**组长：魏鼎坤 3200105652@zju.edu.cn**

**组员：沈韵沨 李毅桐 王粤龙 吴林昱 张洮莼**

**日期：2022/05/15**

**版本：Version 1.0**

——交易客户端子系统

【股票交易系统】

**总体设计说明书**

**目录**

[1 引言 1](#_Toc103588656)

[1.1 编写目的 1](#_Toc103588657)

[1.2 项目背景 2](#_Toc103588658)

[1.2.1 软件系统名称 2](#_Toc103588659)

[1.2.2 任务提出者 2](#_Toc103588660)

[1.2.3 软件开发者 2](#_Toc103588661)

[1.2.4 目标用户 2](#_Toc103588662)

[1.2.5 实现该软件的计算机网络 2](#_Toc103588663)

[1.2.6 项目相关背景介绍 2](#_Toc103588664)

[1.3 相关定义 4](#_Toc103588665)

[1.3.1 HTML 4](#_Toc103588666)

[1.3.2 CSS 4](#_Toc103588667)

[1.3.3 JavaScript 4](#_Toc103588668)

[1.3.4 UTF-8 4](#_Toc103588669)

[1.3.5 Vue 4](#_Toc103588670)

[1.4 系统概述 5](#_Toc103588671)

[1.4.1证券账户业务和资金账户业务 5](#_Toc103588672)

[1.4.2交易客户端 5](#_Toc103588673)

[1.4.3 股票中央交易系统 5](#_Toc103588674)

[1.4.4 网上信息发布 6](#_Toc103588675)

[1.4.5 交易系统管理 6](#_Toc103588676)

[1.5 项目概述 6](#_Toc103588677)

[2 总体设计 7](#_Toc103588678)

[2.1 需求规定 7](#_Toc103588679)

[2.1.1 系统功能 7](#_Toc103588680)

[2.1.2 系统性能 7](#_Toc103588681)

[2.1.3 输入输出要求 8](#_Toc103588682)

[2.1.4 数据管理能力要求 8](#_Toc103588683)

[2.2 运行环境规定 10](#_Toc103588684)

[2.3 基本设计概念和处理流程 11](#_Toc103588685)

[2.4 结构 12](#_Toc103588686)

[2.4.1 交易客户端子系统结构图 12](#_Toc103588687)

[2.4.2 功能IPO图 12](#_Toc103588688)

[3 系统结构 15](#_Toc103588689)

[3.1 系统功能结构 15](#_Toc103588690)

[3.2 技术简介 17](#_Toc103588691)

[3.3 部署图 18](#_Toc103588692)

[3.4 内部接口图 19](#_Toc103588693)

[3.5 顺序图 20](#_Toc103588694)

[3.5.1 登陆客户端 20](#_Toc103588695)

[3.5.2 查询证券账户持有的股票 20](#_Toc103588696)

[3.5.3 查询账户资金 21](#_Toc103588697)

[3.5.3 发布购买股票指令 21](#_Toc103588698)

[3.5.4 发布出售股票指令 22](#_Toc103588699)

[3.5.5 撤销指令 22](#_Toc103588700)

[3.5.6 按名称查询股票 23](#_Toc103588701)

[3.5.7 按涨落情况查询股票 23](#_Toc103588702)

[3.5.8 按涨落情况查询股票 24](#_Toc103588703)

[3.5.9 修改密码 24](#_Toc103588704)

[4 执行概念 25](#_Toc103588705)

[4.1 登录验证 25](#_Toc103588706)

[4.1.1模块概述 25](#_Toc103588707)

[4.1.2 IPO图 25](#_Toc103588708)

[4.1.3 功能 25](#_Toc103588709)

[4.1.4 输入项 25](#_Toc103588710)

[4.1.5 输出项 25](#_Toc103588711)

[4.1.6 设计方法 26](#_Toc103588712)

[4.1.7 流程图 26](#_Toc103588713)

[4.1.8 测试计划 27](#_Toc103588714)

[4.2 查询证券账户持有的股票 28](#_Toc103588715)

[4.2.1 模块概述 28](#_Toc103588716)

[4.2.2 IPO图 28](#_Toc103588717)

[4.2.3 功能 28](#_Toc103588718)

[4.2.4 输入项 28](#_Toc103588719)

[4.2.5 输出项 28](#_Toc103588720)

[4.2.6 设计方法（算法） 29](#_Toc103588721)

[4.2.7 流程图 29](#_Toc103588722)

[4.2.8 测试计划 30](#_Toc103588723)

[4.3 查询资金账户 31](#_Toc103588724)

[4.3.1 模块概述 31](#_Toc103588725)

[4.3.2 IPO图 31](#_Toc103588726)

[4.3.3 功能 31](#_Toc103588727)

[4.3.4 输入项 31](#_Toc103588728)

[4.3.5 输出项 31](#_Toc103588729)

[4.3.6 设计方法（算法） 32](#_Toc103588730)

[4.3.7 流程图 32](#_Toc103588731)

[4.3.8 测试计划 33](#_Toc103588732)

[4.4 发出购买股票指令 34](#_Toc103588733)

[4.4.1模块概述 34](#_Toc103588734)

[4.4.2 IPO图 34](#_Toc103588735)

[4.4.3功能 34](#_Toc103588736)

[4.4.4 输入项 34](#_Toc103588737)

[4.4.5 输出项 34](#_Toc103588738)

[4.4.6 设计方法 35](#_Toc103588739)

[4.4.7 流程图 35](#_Toc103588740)

[4.4.8 测试计划 36](#_Toc103588741)

[4.5 发出出售股票指令 37](#_Toc103588742)

[4.5.1模块概述 37](#_Toc103588743)

[4.5.2 IPO图 37](#_Toc103588744)

[4.5.3 功能 37](#_Toc103588745)

[4.5.4 输入项 37](#_Toc103588746)

[4.5.5 输出项 37](#_Toc103588747)

[4.5.6 设计方法 38](#_Toc103588748)

[4.5.7 流程图 39](#_Toc103588749)

[4.5.8 测试计划 40](#_Toc103588750)

[4.6 撤销指令 41](#_Toc103588751)

[4.6.1模块概述 41](#_Toc103588752)

[4.6.2 IPO图 41](#_Toc103588753)

[4.6.3 功能 41](#_Toc103588754)

[4.6.4 输入项 41](#_Toc103588755)

[4.6.5 输出项 41](#_Toc103588756)

[4.6.6 设计方法 42](#_Toc103588757)

[4.6.7 流程图 42](#_Toc103588758)

[4.6.8 测试计划 43](#_Toc103588759)

[4.7 按名称查询股票 44](#_Toc103588760)

[4.7.1 模块概述 44](#_Toc103588761)

[4.7.2 IPO图 44](#_Toc103588762)

[4.7.3 功能 44](#_Toc103588763)

[4.7.4 输入项 44](#_Toc103588764)

[4.7.5 输出项 44](#_Toc103588765)

[4.7.6 设计方法（算法） 45](#_Toc103588766)

[4.7.7 流程图 46](#_Toc103588767)

[4.7.8 测试计划 47](#_Toc103588768)

[4.8 按代码查询股票 48](#_Toc103588769)

[4.8.1 模块概述 48](#_Toc103588770)

[4.8.2 IPO图 48](#_Toc103588771)

[4.8.3 功能 48](#_Toc103588772)

[4.8.4 输入项 48](#_Toc103588773)

[4.8.5 输出项 48](#_Toc103588774)

[4.8.6 设计方法（算法） 49](#_Toc103588775)

[4.8.7 流程图 50](#_Toc103588776)

[4.8.8 测试计划 51](#_Toc103588777)

[4.9 修改密码 52](#_Toc103588778)

[4.9.1模块概述 52](#_Toc103588779)

[4.9.2 IPO图 52](#_Toc103588780)

[4.9.3 功能 52](#_Toc103588781)

[4.9.4 输入项 52](#_Toc103588782)

[4.9.5 输出项 52](#_Toc103588783)

[4.9.6 设计方法 53](#_Toc103588784)

[4.9.7 流程图 53](#_Toc103588785)

[4.9.8 测试计划 54](#_Toc103588786)

[5 接口设计 55](#_Toc103588787)

[5.1 登录界面 55](#_Toc103588788)

[5.2 个人股票信息页面 55](#_Toc103588789)

[5.3 个人资金信息页面 56](#_Toc103588790)

[5.4 股票购买页面 56](#_Toc103588791)

[5.5 股票出售页面 57](#_Toc103588792)

[5.6 交易记录页面 57](#_Toc103588793)

[5.7 外部接口 57](#_Toc103588794)

[6 运行设计 58](#_Toc103588795)

[6.1 运行模块组合 58](#_Toc103588796)

[6.2 运行控制 58](#_Toc103588797)

[6.2.1查询证券账户持有的股票 58](#_Toc103588798)

[6.2.2查询资金账户 58](#_Toc103588799)

[6.2.3发出购买股票指令 59](#_Toc103588800)

[6.2.4发出出售股票指令 59](#_Toc103588801)

[6.2.5撤销指令 60](#_Toc103588802)

[6.2.6显示交易结果 60](#_Toc103588803)

[6.2.7查询股票 60](#_Toc103588804)

[7 系统出错设计 61](#_Toc103588805)

[7.1 出错信息 61](#_Toc103588806)

[7.2 补救措施 62](#_Toc103588807)

[7.2.1 系统恢复 62](#_Toc103588808)

[7.2.2 定时备份 62](#_Toc103588809)

[7.2.3 人工操作 62](#_Toc103588810)

[7.3 系统维护设计 63](#_Toc103588811)

[8 需求回溯 64](#_Toc103588812)

[8.1 功能性需求回溯 64](#_Toc103588813)

[8.2 性能及安全需求 65](#_Toc103588814)

# 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，是以交易客户端的需求分析说明书为依据，从总体设计的角度，明确交易客户端子系统的总体架构、流程、数据结构。

目的在于：

* 为开发人员提供依据
* 为代码的修改、测试、维护提供条件
* 明确交易客户端各模块的接口
* 项目负责人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：

* 软件用户
* 项目经理
* 测试人员
* 系统管理和开发人员
* 系统维护人员

## 1.2 项目背景

### 1.2.1 软件系统名称

* 股票交易系统 —— 交易客户端子系统

### 1.2.2 任务提出者

* 软件工程基础课程任课教师-王新宇

### 1.2.3 软件开发者

* 浙江大学2021-2022学年夏学期软件工程基础课程C大组第二小组

### 1.2.4 目标用户

* 希望通过该股票交易系统进行投资的普通用户

### 1.2.5 实现该软件的计算机网络

* 由若干台PC机组成的局域网

### 1.2.6 项目相关背景介绍

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。

实验课则采取分组形式。每5个学生为一小组，5个小组为一个大组，每个小组完成该项目的一个部分，最终共同完成本项目。本次课程，教师选取股票交易系统作为综合性实验题目。其中我组主要负责交易客户端子系统。主要任务是开发用户界面，实现用户客户端申请，登陆，查看股票价格，查询资金账户里面的现金，查询证券账户里面的股票情况，发出购买和出售股票信息，显示交易结果等功能。

## 1.3 相关定义

### 1.3.1 HTML

超文本标记语言（英语：HyperText Markup Language，简称：HTML）是一种用于创建网页的标准标记语言。

### 1.3.2 CSS

层叠样式表（英语：Cascading Style Sheets，缩写：CSS；又称串样式列表、级联样式表、串接样式表、阶层式样式表）是一种用来为结构化文档（如HTML文档或XML应用）添加样式（字体、间距和颜色等）的电脑语言。

### 1.3.3 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。

### 1.3.4 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。

### 1.3.5 Vue

一套用于构建用户界面的渐进式框架。

## 1.4 系统概述

股票交易系统包括具有证券账户的开户，挂失，查看股票价格，查询资金账户里面的现金，购买和出售股票等功能。该系统由五个模块构成，分别为：证券账户业务和资金账户业务子系统、交易客户端子系统、股票中央交易子系统、网上信息发布子系统、交易系统管理子系统和成绩管理子系统。具体模块的功能要求如下：

### 1.4.1证券账户业务和资金账户业务

证券账户业务实现证券账户业务相关业务逻辑。开发工作人员管理界面，用于证券账户的开户、挂失与重新开户、销户。资金账户业务实现资金账户相关业务逻辑。开发工作人员界面，用于录入用户信息，审批，开户，修改、密码，存款，取款，挂失，销户，资金信息查询。并实现资金账户和账户的关联。

### 1.4.2交易客户端

实现股票交易客户端的相关业务逻辑。开发用户界面，包括用户客户端申请，首次登陆，登陆，查看股票价格，查询资金账户里面的现金，查询证券账户里面的股票情况，发出购买和出售股票信息，显示交易结果。

### 1.4.3 股票中央交易系统

实现股票场内交易相关业务逻辑。开发一个股票的中央交易系统，对用户发出的购买和出售股票的指令首先保存。然后匹配同一个股票的相关指令，将价格合适的两个指令进行撮合，并将交易结果放入不同的证券账户进行保存，然后修改已经匹配指令的信息。

### 1.4.4 网上信息发布

实现交易信息的实时发布逻辑。对每一个股票最新的交易结果，以及近期的交易结果进行统计，并且在网站上面发布，以供用户查询。用户可以输入股票名字或代码来对股票的相关信息进行查询。

### 1.4.5 交易系统管理

实现交易系统管理员对交易系统监控管理的功能。开发一个工作人员管理界面，通过识别不同用户的授权，可以查看不同股票的所有指令的数量，价格以及买卖数量的全部信息。

## 1.5 项目概述

交易客户端子系统是股票交易系统的模块之一。它主要是展示整个股票交易系统的各种页面，实现用户与股票交易系统的各种交互并返回对应结果。包括用户客户端申请，首次登陆，登陆，查看股票价格，查询资金账户里面的现金，查询证券账户里面的股票情况，发出购买和出售股票信息，显示交易结果。

# 2 总体设计

## 2.1 需求规定

### 2.1.1 系统功能

根据需求，本项目需要提供整个股票交易系统的各种交互页面，并提供各种接口。页面方面，本项目主要提供登陆页面，查询页面和交易页面，用于可视化登陆，查看股票价格，查询资金账户里面的现金，查询证券账户里面的股票情况，发出购买和出售股票信息，显示交易结果等。接口方面主要读取后端数据库的用户信息，股票信息，资金信息和交易信息等，实现登陆，查看股票价格，查询资金账户里面的现金，查询证券账户里面的股票情况，发出购买和出售股票信息，显示交易结果等功能。

### 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。  
   数据发布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。  
   数据发布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 条数据发布记录。
* 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（如用户名和密码）在交换时应加密。密码在存储前应加密。

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

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。

**计算机：**

* CPU: 不小于 2.0GHz
* 内存： 不小于 2.0GB

**通讯设备：**

* 网线：正常联通且数据传输能力好
* 网卡：100M

**客户端外围设备：**

* 键盘：可正常使用
* 鼠标：可正常使用
* 显示器：全彩显示，可正常使用
* 硬盘：大于100GB，转速大于7200rpm

**软件依赖：**

* 操作系统：Windows Vista/7/8/8.1/10, Mac OS，Linux
* 数据库平台：MySQL
* 前端开发框架：Vue3.0
* 后端语言：Python
* 后端框架： node.js
* Web 服务器：Apache
* MySQL 管理软件： MySQL WorkBench、PHPMyAdmin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）
* 建模工具：Microsoft Word
* 浏览器：Chrome、Edge、FireFox

## 2.3 基本设计概念和处理流程

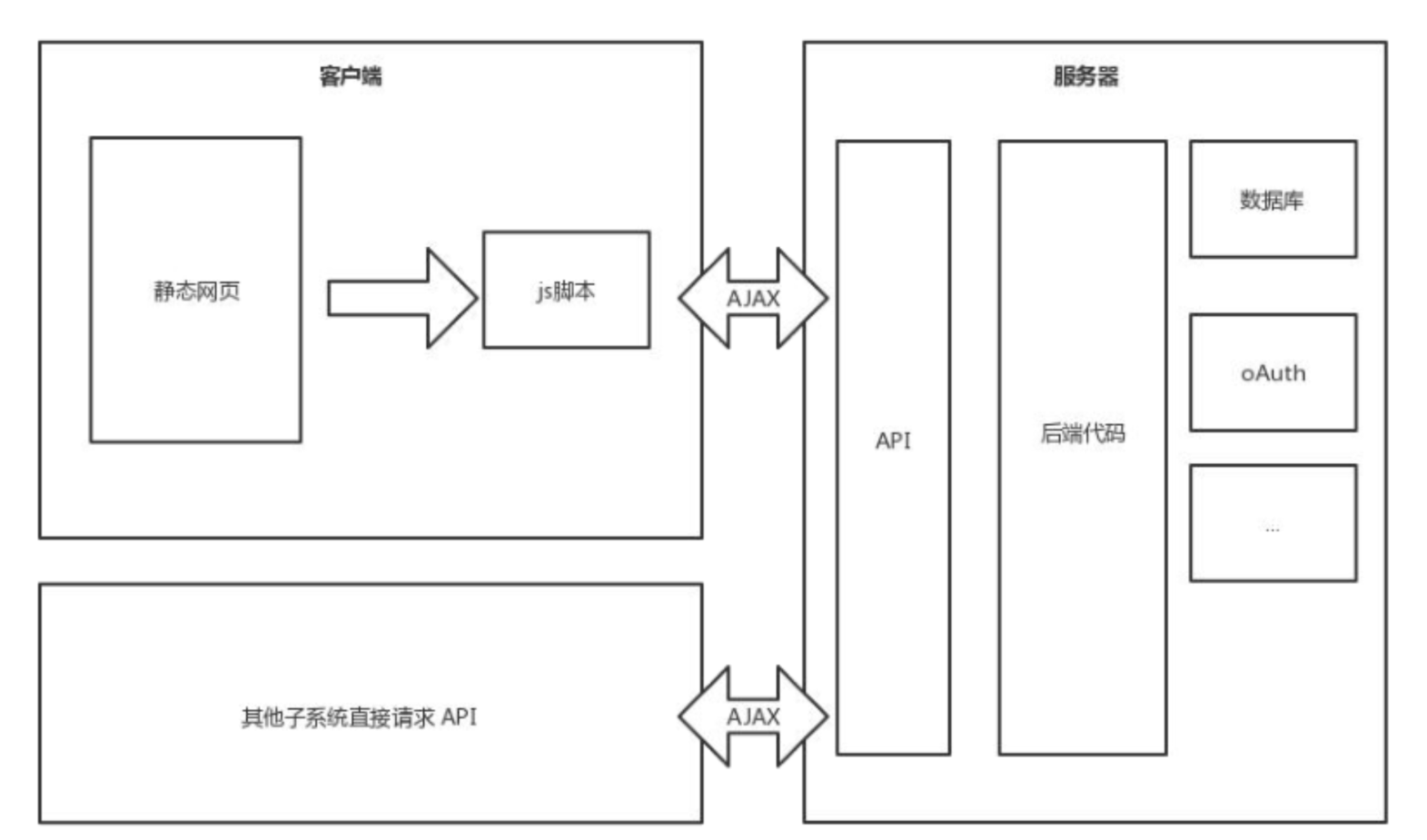
本子系统是一个基本上纯前端的模块，主要负责设计各种页面，并将用户进行的各种操作传入后端，并接收从后端传来的数据，可视化登陆，查看股票价格，查询资金账户里面的现金，查询证券账户里面的股票情况，发出购买和出售股票信息，显示交易结果等。

**前端：**

基于HTML，CSS，JavaScript网络前端，并且采用Vue框架来实现交易客户端各种页面的构建。

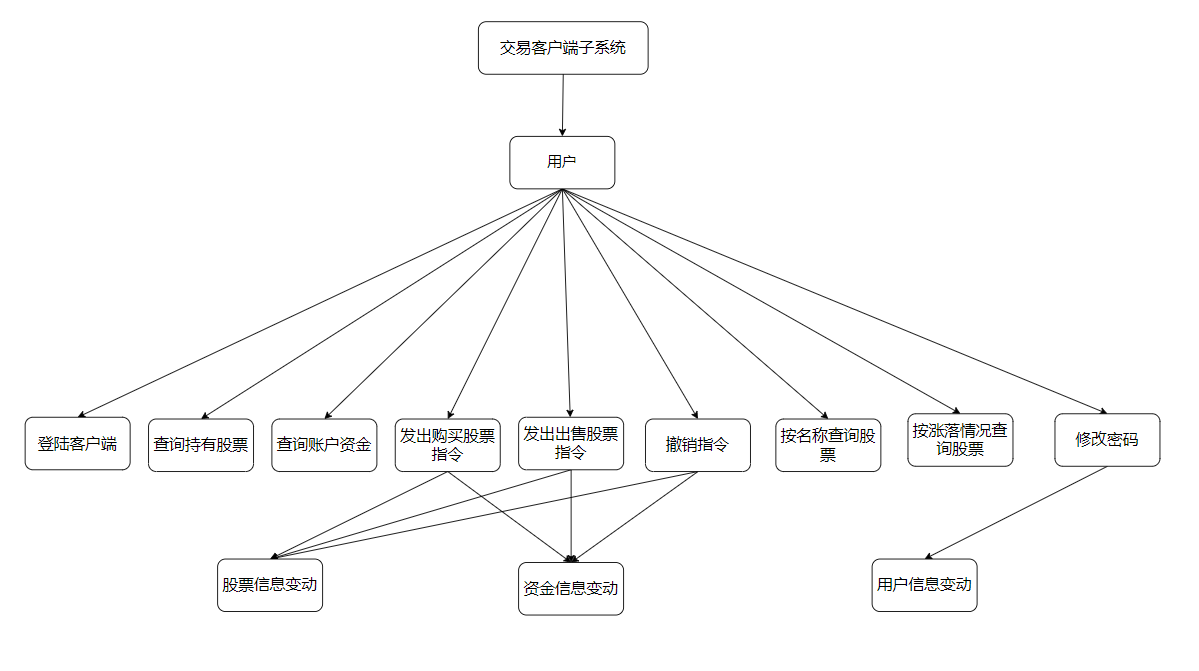
**客户端：**

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， 如需使用，版本须在 11 及以上。

**处理流程图如下：**

## 2.4 结构

### 2.4.1 交易客户端子系统结构图



### 2.4.2 功能IPO图

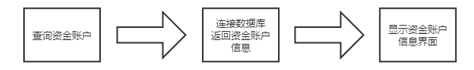
#### 2.4.2.1 登陆客户端



#### 2.4.2.2 查询账户持有的股票



#### 2.4.2.3 查询资金账户



#### 2.4.2.4 发出购买股票指令



#### 2.4.2.5 撤销指令



#### 2.4.2.6 按名称查询股票



#### 2.4.2.7 按涨落情况查询股票



#### 2.4.2.8修改密码



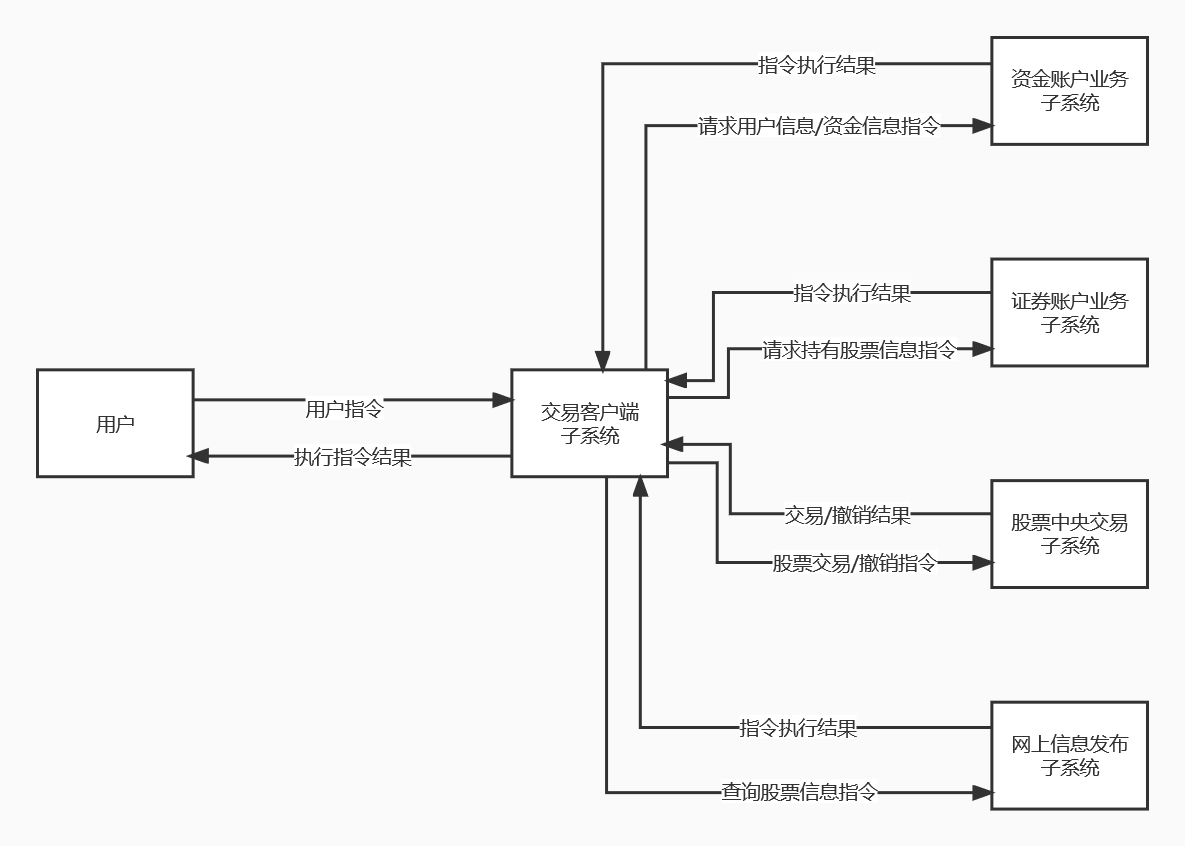
# 3 系统结构

## 3.1 系统功能结构

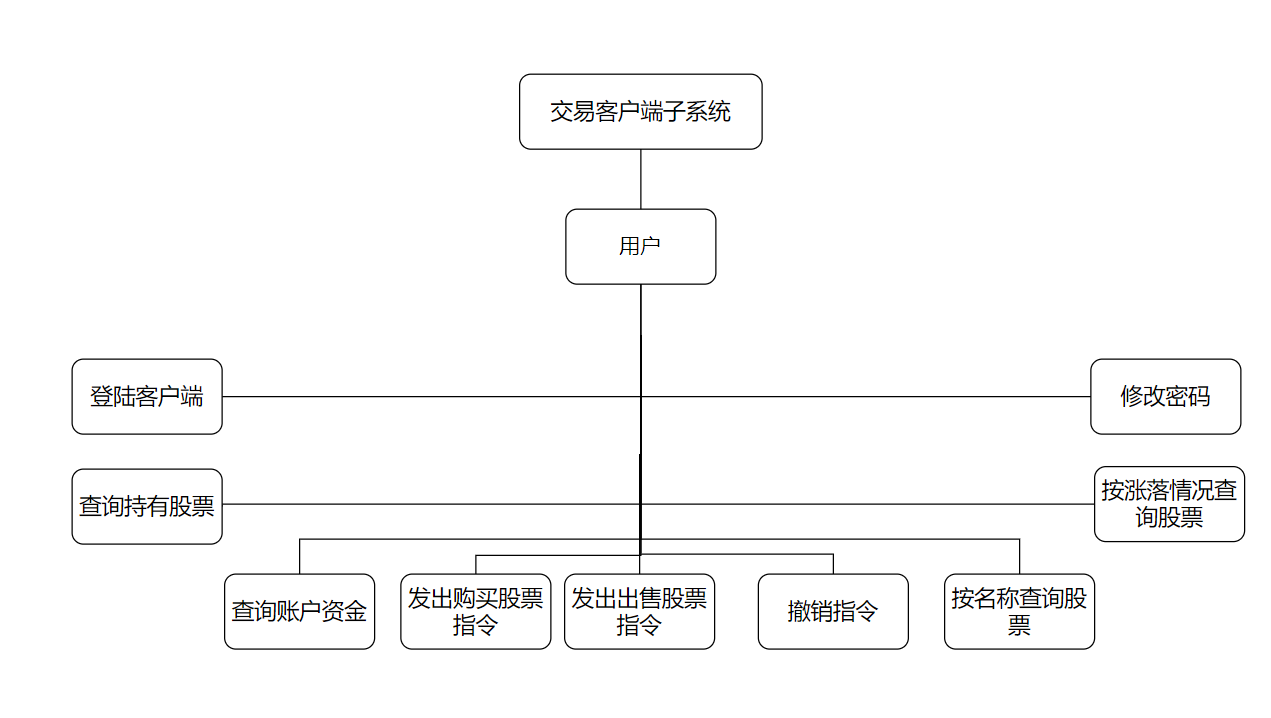
通过需求分析，已经完成对系统的用例分析，具体如下所示：



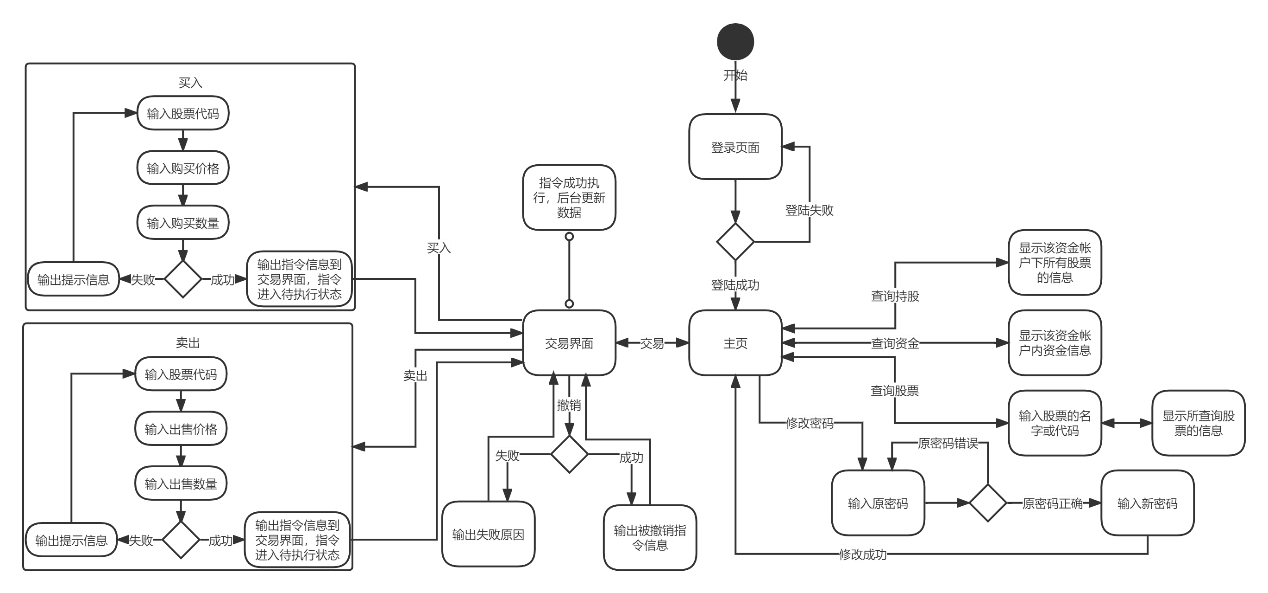
需求分析中，已经完成了对数据流相应的分析，其顶层数据流分析具体如下图所示：



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



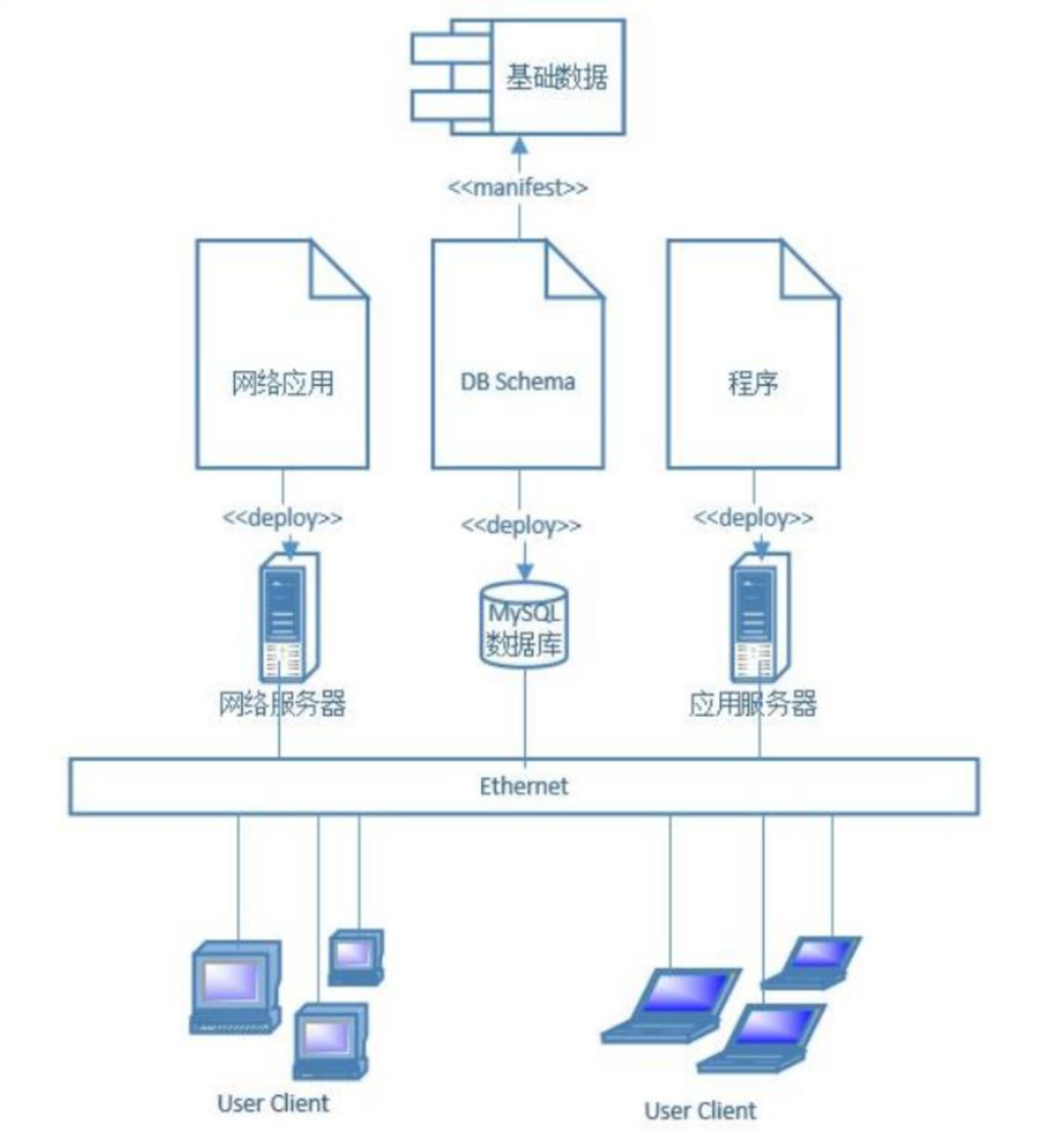
需求分析中完成了系统的状态图设计，这里引用状态图作为系统运行的整体流程参考。具体如下图：



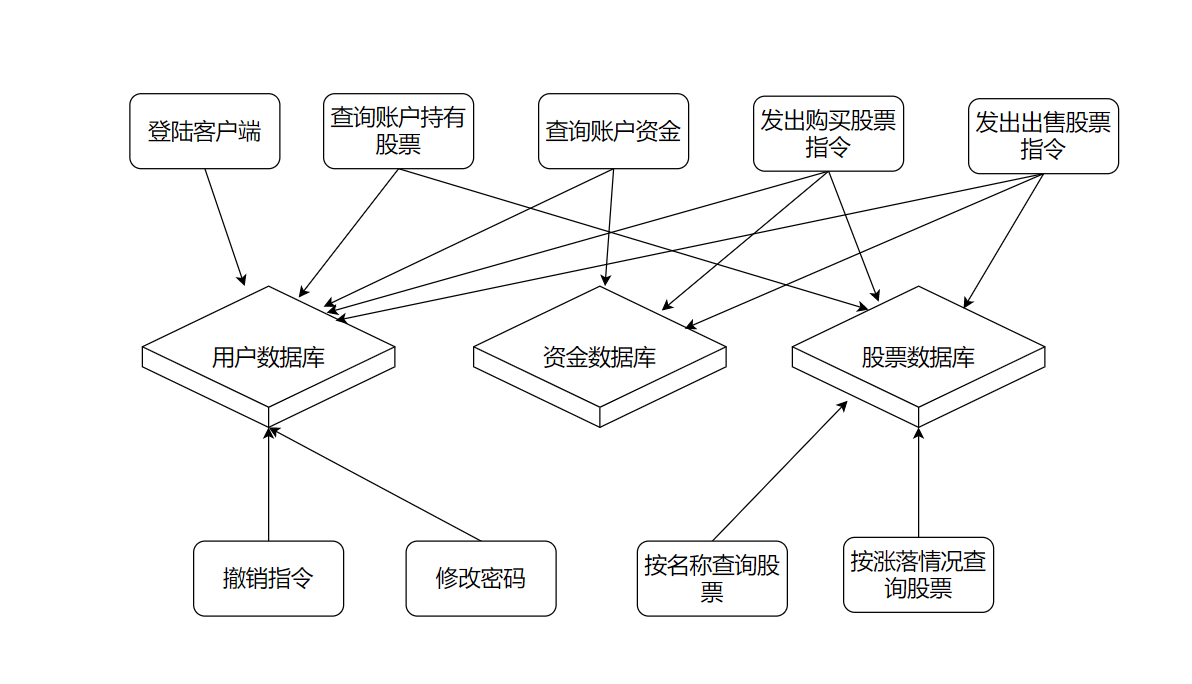
## 3.2 技术简介

在系统设计中，我们将采用基于HTML，CSS，JavaScript网络前端，并且采用Vue框架来实现交易客户端各种页面的构建。Vue是一套用于构建用户界面的渐进式框架。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，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

## 3.3 部署图

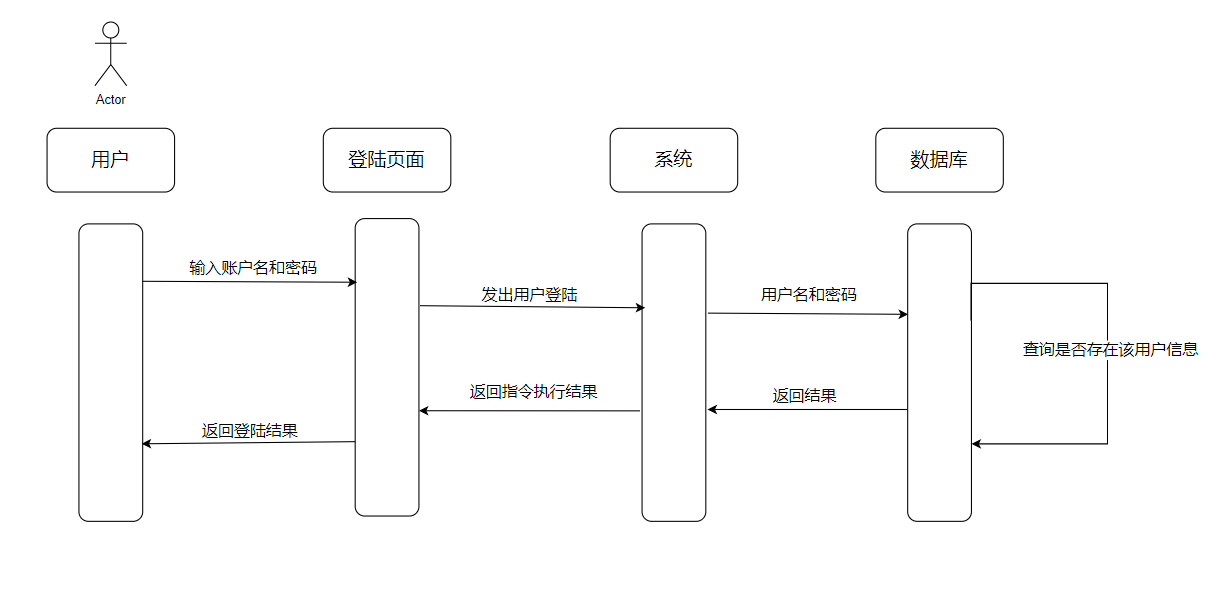


## 3.4 内部接口图

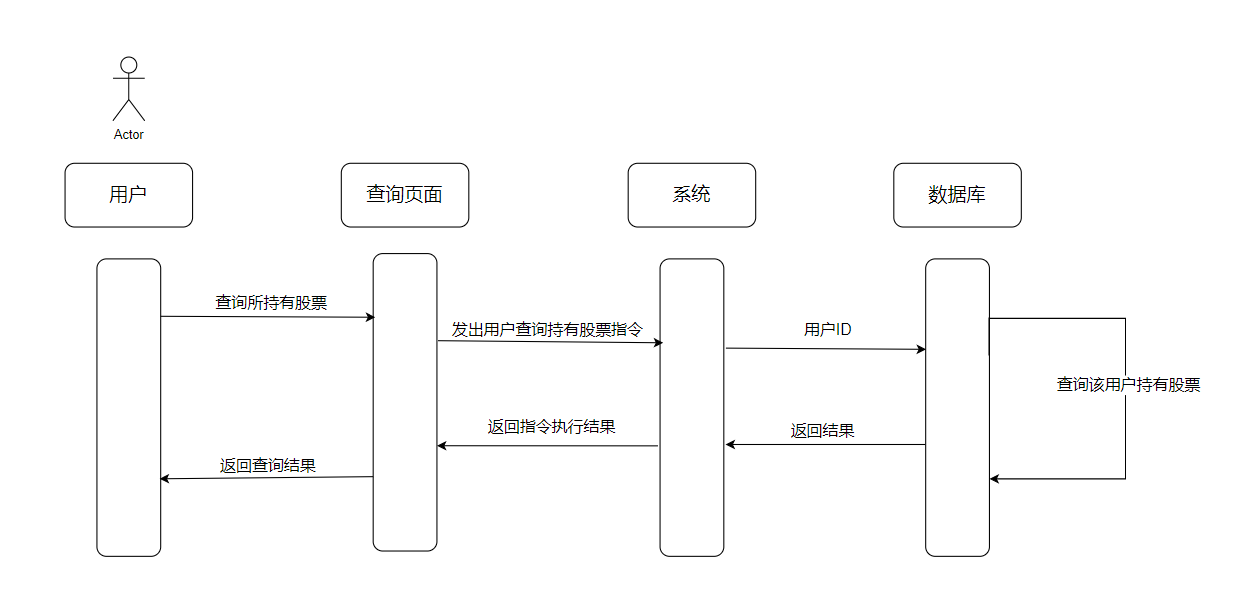


## 3.5 顺序图

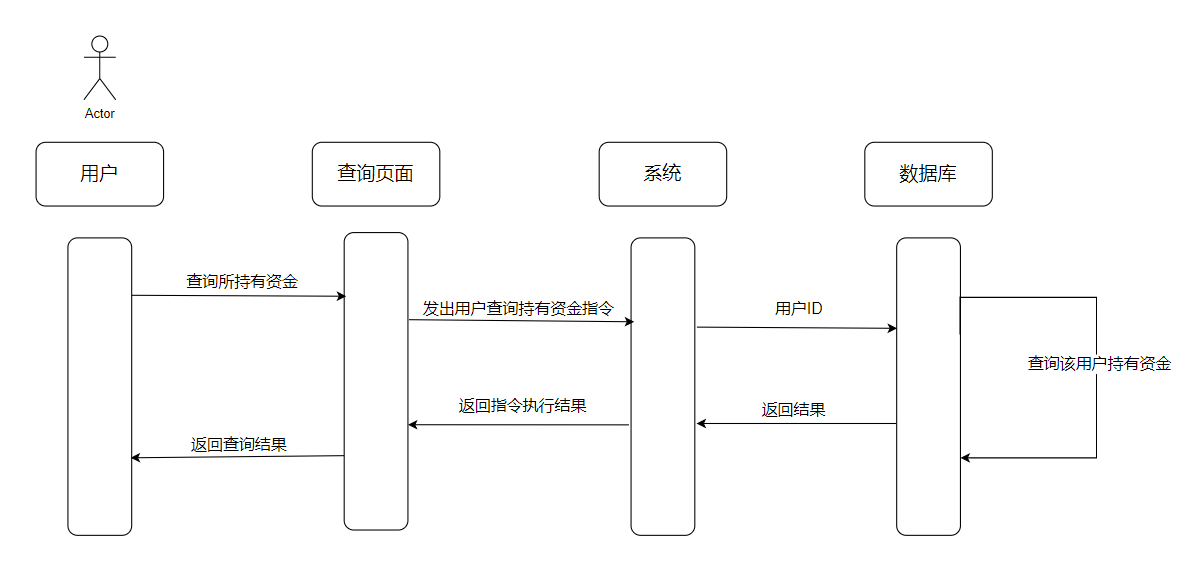
### 3.5.1 登陆客户端



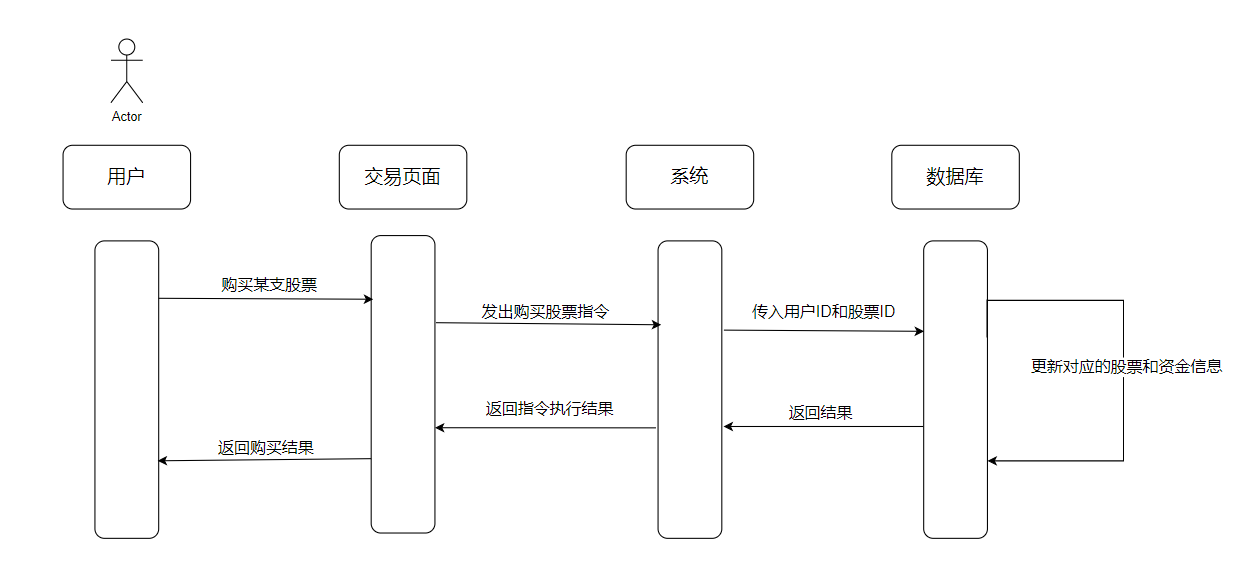
### 3.5.2 查询证券账户持有的股票



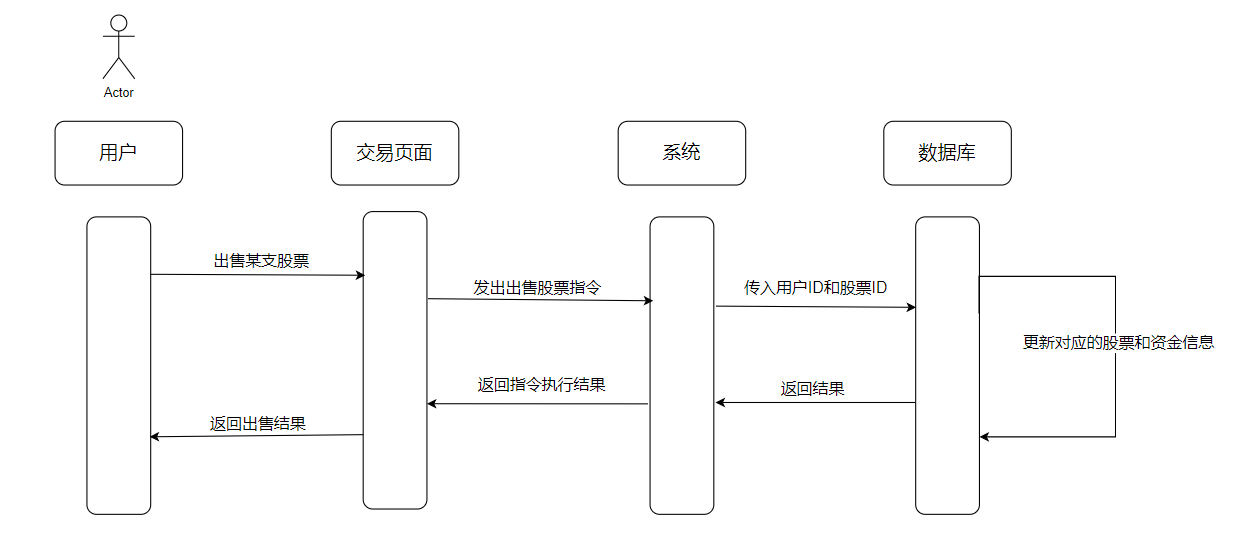
### 3.5.3 查询账户资金



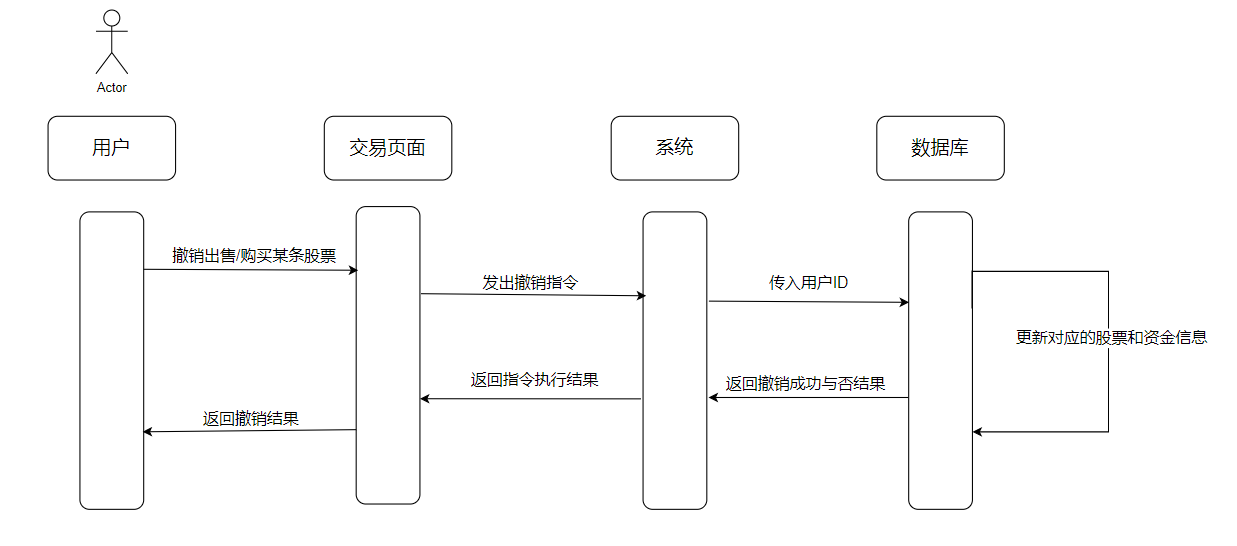
### 3.5.3 发布购买股票指令



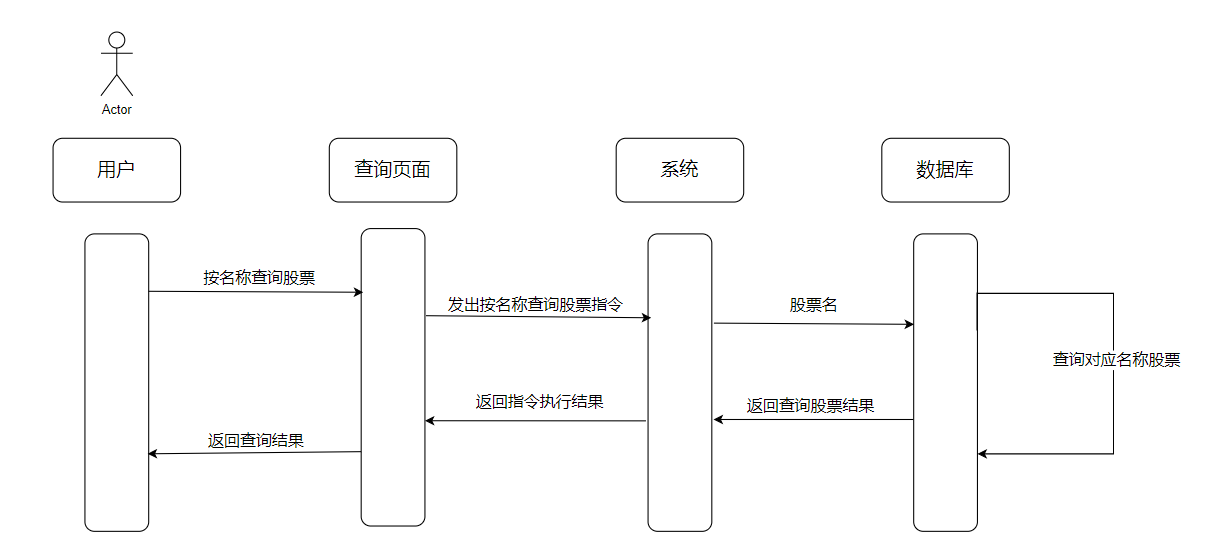
### 3.5.4 发布出售股票指令



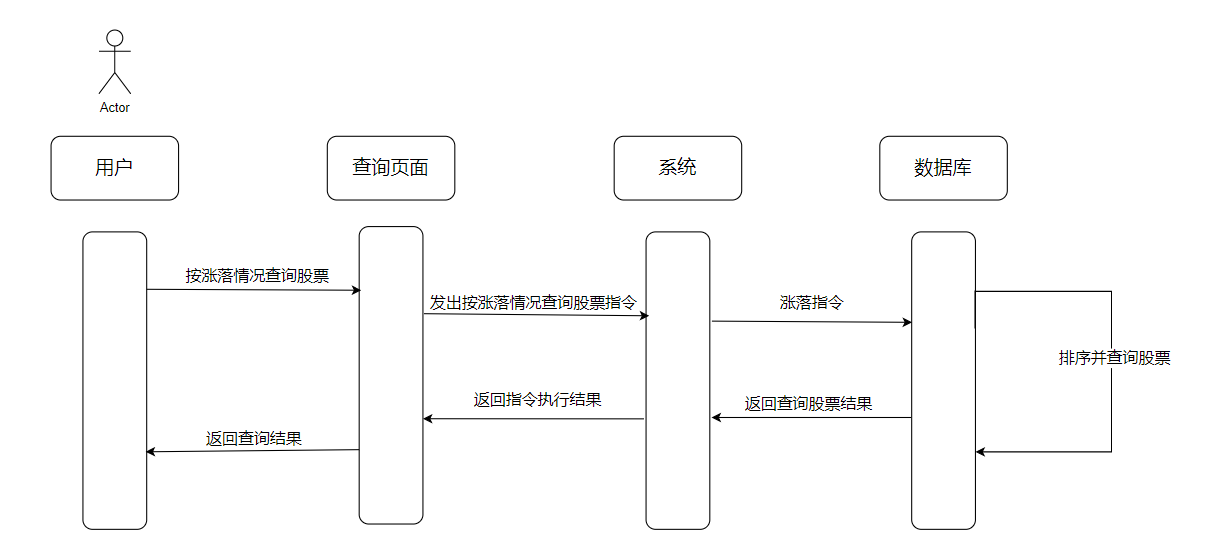
### 3.5.5 撤销指令



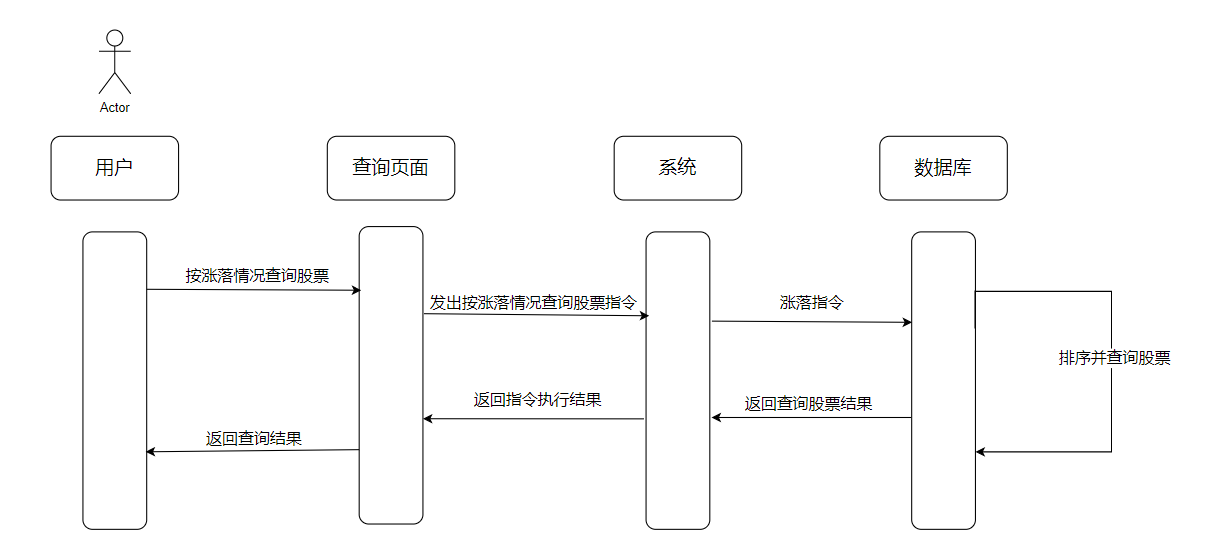
### 3.5.6 按名称查询股票



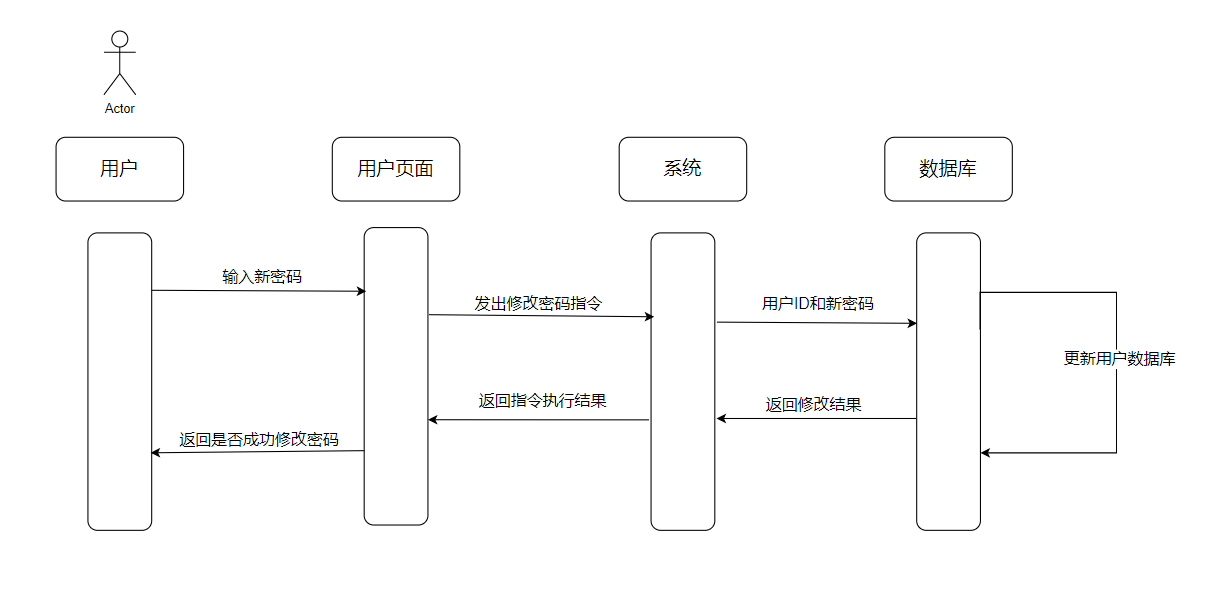
### 3.5.7 按涨落情况查询股票



### 3.5.8 按涨落情况查询股票



### 3.5.9 修改密码



# 4 执行概念

## 4.1 登录验证

### 4.1.1模块概述

本模块为用户登录系统时展示的页面，在此界面进行身份认证成功后才能获得相应的操作权限。存在的目的是确认用户的身份，从而保证系统安全性。

### 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** | **输入方式** |
| 登陆结果 | LoginResult | Bool | 由脚本输出 |

### 4.1.6 设计方法

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

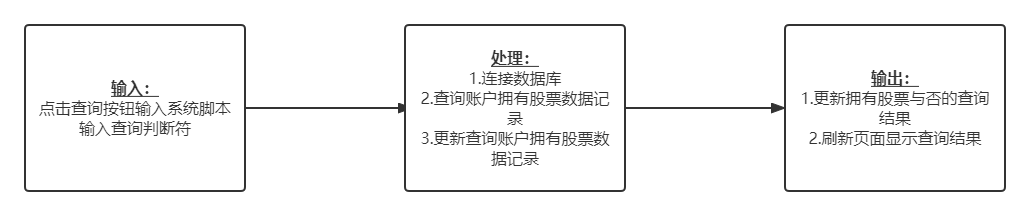
|  |  |
| --- | --- |
| **输入数据** | **预期结果** |
| 输入不合适的信息 | 返回错误信息，输入信息不合法 |
| 输入不匹配的账号和密码 | 返回错误信息，密码错误 |
| 输入匹配的账号和密码 | 返回成功信息，登录成功 |

## 4.2 查询证券账户持有的股票

### 4.2.1 模块概述

​ 在这个模块中，用户可以在证券账户子系统中通过账号密码登录进入本账户页面，从而使用查询用户账户持有的股票的信息的功能。在查询时，用户通过界面的操作提示来获取持有股票代码、持有股票名称、持有股票单价、持有股票数量、持有股票成本、持有股票损盈等信息，若未持有任何股票，系统将显示该账户未持有任何股票。

### 4.2.2 IPO图



### 4.2.3 功能

对用户登录的证券账户所持有的股票信息进行查询并显示。

### 4.2.4 输入项

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **标识** | **类型和格式** | **输入方式** |
| 证券查询标识 | query\_stock\_flag | Bool | 外部选择->脚本转换 |

### 4.2.5 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **标识** | **类型和格式** | **输入方式** |
| 证券账户查询结果 | query\_stock\_result | Bool | 由脚本输出 |

### 4.2.6 设计方法（算法）

if 账户已登录  
连接数据库  
用户点击查询键  
检测账户是否拥有股票  
if 账户拥有股票  
 系统根据数据库内信息返回账户所拥有股票代码、股票名称、股票单价、股票数量、股票成本、股票损盈等信息，刷新页面显示在用户界面上  
else  
显示该账户未拥有任何股票

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

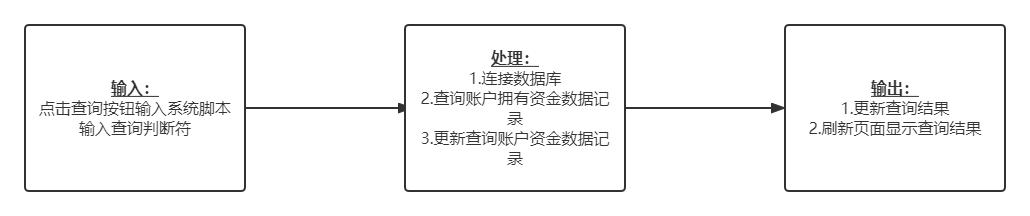
|  |  |
| --- | --- |
| **输入数据** | **预期结果** |
| 未登录，点击查询按钮 | 提示未登陆错误，返回登陆界面 |
| 已登录，账户内未拥有股票， 点击查询按钮 | 显示账户未拥有股票 |
| 已登录，账户内拥有股票， 点击查询按钮 | 显示账户拥有所有股票相关信息 |

## 4.3 查询资金账户

### 4.3.1 模块概述

在这个模块中，用户可以在资金账户子系统中通过账号密码登录进入本账户页面，从而使用查询用户账户资金信息的功能。在查询时，用户通过界面的操作提示来获取已储蓄资金数量、活动资金数量、冻结资金数量、绑定信用卡号、绑定储蓄卡号、活期利息等信息。若用户尚未绑定信用卡/储蓄卡，则显示该账户尚未完成绑定。

### 4.3.2 IPO图



### 4.3.3 功能

对用户登录的资金账户的资金相关信息进行查询并显示。

### 4.3.4 输入项

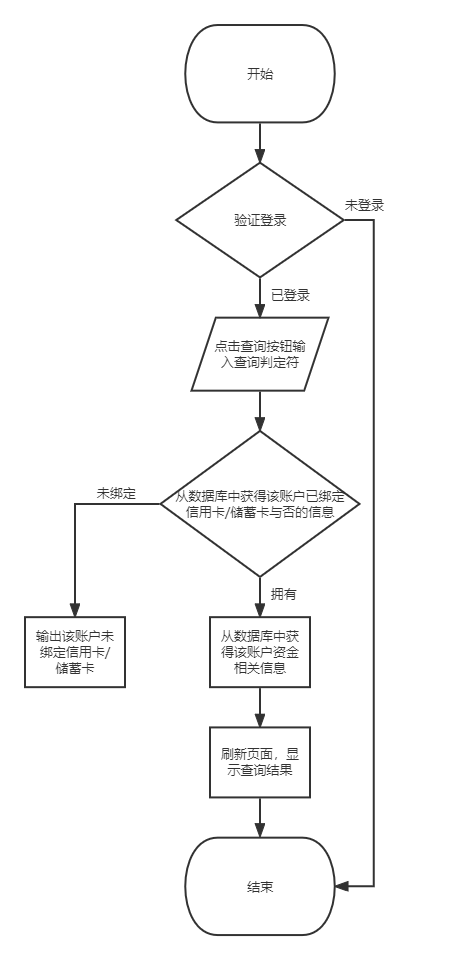
|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **标识** | **类型和格式** | **输入方式** |
| 资金查询标识 | query\_capital\_flag | Bool | 外部选择->脚本转换 |

### 4.3.5 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **标识** | **类型和格式** | **输入方式** |
| 资金账户查询结果 | query\_capital\_result | Bool | 由脚本输出 |

### 4.3.6 设计方法（算法）

if 账户已登录  
连接数据库  
用户点击查询键  
检测账户是否绑定信用卡/储蓄卡  
if 账户绑定信用卡/储蓄卡  
 系统根据数据库内信息返回账户已储蓄资金数量、活动资金数量、冻结资金数量、绑定信用卡号、绑定储蓄卡号、活期利息等信息，刷新页面显示在用户界面上。  
else  
 显示该账户未绑定信用卡/储蓄卡

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，点击查询按钮 | 提示未登陆错误，返回登陆界面 |
| 已登录，账户内未绑定信用卡 /储蓄卡，点击查询按钮 | 显示账户未绑定信用卡/储蓄卡 |
| 已登录，账户内未绑定信用卡 /储蓄卡，点击查询按钮 | 显示账户资金相关信息 |

## 4.4 发出购买股票指令

### 4.4.1模块概述

在这个模块中，用户通过界面上的购买股票功能可以发出购买某一股票的指令。投资需要输入 股票代码、购买价格和购买数量。购买股票总价不能超过资金帐户中可用资金总量。 指令的信息将会显示在投资者的界面上 。

### 4.4.2 IPO图

### 4.4.3功能

按某一价格购买一定数量的特定股票

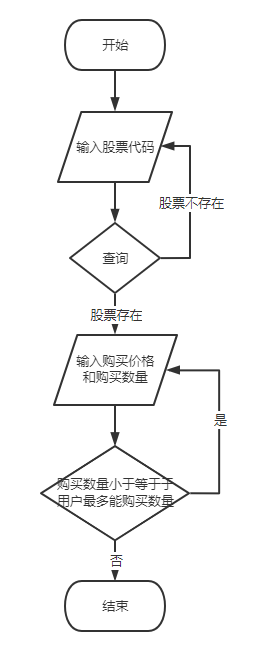
### 4.4.4 输入项

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **标识** | **类型和格式** | **输入方式** |
| 股票代码 | StockID | Varchar(20) | 外部输入 |
| 购买价格 | PurchasePrice | decimal(10.2) | 外部输入 |
| 购买数量 | PurchaseQuantity | int | 外部输入 |

### 4.4.5 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **标识** | **类型和格式** | **输入方式** |
| 购买状态标识 | PurchaseStatus | Bool | 由脚本输出 |

### 4.4.6 设计方法

连接数据库  
用户输入股票代码  
查询股票代码  
if 股票代码存在  
 输出参考价格  
​ 用户输入购买价格  
 用户输入购买数量  
    if 购买数量小于等于于用户最多能购买数量  
       输出成功信息（购买正在执行）  
       将指令的信息将会显示在投资者的界面上  
   else  
       输出错误信息（用户可用资金不足）     
else  
      输出错误信息（该股票不存在）

### 4.4.7 流程图

### 4.4.8 测试计划

|  |  |
| --- | --- |
| **输入数据** | **预期结果** |
| 不存在的股票代码 | 返回错误信息，该股票不存在 |
| 正确的股票代码和价格， 购买数量大于该账户最大购买数量 | 返回错误信息，用户可用资金不足 |
| 正确的股票代码、价格和数量 | 返回成功信息，购买正在执行 |

## 4.5 发出出售股票指令

### 4.5.1模块概述

在这个模块中，用户通过界面上的出售功能，可以发出出售投资者持有股票的指令。指令包括股票代码，出售价格和出售数量。购买价格不得低于下限或超过上限，出售股票总数不能超过投资者持有的总数。指令的信息将会显示在投资者的界面上。

### 4.5.2 IPO图

### 4.5.3 功能

按某一价格出售一定数量的特定股票

### 4.5.4 输入项

| **名称** | **标识** | **类型和格式** | **输入方式** |
| --- | --- | --- | --- |
| 股票代码 | StockID | Varchar(20) | 外部输入 |
| 出售价格 | SellingPrice | decimal(10.2) | 外部输入 |
| 出售数量 | SellingQuantity | int | 外部输入 |

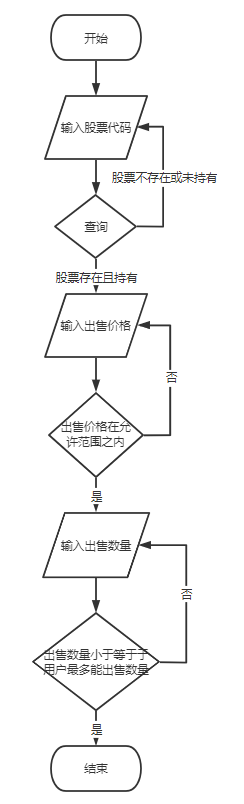
### 4.5.5 输出项

| **名称** | **标识** | **类型和格式** | **输出方式** |
| --- | --- | --- | --- |
| 出售状态标识 | SellingStatus | Bool | 脚本输出 |

### 4.5.6 设计方法

连接数据库  
用户输入股票代码  
查询股票代码  
if 股票代码存在  
   if 用户持有该股票  
 输出参考价格  
​ 用户输入出售价格  
       if 出售价格在允许范围之内  
          用户输入出售数量  
​          if 出售数量小于等于于用户最多能出售数量  
​              输出成功信息（出售正在执行）  
​              将指令的信息将会显示在投资者的界面上  
​          else  
​              输出错误信息（用户持股数不足）              
       else  
          输出错误信息（出售价格高于上限或低于下限）  
  else  
    输出错误信息（该股票未持有）  
else  
   输出错误信息（该股票不存在）

### 4.5.7 流程图



### 4.5.8 测试计划

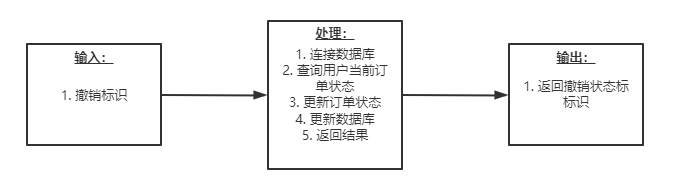
| **输入数据** | **预期结果** |
| --- | --- |
| 不存在的股票代码 | 返回错误信息，该股票不存在 |
| 未持有股票的股票代码 | 返回错误信息，该股票未持有 |
| 正确的股票代码，出售价格高于上限 | 返回错误信息，出售价格高于上限 |
| 正确的股票代码，出售价格低于下限 | 返回错误信息，出售价格低于下限 |
| 正确的股票代码和价格， 出售数量大于该账户拥有该股票数量 | 返回错误信息，持股数不足 |
| 正确的股票代码、价格和数量 | 返回成功信息，出售正在执行 |

## 4.6 撤销指令

### 4.6.1模块概述

在这个模块中，用户通过界面上的撤销功能，可以撤销发出的指令。指令如果尚未执行，则可以撤销，如果 指令已经执行，拒绝投资者的请求。

### 4.6.2 IPO图



### 4.6.3 功能

撤销已发出但未执行完成的购买或出售指令

### 4.6.4 输入项

| **名称** | **标识** | **类型和格式** | **输入方式** |
| --- | --- | --- | --- |
| 撤销标识 | UndoFlag | bool | 外部选择->脚本转换 |

### 4.6.5 输出项

| **名称** | **标识** | **类型和格式** | **输出方式** |
| --- | --- | --- | --- |
| 撤销状态标识 | UndoStatusFlag | bool | 脚本输出 |

### 4.6.6 设计方法

用户发出撤销指令  
连接数据库  
查询用户当前订单状态  
if 用户当前有订单  
   if 订单正在执行且未完成  
       输出成功信息（撤销成功）  
 else  
       输出失败信息（购买或出售指令已执行完成）  
else  
   输出失败信息（当前没有购买或出售指令）

### 4.6.7 流程图

### 4.6.8 测试计划

| **输入数据** | **预期结果** |
| --- | --- |
| 当前没有发出购买或出售指令，撤销 | 返回失败信息， 当前没有购买或出售指令 |
| 购买或出售指令已执行完成，撤销 | 返回失败信息， 购买或出售指令已执行完成 |
| 发出了购买或出售指令，且指令正在执行，撤销 | 返回成功信息，撤销成功 |

## 4.7 按名称查询股票

### 4.7.1 模块概述

在这个模块中，用户可以在信息查询子系统中通过账号密码登录进入查询页面，从而使用按名称查询的功能。在查询时，用户通过界面的操作提示输入股票名称，来获取股票代码、当日股票价格、历史股票价格、股票K线图等信息。若用户输入的股票名称不属于任何一支股票，则显示输入名称错误。

### 4.7.2 IPO图

### 4.7.3 功能

根据用户输入的股票名称，对该股票信息进行查询并显示。

### 4.7.4 输入项

| **名称** | **标识** | **类型和格式** | **输入方式** |
| --- | --- | --- | --- |
| 按名称查询股票标识 | nameQuery\_stock\_flag | bool | 外部选择->脚本转换 |
| 按名称查询股票名称 | nameQuery\_stock\_name | string | 外部输入 |

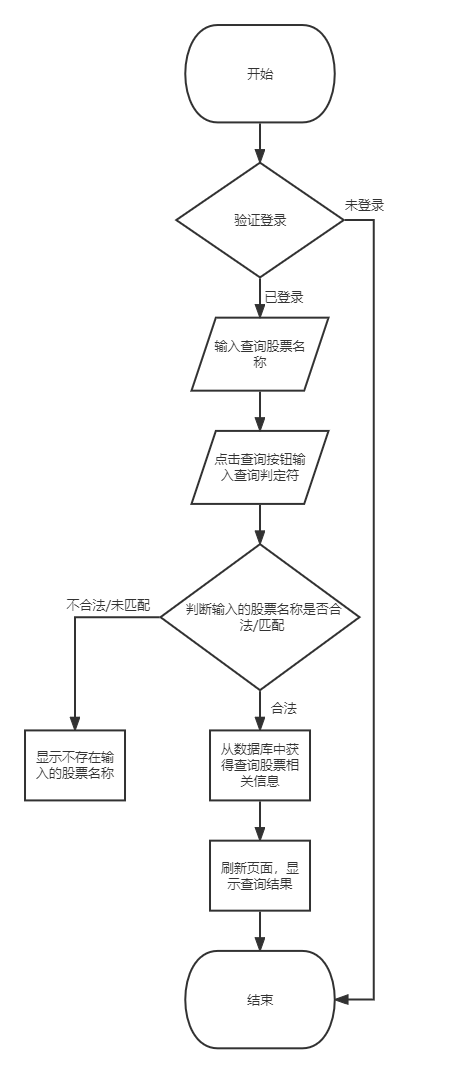
### 4.7.5 输出项

| **名称** | **标识** | **类型和格式** | **输出方式** |
| --- | --- | --- | --- |
| 按名称查询股票结果 | nameQuery\_stock\_result | bool | 由脚本输出 |

### 4.7.6 设计方法（算法）

if 账户已登录  
​ 连接数据库  
​ 用户输入股票名称  
​ 用户点击查询键  
​ 检测输入的股票名称是否匹配  
​ if 用户输入的股票名称是否匹配  
​ 系统根据数据库内信息返回股票代码、当日股票价格、历史股票价格、股票K线图等信息，刷新页面显示在用户界面上。  
​ else  
​ 显示输入了错误的股票名称  
​else  
​ 返回失败信息（跳转到登录界面）

### 4.7.7 流程图



### 4.7.8 测试计划

| **输入数据** | **预期结果** |
| --- | --- |
| 未登录，输入股票名称，点击查询按钮 | 提示未登陆错误，返回登陆界面 |
| 已登录，输入不合法的股票名称， 点击查询按钮 | 显示不存在输入的股票名称 |
| 已登录，输入合法的股票名称， 点击查询按钮 | 显示输入股票名称 所对应的股票相关信息 |

## 4.8 按代码查询股票

### 4.8.1 模块概述

在这个模块中，用户可以在信息查询子系统中通过账号密码登录进入查询页面，从而使用按名称查询的功能。在查询时，用户通过界面的操作提示输入股票代码，来获取股票名称、当日股票价格、历史股票价格、股票K线图等信息。若用户输入的股票名称不属于任何一支股票，则显示输入名称错误。

### 4.8.2 IPO图

### 4.8.3 功能

根据用户输入的股票代码，对该股票信息进行查询并显示。

### 4.8.4 输入项

| **名称** | **标识** | **类型和格式** | **输入方式** |
| --- | --- | --- | --- |
| 按代码查询股票标识 | idQuery\_stock\_flag | bool | 外部选择->脚本转换 |
| 按代码查询股票名称 | idQuery\_stock\_name | string | 外部输入 |

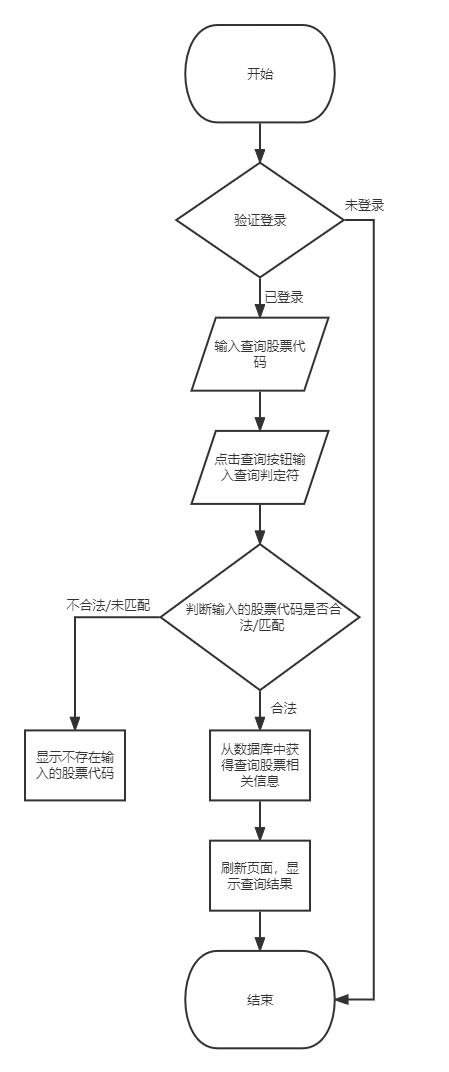
### 4.8.5 输出项

| **名称** | **标识** | **类型和格式** | **输出方式** |
| --- | --- | --- | --- |
| 按代码查询股票结果 | idQuery\_stock\_result | bool | 由脚本输出 |

### 4.8.6 设计方法（算法）

if 账户已登录  
​ 连接数据库  
​ 用户输入股票代码  
​ 用户点击查询键  
​ 检测输入的股票代码是否匹配  
​ if 用户输入的股票代码成功匹配  
​ 系统根据数据库内信息返回股票名称、当日股票价格、历史股票价格、股票K线图等信息，刷新页面显示在用户界面上。  
​ else  
​ 显示输入了错误的股票代码，返回错误值  
​else  
​ 返回失败信息（跳转到登录界面）

### 4.8.7 流程图



### 4.8.8 测试计划

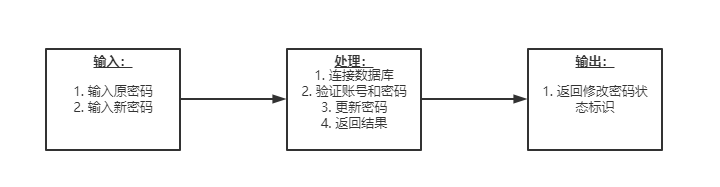
| **输入数据** | **预期结果** |
| --- | --- |
| 未登录，输入股票代码，点击查询按钮 | 提示未登陆错误，返回登陆界面 |
| 已登录，输入不合法的股票代码， 点击查询按钮 | 显示不存在输入的股票代码 |
| 已登录，输入合法的股票代码， 点击查询按钮 | 显示输入股票代码 所对应的股票相关信息 |

## 4.9 修改密码

### 4.9.1模块概述

在这个模块中，用户可以通过界面上的修改密码功能，投资者在正确输入原有密码以后，可以修改资金帐户投资取款密码和交易密码。

### 4.9.2 IPO图



### 4.9.3 功能

用于修改账户密码

### 4.9.4 输入项

| **名称** | **标识** | **类型和格式** | **输入方式** |
| --- | --- | --- | --- |
| 原密码 | OriginalPassword | VarChar(20) | 用户输入 |
| 新密码 | NewPassword | VarChar(20) | 用户输入 |

### 4.9.5 输出项

| **名称** | **标识** | **类型和格式** | **输出方式** |
| --- | --- | --- | --- |
| 修改密码状态标识 | UpdatePasswordStatusFlag | bool | 脚本输出 |

### 4.9.6 设计方法

用户输入原密码  
连接数据库  
查询用户密码  
if 原密码正确  
   用户输入新密码  
   if 新密码与原密码相同  
      输出错误信息（新密码不能与原密码相同）  
    更新用户密码  
else   
   输出错误信息（原密码错误）

### 4.9.7 流程图

### 4.9.8 测试计划

| **输入数据** | **预期结果** |
| --- | --- |
| 输入错误的原密码 | 返回错误信息，原密码错误 |
| 输入正确的原密码和相同的新密码 | 返回错误信息，新密码不能与原密码相同 |
| 输入正确的原密码和新密码 | 返回成功信息，修改成功 |

# 5 接口设计

## 5.1 登录界面

​ 用户输入正确的账户和密码即可登录。需要注意的是，注册即开户功能需要在资金账户和证券账户子系统中实现，之后才可以进行登录。



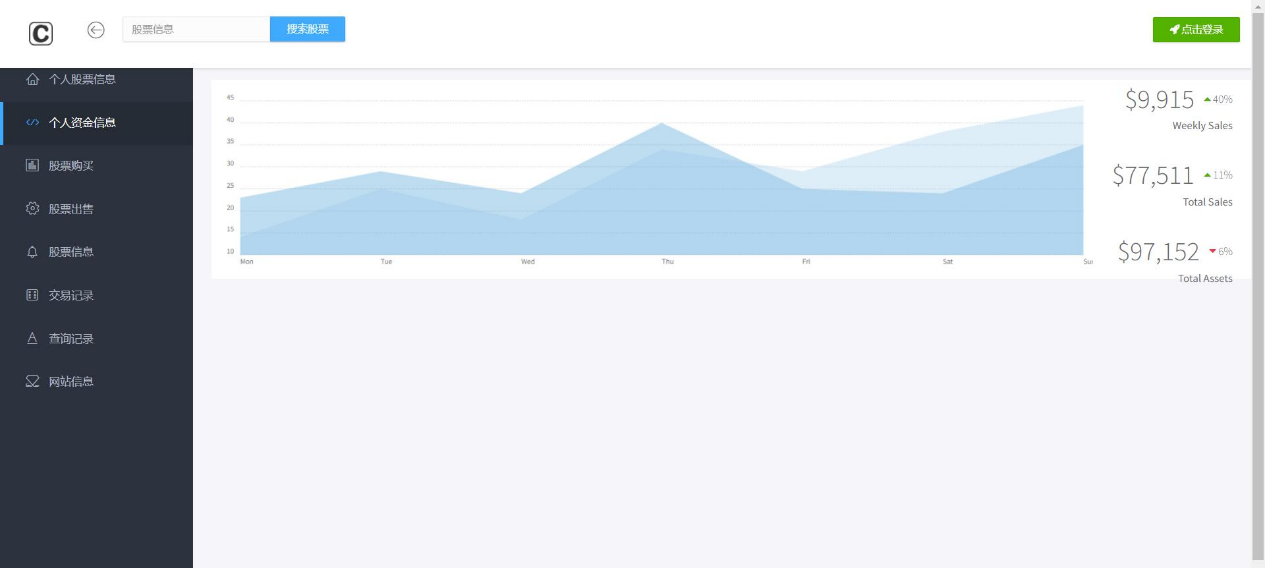
## 5.2 个人股票信息页面

​ 用户可查询自己的股票，并通过点击某一股票查看其涨落情况。



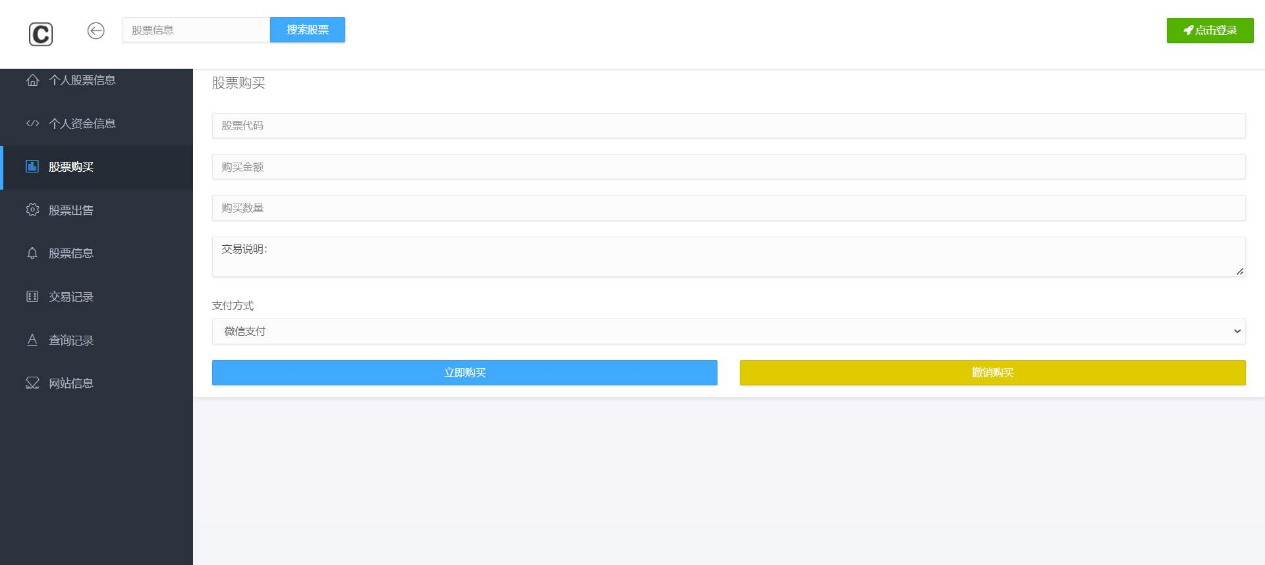
## 5.3 个人资金信息页面

​ 用户可查询自己的资金及最近涨落情况。



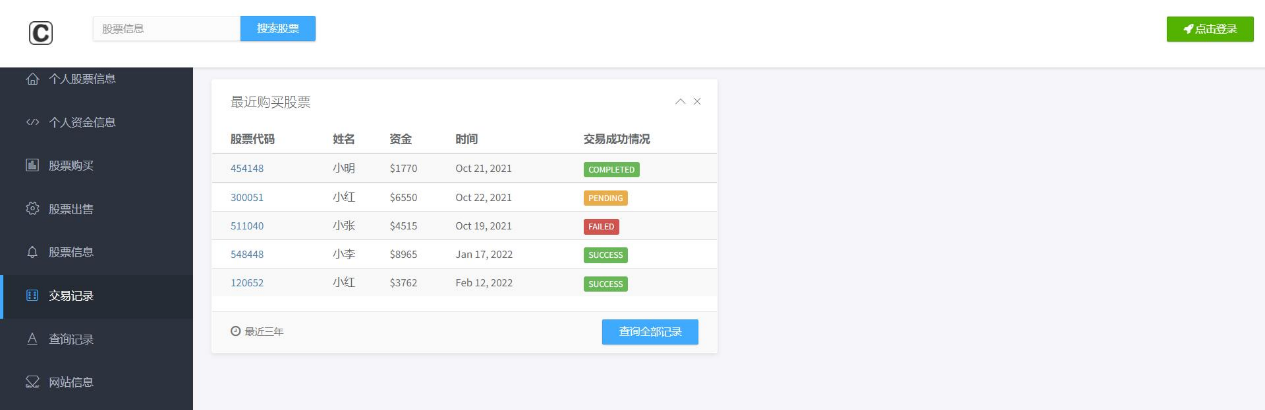
## 5.4 股票购买页面

​ 用户可以通过输入股票代码、购买价格、购买数量进行股票购买的操作。



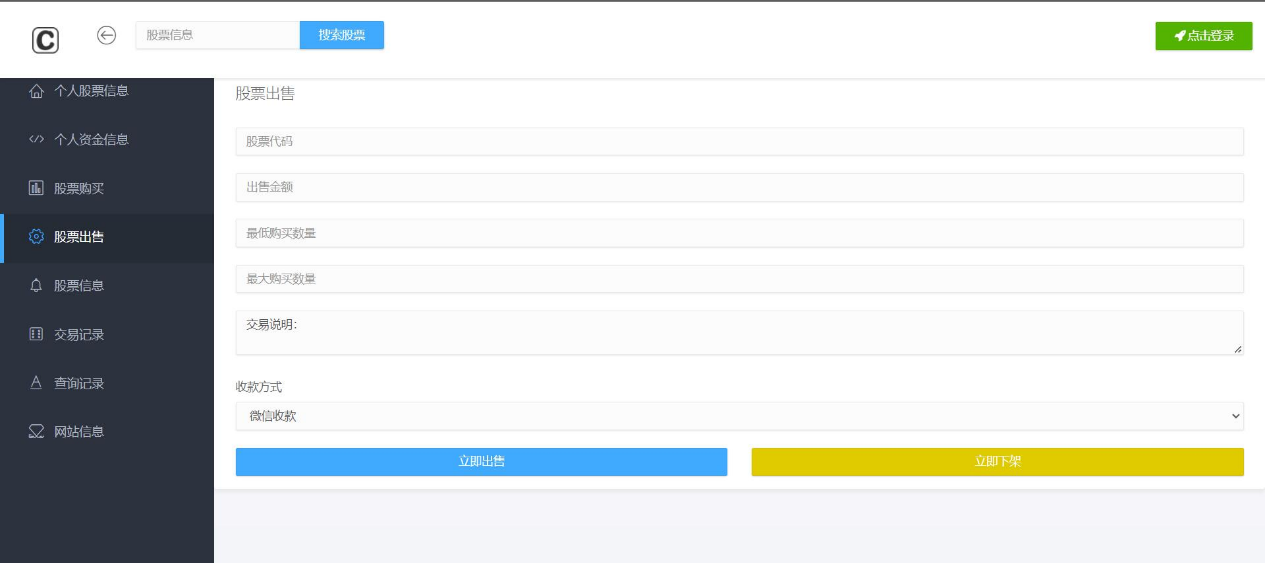
## 5.5 股票出售页面

​ 用户可以通过输入股票代码、出售价格、出售数量进行股票出售的操作。



## 5.6 交易记录页面

​ 用户可查询到最近的交易记录。



## 5.7 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。

# 6 运行设计

## 6.1 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（侧边栏、导航栏等）。

## 6.2 运行控制

用户首先通过输入正确的账户密码在登录模块中进入交易客户端子系统。之后，用户可以通过侧边栏和导航栏选择功能，如查询股票，交易记录、个人资金账户与持仓等。

### 6.2.1查询证券账户持有的股票

通过界面上的证券信息查询功能，投资者，可以查看与该资金帐户相关联的证券帐户内所有股票的信息，具体包括：

* 股票名称：投资者持有的股票名称，股票代码
* 股票总数 S ：投资者持有每个股票的股数。
* 股票现在的价格：投资者持有的某个股票目前的市场价格。
* 股票持有成本：系统统计出投资者对每个股票的每股持有成本。
* 持有股票损益：

系统计算出每种股票目前市场上的价格减去股票持有成本乘积 持有股数，就是股票损益。

### 6.2.2查询资金账户

通过界面上的资金信息查询功能，投资者可以查看资金帐户内资金信息 ，包括：

* 可用资金：

投资者可以用来购买股票或者立即取出的资金数额。

* 冻结资金：

投资者已经发出股票购买指令所需要的资金，虽然股票购买指令尚未执行，但是这部分资金已经冻结。当投资者发出撤销指令成功，冻结的资金又可以转化为可用资金。

### 6.2.3发出购买股票指令

通过界面上的购买股票功能可以发出购买某一股票的指令。投资需要输入以下的信息：

* 股票代码
* 购买价格

系统根据当前股票的价格给出参考价，用户可以接受也可以修改。

* 购买数量。

购买的最大的数量，由帐户中的可用资金数量以及购买价格决定。购买股票总价不能超过资金帐户中可用资金总量。

### 6.2.4发出出售股票指令

通过界面上的出售功能，可以发出出售投资者持有股票的指令。指令包括以下信息：

股票代码，必须是投资者持有的股票。

出售价格（系统根据当前股票的价格给出参考价），用户可以接受也可以修改。

出售数量。出售的最大的数量，由投资者持有的股数决定。出售股票总数不能 超过投资者持有的总数。

### 6.2.5撤销指令

通过界面上的撤销功能，可以撤销发出的指令。指令如果尚未执行，则可以撤销，如果指令已经执行，拒绝投资者的请求。

### 6.2.6显示交易结果

如果指令执行成功将会更新资金帐户信息和证券帐户信息。包括以下几个方面：

* 用户指令的状态改变，从未执行变成执行成功。
* 股票帐户中股票持有数量的改变。
* 资金帐户中，可用资金数量（出售股票）和冻结资金数量（购买股票）的更新。

### 6.2.7查询股票

通过在界面的查询功能中输入股票的名字或代码，实现对股票基本信息的查询，为了简 化期间，我们要求能够输出以下功能：

* 股票最新成交价格
* 当前购买指令的最高价格，当前出售指令的最低价格
* 当日最高，最低成交价格
* 本周最高，最低成交价格
* 本月最高，最低成交价格
* 股票的重要公告

# 7 系统出错设计

## 7.1 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 | 部分用户恶意实施 SQL 注入，导致正常查询出错 | 使用参数绑定法构建 SQL 语句，对用户填入的信息进行字符串过滤 |

## 7.2 补救措施

### 7.2.1 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。

### 7.2.2 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

### 7.2.3 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。

## 7.3 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。

# 8 需求回溯

## 8.1 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 | 用户身份有效性校验 | 用户登录模块 | 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。 |
| A02 | 为用户提供显示个人持仓和资金账户的界面 | 查询持仓和资金账户模块 | 服务端验证Session是否为已登录状态，  若已登录，通过用户ID向数据库查询持仓和资金账户。 得到返回数据后，系统显示用户的持仓和资金账户。 |
| A03 | 为用户提供发出购买股票指令的操作 | 股票交易模块 | 服务端验证股票代码是否正确， 购买价格和购买数量是否符合规则， 如果没有问题，系统就生成购买指令， 并冻结该用户用于购买的资金，形成购买交易记录。 交易成功后更新用户持仓和资金账户。 |
| A04 | 为用户提供发出出售股票指令的操作 | 股票交易模块 | 服务端验证股票代码是否正确， 出售价格和出售数量是否符合规则， 如果没有问题，系统就生成出售指令， 并形成出售交易记录。成功后更新用户持仓和资金账户。 |
| A05 | 为用户提供撤销指令的操作 | 指令控制模块 | 服务端验证该指令是否未完成， 如果完成了，则拒绝撤销指令， 如果未完成，则可以撤销指令， 并恢复持仓和资金账户。 |

## 8.2 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。 |