软件设计说明书

拟制：雷皓迪 日期： 2024-04-11

复核： 日期： 2024-05-14

批准： 日期： 2024-06-11

文档修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **修订日期** | **修订版本** | **修改描述** | **作者** |
| 2024.04.16 | 1.1 | 增加了对数据库技术和前端技术的分析，并对功能性需求做了补充 | 雷皓迪 |
| 2024.04.23 | 1.2 | 对需求分析做了进一步完善 | 雷皓迪 |
| 2024.05.21 | 1.3 | 撰写系统方案设计 | 雷皓迪 |
| 2024.06.05 | 1.4 | 撰写项目接口文档 | 雷皓迪 |

目录

[软件设计说明书 1](#_Toc120110330)

[文档修订记录 1](#_Toc120110331)

[目录 2](#_Toc120110332)

[1 简介 3](#_Toc120110334)

[1.1 目的和对象 3](#_Toc120110335)

[1.2 范围 3](#_Toc120110336)

[2 需求概述 5](#_Toc120110337)

[2.1 需求背景 5](#_Toc120110338)

[2.2 需求列表 5](#_Toc120110339)

[2.3 需求范围及限制 6](#_Toc120110340)

[3 需求分析 6](#_Toc120110341)

[3.1 竞品分析 6](#_Toc120110342)

[3.2 功能性需求 8](#_Toc120110343)

[3.3 设计约束 **错误!未定义书签。**](#_Toc120110344)

[3.4 依赖分析 8](#_Toc120110345)

[3.5 影响分析 11](#_Toc120110346)

[4 系统方案设计 12](#_Toc120110347)

[4.1 零层设计 12](#_Toc120110348)

[4.2 一层设计 13](#_Toc120110349)

[4.2.1 方案选型 13](#_Toc120110350)

[4.2.2 模块设计 14](#_Toc120110352)

[4.2.3 关键流程设计 14](#_Toc120110353)

[4.3 二层设计 15](#_Toc120110354)

[4.4 技术方案风险点 17](#_Toc120110355)

[5 数据结构设计 18](#_Toc120110356)

[6 接口设计 18](#_Toc120110357)

[6.1 提供接口 18](#_Toc120110358)

[6.2 依赖接口 19](#_Toc120110359)

[7 附录 20](#_Toc120110360)

# 简介

## 目的和对象

目的： 本文档旨在详细描述即将开发的量化交易软件的各项功能需求、性能需求、安全需求、用户界面需求以及与其他系统集成的需求，以便于指导软件开发团队明确目标、理清思路、制定有效的开发计划和测试方案。通过对软件进行全面、精确的需求分析和定义，确保最终交付的产品能够满足各类量化交易用户在策略开发、策略回测、实时交易、风险管理等方面的实际需求，助力用户实现资产的科学管理和有效增值。

适用对象：

1. 软件开发团队：包括项目经理、系统架构师、软件工程师、测试工程师等，他们将直接依据本文档进行系统设计、编码实现和测试验证工作。

2. 业务分析师：负责理解和细化量化交易业务流程，确保软件需求与实际业务紧密贴合。

3. 产品经理：根据文档指导产品的功能规划、路线图制定及迭代更新。

4. 用户代表：包括潜在的终端用户、交易员、量化策略师以及资产管理者，他们可以通过阅读此文档预先了解软件的功能特性，并对需求提出反馈和建议。

5. 监管机构：在必要时，文档也可能用于向相关金融监管机构展示软件是如何遵守法规要求和保障用户利益的。

6. 运维人员：理解软件运行所需的硬件配置、系统环境以及运维管理要求。

通过这份需求文档，所有涉众都能够对软件的目标、预期功能、性能指标、安全标准及用户交互界面有共同的理解，确保在整个开发生命周期内各方沟通协调一致，推动项目的顺利实施和高质量产品的诞生。

## 范围

软件范围：

本量化交易软件需求规格说明书涵盖了以下关键组成部分：

1. 策略开发模块：

提供策略设计工具，支持用户使用高级编程语言（如Python、R等）创建和修改量化交易策略。

内置策略模板库，允许用户导入、修改或复用已有的成熟交易策略模型。

支持策略的图形化设计，包括可视化回测结果和策略参数优化。

2. 数据管理模块：

实现与多个数据提供商的接口连接，获取实时和历史金融数据。

提供数据清洗、预处理和存储功能，便于用户进行数据分析和策略研究。

支持本地数据导入导出，以及定期备份和恢复。

3. 回测系统：

设计可配置的回测引擎，支持多种基准指数比较、滑动窗口测试和多周期回测。

输出详细的回测报告，包括收益曲线、胜率、最大回撤、夏普比率等统计指标。

4. 实盘交易模块：

构建与交易所、经纪商API的集成，支持自动化下单、撤单、持仓管理等功能。

实现实时风控功能，设定交易限额、止损止盈规则以及仓位调整策略。

5. 监控与报告系统：

提供实时交易监控面板，展示交易状况、头寸详情和绩效指标。

生成交易日志和交易报表，便于用户审计和总结交易表现。

6. 用户管理与权限控制：

多用户支持，设置不同级别的账户权限，确保用户数据的安全性和隐私保护。

提供账户认证、角色分配和操作记录审计功能。

设计关系：

上述各个模块之间相互关联但又相对独立，策略开发模块产生的策略文件将作为输入传给回测系统和实盘交易模块进行测试和执行。

数据管理模块与策略开发、回测、实盘交易模块均有密切联系，为它们提供必要的数据支撑。

回测系统和实盘交易模块的设计紧密相连，均需遵循统一的接口规范和交易逻辑，实现在回测环境下与实盘环境下的平滑过渡。

监控与报告系统贯穿整个软件生命周期，收集各个模块的运行状态和交易数据，为用户提供实时反馈和历史分析。

# 需求概述

## 需求背景

专业交易员和机构投资者对于交易平台的需求主要集中在高度可定制化的交易策略开发环境、高效执行交易的能力以及实时监控和风险管理工具。

量化交易员和算法交易员对于平台的需求主要集中在强大的数据分析和回测功能、低延迟的交易执行以及与交易所和经纪商的高效接口。

个人投资者和零售交易者可能对金融市场了解较少，但希望通过量化交易平台获得更好的投资回报，对于平台的需求主要集中在简单易用的交易界面、预置的交易策略模板以及实时的交易信号和市场分析工具。

金融科技公司和创业者对于平台的需求主要集中在开放的API和插件系统、灵活的数据接入和处理能力以及与其他金融工具和服务的集成性。

学术界研究人员对于平台的需求主要集中在开放的数据接口和历史数据存储、灵活的模型开发和回测环境以及与学术研究工具的集成性。

## 需求列表

*Replace me: 本章节对来自产品或需求方的原始需求描述*

|  |  |  |
| --- | --- | --- |
| 需求 | 需求说明 | 备注 |
| *处理市场数据* | *对市场数据进行实时的获取与处理* |  |
| *生成交易信号* | 根据预定的交易策略生成交易信号 | 自动执行交易操作 |
| *提供用户界面* | 供用户监控交易状态、调整交易策略和分析交易结果 |  |

## 需求范围及限制

需求范围：

1.实现基本的交易功能，包括交易订单的创建、执行和监控。

2．设计简单易用的用户界面，减少用户学习成本。

3．提供基本的数据分析功能，如简单的市场趋势分析和交易信号生成。

4.支持基于简单规则的交易策略，如均线交叉等。

限制：

技术能力限制：考虑到作为初学者的技术能力有限，需要选择简单易实现的功能和算法。

数据获取限制：可能无法获取到高质量的实时数据，可以考虑使用免费或者延迟的数据源。

交易执行限制：由于缺乏实际的交易账户，可能无法进行实际的交易执行，只能模拟交易。

安全性限制：由于缺乏相关经验，可能无法实现高度安全的交易平台，需要做好安全性风险管理。

# 需求分析

## 竞品分析

# 竞品分析

拟对盈首AI全自动量化交易平台、掘金量化、聚宽量化和米筐量化这四款量化交易平台进行简要分析：

1. 方案实现原理和方法：

盈首AI全自动量化交易平台：侧重于AI技术的应用，提供自定义策略模板和智能策略生成工具，实现自动化交易决策。

掘金量化：提供一套完整的策略开发、回测和实盘交易流程，允许用户使用其平台构建基于各种金融理论和技术分析的量化交易策略。

聚宽量化：主打低门槛的策略编写和研究，支持Python等编程语言，具有强大的可视化策略编辑器和详尽的回测报告，适合量化投资初学者及专业人士。

米筐量化：可能提供高自由度的策略开发环境，支持复杂策略设计，并且可能具备全面的历史数据支持和高性能的交易引擎。

2. 技术参数指标：

执行速度：各平台在交易执行速度上的差异可能体现在API接口响应时间、订单发送速度等方面。

策略开发环境：各自平台支持的编程语言、回测工具的细致程度、模拟交易的真实程度、策略部署的简易程度等。

数据源：各平台提供的数据覆盖范围、数据质量（延迟、完整性）、数据更新频率等。

安全性与稳定性：账户安全性措施、交易系统的稳定性和容错能力、异常情况下的应急处理机制等。

用户体验：界面设计、教程文档、社区支持、增值服务（如培训课程、行业资讯）等。

3. 自身产品与竞品优劣对比：

对标竞品都是行业内经过验证的优秀产品，我们对此主要做学习性分析。

学习与借鉴：

盈首AI：关注其如何整合AI技术实现自动化交易策略的生成与执行，研究其用户界面设计如何简化复杂的量化操作流程，以及AI算法在提升交易效率和风险管理上的具体应用。

掘金量化：探究其策略开发工具的易用性，分析其回测系统的准确性和可靠性，了解其在支持用户进行策略迭代优化上的有效手段。

聚宽量化：学习其如何降低量化投资的入门门槛，尤其是在教育和新手引导方面的成功实践，以及如何构建易于阅读和调试的可视化策略编辑环境。

米筐量化：着重观察其在处理复杂策略场景的技术实力，以及如何确保大规模数据处理能力和高速交易执行效率，此外还可以参考其在策略安全性与隐私保护上的做法。

自我提升：

基础功能完善：对照竞品的基础功能，本产品能够涵盖量化交易的基本要素，包括但不限于策略编写、回测、实盘交易等功能模块。

用户体验优化：吸收竞品在交互设计、用户教程、社区互动等方面的优点，力求让自己的产品更加友好易用，帮助初级用户快速成长。

技术创新尝试：虽然初入行业，但我们也应积极探索前沿技术在量化交易软件中的运用，比如AI、大数据分析等，寻求在特定领域内打造自身的亮点功能。

4. 对我们的启示：

可以从竞品的优点中学习，例如盈首AI的智能化应用，掘金量化的用户友好的策略开发流程，聚宽量化的教育生态和易用性，以及米筐量化在复杂策略支持方面的深度。

针对竞品的短板，我们可以强化自身在相应领域的功能。如平台大多集成度较高，用户自由度较小。我们可以适度开放底层接口，方便进阶用户使用。一些平台的可视化不够完善，不能做到所见即所得。我们可以针对这方面进行优化。

在创新方面，可以尝试融合各家之长，同时探索新的应用场景和技术，形成独特的产品特色和服务模式，以满足更多细分市场的客户需求。

## 功能性需求

以下所有需求在具体实现时都应注意用户界面的易用性，注重界面设计的5W1Y原则，力求让用户感到“越用越好用”。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能性需求编号 | 需求说明 | 优先级 | 备注 |
| 用户登录/注册 | 应当用数据库永久保存，并保证用户名唯一性（不允许重名） | 1 |  |
| 模拟交易 | 实现买入、卖出功能，并检测是否有违规操作（如买入的价格大于已有的货币） | 1 |  |
| 数据收集与展示 | 实现动态展示A股每日最新价，并可允许用户点击获取某一只股票的详细信息，如最近30天价格走势等 | 1 |  |
| 用户个人中心 | 便于用户对感兴趣的股票进行收藏、持续关注等，允许用户定制自身投资喜好 | 2 |  |
| 股票分析 | 帮助用户智能分析特定股票的潜力等，并给出购买建议 | 2 |  |
| 策略选择 | 实现一种或多种量化交易策略 | 2 |  |
| 风险管理 | 设置风险管理手段，如当持仓超过一定限度给予用户提醒，连续提醒达到一定次数系统自动帮助用户做出降低风险的决策 | 2 |  |
| 数据安全保障 | 保证用户数据和交易策略不被窃取，如采取一定的反爬虫手段 | 3 | 实际web软件可能对安全的要求更高，优先级更大。 |

## 设计约束

量化交易平台是一个综合而全面的软件系统，在设计和开发过程中可能的限制主要有以下几个方面：

1. 数据源限制：量化交易所依托的股市数据源可能需要订阅或购买，如果使用爬虫可能受到例如访问频率限制或者历史数据不可靠等问题。

2. 实时性要求：某些交易策略可能需要实时数据和执行能力，而某些策略则可以在延迟较大的情况下执行。如果实现实时性较高的策略可能对设备性能提出较高要求。

3. 法律法规要求：金融领域受到严格的监管，应确保软件设计符合所有相关法律法规，包括数据隐私保护、证券交易规则等。

4. 安全性要求：交易平台涉及到资金和交易数据，安全性是至关重要的。特别是我们的平台在网页端运行，需要具备足够的安全措施来保护用户资金和数据的安全，如设定一些反爬虫操作等。

5. 可扩展性限制：软件在发展过程中可能会面临不断增长的用户和交易量，因此可扩展性是一个重要的设计考虑因素。可以考虑初步设计时提前预留借口以支持水平和垂直扩展，避免因可拓展性不足对软件进一步发展造成限制。

## 依赖分析

为了构建这个量化交易平台，我们除了提到的python库和第三方API之外，还需要考虑以下几个方面的技术和工具：

**Python库：**

pandas：用于数据处理和分析，包括数据清洗、转换、合并等操作。

numpy：提供大量的数学函数库，用于进行高效的数值计算。

matplotlib：用于数据可视化，可以生成各种图表，帮助分析和展示数据。

time：用于处理时间相关的操作，如获取当前时间、时间戳转换等。

pytorch：深度学习框架，用于构建和训练量化模型。

**第三方API：**

akshare：提供丰富的财经数据接口，包括但不限于股票、期货、基金等市场数据，可以大大简化数据获取的流程。

**Web框架：**

Flask：轻量级的Web框架，用于构建交易平台的后端服务，包括API接口、用户认证、权限管理等。

**数据库：**

SQLite 或 MySQL：用于存储交易数据、用户信息、策略配置等。

**前端技术：**

HTML/CSS/JavaScript：构建用户交互界面，包括交易界面、策略配置界面、数据分析界面等。

React/Vue：前端框架，用于构建复杂的前端应用，提升用户体验。

**版本控制和协作：**

Git：用于版本控制，确保团队成员之间的代码协同开发。

**部署和运维：**

Docker：容器化技术，用于打包应用及其依赖，简化部署流程，确保环境一致性。

Kubernetes：容器编排工具，用于管理Docker容器集群，提供高可用性、自动扩展等功能。

**安全和性能：**

HTTPS：确保数据传输的安全性。

负载均衡：用于提高系统的并发处理能力，确保在高负载下系统的稳定运行。

监控和告警：实时监控系统运行状态，及时发现并处理潜在问题。

构建这样一个量化交易平台是一个复杂的工程，需要综合考虑技术选型、架构设计、性能优化、安全性等多个方面。随着项目的推进，可能还需要根据实际情况进行技术调整和优化，我们结合小组成员学习数据库和前端的经验，对常见的数据库和前端技术进行了进一步分析：

常用的数据库包括：MySQL,MongoDB,redis等，其中，MySQL是至今最流行的开源关系型数据库，简单易用，拥有大量的第三方插件，社区活跃、文档丰富，支持快速的复杂查询操作、完整的事务操作和较高的安全性。mongoDB是目前非常流行的非关系型数据库，模式自由，可根据需要随时修改文档格式；支持海量数据的查询和插入，学习成本低。redis是近年来兴起的内存数据库，数据在内存中保证了访问的高效；在保证访问速度的同时，也能够进行持久化存储。

我们选择数据库的原则：从实际量化交易平台的需求出发，同时结合对应数据库的学习成本考虑，由于主要处理财务金融信息，因此决定采用事务型数据库MySQL。

关于前端框架的选择，当前主要流行的框架是React和Vue，其中Vue有着较为轻量化，API 设计简单直观，使用基于 HTML 的模板语法，与传统的 HTML 和 JavaScript 结合较好等优点，同时，Vue的作者尤雨溪是国人，因此它的文档中文版质量较高，因此我们决定采用Vue作为我们项目的前端框架。

## 影响分析

量化交易平台的影响分析，尤其是对其公共框架、控件相关软件的影响，以及对使用者的影响，是一个多维度且复杂的议题。以下是针对这些方面的详细分析：

一、对公共框架的影响

量化交易平台往往基于公共框架进行开发，这些框架为平台提供了基础架构和功能。因此，量化交易平台的设计和实现会对公共框架产生直接影响。

性能影响：量化交易平台的运行需要处理大量的数据和复杂的算法，这会对公共框架的性能提出更高的要求。平台的设计需要充分考虑到这一点，避免对框架造成过大的性能负担。

扩展性影响：随着量化交易市场的不断发展，平台需要不断升级和扩展以满足新的需求。这要求公共框架具有良好的扩展性，能够支持平台的不断迭代和升级。

安全性影响：量化交易平台涉及到大量的资金和敏感信息，因此安全性至关重要。平台的设计需要充分利用公共框架的安全机制，确保数据的安全性和完整性。

二、对控件相关软件的影响

控件是量化交易平台中用于实现特定功能的组件，它们与平台的其他部分紧密相连，相互影响。

集成性影响：控件需要与量化交易平台的其他部分无缝集成，以确保平台的稳定性和可用性。因此，在控件的设计和开发过程中，需要充分考虑其与平台其他部分的兼容性。

用户体验影响：控件的易用性和用户体验直接关系到使用者对量化交易平台的满意度。因此，控件的设计需要符合用户的使用习惯，提供直观、简洁的操作界面。

功能性影响：控件的功能实现需要满足量化交易平台的整体需求。例如，交易控件需要支持多种交易策略和执行方式，以满足不同用户的需求。

三、对使用者的影响

量化交易平台的设计和使用对使用者具有重要影响，主要体现在以下几个方面：

操作效率：平台的设计应简化操作流程，提高操作效率，降低使用者的学习成本。通过合理的界面布局和功能划分，使用者可以快速上手并高效地进行量化交易。

决策支持：平台应提供丰富的数据分析和可视化工具，帮助使用者更好地理解市场趋势和交易机会。这些工具可以提高使用者的决策效率和准确性。

风险管理：量化交易平台应具备完善的风险管理机制，帮助使用者识别和控制交易风险。通过设定风险阈值、监控交易过程等方式，可以有效降低使用者的风险暴露。

四、对其他软件模块的影响

量化交易平台在设计和实现过程中，还需要考虑与其他软件模块的兼容性和影响。

数据接口兼容性：平台需要与其他数据源进行交互，以获取实时的市场数据和其他相关信息。因此，平台的数据接口需要具备良好的兼容性，能够与其他软件模块进行无缝对接。

功能协同性：量化交易平台的功能可能与其他软件模块存在重叠或互补关系。在设计过程中，需要充分考虑这些功能之间的协同性，避免产生冲突或重复劳动。

系统稳定性：多个软件模块的集成可能增加系统的复杂性和不稳定性。因此，在设计和实现过程中，需要采取有效的措施来确保系统的稳定性和可靠性。

综上所述，量化交易平台的影响分析是一个复杂而重要的任务。在设计过程中，需要充分考虑公共框架、控件相关软件以及使用者的影响，以确保平台的稳定性、易用性和高效性。同时，还需要关注与其他软件模块的兼容性和协同性，以实现整体系统的优化和升级。

# 系统方案设计

## 零层设计

本章节将详细展示整个量化交易平台的软件结构，并阐述软件与外部主要交互模块的关系，以及模块间的关系和交互形式。

首先，量化交易平台的核心软件结构将包括以下几个关键模块：数据采集模块、策略分析模块、交易执行模块、风险管理模块以及用户交互模块。这些模块共同协作，实现平台的核心功能。

数据采集模块负责从各种市场数据源获取实时行情数据、历史数据以及其他相关信息。这些数据将作为策略分析模块和交易执行模块的基础输入。

策略分析模块根据用户自定义的交易策略或平台提供的策略模板，对采集到的数据进行分析和计算，生成交易信号或交易建议。

交易执行模块则负责根据策略分析模块的输出，自动或半自动地执行交易操作，包括下单、撤单、修改订单等操作。该模块需要与交易所或经纪商的交易系统进行对接，确保交易的及时性和准确性。

风险管理模块负责对交易过程中的风险进行实时监控和管理，包括设置止损点、控制仓位、调整杠杆等操作，以确保交易的安全和稳定。

用户交互模块则提供了友好的用户界面和交互方式，使用户能够方便地查看行情、管理策略、监控交易等操作。该模块还需要提供必要的安全认证和权限控制机制，确保用户数据的安全性和隐私性。

在与外部交互方面，量化交易平台需要与交易所或经纪商的交易系统、市场数据源、支付系统等进行接口对接，实现数据的传输和交易的执行。同时，平台还需要考虑与其他金融服务提供商的合作，以提供更全面的金融服务。

在模块间的关系方面，各模块之间通过接口进行通信和数据交换，确保信息的准确性和一致性。模块间的交互形式可以采用消息队列、API调用等方式，以实现高效的通信和协同工作。

通过零层设计，我们明确了量化交易平台的软件结构、外部交互关系以及模块间的交互形式，为后续的开发和实施工作提供了清晰的指导。

## 一层设计

## 方案选型

在设计量化交易平台的过程中，我们可能会面临多种可行的方案选择。本章节将重点说明在面临多种方案时，如何进行方案选型。

首先，我们会进行市场调研和竞品分析，了解业界已有的成熟方案和最佳实践。通过对比分析不同方案的优缺点、适用范围以及成本效益等方面，初步筛选出符合我们需求的方案。

其次，我们会结合平台的实际需求和业务特点，对筛选出的方案进行进一步的评估。我们会考虑方案的技术实现难度、系统的稳定性和可扩展性、数据处理的效率和准确性等因素，以确保所选方案能够满足平台的核心需求。

同时，我们还会考虑方案的成本效益和风险因素。我们会评估不同方案的实施成本、维护成本以及潜在的风险点，并综合考虑平台的长期发展规划和预算限制，选择最符合经济效益和风险控制的方案。

最后，我们会组织专家团队进行方案评审和讨论，听取各方面的意见和建议。通过充分讨论和评估，最终确定最适合我们量化交易平台的方案。

在方案选型过程中，我们会注重科学性和客观性，避免主观臆断和盲目跟风。我们将以平台的实际需求和发展方向为导向，选择最适合我们的方案，为平台的成功实施和长期发展奠定坚实的基础。

## 模块设计

本章节将对量化交易平台中的重点模块进行详细设计描述。

策略分析模块设计：

策略分析模块是平台的核心之一，负责根据用户定义的策略进行数据分析并生成交易信号。该模块采用高性能的算法引擎，支持多种策略编写语言和框架，如Python、R等。用户可以通过平台提供的策略编辑器或API接口，自定义策略逻辑，并上传至平台进行测试和回测。策略分析模块会对用户上传的策略进行验证和优化，确保其在实际交易中的有效性。

交易执行模块设计：

交易执行模块负责根据策略分析模块输出的交易信号，自动或半自动地执行交易操作。该模块与交易所或经纪商的交易系统进行对接，采用高并发、低延迟的技术架构，确保交易的及时性和准确性。模块还具备订单管理功能，包括订单的创建、修改、撤销等操作，以及订单状态的实时监控和通知。

风险管理模块设计：

风险管理模块是保障交易安全的重要组成部分。该模块通过实时监控交易过程中的各项风险指标，如仓位、止损点、杠杆等，提供风险预警和自动风险控制功能。当交易风险超过预设阈值时，模块会自动触发相应的风险控制措施，如平仓、调整仓位等，以减小潜在损失。

此外，平台还采用了多种优化措施来提高模块的性能和稳定性，如使用缓存技术加速数据处理速度，采用分布式架构提高系统的可扩展性和容错能力等。

## 关键流程设计

本章节将详细描述量化交易平台的关键流程，包括正常流程和异常流程，以及不同模块、数据、进程之间的运行情况和时序。

正常交易流程：

用户通过用户交互层定义交易策略并上传至平台。

策略分析模块接收用户策略，进行验证和优化后生成交易信号。

交易执行模块根据交易信号自动或半自动地执行交易操作，与交易所或经纪商的交易系统进行对接。

交易结果实时反馈至用户交互层，供用户查看和监控。

风险管理模块实时监控交易过程中的风险，确保交易安全。

异常处理流程：

当交易执行过程中出现异常（如网络故障、交易所系统异常等），交易执行模块会记录异常信息并尝试重试。

若重试失败，交易执行模块会触发异常处理机制，如通知用户或自动执行备选交易策略。

同时，风险管理模块会根据异常类型触发相应的风险控制措施，以减小潜在损失。

在流程设计过程中，我们采用了流程图和数据流图的方式来描述不同模块、数据、进程之间的运行情况和时序，以确保流程的清晰和准确。同时，我们还考虑了异常情况的处理和备份方案，以确保平台的稳定性和可靠性。

## 二层设计

#### 4.3.1 安全性设计

1. **认证机制**：
   * 在登录界面进行前端的路由锁定，保证用户不登录时不能手动跳转到其他页面
2. **权限和访问控制**：
   * 对后端网站的访问权限进行CORS策略设置
3. **定期安全评估**：
   * 定期进行维护和测试，以识别和修复安全漏洞。
   * 更新安全策略，以应对新的威胁和漏洞。

#### 4.3.2 用户界面设计

1. 登录与注册界面

* + **简洁明了**：设计一个简洁的登录界面，提供注册链接。
  + **实时提醒：**实时在用户输入的时候提醒用户输入是否合法

2. 策略开发与回测

* + **回测界面**：展示回测结果，包括收益曲线、交易信号等。
  + **策略库**：展示用户保存的策略，支持版本控制和复用。

3. 交易面板

* + **订单管理**：展示所有活动订单和历史订单，包括订单详情和状态。
  + **市场监控**：实时展示市场动态，如股票价格、交易量、市场深度。
  + **持仓信息**：显示当前持仓情况，包括头寸规模、平均成本、未实现盈亏。

4. 用户个人中心

* + **账户信息**：展示用户的账户详情，包括联系方式、安全设置、API密钥等。
  + **交易历史**：提供交易历史的详细记录，支持搜索和过滤。
  + **偏好设置**：允许用户根据个人喜好设置界面主题、通知选项等。

5. 数据可视化

* + **图表工具**：集成图表工具，支持多种图表类型，如折线图、柱状图、K线图。

6.交互设计

* + **导航**：设计直观的导航菜单，方便用户快速找到所需功能。
  + **反馈**：对于用户操作提供即时反馈，如按钮点击效果、操作成功或失败的提示。

7. 帮助与支持

* + **用户手册**：提供详细的用户手册和FAQ，帮助用户理解平台功能。
  + **反馈收集**：提供反馈渠道，允许用户在遇到问题时快速获得支持。

8. 界面美观性

* + **专业设计**：采用现代、专业的设计风格，提升用户的第一印象。
  + **颜色方案**：使用符合金融行业特性的颜色方案，如蓝白色调，传达信任和专业感。

#### 4.3.3 系统测试计划

1. 单元测试
   * **黑盒测试**：采取针对用户界面的等价类划分法进行黑盒测试，确保覆盖全部的测试场景。我们对注册功能的等价类划分测试详见测试报告。
   * **断言验证**：对每个函数的输入、输出和中间状态进行断言验证，确保逻辑正确性。
   * **边界条件**：特别关注边界条件和异常输入的测试，以确保系统的鲁棒性。
2. 集成测试
   * **接口测试**：验证各模块间的接口是否按预期工作，包括数据格式、传输安全性和错误响应。
   * **数据流测试**：确保数据在系统内部流动正确，包括从数据源到策略分析再到交易执行的整个流程。
   * **错误处理测试**：模拟各种错误情况，验证系统的异常处理机制是否有效。
   * **契约测试**：对服务间的契约进行测试，确保模块间的依赖关系得到满足。
3. 端到端测试
   * **用户旅程**：模拟真实用户的操作流程，从登录到策略开发、交易执行和监控的整个旅程。
   * **业务场景**：设计端到端的业务场景，覆盖所有关键的用户故事和用例。
4. 用户接受测试(UAT)
   * **用户参与**：邀请真实用户参与测试，收集他们的反馈和建议。
   * **场景测试**：确保所有用户提出的业务场景都能得到正确执行。
5. 测试报告
   * 生成详细的测试报告，包括测试覆盖率、缺陷率、性能指标等。

## 技术方案风险点

1. 注意敏感信息泄露：对于财务数据和用户信息，应该避免硬编码。

2. 防止SQL注入：确保所有数据库交互都遵循参数化查询的方式，以避免 SQL 注入。

3. 性能优化：多个数据库操作由 HTTP 请求触发。如果这些操作慢，应用程序的性能将受影响。考虑对数据库进行索引和查询优化。

# 数据结构设计

1. 用户数据 (User 类)

- 属性:

- username: 用户名，用作登录标识。

- password: 用户密码。

- nickname: 用户昵称，用于在应用内显示。

2. 股票持有数据 (stocks\_held)

- 数据格式:

- 存储为 JSON 对象，其中键为股票代码，值为持股数量。如[‘000001’ : 100]

3. 交易历史 (history)

- 数据格式:

- 存储为 JSON 数组，每个条目包含交易类型（买入/卖出）、股票代码、股票名称、数量、价格和交易时间等信息。

# 接口设计

## 提供接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 参数 | 返回值 | 功能描述 | 备注 |
| /login | POST: account, password | {status, nickname, history, balance, message} | 处理用户登录 |  |
| /signup | POST: account, password, name | {status, username, message} | 注册新用户 |  |
| /home | - | {username, nickname} | 显示主页信息 |  |
| /home/change-password | POST: account, password, newPassword | {status, message} | 更改用户密码 |  |
| /home/change-nickname | POST: account, password, newNickname | {status, message} | 更改用户昵称 |  |
| /home/CNstock | POST: code | {status, data, stock\_name, all\_info} | 获取中国股票信息 |  |
| /home/HKstock | POST: code | {status, data, stock\_name, all\_info} | 获取香港股票信息 |  |
| /home/recharge | POST: sum, account | {status, balance} | 处理账户充值 |  |
| /home/buy | POST: code, num, account | {status, balance, message} | 处理股票购买 |  |
| /home/sell | POST: code, num, account | {status, balance, message} | 处理股票销售 |  |
| /home/stock-holding | POST: account | {status, stocks\_info, message} | 显示当前股票持有情况 |  |
| /home/history | POST: account | {status, history, message} | 显示交易历史 |  |
| /home/topFive | POST: account | {status, topFive} | 显示价值前五的股票 |  |
| /home/price | POST: code | {status, price, message} | 获取当前股票价格 |  |
| /home/total | POST: account | {status, total} | 计算总资产 |  |
| /home/foreign-exchange | - | {status, data} | 显示外汇汇率 |  |
| /home/today-and-yesterday | POST: account | {status, today, yesterday, change} | 比较今天和昨天的总资产 |  |
| /home/held-stock | POST: account | {status, stocks\_info} | 给出每只所持有股票的详细信息 |  |
| /home/strategies | POST: account, Strategy\_Code | {status, data, message} | 执行交易策略 |  |

## 6.2依赖接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 参数说明 | 返回值 | 功能说明 | 备注 |
| ak.stock\_zh\_a\_spot\_em | 无 | A股所有股票的完整信息 | 获取A股市场股票信息 |  |
| ak.stock\_hk\_spot\_em | 无 | 港股所有股票的完整信息 | 获取港股市场股票信息 |  |
| ak.stock\_zh\_a\_hist | symbol, period, start\_date, end\_date, adjust | 对应A股股票在相应时间间隔内的股价 | 获取对应A股股票在相应时间间隔内的股价 |  |
| ak.stock\_hk\_hist | symbol, period, start\_date, end\_date, adjust | 对应港股股票在相应时间间隔内的股价 | 获取对应港股股票在相应时间间隔内的股价 |  |

# 附录

附录1：智投用户手册（见附件PDF）

附录2：测试报告（包括单元测试和黑盒测试）