措施。

### 项目成果

经过一段时间的开发和测试，我们成功推出了期货交易模拟软件。以下是一些项目成果的亮点：

1. 教育资源丰富： 我们的软件提供了大量的教育资源，覆盖了期货交易的各个方面，从基础知识到高级策略。
2. 实时模拟交易： 用户可以使用虚拟资金进行实时模拟交易，无需承担实际风险。
3. 真实市场数据： 我们提供了实时的市场数据，帮助用户跟踪市场动态。
4. 用户反馈积极： 我们收到了来自用户的积极反馈，他们表示软件对于他们的期货交易学习和实践非常有帮助。
5. 3. 同类产品的分析与比较

### 分析

1. **产品A - TradingSimulator**

* 功能： TradingSimulator是一款功能强大的期货交易模拟软件。它提供了实时模拟交易、技术分析工具、多种期货合约支持以及交易策略测试功能。用户可以创建虚拟投资组合，使用实时市场数据模拟交易。
* 用户体验： 用户界面直观，易于导航，适合新手。同时，它也提供了高级功能，满足有经验的交易者的需求。
* 教育性： TradingSimulator拥有广泛的教育资源，包括视频教程、文章和在线研讨会。这些资源帮助用户学习期货交易的基础知识和高级策略。
* 实时性： 提供实时市场数据，包括价格、成交量和交易深度。这有助于用户跟踪市场动态。
* 市场数据： 支持多种期货合约，包括商品、外汇和股指。用户可以选择不同市场进行模拟交易。
* 安全性： TradingSimulator采取了强大的安全措施，保护用户数据和资金的安全。
* 可访问性： 提供Web版本和移动应用程序，适用于不同的设备和操作系统。
* 用户支持： 提供在线聊天支持、电子邮件支持和FAQ，以帮助用户解决问题。
* 价格： 提供免费试用版和付费订阅选项，价格相对合理。

1. **产品B - FuturesMaster**

* 功能： FuturesMaster注重交易策略的开发和测试。它提供了强大的回测功能，允许用户测试他们的交易策略，同时还提供实时模拟交易。
* 用户体验： 用户界面简洁，专注于策略开发和测试。适合那些更关注交易策略的用户。
* 教育性： FuturesMaster提供了一些教育内容，但相对较少。主要侧重于交易策略的开发和优化。
* 实时性： 提供实时市场数据，但相对于其他产品来说，它的实时性可能略有滞后。
* 市场数据： 支持多种期货合约，但数量有限。主要关注流行的期货市场。
* 安全性： 有良好的安全性措施，保护用户的数据和资金。
* 可访问性： FuturesMaster提供了Web版本和Windows应用程序，但不支持移动设备。
* 用户支持： 提供电子邮件支持和社区论坛，用户可以在论坛上与其他交易者交流经验。
* 价格： 有限的免费试用版，付费订阅价格较为竞争力。

1. **产品C - 模拟期货交易系统**

* 功能： 模拟期货交易系统是一款专注于实时交易的模拟软件。它提供了快速的模拟交易环境，强调实际市场体验。
* 用户体验： 用户界面简单，适合那些想要快速模拟交易的用户。不提供复杂的分析工具。
* 教育性： 模拟期货交易系统提供基本的教育资源，但不如其他产品的教育性丰富。
* 实时性： 提供真实市场数据，注重快速执行模拟交易。
* 市场数据： 主要关注主要期货市场。
* 安全性： 有基本的安全措施，但相对较少的高级安全特性。
* 可访问性： 提供Web版本应用程序，适用于不同设备和操作系统。
* 用户支持： 提供帮助文档。
* 价格： 提供完全免费版。

### 比较

|  |  |  |  |
| --- | --- | --- | --- |
| **产品** | **TradingSimulator** | **FuturesMaster** | **模拟期货交易系统** |
| 功能 | 全面 | 策略测试 | 实时交易 |
| 用户体验 | 直观、适合新手 | 简介、专注策略 | 简单、快速 |
| 教育性 | 丰富教育资源 | 有限 | 基本 |
| 实时性 | 提供实时市场数据 | 实时模拟 | 提供真实市场数据 |
| 市场数据 | 多种期货合约支持 | 有限 | 多种期货合约支持 |
| 安全性 | 强大安全措施 | 良好 | 基本 |
| 可访问性 | Web和移动应用 | Web和Windows | Web |
| 用户支持 | 在线聊天、Email | Email、社区 | 帮助文档 |
| 价格 | 免费试用版、付费 | 免费试用版、付费 | 完全免费 |

**表1-1 产品比较**

* 1. 项目的应用价值和社会效益



### 应用价值

首先，模拟期货交易系统通过提供实时市场数据，为初学者创造了一个真实交易环境的模拟体验。这种体验使得新手投资者能够在没有真实金融风险的情况下学习和磨练交易技能，不仅增加了其实战经验，也降低了进入金融市场的初始门槛。

其次，平台注重交易数据的可视化功能，使交易者能够实时查看模拟交易市场数据的变化，包括期货价格和交易量。数据的即时更新实现一秒一变，让交易者能够从平台上获取相对实时的阶段性成果。这有助于他们及时发现问题、应对策略变更，从而避免不必要的资金浪费。交易成交价格通过直观的走势图呈现，并且交易者可以根据自己的需要选择绝对时间的流动与否。在交易价格变动过快的情况下，交易者还可自主暂停，并及时进行买卖操作来进行止损。

实践交易机会与风险管理：提供实战体验： 模拟交易系统为投资者提供了在真实市场环境中进行模拟交易的机会，使其能够在没有实际金融风险的情况下学习和磨练交易技能，增强实际操作的信心和能力。风险管理与策略验证： 初学者可以通过系统模拟不同的交易策略，验证其在实际市场中的效果，从而学习如何科学合理地进行风险管理，降低投资风险。模拟交易系统不仅为投资者提供实践机会，还通过交易决策支持、培训资料等方式普及金融知识，提高投资者的金融素养和市场洞察力。

初学者友好：系统以简洁直观的界面和易操作的功能设计，为初学者提供友好的学习环境，降低了金融市场对新手的学习门槛，促使更多人参与到金融领域。

总的来说，模拟交易系统通过提供实践交易机会、教育培训、投资策略优化、实时市场观察和投资者社区等多方面的功能，为投资者提供了全面的应用价值。这不仅有助于个体投资者的职业发展，还促进了整个金融市场的稳健发展。这一全面的应用价值使模拟期货交易系统成为投资者学习、磨练和实践的理想平台。

### 社会效益

1. 提升市场透明度：

模拟交易系统通过为投资者提供实时市场数据、交易记录等信息，积极促进金融市场的透明度。这一举措有助于减少信息不对称，提高市场运作的效率，从而塑造更加公平公正的交易环境，使市场参与者能够更清晰地了解市场动态。

1. 培养理性投资观念：

通过模拟交易系统，投资者得以培养理性的投资观念，降低盲目跟风和冲动交易的发生。这有益于培养投资者更为理性的决策思维，减缓市场过度波动，从而促进金融市场的稳健发展。

1. 培养专业人才：

作为一个培训平台，模拟交易系统有助于培养更多具备实际操作经验的金融专业人才。这对金融行业的长期发展至关重要，有助于提高金融从业人员整体素质，推动金融服务水平的提升。

1. 解决人才短缺问题：

通过模拟交易系统的人才培训功能，有助于缓解金融行业人才短缺的问题，为金融领域提供更充足的人才储备。这不仅为行业注入新鲜血液，还有助于金融产业的可持续发展。

社会稳定与经济效益：

1. 降低投资风险：

投资者通过模拟交易系统学到的投资知识和技能，能够更好地理解市场规律，降低投资风险。这有助于减少由于投资失败导致的财务风险，提高社会整体的经济效益。

1. 推动创业与创新：

模拟交易系统通过提供实践机会，激发投资者的创新意识，促进更多的创业活力。这有助于推动社会经济的发展，增加就业机会，推动创新产业的崛起。

1. 投资者权益保护：

建立诚信环境：模拟交易系统通过提供透明、公正、安全的交易环境，有助于建立起一个诚信的金融社会环境。这对于保护投资者的权益、提高金融从业人员的职业操守具有积极的社会效益。

1. 减少非法交易：

提供安全的投资平台和严格的监管，有助于减少非法交易的发生。这有助于维护金融市场的秩序，增强社会对金融体系的信任感。

总体而言，模拟交易系统通过为投资者提供培训机会、提高金融人才素质、促进创业与创新、保护投资者权益等多方面产生了广泛的社会效益，对于金融市场和整个社会的发展具有重要意义。这种综合效益不仅对个体投资者有益，也有助于整个社会构建更加健康、稳定的金融生态系统。

1. 项目管理计划说明书
   1. 项目概述

### 意图、范围和目标

该项目旨在创建一个面向期货交易新手的期货交易模拟系统，涵盖用户个人资料管理、期货交易、订单信息查询和个人资产查询等服务。项目的主要目标是提供一站式解决方案，满足期货交易员的各种需求。

### 设想和限制

项目设想为期货交易者提供方便、高效的学习机会，但同时受到资源、法规和技术等方面的限制。

### 可交付项

* 模拟期货交易系统
  1. 项目组织

### 外部接口

与供应商、合作伙伴和其他项目之间的接口和交互。

### 内部结构

本系统由多个模块组成，包括注册登录、期货交易、订单查询、宠物培训、宠物寄养、信息管理等多个模块，由前后端多位同学协作完成。

### 角色和责任

* 小组组长：负责项目的规划。
* 开发人员：负责系统开发和维护。
* 测试人员：负责对系统进行测试和验证。

2.3管理过程计划



### 需求控制计划

制定需求变更的控制计划，以确保变更的合理性和影响。

* 变更流程：引入变更控制委员会，审查和评估所有需求变更。
* 责任分配：明确变更的责任人，包括变更的提出者、评估者和批准者。

### 时间表控制计划

描述如何监控和调整项目的时间表。

* 进度追踪：使用甘特图和里程碑来追踪项目进度。定期举行进展会议，检查任务完成情况。
* 延迟问题解决：建立早期警报系统，及时发现和解决延迟问题。设立补救计划，以缩短延期时间。

### 预算控制计划

说明项目预算的监控和控制方法。

* 费用追踪系统： 使用项目管理工具来追踪实际费用与预算的差距。确保项目在预算范围内运作。
* 审查和调整： 定期审查项目的经费使用情况，并根据需要调整预算分配。

### 质量控制计划

描述如何确保项目交付物的质量。

* 质量标准： 制定明确的质量标准，包括代码质量、文档准确性等。
* 测试流程： 实施全面的测试流程，包括单元测试、集成测试和系统测试，以确保交付物的质量。

### 风险管理计划

概述如何识别、评估和应对项目风险的计划。

* 风险识别： 每两周召开风险评估会议，收集团队成员的意见，使用头脑风暴等方法，识别可能的风险。
* 风险评估： 使用定性和定量分析方法，评估每个风险的概率

2.4技术过程计划

### 2.4.1 过程模型

项目采用敏捷开发模型，以确保对需求变更的灵活响应和高质量的交付。敏捷方法将项目划分为短期迭代，每个迭代的周期通常在2至4周之间。每个迭代结束后都会交付一部分可用的产品，同时根据用户反馈进行迭代。

### 2.4.2 方法、工具和技术

* 方法：采用面向对象分析和设计方法，以便更好地理解和建模系统的结构和功能。
* 工具：使用Jira作为项目管理和问题跟踪工具，Git用于版本控制，以确保团队有效协作和代码管理。
* 技术：选择使用Vue作为前端框架，SpringBoot作为后端框架，以构建现代化的、可扩展的期货交易模拟系统。

### 2.4.3 基础设施计划

为了支持项目的开发和测试，我们需要以下基础设施：

* 硬件： 开发人员的工作站、服务器用于部署和测试。
* 软件： 集成开发环境（IDE）、数据库管理系统、操作系统。
* 网络： 确保团队成员之间的良好通信，使用云服务实现团队协作。

2.5支持过程计划



### 配置管理计划

配置管理计划将确保项目的配置项被正确标识、版本化和追踪。具体步骤包括：

* 配置标识： 为每个配置项分配唯一标识符。
* 版本控制： 使用Git进行版本控制，确保每个版本的可追溯性。
* 配置审查： 定期进行配置审查，以确保配置项的正确性和一致性。

### 测试计划

测试计划将确保系统的质量和稳定性。主要步骤包括：

* 测试策略： 确定测试的整体策略和方法。
* 单元测试： 针对每个单元进行测试，确保模块的正确性。
* 集成测试： 测试不同模块之间的集成，确保它们协同工作。
* 系统测试： 对整个系统进行综合测试，确保满足用户需求。

### 归档计划

归档计划将确保项目文档得到妥善管理和保存。具体步骤包括：

* 文档标准： 制定文档标准，确保一致的格式和结构。
* 文档存档： 将项目文档存档至云端或其他可靠的存储设备。
* 访问控制： 设立访问权限，确保只有授权人员可以访问和修改文档。

### 质量保证计划

* 质量标准： 制定明确的质量标准，包括代码规范、性能标准、用户体验等。
* 过程审查： 定期进行过程审查，评估团队是否遵循最佳实践和流程。
* 度量和分析： 收集质量度量数据，分析趋势，及时发现潜在问题并采取纠正措施。
* 培训计划： 提供质量相关的培训，确保团队成员了解并遵循质量标准。

### 评审和审计计划

确保项目在不同阶段进行评审和审计，以提高质量和合规性。

* 阶段评审： 在关键阶段结束时进行内部评审，确保符合项目标准。
* 外部审计： 定期邀请外部专家进行独立审计，确保项目合规性。

### 问题解决计划

确保项目中出现的问题得到及时而有效的解决。

* 问题识别： 建立问题识别机制，通过团队会议和工具收集问题。
* 问题记录： 每个问题都将记录在问题跟踪系统中，包括问题的描述、优先级和解决状态。
* 解决方案： 团队将定期召开问题解决会议，共同研究和制定解决方案。
  1. 制定项目计划

### 任务列表



**表2-1 任务列表**

### 任务依赖关系

1. “软件设计”任务的开始依赖于“需求分析”任务的完成。
2. “需求分析”和“软件设计”任务的结束依赖于“系统设计”的完成
3. “软件实现与测试”任务的开始依赖于“软件设计”任务的完成

### 里程碑



**表2-2 项目里程碑**

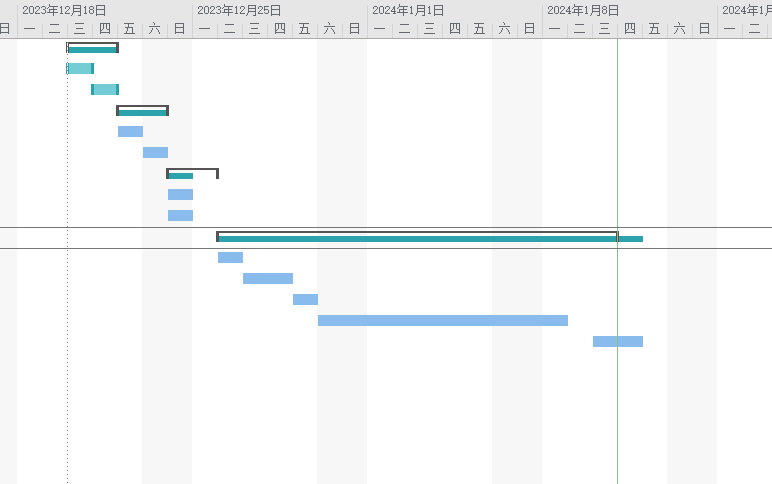
* 1. 资源分配

### 人员和团队

王睿昊，郭增端，齐昊霖，聂梦召，万瑞凯，郑庚灿

### 资源分配

* 系统设计：王睿昊，郭增端，齐昊霖，聂梦召，万瑞凯，郑庚灿
* 需求分析：王睿昊，郭增端，齐昊霖，聂梦召，万瑞凯，郑庚灿
* 软件设计：王睿昊，郭增端，齐昊霖，聂梦召，万瑞凯，郑庚灿
* 软件实现与设计：
  + 前端部分：王睿昊，郑庚灿
  + 后端部分：王睿昊，齐昊霖，万瑞凯。
  1. 进度管理



**图2-1 甘特图**

* 1. 风险管理

### 技术难题

* 需要使用Vue与SpringBoot等新的软件知识，可能导致进度延迟。
* 依赖于第三方工具或库：使用的VScode和IDEA可能存在不稳定性，不支持的问题，影响项目的稳定性和可维护性。

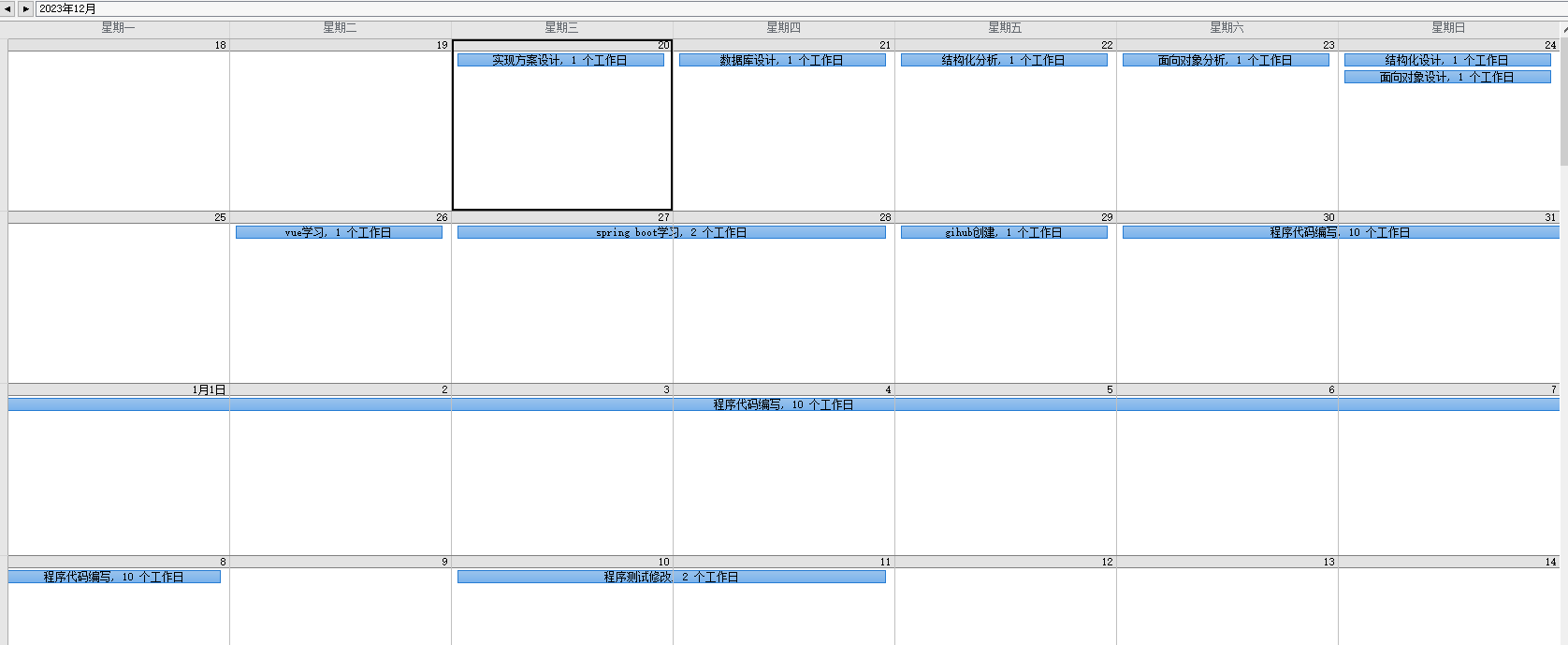
### 人员风险

* 关键团队成员离职：关键团队成员的日常生活琐事可能导致项目知识损失和进度延迟。
* 团队协作问题：沟通不畅、合作问题或团队成员之间的不和谐可能导致项目交付困难。

### 进度风险

* 需求变更：频繁的需求变更与开始的需求分析不明确可能导致进度滞后和额外的开发工作。
* 任务估算不准确：任务估算不足可能导致无法按时完成项目。

### 生成报告



**图2-2 进度报告图**

1. 系统需求说明书
   1. 可行性分析

### 经济可行性

当前金融市场对于投资者的需求不断增长，模拟交易系统满足了广大投资者对于实践操作和风险管理的需求。在金融科技不断发展的趋势下，提供一个可靠、实用的模拟交易平台具备巨大市场潜力。尤其对于初学者，提供零风险的学习平台能够促进更多人参与金融市场，从而刺激金融行业的发展。

### 技术可行性

本平台采用SpringBoot+VUE架构，方便客户的浏览、使用。我们根据J2EE三层结构，分为表现层、业务逻辑层和数据服务层，该分层结构，提高了软件的后期可维护性。三层结构以springboot作为支持，确保网站平台的高效、稳定、安全。

SpringBoot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。Spring Boot是一种基于Spring框架的快速应用程序开发工具，它帮助开发人员快速创建和部署应用程序，而不必编写大量的配置文件和样板代码。Spring Boot为开发人员提供了一种简单且快速的方式来创建和部署高度可扩展和可维护的应用程序。Spring Boot是一个开源框架，它提供了许多工具和组件，以简化Spring应用程序的开发和部署。Spring Boot是基于Spring框架构建的，它使用了Spring的核心功能，如依赖注入和面向切面编程（AOP），并且添加了一些新的功能来简化开发过程、

Vue 是一款用于构建用户界面的 JavaScript 渐进式（即主张最少，每个框架都不可避免会有自己的一些特点，从而会对使用者有一定的要求，这些要求就是主张，主张有强有弱，它的强势程度会影响在业务开发中的使用方式）框架，是MVVM模型的一种应用。它基于标准 HTML、CSS 和 JavaScript 构建，并提供了一套声明式的、组件化的编程模型，帮助使用人员高效地开发用户界面。

与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用，Vue 的核心库只关注视图层。

MVVM 其实表示的是 View-ViewModel-Model，就是视图层-视图模型层-模型层。View是作为视图层，简单来说可以把它理解为HTML页面；Model 是作为模型层，它是负责处理业务逻辑以及和服务器端进行交互的；ViewModel 是作为视图模型层，也就是 Vue 框架所起到的作用了，主要是作为 View 层和 Model 层之间的通信桥梁。本平台使用sqlserver中的SQL Server 版作为数据库，它较之前版本推出了许多新的特性和关键的改进，使得它成为至今为止的最强大和最全面的SQL Server版本。SQL Server 2008出现在微软数据平台愿景上是因为它使得公司可以运行他们最关键任务的应用程序，同时降低了管理数据基础设施和发送观察和信息给所有用户的成本。我们小组成员都有过SQL Server 2008的学习经验，能够熟练使用SQL语言，有过SQL Server 2008上建数据库、建表、触发器编写等经验。

### 社会可行性

随着金融市场对于普通投资者的开放，模拟交易系统有助于提高投资者的金融素养，促进金融市场的健康发展。系统为投资者提供了一个学习和交流的平台，形成了一个学习共同体，有助于培养更多具备实际操作经验的金融初学者。此外，系统的推出还有助于社会稳定，通过风险管理和投资者教育，减少了非法交易的可能性，提高了整个金融市场的透明度。

### 风险因素及对策

在模拟交易系统的实施过程中，可能面临技术风险、市场风险以及用户接受度等方面的风险。为应对这些风险，系统提供了识别和管理风险的流程，建立了透明的风险应对机制，以确保项目的顺利实施。此外，根据用户反馈和市场变化，及时进行系统升级和优化，以提高系统的稳定性和用户体验。

### 可行性分析结论

综合考虑经济、技术、社会以及风险因素，模拟交易系统的设计方案是可行的。系统有望为入门投资者提供一个实践交易的平台，促进金融市场的健康发展，培养更多的金融人才进行入门，同时通过风险管理和投资者教育，减少非法交易，提高整个金融市场的透明度。

* 1. 系统需求

### 需求分析

1. 交易终端模块：
2. 用户界面设计：
   * + - 设计直观友好的界面，支持多PC访问。
       - 提供个性化设置，允许用户自定义界面布局和主题。
3. 用户注册和登录：
   * + - 实现安全的用户注册和登录功能
4. 实时行情数据：
   * + - 提供期货合约的真实行情数据，包括K线图、深度图等。
5. 模拟交易：
   * + - 支持用户进行模拟交易，包括下单、撤单等操作。
       - 提供虚拟资金和实时盈亏展示。
6. 订单服务模块：
7. 撤销和修改：
   * + - 允许用户撤销和修改订单，考虑交易规则的限制。
8. 实时成交和结算：
   * + - 实现订单的实时成交和结算，包括手续费和保证金计算。
9. 订单查询：
   * + - 提供用户查询当前订单的功能。
       - 提供用户查询历史订单的功能。

### 用例图



**图3-1 用例图**

### 用例描述

|  |  |
| --- | --- |
| **用例名称** | 注册 |
| **用例描述** | 用户为自己注册新账号 |
| **参与者** | 用户 |
| **基本路径** | 1.用户点击注册  2.跳转到注册页面  3.注册人填写注册信息  4.注册成功 |
| **拓展路径** | 注册人填写信息无效，注册失败 |

**表3-1 注册用例描述**

|  |  |
| --- | --- |
| **用例名称** | 登录 |
| **用例描述** | 用户登录自己的账号 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户点击登录 2. 填写账号密码 3. 系统显示登录成功并跳转界面 |
| **拓展路径** | 填写的账户或密码不匹配，登录失败 |

**表3-2 登录用例描述**

|  |  |
| --- | --- |
| **用例名称** | 个人信息管理 |
| **用例描述** | 用户填写、删除、更改自己的个人信息 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入个人信息界面 2. 用户选择需要新增或更改的个人信息 3. 用户填写新增或更改后的个人信息 4. 用户点击“保存”按钮保存个人信息 5. 系统显示“保存成功”并退出个人信息编辑页面 |
| **拓展路径** | 用户输入的个人信息不符合格式要求，新增或更改个人信息的操作失败 |

**表3-3 个人信息管理用例描述**

|  |  |
| --- | --- |
| **用例名称** | 资产管理 |
| **用例描述** | 用户管理自己的资产，包括充值和提现 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入个人信息页面 2. 用户点击“充值”按钮充值交易保证金或点击“提现”按钮提取交易保证金 3. 系统显示“充值保证金成功”或“提现保证金失败” |
| **拓展路径** | * 1. 用户当前保证金数额减去希望提现的保证金数额后小于总资产价值的10%，系统提示“提现失败”   2. 用户点击“充值”按钮后并没有向系统支付金额，系统提示“充值保证金失败” |

**表3-4 资产管理用例描述**

|  |  |
| --- | --- |
| **用例名称** | 充值 |
| **用例描述** | 用户向自己的账户中充值保证金 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入个人信息页面 2. 用户点击“充值”按钮充值交易保证金 3. 系统显示“充值保证金成功” |
| **拓展路径** | 用户点击“充值”按钮后并没有向系统支付现金，系统提示“充值保证金失败” |

**表3-5 充值用例描述**

|  |  |
| --- | --- |
| **用例名称** | 提现 |
| **用例描述** | 用户从自己的账户中提取保证金 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入个人信息页面 2. 用户点击“提现”按钮提取交易保证金 3. 系统显示“充值保证金成功”或“提现保证金失败” |
| **拓展路径** | 用户当前保证金数额减去希望提现的保证金数额后小于总资产价值的10%，系统提示“提现保证金失败” |

**表3-6 提现用例描述**

|  |  |
| --- | --- |
| **用例名称** | 客户支持 |
| **用例描述** | 用户寻求系统使用方面的帮助或反馈对系统的意见 |
| **参与者** | 用户、客服 |
| **基本路径** | 1. 用户在个人主页点击“客户支持”按钮，进入“客户支持”界面 2. 用户选择需要的客户支持功能 3. 用户获得了系统提供的相应客户支持功能 4. 用户退出“客户支持”界面 |
| **拓展路径** | 客户在寻求客服帮助时，出现了通讯故障或客户长时间不在线，导致客户无法很快地获取帮助 |

**表3-7 客户支持用例描述**

|  |  |
| --- | --- |
| **用例名称** | 反馈意见 |
| **用例描述** | 用户向系统开发方反馈系统存在的Bug或提出自己的改进意见 |
| **参与者** | 用户、客服 |
| **基本路径** | 1. 用户在个人主页点击“客户支持”按钮，进入“客户支持”界面 2. 用户选择“反馈意见”功能，进入反馈信箱界面 3. 用户填写想要反馈的系统Bug或希望系统改进的部分 4. 用户点击“提交”按钮提交反馈意见 |
| **拓展路径** | 用户填写的内容为空或超出最大字数限制，提交反馈信息失败 |

**表3-8 反馈意见用例描述**

|  |  |
| --- | --- |
| **用例名称** | 寻求客服帮助 |
| **用例描述** | 用户寻求系统使用方面的帮助 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户在个人主页点击“客户支持”按钮，进入“客户支持”界面 2. 用户选择“寻求客服帮助”界面 3. 用户在和客服交流的过程中成功解决了系统使用方面的问题 4. 用户退出“客户支持”界面 |
| **拓展路径** | 客户在寻求客服帮助时，出现了通讯故障或客户长时间不在线，导致客户无法很快地获取帮助 |

**表3-9 寻求客服帮助用例描述**

|  |  |
| --- | --- |
| **用例名称** | 找回密码 |
| **用例描述** | 用户忘记了账户密码，需要重设账户密码来登录账号 |
| **参与者** | 用户、客服 |
| **基本路径** | 1. 用户进入登录界面 2. 用户点击“找回密码”按钮 3. 进入找回密码界面 4. 用户输入用户名和预设的密保问题答案 5. 系统对用户名和密保问题进行验证 6. 如果验证通过，用户输入新密码 7. 密码找回成功，系统回到登录界面 |
| **拓展路径** | 系统验证用户名和密保问题答案失败，用户重新输入密保问题答案 |

**表3-10 找回密码用例描述**

|  |  |
| --- | --- |
| **用例名称** | 用户名密码登录 |
| **用例描述** | 用户通过输入用户名和密码实现登录 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入登录界面 2. 用户在输入框中分别输入用户名和密码 3. 用户点击登录按钮 4. 系统验证用户名和密码是否匹配 5. 如果验证通过，则用户登录成功，用户成功进入系统 |
| **拓展路径** | 系统验证用户名和密码不匹配，用户登录失败，需要用户重新输入用户名和密码 |

**表3-11 用户名密码用例描述**

|  |  |
| --- | --- |
| **用例名称** | 扫码登录 |
| **用例描述** | 用户通过输入用户名和密码实现登录 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入登录界面 2. 用户扫描登录二维码 3. 系统验证用户身份 4. 如果验证通过，则用户登录成功，用户成功进入系统 |
| **拓展路径** | 系统验证用户身份失败，用户登录失败，需要用户重新扫码或使用其他方式登录 |

**表3-12 扫码登录用例描述**

|  |  |
| --- | --- |
| **用例名称** | 期货交易 |
| **用例描述** | 用户进行期货交易操作 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入开仓交易界面或平仓交易界面 2. 用户调整交易参数 3. 用户点击确认交易按钮 4. 系统验证用户的交易参数是否合法 5. 如果合法，则提示交易成功 |
| **拓展路径** | 系统判断用户提交的交易订单不合法，期货交易操作失败 |

**表3-13 期货交易用例描述**

|  |  |
| --- | --- |
| **用例名称** | 开仓交易 |
| **用例描述** | 用户进行开仓交易操作 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入期货信息详情界面 2. 用户在输入框中输入委托价格和委托数量 3. 用户点击交易按钮 4. 系统判断用户提交的交易订单是否合法 5. 如果合法，则提示“您的订单已创建成功”，并建立相应的委托订单 |
| **拓展路径** | 系统判断用户提交的交易订单不合法，委托订单创建失败 |

**表3-14 开仓交易用例描述**

|  |  |
| --- | --- |
| **用例名称** | 平仓交易 |
| **用例描述** | 用户进行平仓交易操作 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入个人主页 2. 用户在期货持仓表中选择需要进行平仓操作的期货合约 3. 用户在数据框中输入需要平仓的期货数量 4. 用户点击平仓按钮 5. 系统判断用户的平仓交易参数是否合法 6. 如果平仓操作合法，则提示“平仓交易成功” |
| **拓展路径** | 用户输入的平仓交易参数不合法，操作失败 |

**表3-15 平仓交易用例描述**

|  |  |
| --- | --- |
| **用例名称** | 买入 |
| **用例描述** | 用户进行买开或买平操作 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入期货信息详情界面或用户信息界面 2. 用户在输入框中输入买开参数或买平参数 3. 用户点击提交按钮 4. 系统判断用户的交易参数是否合法 5. 如果操作合法，则提示“交易成功” |
| **拓展路径** | 用户输入的交易参数不合法，操作失败 |

**表3-16 买入用例描述**

|  |  |
| --- | --- |
| **用例名称** | 卖出 |
| **用例描述** | 用户进行卖开或卖平操作 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入期货信息详情界面或用户信息界面 2. 用户在输入框中输入买开参数或买平参数 3. 用户点击提交按钮 4. 系统判断用户的交易参数是否合法 5. 如果操作合法，则提示“交易成功” |
| **拓展路径** | 用户输入的交易参数不合法，操作失败 |

**表3-17 卖出用例描述**

|  |  |
| --- | --- |
| **用例名称** | 期货行情查询 |
| **用例描述** | 用户查询期货的行情信息 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入期货信息界面 2. 用户选择希望查看行情信息的期货 3. 用户进入期货行情详情界面 |
| **拓展路径** | 无 |

**表3-18期货行情用例描述**

|  |  |
| --- | --- |
| **用例名称** | 所有委托订单查询 |
| **用例描述** | 用户查询所有委托订单 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入订单管理界面 2. 用户选择需要查看的订单类型，包括当前订单和历史订单 |
| **拓展路径** | 无 |

**表3-19 所有委托订单查询用例描述**

|  |  |
| --- | --- |
| **用例名称** | 特定期货委托订单查询 |
| **用例描述** | 用户查看特定期货的委托订单 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入期货信息详情界面或用户信息界面 2. 用户在页面最下方选择需要查看的订单类型，包括当前订单和历史订单 |
| **拓展路径** | 无 |

**表3-20 特定期货委托订单查询用例描述**

|  |  |
| --- | --- |
| **用例名称** | 持仓信息查询 |
| **用例描述** | 用户查询自己的持仓信息 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入用户信息界面 2. 用户查看自己的持仓信息 |
| **拓展路径** | 无 |

**表3-21 持仓信息查询用例描述**

|  |  |
| --- | --- |
| **用例名称** | 资产盈亏查询 |
| **用例描述** | 用户查询资产盈亏 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入用户信息界面 2. 用户查看资产盈亏 |
| **拓展路径** | 无 |

**表3-22 资产盈亏查询用例描述**

|  |  |
| --- | --- |
| **用例名称** | 信息查询 |
| **用例描述** | 用户查询各类信息 |
| **参与者** | 用户 |
| **基本路径** | 用户查询信息 |
| **拓展路径** | 无 |

**表3-23 信息查询用例描述**

|  |  |
| --- | --- |
| **用例名称** | 发送交易达成通知 |
| **用例描述** | 用户的委托订单达成，系统向用户发送交易达成通知 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户的委托订单达成 2. 系统向用户发送交易达成通知 3. 用户查收交易达成通知 |
| **拓展路径** | 无 |

**表3-24 发送交易达成通知用例描述**

|  |  |
| --- | --- |
| **用例名称** | 发送价格异动通知 |
| **用例描述** | 某个期货在短时间内价格大幅变化，系统向持有该期货的用户发送价格异动通知 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户持有的期货在短时间内价格发生大幅度变动 2. 系统向用户发送价格异动通知 3. 用户查收通知 |
| **拓展路径** | 无 |

**表3-25 发送价格异动通知用例描述**

|  |  |
| --- | --- |
| **用例名称** | 发送通知 |
| **用例描述** | 系统发送通知 |
| **参与者** | 用户 |
| **基本路径** | 1. 系统因为各种原因发送通知 2. 用户查收通知 |
| **拓展路径** | 无 |

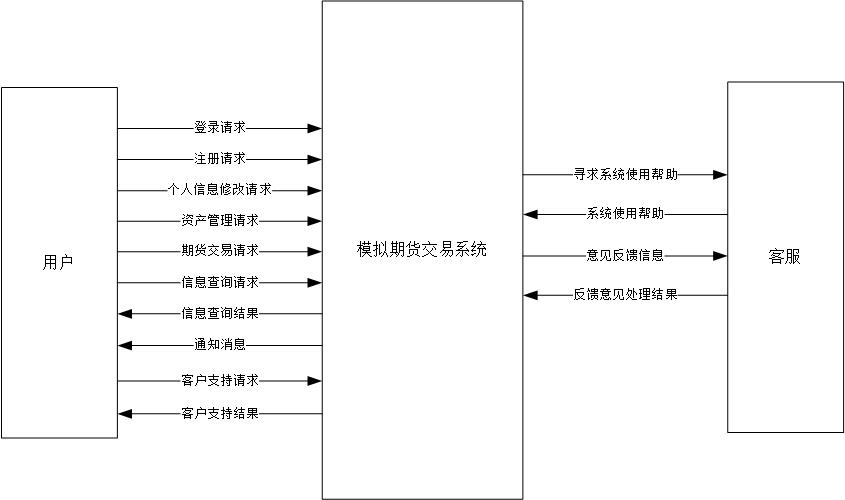
**表3-26 发送通知用例描述**

|  |  |
| --- | --- |
| **用例名称** | 撤单 |
| **用例描述** | 用户撤销委托订单 |
| **参与者** | 用户 |
| **基本路径** | 1. 用户进入订单管理界面或进入某只期货的行情详情界面 2. 用户选择需要进行撤单操作的订单 3. 用户点击撤单按钮 4. 系统判定撤销订单操作是否合法 5. 如果合法，发送“撤销订单成功”通知 |
| **拓展路径** | 系统判定撤销订单操作不合法，撤销订单失败 |

**表3-27 撤单用例描述**

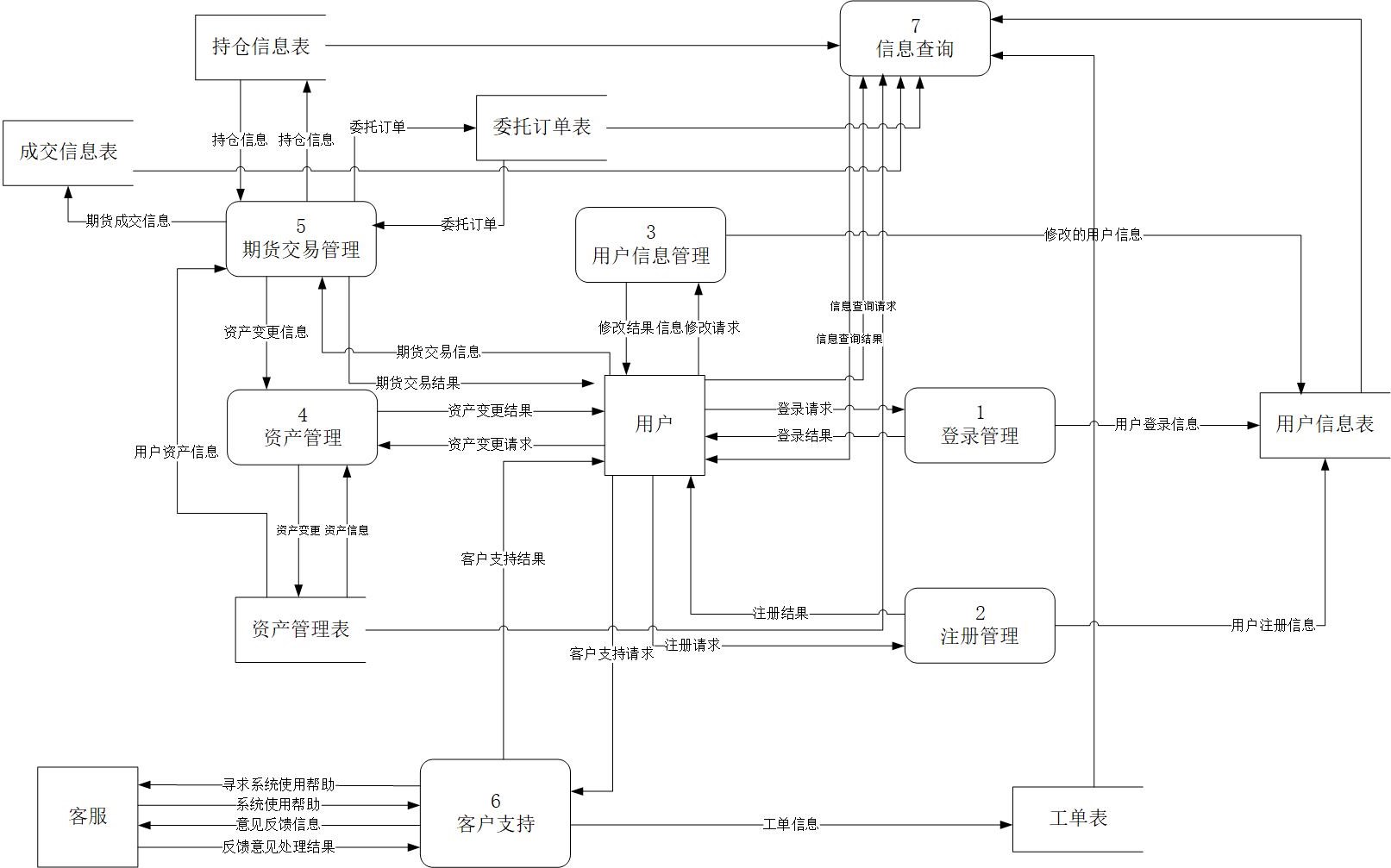
1. 传统的分析说明书
   1. 数据流图

### 顶层数据流图



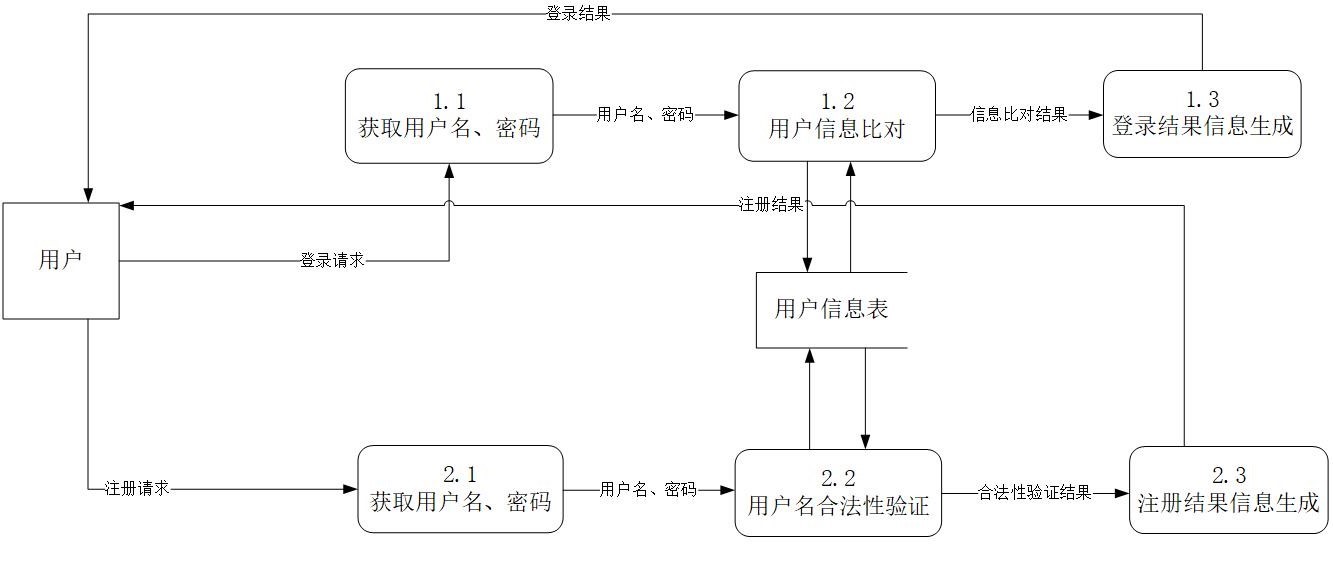
**图4-1 顶层数据流图**

### 0层数据流图

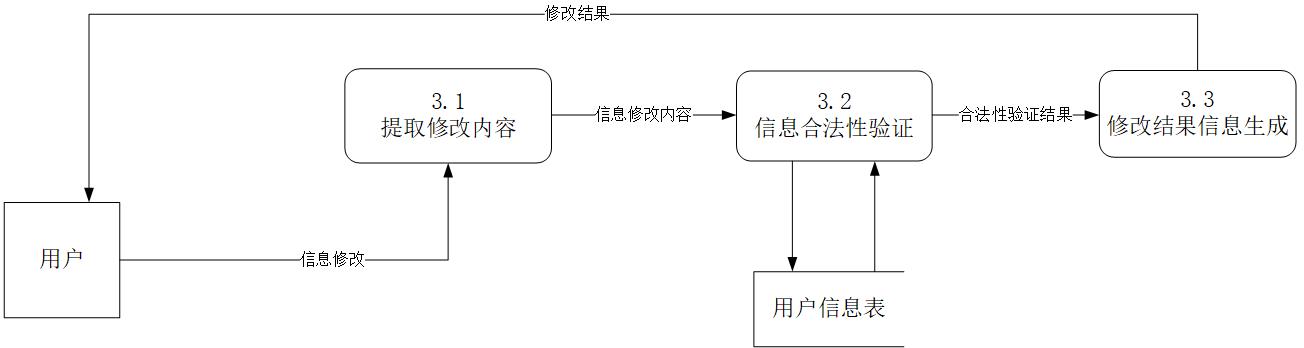


**图4-2 0层数据流图**

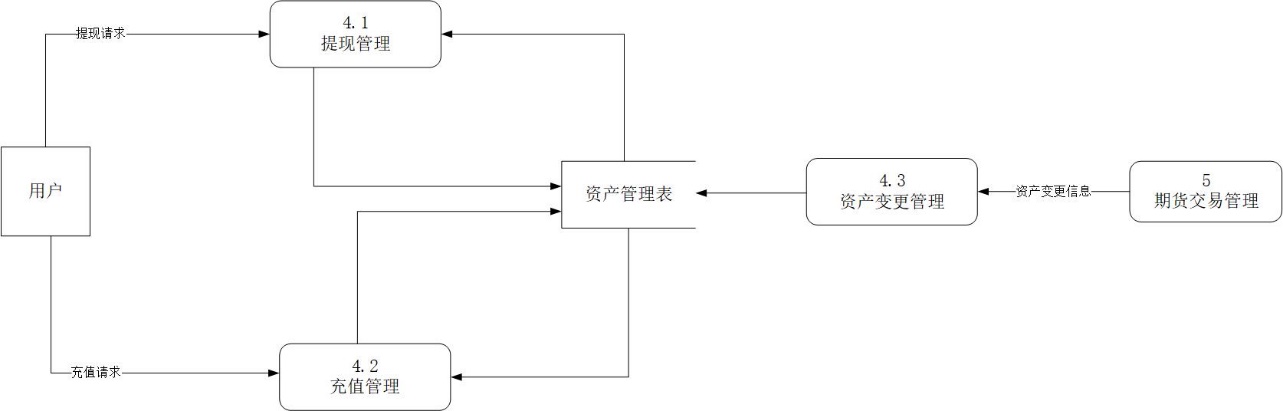
### 1层数据流图

****

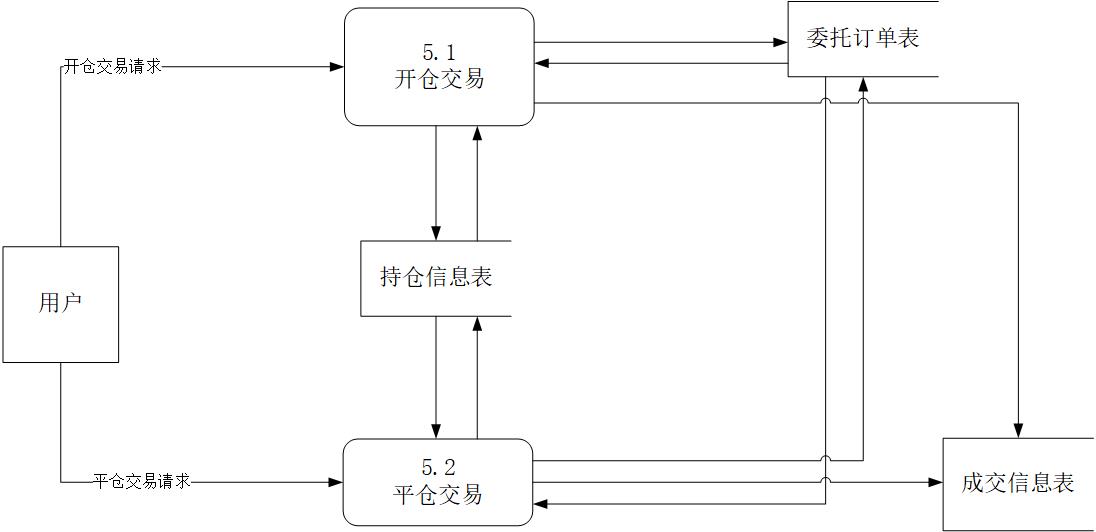
**图4-3 1层数据流图（登录、注册）**

****

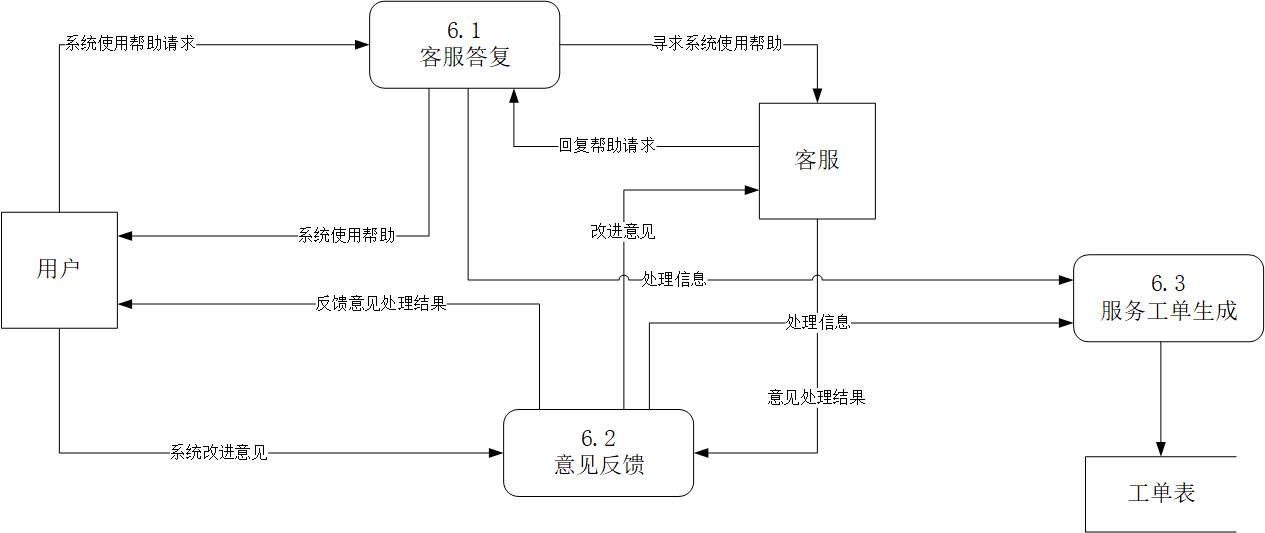
**图4-4 1层数据流图（信息修改）**

****

**图4-5 1层数据流图（资产管理）**

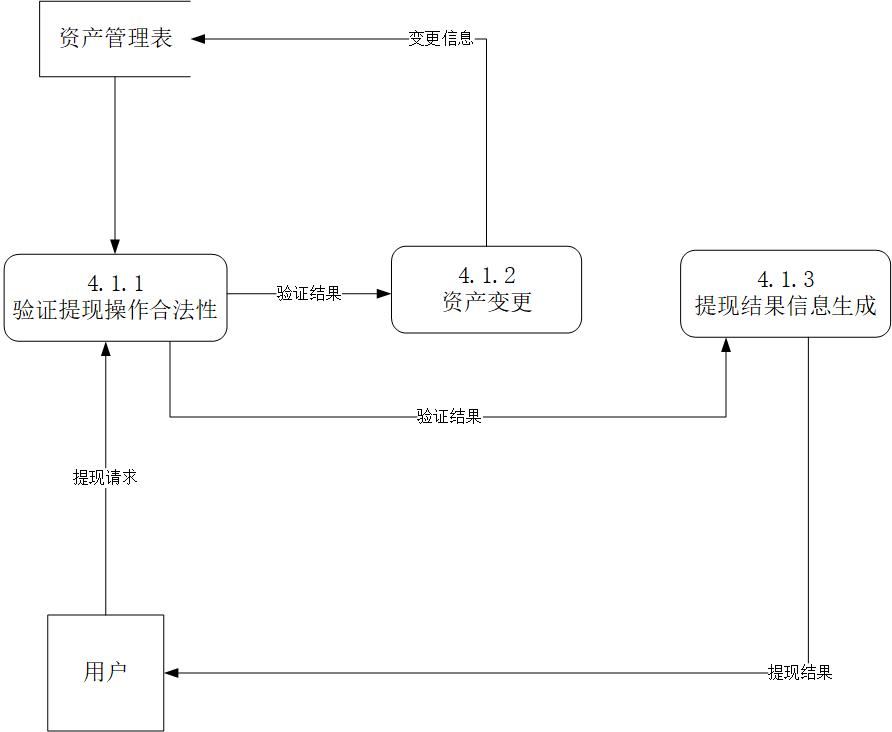
****

**图4-6 1层数据流图（期货交易）**

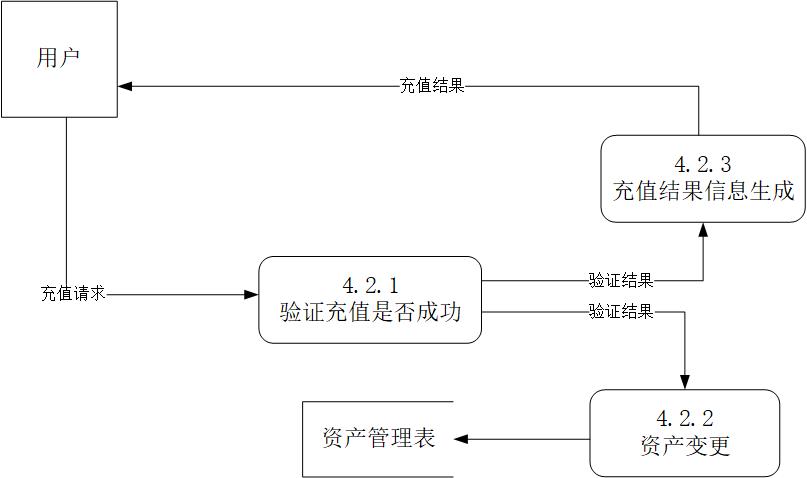
****

**图4-7 1层数据流图（客户支持）**

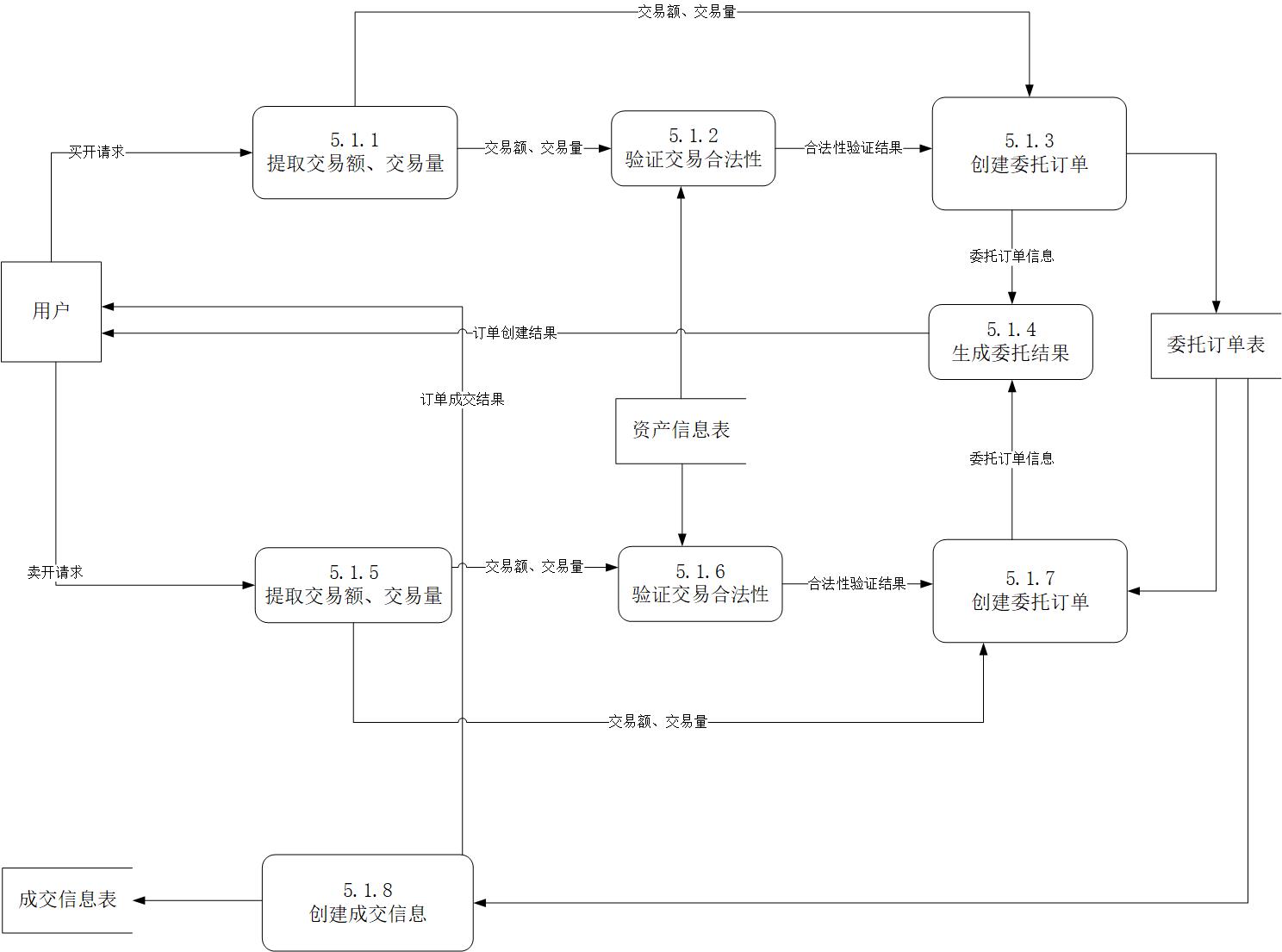
### 2层数据流图

****

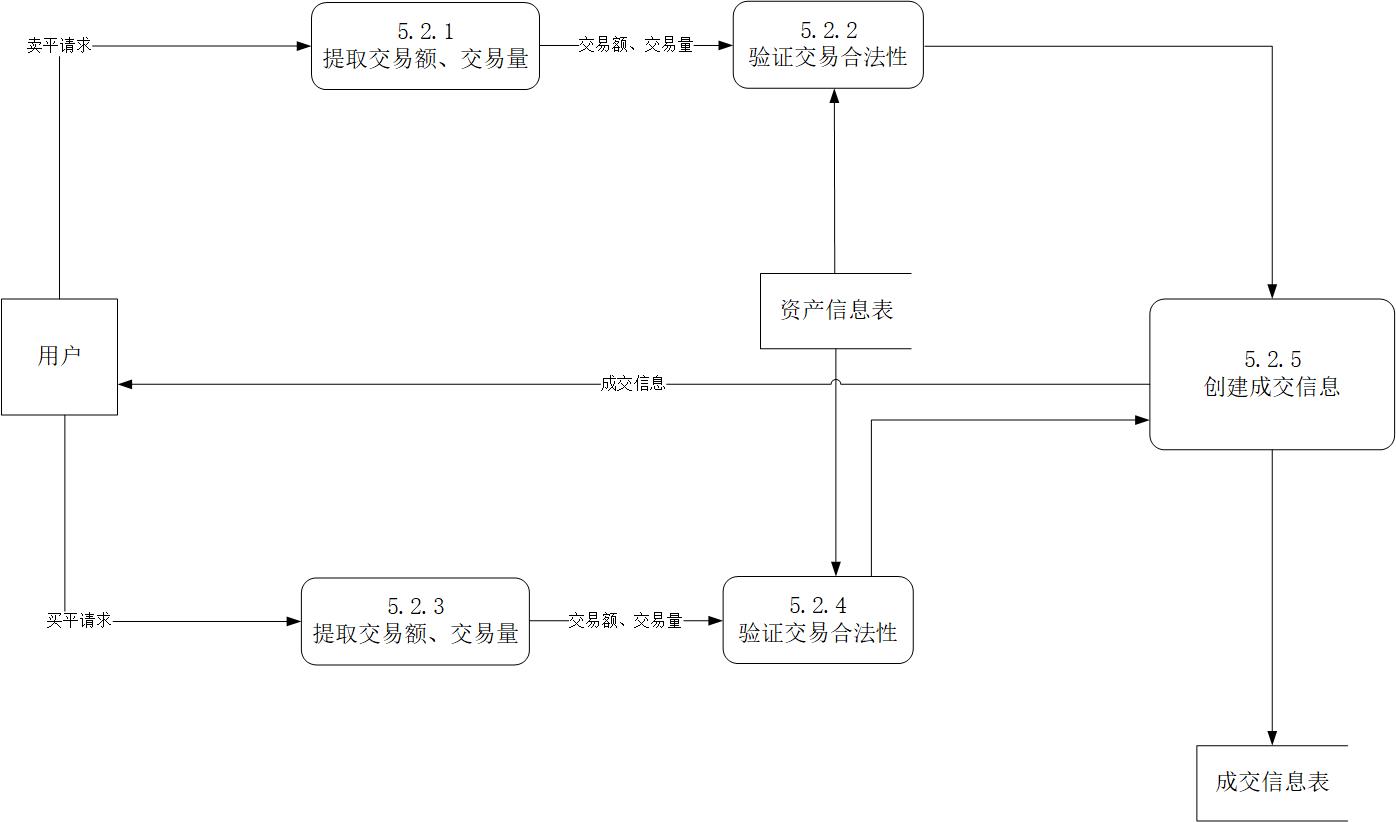
**图4-8 2层数据流图（提现管理）**

****

**图4-9 2层数据流图（充值管理）**

****

**图4-10 2层数据流图（开仓交易）**

****

**图4-11 2层数据流图（平仓交易）**

* 1. 数据字典

### 数据流

|  |  |
| --- | --- |
| 数据流名 | 信息修改请求 |
| 说明 | 用户向系统提交的修改个人信息的请求 |
| 数据流来历 | 用户 |
| 数据流去向 | 用户信息管理 |
| 组成 | 用户ID、需要修改的信息的ID、修改后的信息内容 |

|  |  |
| --- | --- |
| 数据流名 | 修改结果 |
| 说明 | 系统向用户返回的修改个人信息的结果 |
| 数据流来历 | 用户信息管理 |
| 数据流去向 | 用户 |
| 组成 | 信息是否成功修改 |

|  |  |
| --- | --- |
| 数据流名 | 登录请求 |
| 说明 | 用户向系统提交的用于登录的请求信息 |
| 数据流来历 | 用户 |
| 数据流去向 | 登录管理 |
| 组成 | 用户名、用户密码 |

|  |  |
| --- | --- |
| 数据流名 | 登录结果 |
| 说明 | 系统向用户返回的登录操作的结果 |
| 数据流来历 | 登录管理 |
| 数据流去向 | 用户 |
| 组成 | 是否成功登录 |

|  |  |
| --- | --- |
| 数据流名 | 注册请求 |
| 说明 | 用户向系统提交的用于注册的请求信息 |
| 数据流来历 | 用户 |
| 数据流去向 | 注册管理 |
| 组成 | 用户名、用户密码 |

|  |  |
| --- | --- |
| 数据流名 | 注册结果 |
| 说明 | 系统向用户返回的注册操作的结果 |
| 数据流来历 | 注册管理 |
| 数据流去向 | 用户 |
| 组成 | 是否成功注册 |

|  |  |
| --- | --- |
| 数据流名 | 客户支持请求 |
| 说明 | 用户向系统提交的用户寻求客户支持的请求信息 |
| 数据流来历 | 用户 |
| 数据流去向 | 客户支持 |
| 组成 | 用户ID、客户支持类型、用户提交的信息 |

|  |  |
| --- | --- |
| 数据流名 | 资产变更请求 |
| 说明 | 用户向系统提交的变更资产的请求 |
| 数据流来历 | 用户 |
| 数据流去向 | 资产管理 |
| 组成 | 用户ID、资产变更类型、资产变更数额 |

|  |  |
| --- | --- |
| 数据流名 | 客户支持结果 |
| 说明 | 系统向用户返回的客户支持的结果 |
| 数据流来历 | 客户支持 |
| 数据流去向 | 用户 |
| 组成 | 客户支持信息 |

|  |  |
| --- | --- |
| 数据流名 | 资产变更结果 |
| 说明 | 系统向用户返回的资产变更的结果 |
| 数据流来历 | 资产管理 |
| 数据流去向 | 用户 |
| 组成 | 资产是否成功完成变更 |

|  |  |
| --- | --- |
| 数据流名 | 期货交易信息 |
| 说明 | 用户向系统提交的关于期货交易的请求信息 |
| 数据流来历 | 用户 |
| 数据流去向 | 期货交易管理 |
| 组成 | 用户ID、期货ID、交易数额、交易量 |

|  |  |
| --- | --- |
| 数据流名 | 期货交易结果 |
| 说明 | 系统向用户返回的期货交易的结果 |
| 数据流来历 | 期货交易管理 |
| 数据流去向 | 用户 |
| 组成 | 期货交易是否完成、委托订单是否成功创建 |

|  |  |
| --- | --- |
| 数据流名 | 信息查询请求 |
| 说明 | 用户向信息查询模块提交的查询信息的请求 |
| 数据流来历 | 用户 |
| 数据流去向 | 信息查询 |
| 组成 | 用户ID、需要查询的信息 |

|  |  |
| --- | --- |
| 数据流名 | 信息查询结果 |
| 说明 | 系统向用户返回的信息查询的结果 |
| 数据流来历 | 信息查询 |
| 数据流去向 | 用户 |
| 组成 | 用户希望查询的信息 |

|  |  |
| --- | --- |
| 数据流名 | 委托订单 |
| 说明 | 期货交易管理模块创建的委托订单信息 |
| 数据流来历 | 期货交易管理、委托订单表 |
| 数据流去向 | 委托订单表、期货交易管理、信息查询 |
| 组成 | 委托订单信息（委托价格、委托时间、委托交易量等） |

|  |  |
| --- | --- |
| 数据流名 | 持仓信息 |
| 说明 | 用户的持仓信息 |
| 数据流来历 | 期货交易管理、持仓信息表 |
| 数据流去向 | 持仓信息表、期货交易管理、信息查询 |
| 组成 | 持仓信息（期货ID、持仓量、成本价等） |

|  |  |
| --- | --- |
| 数据流名 | 期货成交信息 |
| 说明 | 期货的成交记录信息 |
| 数据流来历 | 期货交易管理 |
| 数据流去向 | 信息查询 |
| 组成 | 成交信息（期货ID、成交时间、用户ID等） |

|  |  |
| --- | --- |
| 数据流名 | 用户资产信息 |
| 说明 | 用户账户上拥有的总资产信息 |
| 数据流来历 | 资产管理表 |
| 数据流去向 | 期货交易管理 |
| 组成 | 用户资产总量、总盈亏、盈亏率等 |

|  |  |
| --- | --- |
| 数据流名 | 寻求系统使用帮助 |
| 说明 | 客户支持模块向客服发送的用户的使用帮助请求 |
| 数据流来历 | 客户支持 |
| 数据流去向 | 客服 |
| 组成 | 用户的使用帮助请求信息 |

|  |  |
| --- | --- |
| 数据流名 | 系统使用帮助 |
| 说明 | 客户向客户支持模块返回的使用指南 |
| 数据流来历 | 客服 |
| 数据流去向 | 客户支持 |
| 组成 | 用户需要的系统使用指南 |

|  |  |
| --- | --- |
| 数据流名 | 意见反馈信息 |
| 说明 | 客户支持模块发送给客服的用户的反馈意见 |
| 数据流来历 | 客户支持 |
| 数据流去向 | 客服 |
| 组成 | 用户的意见反馈信息 |

|  |  |
| --- | --- |
| 数据流名 | 反馈意见处理结果 |
| 说明 | 客服向客户支持模块返回的反馈意见处理结果 |
| 数据流来历 | 客服 |
| 数据流去向 | 客户支持 |
| 组成 | 反馈意见处理结果 |

|  |  |
| --- | --- |
| 数据流名 | 工单信息 |
| 说明 | 客户支持服务的服务记录信息 |
| 数据流来历 | 客户支持 |
| 数据流去向 | 工单表 |
| 组成 | 工单信息（服务ID、服务内容、服务时间等） |

|  |  |
| --- | --- |
| 数据流名 | 资产变更信息 |
| 说明 | 用户期货交易达成而产生的资产变更 |
| 数据流来历 | 期货交易管理 |
| 数据流去向 | 资产管理 |
| 组成 | 用户ID、资产变更数额 |

|  |  |
| --- | --- |
| 数据流名 | 资产变更 |
| 说明 | 由于用户资产变更操作或由于期货交易达成而产生的资产变更 |
| 数据流来历 | 资产管理 |
| 数据流去向 | 资产管理表 |
| 组成 | 用户ID、资产变更数额 |

|  |  |
| --- | --- |
| 数据流名 | 用户登录信息 |
| 说明 | 请求登录的用户的用户名和用户密码 |
| 数据流来历 | 登录管理 |
| 数据流去向 | 用户信息表 |
| 组成 | 用户名、用户密码 |

|  |  |
| --- | --- |
| 数据流名 | 用户登录信息 |
| 说明 | 请求注册的用户名和用户密码 |
| 数据流来历 | 注册管理 |
| 数据流去向 | 用户信息表 |
| 组成 | 用户名、用户密码 |

### 数据存储

|  |  |
| --- | --- |
| 数据存储名 | 用户信息表 |
| 说明 | 存储用户的个人信息 |
| 流入的数据流 | 用户登录信息、用户注册信息、修改的用户信息 |
| 流出的数据流 | 用户信息 |
| 组成 | 用户ID、用户名、用户密码等 |

|  |  |
| --- | --- |
| 数据存储名 | 持仓信息表 |
| 说明 | 存储用户的持仓信息 |
| 流入的数据流 | 持仓信息 |
| 流出的数据流 | 持仓信息 |
| 组成 | 期货ID、用户ID、持仓量、成本价、现价等 |

|  |  |
| --- | --- |
| 数据存储名 | 成交信息表 |
| 说明 | 存储委托订单的成交信息 |
| 流入的数据流 | 期货成交信息 |
| 流出的数据流 | 期货成交信息 |
| 组成 | 期货ID、委托订单ID、成交时间、手续费等 |

|  |  |
| --- | --- |
| 数据存储名 | 委托订单表 |
| 说明 | 存储所有委托订单，包括已完成的和未完成的 |
| 流入的数据流 | 委托订单 |
| 流出的数据流 | 委托订单 |
| 组成 | 订单ID、委托时间、委托价格、委托交易量等 |

|  |  |
| --- | --- |
| 数据存储名 | 资产管理表 |
| 说明 | 存储用户的资产信息 |
| 流入的数据流 | 资产变更、用户资产信息 |
| 流出的数据流 | 资产信息 |
| 组成 | 用户ID、资产信息等 |

|  |  |
| --- | --- |
| 数据存储名 | 工单表 |
| 说明 | 存储客户支持服务的记录 |
| 流入的数据流 | 工单信息 |
| 流出的数据流 | 工单信息 |
| 组成 | 工单ID、服务时间、服务内容、用户ID、客服ID等 |

### 处理过程

|  |  |
| --- | --- |
| 处理过程名 | 登录管理 |
| 说明 | 用于处理用户的登录请求 |
| 输入 | 登录请求 |
| 输出 | 用户登录信息、登录结果 |
| 处理 | 对用户信息进行验证，返回登录结果（成功与否） |

|  |  |
| --- | --- |
| 处理过程名 | 注册管理 |
| 说明 | 用于处理用户的注册请求 |
| 输入 | 注册请求 |
| 输出 | 用户注册信息、注册结果 |
| 处理 | 对用户的注册请求进行验证，返回注册结果（成功与否） |

|  |  |
| --- | --- |
| 处理过程名 | 用户信息管理 |
| 说明 | 用户处理用户的修改个人信息请求 |
| 输入 | 信息修改请求 |
| 输出 | 修改结果 |
| 处理 | 判断用户希望修改的信息是否合法，并根据判断结果修改个人信息或不修改个人信息，并返回修改结果（成功与否） |

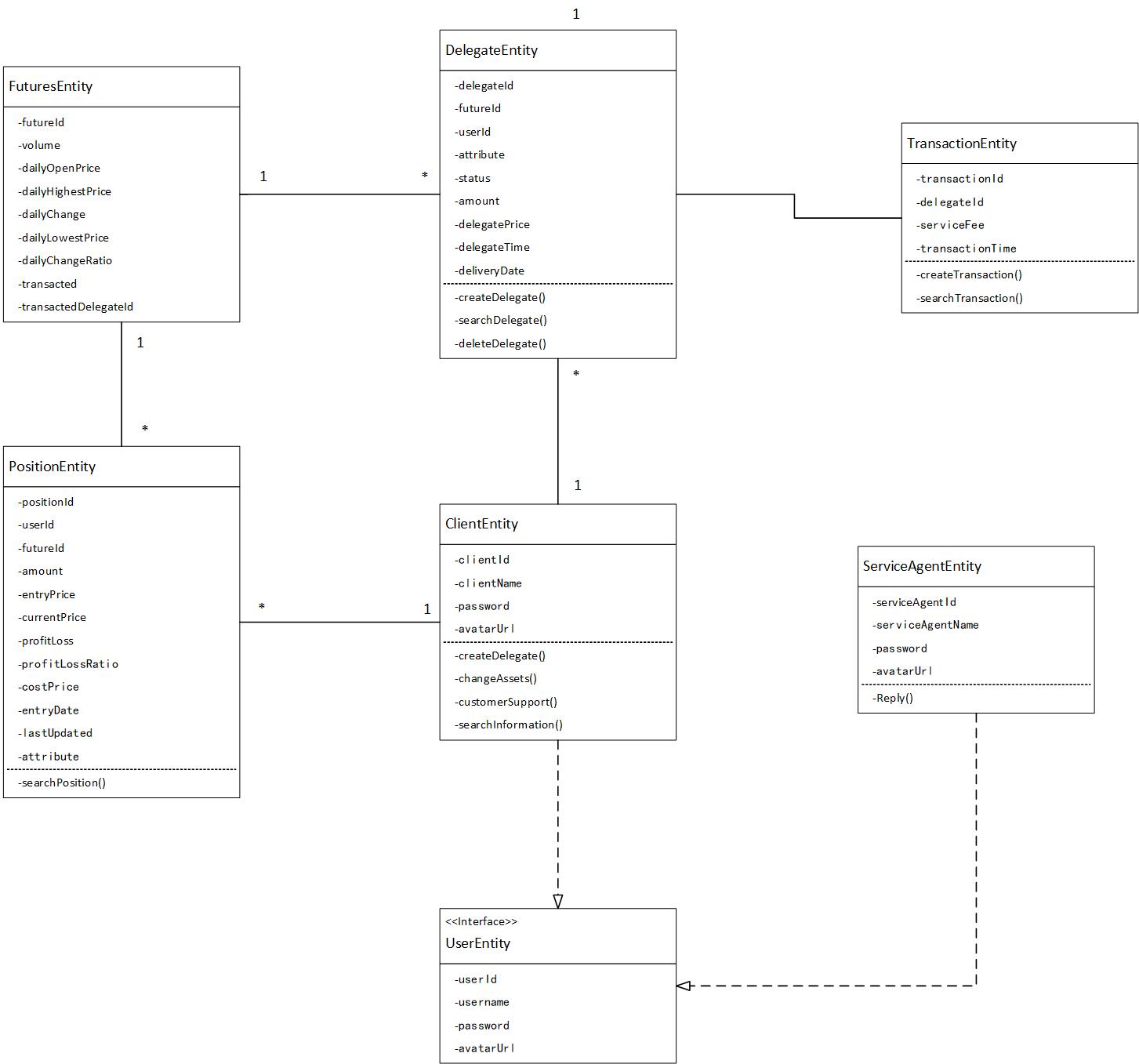
|  |  |
| --- | --- |
| 处理过程名 | 资产管理 |
| 说明 | 用于处理用户的资产变更 |
| 输入 | 资产变更请求、资产变更信息 |
| 输出 | 资产变更结果、资产变更 |
| 处理 | 根据用户的资产变更请求或期货交易管理模块的资产变更信息更改用户的资产，并根据情况返回资产变更请求 |

|  |  |
| --- | --- |
| 处理过程名 | 期货交易管理 |
| 说明 | 用户处理期货交易操作 |
| 输入 | 期货交易信息、持仓信息、用户资产信息、委托订单 |
| 输出 | 期货交易结果、持仓信息、期货成交信息、资产变更信息、委托订单 |
| 处理 | 接收用户的期货交易信息，判断是否满足创建委托订单的条件并根据情况创建委托订单，同时执行订单的成交操作 |

|  |  |
| --- | --- |
| 处理过程名 | 客户支持 |
| 说明 | 用户和客服沟通的中转站 |
| 输入 | 客户支持请求 |
| 输出 | 寻求系统使用帮助 |
| 处理 | 意见反馈信息 |

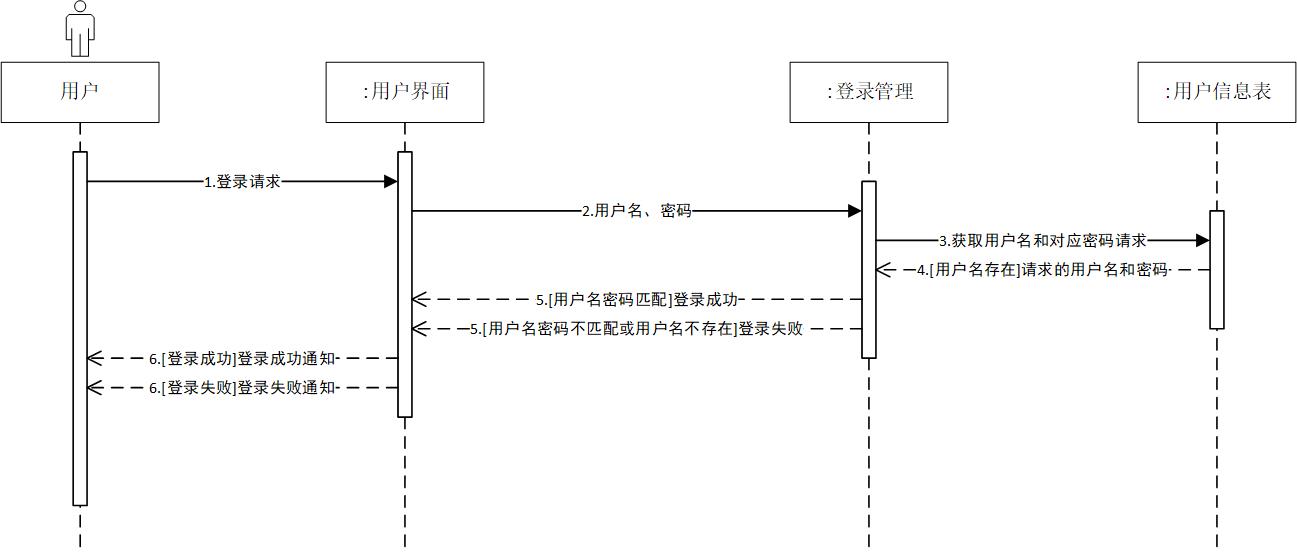
|  |  |
| --- | --- |
| 处理过程名 | 信息查询 |
| 说明 | 用来处理用户的信息查询请求 |
| 输入 | 信息查询请求、持仓信息、成交信息、委托订单、用户信息、资产信息、工单信息 |
| 输出 | 信息查询结果 |
| 处理 | 根据接收到的用户查询请求，从数据表中查询对应的信息，并将查询到的信息返回给用户 |

1. 面向对象的分析说明书
   1. 类图

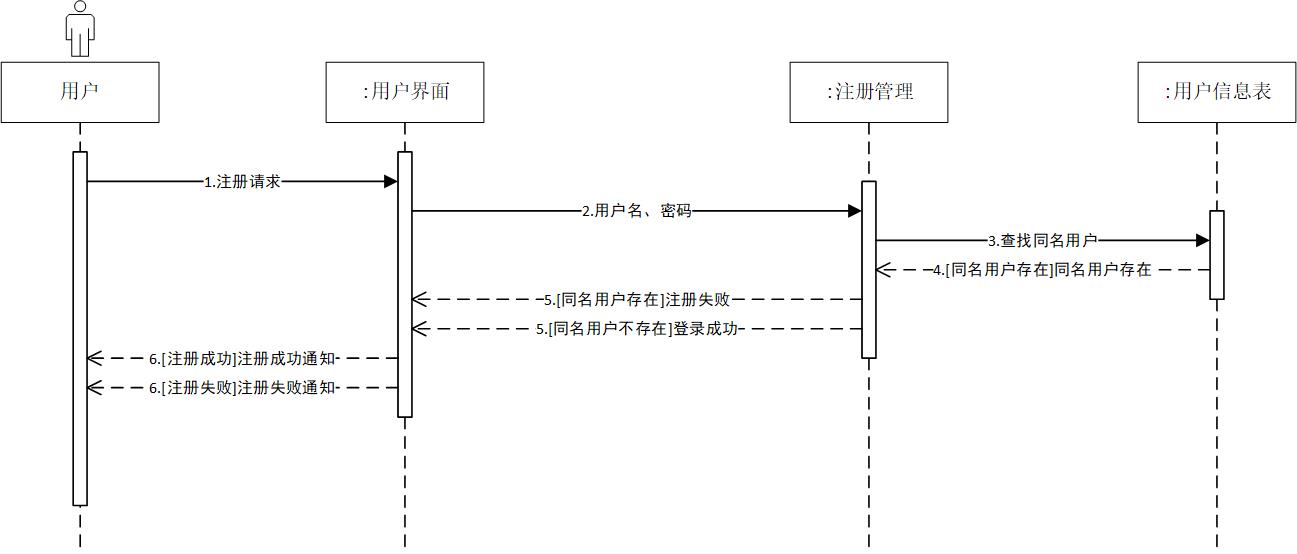


**图5-1 类图**

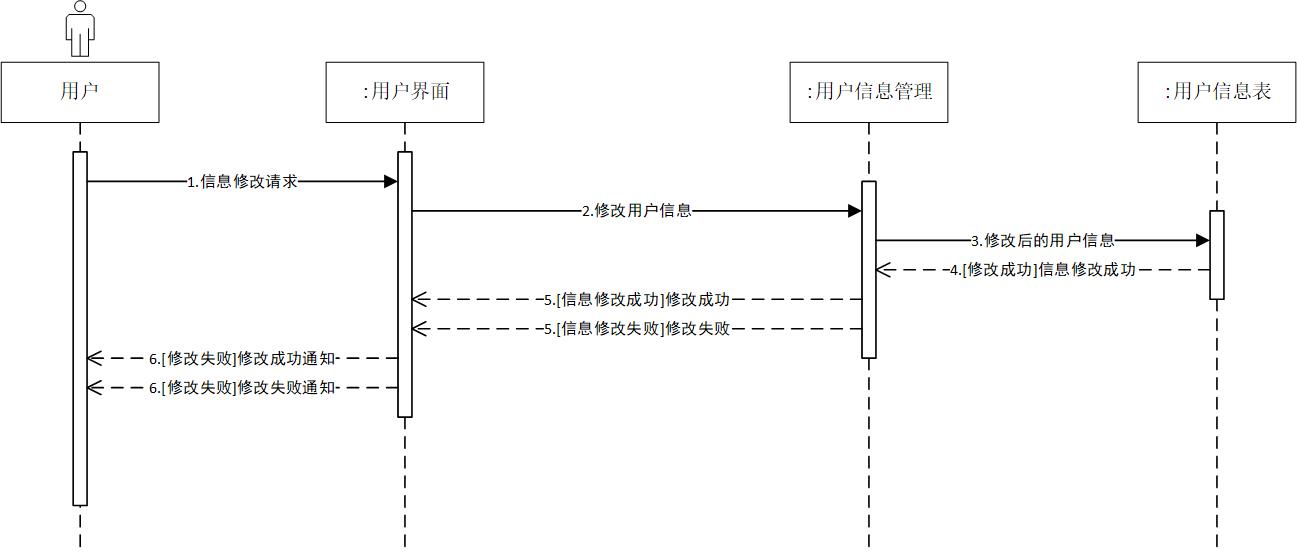
* 1. 顺序图



**图5-2登录顺序图**



**图5-3注册顺序图**



**图5-4信息修改顺序图**



**图5-5 资产变更顺序图**



**图5-6 开仓顺序图**



**图5-7 平仓顺序图**

* 1. 协作图



**图5-9 登录协作图**



**图5-10 注册协作图**



**图5-11 修改信息协作图**



**图5-12 资产管理协作图**



**图5-13 开仓协作图**



**图5-14 平仓协作图**

1. 面向对象的设计说明书
   1. 系统结构图



**图6-1 系统结构图**

* 1. 主要类信息

### 实体类

|  |  |  |
| --- | --- | --- |
| 类名 | 属性 | 基本方法 |
| DelegateEntity | delegateId : int  futureId : int  userId : int  attribute : String  status : String  amount : int  delegatePrice: double  delegateTime: String  deliveryDate: String | getDelegateId() : int  setDelegateId(int) : void  getFutureId() : int  setFutureId(int) : void  getUserId() : int  setUserId(int) : void  getAttribute() : String  setAttribute(String) : void  getStatus() : String  setStatus(String) : void  getAmount() : int  setAmount(int) : void  getDelegatePrice() : double  setDelegatePrice(double) : void  getDelegateTime() : String  setDelegateTime(String) : void  getDeliveryDate() : String  setDeliveryDate(String) : void |
| FuturesEntity | price : double  volume : int  dailyOpenPrice : double  dailyHighestPrice : double  dailyLowestPrice : double  dailyChange : double  dailyChangeRatio : double  transacted : boolean  transactedDelegateId : int | getPrice() : double  setPrice(double) : void  getVolume() : int  setVolume(int) : void  getDailyOpenPrice() : double  setDailyOpenPrice(double) : void  getDailyHighestPrice() : double  setDailyHighestPrice(double) : void  getDailyLowestPrice() : double  setDailyLowestPrice(double) : void  getDailyChange() : double  setDailyChange(double) : void  getDailyChangeRatio() : double  setDailyChangeRatio(double) : void  isTransacted() : boolean  setTransacted(boolean) : void  getTransactedDelegateId() : int  setTransactedDelegateId(int) : void |
| PositionEntity | positionId : int  userId : int  futureId : int  amount : double  entryPrice : double  currentPrice : double  profitLoss : double  profitLossRatio : double  costPrice : double  entryDate : String  lastUpdated : String  attribute : String | getPositionId() : int  setPositionId(int) : void  getUserId() : int  setUserId(int) : void  getFutureId() : int  setFutureId(int) : void  getAmount() : double  setAmount(double) : void  getEntryPrice() : double  setEntryPrice(double) : void  getCurrentPrice() : double  setCurrentPrice(double) : void  getProfitLoss() : double  setProfitLoss(double) : void  getProfitLossRatio() : double  setProfitLossRatio(double) : void  getCostPrice() : double  setCostPrice(double) : void  getEntryDate() : String  setEntryDate(String) : void  getLastUpdated() : String  setLastUpdated(String) : void  getAttribute() : String  setAttribute(String) : void |
| TransactionEntity | transactionid : int  delegateid : int  serviceFee : double  transactionTime : String | getTransactionId() : int  setTransactionId(int) : void  getDelegateId() : int  setDelegateId(int) : void  getServiceFee() : double  setServiceFee(double) : void  getTransactionTime() : String  setTransactionTime(String) : void |
| ClientEntity | userId : int  username : String  password : String  avatarUrl : String | getUserId() : int  setUserId(int) : void  getUsername() : String  setUsername(String) : void  getPassword() : String  setPassword(String) : void  getAvatarUrl() : String  setAvatarUrl(String) : void |
| ServiceAgentEntity | userId : int  username : String  password : String  avatarUrl : String | getUserId() : int  setUserId(int) : void  getUsername() : String  setUsername(String) : void |

**表6-1 实体类**

### 控制类

|  |  |  |
| --- | --- | --- |
| 类名 | 属性 | 方法 |
| AvatarUploadController | userServiceImpl : UserServiceImpl | uploadImg (MultipartFile  , String): void  getUserAvatar(UserEntity) : void |
| CancelDelegateController | delegateService : DelegateService | cancelDelegate(CancelDelegateRequest) : CancelDelegateResponse |
| ClosePositionController | delegateService : DelegateService  positionServiceImpl : PositionServiceImpl  transactionServiceImpl : TransactionServiceImpl  userServiceImpl : UserServiceImpl | closePosition(ClosePositionRequest) : ClosePositionResponse |
| CreateDelegateController | delegateService : DelegateService  userServiceImpl :  UserServiceImpl | createDelegate(CreateDelegateRequest) : CreateDelegateResponse |
| GetAssetsController | userServiceImpl :  UserServiceImpl  positionServiceImpl :  PositionServiceImpl | getAssets(GetAssetsRequest) : GetAssetsResponse |
| GetCurrentDelegateController | delegateService :  DelegateService | getCurrentDelegate(GetCurrentDelegateRequest) : GetCurrentDelegateResponse |
| GetFuturesDataController | futuresServiceImpl :  FuturesServiceImpl  positionServiceImpl :  PositionServiceImpl  delegateService :  DelegateService  transactionServiceImpl :  TransactionServiceImpl  currentRowList :  List<Integer> | getFuturesData(GetFutureDataRequest) : GetFuturesDataResponse |
| GetHistoricalDelegateController | delegateService :  DelegateService | selectHistoricalDelegate(GetHistoricalDelegateRequest) : GetHistoricalDelegateResponse |
| GetPositionController | positionServiceImpl : PositionServiceImpl | getPosition(GetPositionRequest) : GetPositionResponse |
| LoginController | userServiceImpl : UserServiceImpl | Login(LoginRequest) : LoginResponse |
| RechargeController | userServiceImpl : UserServiceImpl | Recharge(RechargeRequest) : RechargeResponse |
| RegisterController | userServiceImpl : UserServiceImpl | register(RegisterRequest) : RegisterResponse |

**表6-2 控制类**

### 边界类

|  |  |  |
| --- | --- | --- |
| 类名 | 属性 | 基本方法 |
| DelegateEntity | delegateId : int  futureId : int  userId : int  attribute : String  status : String  amount : int  delegatePrice: double  delegateTime: String  deliveryDate: String | getDelegateId() : int  setDelegateId(int) : void  getFutureId() : int  setFutureId(int) : void  getUserId() : int  setUserId(int) : void  getAttribute() : String  setAttribute(String) : void  getStatus() : String  setStatus(String) : void  getAmount() : int  setAmount(int) : void  getDelegatePrice() : double  setDelegatePrice(double) : void  getDelegateTime() : String  setDelegateTime(String) : void  getDeliveryDate() : String  setDeliveryDate(String) : void |

**表6-3 边界类**

窗体顶端

1. 测试技术说明书
   1. 等价类划分法

### 保证金充值

**等价类划分：**

|  |  |
| --- | --- |
| **有效等价类** | **编号** |
| 充值金额大于0 | 1 |

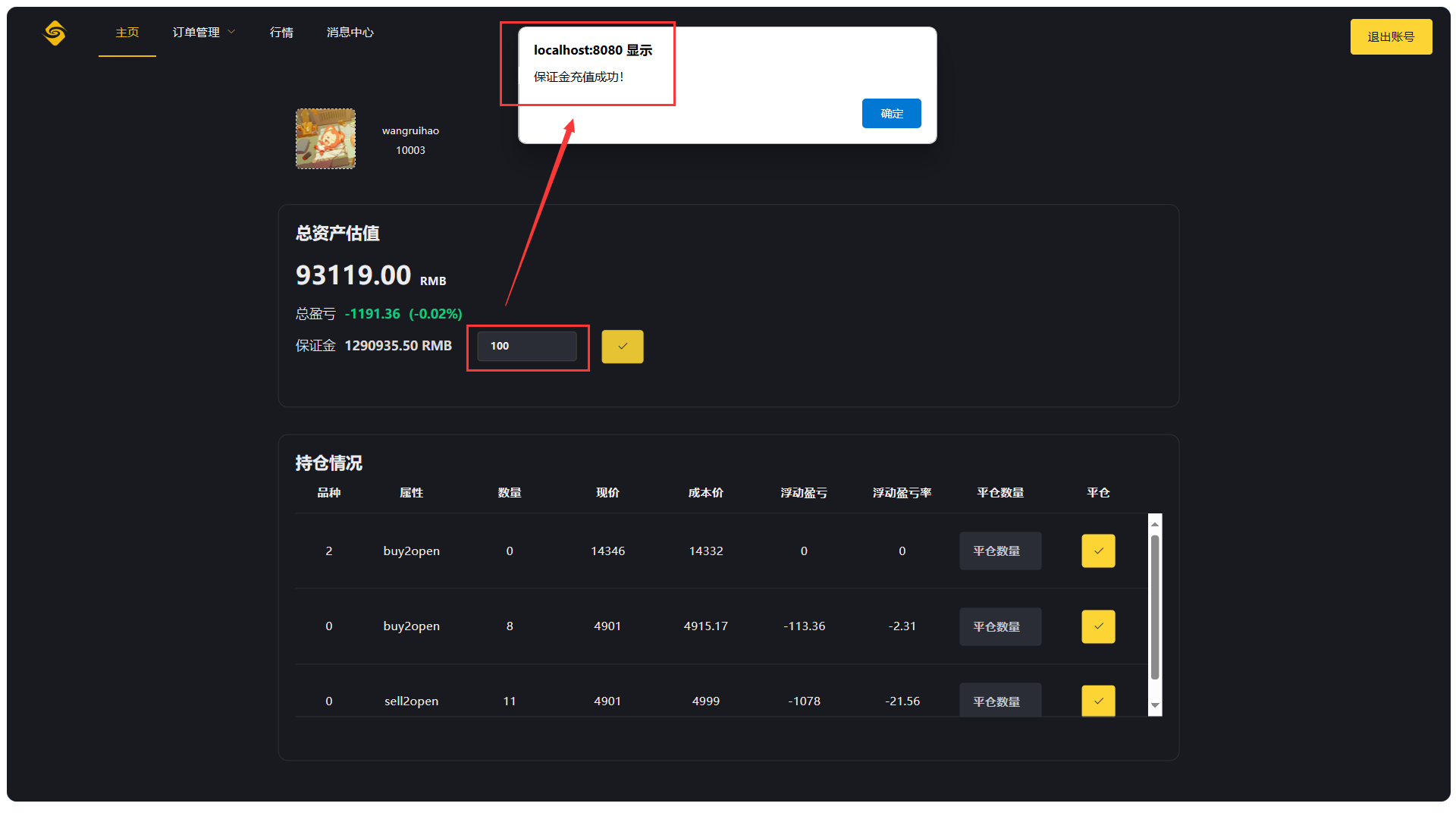
|  |  |
| --- | --- |
| **无效等价类** | **编号** |
| 充值金额小于等于0 | 2 |

**测试用例：**

1. 充值金额为100 覆盖1
2. 充值金额为-100 覆盖2

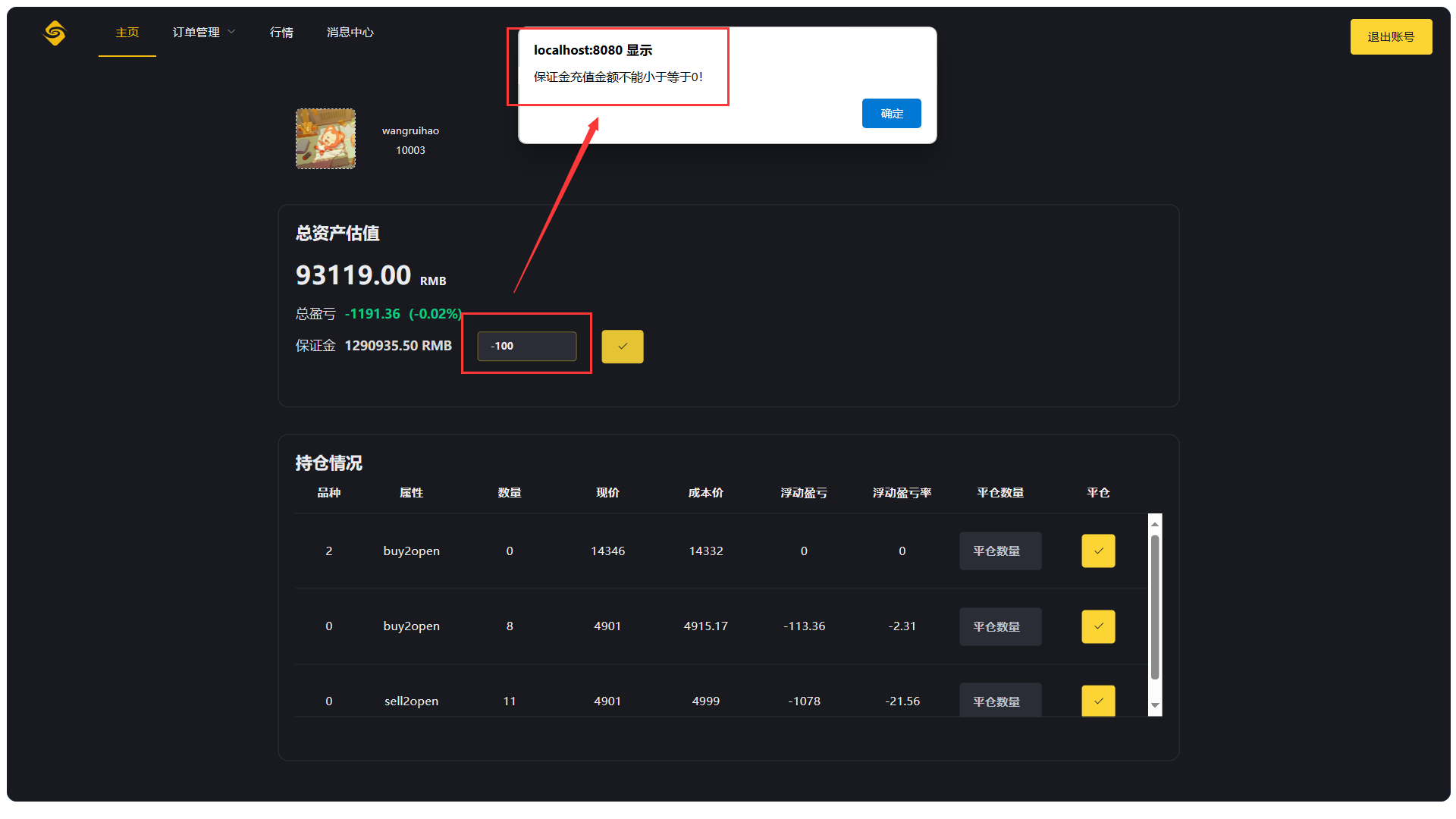
**测试结果：**

1. 充值金额为100



**图7-1 测试结果（充值金额为100）**

1. 充值金额为-100



**图7-2 测试结果（充值金额为-100）**

### 平仓交易

**等价类划分：**

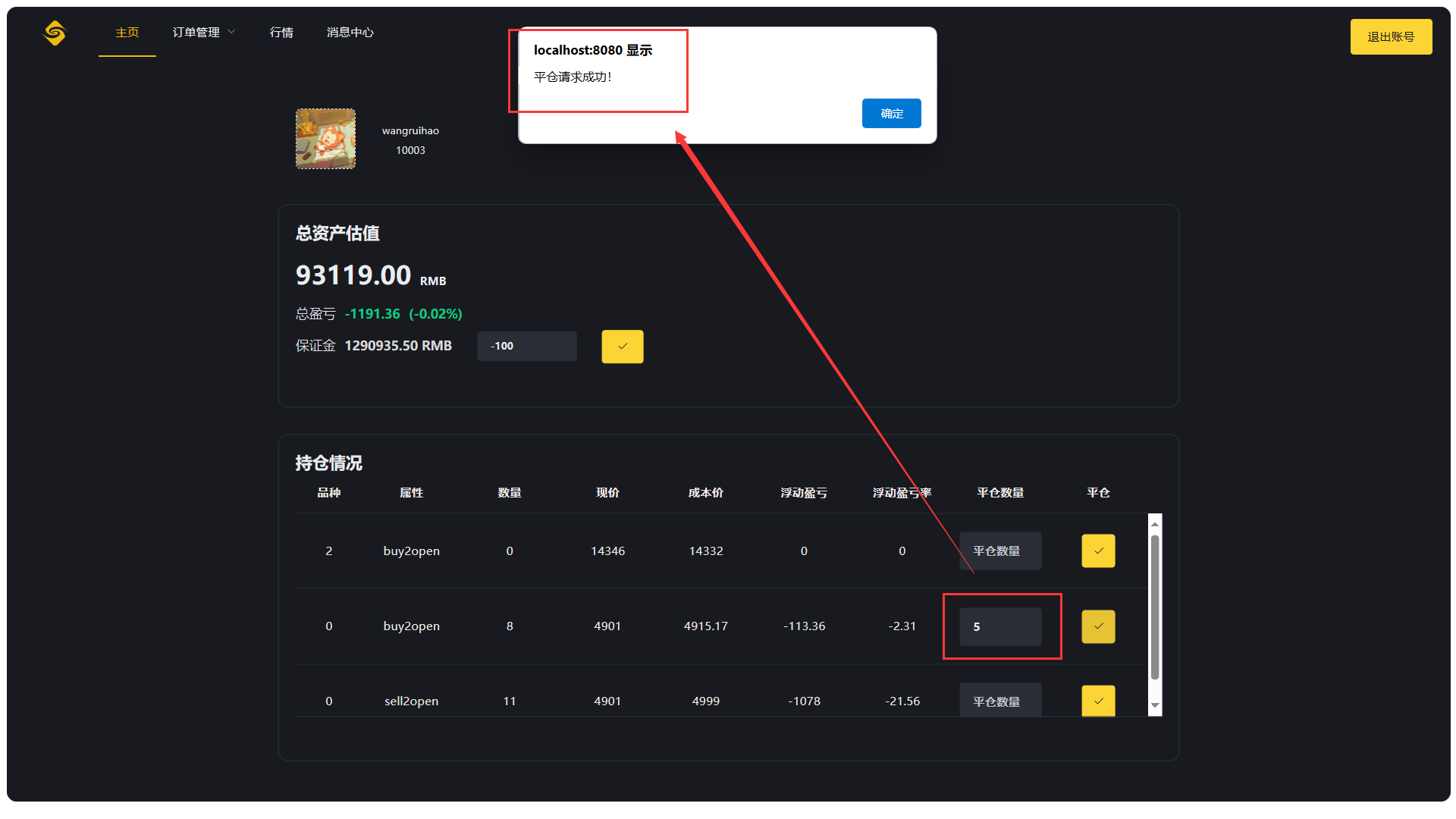
|  |  |
| --- | --- |
| **有效等价类** | **编号** |
| 平仓数量大于0且小于持仓数 | 1 |

|  |  |
| --- | --- |
| **无效等价类** | **编号** |
| 平仓数量小于等于0 | 2 |
| 平仓数量大于持仓数 | 3 |

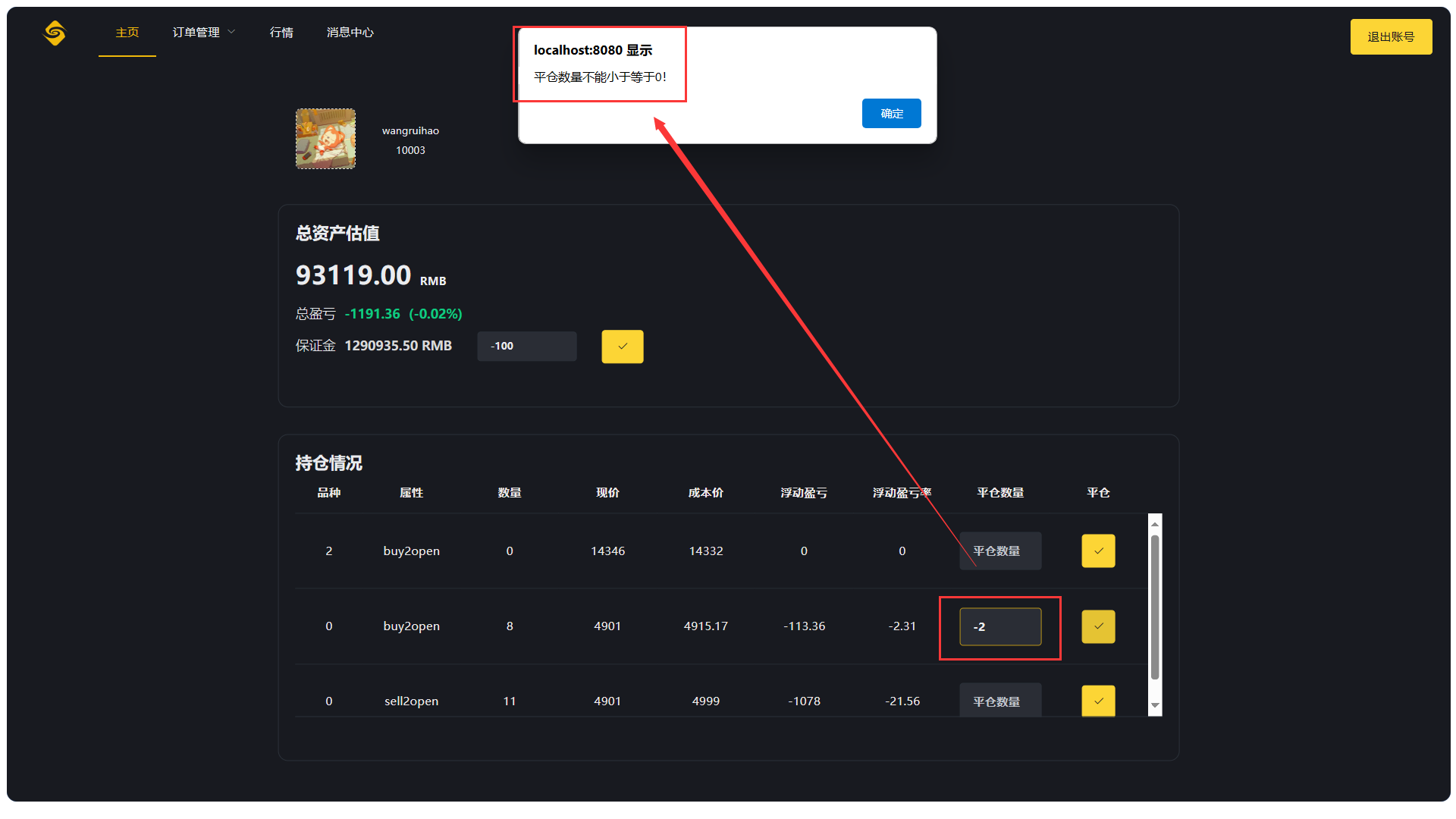
**测试用例：**

1. 平仓数量为5 覆盖1
2. 平仓数量为-2 覆盖2
3. 平仓数量为13 覆盖3

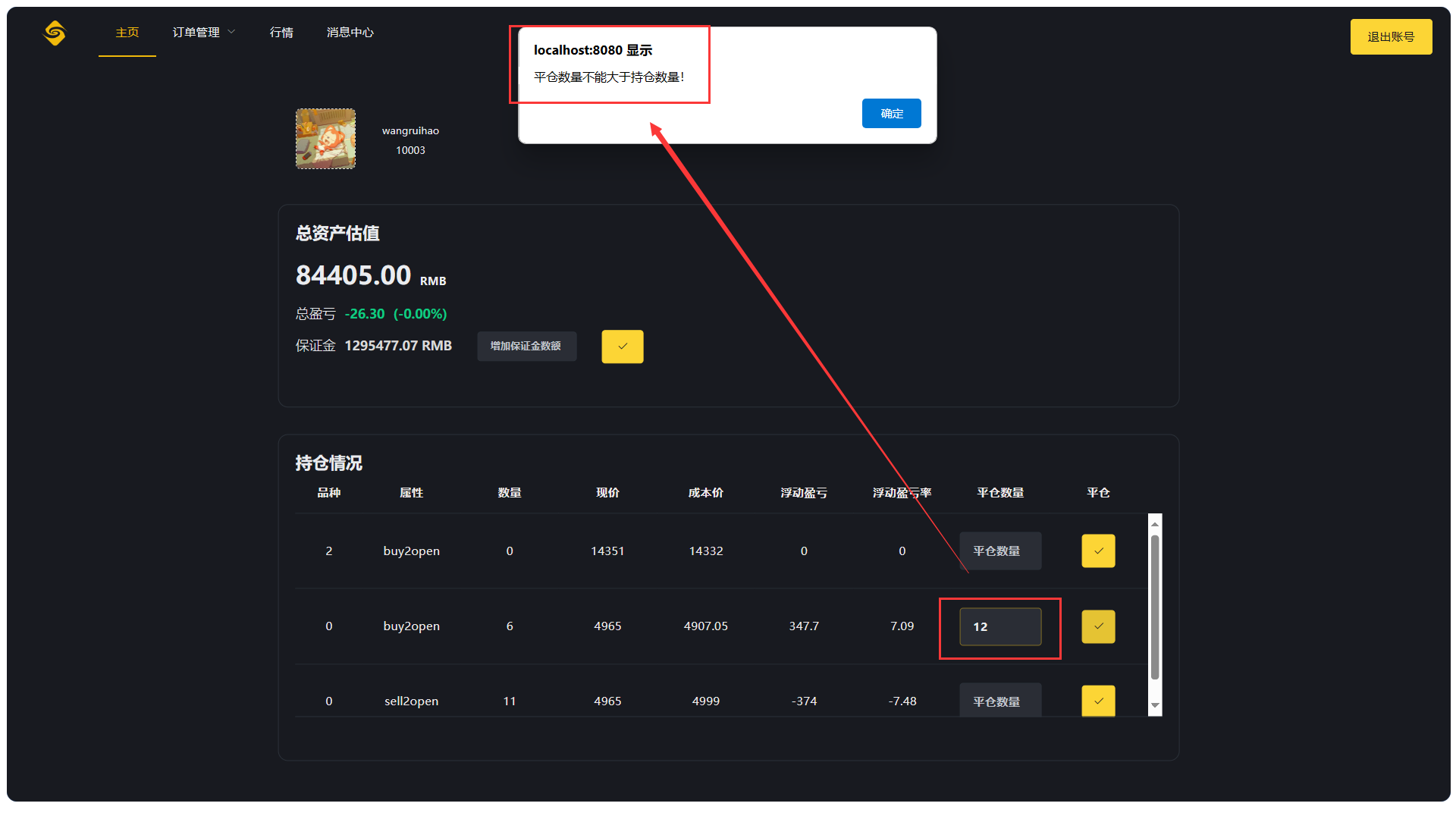
**测试结果：**



**图7-3 测试结果（平仓数量为5）**



**图7-4 测试结果（平仓数量为-2）**



**图7-5 测试结果（平仓数量为12）**

### 开仓交易

**等价类划分：**

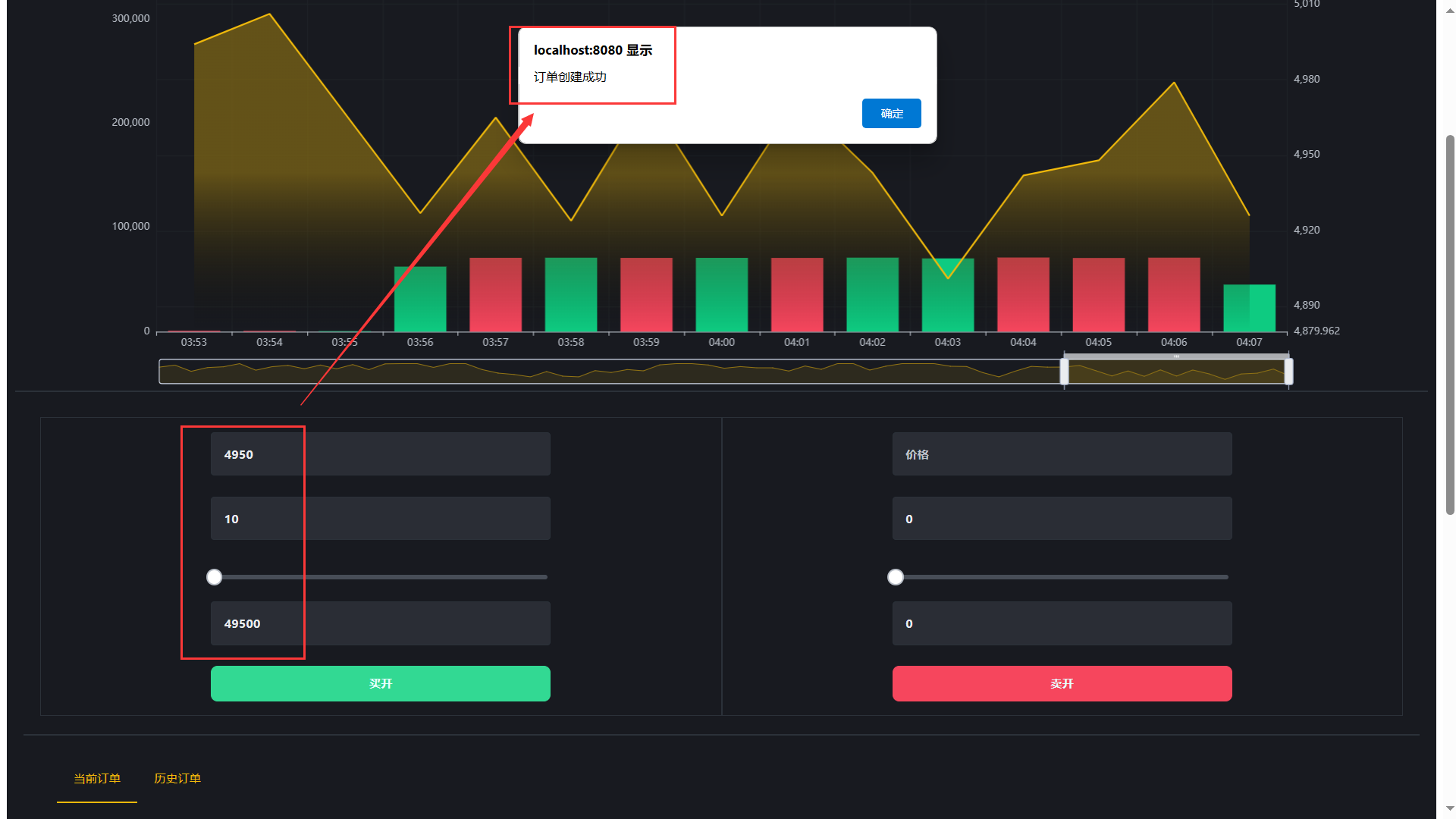
|  |  |
| --- | --- |
| **有效等价类** | **编号** |
| 价格大于0 | 1 |
| 数量大于0 | 2 |
| 总金额的10%小于等于保证金数额 | 3 |

|  |  |
| --- | --- |
| **无效等价类** | **编号** |
| 价格小于等于0 | 4 |
| 数量小于等于0 | 5 |
| 总金额的10%大于保证金数额 | 6 |

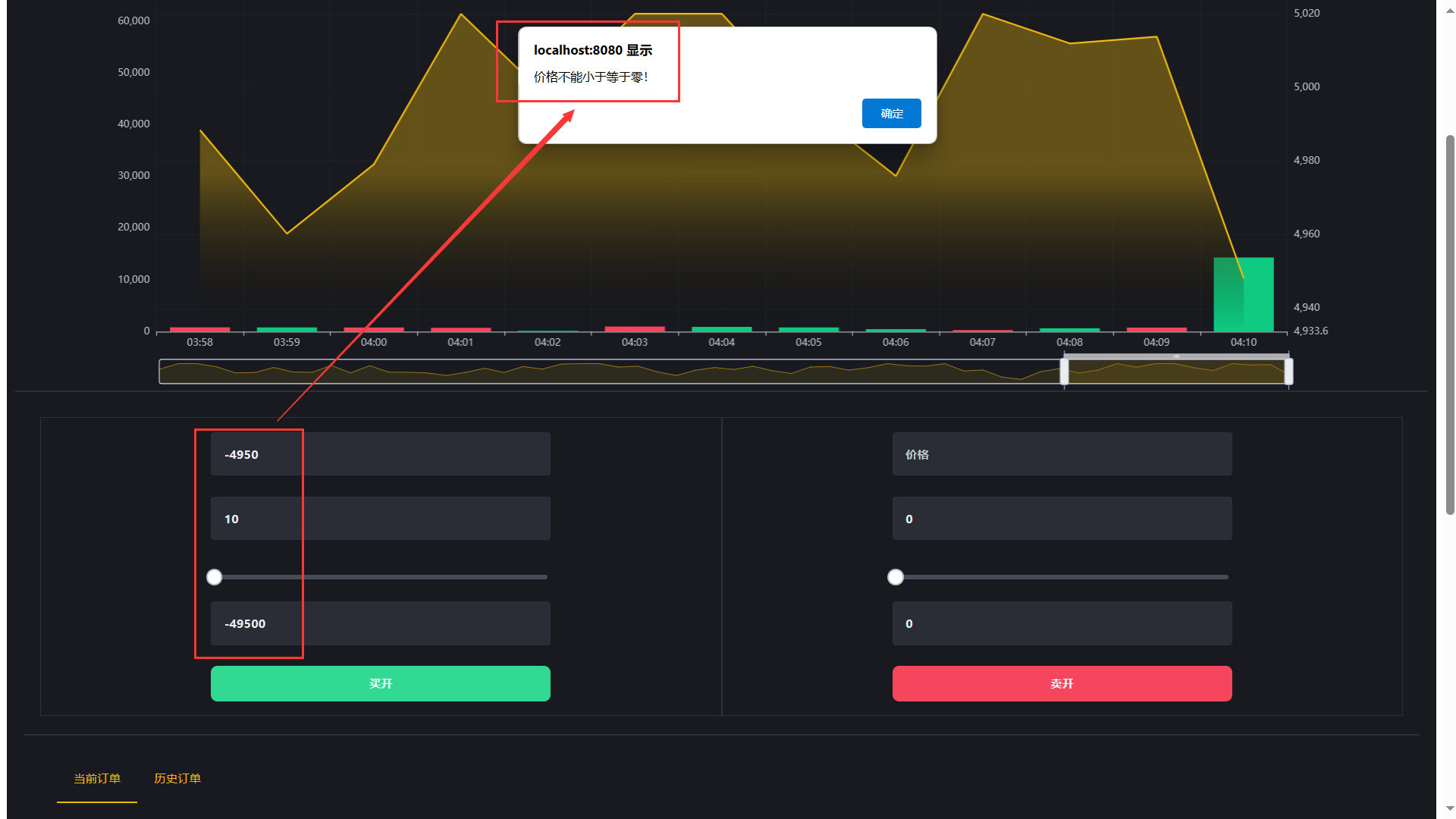
**测试用例：**

1. 价格为4950，数量为10，总金额为49500 覆盖1、2、3
2. 价格为-4950 覆盖4
3. 数量为-10 覆盖5
4. 总金额的10%为99999999 覆盖6

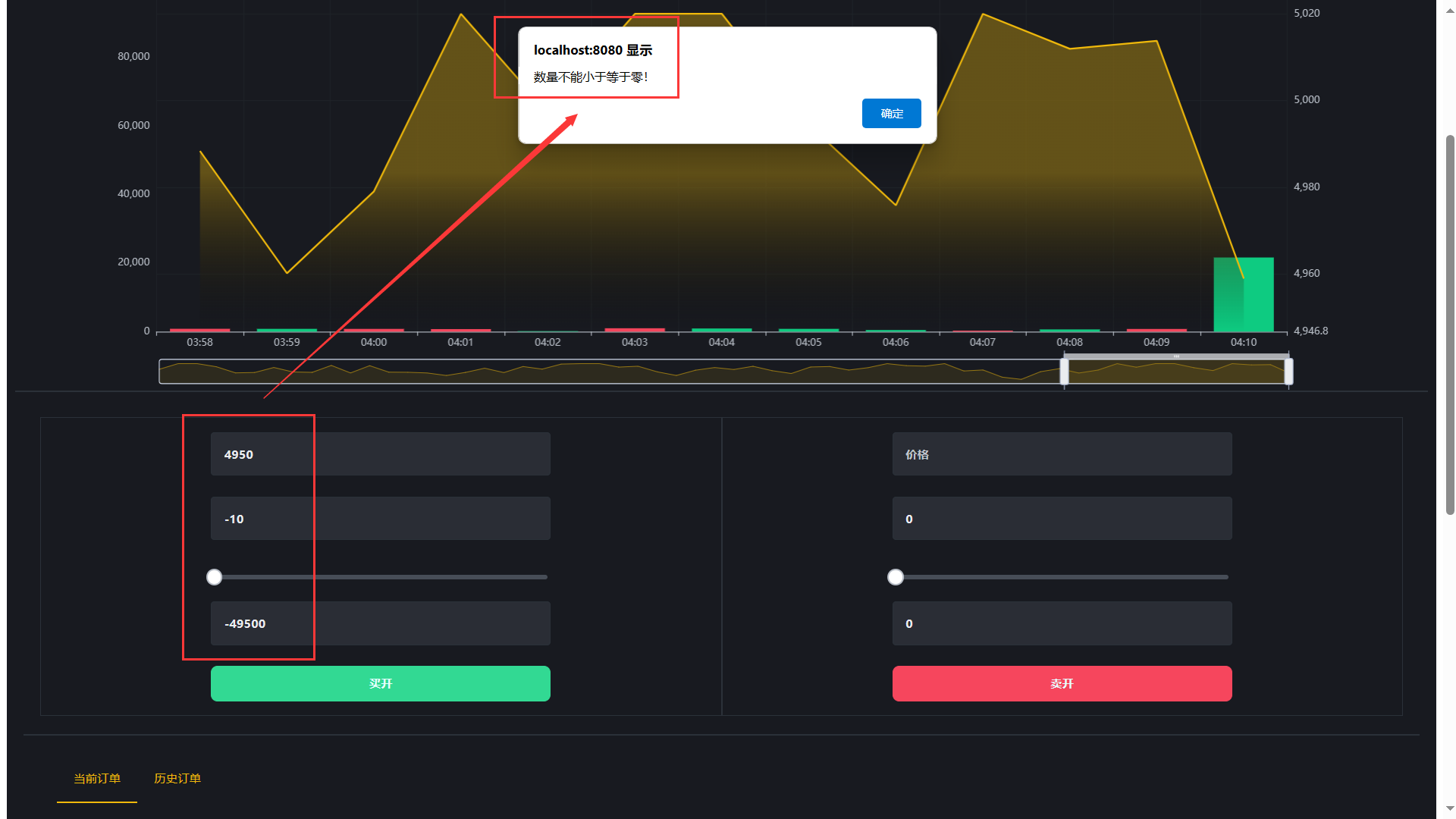
**测试结果：**



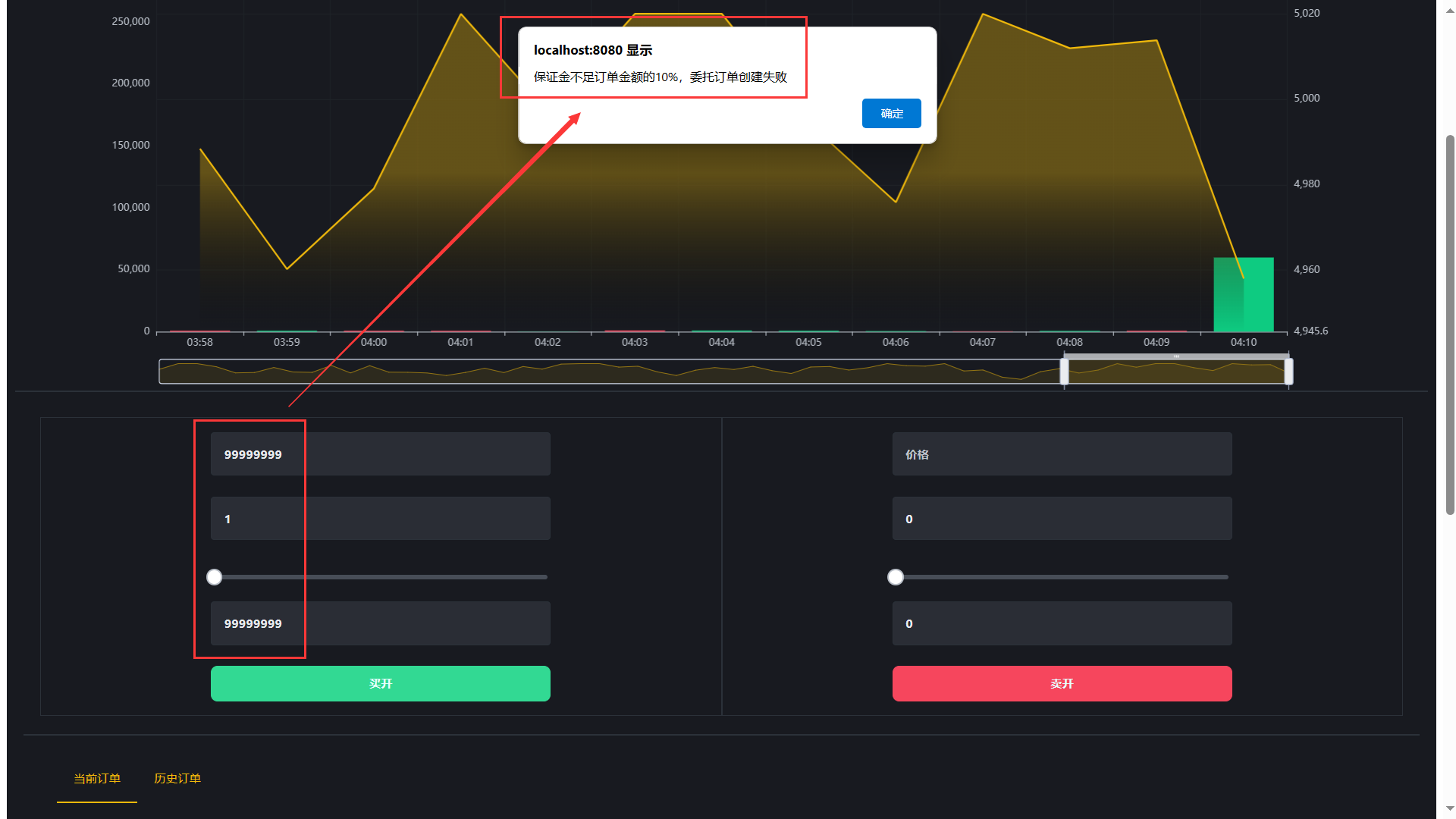
**图7-6 测试结果（价格为4950，数量为10，总金额为49500）**



**图7-7 测试结果（价格为-4950）**



**图7-8 测试结果（数量为-10）**



**图7-9 测试结果（总金额为99999999）**

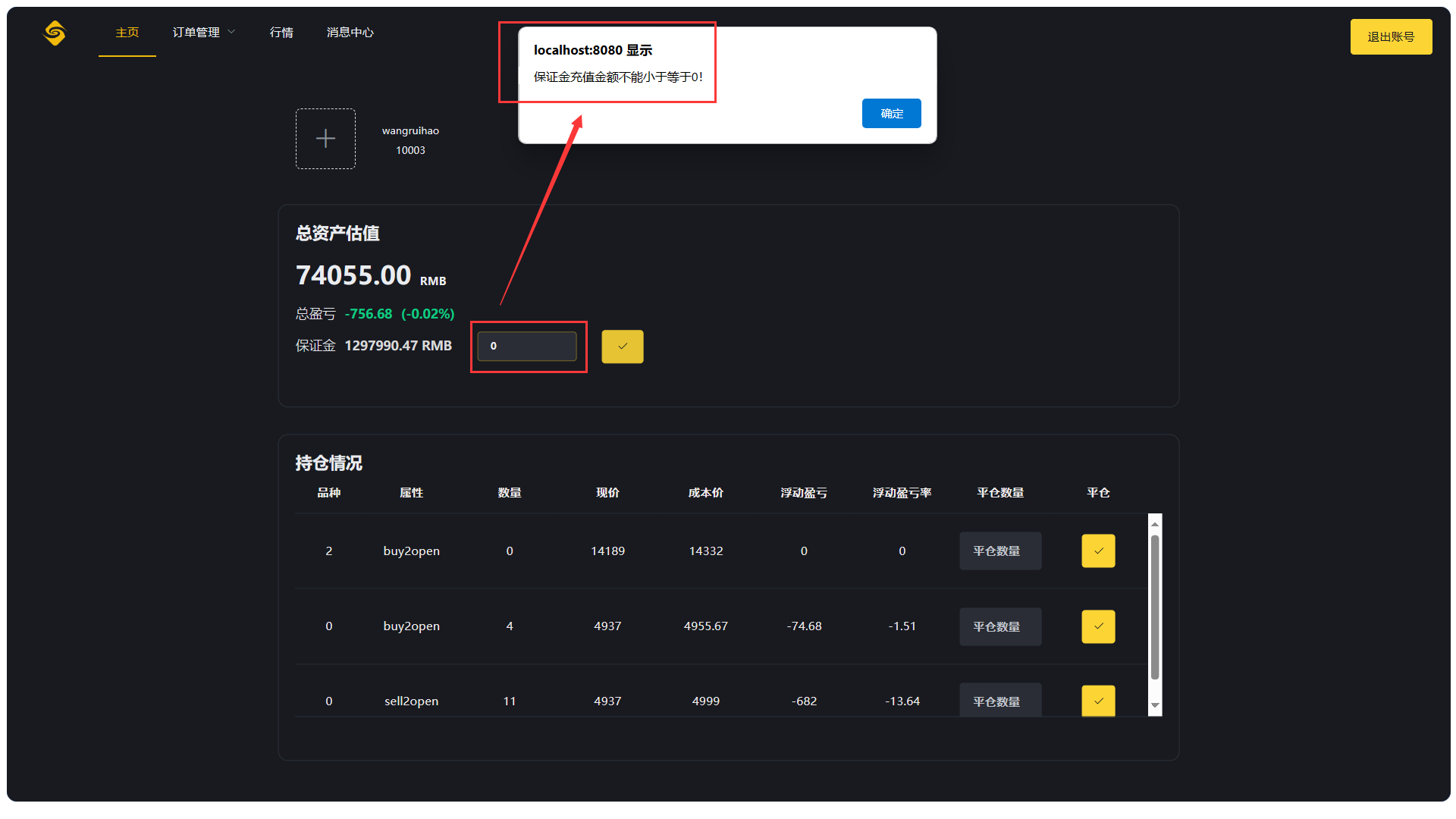
* 1. 边界值分析法

### 保证金充值

**测试用例：**

1. 充值数额为0

**测试结果：**



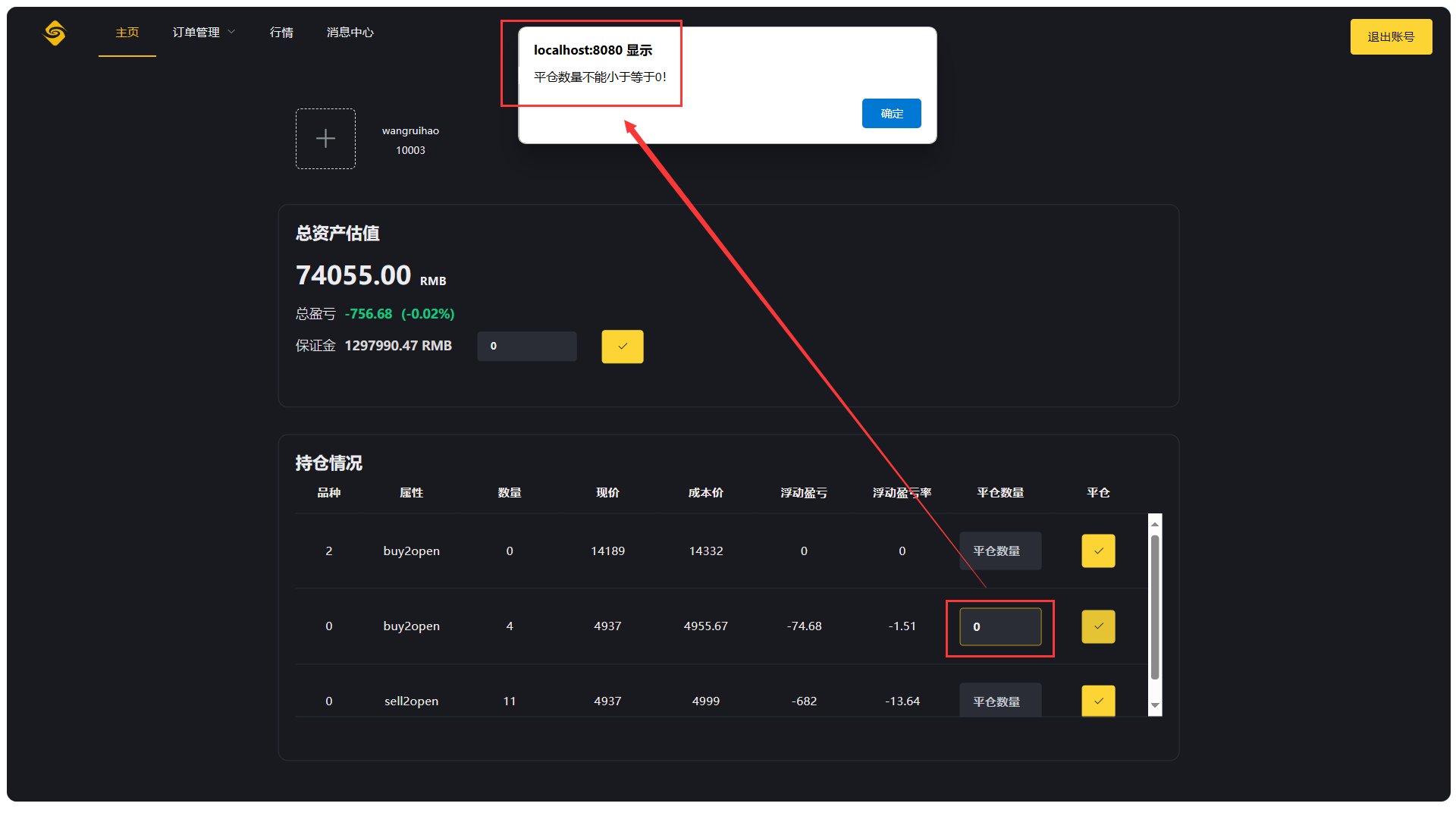
**图7-10 测试结果（充值金额为0）**

### 平仓交易

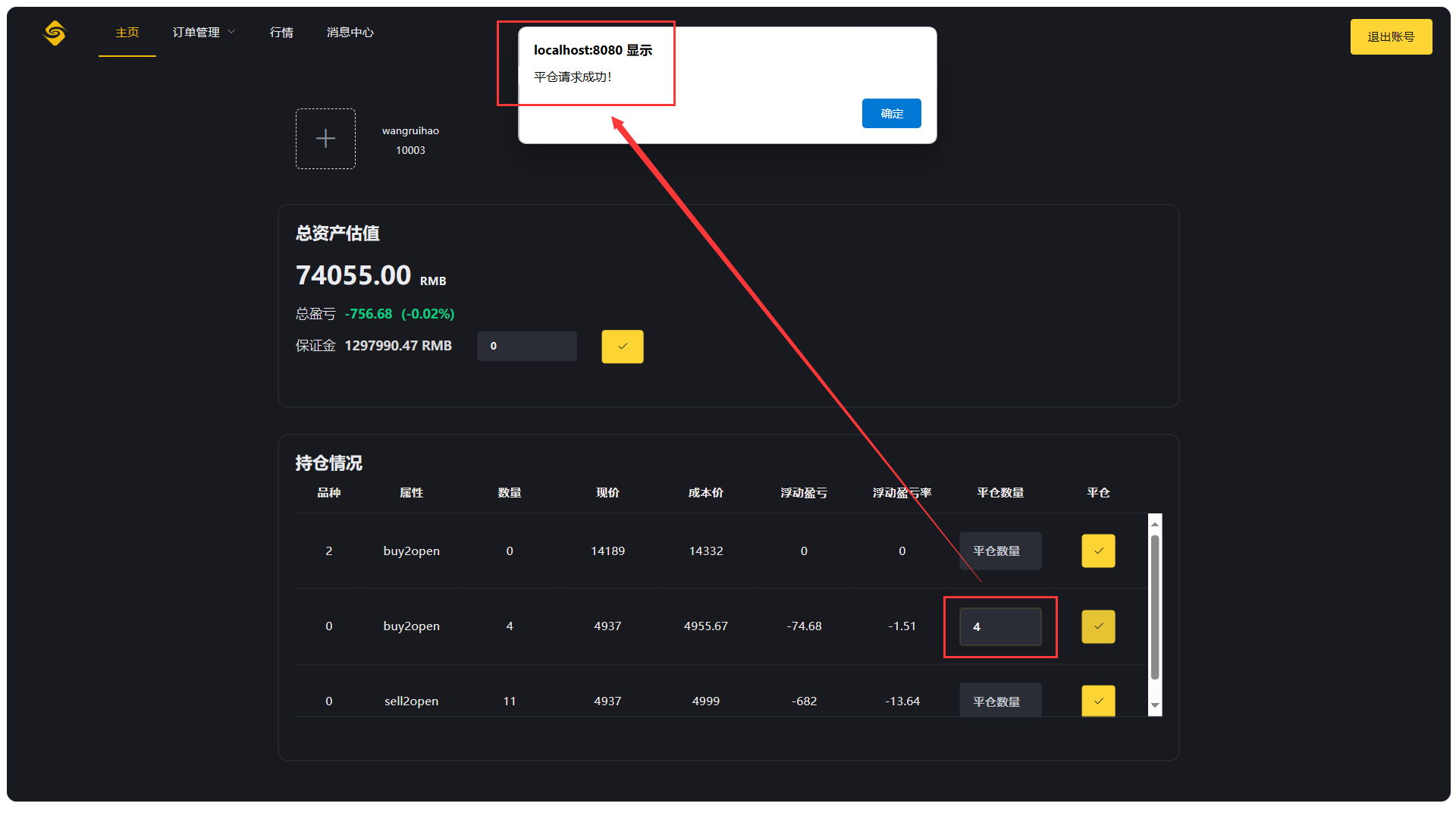
**测试用例：**

1. 平仓数量为0
2. 平仓数量为持仓量

**测试结果：**



**图7-11 测试结果（平仓数量为0）**



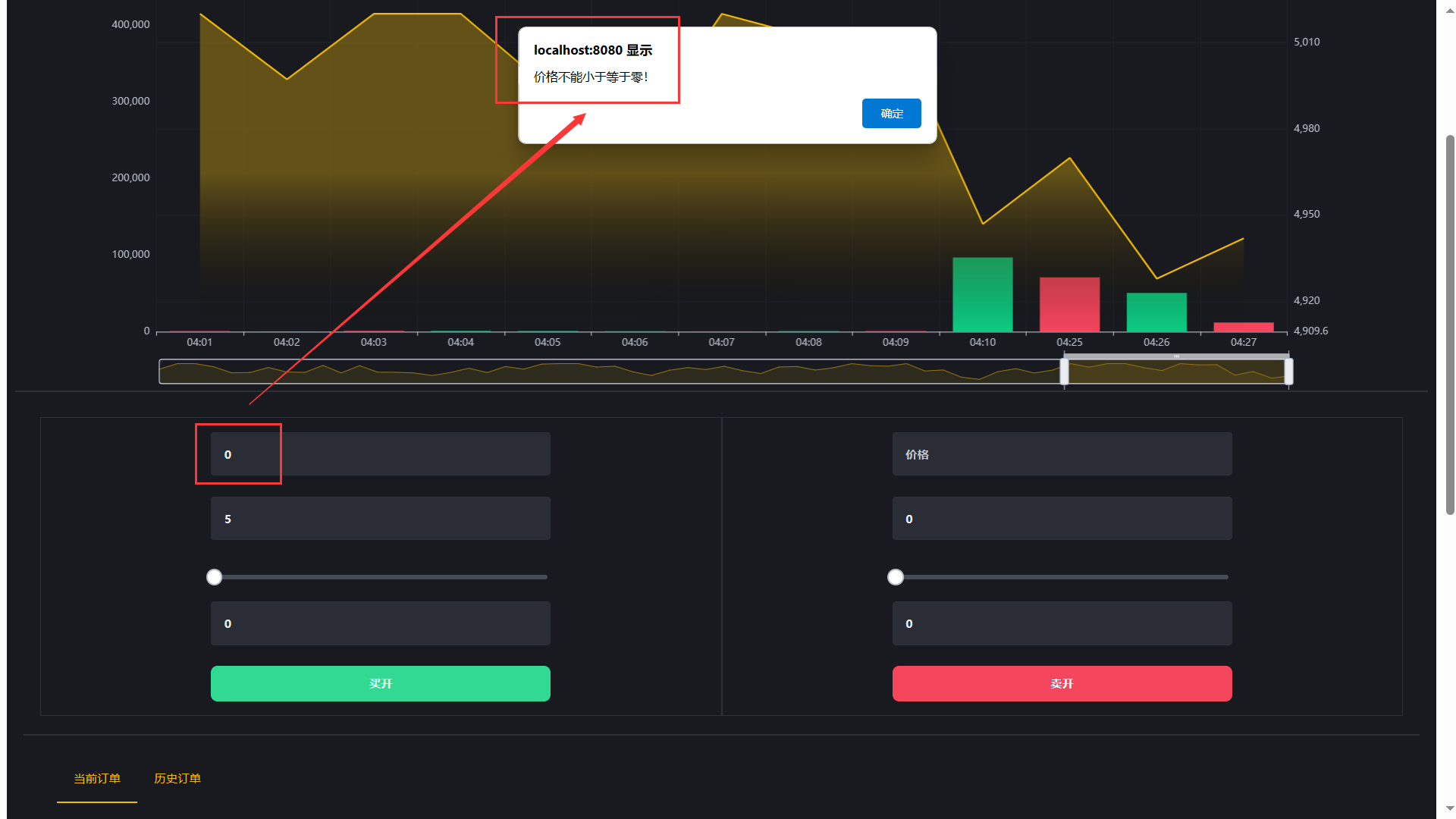
**图7-12 测试结果（平仓数量为持仓量）**

### 开仓交易

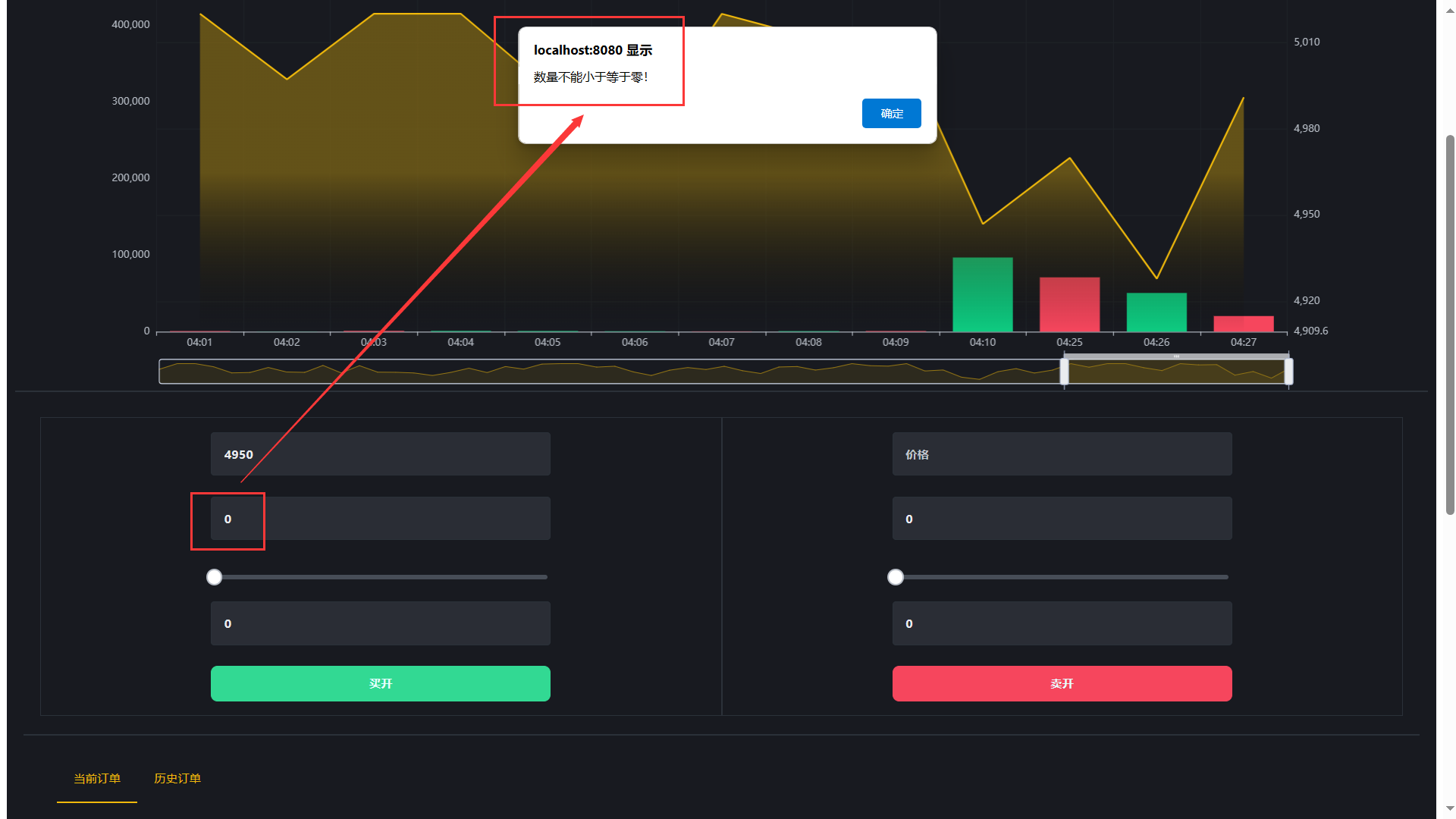
**测试用例：**

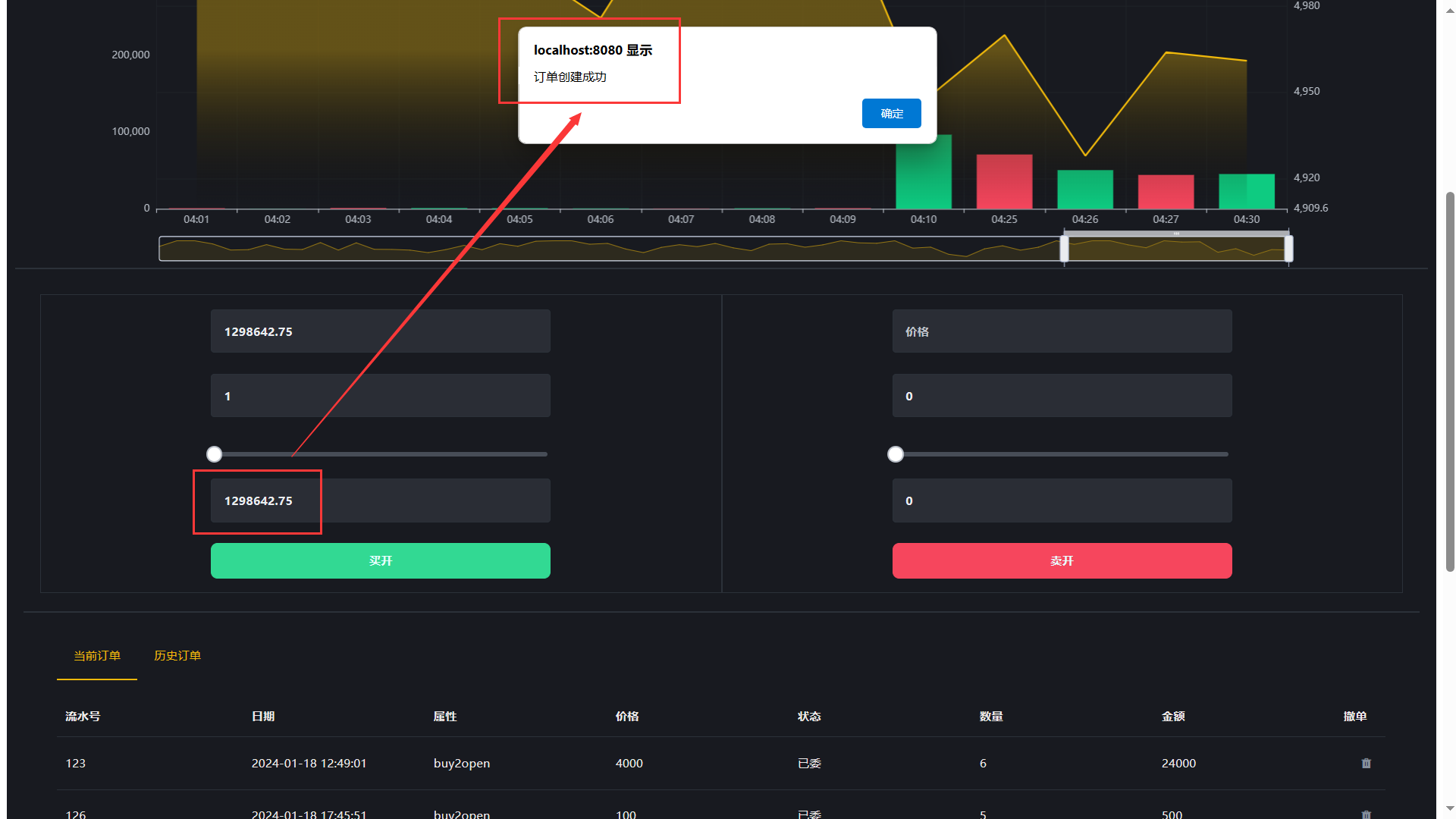
1. 价格为0
2. 数量为0
3. 总金额的10%等于保证金

**测试结果：**



**图7-13 测试结果（价格为0）**

**图7-14 测试结果（数量为0）**



**图7-15 测试结果（总金额等于保证金）**

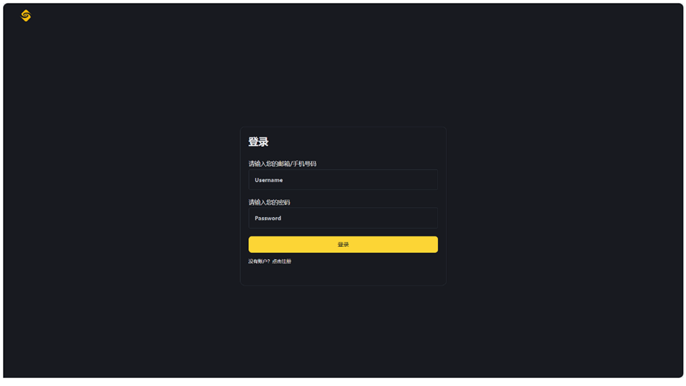
1. 用户使用说明书
   1. 系统功能概述

本期货交易系统用户角色拥有的功能有：注册、登录、查看个人资产、查看盈亏、查看持仓情况、查看期货总览信息、查看期货走势图、进行开仓操作、进行平仓操作、查看当前订单、查看历史订单等。

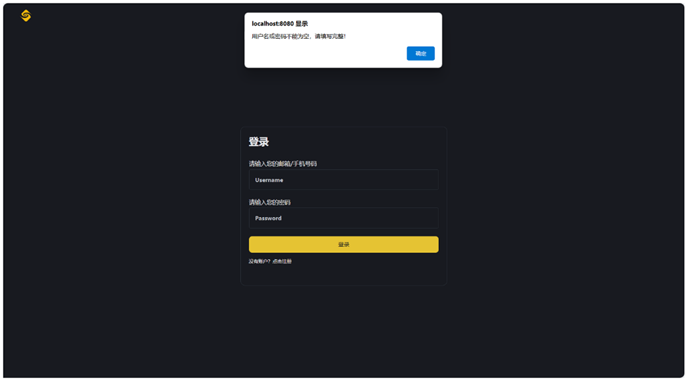
* 1. 系统使用说明

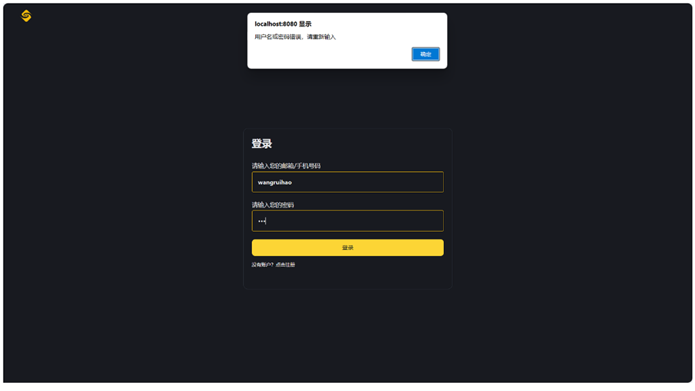
### 登录

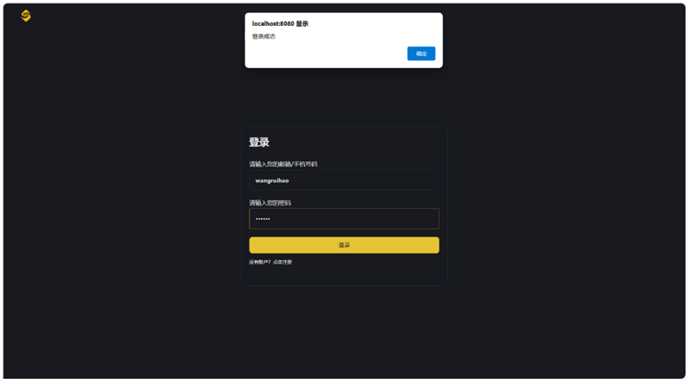
用户登录界面如图所示。用户可在用户名输入框中输入用户名，在密码输入框中输入密码，然后点击“登录”按钮进行登录。如果出现用户名或密码输入为空、用户名和密码不匹配等情况，登录会失败。



**图8-1 登录界面**

**图8-2 用户登录界面（用户名或密码为空）**

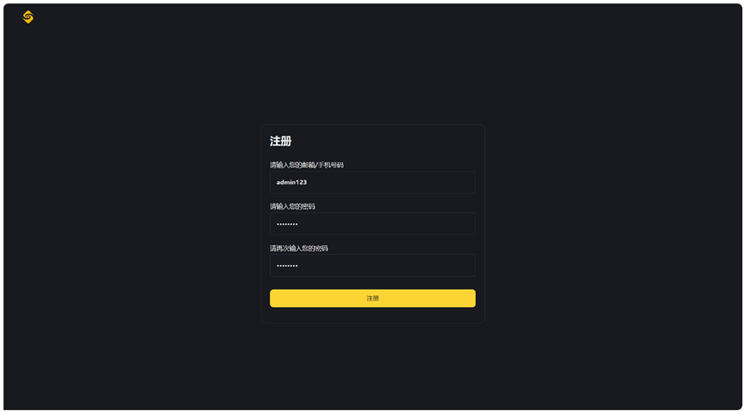
**图8-3 用户登录界面（用户名或密码错误）**



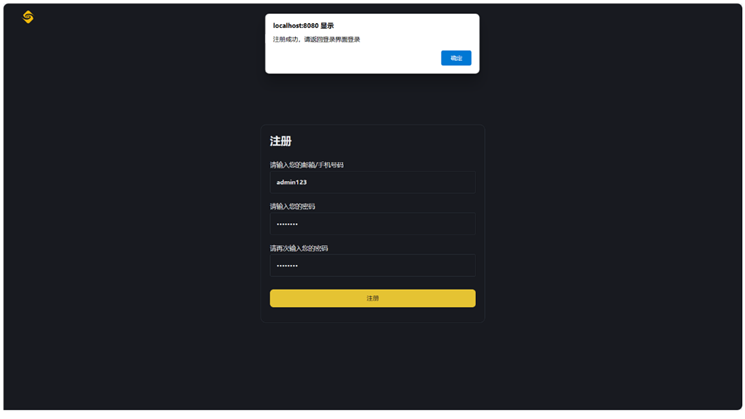
**图8-4 用户登录界面（登录成功）**

### 注册

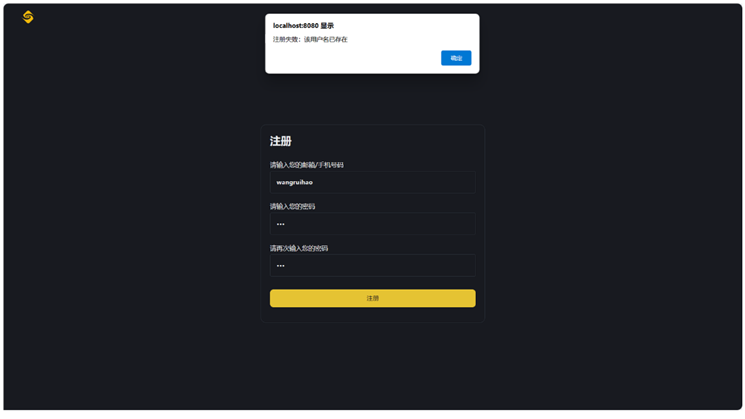
用户注册界面如图所示。用户可在用户名输入框中输入用户名，在密码输入框中输入密码，在确认密码输入框中再次输入密码，然后点击“注册”按钮进行注册。如果出现用户名重复、两次输入密码不匹配等情况，注册会失败。



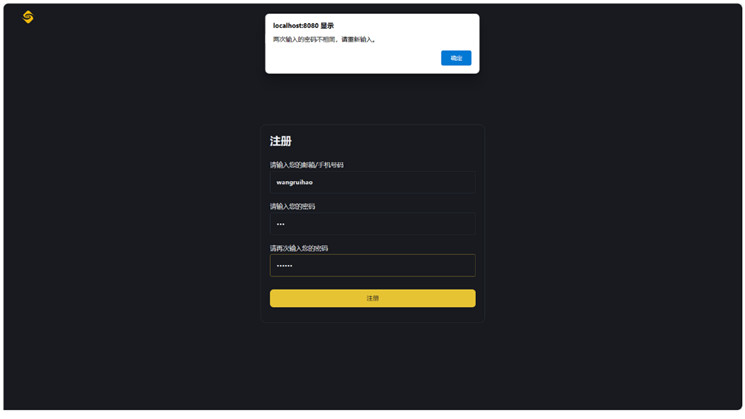
**图8-5 用户注册界面**



**图8-6 用户注册界面（注册成功）**



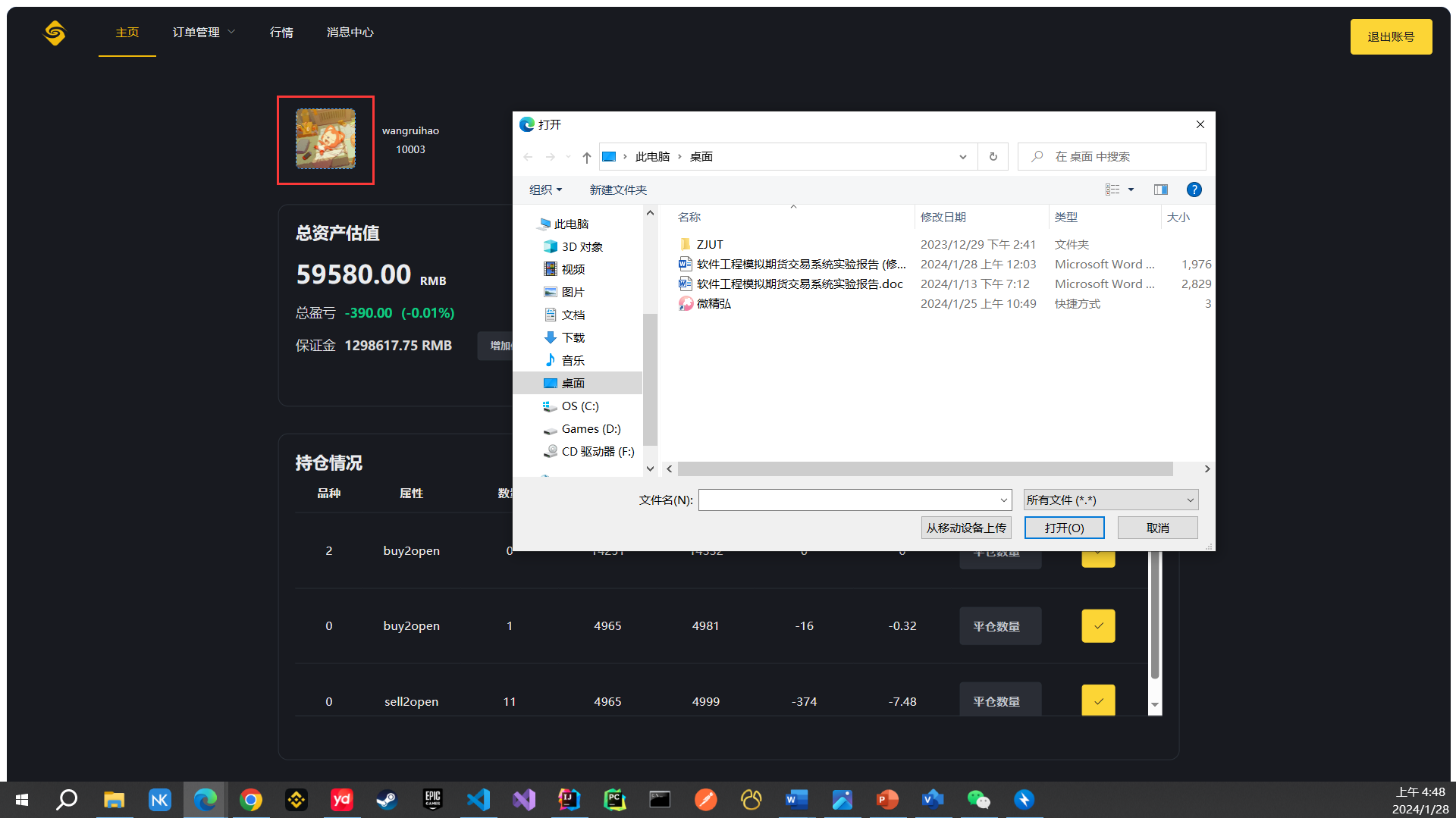
**图8-7 用户注册界面（注册失败：用户名已存在）**



**图8-8 用户注册界面（注册失败：两次输入的密码不相同）**

### 上传头像

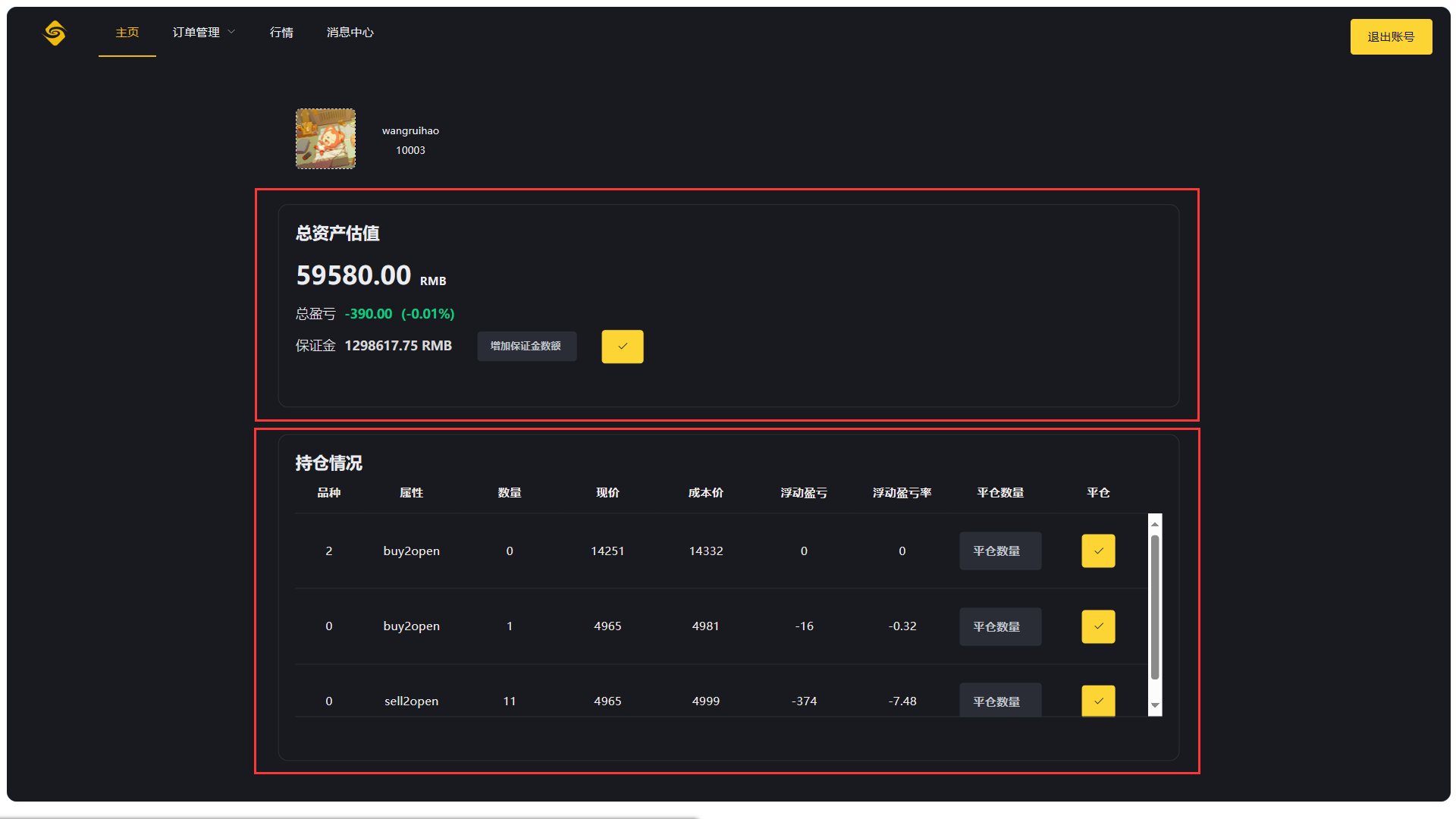
用户登录后可点击头像框更换头像。



**图8-9 用户头像上传**

### 查看个人信息

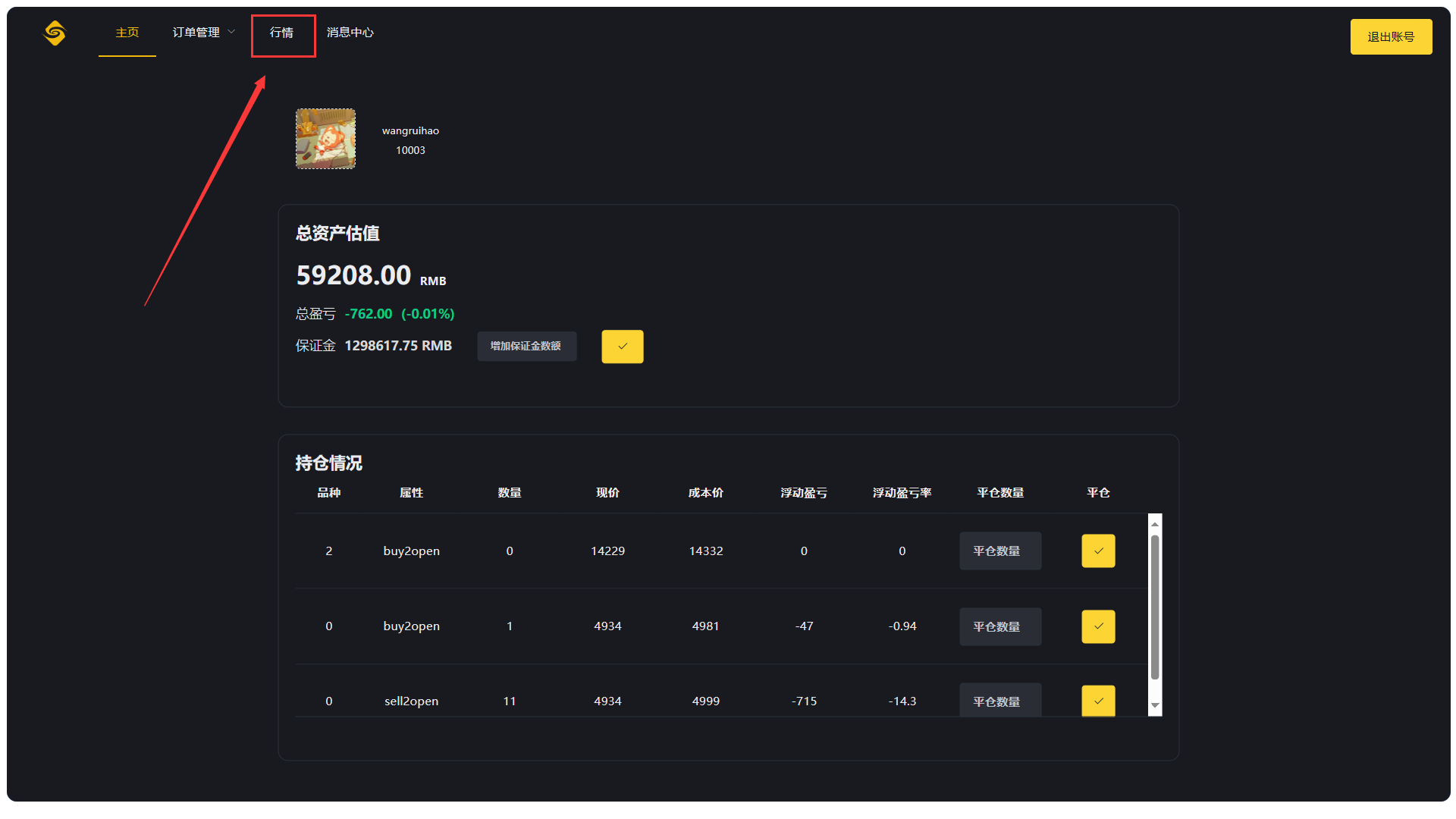
用户进入主页时，可以查看自己的总资产估值、持仓情况等个人信息。



**图8-10 查看个人资料**

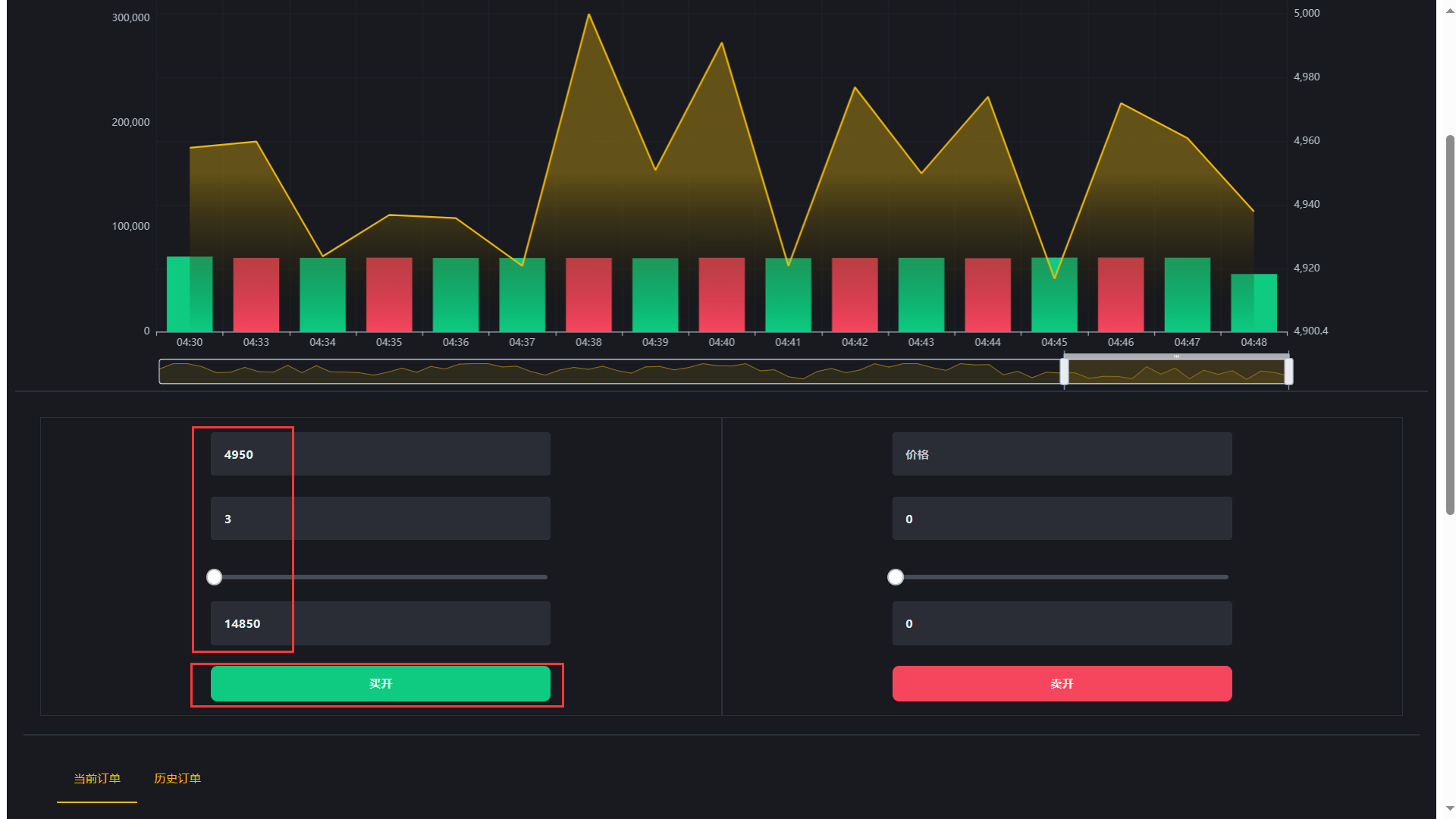
### 开仓

用户点击各页面上方的主菜单栏中的“行情”界面查看期货行情总览，在期货行情总览界面选择想要交易的期货，并点击进入该期货的期货行情详情界面。

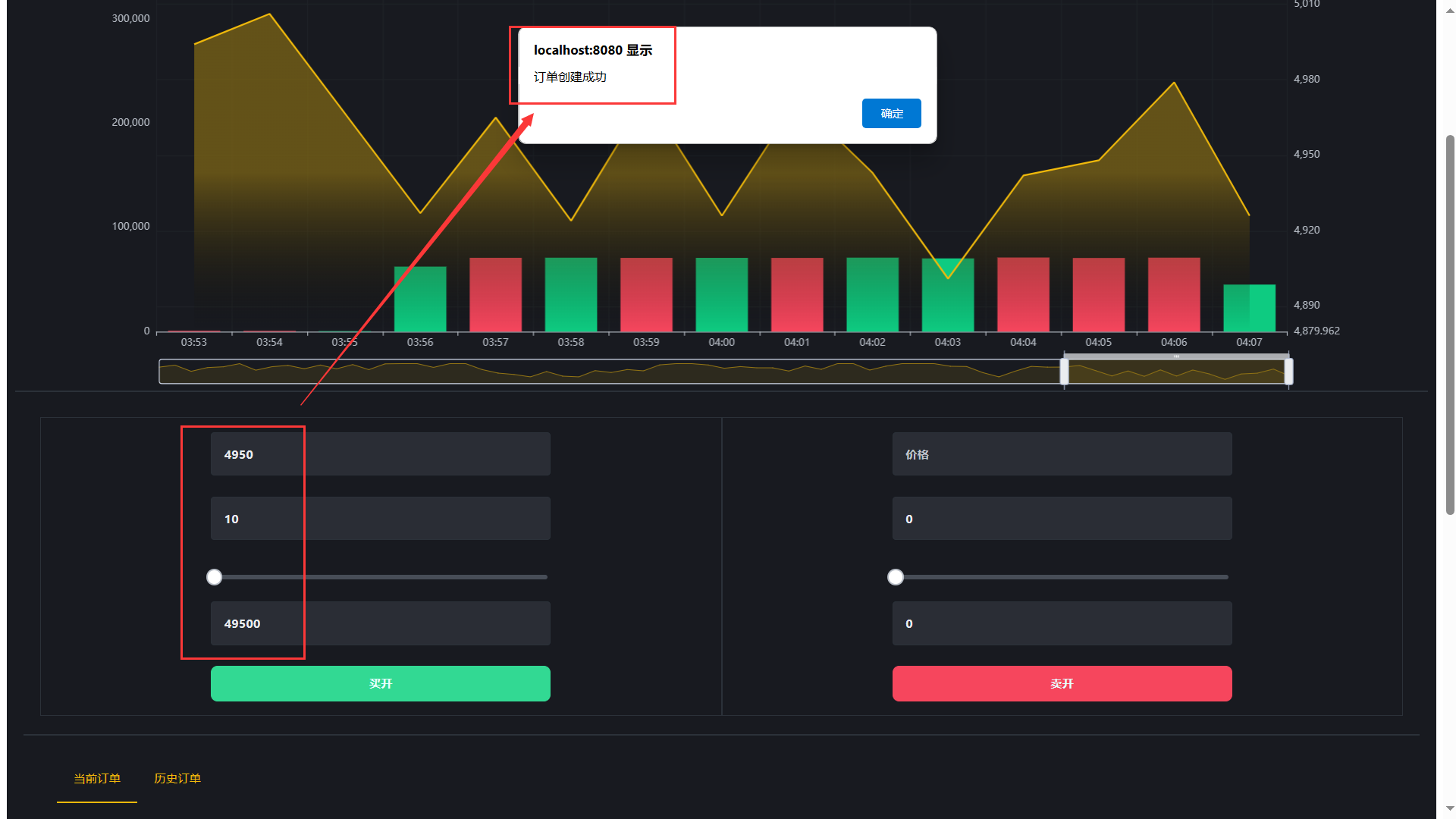


**图8-11 行情界面入口（图示红框处）**

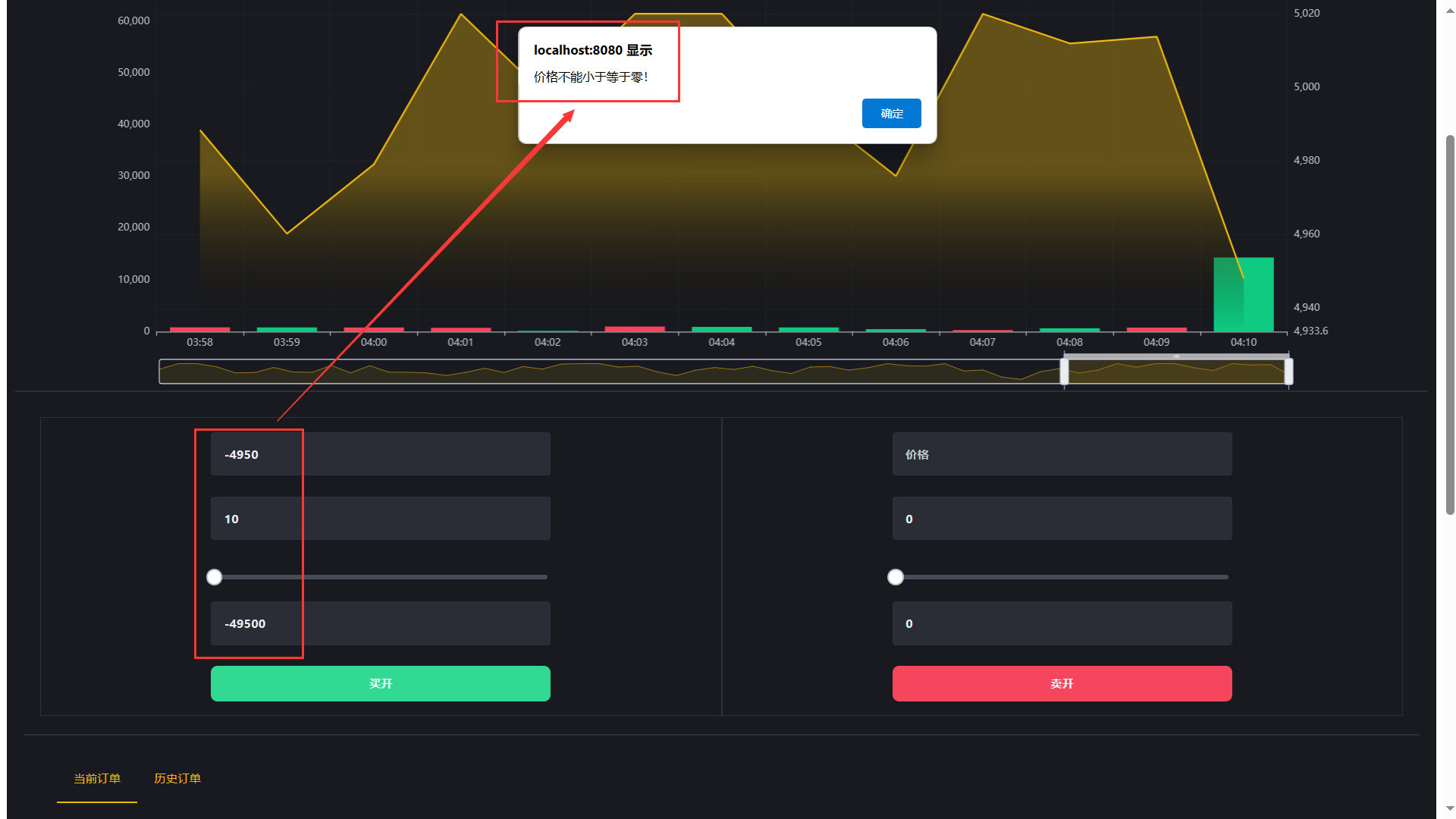
用户期货行情详情界面下方的交易板块中输入委托价格和委托数量，并点击交易按钮，实现委托订单的创建。如果出现价格小于等于0、数量小于等于0、总金额的10%超过保证金等情况，委托订单的创建会失败。



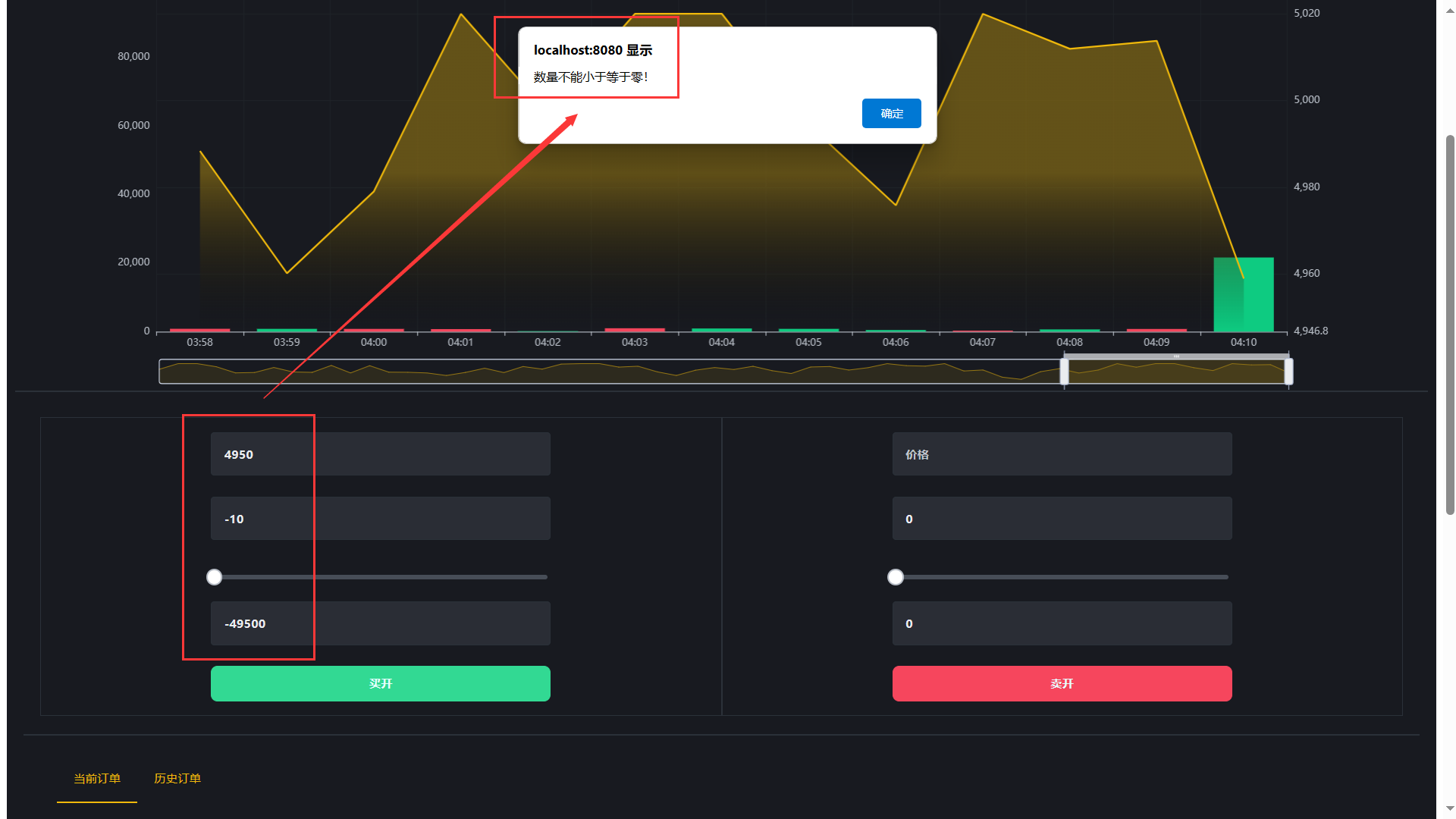
**图8-12 开仓板块**



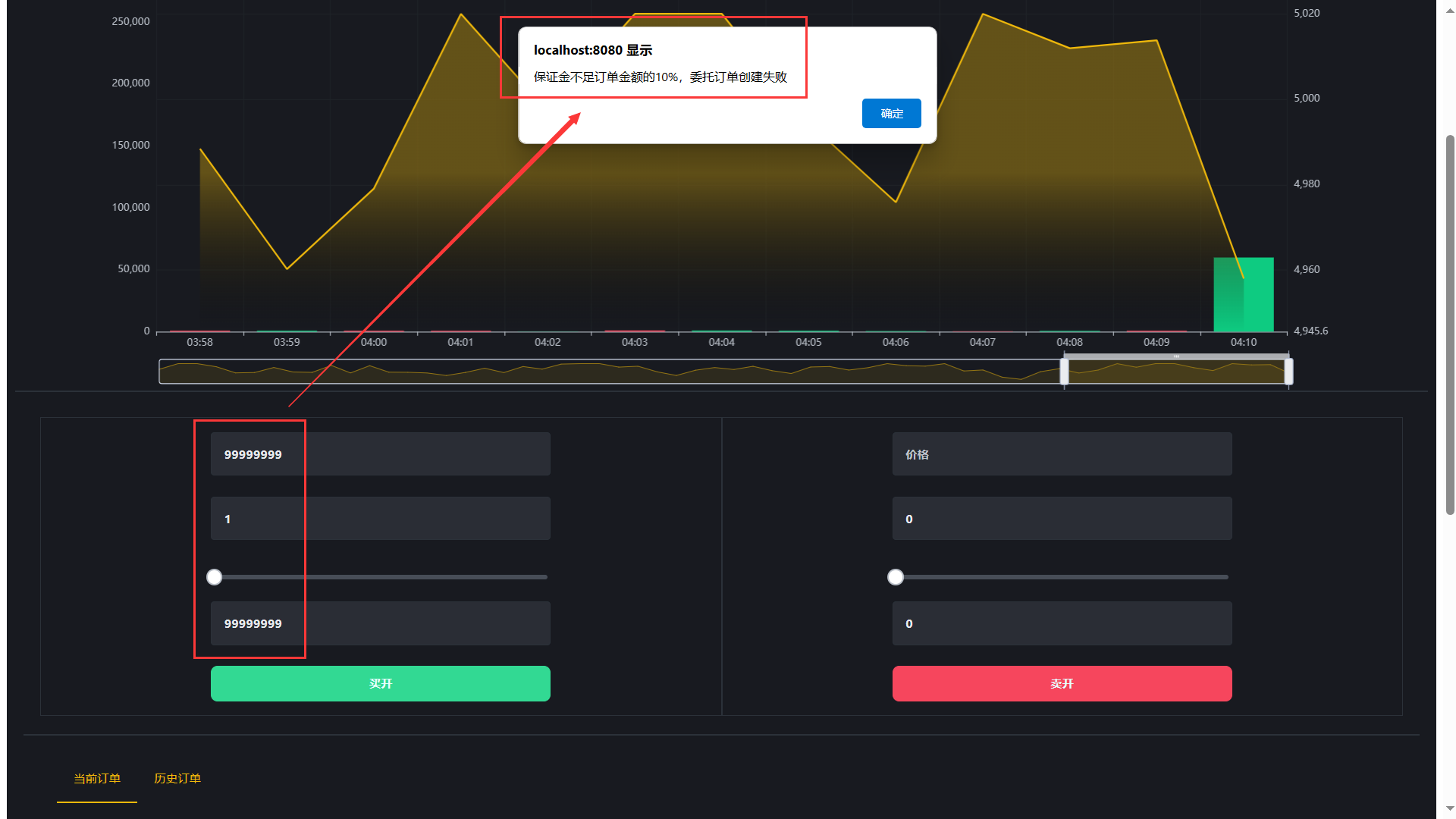
**图8-13 委托订单创建成功**



**图8-14 委托订单创建失败（价格为-4950）**



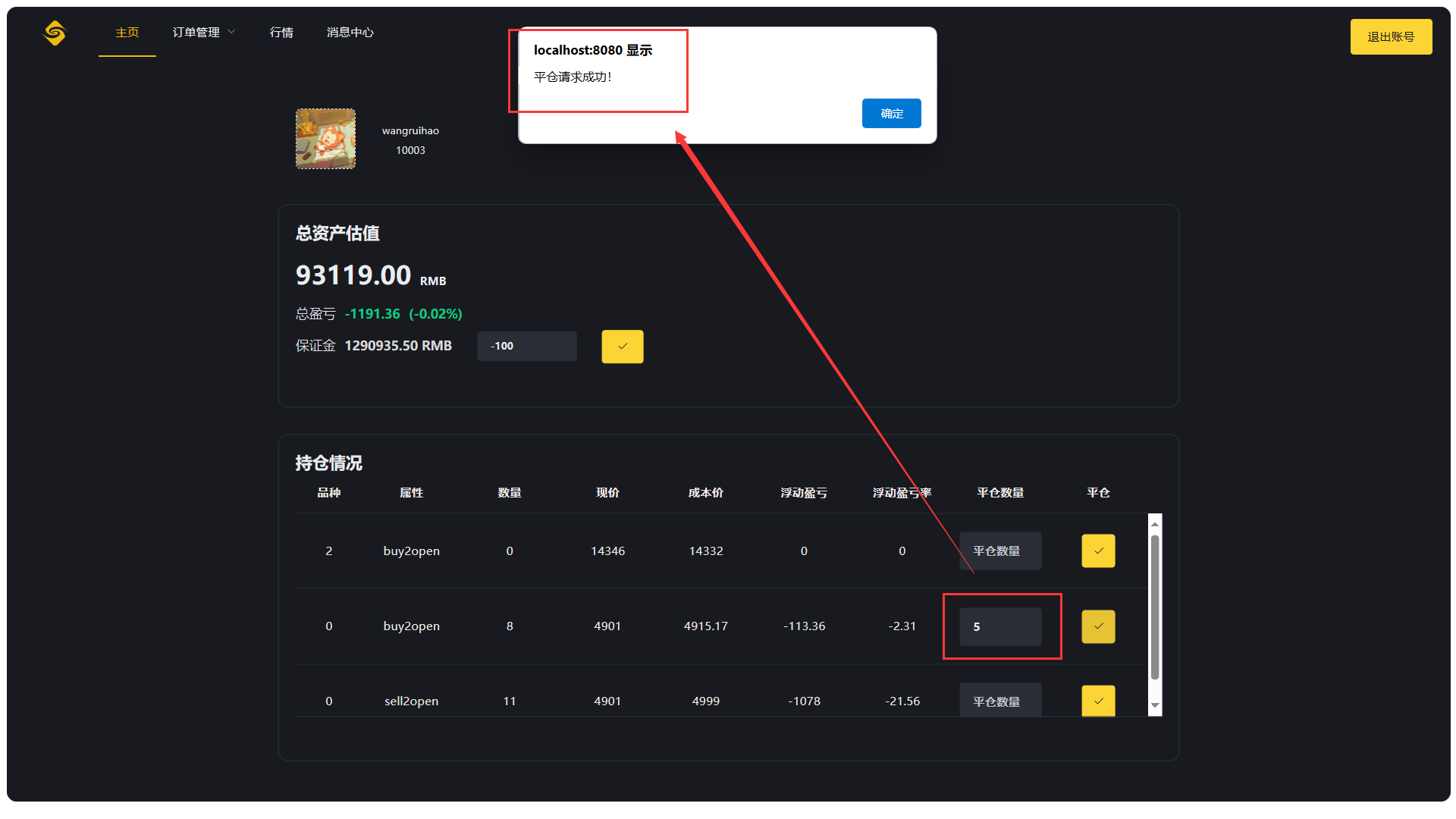
**图8-15 委托订单创建失败（数量为-10）**



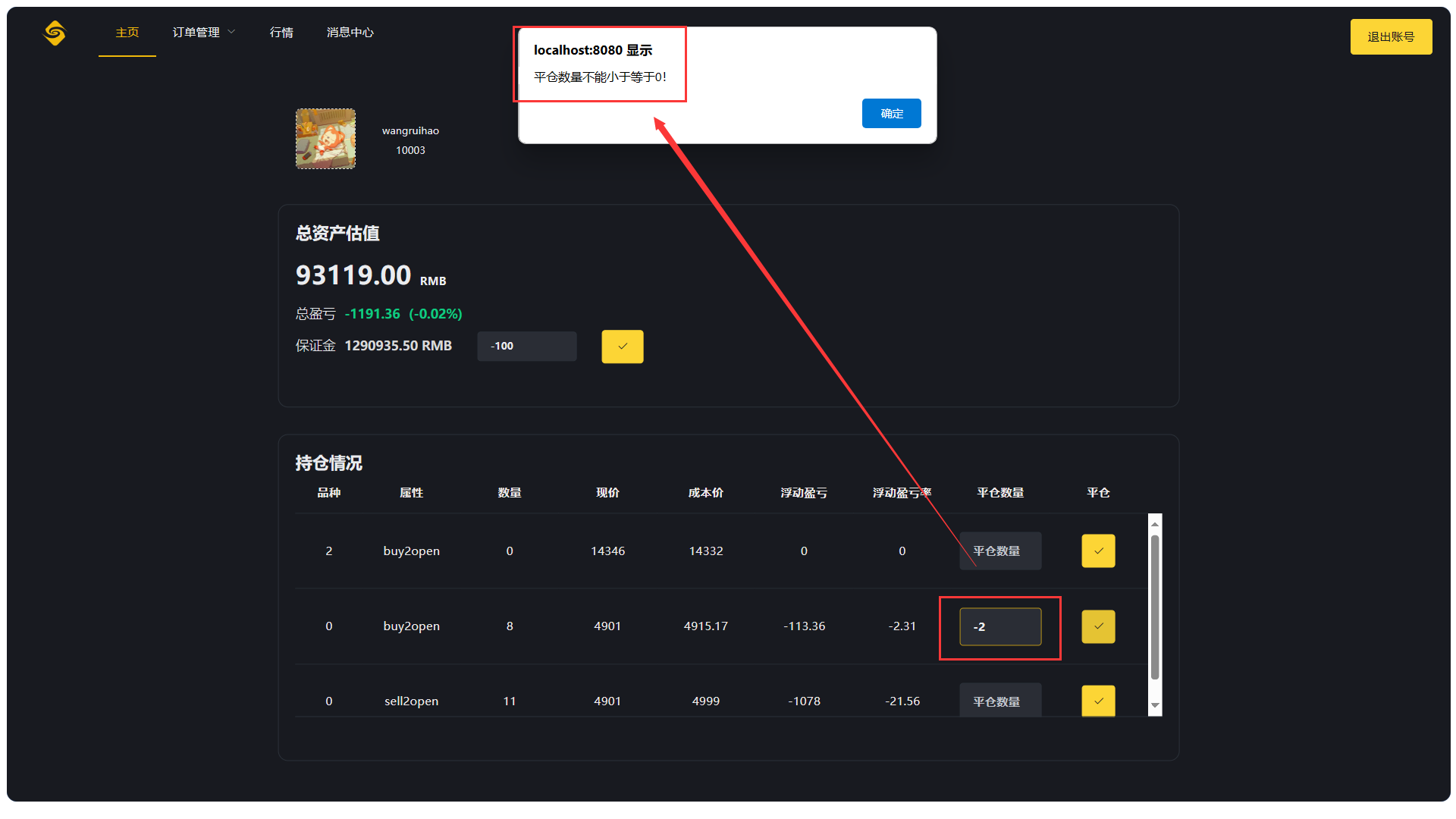
**图8-16 委托订单创建失败（总金额为99999999）**

### 平仓

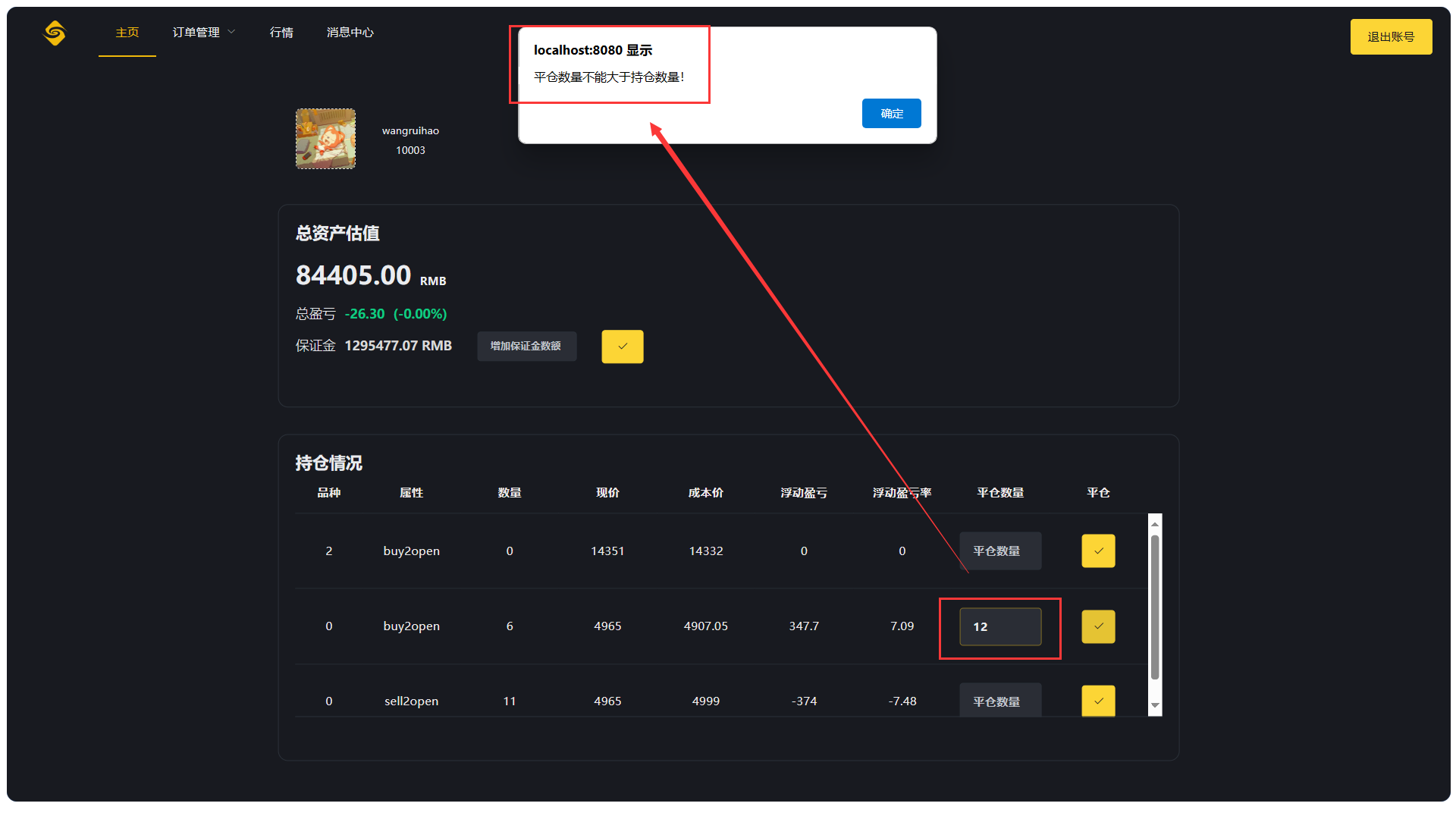
用户可在主页的持仓信息板块中输入平仓数量并点击平仓按钮实现平仓。如果平仓数量小于等于0，或平仓数量大于持仓数，那么平仓会失败。



**图8-17平仓成功**



**图8-18 平仓失败（平仓数量小于等于0）**

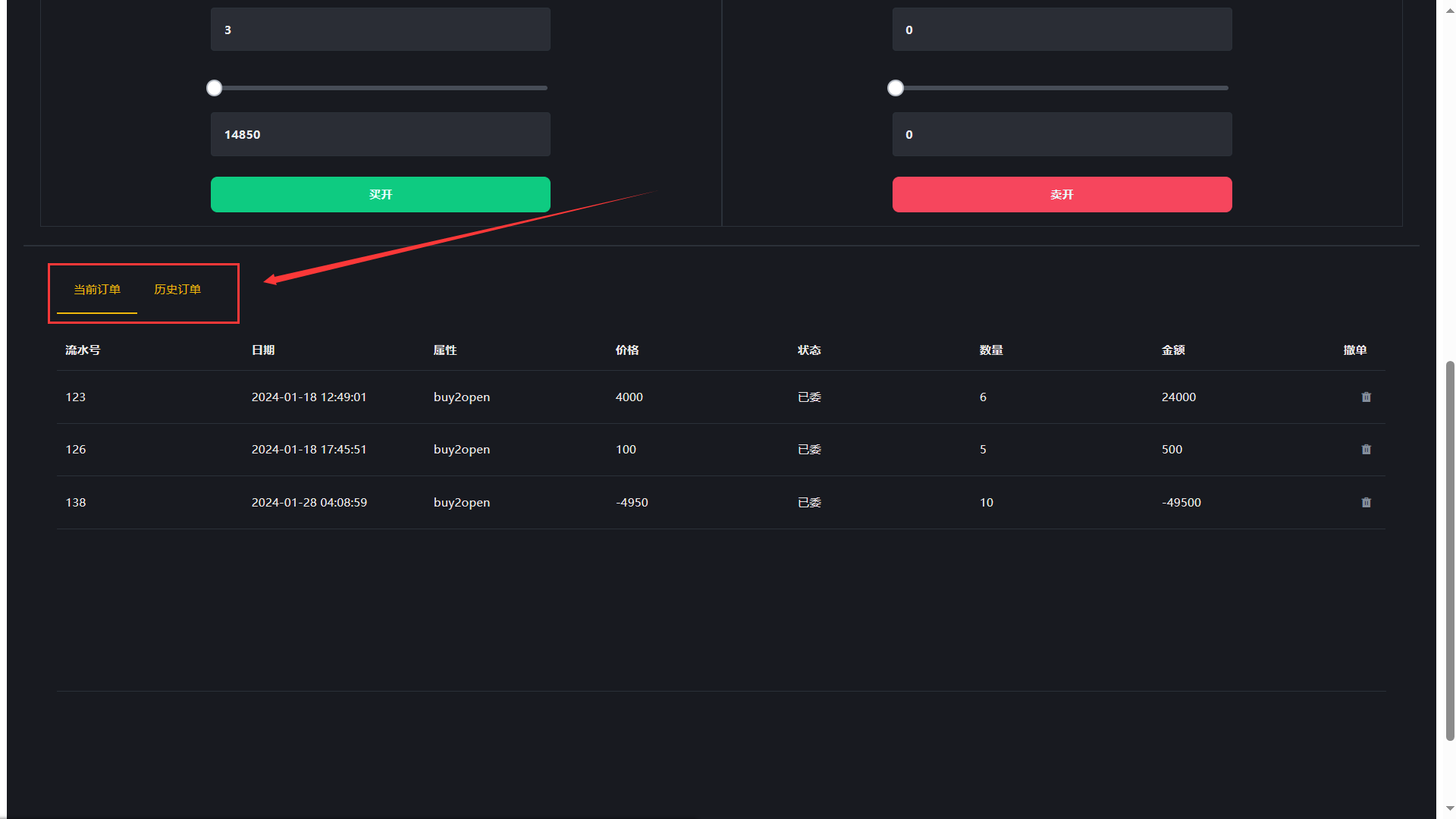


**图8-19 平仓失败（平仓数量大于持仓量）**

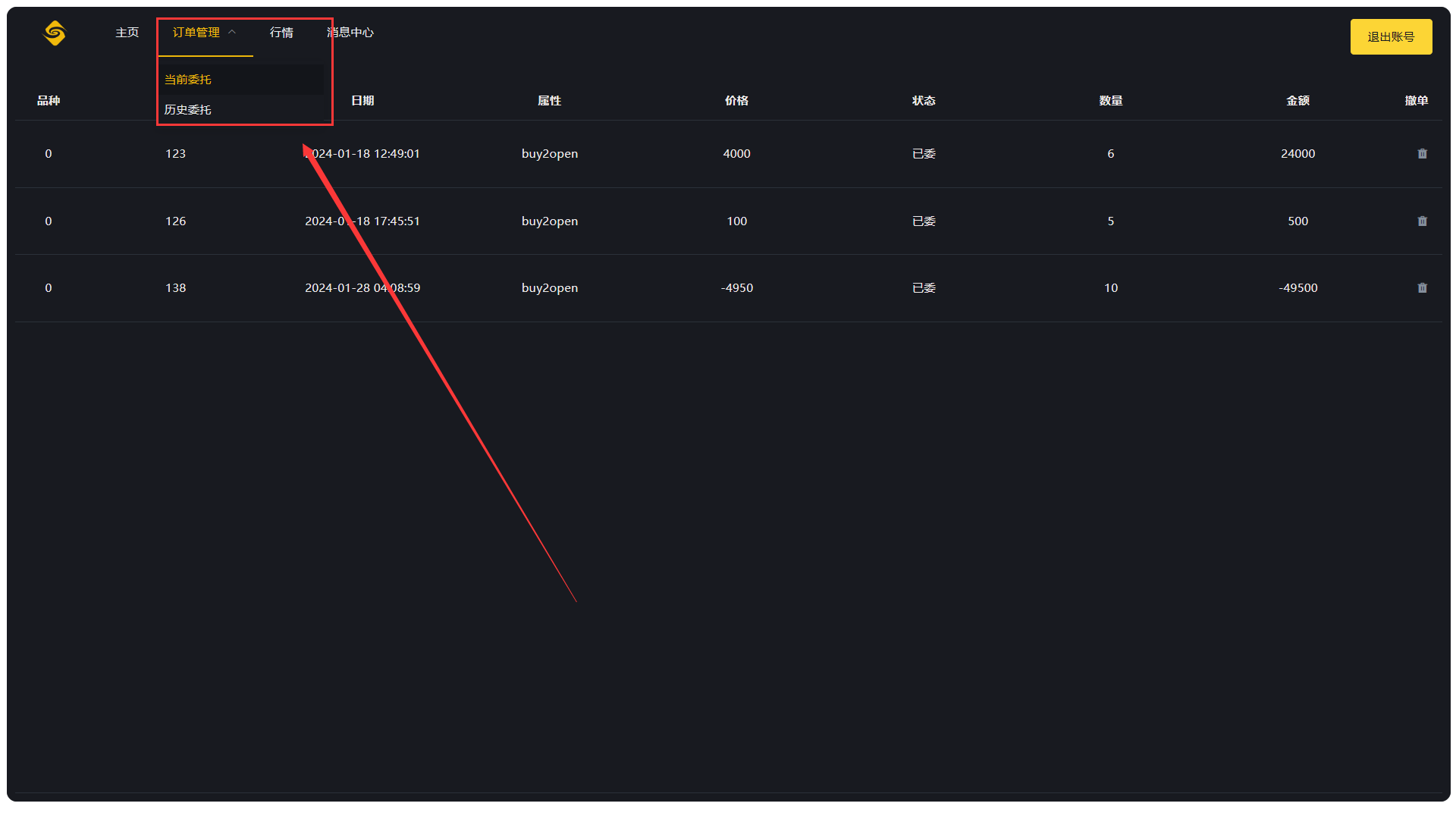
### 查看订单

用户可在期货行情详情界面查看当前期货的订单，包括历史订单和当前订单。

如果用户希望查看所有订单，可以在导航栏中点击“订单管理”并选择需要查看的订单类型（“历史订单”和“当前订单”），从而查看对应的订单。



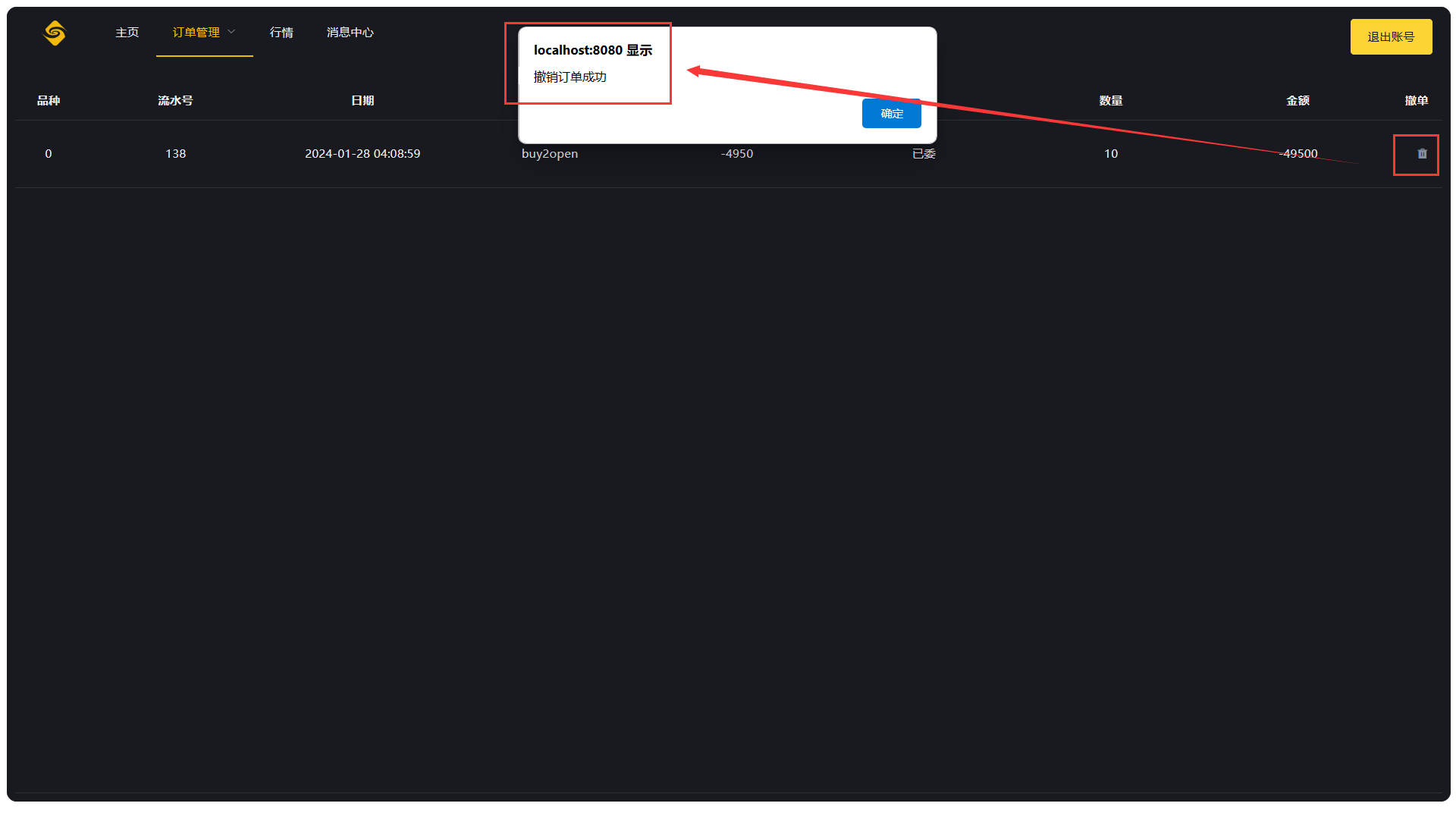
**图8-20 当前期货订单板块（图示红框为切换订单类型菜单栏）**



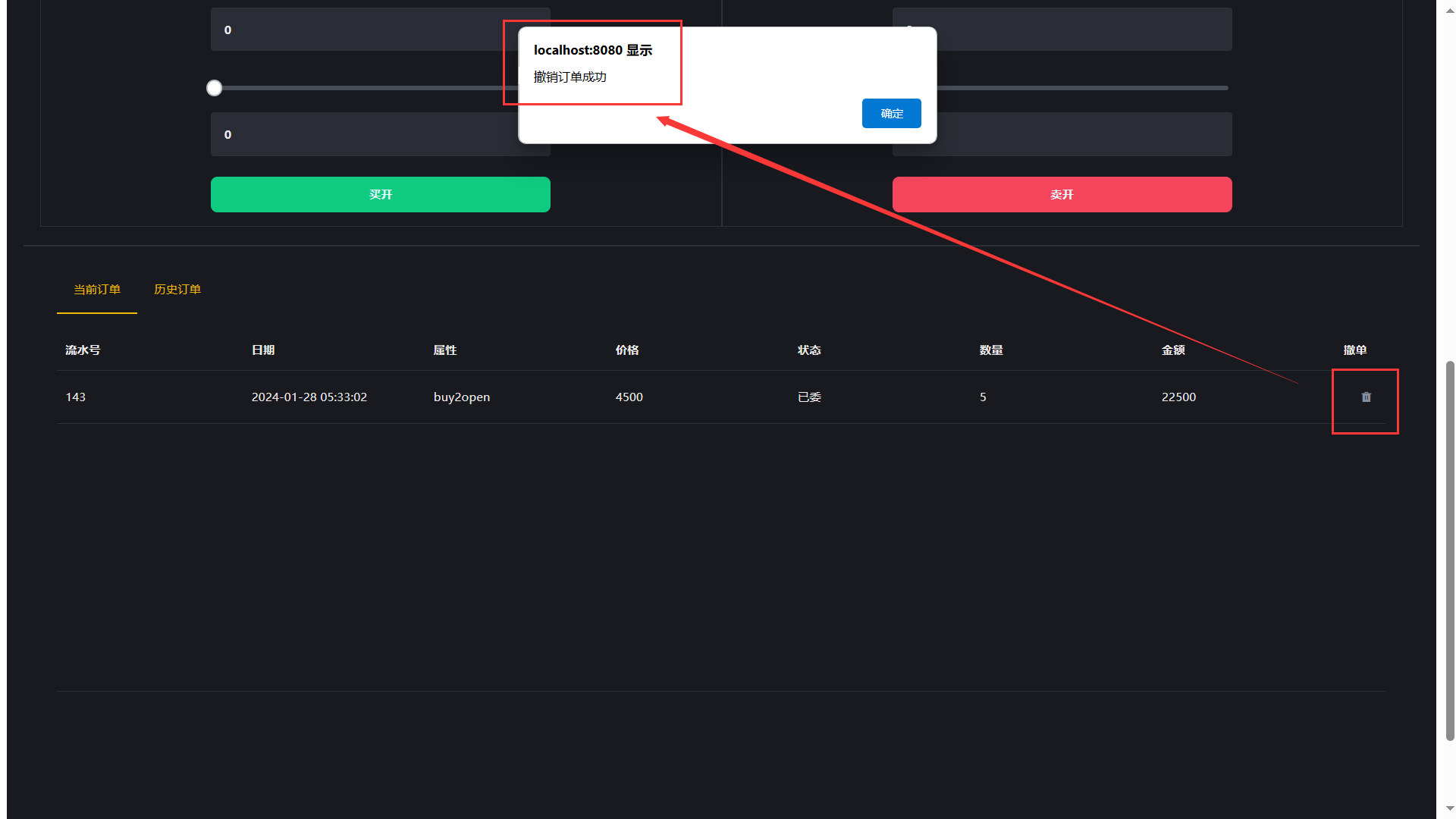
**图8-21 所有期货订单页面（图示红框为该页面的入口）**

### 撤单

用户可以在当前订单板块（包括当前期货当前订单和所有期货当前订单）中通过点击撤单按钮实现撤单。



**图8-22 撤销订单成功（所有期货当前订单界面）**



**图8-23 撤销订单成功（当前期货当前订单界面）**

1. 小组项目的体会与心得
   1. Mcrivers

毫无疑问地，软件工程课程的课程设计是我自2021年入学以来接触的难度最大、项目复杂度最高、代码量最大、工作量最多的一个项目。而我又在项目中担任了“全栈开发工程师”的角色，全面负责前后端开发、前后端对接、项目调试、项目测试等工作。这对于从来没有项目开发经验的我来说，无疑是一个巨大的挑战。

项目的起步并不顺利，组内的六个成员，对JavaEE开发的掌握程度差异大，代码水平也层次不齐，这就导致分工困难；又因为人数较多的原因，初期在敲定项目选题、进行需求分析、项目功能设计时消耗了大量的时间统一意见。同时，因为这学期大家的课业都十分繁忙，几乎整个学期都在忙于应付各个课程的各种ddl，我更是在学期初由于参加中国机器人大赛缺席了数周的课程，所以项目起步时间很晚，直到十二月下旬才启动。在有几名成员代码水平一般的情况下，想要在一个月内完成一个大型项目，确实不是一件易事。

我们的项目选题“期货交易模拟系统”并不常见，网上并没有类似的项目可以作为模板，这意味着我们必须从零开始。为了实现项目的快速开发，我们并没有在项目开始前做非常完整的需求分析，而是采用了软件工程中经典的“敏捷过程开发”，即“个体和交互胜过过程和工具，可以工作的软件胜过面面俱到的文件，客户合作胜过合同谈判，响应变化胜过遵循计划”，快速地分别开发出了第一版的前端页面和后端服务器。

在项目刚起步时，我负责的是前端的开发。在我的想象中，似乎在前后端分离的开发项目中，只需要前端工程师和后端工程师商量好接口的形式，再稍加调试，系统就可以正常运行了。然而，但是在我耗时两周时间开发出前端页面，转而将目光转向后端服务器时，才发现情况和我的想象相去甚远。由于我没有很好地向负责后端开发的同学说明前端的接口需求，导致负责后端开发的同学对项目逻辑产生了误解，项目并不能成功运行。而两人之间的信息交流一定会有信息损失，从而降低开发效率，延长开发周期。而ddl临近，为了尽快完成项目，我决定加入后端开发工作，正式化身为所谓“全栈开发工程师”。

好在，原本负责后端开发同学得知有相当多代码要返工以适应前端需求时，并没有灰心丧气，丧失斗志，而是积极地和我一起投入到后端的开发工作中来。多了一个劳动力，又有了更加明确的需求，所以后端推进的速度有了较大的提高。在这个过程中，我重新设计了数据库、重新编写了后端一些操作的实现逻辑，并全面负责了前后端接口的对接工作。最终，在我们小组成员的齐心协力之下，项目终于顺利完成，并通过了各项测试。

本次实验让我收获许多。

首先，比代码开发更重要的，是良好的分工合作。没有分工合作的项目组，就像无法让进程并发执行的操作系统，不仅效率低下，还造成了极大的资源浪费。

其次，前端工程师和后端工程师都需要对项目需求有一个明确的认识，以实现前后端的分析开发，尽可能避免前后端“扯皮”的情况发生。

最后，一切工作最好尽早完成，拒绝拖沓，否则只会忙中出错，越忙越错。

* 1. A

本次实验是模拟期货交易系统,我做的后端的部分代码,虽然不多但是通过这次实验我对springboot框架及mvc设计模式以及为什么要使用框架有了更多的认识。

我个人认为spring的解耦做的是真的好,目前做完这次实验我对spring的认识是它就好像是一个大工厂,他能创建对象并维护这些对象的生命周期,这对于像adminservice,admindao这种只需要创建一次的工具类非常合适,这个控制权不在action里了.后期要给action加一个工具类或者把这个工具类换成实现了同一接口的另一个工具类将不再需要在java代码里进行修改,只需要修改spring配置文件即可.

这次实验开发几经波折,起初是想要做一个类似于网上TEX交易练习者的模拟交易平台,但由于网络编程方面知识的欠缺和这个项目的核心功能和javaee相关度不是很大,所以改成了一个拥有基础交易功能的模拟期货交易平台,这个项目的进行过程中也遇到了很多问题,初步是数据库不合理,导致某些功能进行起来非常困难,然后我们对数据库进行了重构.之后的过程就进行得相对比较顺利,但由于数据库表的设计导致了后期某些功能怎么做都不太合理.在这次实验中我充分地认识到了处理异常的重要性,无论如何都要把异常处理掉并把执行栈轨迹打印出来。

而且我深切地体会到了需求分析的重要性,如果起初的设计有问题那么后面很有可能在某些功能上面陷入僵局.,还有就是用户体验的问题,我的某些功能设计的确实用户体验不是很好,以后再开发过程中一定会注意不只是技术的问题还有功能用起来舒适与否。

* 1. B

在此次期货模拟交易平台的项目中，在项目实现中我主要负责后端开发部分，在开发过程我最大的感触是在项目开始的初期一定在先明确好需求分析，确定好基本的框架设计，明确各个api的命名以及前后端传输了分别哪些数据和他们的属性名。

我们项目在初期的需求分析时，并未深入理解期货的交易机制以及期货中各个专属名词所代表的含义和功能，只是很肤浅的依据市面上存在的期货交易软件和炒币软件，将它们的部分内容进行结合，形成了我们项目初期的需求分析以及功能设计，在后面的开发过程中并未再进行深入的再需求分析和功能分析以及api和属性名的命名规范。

因为在初期并未做好详细的基础功能设计，加上我对springboot开发框架的不熟悉不熟练导致了在后端开发进度较为缓慢，不过也算是慢慢的开发出一些基础功能，像基本的数据库设计和一些简单的查找操作，在实现了基础的开仓，以及查找更新删除功能开发后，由于另一个项目较为紧急，所以我先去进行另一个项目的后端开发操作，在回到此项目时，组员对后端代码以及数据库进行了优化，这里的优化是因为我在设计数据库时存在部分表内内容冗余以及代码中api的名字和属性名与前端开发不一致，所以在后面的开发新功能时，第一部便是明确好api的名字和属性名，把功能的需求分析落实到位后，接着在进行开发，这一转变让后端开发轻松了很多，基本不会在去花时间在一个已开发的功能修改完善上。

所以这次项目我自身的感受最深的便是在开发过程中，首先要明确的一定是需求分析，这个api的功能是什么，前端传什么样子的数据给后端，后端处理完之后传完什么样的数据给前端，这是重中之重，明确了一个api长什么样子能够在开发时少许多烦恼，初期明确的功能设计，也能避免已完成的api改来改去增加时间成本。

在完成项目的路上，受益良多，学习并渐渐熟悉了基础的springboot框架，让我后端开发的能力成长了许多，小组合作也让我学习到在软件开发中，哪些部分的讨论和明确是更为重要的，这些都为我在今后的发展打下了坚实的基础。

* 1. C

在完成这个项目时，我负责部分前端页面的开发，这对我来说是一次非常有价值的学习和经验积累的机会。首先，我深刻理解到了Spring Boot和Vue.js在现代Web应用开发中的重要性。

通过学习Spring Boot，我掌握了构建后端服务的关键技能。它的简洁性和快速开发特性让我能够迅速创建可靠的后端服务，并处理用户登录、注册、订单管理等核心功能。Spring Boot的强大功能和广泛的社区支持能够更高效地编写稳定的后端代码，这对于整个项目的成功至关重要。

另一方面，Vue.js作为一种流行的前端框架，为我提供了强大的工具来创建交互式和响应式的用户界面。通过Vue.js，我能够轻松地构建了订单界面、行情界面和用户界面等前端页面，并实现了期货价格走势图的展示。这个过程不仅提高了我的前端开发技能，还让我更好地理解了前后端之间的协作和数据传输。

通过参与这个项目，我不仅学到了Spring Boot和Vue.js的重要知识和技能，还深刻认识到了它们在现代Web应用开发中的关键作用。这次经验不仅丰富了我的技术背景，还为我今后的职业发展提供了宝贵的经验和见识。

* 1. D

在完成这个项目时，我负责了部分后端的开发，这对我这个从未有小组项目经验的人来说是一次前所未有的挑战。起初我们的小组分工并不明确，导致了我们初期开发的困难，对我们小组造成了许多的障碍。而且受限于对javaee知识的不明确以及困乏，导致我们的项目难以持续有效的进行。但是在我们小组的共同努力之下，我们组有了自己的前端与后端的开发师，成功并且出色的完成了各种的任务。而在spring 开发模式中，spring 容器使用了工厂模式为我们创建了所需要的对象，不需要我们自己创建了，直接调用spring 提供的对象就可以了，这是控制反转的思想面向切面编程：在面向对象编程思想中，我们将事物纵向抽成一个个的对象。而在面向切面编程中，我们将一个个的对象某些类似的方面横向抽成一个切面，对这个切面进行一些如权限控制、事物管理，记录日志等公用操作处理的过程就是面向切面编程的思想。并且Spring 使用 JavaBean 对象的 set 方法或者带参数的构造方法为我们在创建所需对象时将其属性自动设置所需要的值的过程，就是依赖注入的思想。这些思想都在项目设计中不断体现。同时老师也指出了我们项目的许多不足，包括后端项目结构中具体的功能实现并不清楚，以及存在逻辑错误，实用性较差的问题。这些都需要我们在以后得学习中加以更正。我认为除了具有专业的课程知识以外，非常有必要进行其他相关方面知识的学习而良好的团队分工可以大大加快开发进程。软件工程不仅是代码方面的能力，还需要管理能力。如何让一个团队发挥更大的作用，是每一个参与到软件工程中的人都应该思考的问题。本次实验中，我们明确了各自的分工与职责，尽力尽快完成自己的工作，再去帮助那些进度慢的组员，让我们的开发效率有所提高。此外，听懂与真正会运用还是有很大差别的，平时学到的知识不能照搬到实验中，要根据具体的情况调整。这次实验让我对SSH框架有了更深的了解，对以后开发类似的项目有了良好的经验。

参考文献

[1] 赵捷. 基于Web的期货交易模拟系统的设计与实现.大连理工大学,2015.

附录（电子版本材料）

附录A 源代码.zip