

分类号 _____
学校代码 10487

学号 M201676065
密级 _____

华中科技大学

硕士学位论文

基于 B/S 结构的实盘训练营系统 的设计与实现

学位申请人：刘喆

学科专业：软件工程

指导教师：吴涛 教授

答辩日期：2018.12.25

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree for the Master of Engineering**

**Design and Implementation Of Shi Pan
Training Camp System Based On B/S
Structure**

Candidate : Liu Zhe

Major : Software Engineering

Supervisor: Assoc. Prof. Wu Tao

Huazhong University of Science and Technology

Wuhan 430074, P.R.China December, 2018

摘要

随着前端技术日新月异的发展，前端页面不仅仅再是承担提供信息的功能，更承载着人机交互的重要作用。许多以往的平台型产品，渐渐 Web 化，变成轻量级交互设计，更注重简洁而方便的用户体验，同时还保证了信息资源的利用效率。股票实盘训练营便是这样一个基于 B/S 结构的股票模拟交易系统，为股票投资交易新手用户提供的学习、练习股票交易的平台。用户可以在如同真实的股市环境中，完成买入、卖出等操作。

为顺应时代的潮流，实盘训练营主要采用了基于 Web 端的设计理念，信息接收，交互、应答均在前端完成。股票实盘训练营系统基于 JavaScript 实现了前端交互逻辑，凭借 HTML 语言展现内容，利用 CSS 控制版式页面。该平台恪守着最新的结构规范，按照 HTML DOM 规则控制页面结构。同时该平台基于 bootstrap 样式模板进行构建，既保证了页面的美观整洁又实现了响应式布局，在不同平台不同浏览器下均可保证正常使用。模块内部采用 MVC 模式，保持展示结构与功能逻辑结构的分离，避免了组件之间的耦合，且保证了在大量用户同时使用时的稳定性、安全性、敏捷性。在功能、测试和监控方面，很多程序都加入了持续集成，遵循这样的原则让我们写的程序也越来越健壮。在模拟交易系统的构建过程中同样注意了对于代码模块与组件的复用，依循模块化开发的准则进行架构，利于后期的版本迭代与维护。

股票交易系统会从需求分析、系统设计、系统实现、系统测试四个主要方面进行剖析论述，基于最新 Web 端前沿技术标准，通过总结前端模块框架的共同之处，帮助功能实现分层架构，有效减少大量复杂且冗余的低质量代码，构建高质量的、重体验、重稳定、重迭代性的股票模拟交易系统。目前该系统已上线并保持了良好的运营状态，日活跃用户过万人，收获了大量用户的好评。

关键词：股票交易 前端架构 响应式

Abstract

With the rapid development of front-end technology, the front-end page not only provides information, but also plays an important role in human-computer interaction. Many of the previous platform-based products have gradually become Web-based and use lightweight interactive design, paying more attention to the simple and convenient user experience, while also ensuring the utilization efficiency of information resources. The Stock Training Camp is such a B/S construction trading system. Users can buy and sell stocks like in a real stock market environment.

In order to comply with the trend of the times, the system mainly adopts the design concept based on the Web, and information reception, interaction and response are completed at the front end. The Stock Training Camp implements front-end interactive logic based on JavaScript, displaying content in HTML language, and controlling layout pages using CSS. The platform adheres to the latest structural specifications and controls the page structure in accordance with HTML DOM rules. At the same time, the platform is built based on the bootstrap style template, which not only ensures the beautiful and clean page, but also realizes the responsive layout. It can be used normally under different browsers of different platforms. The MVC mode is adopted inside the module to maintain the separation of the display structure and the functional logic structure, avoiding the coupling between the components, and ensuring the stability, security and agility when a large number of users are simultaneously used. In terms of functionality, testing, and monitoring, many programs have joined continuous integration, and following this principle, the programs we write are becoming more robust. In the construction process of the simulation trading system, the reuse of code modules and components is also paid attention to, and the architecture is developed according to the criteria of modular development, which is beneficial to the later version iteration and maintenance.

华 中 科 技 大 学 硕 士 学 位 论 文

The stock trading system will analyze the four main aspects of demand, system design, system implementation and system testing. Based on the latest front-end technology standards of the Web, the functional layer will be implemented by summarizing the commonalities of the front-end module framework. We Effectively reduce a large number of complex and redundant low quality codes and finally made it out a high-quality, experienced, stable, and iterative stock simulation trading system. At present, the system has been used on line. It has maintained a good operating status. The number of daily active users is nearly ten thousand, which has won a lot of praises from a large number of users.

Key words: Stock Trading Front-end Architecture Responsive

目 录

摘 要.....	I
Abstract.....	II
1 绪论	
1.1 研究背景与意义	(1)
1.2 国内外研究现状	(2)
1.3 相关技术介绍	(5)
1.4 主要研究内容	(6)
2 系统需求分析	
2.1 系统功能性需求分析	(8)
2.2 系统非功能性需求分析	(12)
2.3 本章小结.....	(14)
3 系统设计	
3.1 系统架构设计	(15)
3.2 系统功能设计	(17)
3.3 系统数据库设计	(23)
3.4 系统性能设计	(28)
3.5 本章小结.....	(29)
4 系统实现	
4.1 开发环境及服务器部署	(30)
4.2 系统功能实现	(31)
4.3 性能优化实现	(41)
4.4 本章小结.....	(42)

5 系统测试

5.1 系统测试环境	(44)
5.2 系统功能测试	(44)
5.3 非功能性测试	(47)
5.4 本章小结.....	(49)

6 总结与展望

6.1 全文总结.....	(50)
6.2 展望.....	(50)

致 谢.....	(52)
----------	------

参考文献.....	(53)
-----------	------

1 绪论

1.1 研究背景与意义

当前股票交易平台众多，为拉拢股票交易的新手用户，各大股票交易平台均开发了模拟交易功能，为股市初学者提供学习、尝试、交流股票交易的学习的平台。股票实盘训练营为注册用户提供免费初始虚拟金用来购入股票，股票的数量、名称、涨幅跌幅均与市场真实股市保持一致，交易时间与股票实际交易时间保持一致，交易品种为在深圳、上海证券交易所上市的 A 股。为用户带来了趋于真实的股市交易操作体验。近年来凭借良好的用户体验以及高度好评的系统稳定性，在股票投资交易学习者中赢得了良好的口碑，迅速积累了数万的活跃用户数量。

为达到实盘训练营的轻量级简洁易用特点，该交易平台前端基于 Bootstrap 框架，JavaScript ES6 语言规范，后台基于 Java 语言进行搭建^[1]。基于前端的股票模拟交易系统的设计免去了股票交易环境的安装，用户只需在浏览器端即可进行访问、操作，并将信息记录在服务器端，既保证了用户的访问速度，也保证了用户数据的安全性^[2]。

随着时代的发展，后台功能向前端的迁移。Web 的历史出现了几个阶段。

Web1.0 时期网站多以单向输出信息为主，用户通过 Web 页面接受相关信息、资讯，但并不可以完成交互，无法向 Web 端传输数据包。此时的 Web 技术亦相当简单^[3]。

Web2.0 的概念大约在 2004 年出现，立刻掀起了互联网的风波。页面不再仅仅是信息的展示栏，用户可以直接在网页上进行交互，他们既是浏览者的身份，同时也是内容的创造者、分发者、分享着。Web2.0 达到了去中心化的效果。这一次升级换代，让互联网的主角不再是少数人，广大互联网用户得以登上信息交流的舞台。

虽然前端技术已出现近 30 年，但前端技术的快速发展与蓬勃期确实最近的十年。伴随着 Web 进入 2.0 时代，人们发现越来越多的应用、可以建立在前端基础上得以实现，既保证了系统的完整性、人与设备的交互性，也保证了安全的数据传输。更

重要的是，基于前端的应用可以免去用户在平台安装复杂臃肿的程序，达到快捷使用的目的。在人人追求效率的互联网时代，前端技术的蓬勃发展完美的契合了人们的使用需求。随着各类前端技术的涌现，如今各类应用、游戏、工具得以在前端面向人群。

正是因为前端技术在近十年来的高速发展，使得前端的技术变得越来越复杂。用户的需求层出不穷，页面的效果愈发精致。如何选用合适的平台、框架、技术来搭建成为了前端架构师们不可逃避的问题^[4]。本文主要研究股票模拟交易系统的前端架构与实现，利用截至目前的最新前端技术与框架，如何搭建出具有可靠性、安全性、易用性、易维护的股票模拟交易系统。

1.2 国内外研究现状

（1）市场上的股票交易平台

在上世纪 90 年代上海证券交易所于深圳证券交易所的成立之后，中国的股民数量不断壮大，同时也涌现了不少的股票交易平台。

用户量最大的是同花顺股票交易平台。主要支持 A 股交易，提供股票模拟盘与 K 线训练营功能供股票交易学习者使用。作为一个经验丰富的股票市场平台，其始终站在潮流的风口浪尖上，凭借其在移动端的功能和布局，它已经赶上了移动互联网最大的热潮。目前该产品在 PC 端、移动端、Web 端均可使用。

东方财富亦是用户量极大的股票交易平台。其特色功能“股吧”为股民提供了调股、论股、诊股的交流空间。随着移动互联网的影响，其活跃程度一直在下降，但经过十多年的积累，东方财富仍然保持着该行业在股市信息和数据领域的第一阵营。

牛股王凭借 2015 年的大牛市，又恰好赶上“互联网+”领域的东风，迅速赢得了海量的用户，吸引了众多炒股学习者的使用。在模仿其他传统股票交易平台的基础上，开发了一些独特的功能，如：自动定量警告，与公牛，一键跟进等。然而，随着牛市的结束，牛股王的活跃用户数量也开始显着下降。

雪球 2011 年发布至今，该平台已有 8 年的积淀。从最初的社交领域开始，它已经发展成为包括上市公司，经纪人，基金和投资者的完整生态系统。近年来，雪球已经成为最成功的投资和交换社区之一。

富途牛牛专注于传统的港美股票交易。由于本身带有互联网的过往成功经验，重型产品经验和技术能力，2015 年香港股市炙手可热时，由于订单，崩溃，停机和交易失败的突然增加，许多香港-美国股票投资交易平台已经出现。富图依靠其雄厚的科技实力，成为唯一能够承受证券交易高峰的互联网经纪公司。其系统在高流量峰值之下的稳定性同样是业界最出色的标杆。

老虎证券成立于 2014 年，凭借海外资产配置的潮流，迅速在香港股市投资领域站稳脚跟。老虎证券发布了各种维度的市场波动列表供用户参考，包括股票和股票，股票交易图表等。许多股票交易新手用户的愿景已经从中国股市扩展到整个美国股市。

Tradestation 是一个专业的股票交易平台，在美国流行了 30 多年，专注于为活跃的交易客户提供服务。拥有低延时、高速时情的优点。支持多窗口，多工作区模式，自定义市场界面，全市场扫描，实时警告，动态排序。

在众多股票交易平台中，最重要的功能是股票市场，股票市场信息，股票分析等。但是，这些平台的主要功能大多相似。因此，它在一定程度上造成了严重的同质化。因此，在目前市场环境不理想的情况下，如何吸引用户并满足用户的其他需求已成为决定股票交易平台企业生存的重要因素。开发自己的创新模式，建立自己的产品功能，在激烈的竞争中不断创新，赢得更多用户的青睐。实盘训练营便是一个针对于新手股票学习者的重大吸引。

（2）前端技术研究现状

如果说近年来前端蓬勃发展的根本原因，应当归结于人们对 JavaScript 语言的运用，它起到了最大的正面影响效果。

JavaScript 出现之初，人们认为它只是一门脚本语言，只能用来做一些旋转样式，或在页面中添加一些具有视觉效果交互。当时，他的主要目的还在于接收反馈服务端脚本语言负责的一些验证表单。当时表单数据需要传送到服务器以后才能确定用户是否填写数据域，JavaScript 则巧妙的解决了在客户端完整验证的问题。直到 Ajax 的出现^[5]，成全了 JavaScript 语言的发展与前端技术的发展。JavaScript 的实现主要包括核心 ECMAScript、文档对象模型（DOM）^[6]、浏览器对象模型（BOM）。

最先出现的 JQuery/Prototype, 前者主要与浏览器兼容, 这样开发人员就不再把精力投入到浏览器的差异中; 后者是语言本身的扩展, 扩展了各种类型的 JavaScript 并提供了一组扩展任何对象的功能。从 Github 上用户使用编程语言的范围、数量的角度来看, JavaScript 已经成为了当今世界上最流行的编程语言。它服务于全球上千万网民的 Web 浏览器之中, 以保证服务并提供 Web 应用程序的交互性。

13 年移动时代的到来推动了前端的响应式发展。在 13 年里, 出现了许多移动终端框架, 如 Sencha Touch/Zepto.js 等, 他们相对于传统的 PC 端框架更加的轻便了, 灵活了。如 Twitter 推出的 CSS 框架 Bootstrap, 它解决了过往难题, 如不同终端自动适应、多分辨率自适应、远程调试自适应等关键问题。它同时方便了前端、后端的开发, 让建站变得更加易如反掌。Node.js 闪耀四方, 许多公司在生产环境中使用 Node.js, 同时也出现了诸如 Express、Meteor 等小巧的快速搭建 Node.js Server 的应用框架^[7]。

随着移动终端 14 年的崛起, HTML5 标准正式登陆, 标志着 Web Page 向 Web Application 的转变。ES6 也成为了前端的主流标准, 它的模块、类等特性让 JavaScript 这门语言变得更为饱满, 具备了开发大型应用的一切能力。Web 跨终端组件的快速壮大发展, 前端开发迎来了新的台阶^[8]。伴随 Node.js 在前端和后端分离的变化, 中间层的出现让过去的传统合作模式得到了改变。在这个颠覆性的一年中, 前端技术开始形成短期稳定的格局^[9]。

15 年后, 前端进入了工业生产的时代。伴随着前端应用的大量出现以及前端程序的复杂性日益提高, 出现了对前端技术的大量需求, 无数的框架涌现而出。JavaScript 框架 React、Vue.js 替代过往的 JQuery 成为了当今前端工程师们的法宝。同时众多构建工具百花齐放, 在 grunt、webpack 等包管理工具之后, npm 开始主导整个市场。为了响应端到端之间的数据传输, 后端更注重数据的可用性和接口包的可扩展性。是由于后端服务的考虑, 前端差距非常明显, 越来越多的基于 Web 的应用程序已经从后端发布, 而前端完全异步获取数据^[10]。

截止至 2018 年, React 继续在前端框架领域占领着领先的地位, 它囊括了可以实现异步 UI 渲染的 fiber 结构。网格布局终于被 CSS 纳入了标准, 各大浏览器厂商也进行了快速的跟进。Vue.js 继续流行并被 Github 在内的大型互联网公司所采用^[12]。

而 JavaScript 的最新版本为 ES8^[13]。

1.3 相关技术介绍

(1) JavaScript

JavaScript 是一种专门为网页交互而设计的语言。它可以通过三个模块进行描述,即为 ECMAScript、文档对象模型与浏览器对象模型。ECMAScript 由 ECMA-262 所定义,是语言核心功能模块。而文档对象模型则是提供网页内容呈现的函数与接口。浏览器对象模型承担提供与浏览器交互操作的接口的作用^[12]。这三个模块全部在主流浏览器中得到了广泛的支持。随着时代的发展,JavaScript 已发展到了 ES8 版本。

JavaScript 是一种函数优先的解释型轻量级语言,同时也是即时编译性语言。虽然它由于在前端开发的重要作用而被广为人知,但当前已被用于各种的非浏览器环境之下,比如为人所熟知的 Node.js^{[11][13]}。它也是一种基于原型编程、多范式的动态语言,具有支持面向对象、命令式、声明式风格的特性。

JavaScript 具有原型链继承的特点。相比于其他基于类的开发语言,它不需要通过类实现,JavaScript 的动态性令他独具一格,因为它是基于原型的。JavaScript 的继承只用考虑到对象类型。每个对象类型的私有属性 Prototype 指向它的原型对象

(2) HTML DOM

DOM 为文档对象模型,承担支撑页面的结构的作用,是 W3C 组织设置的一种拓展语言标准编程接口标准。用户通过脚本语言可以对页面的结构进行编辑,实现节点的插入、删除、改变属性等操作。结构图如图 1-1 所示。

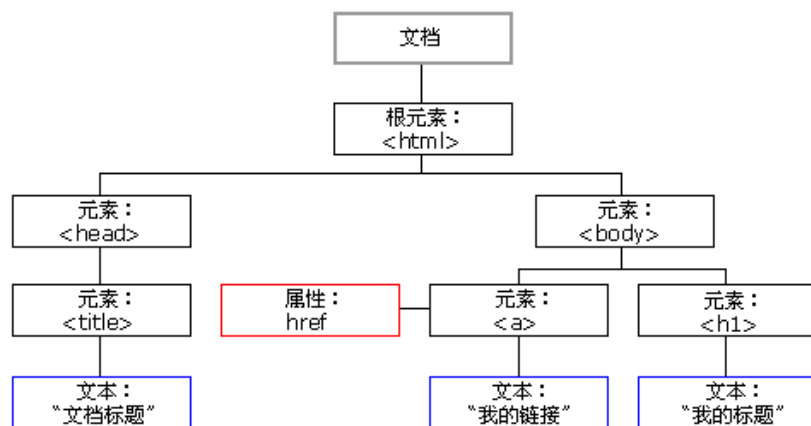


图 1-1 文档对象模型

(3) Canvas

Canvas 是通过脚本完成页面的绘图操作。使得页面可以直接根据代码直接实现图片甚至动画效果。他是 HTML5 新增的特性之一，最初由苹果的 Webkit 推出，供 Safari 浏览器使用。在 Canvas 元素中，元素的绘制依赖于坐标的描点，起点为左上角 (0,0)。所有元素相对于远点进行定位。如图 1-2 所示。

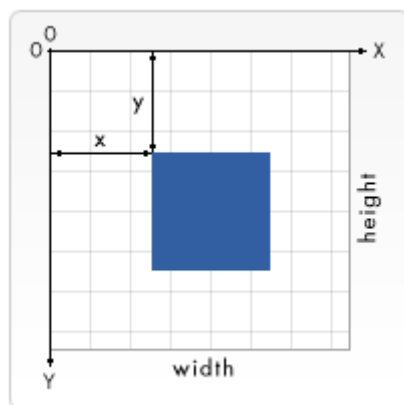


图 1-2 Canvas 坐标系

1.4 主要研究内容

本系统是一个模拟股票交易系统，文章主要论述前端的设计与实现。该系统为小白用户提供股票的买入、卖出、查看自选股，查看资产，自选股讨论等诸多业务

功能。它拥有同 A 股一样真实的行情，并保持实时性。让用户无投资成本无风险的进行股票模拟交易操作，有利于股票交易学习者了解自己的投资水平，总结投资经验。

该系统结合最新的前端技术与标准，基于 Bootstrap 框架进行开发，采用了 MVC 结构^[14]，按照软件工程开发要求进行研究开发，论文的研究内容主要包括以下几个方面：

（1）针对于现有的股票模拟交易系统与前端技术发展局面，研究分析开发该系统的可行性与比较性。

（2）对股票模拟交易系统进行用户需求调研，明确具体的功能性需求与非功能性需求，确定需求的范围。

（3）结合对系统的需求调研，对系统的业务模块进行系统设计，包括功能设计、架构设计、可维护性设计与性能优化设计。并对系统的结构流程与接口进行说明。

（4）根据需求与业务的系统设计对系统前端实现。从内容角度、版式角度、交互角度实现，包括了用户侧、管理员侧的用例以及非功能需求的实现。同时给出了性能及稳定性的优化策略。

（5）对股票实盘训练营平台进行功能性测试、非功能性测试、性能测试、稳定性测试^[15]，验证了该系统在大流量下的稳定性运行。

（6）对论文进行总结，论述该系统设计与实现的优点、意义与后续可优化的展望。

2 系统需求分析

自从 90 年 10 个 A 股上市后，人们对于投资股票的需求变慢慢涌现了出来。随着时代的发展、经济的繁荣，网络的进步，人们投资股票的方式慢慢发生了巨大的转变。

随之而出的便是人民对于股票在网络上交易的需求^{[16][17]}。各大股票交易平台如雨后春笋般破荒而出。当人们想要股票投资的时候，第一时间想的是在互联网上进行交易。

如今虽然几大股票交易平台已经瓜分了大部分市场，但如何拉拢股市交易用户成为了各大互联网金融公司、证券交易公司的战略重点。

实盘训练营是一个针对于股票交易新手用户进行模拟股票交易学习的平台，主要面向无经验、有极少经验、没接触过股市或接触很少的新手用户群体。这类群体代表着未来互联网股票交易市场的新鲜血液，如何抓住这类人群的需求成为了最大的重点。

本章主要分析实盘训练营的如账户管理、浏览行情、管理资金账户、买入卖出、浏览咨询、盈利兑换京豆等业务性功能需求，以满足此类需求来抓住股票新手的痛点^{[18][19]}。

同时也分析了性能需求、可维护性需求、可拓展性需求等非功能性需求，以保证系统的可迭代性与稳定性。

2.1 系统功能性需求分析

本系统完全从用户的角度触发，参考了同类竞品后得出如下功能性需求，会通过用例图的方式对功能性需求进行展现与讲解。

(1) 账户管理需求

实盘训练营是必须用户登录账号才可以进行操作，故每个用户的账号都有该用户独有的账户信息。

游客访问训练营后可以选择注册或者用已有账号登录。进入实盘训练营个人账户系统后，可以选择查询个人账户信息、查询任务列表、查询个人关注、查询可兑换额度、查询金币^[20]，如图 2-1 所示。

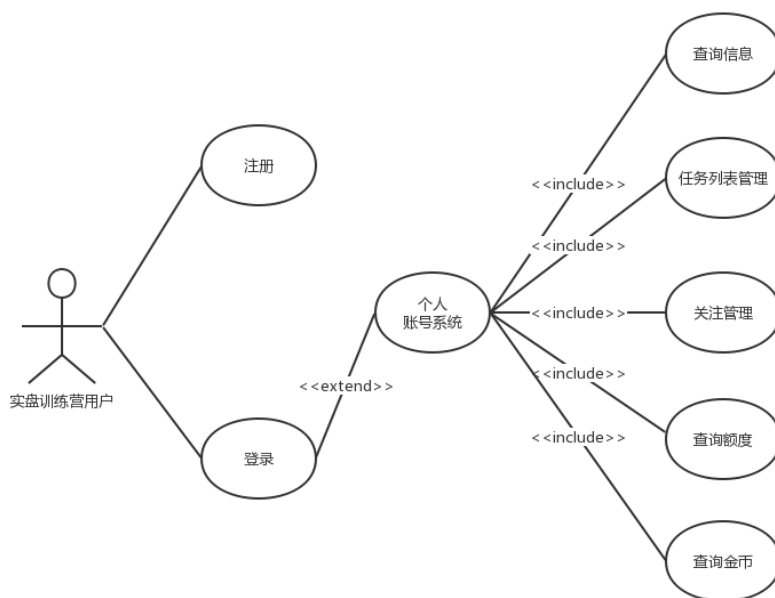


图 2-1 账户管理用例图

(2) 查询行情需求

所有股民都会根据行情进行择股。行情是反应当天股市状况的风向标。用户可查看 A 股、港股、美股、指数或基金等各类大盘走势。并且用户有了解当日涨幅榜、跌幅榜、各行业模块跌涨幅的需求，如图 2-2 所示。

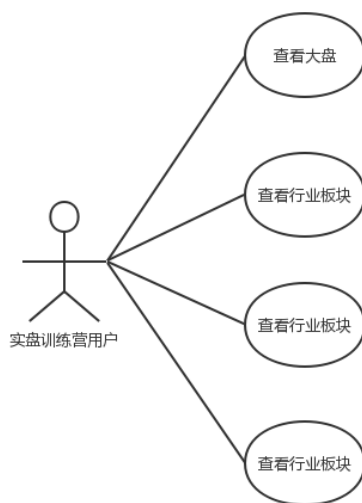


图 2-2 查看行情用例图

(3) 个股查询需求

用户存在查询个股信息的需求，查询自己希望了解的股票代码或名称。用户点击个股后，应进入该股票的详情页以了解股票的具体信息如今日开盘价、昨日收盘价、今日最高价、今日最低价、成交量、成交额、换手率、市盈率、总市值、总股本等基本信息。同时 K 线展示有利于用户分析股票的走势，为用户的投资提供参考。同时用户会希望了解关于该支股票的公告、研报、评分、简况、资金财务等，作为投资的辅助参考信息，如图 2-3 所示。

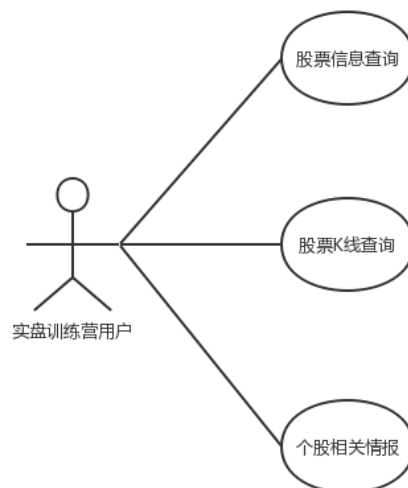


图 2-3 个股查询用例图

(4) 自选股管理需求

大多数用户关注的股票数量都不唯一，为避免每次搜索股票的繁琐，建立用户个人的自选股股票池十分有必要。用户需要查询自选股票、添加自己的自选股票以保持对该股票的关注、删除自选股票。自选股管理用例图如图 2-4 所示。

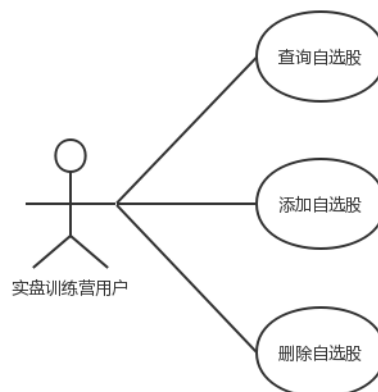


图 2-4 自选股管理用例图

（5）股票模拟交易需求

实盘训练营的核心功能，在于让用户迈出股票交易的第一步，股票交易。只有发生交易，才能够让股票投资者体验到股市的乐趣。为保证与实盘的相似性，应调取与实盘一样的数据接口，以复刻真实的股市信息。该模块主要功能包括股票买入、股票卖出、查看账户资金及持仓的浮动盈亏状况、持仓股票实施浮动信息以及查询今日成交与历史成交记录，如图 2-5 所示。

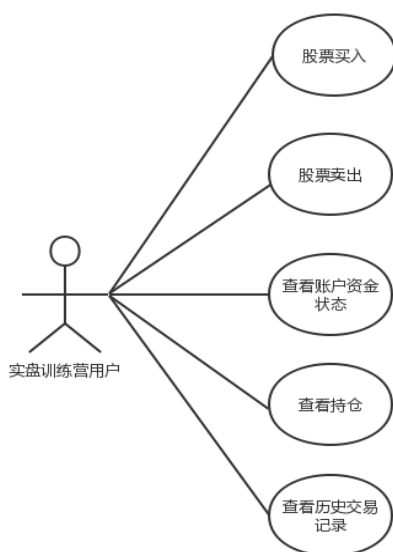


图 2-5 模拟交易用例图

（6）盈利兑换京豆需求

为鼓励新手用户的使用、参与，出台了盈利兑换京豆的规则，所兑换的京豆可以在商城中购买商品时抵扣。用户兑换京豆需满足两个条件：浮动盈亏为正且兑换额度为正。兑换额度为平时在股票平台做任务获得（如浏览资讯、关注牛人等），如图 2-6 所示。

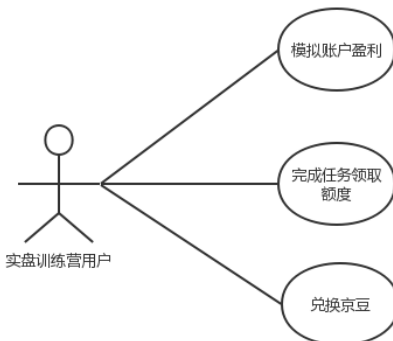


图 2-6 兑换京豆用例图

(7) 浏览资讯需求

新手股市学习者对于股市操作存在迷茫性，面对海量信息，希望获取系统性的学习文章。资讯、新闻、讲解等栏目便提供了这样的平台供股票交易学习者进行学习。同时，用户阅览资讯、新闻可以获得一定的额度，可用于实盘训练营的盈利兑换京豆，如图 2-7 所示。

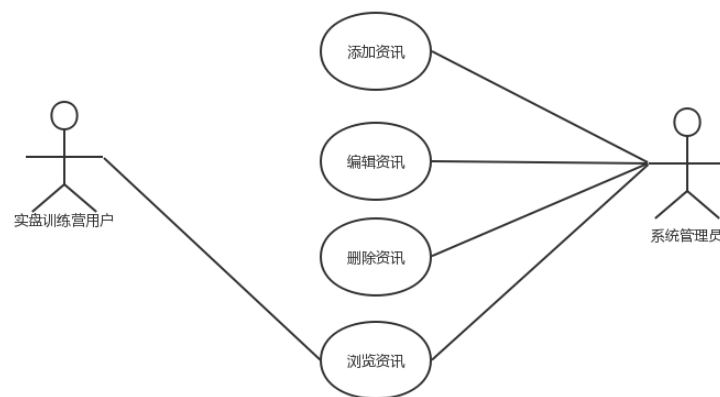


图 2-7 浏览资讯用例图

2.2 系统非功能性需求分析

1) 硬件需求

为达到正常、流畅的使用效果，至少需要的硬件条件如表 2-1 所示。

表 2-1 硬件需求表

硬件类型	要求标准
机器类型	台式 PC 机、便携式 PC 机、平板类设备
CPU	Core i3 及以上
内存	2GB 及以上
显卡	512MB 及以上
网络带宽	10Mbit 及以上
网络类型	ADSL、FTTB-LAN、FTTH

2) 系统及浏览器平台需求

因本系统基于前端开发，开发所用采用技术及标准遵守最新的 ES6 标准，且用

到了 CSS 中的盒模型，且用户使用该系统时，数据的接受、发送、存储均在前端完成，故对浏览器版本有较高的要求，过于落后的浏览器版本可能会导致效果显示不全或无法显示。具体需求如表 2-2 所示。

表 2-2 系统及浏览器平台需求表

平台类型	要求标准
系统	WINDOWS XP/Vista/7/8/8.1/10 MAC OS 8.0 及以上
浏览器	IE10/EDGE/Safari9.0 及以上/Chrome43.0 及以上/Firefox17.0 及以上/Opera45.0 及以上

3) UI 需求

互联网时代的产品对于界面的要求非常高，很多时候一个清爽的界面可以迅速抓住用户的尝试心理，同时能够降低用户的学习成本，从而将新用户拉拢转化为自己的忠诚用户。UI 需求包括：

- (1) 界面清晰，边界清晰无重合。
- (2) 功能区独立
- (3) 基于用户使用习惯的基础上进行构建
- (4) 多页面保持统一的风格。

4) 性能需求

(1) 峰值要求

实盘训练营为潜在面对对象为中国互联网股民中的新手股民。中国股民数量约 1 亿，新手用户占不到 20%。再考虑到股票平台的市场占有份额及未来市场趋势。本系统保证了 100 万用户的峰值在线稳定性。支持 5 万人并行交易操作。

(2) 相应时间要求

股票交易中每一秒都价值千金，每秒价格都再发生变化，为保证系统的响应速度，我们精简了 JavaScript DOM 的结构与数量，让信息的发送与接受路径尽可能的短，同时谨慎的使用了 Cookie，以减少对本地文件的写入与读取^{[20][21]}。以此减少了

系统的响应时间。系统响应时间控制在了 1.2s 内。

5) 拓展性需求

为切合公司未来的战略发展,实时跟进用户与市场需求的变化。为便于日后系统的迭代升级,本系统在开发时注重了拓展性的需求。严格遵循 W3C 标准及 ES6 标准进行开发,严格采用模块化、结构化的设计理念,注重模块的复用。并清晰且完整的定义了数据接口与模块接口。为日后系统功能的修改、迭代奠定了扎实的基础。

6) 稳定性需求

为保证用户体验,对实盘训练营系统的稳定性做了较高要求。尽量保证业务层与数据层传输的独立性^[22]。使用户在低 CPU 占用、低内存占用环境下,系统崩溃发生率不高于 0.1%。在高 CPU 占用、高内存占用环境下,系统崩溃发生率不高于 1%。

7) 可靠性

当系统崩溃时,用户的数据不会因系统崩溃而丢失。在系统在此打开时,数据可得到恢复。

8) 安全性需求

因股票交易涉及资金流动,故对安全性需求要求很高。系统的账号密码等数据需通过算法进行加密^[23]。

用户的个人信息、投资信息、言论信息均受到严格的保护,使保证股票交易市场的隐私性与公平性得到了保证。

2.3 本章小结

本章从用户角度与系统构建角度出发,详细介绍了实盘训练营系统的功能性需求与非功能性需求,并通过用例图与图表进行了讲解。为后续系统的设计划定了范围。

3 系统设计

3.1 系统架构设计

3.1.1 系统架构

实盘训练营整体系统架构采用 MVC 框架设计模式，通过 B/S 结构实现，摒弃了传统的客户端，免去了复杂的安装流程，使用户可以快速上手使用。MVC 模式通过将复杂的系统分为模型、视图与控制器，保证了低耦合性、高重用性、提高了系统的可维护性，并且在开发过程中由于模块的分离更利于开发人员进行分别开发，兼具了效率^[24]。模型用来处理业务层的逻辑模块，像数据库发送读取数据的请求，视图用来呈现数据处理的结果，它根据数据的模型而创建，控制器则是处理用户交互的窗口^[25]。MVC 框架设计模式如图 3-1 所示。

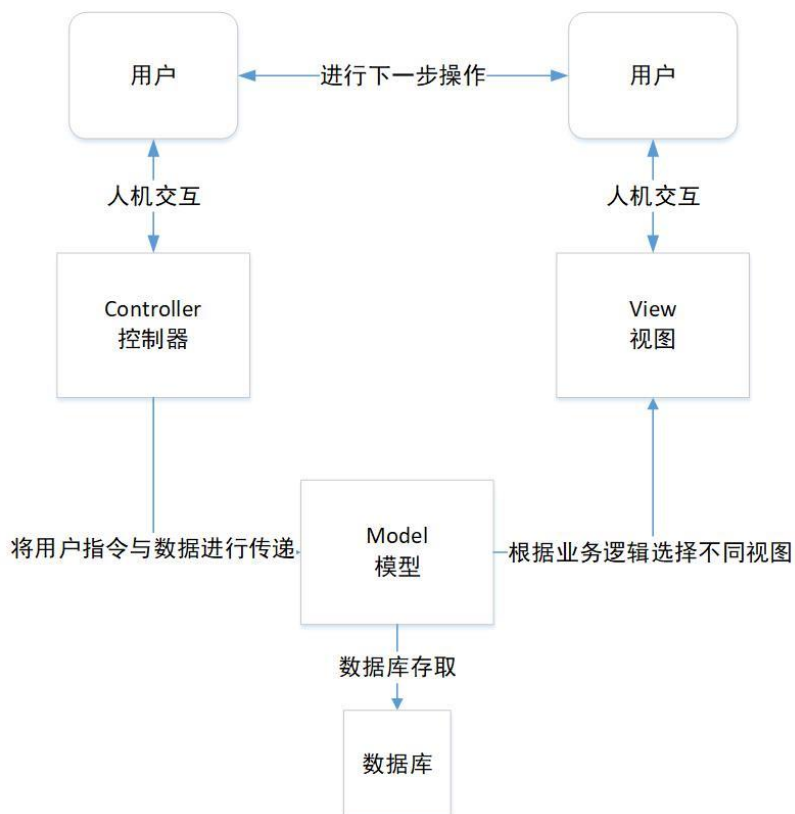


图 3-1 MVC 模式图

B/S 结构与传统的 C/S 结构相比，处理数据负荷对服务器给与的运行压力较小。数据的存储、调用也更加的清晰透明^[26]。并将过往的客户端压力转向了服务器上。降低了升级与维护的成本，提高了访问速度与业务拓展性。

B/S 结构系统主要分为三层：表现层、功能层、数据层^[27]。

表现层为用户可见的操作界面，通过 HTML 与 CSS 框架 Bootstrap 框架构建^[28]。通过设计层叠样式表呈现美观的样式与界面。主要分为四个模块：账户管理、自选股、行情与股票交易。

具体的系统架构如图 3-2 所示。

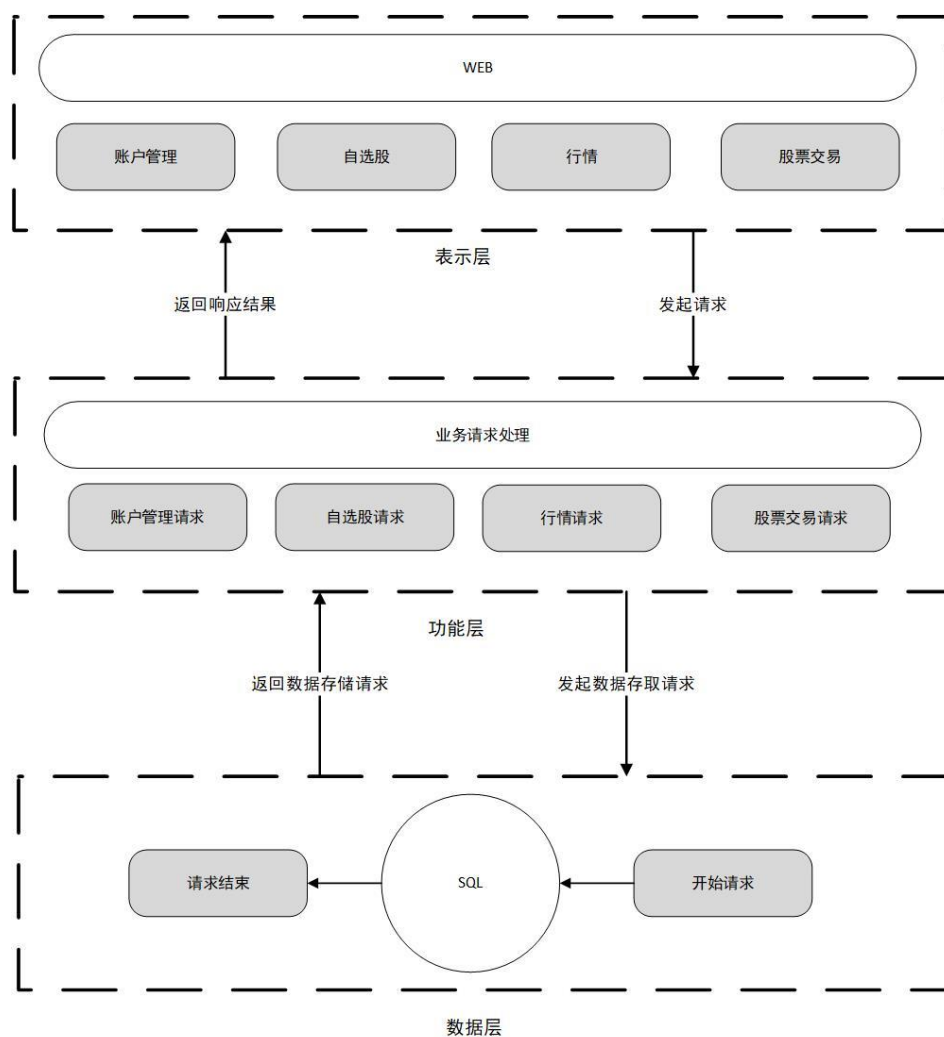


图 3-2 B/S 结构图

3.1.2 前端架构

系统的前端采用了内容、样式、交互分离的设计模式，保证了各模块之间的独立性。我们将 JavaScript 代码与 HTML 文档、CSS 文档严格区分存放，系统隔离了不同的逻辑单元，每一个独立的逻辑单元只存放于独立的文件中，这样便于后期的维护与问题出现时的定位^[29]。

在交互模块中，当用户在表现层进行操作，JavaScript 通过其独特的异步加载模式将用户的指令、操作的结果保存为 JSON 格式传输到功能层，遵循异步加载与按需执行的原理^[30]。相比于传统的瀑布模型加载网络，可以并发的高效处理业务逻辑，同时为保证页面为快速动态页面，使用了 Ajax 技术。

在构建文档对象模型时，为尽可能的提高系统的性能。我们尽可能减少了 DOM 节点的数量，将脚本文件合并存放。同时尽可能的调用自然的 DOM 元素节点与属性，以保证系统运行的时效性。为保证在多浏览器平台上的通用性。开发结构严格遵循平稳退化、渐进增强原则^[31]。即使用户的浏览器无法支持系统中使用的最新 JavaScript 特性，也有其备选方案达到同样的效果，即向后兼容性，以迎合系统的各类非功能性需求标准^[32]。

3.2 系统功能设计

结合系统性功能需求，实盘训练营主要功能结构为用户账户管理、行情查询、个股查询、自选股管理、模拟交易、京豆兑换、资讯浏览、智能荐股，如图 3-3 所示。

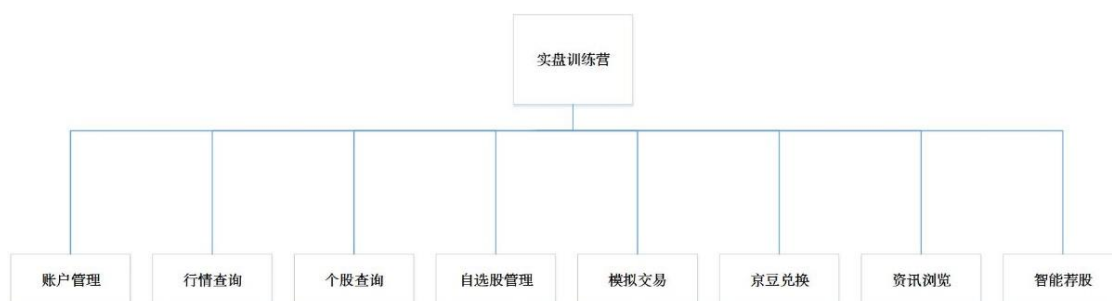


图 3-3 系统功能结构

(1) 账户管理功能设计

用户使用实盘训练营首先要登录系统，这是前提条件，如图 3-4 所示。

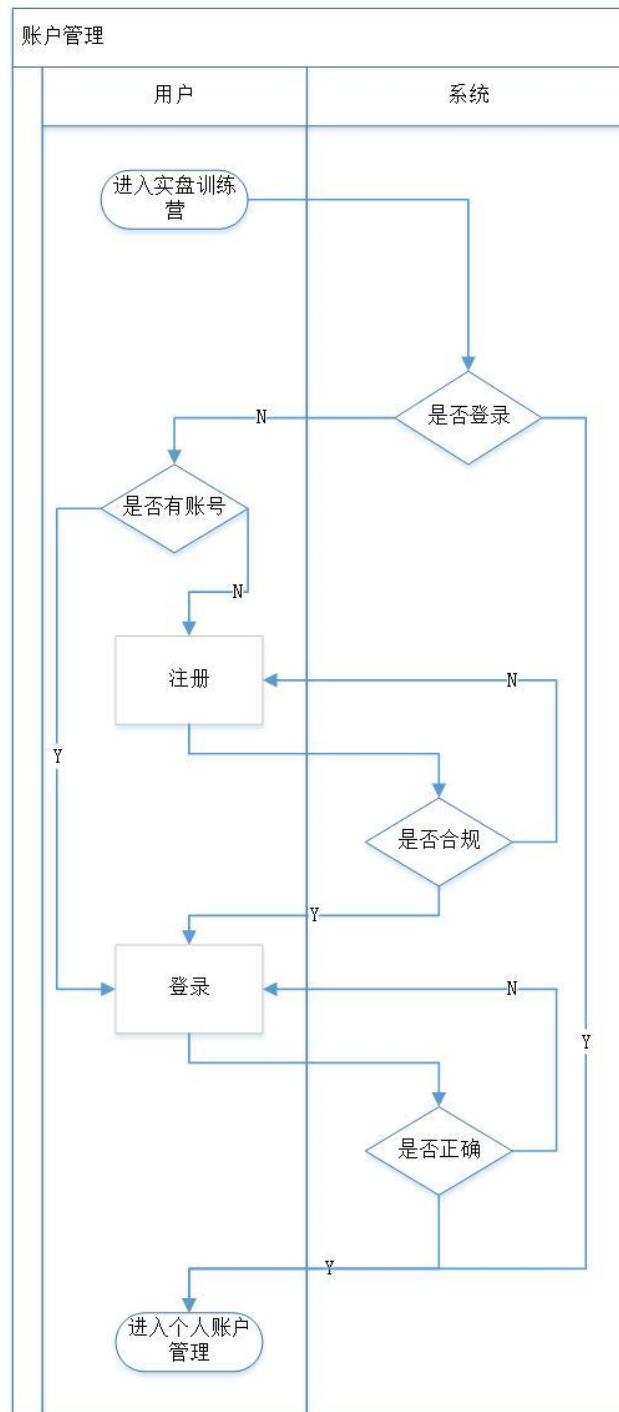


图 3-4 账户管理登录流程图

在用户点击实盘训练营后，系统开始判定账号是否登录，若账号处于已登录状态，则直接跳转至个人账户管理页面。若系统认定用户处于未登录状态，则向用户发出询问是否已有账号。若用户已有账号，则跳转至登录界面，若用户没有账号则进入注册页面。用户键入字符后系统判定是否合规，（合规的条件为账号名称未被占用且密码位数大于 8 位）若不合规重新犯规注册界面进行注册，若合规则注册成功跳转至登录界面。用户输入账户名与密码后系统判定是否正确，若正确则进入个人账户管理页面，若不正确则重新返回登陆界面进行登录。

进入个人账户管理页面后，分为我的任务、我的额度、我的关注、我的账户四个板块。进入我的任务后可以通过浏览资讯、关注牛人获得体现额度。进入我的额度后，可以看到自己还有多少虚拟盈利，可以兑换成为京豆在商城中进行消费。进入我的关注后可以看到自己关注的股票交易牛人列表。进入我的账户后，可以查看今日的收益，持仓浮动盈亏与我的总资产。今日收益为当前总资产减去今日开盘时总资产所得;持仓浮动盈亏为历史持仓盈亏加上当日开盘持仓盈亏所得。

个人账户结构图如图 3-5 所示。

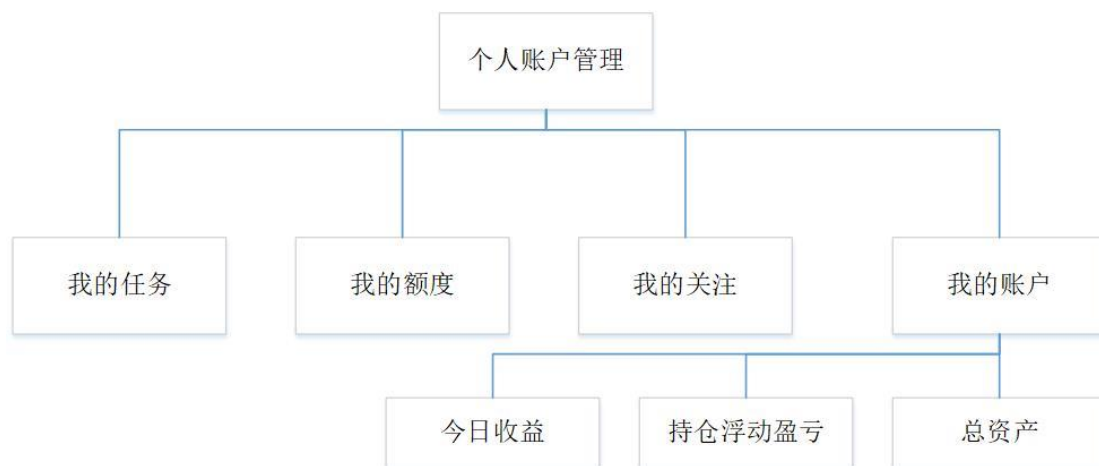


图 3-5 个人账户结构图

(2) 行情查询设计

用户点击行情进入行情页面，会显示 5 个标签栏进行分别切换。分别对应着沪深行情、港股行情、美股行情、指数行情与基金行情。实盘训练营通过连接上海证

券交易所与深圳证券交易所开发的接口,获取他们的 DBF 数据库获取股市行情数据。在沪深行情标签页下有上证、深证及创业板的 K 线走势图以及领涨行业板块与沪深涨幅榜。在港股行情标签页下有恒生指数 K 线走势图与 AH 股榜单与港股涨幅榜。在美股行情标签页下有道琼斯指数、纳斯达克、标普 500K 线走势图以及 EFT 榜和中概股榜单。在指数行情标签页下各类证券市场的指数。在基金行情标签页下有 A 类、B 类及其他基金,如图 3-6 所示。

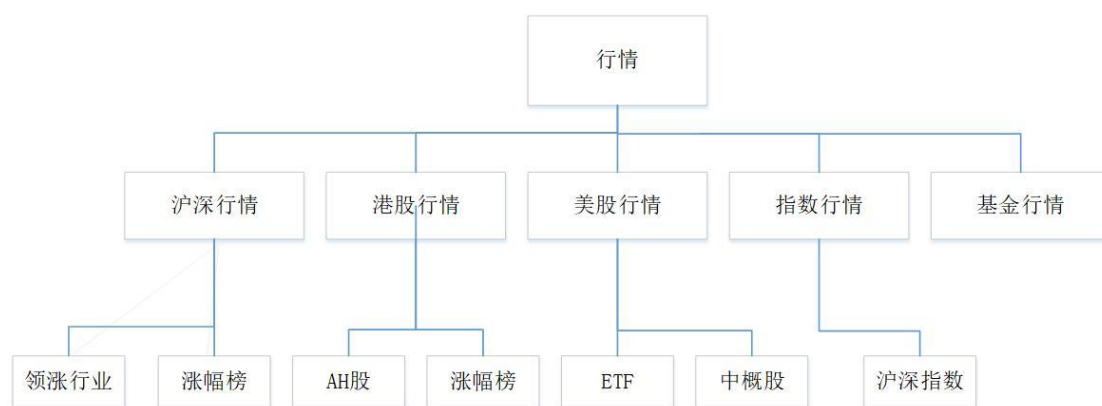


图 3-6 行情结构图

(3) 个股详情页设计

用户通过行情页或者检索股票代码或名称,可以检索到个股。进入个股详情页后,会显示股票的当前价格、今日涨跌幅度、今日涨跌值、总市值、市盈率、当日开盘价、昨日收盘价、当日最高价、当日最低价、当日成交额、当日成交量、换手率、振幅区间等信息。

实盘训练营连接了上海证券交易所与深圳证券交易所开发的接口,通过连接他们的 DBF 数据库获取股市行情数据,在通过 JavaScript 的 Canvas 功能绘制出 K 线图。K 线图按照 5 个不同的时间维度进行浏览,方便用户从长线、中线、短线的角度对该支股票的走势进行分析。

在股票的相关情报选项卡中,可以查看该股票近日来的重要公告、报告、摘要、新闻、来自各大金融机构发布的研究报告。同时本系统还与金融信息服务公司建立合作,推出了股票智能评分功能。通过分析该股票的基本信息,给出成长性、规模、

估值、技术、四项评分。以用来反应该只股票在市场各种因素的综合作用之下的得分。此外还展示股票的简况、资金流向、财务报表等信息。

每个股票的详情页都有评论空间。用户进入评论空间后，可以在迷你论坛中进行留言讨论该只股票的走向。同时也设有投票系统，向用户展现看涨、看跌两个选项。用户投票后可以看到股民对该只股票的看好程度。

个股详情页的结构如图 3-7 所示。

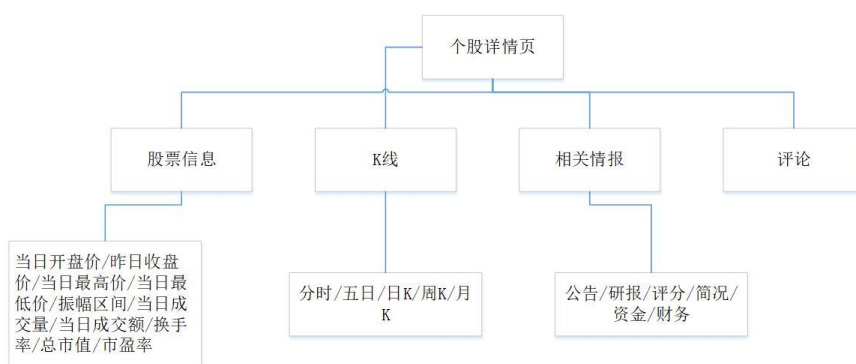


图 3-7 个股详情页结构图

（4）自选股管理设计

用户可以将关注的股票添加至自选股票池，方便每日查看自己关注的股票。股票池每页展示 10 条股票的名称、当前价格、当日涨跌幅。并将当日涨幅为正的股票颜色标红，涨幅为负的股票标绿以作区分。用户在自选股票池中可以根据股票类型（沪深、港股、美股）、当前价格由高到低或由低到高、涨跌幅进行筛选排序。

（5）模拟交易设计

实盘训练营在用户账号初始化时会设置初始资金为 50 万人民币。因为模拟交易系统故不收取用户交易手续费。实盘训练营中的交易时间与沪深股票市场的交易时间保持一致。在股票交易功能中包含三个功能：买入、卖出、查询历史成交。

用户在买入股票时，通过键入证券代码或者股票的拼音或股票的名称，即可选中该只股票。同时会显示该只股票未成交的最低 5 个卖出价格以及未成交的最高 5 个买入价格，并自动生成当前可购买价格。用户可选择买入量。买入量的限制条件为 100 股的倍数即整数手股票。在买入数量栏下设置三挡：全仓买入、1/2 仓买入、

1/4 仓买入。选择相应的档位后可自动计算账户内可用金额所对应的可买股票数自动键入数量栏中，并经系统计算后返回手续费价格。点击买入后系统进行判定，当前是否为交易时间，若处于交易时间区间内，则当价格匹配到最低卖出价后便会买入成功并显示在我的持仓中。

用户卖出股票的交易逻辑与买入股票的交易逻辑保持一致。区别在于用户卖出的股票只能为自身所持有的股票。以买入股票为例展示股票交易逻辑，如图 3-8 所示。

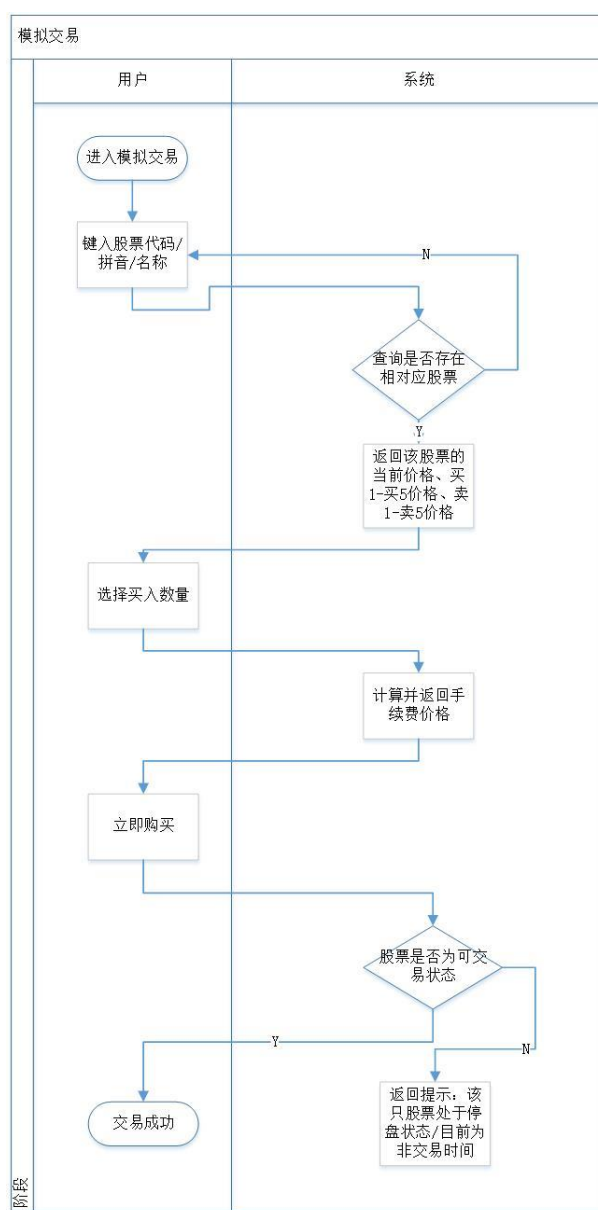


图 3-8 模拟交易流程图

（6）京豆兑换功能设计

为鼓励股票交易学习用户的积极性，设立了股票盈利兑现京豆的服务。京豆可在商城中消费时抵扣。当用户模拟账户中的持有股票均卖出后，若总资产大于 50 万，且拥有兑换额度，则可兑换相应比例的京豆。

3.3 系统数据库设计

考虑到该平台的用户量较大，交易信息复杂的特点，我们将用户个人信息及交易信息的数据存储于 Oracle 数据库中，相比于 MySQL、SqlServer 等关系型数据库，Oracle 对大型数据的存储、读取更为友好，它拥有高敏捷性、低延迟性、高安全性、均衡负载等诸多优点。

结合前文所述的需求分析与功能设计，下文主要展示几个关键性数据的存储数据库表如用户个人账号表、个人交易账户表、个股详情表与实盘训练营运营监控表。数据库表对应关系如 E-R 图 3-9 所示。

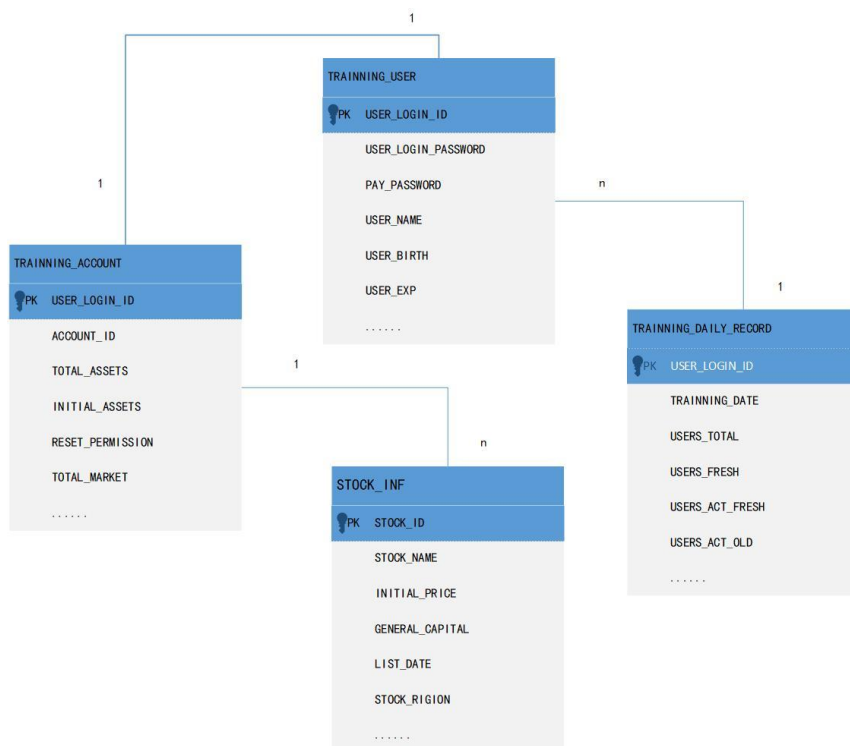


图 3-9 数据库 E-R 关系图

(1) 用户个人账号表

该表用来储存用户账号的相关信息，诸如个人信息，系统信息等。
 USER_LOGIN_ID 为该表的主键。用户的个人信息当用户在表示层建立账号时，根据用户在 HTML 表单中填写的内容，转化为 JSON 格式文件，传入后台，写入数据库，如表 3-1 所示。

表 3-1 个人账号表

表名: TRAINNING_USER			
字段名	数据类型	是否为空	描述
USER_LOGIN_ID	VARCHAR	N	用户账号
USER_LOGIN_PASSWORD	INT	N	用户密码
PAY_PASSWORD	VARCHAR	N	支付密码
USER_NICKNAME	VARCHAR	N	用户昵称
USER_NAME	VARCHAR	N	用户姓名
USER_NAME_PY	VARCHAR	N	用户姓名拼音
USER_AGE	INT	N	用户年龄
USER_TEL	VARCHAR	N	用户联系方式
USER_IDEN_NUM	INT	N	用户身份证号
USER_CRE	DATETIME	N	用户注册时间
USER_COUNTRY	VARCHAR	N	用户国籍
USER_REGION	VARCHAR	N	用户所在地区
USER_BIR	DATETIME	N	用户出生年月日
USER_SEX	CHAR	N	用户性别
USER_STOCK_TYPE	VARCHAR	N	用户关注股市类型
USER_PIC_ADDR	VARCHAR	N	用户上传头像储存路径
USER_PASSWORD_Q	VARCHAR	N	用户找回密码设置问题
USER_PASSWORD_A	VARCHAR	N	用户找回密码问题答案
USER_EMAIL	VARCHAR	Y	用户邮箱地址
USER_EXP	VARCHAR	N	用户投资经验年限
USER_RELATED	INT	N	是否关联商城账号
TASK_NUM	INT	N	今日任务完成数量
NEWS_NUM	INT	N	今日资讯阅读数量
TASK_FINISHED_NUM	INT	N	任务总完成数
NEWS_READ_NUM	INT	N	资讯总阅读数
MASTER_FOLLOW_NUM	INT	N	关注牛人数量
JBEAN_LIMIT	INT	N	提现京豆额度
JBEAN_EXCHANED	INT	N	已提现京豆数量
PUSH_PERMISSION	INT	N	推送权限
CAM_PERMISSION	INT	N	相机访问权限
PHOTO_PERMISSION	INT	N	相册访问权限
MICROPHONE_PERMISSION	INT	N	麦克风访问
STATE	INT	N	账号状态 0 为正常 1 为被封禁 2 为账号已注销

(2) 个人交易账户表

个人交易账户表用来存储用户的虚拟资产等信息，USER_LOGIN_ID 为主键，如表 3-2 所示。

表 3-2 交易账户表

表名: TRAINNING_ACCOUNT			
字段名	数据类型	是否为空	描述
USER_LOGIN_ID	VARCHAR	N	用户账号
ACCOUNT_ID	VARCHAR	N	模拟账户 ID
TOTAL_ASSETS	INT	N	模拟账户总资产
INITAIL_ASSESTS	INT	N	初始资产金额
RESET_PERMISSION	INT	N	是否持有重置权限
TOTAL_MARKET	INT	N	模拟账户持仓总市值
USABLE_ASSETS	INT	N	账面可用资金
POP_ASSEST	INT	N	可取出资金
TODAY_EARNING	INT	N	今日收益
FLOATING_EARNING	INT	N	持仓浮动盈亏
TODAY_EARNING_RATE	INT	N	本日收益率
WEEK_EARNING	INT	N	本周盈亏
WEEK_EARNING_RATE	INT	N	本周收益率
MONTH_EARNING	INT	N	本月盈亏
MONTH_EARNING_RATE	INT	N	本月收益率
DAY_RANKING	INT	N	本日排名
WEEK_RANKING	INT	N	本周排名
MONTH_RANKING	INT	N	本月排名
TOTAL_EARNING	INT	N	总收益排名
LAST_TRADE_TIME	DATETIME	N	上次交易时间
LAST_TRADE_TYPE	VARCHAR	N	上次交易股票品种
STOCK_ID	INT	N	持有股票代码
STOCK_NAME	VARCHAR	N	持有股票名称
STOCK_NUM	INT	N	持有股票数量

(3) 个股详情表

表格包含了个股的关键信息，STOCK_ID 为主键，如表 3-3 所示。

表 3-3 股票详情表

表名: STOCK_INF			
字段名	数据类型	是否为空	描述
STOCK_ID	INT	N	股票代码
STOCK_NAME	VARCHAR	N	股票名称
STOCK_PY	VARCHAR	N	股票拼音
CO_NAME	VARCHAR	N	公司名称
INITIAL_PRICE	INT	N	股票发行价
GENERAL_CAPITAL	INT	N	总股本
DISTRI_QUANTITY	INT	N	股票发行数量
HOLDER_QUANTITY	INT	N	股东人数
STOCK_QUANTITY_AVE	INT	N	平均股数
LIST_DATE	DATETIME	N	上市日期
REGISTERED_CAPITAL	INT	N	注册资金
STOCK_REGION	VARCHAR	N	所属区域
STOCK_TOD_INI	INT	N	当日开盘价
STOCK_TOD_CLO	INT	N	昨日收盘价
STOCK_TOD_TOP	INT	N	当日最高价位
STOCK_TOD_LOW	INT	N	当日最低价位
STOCK_VOLUME	INT	N	成交量
STOCK_TURNOVER	INT	N	成交额
STOCK_TURNOVER_RATE	INT	N	换手率
STOCK_PE_RATIO	INT	N	市盈率
STOCK_RATIO	INT	N	量比
STOCK_AMP	INT	N	振幅
STOCK_TBT_PRICE	INT	N	实时价格

(4) 实盘训练营运营监控表

为记录实盘训练营的运营状况，记录实盘训练营每日用户的使用情况，利于后期分析迭代升级的方向，如表 3-4 所示。

表 3-4 实盘训练营运营监控表

表名: TRAINNING_DAILY_RECORD			
字段名	数据类型	是否为空	描述
TRANNING_DATE	DATETIME	N	日期
USERS_TOTAL	INT	N	总用户数
USERS_FRESH	INT	N	新用户数
USERS_ACT_FRESH	INT	N	当日登录新用户数
USERS_ACT_OLD	INT	N	当日登录老用户数
USERS_ASSETS_TOTAL	INT	N	用户账户总金额
USERS_REC_ASSEST	INT	N	用户今日领取金额
USERS_TRA_NUM	INT	N	参与交易用户数

续表 3-4 实盘训练营运营监控表

表名: TRAINNING_DAILY_RECORD			
字段名	数据类型	是否为空	描述
USERS_STOCK_TOA	INT	N	用户持股总数
USERS_STOCK_AVE	INT	N	人均持股数
USERS_TRA_TOA	INT	N	用户总参与次数
USERS_TRA_AVE	INT	N	用户人均参与次数
USERS_PLUS_NUM	INT	N	总盈利用户数
USERS_MINUS_NUM	INT	N	当日盈利用户数
USERS_EXHCANGED_NUM	INT	N	当日兑换京豆用户数
USERS_EXCHANGED_TOTAL	INT	N	兑换京豆总用户数
JBEAN_EXCHANGED_TOTAL	INT	N	京豆兑换总量
JBEAN_EXCHANGED_VALID	INT	N	京豆可兑换余额
USERS_EXCHANGED_AVE	INT	N	人均京豆兑换数

用户体验最重要的因素之一就是响应反馈时间,为保证用户得到响应的及时性,我们在数据库读取上做了优化。

首先最重要的是存储介质的选择,不同的存储介质带来的读取、写入速度具有极大的差别。机械硬盘的带宽普遍为 50-250mb/s,延时 5-20ms;而机械硬盘可将带宽提升至 2Gb/s,保证了数据的持续读写能力,同时延时将至 1ms 以下,提高了突发事件的处理能力。故本系统的数据存储选择了固态硬盘,以获得大幅度的读写速度的提升。

其次为减少数据的不必要访问。为此本系统通过将字段的分类建立了索引,通过这种成本最低的方式达到了最大的效果。减少了数据的不必要遍历读取。

在返回数据的方式上,我们选择了只返回必要字段。以此削减读取数据时在网络传输中的损耗,并降低了数据处理在服务器端的开销、减少了客户端对于内存的占用。

同时,数据库的读写尽可能的减少了比较操作。在海量数据情况下,尽可能选择 search 进行全文检索以替代 like 模糊查询。在大量复杂运算出现的情况下,为避免数据库服务器的过载,我们选择将处理行为放置在客户端进行处理,以减少服务器的高并发处理情况。

本系统在数据大于百万级时采用数据并行处理操作。服务端每接收一条来自客户端的 SQL 指令，都会调用数据库服务器的并行调度进程，将指令分解成为若干进程进行操作。它充分利用服务器端的 IO 资源，减少性能闲置。但这样的处理在单个会话中会占用大量的系统资源，从而影响其他数据读写操作，故并行处理包含前置条件：在服务器 CPU 占用率低于 20% 时触发。以此保证在服务器过载时的数据正常传输。

最后本系统建立的缓存机制。将常用的高频字段保存于缓存之中，以减少对数据库的读写操作，当大量数据访问时，避免了长时间的等待延时，提升系统读取效率。

3.4 系统性能设计

考虑到系统为 B/S 结构，用户在浏览器端所接受数据的响应速度直接影响了用户的体验。故在系统的响应速度上尽可能的做了性能优化处理。

（1）减少 HTTP 请求次数

对于前端系统响应最大的便是用户的请求次数。用户每次发出 URL 连接请后，便产生了一次连接指令，服务器接收请求后再返回数据。用于连接的过程会大大加大响应耗时。故我们尽可能的将文件打包，减少用户的总请求次数。在处理图片方面，用到了 CSS Sprites 技术，将多种 Logo、UI 图片进行合并。用户在向服务器发出第一次请求后，便会一次性下载所有的图片集合。用户在访问不同页面的图片时，会调用该图片的切割小图。在实现了减少请求、提高响应的效果之上，用户的视觉效果并不受影响^[33]。

（2）限制 DOM 节点的访问次数

每一次访问 DOM 节点都会产生一层层的向下遍历，在 DOM 结构设计时，尽可能简化了 DOM 节点的数量，以减少请求耗时。在一些需要使用深层 DOM 节点的情况中，利用了 DocumentFragment 来暂时存放节点^[34]。

（3）JSON 数据传输

以往 JavaScript 向后台传输数据多用 XML 格式，为达到性能优化的目的。本系

统采用了 JSON 这种轻量级数据交换格式。得益于其原生的特点，JavaScript 在调用时无需调用 API^[35]。

（4）初始化负载分解

系统将所有 CSS 代码单独储存，保持与 HTML 文件的绝对分离，将 CSS 文件放置在结构头部，JS 文件存放于结构尾处。在浏览器解析页面时，会预先加载 CSS 文件，而用户在刚打开页面时不会进行交互操作，故 JS 文件可在 CSS 文件加载后再进行加载，以保证页面呈现的优先性^[36]。做到单位时间内对数据的有效利用，节省了读取时间。

在页面加载时，对 JS 文件的功能性加以区分，与页面内容、样式有关联的 JS 文件首先加载，而主导交互与逻辑判断的 JS 文件放置在 Onload（）函数触发后加载^[37]。

3.5 本章小结

本章主要介绍了实盘训练营系统的整体架构设计、前端架构设计、系统主要功能的逻辑设计、主要数据库的关联与字段设计与数据库的性能优化。最后阐述了为打造高性能系统而做的优化性设计。

4 系统实现

实盘训练营系统是基于 B/S 结构实现的。在实现章节，挑选重要及关键业务环节、功能模块进行实现方式的论述，主要包括账户管理功能、行情功能、交易功能、自选股功能、兑换功能等。部分设计的实现包含核心代码的描述。并对系统的性能优化通过示例进行实现论述。系统的功能界面通过截图进行展示说明。

4.1 开发环境及服务器部署

系统内容通过 HTML 语言呈现，并遵循 HTML5 标准。系统基于 Bootstrap 框架对页面样式进行布局，系统交互逻辑通过 JavaScript 控制，语言遵循 ES6 标准。数据通过 JSON 格式文件传输，数据库采用 Oracle 进行存储。系统模块通过 Webpack 进行打包，实现类库模块化。在构建股票、行情 K 线图时，采用 JavaScript Canvas 进行绘制。开发工具采用 WebStorm 与 HBuilder，采用 Chrome Dev Tools 进行调试，如表 4-1 所示。

表 4-1 开发环境及工具

类别	细分	开发环境或开发工具及版本
实现语言	前端	JavaScript ES6
	数据库	Oracle11
	页面内容实现	HTML5
	页面结构实现	Bootstrap2.3.2、LESS2.53
实现环境	CPU	Core I7
	内存	16GB
	操作系统	Windows7
开发工具及平台	开发工具	Webstorm2018、HBuilder2018、Sublime3、Eclipse5
	项目管理工具	Webpack、Tortois SVN1.9.0
	调试工具	Chrome Dev Tools
	测试工具	Unit JS
	设计工具	Microsoft Visio2016、Axure RP8.0、PlantUML、ProcessOn

前端服务器、数据库服务器、行情服务器与委托服务器均采用了 Xeon5 处理器以保证系统的快速响应。服务器部署如表 4-2 所示。

表 4-2 服务器部署

服务名称	硬件配置	操作系统
前端服务器	Xeon 5	Linux Centos 6.5
	16GB	
	1T	
数据库服务器	Xeon 5	Linux Centos 6.5
	32GB	
	2T	
行情服务器	Xeon 5	Linux Centos 6.5
	32GB	
	1T	
委托服务器	Xeon 5	Linux Centos 6.5
	32GB	
	1T	

4.2 系统功能实现

本小节根据系统的功能需求设计对于系统的主要功能、核心功能的关键实现步骤进行讲解。系统实现界面如图 4-1 所示。



图 4-1 主界面

(1) 账户管理功能

用户进入系统后,首先调用 `isOnload()` 函数判断用户是否处于登录状态,若处于未登录状态则跳转至注册页面。在页面中设置监听事件,当用户完成提交操作时,触发监听事件,利用 JavaScript DOM 寻找关键表单节点。将本文框中的信息记录成为 JSON 文件,返回后台。

事件绑定函数代码如下:

```
function addEvent(element, event, listener) {  
  if (element.addEventListener) {  
    element.addEventListener(event, listener, false);  
  }  
  else if (element.attachEvent) {  
    element.attachEvent("on" + event, listener);  
  }  
  //对 IE 内核浏览器添加事件监听  
  else {  
    element["on" + event] = listener;  
  }  
}
```

//通过 `stringify()` 方法将 JavaScript 对象解析成为序列化的 JSON 字符串保存,再传送至数据库中。

```
var inf = document.getElementById("num-input").value;  
var Text = JSON.stringify(inf);
```

注册完成后,进入用户登录界面。为保证页面的无卸载式直接交互型体验,采用了 Ajax 异步加载。这样可实现将前端数据表格中的信息与后台交互并反应回来后,页面仍然不重新加载,大大提高用户体验。

Ajax 核心代码如下:

```
xmlhttp.open("GET","inf.txt",true);  
xmlhttp.send();
```

//在这里选用了 GET 方法,因为信息的传输数据量并不大,

且 GET 可以保证用户体验的速度响应, 且表格空间中的字符均为限制性字符, 故不采用 POST 方法。

```
function transInf()
{
    loadXMLDoc(inf,function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
//检查响应状态码
        {
            document.getElementById("infText").innerHTML=xmlhttp.responseInf
        }
    });
}
```

在这里使用 XHR 对象调用 open()函数, 它指向处于单个域当中的一致端口和协议 URL 发送。请求分派到服务器后, 响应的数据会对 XHR 对象属性进行填充, 再返回前端页面。其中 responseInf 用来返回文本信息, status 表示响应 HTTP 的状态码。响应被接收后, 开始检查 status 值, 当值为 200 时表示响应返回成功, 故在此添加了条件判断语句。注册页实现效果如图 4-2 所示。

The image shows a registration form with a horizontal tab bar at the top containing five tabs: '基本资料' (Basic Information), '设置头像' (Set Avatar), '账户安全' (Account Security), '修改密码' (Change Password), and '账号绑定' (Account Binding). The '基本资料' tab is selected. Below the tabs, the form contains the following fields and controls:

- 昵称:** A text input field with a red asterisk indicating it is required.
- 姓名:** A text input field.
- 性别:** Radio buttons for '男' (Male), '女' (Female), and '保密' (Keep Private).
- 生日:** Three dropdown menus for year, month, and day.
- 所在地:** Two dropdown menus for province/straight city and region/city.
- 联系地址:** A text input field.
- 从事行业:** A text input field.
- 个人简介:** A text area containing the placeholder text '这个人很懒, 什么都没有留下'.
- 保存设置:** A blue button at the bottom right of the form.

图 4-2 注册页面

若 onLoad () 函数判定用户为登录状态或用户完成注册成功后。跳转至个人账户页面。执行 isNew () 函数判定用户是否为新用户, 新用户则启用 user.initial()方法初始化用户资产为 20 万元。若用户为老用户则直接读取用户所对应的数据库信息。

个人账户实现效果如图 4-3 所示。



图 4-3 个人账户页面

(2) 行情功能实现

股市行情通过深圳证券交易所数据库与上海证券交易所数据库的 DBF 文件获得, 通过 JavaScript Canvas 进行绘制而成。考虑到 K 线图为 2D 图像, 故在此采用 Canvas 绘制。我们首先将<canvas></canvas>标签嵌入文件中, 设定绘制区域的坐标 width 与 height。K 线中的柱状形图形通过 fillRect()函数绘制。

K 线中分别用红色和绿色代表涨跌, 以红色柱状为例。

```
var canvas=document.getElementById('myCanvas');
```

```
//通过节点 id 找到画布元素
```

```
var =canvas.getContext('KChart');
```

```
stockInc.fillStyle='#FF0000';
```

```
//设置颜色
```

```
stockInc.fillRect(x,y,a,b);
```

```
//设置坐标
```

调取 canvas 画布，定位 Kchart 节点，设定填充颜色与坐标及大小。K 线中柱状的大小由股票的涨幅/跌幅所决定，股票的涨跌幅由证交所传递的数据所决定。故我们将证交所传来的信息输入至 fillRect()函数即可表现出股票的涨幅。

K 线定义如下：

```
function KChart(begin,high,low,last,width){、  
    //设置边界  
    this.begin= begin;  
    this.high = high;  
    this.low = low;  
    this.last= last;  
    this.width = width | 8;  
}
```

其中 begin 表示开市时价格，last 表示收盘时价格，high、low 分别表示股票的最高最低价格。

```
KChart.prototype.draw = function(place,x,low,high,screen){  
    //根据 low,high 计算坐标  
    var _oplace = screen - (this.oplace - low)*high - 10;  
    var _high = screen - (this.high - low)*high - 10;  
    var _low = screen - (this.low - low)*high - 10;  
    var _close = screen - (this.close - low)*high - 10;  
    //根据 low,high 画 K 线  
    place.save();  
    //颜色设置  
    if(this.close >= this.oplace){  
        place.fillStyle = "#FF0000";  
        place.strokeStyle = "#FF0000";  
    }else{
```

```
        place.fillStyle = "#00FF00";

        place.strokeStyle = "#00FF00";

    }

    //线条绘制

    place.beginPath();

    place.moveTo(20 + x * this.width*2 + this.width,_high);

    place.lineTo(20 + x * this.width*2 + this.width,_low);

    place.stroke();

    //坐标定位

    var temp = ((_oplace > _close)?_oplace:_close);

    if(Math.abs(_oplace - _close) == 0 ){

        place.moveTo(35 + x * this.width*2,y);

        place.lineTo(20 + (x+1) * this.width*2,y);

        place.stroke();

    }else{

        place.fillRect(35+,Math.abs(_oplace - _close));

    }

    place.restore();

}
```

Place 代表画布的位置，high 是单位价格对应的高度，screen 是画布的整体高度，因 Canvas 是左上角坐标点固定，故需 screen 做坐标转换。

首先根据价格计算最小坐标，再根据最小坐标绘制 K 线，根据收盘价和开盘价确定阴柱与阳柱，设置相应的颜色，根据最高价与最低价画线，根据开盘价与收盘价画矩形。

K 线图通过坐标系创建图表类。

```
function KChart(dom){

    var KWidth = parseFloat(dom.style.KWidth.replace("px",""));
```

```
var KHeight = parseFloat(dom.style.KHeight.replace("px",""));

var KChart = document.createElement("KChart");

//创建绘图元素

KChart.style.KWidth = KWidth + "px";

KChart.style.KHeight = KHeight + "px";

KChart.KWidth = KWidth;

KChart.KHeight = KHeight;

    //将 KChart 节点插入文档对象模型

    dom.appendChild(KChart);

var context = KChart.getContext("2d");

this.KWidth = KWidth;

this.KHeight = KHeight;

this.pen = context;

}
```

构建 KChart()函数，通过 DOM 寻找元素节点，为图标类赋予坐标系属性。

K 线图并不是静止的图片，用户鼠标的滑动及操作均可实现对线图的操作。故我们需添加对鼠标移动与鼠标拖拽两个行为的监听。通过 addMouseMove()与 addMouseDrag()两个函数实现。K 线实现效果如图 4-4 所示。



图 4-4 K 线效果图

(3) 自选股功能实现

构建自选股数据列表,用户可添加自己关注的股票进入自选股股票池,方便每日观察股票涨跌幅行情。股票的属性包括股票的名称、代码、所属股票市场、实时价格、涨跌幅价格。通过 `stock.fresh()` 方法保持 5 秒的刷新频率以展示股票的最新价格。关键代码如下:

```
box.put('id', function(req, res, next) {  
    var body = req.body;  
    var id = req.param('id');  
    if(!id) return next(new Error('Stock id cannot be blank.'));  
    //判断股池状态  
    var stockAttrs = ['symbol', 'type', 'setting'];  
    body = req.pick(body, stockAttrs);  
    //读取股票列表  
    if(body.setting) {  
        var allowSettingAttrs = ['enable', 'count', 'range_percent', 'interval'];  
        body.setting = _.pick(body.setting, allowSettingAttrs);  
    }  
    //自选股更新  
    Stock.findByIdAndUpdate(id, body, {upsert: true}, function(err, doc){  
        if(err) next(err);  
        res.json({result: true, data: _.extend(doc, body)});  
    });  
});  
//删除自选股  
box.delete('/:id', function(req, res, next) {  
    var id = req.param("id");  
    Stock.findByIdAndRemove(id, function(err, doc) {
```

```
    if(err) next(err);

    res.json({result: true, data: {id: id}});

  });

});
```

box 下的 put()与 delete()函数通过对股票名称或股票代码的识别实现对自选股票池中的股票添加与删除。

(4) 交易功能实现

用户键入自己希望买入的股票代码或名称, 读取相关股票的价格, 并根据户个人账户中的可用资金数量, 计算可买入的股票数。自动生成可购买数字, 并提供全仓买入、1/2 仓买入、1/4 仓买入三个快捷选择。点击立即买入后, 系统计算是否为正常交易时间, 若交易时间正常股票进入持仓池中。核心代码如下:

```
const { mainDB, redisClient, jpushClient } = singleton ;

//async 异步加载

export default async({ Id: OrderId, Commission, delta, AccountNo, OrdType, Side, OrderQty,
Price, SecuritiesType, SecuritiesNo, MemberCode, Amount }) => {

  let Type = ((OrdType - 1) / 3 >> 0) + 1;

  //等待异步加载完成

  let{ Cash,UsableCash }=await;

  singleton.selectMainDB0("wf_street_practice_account", { AccountNo });

  let{ Positions=0,TradAble=0,CostPrice,Id:PositionsId}=await;

  //条件判断股票状态

  if (Side == "null" [Type - 1]) {

    Positions -= OrderQty;

    if (Type == 2) delta += OrderQty * (CostPrice - Price)

    if (Positions > 0) {

      await singleton.updateMainDB("wf_street_practice_positions", { Positions },

null, { Id: PositionsId }, t);
```

```
} else if (Positions == 0) {  
    await singleton.deleteMainDB("wf_street_practice_positions",  
    UsableCashDelta = Amount + delta  
}
```

实现效果如图 4-5 所示。

图 4-5 交易功能

(5) 京豆兑换功能实现

为鼓励用户使用实盘训练营的积极性，每日系统支出 100000 京豆用于用户兑换盈利，设立奖金激励池用于当日盈利用户的京豆兑换。

每日的盈利用户根据其盈利数额占有所有盈利用户的盈利数额总数的比例进行等比例投放。当用户持仓的股票卖出后，根据其盈利金额及兑换额度，可进行京豆兑换操作。

首先系统通过 `profitAll()` 函数计算当日股市交易结束时间后的所有盈利用户的盈利金额，利用读取 `profitUserNum` 值，即计算当日盈利用户数数量，再通过 `profitIndivisual()` 函数读取用户在该交易日中的盈利总金额。

当数值大于 0 时视为有效，然后调用 `profitAll()` 的计算结果，进行计算用户盈利占盈利用户盈利总金额占比。再判断提现额度上限，并返回用户当日可兑换的京豆数量。

4.3 性能优化实现

为保证前端系统的高响应性，针对于 HTML DOM 文档结构的节点进行了优化，尽可能将树形结构的层数减少，以减少用户访问时的遍历次数。同时为提高加载速度，我们采用了无阻塞的加载脚本。

目前市面上大多数用户所使用的主流浏览器在加载页面时，会对页面中的除脚本文件外的不同组件元素进行并行下载操作。在脚本文件被用户下载直至脚本文件下载完成后被浏览器解析的过程之中，不会并行下载其他组建。且考虑到脚本间的功能可能存在相互作用的关系，即后面下载的脚本可能为前面下载的脚本运行的基础，若并行下载脚本，则含有前置条件的脚本文件无法正常运行。故脚本文件往往放置于<body>标签之前，以前少上述情况的发生。在实盘训练营系统中，我们采用了 Script Defer、动态脚本元素等技术是脚本的下载阻塞时不至于阻碍页面正常运行。我们通过在<script>标签中加入 defer 限制条件，代码如下：

```
<script type="text/javascript" src="xxxx.js" defer></script>
```

当脚本不会改变文档内容的前提下，使用 defer 属性可以让浏览器在不执行脚本的前提下解析并安全顺利的读取文档剩余内容，当文档被完整解析后再进行脚本处理，从而加快页面的加载速度。

为保证脚本的下载不阻塞其他页面的处理，我们针对于某些脚本采用了动态创建 script 标签的方法。代码如下所示。

```
var sp = document.createElement ("script");  
sp.type = "text/javascript";  
sp.src = "xxx.js";  
document.body.appendChild(script);
```

通过 JavaScript 为文档添加脚本元素，并赋予脚本文件的属性，最后利用对 HTML DOM 的操作将脚本插入进页面内。

此外我们通过使用 CSS Sprite 将图片、图标进行整合，形成单独资源包，以减少用户的访问下载次数，减少传输损耗。以往的图片读取往往是每一个小的图标单

独占一个资源位，而整合后的图标集合在一张大图之中，系统只需要下载一次。每次访问调用时，利用 `background-position` 属性对图片进行定位切割，选取所用图标，进行显示。

虽然使用 `Sprites` 可能会导致图片质量的下降，因为在图片合成时会存在连接处的空白区域，但凭借削减颜色表与格式信息的特点，大部分情况下降低了图片的开销。

一般在谈及 B/S 系统性能时，会以系统对用户操作的反馈时间、系统稳定性、系统对于用户并行访问的性能等等方面进行评估。而在 B/S 结构的系统中，数据库又是系统的关键中间节点。应用层的 Web 服务器发出请求调用的信号，并读取数据后把信息发送给用户。因为为保证响应性，采用了高性能的数据库服务器。

为了让系统页面在用户读取时能够更快的显示。我们将样式表文件统一放在了头部位置，使内容达到逐步呈现的效果，以此避免样式变化带来的重绘页面，白屏的出现。若将样式表放在底部，则会妨碍浏览器中内容的逐步呈现效果，导致解析资源的浪费。

同样为了页面的加载更快速，我们将脚本文件放在了最底部。因为 JavaScript 文件在下载状态下执行，会妨碍页面 DOM 树形结构的解析，或者说是中断了树形结构的更新。即使在使用不同主机名的情况下，脚本的下载依然不可并行操作。由于脚本文件的执行可能包含修改页面样式、内容的操作信息，所以浏览器会在解析页面时呈等待状态，以保证脚本文件的按顺序性执行。

4.4 本章小结

在本章中主要介绍了实盘训练营系统在开发环境和服务器的部署结构以及系统主要核心功能的实现逻辑与部分核心代码，并对系统实现效果通过图片的形式进行展示。

5 系统测试

为保证实盘训练营系统的可用性、稳定性，针对于系统的功能与性能进行了功能测试与性能测试。在功能测试中，严格对照系统功能需求说明文档，进行了单元测试、集成测试、系统测试以及验收测试。测试了每条行为路径、边界条件，以保证系统功能的完整性、可用性。为保证测试的合理性，我们选用用户常使用的主流硬件平台上进行了测试。

在测试中，将实盘训练营划分成了若干功能模块，模块划分参照了系统功能设计，再针对于不同的模块进行分别测验。当单元测试均通过后，进行回归主干的集成测试，寻找是否含有模块间的功能、逻辑、代码冲突，以验证系统整体的正确可执行性。

因系统采用 B/S 结构设计，故用户使用均在浏览器端进行。浏览器是用户访问本系统最核心的介质，起初 ActiveX 本是微软公司为自家浏览器产品 IE 浏览器设计的，Javascript 则是网景公司的产品。而不同的浏览器对于 plug-ins、ActiveX、Java、JavaScript 的支持规格与程度都不尽相同。这就导致了前端的最大难题兼容性问题。实盘训练营系统采用兼容性矩阵的测试方法，通过设置不同平台不同厂商不同浏览器版本以检测对不同浏览器的兼容情况。考虑到当今市场浏览器百花齐放，我们针对于市场占有率最高的几款主流浏览器进行了测试。同时对前端的功能进行了功能测试。其中包括链接测试，以测试 Web 系统中不同页面的跳转与页面存在性、可用性。系统通过表单测试以检验用户在填写表单信息时提交数据的完整性，避免信息不完整而导致后续的操作错误，同时在表单中，对于不同表格的边界、特定值、相互关联性进行测试。从而保证向服务器提交的表单是有效、完整的、正确的。用户在实盘训练营系统的操作记录与个人信息会储存于 Cookies 当中，所以对于 Cookies 也进行了测试以检查是否可正常工作。当实盘训练营系统被用户所使用的 Cookies 反问时，用户的个人信息会通过 Web 服务器进行发送，包括 Cookies 有效性、按与其时间保存正确性，刷新影响等^{[38][39]}。

最后考虑到系统设计是面对海量级用户的系统平台，我们进行了系统的性能测试，以考察系统在万级、十万级、百万级用户同时使用情况下，系统的负载及抗压

稳定性。其中包括了链接速度测试、负载测试、压力测试。速度链接测试即是用户在不同网络硬件环境、条件下对系统进行访问所耗时间的测试，访问速度取决于用户的网络类型、带宽^[40]。为保证系统的及时响应性、我们控制最大反应速度不超过 3000ms，常用功能的反应速度小于 500ms。负载测试即为测试该系统在同一时刻不同链接数量的反应时间以及不同链接的反应效果。系统测试步骤如图 5-1 所示。

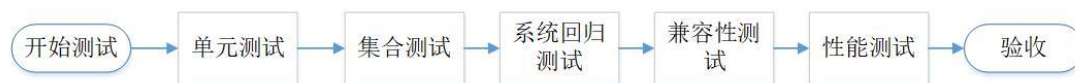


图 5-1 系统测试步骤

5.1 系统测试环境

为保证与真实用户的使用环境一致，我们选取了贴近用户实际使用环境的硬件条件与系统。

测试环境如表 5-1 所示。

表 5-1 测试环境

类型	说明
硬件	台式机采用 I5 6500、8G 内存、1TB 硬盘 笔记本采用 I5 6700HQ、4G 内存、500GB 硬盘
操作系统	Windows 7/8/8.1/10、MacOS High Sierra 10.13.4
浏览器	Windows IE10、Windows EDGE、Chrome60.0.0.0、Opera49.0、Safari11.0、Firefox 58.0
网络环境	ADSL 1000M/200M/100M/50M/10M

5.2 系统功能测试

在对系统的不同功能进行分类后，为每个功能编写了详细的测试用例。为保证测试的完整性，对系统的整个功能路径进行了覆盖^[41]。

考虑到完整系统测试用例过于复杂，下面对主要路径、功能的测试用例进行讲解。

(1) 用户个人账户管理功能测试用例

个人账户的管理包括了用户的注册、登录及个人账户操作。

注册功能主要测试用例如表 5-2 所示。

表 5-2 注册功能测试用例

系统版本	1.9.0			
功能模块	注册功能			
测试用例	输入数据	预期输入	返回结果	测试结果
用户昵称输入	输入字符 2-8 位之间，不包含敏感词。且与数据库中已存在的 ID 不同	输入字符 2-8 位之间，不包含敏感词，且与数据库中已存在的 ID 不同	昵称可用	通过测试
用户昵称输入	输入字符大于 8 位	输入字符 2-8 位之间，不包含敏感词，且与数据库中已存在的 ID 不同	字符应小于 8 位	通过测试
用户昵称输入	输入字符小于 2 位	输入字符 2-8 位之间，不包含敏感词，且与数据库中已存在的 ID 不同	字符应大于 2 位	通过测试
用户昵称输入	输入包含敏感字符	输入字符 2-8 位之间，不包含敏感词，且与数据库中已存在的 ID 不同	该昵称包含敏感字符	通过测试
用户昵称输入	输入已存在的名称	输入字符 2-8 位之间，不包含敏感词，且与数据库中已存在的 ID 不同	该昵称已存在	通过测试
性别输入	选择男/女/保密	选择男、女、保密，不可为空	通过	通过测试
性别输入	不选择	选择男、女、保密，不可为空	请选择性别	通过测试
生日输入	进行选择	不可为空	通过	通过测试
生日输入	不选择	不可为空	请选择	通过测试
密码输入	输入密码长度 8-16 位，包含英文字母，不含空格	密码长度为 8-16 位，需包含英文字母，不可包含空格	密码合格	通过测试
密码输入	输入密码长度 8 位以下或 16 位以上	密码长度为 8-16 位，需包含英文字母，不可包含空格	密码长度应在 8-16 位之间	通过测试
密码输入	输入密码不包含英文字母或包含空格	密码长度为 8-16 位，需包含英文字母，不可包含空格	密码需包含英文字母，不可含空格	通过测试
重复密码输入	与第一次输入密码一致	与第一次输入密码一致	通过	通过测试
重复密码输入	与第一次输入密码不一致	与第一次输入密码一致	与第一次输入的密码不一致	测试通过

(2) 个股功能测试用例

主要测试个股的检索、添加及删除，主要测试用例如表 5-3 所示。

表 5-3 个股功能测试用例

系统版本		1.9.0		
功能模块	个股功能			
测试用例	输入数据	预期输入	返回结果	测试结果
个股检索	正确股票名称/代码/拼音	输入股票名称/代码/拼音	返回个股	通过
个股检索	错误股票名称/代码/拼音	输入股票名称/代码/拼音	请输入正确的名称或代码	通过
添加个股进入自选	添加个股	添加个股	添加成功，进入自选池	通过
自选池中删除个股	删除个股	删除个股	个股删除成功	通过
自选股排序	按照涨幅排列	按照涨幅排列	按涨幅进行排列	通过
自选股排序	按照跌幅排列	按照跌幅排列	按跌幅进行排列	通过

(3) 交易功能测试用例

主要检测模拟股票交易的买入卖出逻辑正常运行，主要测试用例如表 5-4 所示。

表 5-4 交易功能测试用例

系统版本		1.9.0		
功能模块	交易功能			
测试用例	输入数据	预期输入	返回结果	测试结果
显示股票	输入股票代码/拼音/名称	在可交易时间段输入股票代码/拼音/名称，显示相对股票价格	显示相对股票价格	通过
买入股票	输入可买股数以内的数字，点击购买	在可交易时间段输入可买股数以内的数字，点击购买	购买成功	通过
买入股票	输入大于可买股数以内的数字，点击购买	在可交易时间段输入可买股数以内的数字，点击购买	资金不足购买	通过
买入股票怕	非交易时间进行买入	在可交易时间段输入可买股数以内的数字，点击购买	非交易时间不能购买	通过
卖出股票	交易时间内卖出股票	在交易时间内进行股票卖出	卖出成功	通过
卖出股票	非交易时间内卖出股票	在交易时间内进行股票卖出	非交易时间不可卖出股票	通过

(4) 京豆兑换功能测试用例

本模块主要检测盈利用户、亏损用户在兑换京豆时的系统运行逻辑正确性，主要测试用例如表 5-5 所示。

表 5-5 京豆兑换功能测试用例

系统版本	1.9.0			
功能模块	京豆兑换功能			
测试用例	输入数据	预期输入	返回结果	测试结果
京豆兑换	用户在盈利状态下兑换京豆	用户在盈利状态下进行京豆兑换	京豆兑换成功	通过
京豆兑换	用户在亏损状态下兑换京豆	用户在盈利状态下进行京豆兑换	弹出提示：只有盈利状态才可兑换京豆	通过

功能性测试共执行测试用例 381 个，覆盖了系统的全部路径，测试过程中共发现 bug46 个，其中影响功能逻辑的 bug21 个，影响用户体验的 bug25 个。经过 3 轮修复，所有 bug 得到全部修复。

5.3 非功能性测试

(1) 健壮性测试

为保证系统在各类严峻、不良、突发状态的应对处理能力，强化容错性，针对本系统的健壮性进行了测试。主要考察面对系统崩溃、浏览器崩溃、或玩过终端等特殊情况发生时，系统的应对。

主要测试用例如表 5-6 所示。

表 5-6 健壮性测试用例

系统版本	1.9.0	
异常条件	预期结果	测试结果
浏览器崩溃	用户信息得以保存，不会丢失数据。	通过
网络中断	用户信息得以保存不会丢失数据。	通过

（2）平台兼容性测试

考虑到系统为基于 B/S 端而设计，故用户可能使用不同的系统平台、不同的浏览器对系统进行访问操作。而不同系统、浏览器存对于页面的解析存在着差异，有时可能会影响到页面正常的功能交互，故针对于目前市场占有率较高的系统及浏览器进行了测试，如表 5-7 所示。

表 5-7 兼容性测试用例

浏览器	测试结果
IE10	CSS 水平翻转样式失效
EDGE	测试通过
Chrome60.0.0.0	测试通过
Opera49.0	测试通过
Safari11.0	测试通过
Firefox 58.0	测试通过

在测试中，发现了在 IE10 浏览器中存在页面样式解析失败的问题，通过对问题的定位，发现 IE10 不支持 filter 水平翻转滤镜，为解决 IE10 的样式解析问题，修改了水平翻转的表现形式，通过 transform:rotate 属性进行代替，问题得以解决。

（3）性能测试

因证券交易系统与其他传统的互联网系统最大的区别在于用户的使用时间段比较集中。实盘训练营为保持与真实交易环境的一致性，在交易时间上选取了与真实沪深市场交易时间同步的逻辑。故用户会集中于周一至周五的上午 9:30-11:30 与下午 13:00-15:00。用户行为较为集中，对于系统在高并发的情况下的稳定性、响应性有了更高的要求。故针对于海量用户同时在线的情况进行了性能测试。考虑到历史最大活跃用户数为约 100 万人，我们设定了高峰在线操作用户数量为 10 万，在此条件下进行测试。

在此我们选择股票模拟交易下单功能进行展示。首先通过 Virtual User Generator 建立测试脚本，并设置 1 万、5 万、10 万三个峰值用户区间^[42]。

测试结果如表 5-8 所示。

表 5-8 性能测试用例

并发用户数	10000	50000	100000
最小响应时间	1.24s	1.66s	2.11s
最大响应时间	1.45s	1.89s	2.90s
平均响应时间	1.29s	1.71s	2.17s
事务成功率	100.0%	99.6%	96.1%

通过上述测试我们可看到随着并发用户数的增高，平均响应时间也响应的提升，但提升幅度较小，并不影响用户的体验。在并发用户数超过 50000 时，事务成功率开始降低，但失败率依然较低，考虑到并发数量超越 50000 的情况较少，本次测试结果可以接受，满足实盘训练营的业务需求及系统设计目标。

5.4 本章小结

本章对于实盘训练营系统的测试环境、功能逻辑测试进行了核心功能与典型测试用例的介绍，并对系统的健壮性、兼容性、性能进行了测试。通过一系列测试杜绝了发现的 bug，验证了系统的可靠性。

6 总结与展望

6.1 全文总结

实盘训练营系统是通过历时 4 个月迭代而出产生的产品，在开发过程中遇到了无数的问题，通过系统的分析，追根溯源得到了解决。在实践中获得了巨大的技术提升，对于系统的关键需求、关键设计、关键实现方法等核心之处进行了细致讲解与介绍。目前系统已正式上线投放于市场中使用，服务了数以万计的股民用户，尤其是对于股票具有学习、认知需求的新手用户。

实盘训练营顺应时代的潮流，采取了基于 B/S 结构的设计理念。它具有易于维护和易于快速迭代的优点。相比于传统的 C/S 结构，它在维护时可将主要经历放在服务器端，帮助企业节省了后期维护的人力成本，更具有易用性、稳定性等优点。同时，它降低了用户的使用门槛，帮助企业更容易的获得了更多的新用户数。在需求设计方便，我们结合企业自身的优势，开创了盈利兑换京豆的模式亮点设计，大大激发了用户的使用热情。系统经过多轮的功能测试、性能测试，解决了若干潜在问题，最终得以成功上线。

实盘训练营系统严格遵循软件工程开发规范，并对于近年来新出的规范、技术进行了大胆的应用。秉着开创式的创新精神，大胆应用新技术、新平台，顺应了时代的发展趋势。

6.2 展望

实盘训练营自上线后，虽收获了众多用户的好评，也存在如下一些问题需要日后解决。

(1) 在需求设计阶段，对于用户提现逻辑欠缺细致的思考，收到了少部分用户因不能顺利提取京豆而产生的投诉，在下一阶段会对用户提现逻辑进行修改，设计奖金池提现模式，已达到用户可以按收益大小获得相对奖励的效果。

(2) 部分浏览器下，系统响应速度仍有提升空间。在 IE 浏览器下的行情展示

页面，切换不同的页面仍存在卡顿问题。考虑到 IE 浏览器仍存在大量用户群体，这一问题可能会影响到一定数量的用户，故下一版本会对 IE 兼容性问题产出更好的替代方案。

(3) 高并发数量下，事务成功率开始下滑。在 10000 活跃用户同时并发操作时，系统存在着少量的事务不成功概率。随着平台的推广与时间的推移，实盘训练营会迎来更多的新用户，那么服务器的高并发处理将会受到更大的挑战，故在下一阶段会对系统在压力下的处理成功率与速度进一步的优化。

致 谢

自 2016 年考入华中科技大学软件学院后，我在近 3 年的研究生阶段跟随我的导师吴涛教授学习软件工程理论与知识，并保持着重实践的学习习惯。在这三年里，我受益匪浅，让我更进一步的了解到了软件工程这一学科的博大精深与代码的美妙之处。

研究生生涯即将结束，我在吴涛老师的指导下，完成了本篇论文。我由衷地感谢吴涛老师多次细心的指导，以及孜孜不倦的教诲，同时感谢研究生阶段所有教授我知识、指导我人生方向的老师们，最后感谢平日里与我互帮互助的同学们，一起学习，一起进步的时光。

感谢各位老师评委对我的论文的悉心批评与指正！谢谢大家！

参考文献

- [1] 郁抒思, 周水庚, 关侗红. 软件工程数据挖掘研究进展. 计算机科学与探索, 2012(1): 1-31
- [2] 程勇, 秦丹, 杨光. 针对 JavaScript 浏览器兼容性的变异测试方法. 计算机应用, 2017, 11(4): 1143-1148
- [3] 代庆梅. 浅析 JavaScript MVC 框架在 Web 开发中的应用. 电脑知识与技术, 2014, 10: 2242-2245
- [4] 吴通, 陈雨亭. 基于动态分析的 JavaScript 代码推荐. 计算机工程, 2014, 12(10): 66-67
- [5] 杨国瑞, 张思博. 基于 AJAX 的 Web 应用架构设计. 现代电子技术, 2006, 23(15): 95-98
- [6] 周林, 步丰林. 嵌入式浏览器中 JavaScript 和 DOM 的支持. 计算机工程, 2004, 7: 114-117
- [7] 韩利峰, 李嘉曾, 吴丽梅. 基于 Nodejs 实现 Web 端实时在线监测. 仪器仪表用户, 2018, 2(10): 80-83
- [8] Ali Mesbah, Arie van Deursen. A component- and push-based architectural style for ajax applications. The Journal of Systems & Software, 2008, 81(12): 13-16
- [9] Ivan Drnasin, Mislav Grgić, Goran Gogić. JavaScript Access to DICOM Network and Objects in Web Browser. Journal of Digital Imaging, 2017, 30(5): 537-546
- [10] Todd Hawley. The Complete Idiot's Guide to HTML5 and CSS3. Technical Communication, 2012: 205-207
- [11] 布莱恩特. 高性能 HTML5. 北京: 电子工业出版社, 2014: 81-82
- [12] 威诺格拉德. 软件设计的艺术. 北京: 机械工业出版社, 2011: 162-169
- [13] 布鲁格. HTML5 与 CSS3 实战指南. 北京: 人民邮电出版社, 2011: 299-301
- [14] 古德斯特恩. 面向对象软件工程: 使用 UML、模式与 Java. 北京: 清华大学出版

- 社, 2011: 78-79
- [15] 李旭芳, 陈家琪. Web 应用中 XSS 攻击的分析和防御. 信息技术, 2014(11): 16-19
- [16] 丁茂华, 王海航. 股票信息系统的组成与应用. 计算机应用, 2008(2): 45-47
- [17] Chao Huang, Jin-Guan Lin. Metrika. Modified maximum spacings method for generalized extreme value distribution and applications in real data analysis, 2014(7): 23-25
- [18] Shaikh Y H, Mundae SV, Khan. Stock market data analysis using Rescaled Range(R/S)analysis technique. A R. International Journal of Engineering Research&Technology, 2014: 17-19
- [19] 姜晓冬. 基于 HTML5 的交互式股票分析系统的设计与实现. 天津: 天津大学图书馆, 2017
- [20] 胡翠华. 基于业务流程分析的移动证券产业价值链. 图书情报知识, 2006(3): 29-30
- [21] NicholasC. Zakas. Ajax 高级程序设计. 北京: 人民邮电出版社, 2006: 19-21
- [22] Chao Huang, Jin-Guan Lin. Modified maximum spacings method for generalized extreme value distribution and applications in real data analysis. Metrika, 2014(7): 29-30
- [23] Benjamin Appiah. SQL Injection Attack Detection Using Fingerprints and Pattern Matching Technique. The Institute of Electrical and Electronics Engineers, 2015(6): 29-30
- [24] Di Wu. The Creation of Interactive Three-Dimensional Microscope Based on VRML and JavaScript. 中国自动化学会智能自动化专业委员会: 中国自动化学会智能自动化专业委员会, 2013: 19-29
- [25] 李晓峰, 孙广中, 陈国良. 基于预测的 JavaScript 类型系统研究. 计算机研究与发展, 2012, 49(2): 421-431
- [26] 王世克, 吴集, 金士尧. Web 缓存技术概述. 计算机与信息技术, 2005(6): 19-21

- [27] Michael Stonebraker. SQL databases v. NoSQL databases. *Communications of the ACM*, 2010(4): 24-28
- [28] 熊慕舟. 一种基于 Radix 树的数据库前端缓存. *华中科技大学学报(自然科学版)*, 2013, 41(2): 205-208
- [29] 简玲. B/S 系统性能测试的设计与实现. *计算机工程*, 2009, 35(10): 51-53
- [30] 古玲, 苑志勇. 基于 B/S 结构的档案管理信息系统研究. *华中科技大学学报(自然科学版)*, 2005(1): 50-54
- [31] Wenzhi Zhu. Design of LAN Security Analysis System Based on B/S Architecture. Institute of Management Science and Industrial Engineering. *Proceedings of 2018 5th International Conference on Electrical & Electronics Engineering and Computer Science(ICEEECS 2018)*. Institute of Management Science and Industrial Engineering: 计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society), 2018: 24-26
- [32] 朱向明. 期权券商订单交易系统的设计与实现. 哈尔滨: 哈尔滨工业大学图书馆, 2016
- [33] 勒纳. *AngularJS 权威教程*. 北京: 人民邮电出版社, 2014: 187-211
- [34] Natasha Burns, Simi Kedia. The impact of performance-based compensation on misreporting. *Journal of Financial Economics*, 2005(1): 111-115
- [35] 李晓峰. 电子科技. Web 工程前端性能优化, 2015(5): 211-212
- [36] 吴贺. 前后端解耦模式及开发. *计算机系统应用*, 2017, 26(2): 217-221
- [37] 李海平. 高性能万兆网络迎接云计算应用. *网络世界*, 2012. 1. 16(15)
- [38] 冯细光. 开放 Web 服务测试平台的设计与实现. 中国图像图形学会多体委员会、中国计算机学会多媒体专业委员会, 2009: 16-17
- [39] 贺财平, 覃事刚, 刘建勋. Web 服务搜索引擎的设计与实现. *计算机应用与软件*, 2011(1): 56-57
- [40] 王亚楠, 吴华瑞, 黄锋. 高并发 Web 应用系统的性能优化分析与研究. *计算机工程与设计*, 2014, 35(8): 2976-2981

- [41] Quoc Huy Do, Richard Bubel, Reiner Hähnle. Automatic detection and demonstrator generation for information flow leaks in object-oriented programs. Computers & Security, 2016: 54-58
- [42] Peter Y. H. Wong, Richard Bubel, Frank S. Boer, Miguel. Testing abstract behavioral specifications. International Journal on Software Tools for Technology Transfer, 2015(1): 47-50