

股票交易系统

之交易客户端子系统

总体设计说明书

[V4.0]

拟 制 人__赵一帆 3080103820__

审 核 人__李佩聪 3080102538__

批 准 人__叶志强 3080103781__

组长：叶志强

组员：赵一帆、李佩聪、杨诚、毛诗扬

日期：2010-10-17

目 录

- 1.引言.....5
 - 1.1 编写目的.....5
 - 1.2 背景.....5
 - 1.3 定义.....6
 - 1.4 参考资料.....7
 - 1.5 项目概述.....8
- 2.总体设计.....9
 - 2.1 需求规定.....9
 - 2.1.1 系统功能.....9
 - 2.1.2 系统性能.....9
 - 2.1.3 输入输出要求.....9
 - 2.1.4 数据管理能力要求.....10
 - 2.2 运行环境.....10
 - 2.2.1 设备.....10
 - 2.2.2 支持软件.....11
 - 2.3 基本设计概念和处理流程.....11
 - 2.4 结构.....12
 - 2.4.1 按功能划分（横向划分）：.....12
 - 2.4.2 按处理流程(客户端/服务器)划分（纵向划分）.....15
 - 2.5 内部接口.....17
 - 2.6 人工处理过程.....17
 - 2.7 尚未解决的问题.....17
- 3.详细设计.....18
 - 3.1 用户登录模块设计说明.....18
 - 3.1.1 模块描述.....18
 - 3.1.2 功能.....18
 - 3.1.3 性能.....18
 - 3.1.4 输入项.....19
 - 3.1.5 输出项.....19
 - 3.1.6 设计方法（算法）.....19
 - 3.1.8 接口.....21
 - 3.1.9 测试计划.....21
 - 3.2 股票查询模块设计说明.....22
 - 3.2.1 模块描述.....22
 - 3.2.2 功能.....22
 - 3.2.3 性能.....22
 - 3.2.4 输入项.....22
 - 3.2.5 输出项.....23
 - 3.2.6 设计方法.....24
 - 3.2.7 流程逻辑.....25

| | |
|------------------------|----|
| 3.2.8 接口: | 25 |
| 3.2.9 测试计划: | 26 |
| 3.3 股票购买模块设计说明 | 27 |
| 3.3.1 模块描述 | 27 |
| 3.3.2 功能 | 27 |
| 3.3.3 性能 | 27 |
| 3.3.4 输入项 | 27 |
| 3.3.5 输出项 | 28 |
| 3.3.6 设计方法 | 28 |
| 3.3.7 流程逻辑 | 29 |
| 3.3.8 接口 | 30 |
| 3.3.9 测试计划 | 30 |
| 3.4 股票出售模块设计说明 | 31 |
| 3.4.1 模块描述 | 31 |
| 3.4.2 功能 | 31 |
| 3.4.3 性能 | 31 |
| 3.4.4 输入项 | 31 |
| 3.4.5 输出项 | 32 |
| 3.4.6 设计方法 | 32 |
| 3.4.7 流程逻辑 | 34 |
| 3.4.8 接口 | 34 |
| 3.4.9 测试计划 | 35 |
| 3.5 买卖指令撤销模块设计说明 | 35 |
| 3.5.1 模块描述 | 35 |
| 3.5.2 功能 | 35 |
| 3.5.3 性能 | 36 |
| 3.5.4 输入项 | 36 |
| 3.5.5 输出项 | 36 |
| 3.5.6 设计方法 | 36 |
| 3.5.7 流程逻辑 | 38 |
| 3.5.8 接口 | 38 |
| 3.5.9 测试计划 | 39 |
| 3.6 资金账户查询模块设计说明 | 40 |
| 3.6.1 模块描述 | 40 |
| 3.6.2 功能 | 40 |
| 3.6.3 性能 | 40 |
| 3.6.4 输入项 | 41 |
| 3.6.5 输出项 | 41 |
| 3.6.6 设计方法 | 41 |
| 3.6.7 流程逻辑 | 42 |
| 3.6.8 接口 | 42 |
| 3.6.9 测试计划 | 42 |
| 3.7 修改密码模块设计说明 | 43 |
| 3.7.1 模块描述 | 43 |

| | |
|---------------------------|-----------|
| 3.7.2 功能..... | 43 |
| 3.7.3 性能..... | 43 |
| 3.7.4 输入项..... | 44 |
| 3.7.5 输出项..... | 44 |
| 3.7.6 设计方法..... | 44 |
| 3.7.7 流程逻辑..... | 45 |
| 3.7.8 接口..... | 46 |
| 3.7.11 测试计划..... | 46 |
| 3.8 账户持股信息查询设计说明..... | 47 |
| 3.8.1 模块描述..... | 47 |
| 3.8.2 功能..... | 47 |
| 3.8.3 性能..... | 47 |
| 3.8.4 输入项..... | 47 |
| 3.8.5 输出项..... | 48 |
| 3.8.6 设计方法..... | 48 |
| 3.8.7 流程逻辑..... | 49 |
| 3.8.8 接口..... | 49 |
| 3.8.11 测试计划..... | 49 |
| 3.9 数据安全性检测模块设计说明..... | 50 |
| 3.9.1 模块描述..... | 50 |
| 3.9.2 功能..... | 50 |
| 3.9.3 输入项..... | 50 |
| 3.9.4 输出项..... | 51 |
| 3.9.5 设计方法..... | 51 |
| 3.9.6 测试计划..... | 51 |
| 3.10 同步服务器信息模块..... | 51 |
| 3.10.1 模块描述..... | 51 |
| 3.10.2 功能..... | 52 |
| 3.10.3 输入项..... | 53 |
| 3.10.4 设计方法..... | 53 |
| 3.11 ConnectionDB 设计..... | 53 |
| 4.接口设计..... | 55 |
| 4.1 用户接口（界面设计）..... | 55 |
| 4.1.1 主界面..... | 55 |
| 4.1.2 账户登陆模块..... | 56 |
| 4.1.3 修改密码模块..... | 56 |
| 4.1.4 股票购买模块..... | 57 |
| 4.1.5 股票出售模块..... | 57 |
| 4.1.6 撤销买卖指令模块..... | 58 |
| 4.1.7 资金账户查询..... | 59 |
| 4.1.8 股票查询模块..... | 59 |
| 4.1.9 持股信息查询模块..... | 60 |
| 4.2 外部接口..... | 60 |
| 4.3 内部接口..... | 60 |

5.系统数据结构设计.....60

5.1 概念结构设计.....61

5.2 逻辑结构设计.....61

5.3 物理结构设计.....62

5.4 本组附加数据库（V4.0 新增）.....65

6.运行设计..... 66

6.1 运行模块组合.....66

6.2 运行控制.....66

6.3 运行时间.....67

7.系统出错处理设计.....67

7.1 出错信息.....67

7.2 补救措施.....68

7.3 系统维护设计.....68

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。本设计说明书的编写目的，是以本项目的需求分析说明书为依据，从总体设计的角度，明确股票交易系统的总体架构，流程，数据结构，数据库设计。并且，本说明书将为今后关于股票交易系统子系统—交易客户端详细设计说明书提供依据。

目的在于：

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

本说明书的预期读者包括：

- ❖ 软件客户
- ❖ 项目经理
- ❖ 项目开发人员
- ❖ 软件质量分析员
- ❖ 软件维护人员

1.2 背景

a. 软件系统名称

股票交易系统

b. 任务提出者

浙江大学软件工程基础课程任课老师—王新宇

c. 开发者

由浙江大学 2010-2011 学年秋学期软件工程基础课程部分学生组成的项目组

d. 用户

股民、系统管理员、证券工作人员（由于此软件是作为课程实践而布置的作业，我们将用户假定为浙江大学软件工程基础课程助教）

e. 实现该软件的计算机网络

由若干台 PC 机组成的局域网

f. 该软件系统同其他系统或其他机构的基本的相互来往关系

项目组可使用一个网上现有的股票交易系统（中信证券至信版网上交易系统）来加深对股票交易系统的了解，并通过学习一些基本的股票交易知识来为开发工作打下基础。

g. 相关背景介绍

为全面提高学生创新和实践能力，浙江大学软件工程基础课程分为课堂教学和综合性实验两部分。综合性实验采取分组形式完成，每5~6 个学生为1 组，分别设有组长、主程序员、程序员、测试员、文档员等角色，全面锻炼学生的系统设计与实现能力、测试能力、组织文档能力、以及团队合作能力。综合性实验从实际问题中抽取，贯穿全部重要知识点，通过7 个步骤的阶段性验收，锻炼学生综合运用每个环节所学知识解决实际问题的能力。本次课程，教师选取股票交易系统作为综合性实验题目。

1.3 定义

MySQL:一个小型关系型数据库管理系统。

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

PHP:PHP 是一种 HTML 内嵌式的语言，是一种在服务器端执行的嵌入 HTML 文本的脚本语言，语言的风格有类似于 C 语言，被广泛的运用。

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。

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

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

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

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。

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

1.4 参考资料

- 《股票交易系统介绍文档》(课程资料)

提供者浙江大学软件工程基础课程任课教师-王新宇

- 《股票交易系统可行性分析报告》

作者浙江大学软件学院学生-李佩聪

- 《股票交易系统项目开发计划》

作者浙江大学软件学院学生-叶志强

- 《股票交易系统项目需求说明书》

作者浙江大学软件学院学生-叶志强

- 《软件工程-实践者的研究方法》

作者Roger S. Pressman (美)

译者郑人杰马素霞白晓颖

出版社机械工业出版社

出版时间2007 年1 月

- 《软件编档导论》

作者金波

出版社清华大学出版社

出版时间2008 年5 月

软件需求说明书国家标准规范 (GB856T——88)

提供者浙江大学软件文档写作课程任课教师-金波

1.5 项目概述

一个完整的股票交易系统，应实现以下功能模块：

➤ 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。

➤ 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。

➤ 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。

➤ 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查询。

➤ 交易系统管理系统

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。

其中，我组负责的是交易客户端。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。同时还要保证第一次登陆时的安全证书的认证。

交易客户端的用户为普通股民。股民教育水平和技术专长参差不齐，有经验丰富的资深股民，也有刚刚进入股市的新手。此外，股民频繁使用该系统进行股票交易及获取股票信息。

因此，本软件客户端的界面应尽量直观、易懂，操作应尽量简单、容易上手、易于操作，性能应尽量实时、准确。

此次软件开发出于实验性目的，暂无经费划拨，希望项目开发团队能够充分利用现有资源高效地开发出符合需求的系统。

股票交易系统由 5 个子系统构成，每个子系统完成其中一项功能，最后进行系统集成。此项目开发期限较短，2010 年 11 月 7 日下午 5:00 之前要交付本系统，因此各个子系统务必要安排好工作进度，保证有一定的时间进行系统集成。交易客户端与其它几个子系统均有关联，因此，务必要做好与其它几个子系统开发团队的沟通协商工作。

此次软件开发本着实验教学的目的，同时访问本系统的客户端较少，故对数据处理能力等方面要求不高。

2. 总体设计

2.1 需求规定

2.1.1 系统功能

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。

2.1.2 系统性能

本系统的性能将由服务器端数据库，网络数据传输延时，以及并发访问该系统的用户数量决定。客户端方面将实现 AJAX 技术，减轻服务器压力，增加带宽利用率，并为用户提供良好的界面体验。

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。

2.1.4 数据管理能力要求

安全：服务器将予以数据库最高等级的保护，防止黑客从后台下载数据库，防止通过网页SQL注入的方式从数据库中获取信息或者破坏数据库。

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

条件有限，本项目不能提供强大可靠的服务器，运行时只能用性能较好的PC机代替。

2.2.1 设备

计算机：

CPU：≥2.0GHz(如Intel Mobile Core 2 Duo T5800)

内存：≥2.0GB

外围设备：

键盘：能用即可

鼠标：能用即可

显示器：能用即可

硬盘：≥100GB

硬盘转速：≥7200rpm

通讯设备：

网线：具有良好数据传输能力

网卡：100M

客户端的设备要求如下：

能够上网的计算机。

2.2.2 支持软件

服务器端：

操作系统：Windows 2003 Server 或 Windows xp

Php:php5.0

数据库：MySQL

Web 服务器：Apache

MySQL 管理软件：PHPMYAdmin 或其他软件

开发工具：记事本，dreamweaver3.0

客户端：

浏览器：IE6、IE7 浏览器

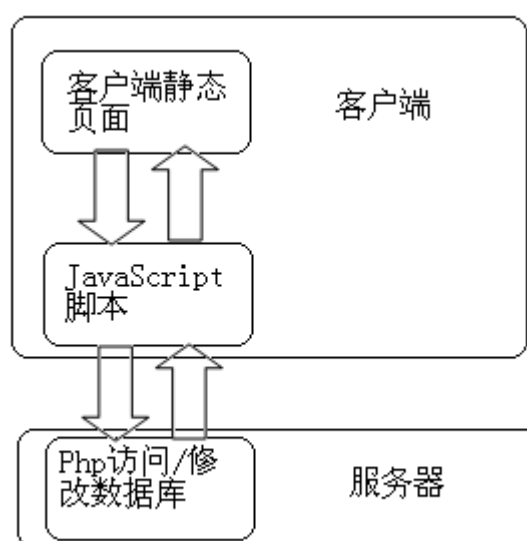
2.3 基本设计概念和处理流程

本系统虽然名为交易客户端，其实也分为客户端和服务端：

服务器：以 apache 为 web 服务器，服务器端采用 php 脚本语言编写。数据库采用 mysql。

客户端：浏览网页主要采用 IE6.0，7.0 或者 Firefox。采用 Ajax 技术减轻服务器端压力，增加服务器带宽利用率，并使用 JQuery 类库对客户端进行美化。

处理流程图如下：



图表 2.1-交易客户端基本流程

客户端静态页面：通过 JQuery 库，设计一个简洁，美观的界面。

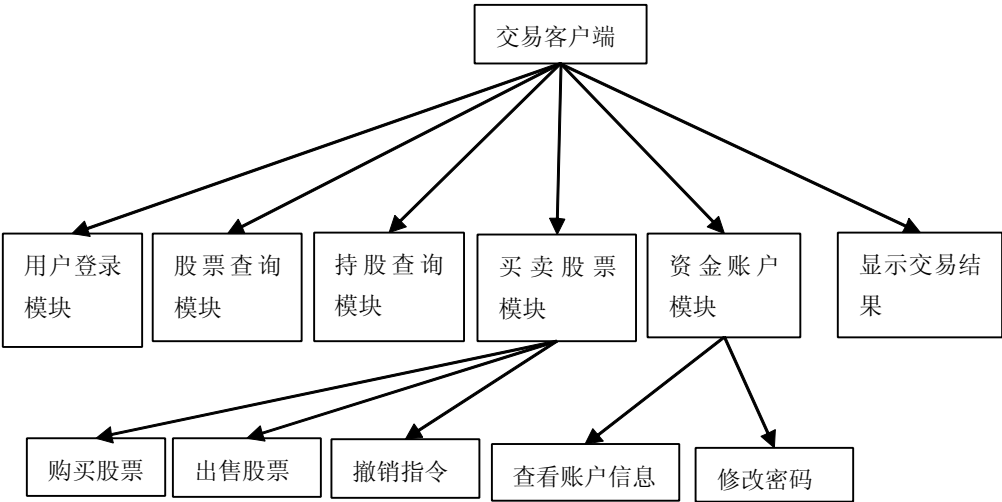
JavaScript 脚本：客户端静态页面中，各种文本框与按键的操作均能触发脚本函数，脚本通过创建 XMLHttpRequest 对象，与服务器进行异步交互。并且，对象接受服务器反馈信息后，能通过脚本函数对客户端静态页面实现无刷新的更新信息。

Php 访问/修改数据库：此模块通过 url 参数方式从客户端得到数据与命令，并且对其进行安全检测，而后按要求访问/修改数据库，并且返回操作结果或是查询结果。

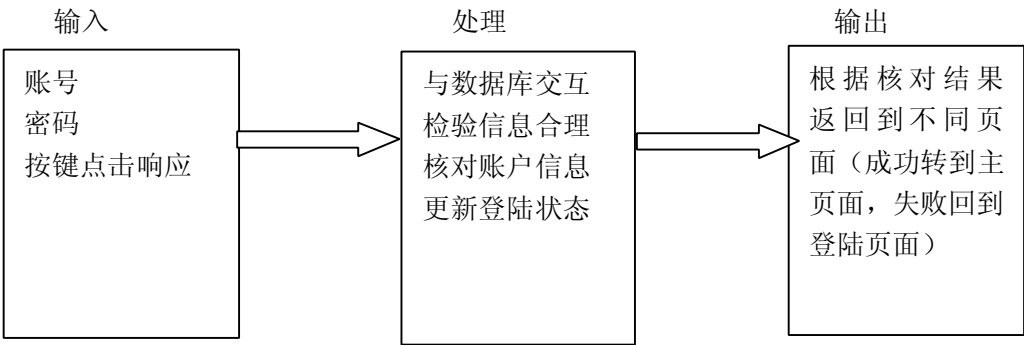
2.4 结构

2.4.1 按功能划分（横向划分）：

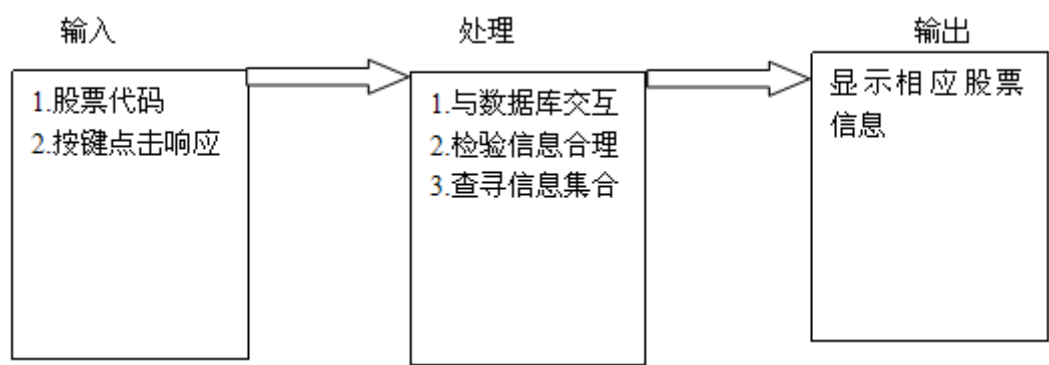
1. 交易客户端 HIPO 图



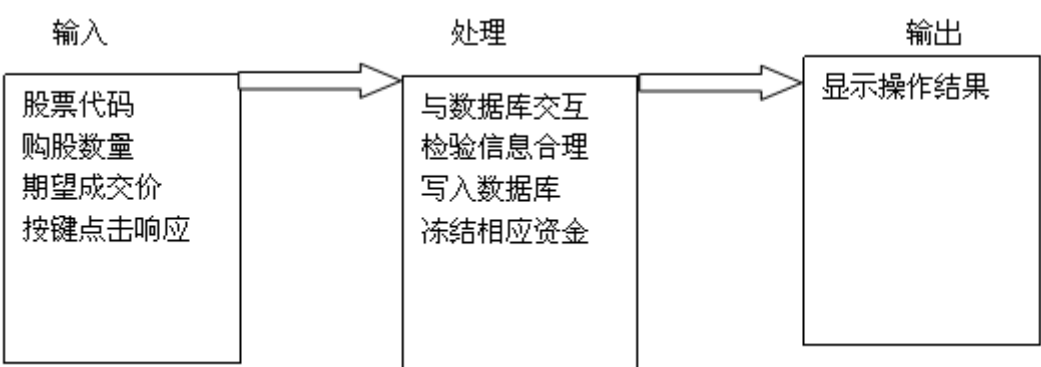
图表 2. 2—交易客户端层次图



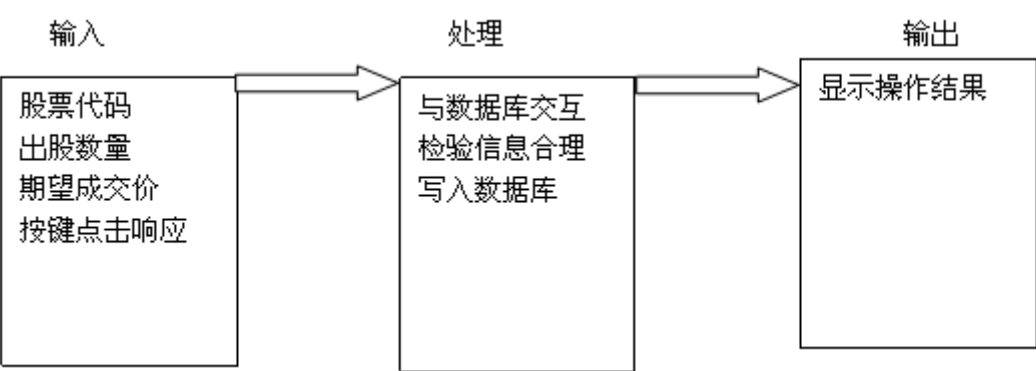
图表 2.3-用户登录模块 IPO 图



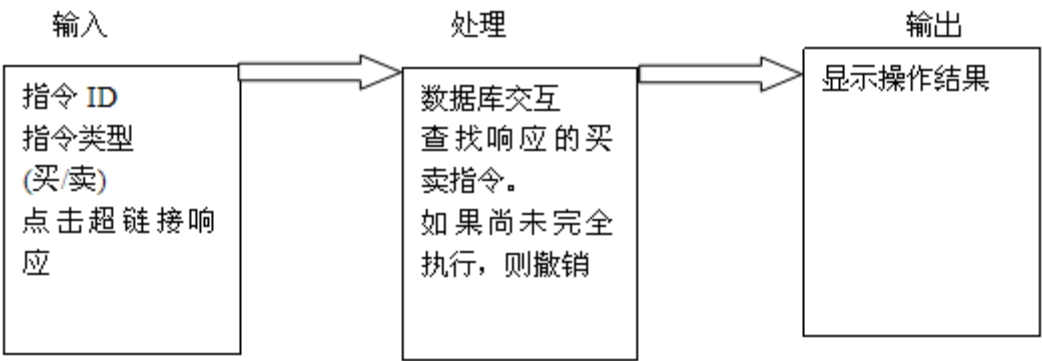
图表 2.4-股票查询模块 IPO 图



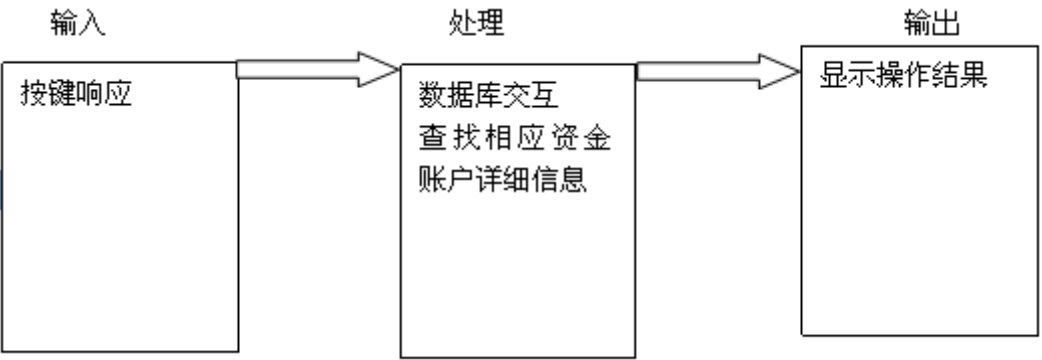
图表 2.5-股票购买模块 IPO 图



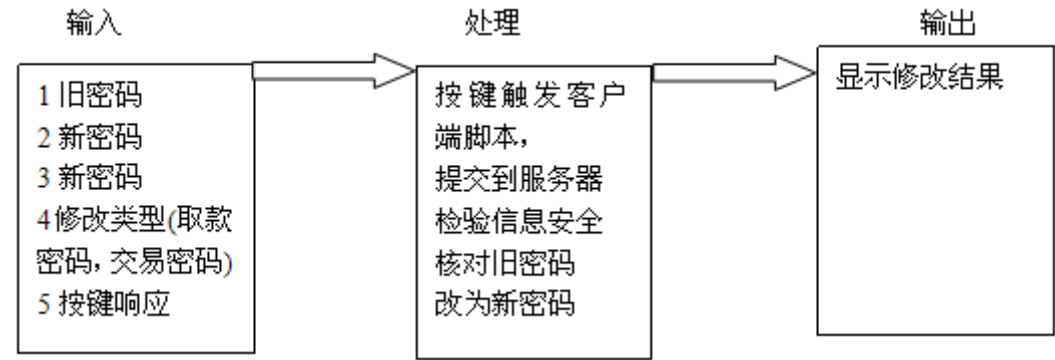
图表 2.6-股票出售模块 IPO 图



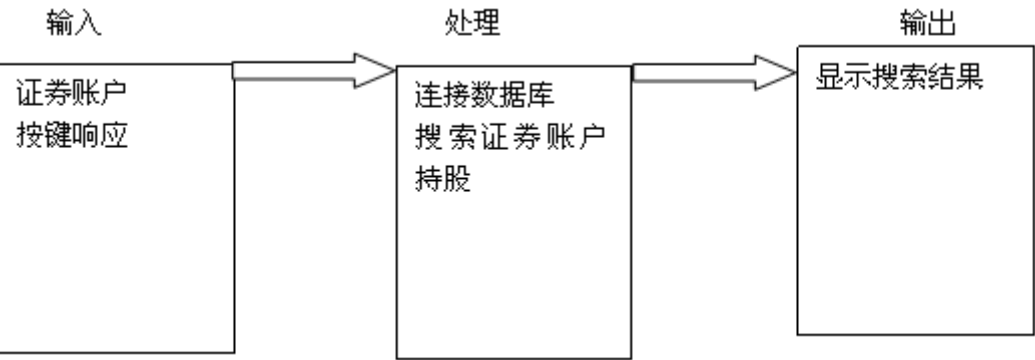
图表 2.7-指令撤销模块 IPO 图



图表 2.8-资金账户查询模块 IPO 图



图表 2.9-密码修改模块 IPO 图

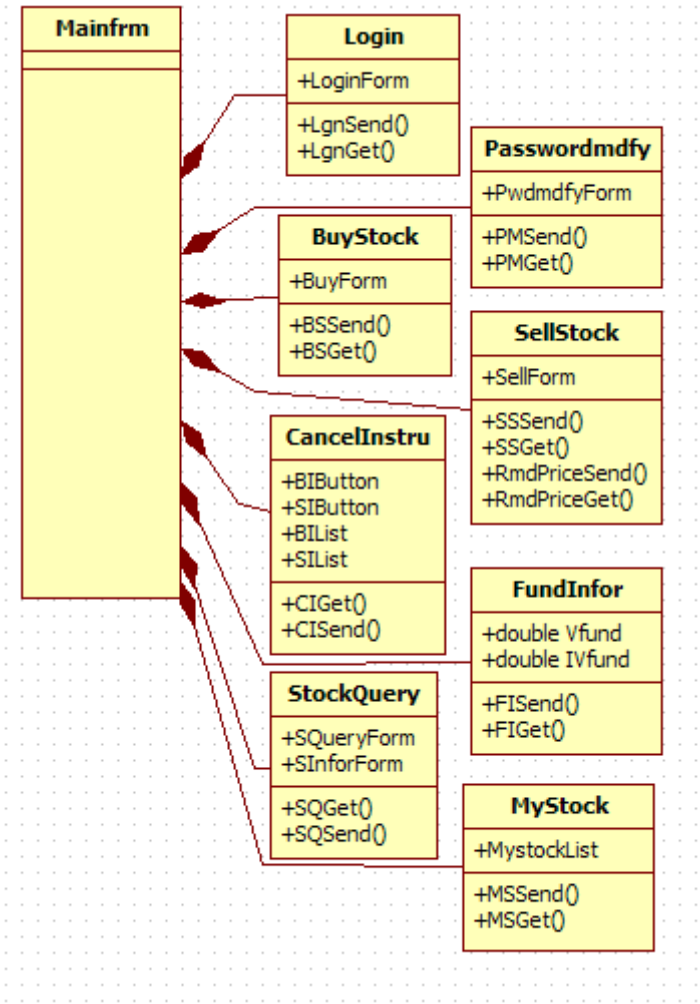


图表 2.10-账户持股查询模块 IPO 图

2.4.2 按处理流程(客户端/服务器)划分（纵向划分）

2.4.2.1 客户端

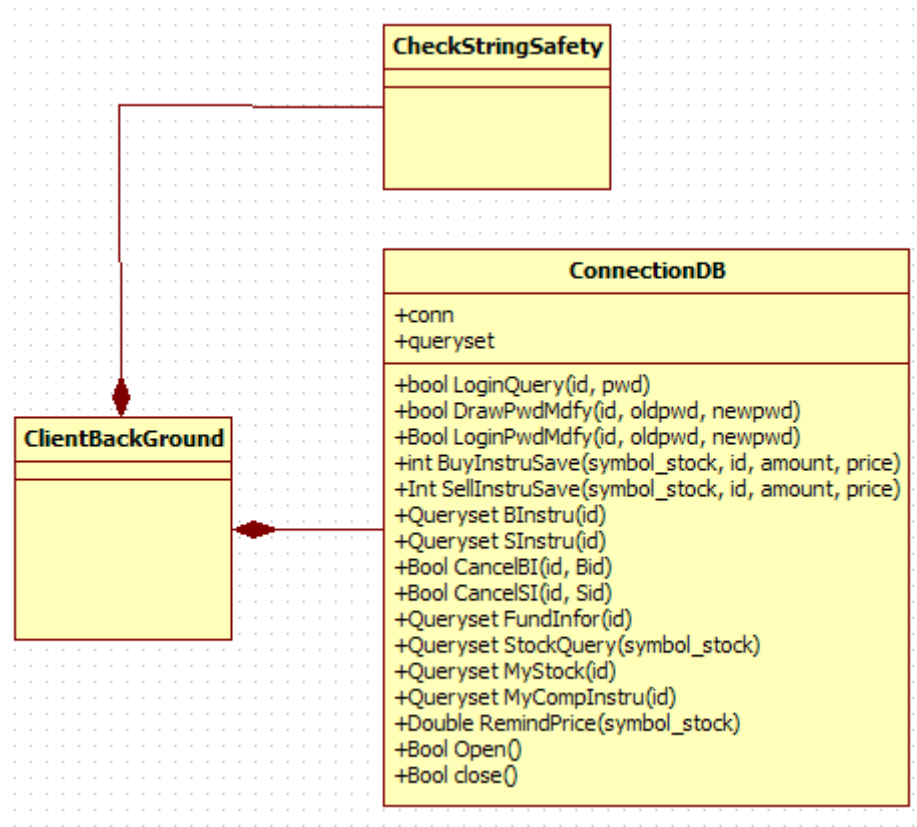
客户端将功能抽象为类,其中 mainfrm 主界面包含了用户的 9 个功能.主要通过 JavaScript 实现。



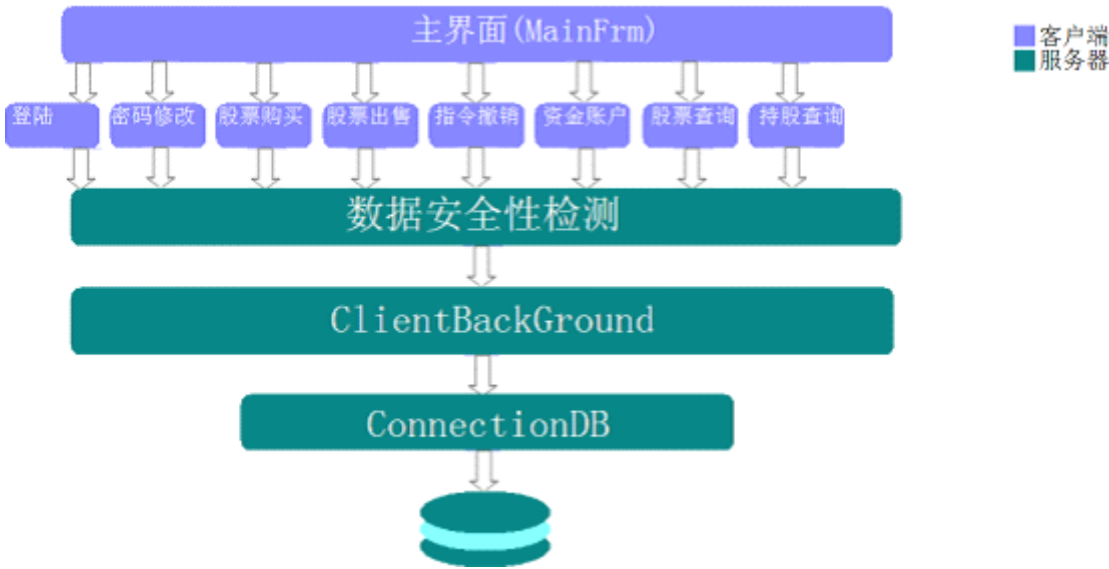
图表-客户端功能抽象类图

2.4.2.2 服务器

服务器主要由 php 实现。



图表-服务器端程序抽象后类图



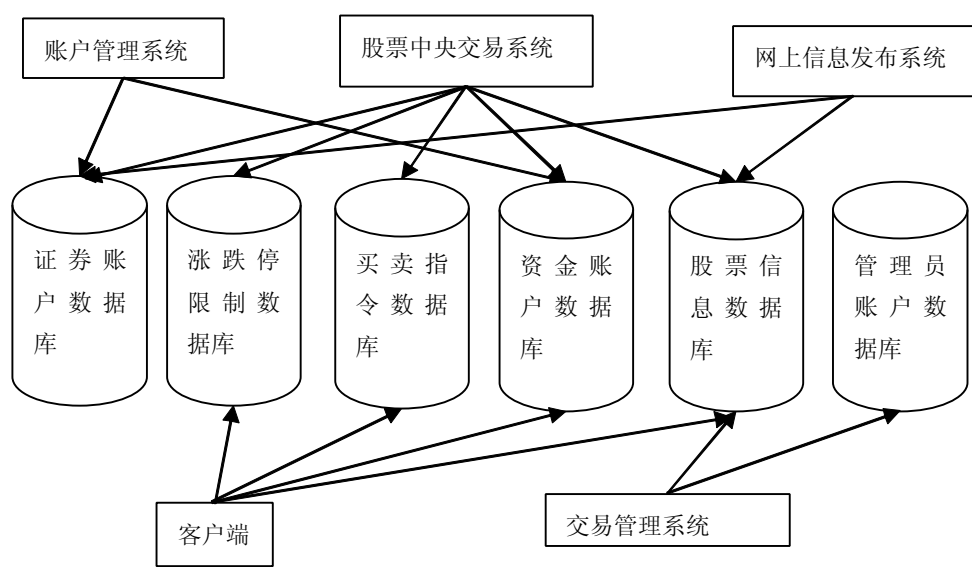
图表-各模块隶属调用关系

2.5 内部接口

本系统内不同模块通过共享数据库完成通信。

| | 账户管理系统 | 股票中央交易系统 | 网上信息发布系统 | 交易系统管理模块 |
|-----|-----------|--------------------------|-----------|-----------|
| 客户端 | 共享资金账户数据库 | 共享买卖指令/涨跌停限/资金账户/股票信息数据库 | 共享股票信息数据库 | 共享股票信息数据库 |

图表 3. 1-五大模块间共享数据库表



图表 3. 1-各模块共享数据库直观图

2.6 人工处理过程

系统出现无法修复错误后，需要开通人工业务帮助用户实现买卖股票业务。

2.7 尚未解决的问题

- 1、客户端登录时，如果是第一次登陆，需要进行安全证书认证，目前尚不知采用何种手段技术对此功能进行实现。

3.详细设计

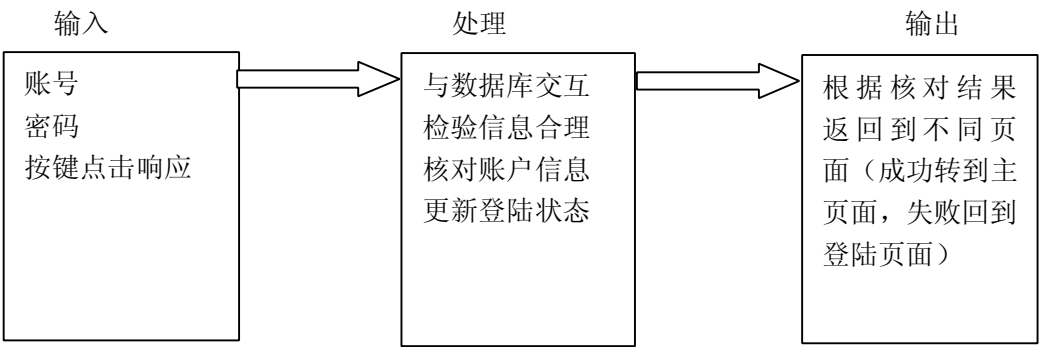
3.1 用户登录模块设计说明

3.1.1 模块描述

本模块以网页表单的形式显示，是用户使用客户端时需要使用的第一个界面，在身份认证成功后主页面的其他功能才能获得授权使用。存在的目的是对用户的身份进行认证。本模块需要在用户第一次登陆时做严格安全认证。

3.1.2 功能

IPO 图如下：



图表 3.1-用户登录模块 IPO 图

3.1.3 性能

输入账号密码后点击登陆，便能进行身份认证。

3.1.4 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|----|----------|--------|----------|
| 账户 | fund_id | string | 在账号框手工输入 |
| 密码 | password | string | 在账号框手工输入 |
| 登陆 | Login | Button | 单击按钮 |

图表 3.2-用户登录模块输入项列表

3.1.5 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|-------------|------------------|-------|
| 登陆结果 | loginresult | Enum{登陆成功， 登陆失败} | 由脚本输出 |

图表 3.3 用户登录模块输出项列表

3.1.6 设计方法（算法）

当客户端点击登陆时，触发脚本函数 LgnSend()，将信息提交给服务器，服务器 ClientBackGround 执行如下 php 代码

```
<?
    If (CheckStringSafety(form.fund_id)
        &&CheckStringSafety(form.password))
        //检测表单中数据安全
    {
        ConnectionDB Db;
        //
        Db.open();
        //创建数据库连接
        If (Db.LoginQuery(form.fund_id,form.password))
        {
            //如果找到相应账户
            If(FirstLogin() &&SecurityCheck())
```

```

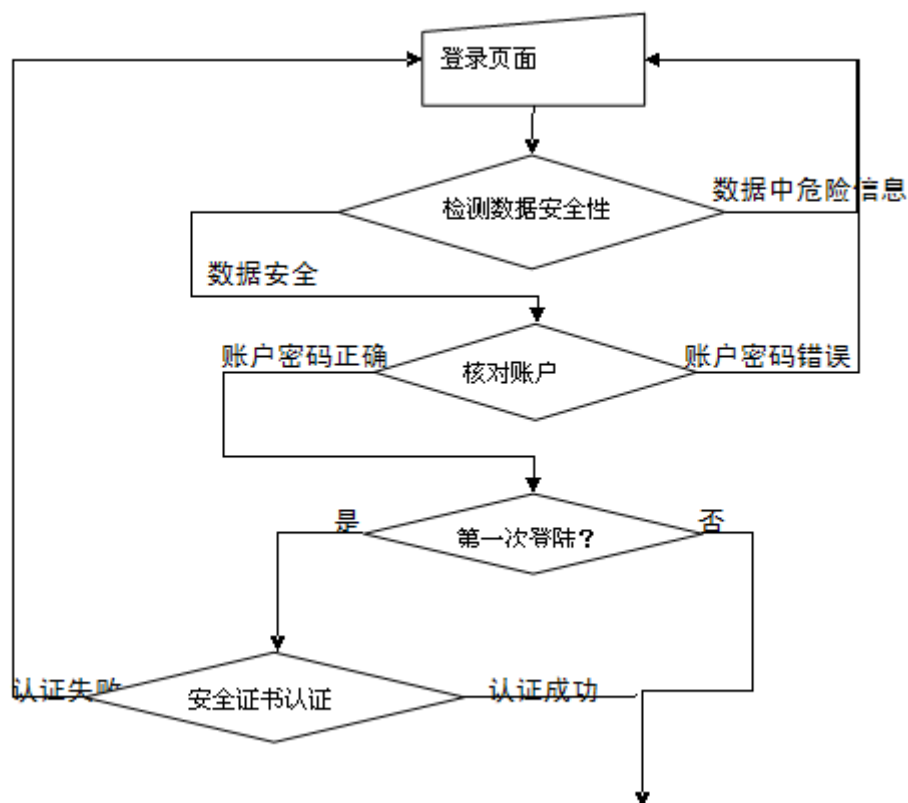
//如果是第一次登陆，安全认证不通过
return;
Session("fund_id")=form.fund_id;
Session("login_state")=true;
Session("time")=time();
//存储登陆状态，证券账户与资金账户
Db.close();
//关闭数据库连接
echo form.fund_id+"先生,您已登陆成功! "
}
echo "您的账户或密码错误! "
}
echo "您的行为非常恶意! "

```

? >

而后执行成功，客户端脚本的 LgnGet() 函数被触发，将信息写入客户端

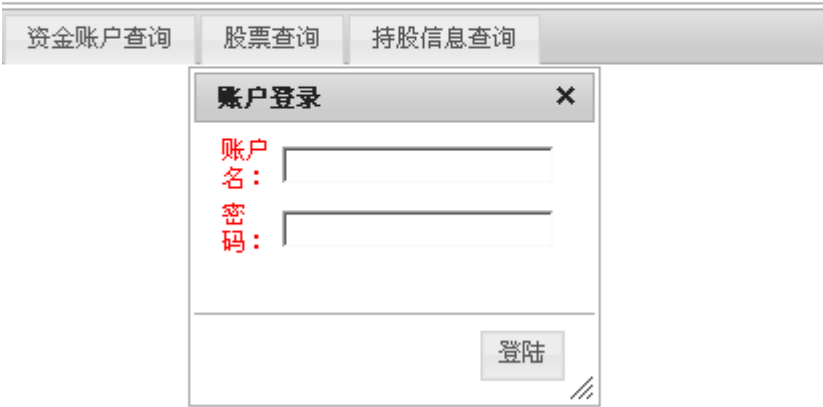
3.1.7 流程逻辑



图表 3.4-用户登录模块流程图

3.1.8 接口

Logo



图表 3.5-用户登录模块用户接口

3.1.9 测试计划

模块单独测试计划。

测试要求通过远程的计算机，要求创建相应证券账户数据库，并且添加测试数据。

| | |
|-----------------------|-------------|
| 输入数据 | 预期结果 |
| 数据库中存在的账户密码 | 获得其他功能使用的授权 |
| 数据库中存在的账户与错误的密码 | 提示错误 |
| 数据库中存在的错误的账户与正确的密码 | 提示错误 |
| 包含特殊字符或者 sql 语句的账户或密码 | 提示警告 |

图表 3.7-用户登录模块测试表

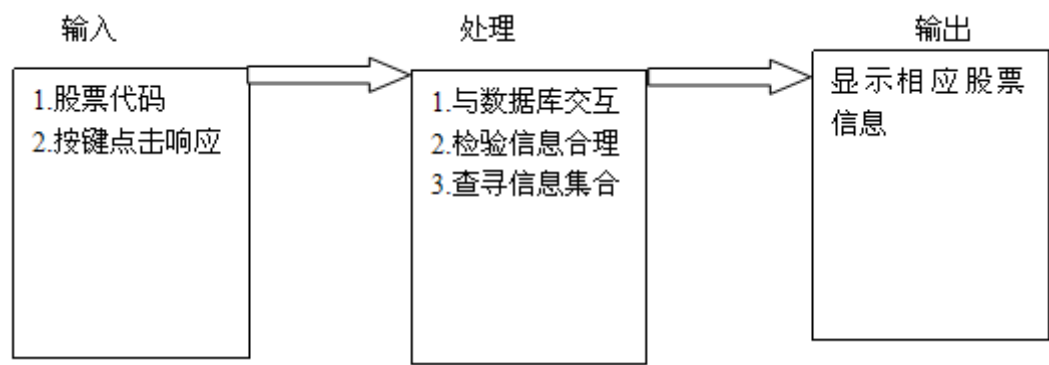
3.2 股票查询模块设计说明

3.2.1 模块描述

此模块是用户身份认证后可以使用的功能模块，通过网页表单传送数据，存在目的是为了查询股票详细信息。

3.2.2 功能

IPO 图如下：



图表-股票查询模块 IPO 图

3.2.3 性能

只要输入股票代码，按键点击，就能转到相应页面查看股票信息。

3.2.4 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|------|--------------|--------|----------|
| 股票代码 | stock_symbol | string | 在文本框手工输入 |
| 查询 | check | Button | 单击按钮 |

图表-股票查询系统输入项列表

3.2.5 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|--------------------|----------------------|--------|------|
| 股票代码 | stock_symbol | string | 文本框 |
| 股票名称 | stock_name | string | 文本框 |
| 股票最新 成交价 | Last_trade_price | float | 文本框 |
| 当前购买 指令最高 价格 | highest_buy_price | float | 文本框 |
| 当前出售 指令最低 价格 | lowest_sell_price | float | 文本框 |
| 当日最高 成交价 | Highest_daily_record | float | 文本框 |
| 当日最低 成交价 | Lowest_daily_record | float | 文本框 |
| 本周最高 成交价 | Highest_week_record | float | 文本框 |
| 本周最低 成交价 | Lowest_week_record | float | 文本框 |
| 本月最高 成交价 | Highest_month_record | float | 文本框 |
| 本月最低 成交价 | Lowest_month_record | float | 文本框 |
| 股票重要 公告 | stock_event | string | 文本框 |

图表-股票查询模块输出项列表

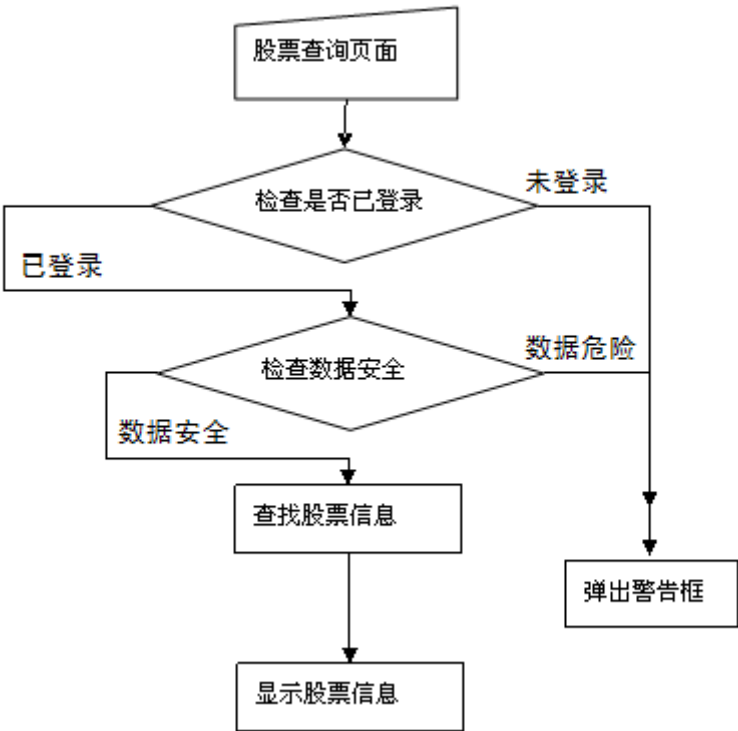
3.2.6 设计方法

当客户端输入股票代码点击查询时，触发脚本函数 SQSend() 将信息提交给服务器，服务器 ClientBackGround 执行如下 php 代码

```
<?
If(session("login"))
{
//如果用户已登陆
    If(CheckStringSafety(form.stock_symbol))
        //如果表单信息安全
        {
            ConnectionDB Db;
            Db.open();
            //创建数据库连接
            $queryset=Db.StockQuery(form.stock_symbol);
            //在数据库中搜索相关股票的相关信息,将结果存入一个包含股票所有详细信
            息域的对象
            Echo $queryset;
            //返回结果
            Db.close();
            //关闭数据库
        }
}
else echo "您尚未登陆"
? >
```

而后客户端的 SQGet() 函数将被调用，将信息写入客户端

3.2.7 流程逻辑



图表-股票查询模块流程图

3.2.8 接口：

内部接口：

本模块作为一个较大的功能模块，通过内部包含其他小模块实现其功能。

用户接口：

你好，游客

账户登陆

修改密码

股票购买

股票出售

撤销指令

资金账户查询

股票查询

股票代码：

查询

股票重要公告：

股票代码：

CA00001

股票名称：

股票名称1

最新成交价：

100

当前购买最高价：

100

当前出售最低价：

100

当日最高成交价：

100

当日最低成交价：

100

当周最高成交价：

100

当周最低成交价：

100

当月最高成交价：

100

当月最低成交价：

100

图表-股票查询模块系统用户接口

3.2.9 测试计划：

与用户登录模块集成测试：

测试要求通过远程的计算机，创建相应证券账户/股票信息数据库，并且添加测试数据。

| | |
|--------------------|----------|
| 输入数据 | 预期结果 |
| 数据库中存在的股票代码 | 返回相关股票信息 |
| 数据库中不存在的股票代码 | 无返回 |
| 数据中包含 sql 语句或特殊字符 | 弹出警告 |
| 用户尚未登陆，绕过登陆窗口访问此模块 | 弹出警告 |

图表-股票查询模块测试计划列表

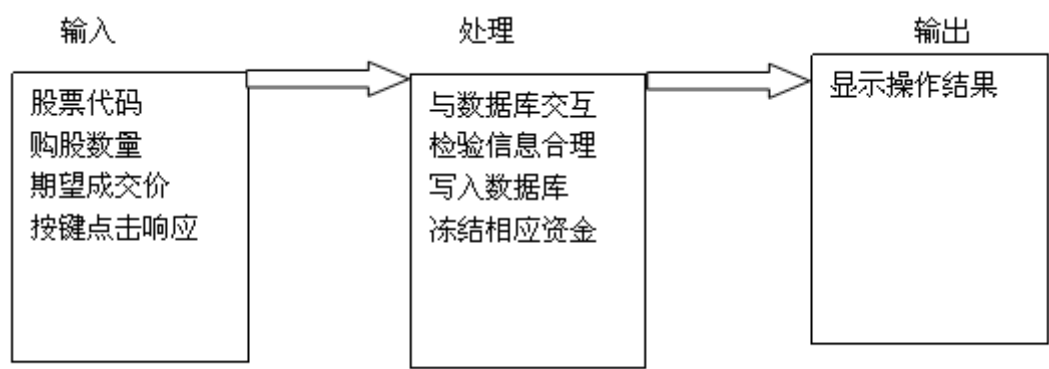
3.3 股票购买模块设计说明

3.3.1 模块描述

此模块是用户身份认证后可以使用的功能模块，通过网页表单传送数据，存在目的是为了发送购买股票指令到中央交易系统。

3.3.2 功能

IPO 图如下：



图表-股票购买模块 IPO 图

3.3.3 性能

表单输入股票代码，购股数量，期望成交价，按键后就能将指令提交到中央交易系统等待撮合。

3.3.4 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|-------|--------------|--------|----------|
| 股票代码 | stock_symbol | string | 在文本框手工输入 |
| 购股数量 | amount | int | 在文本框手工输入 |
| 期望成交价 | price | float | 在文本框手工输入 |

| | | | |
|----|-----|--------|------|
| 购买 | buy | button | 单击按钮 |
|----|-----|--------|------|

图表-股票购买模块输入项列表

3.3.5 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|--------------|--|--------|
| 操作结果 | Instru_Store | Enum(修改成功，资金不足，尚未登陆，价格在涨跌停限外，该股票不存在，该股票不能交易) | msgbox |

图表-股票购买模块输出项列表

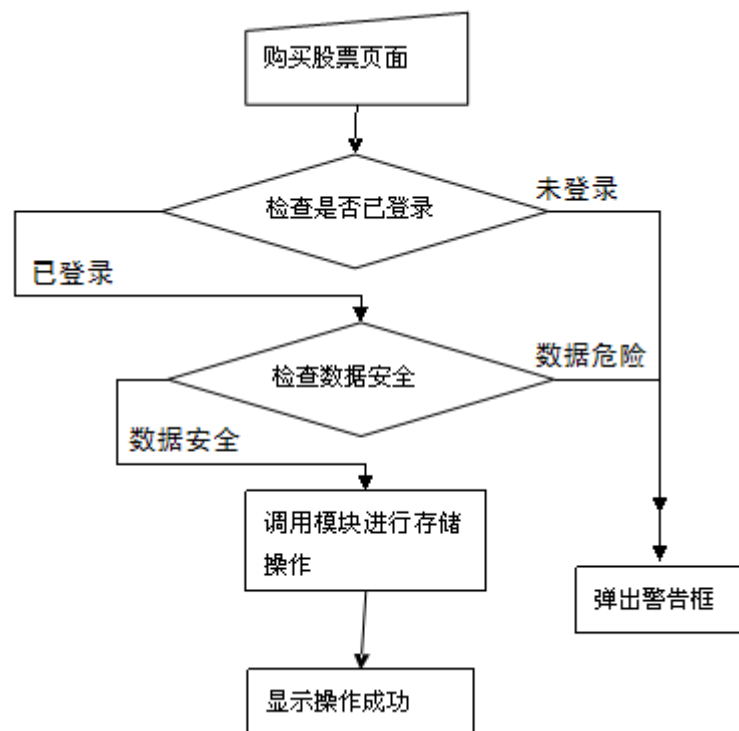
3.3.6 设计方法

客户端点击购买按键后，触发 BSSend() 函数将信息提交到服务器，服务器 ClientBackGround 执行以下代码

```
<?
If(session("login"))
//如果用户已登陆
{
    If(CheckStringSafety(form.stock_symbol)
    &&CheckStringSafety(form.amount)
    &&CheckStringSafety(form.price)
    )
    //如果表单中信息均安全
    {
        ConnectionDB Db;
        Db.open();
        Switch(Db.BuyInstruSave(session("fund_id"),form.stock_symbol,form.a
        mount,form.price))
        //如果账户中可用资金大于当前预购的股票价格与购买数量的积，并且交易价
        格在股票涨停跌限之间，交易成功
```

```
Case 1:  
    echo"修改成功";  
Case 2:  
    echo"可用资金不足";  
Case 3:  
    echo"股票不存在";  
Case 4:  
    echo"交易价格在涨跌停限外"  
Case 5:  
    echo"该股票尚不能交易"  
Db.close();  
}  
}  
echo" 用户尚未登陆";  
?>  
而后客户端通过 BSGet() 函数将服务器返回信息写入客户端
```

3.3.7 流程逻辑



图表-股票购买模块流程图

3.3.8 接口

内部接口：

本模块作为一个较大的功能模块，通过内部包含其他小模块实现其功能。

用户接口：

你好，游客

账户登陆

修改密码

股票购买

股票出售

股票代码：

期望成交价：

期望成交量：

购买

图表-股票购买模块用户接口

3.3.9 测试计划

与用户登录模块集成测试：

测试要求通过远程的计算机，创建相应证券账户/资金账户/股票信息/买指令数据库，并且添加测试数据。

| | |
|--------------------|---------------|
| 输入数据 | 预期结果 |
| 合理的购买指令 | Msgbox (购买成功) |
| 资金不足的购买指令 | Msgbox (资金不足) |
| 数据中包含 sql 语句或特殊字符 | 返回警告 |
| 用户尚未登陆，绕过登陆窗口访问此模块 | Msgbox (尚未登陆) |
| 交易价格在涨停跌限之外 | Msgbox (交易失败) |
| 不存在的股票代码 | Msgbox (交易失败) |

图表-股票购买模块测试计划列表

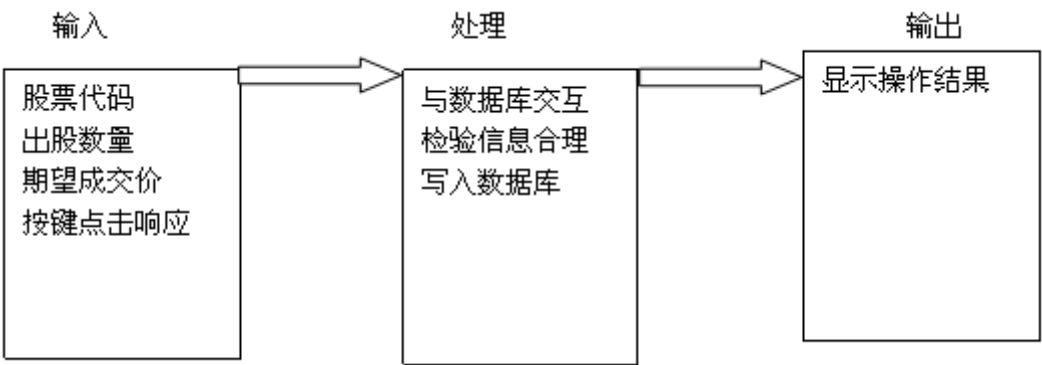
3.4 股票出售模块设计说明

3.4.1 模块描述

此模块是用户身份认证后可以使用的功能模块，通过网页表单传送数据，存在目的是为了发送出售股票指令到中央交易系统。

3.4.2 功能

IPO 图如下：



图表-股票出售模块 IPO 图

3.4.3 性能

按要求输入股票代码，出股数量，期望成交价，点击按键响应便可提交卖指令。

3.4.4 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|-------|--------------|--------|----------|
| 股票代码 | stock_symbol | string | 在文本框手工输入 |
| 出股数量 | amount | int | 在文本框手工输入 |
| 期望成交价 | price | float | 在文本框手工输入 |
| 出售 | buy | button | 单击按钮 |

图表-股票出售模块输入项列表

3.4.5 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|--------------|---|--------|
| 操作结果 | Instru_Store | Enum(修改成功,持股数不足,尚未登陆,价格在涨跌停限外,该股票不存在,该股票不能交易) | msgbox |

图表-股票出售模块输出项列表

3.4.6 设计方法

客户端输完股票代码后，光标离开此文本区域，将触发脚本函数 RmdPriceSend(),从而使服务器 ClientBackGround 执行如下代码

```
:
<?
    If(session("login"))
        //如果用户已登陆
        {
            if(CheckStringSafety(form.stock_symbol))
                //如果信息安全
                {
                    $Db=new ConnectionDB();

                    echo
                    $Db.RemindPrice(form.stock_symbol)
                    //返回该股票的提醒价格
                }
        }
?>
```

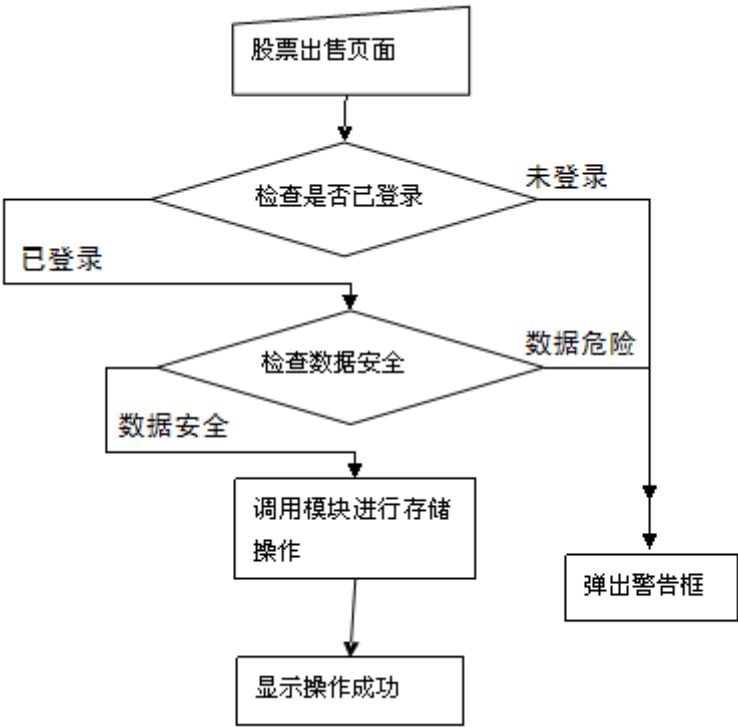
而后客户端执行 RmdPriceGet()将返回信息写入客户端
客户端点击出售按钮后，触发 SSSend（）函数将信息提交给服务器，服务器端 ClientBackGround 运行如下代码：

```
<?
    If(session("login"))
```

```
//如果用户已登陆
{
    If(CheckStringSafety(form.stock_symbol)
    &&CheckStringSafety(form.amount)
    &&CheckStringSafety(form.price)
    )
    //如果表单中信息均安全
    {
        ConnectionDB Db;
        Db.open();
        switch(Db.SellInstruSave(session("fund_id"),form.stock_symbol,form.
amount,form.price))
        //如果账户中持股数足够,并且交易价格在涨停跌限之间,存储指令成功
        Case 1:
            echo"修改成功";
        Case 2:
            echo"持股不足";
        Case 3:
            echo "股票不存在";
        Case 4:
            echo"交易价格在涨跌停限外"
        Case 5:
            echo"该股票尚不能交易"
    }
}
echo" 用户尚未登陆";
?>
```

而后客户端将调用 SSGet() 函数将服务器返回结果写入客户端

3.4.7 流程逻辑



图表-股票出售模块流程图

3.4.8 接口

内部接口：

本模块作为一个较大的功能模块，通过内部包含其他小模块实现其功能。

用户接口：

你好，游客

账户登陆

修改密码

股票购买

股票出售

撤销

股票代码：

期望成交价：

期望成交量：

出售

图表-股票出售模块用户接口

3.4.9 测试计划

与用户登录模块集成测试：

测试要求通过远程的计算机，创建相应证券账户/账户持股/股票信息/卖指令数据库，并且添加测试数据。

| 输入数据 | 预期结果 |
|--------------------|---------------|
| 合理的出售指令 | Msgbox (修改成功) |
| 持股数量不足的出售指令 | Msgbox (持股不足) |
| 数据中包含 sql 语句或特殊字符 | 提示警告 |
| 用户尚未登陆，绕过登陆窗口访问此模块 | Msgbox (尚未登陆) |
| 出售价格在涨停跌限之外的指令 | Msgbox (交易失败) |
| 不存在的股票代码 | Msgbox (交易失败) |

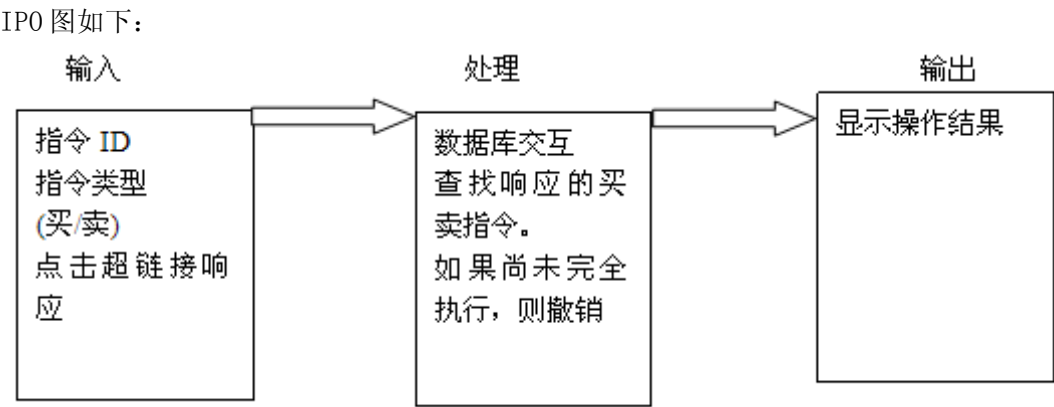
图表-股票出售模块测试计划列表

3.5 买卖指令撤销模块设计说明

3.5.1 模块描述

此模块是用户身份认证后可以使用的功能模块，通过 url 参数传送数据，存在目的是为了发送撤销买卖指令的请求到中央交易系统。

3.5.2 功能



图表-撤销指令模块 IPO 图

3.5.3 性能

页面将会列出一个买卖指令的列表，每条指令都是超链接，用户点击便可转到相关页面进行撤销操作。

3.5.4 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|---------|-------------|-----------------|--------|
| 买卖指令 ID | Instru_ID | int | Url 参数 |
| 买卖指令类型 | Instru_Type | Enum{buy, sell} | Url 参数 |
| 撤销 | Cancel | url | 单击超链接 |

图表-撤销指令模块输入项

3.5.5 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|--------------|-------------------------|--------|
| 操作结果 | Cancelresult | Enum(修改成功，指令已完全执行，尚未登陆) | msgbox |

图表-撤销指令模块输出项

3.5.6 设计方法

客户端需要显示一个等待交易的指令的列表，而这个列表必须从数据库中获取因此在点击此功能后，客户端调用 CIGet ()刷新此页时，需要服务器 ClientBackGround 执行以下代码

```
<?
    If (session("login"))
    {
        ConnectinDB Db;
```

```

        Db.open();
        Queryset=Db.BInstru(session("fund_id"));
        //从指令数据库中取出尚未完全执行的指令集
        echo Queryset+撤销按键+撤销脚本
        //返回等待交易的指令列表
        Db.close();
    }
    else
        echo"您尚未登陆";
? >

```

而后客户端调用 CISend() 将信息写入客户端

用户通过生成的指令列表选择点击相应指令的撤销按键, 调用生成列表时生成的脚本函数转到服务器 ClientBackGround 并运行如下代码:

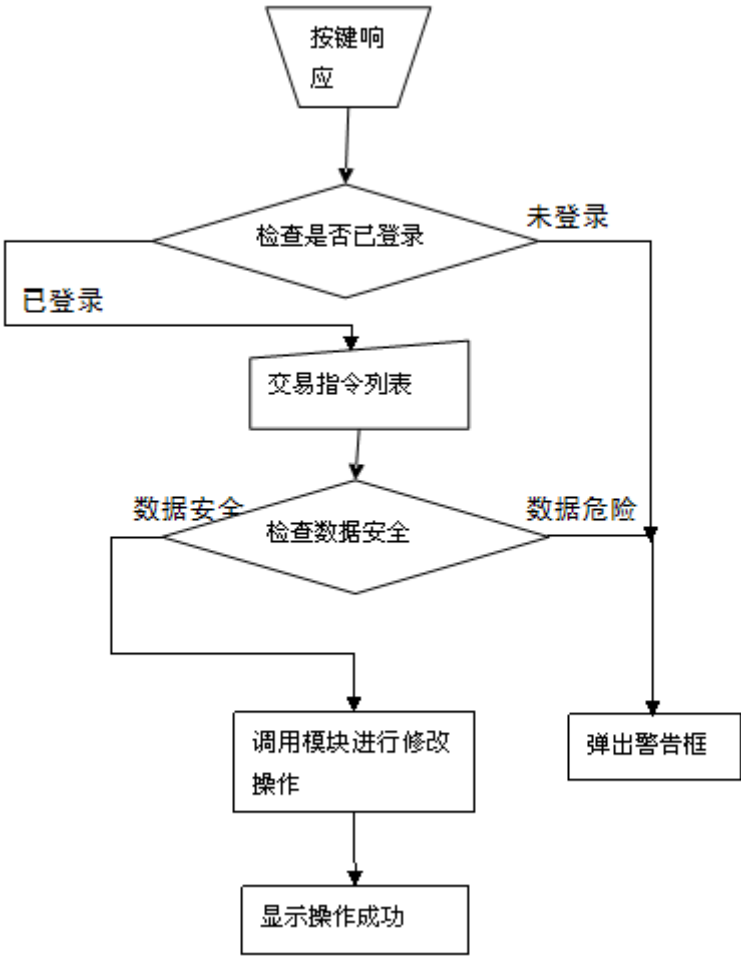
```

<?
    If(session("login"))
    {
        $instru_type=GET("Instru_Type");
        $instru_ID=GET("Instru_ID");
        //获取 URL 参数
        If(CheckStringSafety($instru_type)
        &&CheckStringSafety($instru_ID))//如果参数安全
        {
            ConnectionDB Db;
            Db.open();
            If(Db.CancelInstru($instru_type,$instru_ID))
            //如果指令撤销成功
                echo"撤销成功";
            Else
                echo"指令已执行";
        }
    }
? >

```

而后客户端调用生成列表时生成的客户端脚本, 将信息返回客户端

3.5.7 流程逻辑



图表-撤销指令模块流程图

3.5.8 接口

内部接口：

本模块作为一个较大的功能模块，通过内部包含其他小模块实现其功能。

用户接口：



图表-撤销指令模块超链接列表

3.5.9 测试计划

与用户登录模块集成测试：

测试要求通过远程的计算机，创建相应证券账户/买/卖指令数据库，并且添加测试数据。

| | |
|--------------------|-------|
| 输入数据 | 预期结果 |
| 尚未被执行的指令 | 撤销成功 |
| 完全被执行的指令 | 撤销失败 |
| 部分被执行的指令 | 撤销成功 |
| 数据中包含 sql 语句或特殊字符 | 返回空页面 |
| 用户尚未登陆，绕过登陆窗口访问此模块 | 尚未登陆 |

图表-撤销指令模块测试计划

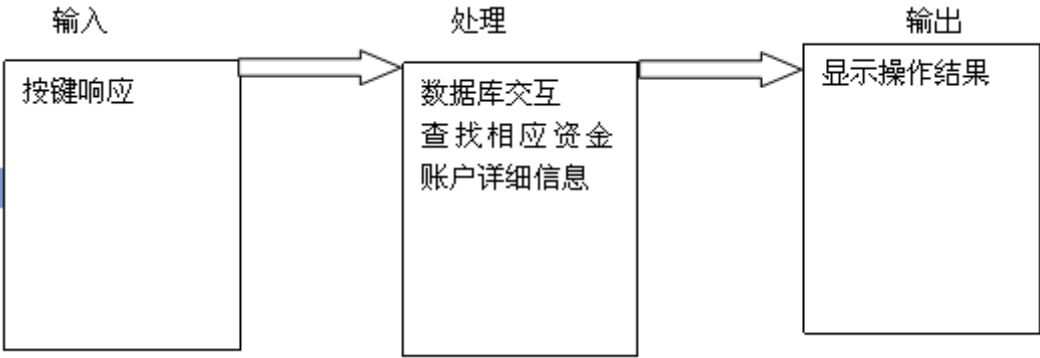
3.6 资金账户查询模块设计说明

3.6.1 模块描述

本模块是用户身份认证后可使用的模块,通过点击按钮触发响应便能打开显示详细信息的页面

3.6.2 功能

IPO 图如下：



图表-资金账户模块 IPO 图

3.6.3 性能

客户端在主界面设计时，会提供一个导航，用户只要点击导航中的资金账户查询标签就能使用此功能

3.6.4 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|--------|-----------------|---------|--------|
| 资金账户 | Fund_id | session | Url 参数 |
| 资金账户查询 | Guide_Fundsinfo | url | 鼠标点击 |

图表-资金账户模块输入项列表

3.6.5 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|--------------|-------|------|
| 可用资金 | Fund_avail | float | 文本 |
| 冻结资金 | Fund_unavail | float | 文本 |

图表-资金账户模块输出项列表

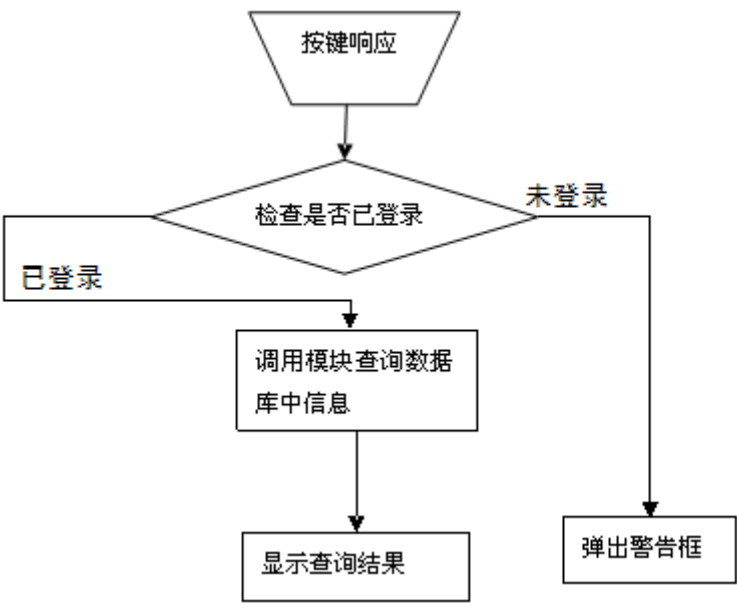
3.6.6 设计方法

客户端点击资金账户查询导航键后，调用脚本函数 FISend() 转到 ClientBackGround 执行如下代码

```
<?
    If(session("login"))
    //如果已登陆
    {
        ConnectionDB Db;
        Db.open();
        Queryset=Db.FundInfo(session("fund_id"));
        //将搜索结果中的可用资金冻结资金存入变量
        Db.close();
        //通过在客户端弹出窗口的方式将信息反馈给用户
        echo"您的可用资金为:"+Queryset.get("fund_available")+"冻结资金为：
        "+Queryset.get("fund_unavailable");
    }
    echo"您尚未登陆";
? >
```

客户端调用 FIGet() 将信息写入客户端

3.6.7 流程逻辑



图表-资金账户模块流程图

3.6.8 接口

内部接口：

本模块作为一个较大的功能模块，通过内部包含其他小模块实现其功能。

用户接口：

系统返回

你好，游客

| | | | | | |
|------|------|------|------|------|--------|
| 账户登陆 | 修改密码 | 股票购买 | 股票出售 | 撤销指令 | 资金账户查询 |
|------|------|------|------|------|--------|

您的可用资金为：

100

您的冻结资金为：

100

图表-资金账户模块系统返回信息

3.6.9 测试计划

与用户登录模块集成测试：

测试要求通过远程的计算机，创建相应证券账户/资金账户数据库，并且添加测试数据。

| | |
|--------------------|---------------------------|
| 输入数据 | 预期结果 |
| 按键响应 | 您的可用资金是：100，冻结资金是： 100 |
| 用户尚未登陆，绕过登陆窗口访问此模块 | 尚未登陆 |

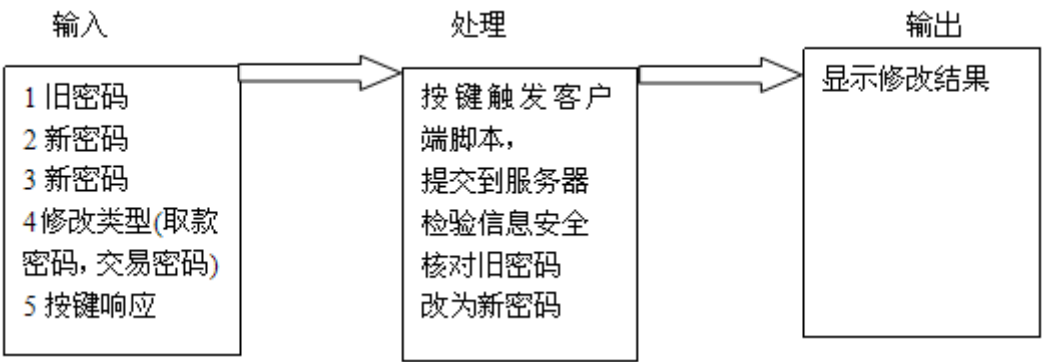
图表-资金账户模块测试计划

3.7 修改密码模块设计说明

3.7.1 模块描述

此模块是用户身份认证后能使用的模块,用户通过表单提交旧密码与新密码给服务器进行修改。

3.7.2 功能



图表-修改密码模块 IPO 图

3.7.3 性能

用户先通过选择修改密码类型（取款密码，交易密码），然后通过同一个表单输入所需填写项目，按键提交。

3.7.4 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|------|---------------|--------------------|-------|
| 旧密码 | O_password | string | 文本框输入 |
| 新密码 | N_password | string | 文本框输入 |
| 新密码 | N_password2 | string | 文本框输入 |
| 修改类型 | Modify_type | Enum {Draw, Login} | 列表选择框 |
| 修改 | Modyfy_button | button | 鼠标点击 |

图表-修改密码模块输入项列表

3.7.5 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------|-----------------|--------------------|--------|
| 修改结果 | Modify_feedback | Enum {修改成功, 旧密码错误} | msgbox |

图表-修改密码模块输入项列表

3.7.6 设计方法

由于在客户端需要进行两次新密码的校对，在用户按键后触发 PMSend() 函数检测两次新密码无误，才提交到 ClientBackGround，服务器执行如下代码

```
<?
If(session("login"))
//如果用户已登陆
{
    If(CheckStringSafety(form.O_password)
    &&CheckStringSafety(form.N_password)
    &&CheckStringSafety(form.Modify_type)
    )//如果表单中数据均安全
    {
        ConnectionDB Db;
        Db.open();
        If(form.Modify_type=="Draw")
        //如果是修改取款密码
```

```

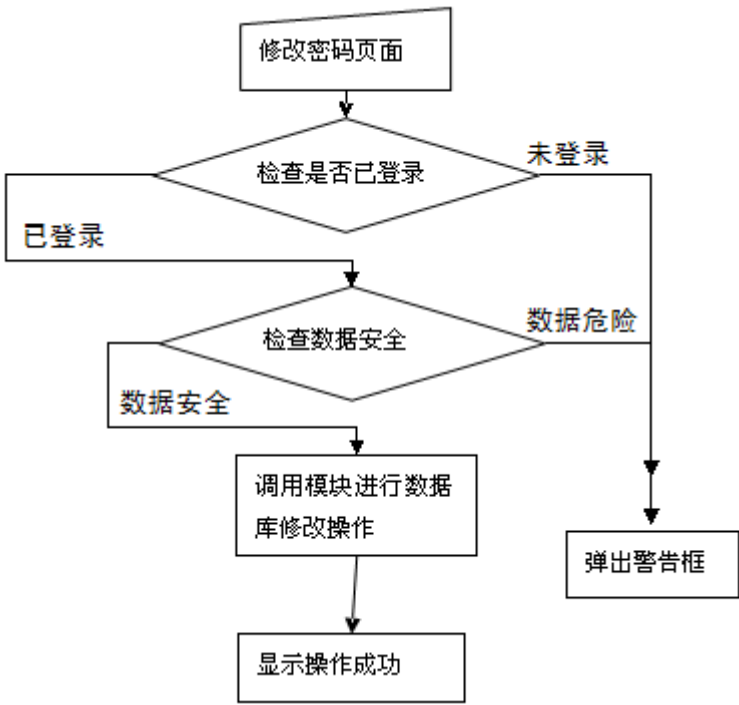
        If(Db. DrawPwdMdfy(session("fund_id"), form. 0_password, form. N_password))
        //如果旧密码核对正确，修改新密码
        echo"密码修改成功";
    Else
        echo"旧密码错误";

Else
    If(Db. LoginPwdMdfy(session("fund_id"), form. 0_password, form. N_password))
    //如果旧密码核对正确，修改新密码
    echo"密码修改成功";
    else
        echo"旧密码错误";

    Db.close();
}
}
echo"您尚未登陆";
? >
而后客户端调用脚本 PMGet()将信息写入客户端

```

3.7.7 流程逻辑



图表-修改密码模块流程图

3.7.8 接口

内部接口：

本模块作为一个较大的功能模块，通过内部包含其他小模块实现其功能。

用户接口：

你好，游客

账户登陆

修改密码

股票购买

股票出售

撤销指令

旧密码：

新密码：

确认新密码：

修改密码类型：

取款密码

取款密码

交易密码

图表-修改密码模块用户接口

3.7.11 测试计划

与用户登录模块集成测试：

测试要求通过远程的计算机，创建相应证券账户/资金账户数据库，并且添加测试数据。

| | |
|--------------------|-----------------|
| 输入数据 | 预期结果 |
| 正确的旧密码，两次新密码 | Msgbox(密码修改成功) |
| 正确的旧密码，两次不同的新密码 | 表单不能提交 |
| 错误的旧密码，两次相同的新密码 | Msgbox(“旧密码错误”) |
| 用户尚未登陆，绕过登陆窗口访问此模块 | Msgbox（您尚未登陆） |

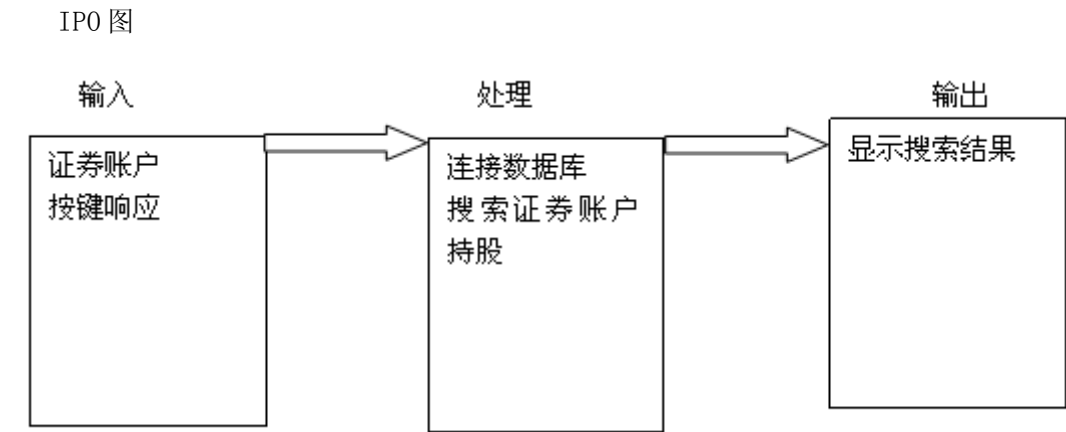
图表-修改密码模块测试计划

3.8 账户持股信息查询设计说明

3.8.1 模块描述

本模块是用户身份验证之后可以使用的模块，模块提供一个页面，显示含有用户持有股票列表，

3.8.2 功能



图表-账户持股模块 IPO 图

3.8.3 性能

通过点击主页面上的导航键，就能显示此模块所对应的页面。

3.8.4 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|------|---------------|---------|---------|
| 证券账户 | Bond_id | session | session |
| 按键 | Guide_MyStock | button | 鼠标点击 |

图表-账户持股模块输入项列表

3.8.5 输出项

| 名称 | 标识 | 类型和格式 | 输出方式 |
|------------|--------------|--------|------|
| 股票名称 | Stock_symbol | string | 文本 |
| 持股总数 | Stock_amount | int | 文本 |
| 股票当前 价格 | Stock_price | float | 文本 |
| 股票持有 成本 | Stock_cost | float | 文本 |
| 持股损益 | Stock_profit | float | 文本 |

图表-账户持股模块输出项列表

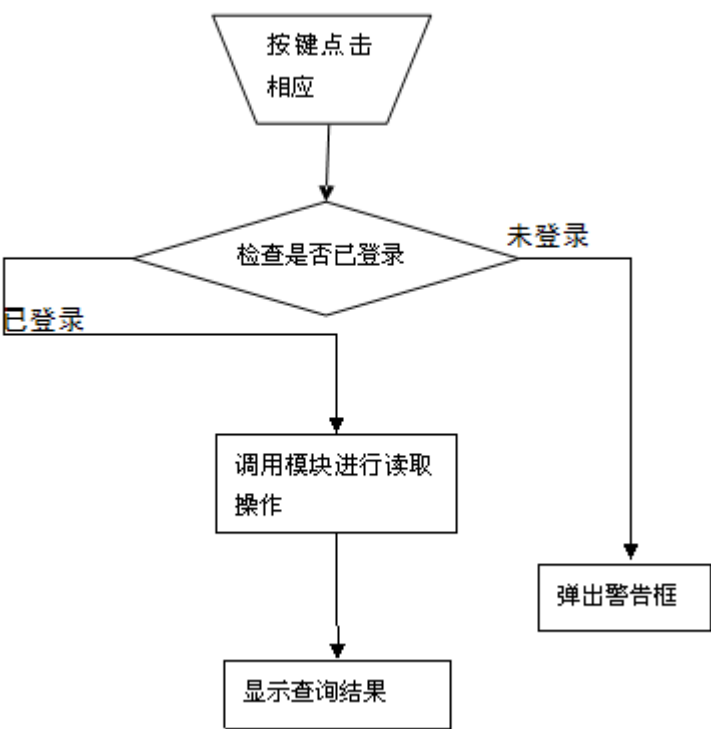
3.8.6 设计方法

用户通过主页面导航点击，触发 MSSend() 函数将信息提交到 ClientBackGround，然后实现如下代码

```
<?
    If(session("login"))
    //如果用户已经登录
    {
        ConnectionDB Db;
        Db.open();
        Queryset=Db.MyStock(session("fund_id"));
        //从数据库获得该账户的持股集合
        echo Queryset;
        Db.close();
    }
%>
```

而后客户端调用 MSGet() 将信息写入客户端

3.8.7 流程逻辑



图表-账户持股模块流程图

3.8.8 接口

内部接口：

本模块作为一个较大的功能模块，通过内部包含其他小模块实现其功能。

用户接口：

你好，游客

| 账户登陆 | 修改密码 | 股票购买 | 股票出售 | 撤销指令 | 资金账户查询 | 股票查询 | 持股信息查询 |
|----------|-------|-------|-------|--------|--------|------|--------|
| 股票代码 | 股票名称 | 当前交易额 | 当前持股数 | 股票持有成本 | 持股损益 | | |
| CA000001 | 股票名称1 | 100 | 100 | 100 | 100 | 出售 | 购买 |
| CA000002 | 股票名称2 | 200 | 300 | 400 | 500 | 出售 | 购买 |
| CA000003 | 股票名称3 | 400 | 500 | 600 | 700 | 出售 | 购买 |

图表-账户持股模块用户接口系统返回列表

3.8.11 测试计划

与用户登录模块集成测试：

测试要求通过远程的计算机，创建相应证券账户/交易记录/证券账户持股/股票信息数据库，并且添加测试数据。

| | |
|--------------------|--------|
| 输入数据 | 预期结果 |
| 按钮响应 | 证券持股列表 |
| 用户尚未登陆，绕过登陆窗口访问此模块 | 您尚未登陆 |

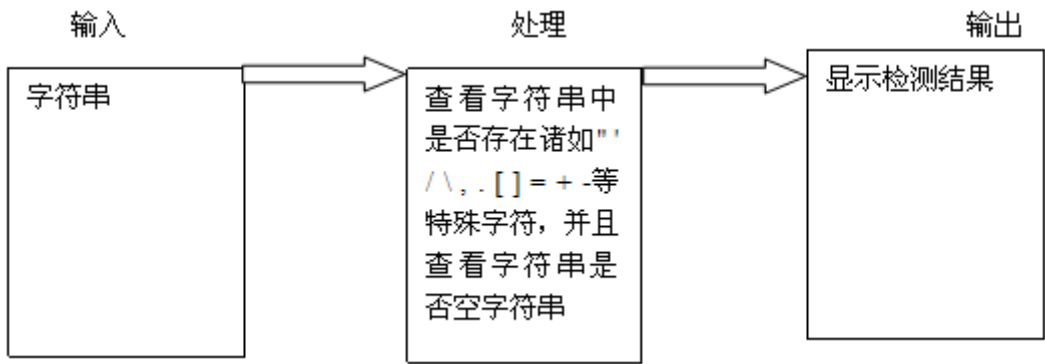
图表-账户持股模块测试计划

3.9 数据安全性检测模块设计说明

3.9.1 模块描述

本模块是用于对客户端传输的数据做安全性检测的模块，之所以做安全性检测，是为了防止一些用户恶意使用 SQL 注入的方式获取或破坏数据库信息。

3.9.2 功能



图表-数据安全性检测模块 IPO 图

3.9.3 输入项

| 名称 | 标识 | 类型和格式 | 输入方式 |
|--------|----------|--------|------|
| 待检测字符串 | StrCheck | string | 函数参数 |

图表-数据安全性检测模块输入项列表

3.9.4 输出项

| | | | |
|------|-------------|-------|------|
| 名称 | 标识 | 类型和格式 | 输出方式 |
| 检测结果 | CheckResult | bool | 函数返回 |

图表-数据安全性检测模块输出项列表

3.9.5 设计方法

```
<?
Function bool CheckStringSafety(string StrCheck)
{
    Char example[]={"','/,\\, ,, . ,[, ], =, +, -,}
    If(none of example in StrCheck)
        Return true;
    Else
        Return false;
}
? >
```

3.9.6 测试计划

与用户登录模块集成测试

测试要求通过远程的计算机，创建相应证券账户数据库，并且添加测试数据。

| | |
|------------|------|
| 输入数据 | 预期结果 |
| 含有禁止字符的字符串 | 返回警告 |
| 正确的账户 | 登录成功 |

图表-数据安全性检测模块测试计划

3.10 同步服务器信息模块

3.10.1 模块描述

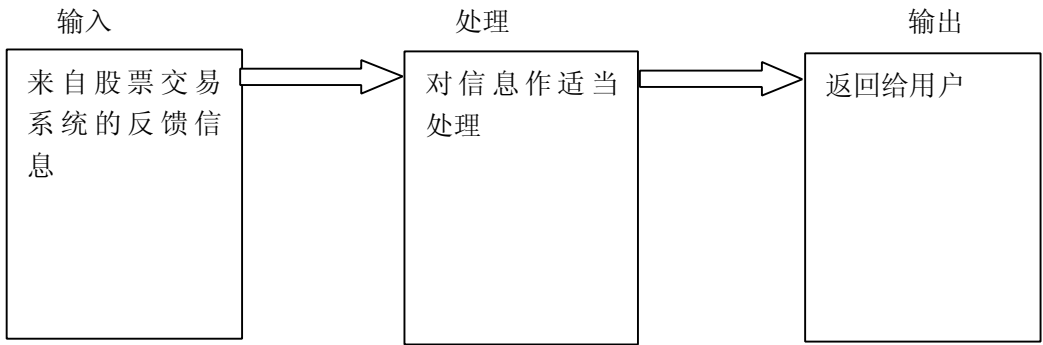
本模块是 V4.0 新添加的模块，主要实现两个功能：显示交易信息，高级提醒功能。

其中,显示交易结果是在中央交易系统撮合指令完成后,向客户端发送信息,客户端于是对用户作出提示。

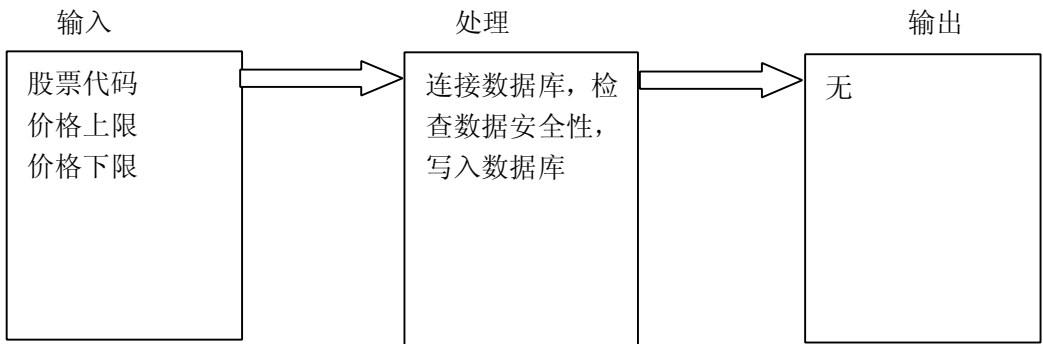
高级提醒功能，是用户可以设置关注股票的代码，价格变动的上下限，从而使系统更好的让用户掌握信息。

3.10.2 功能

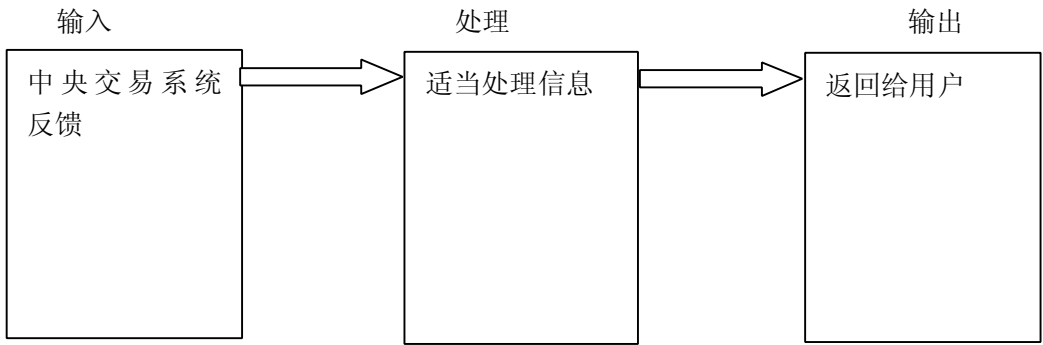
显示交易信息



高级提醒功能



图表-高级提醒功能存储关注股票信息 IPO 图



图表-高级提醒功能返回关注股票信息 IPO 图

3.10.3 输入项

| | | | |
|------|--------------|--------|------|
| 名称 | 标识 | 类型和格式 | 输入方式 |
| 股票代码 | Symbol_Stock | string | 文本框 |
| 价格下限 | Price_llimit | float | 文本框 |
| 价格上限 | Price_hlimit | float | 文本框 |

图表-高级提醒功能存储关注股票信息输入项列表

3.10.4 设计方法

由于中央交易系统与客户端并未建立有效的连接，因此无法主动发送给客户端信息，因此只能通过数据库交互数据。而实时交互的方法主要是：因为中央交易系统每 3 秒钟执行 1 次撮合，即每三秒钟，数据库信息发生改变，交易客户端于是以 3 分钟的间隔访问服务器数据库。查看关心的数据是否已经改变。

客户端通过 window.setInterval(TimingRefresh,180000)的方法对服务器进行 3 分钟一次的访问，而每 3 分钟 ClientBackGround 将执行一次

```
<?
    If(session("time")-time(>3min)//如果刷新时间大于 3 分钟，否则判定为恶意刷性
    {
        Session("time")=time();
        $conn=new ConnectionDB;//
        echo  $conn..MyComplInstru(session("fund_id"));
        //将以执行的指令发给客户端
    }
?>
```

而后客户端通过 TimingWriting 将信息写入客户端

3.11ConnectionDB 设计

3. 11. 1 配置文件 ClientConfig. php

```
<?
    $database=Stock_Transaction
    $account=root
    $password=root
    $server=localhost
    $port=3306
?>
```

3. 11. 2ConnectionDB 定义:

| | |
|-----|----|
| 函数名 | 功能 |
|-----|----|

| | |
|---|--|
| Open() | 通过 php 内置的 mysql_connect() 函数创建数据库连接, 成功返回 true, 失败返回 false |
| close() | 通过 php 内置的 mysql_close() 函数关闭数据库连接, 成功返回 true, 失败返回 false |
| LoginQuery(id, pwd) | 查询数据库中账户为 id, 密码为 password 的账户是否存在, 如果是, 则将登陆状态存入 session 返回 true, 否则返回 false. |
| bool DrawPwdMdfy(id, oldpwd, newpwd) | 查询资金账户为 fund_id 的账户信息, 如果取款密码为 oldpwd, 则将取款密码修改为 newpwd, 并返回 true, 否则返回 false. |
| bool LoginPwdMdfy(id, oldpwd, newpwd) | 查询资金账户为 fund_id 的账户信息, 如果登陆密码为 oldpwd, 则将取款密码修改为 newpwd, 并返回 true, 否则返回 false. |
| BuyInstruSave(symbol_stock, id, amount, price) | 查询数据库中资金账户为 fund_id 的资金账户中可用资金是否大于 amount*price, 如果是, 则判断价格是否在股票代码为 stock_symbol 的涨停跌限之间, 如果是, 则将此购买指令存入数据库, 并修改资金账户中的冻结资金与可用资金。 |
| SellInstruSave(symbol_stock, id, amount, price) | 查询数据库中资金账户为 fund_id, 股票代码为 stock_symbol 所对应的持股数量是否大于 amount, 如果是, 则判断 price 是否在股票代码为 stock_symbol 的股票涨停跌限之间, 如果是, 则将卖指令写入数据库, 并且返回 true, 否则返回 false. |
| BInstru(id) | 查询数据库中资金账户为 fund_id 的对应的股票购买指令。并且返回结果集的句柄 |
| SInstru(id) | 查询数据库中资金账户为 fund_id 的对应的股票出售指令。并且返回结果集的句柄 |
| CancelBI(id, Bid) | 通过索引 Instru_ID 查找指令集中的指令, 如果指令已经被完全执行, 则返回 false, 如果指令尚未执行完毕, 则撤销指令, 并返回 true |
| CancelSI(id, Sid) | 通过索引 Instru_ID 查找指令集中的指令, 如果指令已经被完全执行, 则返回 false, 如果指令尚未执行完毕, 则撤销指令, 并返回 true |
| FundInfor(id) | 返回资金账户为 fund_id 的对应的冻结资金与可用资金。 |
| StockQuery(symbol_stock) | 查询数据库中股票代码为 symbol_stock 的股票信息, 并返回。 |
| MyStock(id) | 返回账户为 id 的对应的持有股票集合句柄 |
| MyCompInstru(id) | 返回资金账户 id 的反馈数据库中的相关账户已经执行完毕的买卖指令 ID 和买卖指令的详细信息, 并且删除此指令, 以此保证下次不会遍历到此指令 |
| RemindPrice(symbol_stock) | 通过股票代码 symbol_stock 返回当前股票当前的建议价格 |

4.接口设计

4.1 用户接口（界面设计）

由于采用网页开发，因此，用户只要点击网页上相关按键，在相关需要填写处填写正确的信息，就能使用本子系统提供的各种功能。并且，客户端使用 JQuery 库增加用户的体验指数。

4.1.1 主界面



图表-交易客户端主界面（导航）

4.1.2 账户登陆模块

Logo

资金账户查询

股票查询

持股信息查询

账户登录

×

账户名：

密码：

登陆

图表-用户登录模块（浮动窗口模式）

4.1.3 修改密码模块

你好，游客

账户登陆

修改密码

股票购买

股票出售

撤销指令

旧密码：

新密码：

确认新密码：

修改密码类型：

取款密码

取款密码

交易密码

图表-用户修改密码模块（表单模式）

4.1.4 股票购买模块

你好，游客

账户登陆

修改密码

股票购买

股票出售

股票代码：

期望成交价：

期望成交量：

购买

图表-股票购买模块（表单模式）

4.1.5 股票出售模块

你好，游客

账户登陆

修改密码

股票购买

股票出售

撤销

股票代码：

期望成交价：

期望成交量：

出售

图表-股票出售模块（表单模式）

4.1.6 撤销买卖指令模块



图表-指令撤销模块界面（列表模式）

4.1.7 资金账户查询

你好，游客

账户登陆

修改密码

股票购买

股票出售

撤销指令

资金账户查询

您的可用资金为：

100

您的冻结资金为：

100

图表-资金账户查询模块

4.1.8 股票查询模块

你好，游客

账户登陆

修改密码

股票购买

股票出售

撤销指令

资金账户查询

股票查询

股票代码：

查询

股票重要公告：

股票代码：

CA00001

股票名称：

股票名称1

最新成交价：

100

当前购买最高价：

100

当前出售最低价：

100

当日最高成交价：

100

当日最低成交价：

100

当周最高成交价：

100

当周最低成交价：

100

当月最高成交价：

100

当月最低成交价：

100

图表-股票查询模块界面

4.1.9 持股信息查询模块



图表-持股信息模块界面

4.2 外部接口

通过 php 内置的 mysql_connect () 函数与数据库 mysql 建立连接。

4.3 内部接口

本系统与其他子系统通过共享数据库完成通信。

| | | | | |
|-----|------------------------|---------------|---------------|----------|
| | 账户管理系统 | 股票中央交易系统 | 网上信息发布系统 | 交易系统管理模块 |
| 客户端 | 共享证券账户/ 资金账户数据 库 | 共享买卖指令数据 库 | 共享股票信息数据 库 | 无通信 |

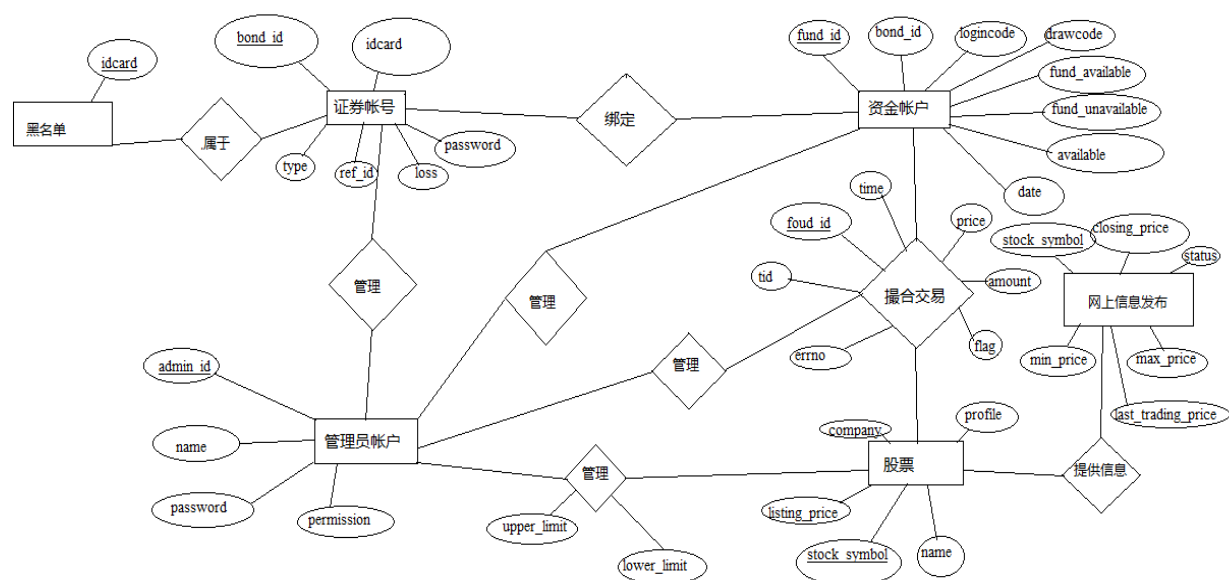
图表-客户端与其他子系统共享数据库表

另外，股票交易系统需要在指令撮合成功后，对指令对应的客户端发送反馈信息。对此，是股票交易系统来实现，还是客户端尚未决定。

5.系统数据结构设计

作为交易客户端，虽然包括服务器后台程序，但是本子系统没有自己设计的数据库，而且因为整个大组对于数据库设计层面上可能时常发生更改，因此，本组在后台处理数据库时，将对数据库的连接，对不同数据库的操作进行封装。从而使本子系统容易适应变化。

5.1 概念结构设计



5.2 逻辑结构设计

管理员账户: stock_admin(admin_id, name, password, permission)

资金账户: stock_account(fund_id, bond_id, logincode, drawcode, fund_available, fund_unavailable, available, date)

证券账户: stock_bond(bond_id, password, type, ref_id, idcard, loss)

个人证券账户:

stock_individual(individual_id, idcard, name, sex, address, job, education, workplace, phone, agent_IDcard, loss)

法人证券账户: stock_legal(legal_id, idcard, agent_id, commercial_id, name, phone, address, executive_name, executive_idcard, executive_phone, executive_address, loss)

身份证黑名单: stock_blacklist(idcard)

股票信息: stock_info(stock_symbol, name, listing_price, company, profile)

证券账户所持有的股票: stock_holder(bond_id, stock_symbol, amount)

股票每日收盘价、最高价等: stock_daily_record(stock_symbol, date, closing_price, max_price, min_price)

交易结果记录： stock_trading_record (tid, stock_symbol , buyer_id , seller_id, price, amount, time)

撮合结果信息： stock_matching_record (foud_id, tid, errno, flag)

当日待撮合买指令列表： stock_buy_order (oid, stock_symbol, fund_id, price, amount, time, flag)

当日待撮合卖指令列表： stock_sell_order (oid, stock_symbol, fund_id, price, amount, time, flag)

股票实时数据： stock_realtime (stock_symbol, upper_limit, lower_limit, last_trading_price, max_price, min_price , status)

5.3 物理结构设计

1、管理员账户

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|------------|------------|------|------|----------|
| admin_id | int (11) | N | Y | 管理员账户 ID |
| name | char (30) | N | N | 管理员姓名 |
| password | char (50) | N | N | 账户密码 |
| permission | tinyint(1) | N | N | 是否授权 |

2、资金账户

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|------------------|---------------|------|------|-------------|
| fund_id | int UNSIGNED | N | Y | 资金账户 ID |
| bond_id | int UNSIGNED | N | N | 所关联的证券账户 ID |
| logincode | char (20) | N | N | 登录用户名 |
| drawcode | char (20) | N | N | 登录密码 |
| fund_available | decimal(16,2) | N | N | 账户可用资金 |
| fund_unavailable | decimal(16,2) | N | N | 账户不可用资金 |
| available | int(1) | N | N | 账户是否可用 |
| date | datetime | N | N | 日期 |

3、证券账户

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|----------|--------------|------|------|-------------|
| bond_id | int UNSIGNED | N | Y | 证券账户 ID |
| password | char (50) | N | N | 密码 |
| type | int (11) | N | N | 用户类型 |
| ref_id | int(11) | N | N | 证券账户资料表的 ID |
| idcard | char (18) | N | N | 身份证号 |
| loss | tinyint(1) | N | N | 账户是否可用 |

4、个人证券账户

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|---------------|------------|------|------|-----------|
| individual_id | int(11) | N | Y | 个人证券账户 ID |
| idcard | char(18) | N | N | 身份证号 |
| name | char(30) | N | N | 姓名 |
| sex | tinyint(1) | N | N | 性别 |
| address | char(200) | N | N | 地址 |
| job | char(100) | N | N | 职业 |
| education | tinyint(1) | N | N | 教育水平 |
| workplace | char(100) | N | N | 工作地点 |
| phone | char(11) | N | N | 联系电话 |
| agent_IDcard | char(18) | N | N | 代理人 ID |
| loss | tinyint(4) | N | N | 是否可用 |

5、法人证券账户

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|-------------------|------------|------|------|---------|
| legal_id | int(11) | N | Y | 法人账户 ID |
| idcard | char(18) | N | N | 身份证号 |
| agent_id | char(10) | N | N | 法人 ID |
| commercial_id | char(10) | N | N | 商用 ID |
| name | char(100) | N | N | 姓名 |
| phone | char(11) | N | N | 联系电话 |
| address | char(200) | N | N | 地址 |
| executive_name | char(30) | N | N | 经理姓名 |
| executive_idcard | char(18) | N | N | 经理身份证 |
| executive_phone | char(11) | N | N | 经理联系电话 |
| executive_address | char(200) | N | N | 经理地址 |
| loss | tinyint(4) | N | N | 是否可用 |

6、身份证黑名单

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|--------|----------|------|------|-----------|
| idcard | char(18) | N | Y | 黑名单中的身份证号 |

7、股票信息

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|---------------|--------------|------|------|------|
| stock_symbol | char(6) | N | Y | 股票代码 |
| name | char(50) | N | N | 股票名 |
| listing_price | decimal(6,2) | N | N | 上市价格 |

| | | | | |
|---------|--------------|---|---|------|
| ce | | | | |
| company | char(50) | N | N | 发行公司 |
| profile | varchar(400) | N | N | 简介 |

8、证券账户所持有的股票

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|--------------|--------------|------|------|---------|
| bond_id | int UNSIGNED | N | N | 证券账户 ID |
| stock_symbol | char(6) | N | N | 股票代码 |
| amount | int UNSIGNED | N | N | 持有量 |

9、股票每日收盘价、最高价等

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|---------------|--------------|------|------|-------|
| stock_symbol | char(6) | N | Y | 股票代码 |
| date | date | N | N | 时间 |
| closing_price | DECIMAL(7,2) | N | N | 收盘价 |
| max_price | DECIMAL(7,2) | N | N | 最高成交价 |
| min_price | DECIMAL(7,2) | N | N | 最低成交价 |

10、交易结果记录

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|--------------|--------------------------|------|------|-------|
| tid | int UNSIGNED | N | Y | 流水号 |
| stock_symbol | char(6) | N | N | 股票代码 |
| buyer_id | int UNSIGNED | N | N | 买方 ID |
| seller_id | int UNSIGNED | N | N | 卖方 ID |
| price | DECIMAL(7, 2) UNSIGNED | N | N | 成交价格 |
| amount | int UNSIGNED | N | N | 成交量 |
| time | TIMESTAMP | N | N | 交易时间 |

11、撮合结果信息

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|---------|--------------|------|------|------|
| fund_id | int UNSIGNED | N | Y | 资金账户 |
| tid | int UNSIGNED | N | N | 流水号 |
| errno | int | N | N | 撮合结果 |
| flag | char(1) | N | N | 状态 |

12、当日待撮合买指令列表

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|--------------|-----------------|------|------|---------|
| oid | int UNSIGNED | N | Y | 流水号 |
| stock_symbol | char(6) | N | N | 股票代码 |
| fund_id | int UNSIGNED | N | N | 资金账户 ID |
| price | DECIMAL(7, 2) | N | N | 报价 |
| amount | int UNSIGNED | N | N | 购买数量 |
| time | TIMESTAMP | N | N | 时间 |

| | | | | |
|------|-----------|---|---|----|
| flag | char(1) | N | N | 状态 |
|------|-----------|---|---|----|

13、当日待撮合卖指令列表

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|--------------|-----------------|------|------|---------|
| oid | int UNSIGNED | N | Y | 流水号 |
| stock_symbol | char(6) | N | N | 股票代码 |
| fund_id | int UNSIGNED | N | N | 资金账户 ID |
| price | DECIMAL(7, 2) | N | N | 报价 |
| amount | int UNSIGNED | N | N | 卖出数量 |
| time | TIMESTAMP | N | N | 时间 |
| flag | char(1) | N | N | 状态 |

14、股票实时数据

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|--------------------|--|------|------|-------|
| stock_symbol | char(6) | N | Y | 股票代码 |
| upper_limit | DECIMAL(7,2) | N | N | 涨停限制 |
| lower_limit | DECIMAL(7,2) | N | N | 跌停限制 |
| last_trading_price | DECIMAL(7, 2) | N | N | 最新成交价 |
| max_price | DECIMAL(7,2) | N | N | 最高成交价 |
| min_price | DECIMAL(7,2) | N | N | 最低成交价 |
| status | ENUM('normal', 'stop', 'limitup', 'limitdown') | N | N | 状态 |

5.4 本组附加数据库（V4.0 新增）

交易完成反馈表 instru_feedback

| 字段 | 类型 | 是否为空 | 是否主键 | 备注 |
|--------------|---------------------|------|------|---------|
| oid | int UNSIGNED | N | Y | 流水号 |
| stock_symbol | char(6) | N | N | 股票代码 |
| fund_id | int UNSIGNED | N | N | 资金账户 ID |
| price | DECIMAL(7, 2) | N | N | 报价 |
| amount | int UNSIGNED | N | N | 购买数量 |
| time | TIMESTAMP | N | N | 时间 |
| type | ENUM('buy', 'sell') | N | N | 类型/买或卖 |

注：中央交易系统需要在指令撮合后将撮合成功的指令写入此表

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。功能模块之间相互不会共享界面（主界面除外），客户端脚本只共用一个创建XMLHttpRequest 类的方法，后台程序只共享建立数据库连接的方法。

6.2 运行控制

用户通过主页面上的导航选择功能：

○用户登录模块：用户通过在表单中输入正确的账户密码，从而登录系统，获得应有的使用其他功能的权限。

○股票查询模块：用户通过登陆后，获得授权，便能在此表单输入存在的股票代码，查询股票信息。

○股票购买模块：用户通过登陆后，获得授权，便能在此表单输入存在的股票代码，在涨跌停限之间的交易价格，以及合理的交易数量（不会用完可用资金），发送给服务器。

○股票出售模块：用户通过登陆后，获得授权，便能在此表单输入存在的股票代码，在涨跌停限之间的交易价格，以及合理的交易数量（持股数量足够），发送给服务器。

○指令撤销模块：用户通过登陆后，获得授权，便能在此页面通过通过查看本日未执行的交易指令，每条记录后提供撤销的按键，供点击触发撤销事件。因为页面不是实时刷新，所以存在撤销失败的情况。

○资金账户模块：用户通过登陆后，获得授权，便能通过主页面上的导航选择此页面查看详细信息。

○修改密码模块：用户通过登陆后，获得授权，便能在此表单输入正确的旧密码，两次相同的新密码，选择修改密码类型（取款密码，交易密码）。

○持股查询模块：用户通过登陆后，获得授权，便能通过主页面上的导航选择此页面查看详细信息。

6.3 运行时间

由于采用 Ajax 技术，实现客户端静态更新信息，静态分页等等，因此，服务器的带宽利用率大大提高，唯一会影响服务器的运行时间的，应该只是客户端频繁建立销毁数据库连接所带来的时间开销。

7.系统出错处理设计

7.1 出错信息

| 系统输出信息的形式 | 含义 | 处理办法 |
|------------|--|--|
| 数据库连接不上 | 由于并发操作的用户数量很大,导致数据库访问读写效率降低。或者数据库配置不对,导致数据库连接失败。 | 修改数据库配置,并限制同一个用户两次访问数据库之间的时间间断限制。 |
| Sql 语句执行错误 | 由于一些用户恶意在表单中插入 sql 语句,企图破坏数据库,导致访问数据库出错。 | 对表单中的数据在服务器端进行过滤,检测。 |
| 网站信息丢失 | 由于一些用户恶意不通过登陆界面而直接访问网站或者后台,导致网站信息破坏或被窃取。 | 禁止直接访问一些需要登陆的网页。 |
| 服务器崩溃 | 服务器由于长时间连续运作,导致错误积累过多,负荷过大,最终崩溃 | 定期对服务器进行维护 |
| 账户被盗 | 由于客户端被黑客植入木马或者其他方式引起用户账户被盗,密码被修改。 | 建议用户经常对电脑进行杀毒,不要在陌生的地点用其他人的电脑登陆客户端。 |
| 数据库中账户信息泄露 | 黑客通过后台下载服务器中的数据库,盗取所有用户的账户信息。 | 对账户中密码进行加密,如 3 重 DES 加密算法, MD5 加密算法等等。 |

| | | |
|----------------|------------------------------|--------------------------------------|
| 磁盘损坏 | 由于需要频繁地操作数据库,导致磁盘损坏,数据丢失。 | 使用磁带对磁盘进行周期性地备份。 |
| 系统无法处理一些非法交易指令 | 股票代码不存在,购股需要金额远大于资金账户中可用金额。 | 在将指令存入数据库时,需要做指令合理性判断。 |
| 数据库读取乱码 | 在操作数据库过程中,由于写入出错或者读出出错导致出现乱码 | 一般出现乱码的错误为数据库编码与写入数据编码不一致缘故,此时需要统一编码 |

图表-系统出错一览表

7.2 补救措施

a. 后备技术

周期，频繁备份数据库，并且将其储存在更加稳定的介质上，比如磁带。

将系统所需的数据库部署到不同的计算机上，减小因硬件问题而导致数据全部丢失的可能性。

b. 降效技术

紧急情况下，启用人工，代替系统撮合买卖指令。

c. 恢复及再启动技术

系统崩溃后，通过系统运行日志记录恢复数据。

7.3 系统维护设计

连接数据库方面，需要在创建数据库连接，销毁数据库连接，执行 sql 语句的模块使用 try catch 语句捕获异常。于是发生此类错误便能很快得知。

网络方面需要检测是否有特定的 IP 地址频繁访问系统，不能排除其攻击服务器的可能性。

中央交易系统中，需要检测撮合进程是否进入死锁，一些未被处理的指令永久处于饥饿状态。

内部人员如（开户人员，系统维护人员）操作留下操作痕迹，使用权管理层可以定期或不定期地稽核系统。