

大 连 民 族 大 学 本 科 毕 业 设 计（论 文）

基于 Laravel 的网络股评分析研究及应用

学 院： 计算机科学与工程学院

（系）专 业： 计算机科学与技术

学 生 姓 名： 杜宇森

学 号： 2018081510

指 导 教 师： 孟佳娜

评 阅 教 师： 王 巍

完 成 日 期： 2022 年 5 月 31 日

大连民族大学

摘 要

股票作为一种传统投资融资手段，目前国内证券交易规模已达世界第二，2021 年，沪、深股市累计成交 258 万亿元人民币，是普通民众最重要的投资理财手段之一。而随着新冠疫情的持续，互联网社区的极大兴起，越来越多的人选择在线进行证券交易与交流，因此对互联网社区讨论进行收集分析对整体市场分析有着至关重要的意义。

基于以上背景，本设计旨在实现一个基于 Laravel 框架的分析平台，平台针对国内上海证券交易所与深圳证券交易所挂牌交易的股票信息进行收集整理并展示，同时针对各个股票的相关讨论评价进行收集整理，将收集整理后的数据通过相关情感分析模型进行分析研究，通过网站系统展示分析结果以供为用户交易提供帮助。系统主体分为三大部分，即网站系统应用，数据服务与模型服务。其中，网站系统为核心，用于向用户提供包括用户账户相关系统，新闻时事汇总，股票自选关注，股票讨论汇总，股票评价分析结果等功能。数据服务为网站系统提供实时股票数据与股票评价数据，同时将数据保存为大数据数据库。模型服务则提供情感分析相关模型，LSTM 等算法能有效地解决数据输入相关问题，解决情感分析中的难点，支撑网站系统中分析功能。

通过网络股评分析平台，用户可以方便及时的获取关注股票的详细情况与动态信息，同时可以快速了解当下市场情绪，为下一步交易决策提供有力支持。

关键词：Laravel；网站系统；股票；情感分析

Research and Application of Online Stock Reviews Based on Laravel

Abstract

As a traditional means of investment and financing, China securities market is now the second largest market in the world. In 2021, the cumulative turnover of Shanghai and Shenzhen stock markets is 258 trillion RMB, making them one of the most important means of investment and financial management for the general public. And with the continuation of the COVID19 and the great rise of Internet, more and more people choose to trade and discuss online, so the analysis of Internet community discussions is of vital importance to the overall market analysis.

So, this design aims to implement a analyze system based on Laravel, which collects and shows information about stocks listed on the SSE and SZSE in China, and collects and organizes discussions for each stock, analyzes and researches the collected data through sentiment analysis models, and shows the results through the web system. This will help users in their decision making. The main body of the system is divided into three major parts, namely web system, data service and model service. Among them, the web system is the core, which is used to provide users with functions including user system, news collection, self-selected shares, stock discussion summary, analysis of stock discussion, etc. The data service provides real-time stock data and stock discussion for the web system, and saves the data as a large database. The modeling service provides sentiment analysis models. Algorithms such as LSTM can effectively solve data input related problems and address the difficult points in sentiment analysis, these can support the analysis functions in the Web system.

Through online stock reviews analyzes system, users can easily and timely get share's status for next decision in trade.

Key Words: Laravel; Web System; Stock; Sentiment Analysis

目 录

摘 要	I
Abstract	II
1 绪论	1
1.1 项目背景与意义	1
1.2 设计内容	2
1.2.1 数据收集	2
1.2.2 数据分析	2
1.2.3 交互系统	2
1.3 方案介绍与技术选型	3
1.3.1 方案介绍	3
1.3.2 技术选型	3
1.4 论文组织结构	4
2 系统需求与可行性分析	5
2.1 需求分析	5
2.1.1 功能性需求	5
2.1.2 非功能性需求	6
2.1.3 设计约束	7
2.2 可行性分析	8
2.2.1 技术可行性	8
2.2.2 经济可行性	10
2.2.3 风险控制可行性	11
3 系统总体设计	12
3.1 系统架构设计	12
3.1.1 系统支撑结构	12
3.1.2 系统部署方案	14
3.2 数据库设计	15
3.2.1 数据库设计分析	15

3.2.2 数据库物理模型	16
3.2.3 数据库迁移	17
4 系统详细设计	18
4.1 数据收集	18
4.1.1 数据来源	18
4.1.2 数据获取	18
4.1.3 框架设计	19
4.1.4 系统运行	19
4.2 数据分析	19
4.2.1 分析算法	19
4.2.2 模型训练	20
4.2.3 模型使用	24
4.3 交互系统	24
4.3.1 前端框架	24
4.3.2 后端框架	25
4.3.3 用户系统	25
4.3.4 股票系统	27
4.3.5 文件系统	30
5 安全与测试	31
5.1 网络安全	31
5.1.1 用户安全	31
5.1.2 攻击抵御	31
5.1.3 计算稳定性	33
5.2 系统健壮性	33
5.2.1 异常处理	33
5.2.2 数据库容灾	33
5.3 系统测试	34
5.3.1 单元测试	34

5.3.2 集成测试.....	34
5.3.3 测试自动化.....	34
结 论.....	35
参 考 文 献.....	36
致 谢.....	37

1 绪论

1.1 项目背景与意义

随着新冠疫情的爆发与肆虐，受到防疫政策影响，全球各国实体经济不同程度上受到了负面影响。市场经济流动阻滞，民众情绪倾向于降低消费水平，将流动资产转换为各类不动产，或进行一定程度上的投资。其中，证券市场作为准入门槛较低，规模体量庞大的投资手段，受到越来越多的人关注。2021 年，沪、深股市累计成交 258 万亿元，日均成交 1.1 万亿元，同比增长 24.7%；创业板累计成交 54.3 万亿元，同比增长 16.4%^[1]。由此可见，在疫情期间越来越多的人选择通过证券市场进行投资与交易。

而与此同时，由于“疫情熔断机制”，一旦发现疫情即停止非必要社会活动，使得居家办公与在线学习的需求大大增加。由于需求大幅度增长，在线会议等领域获得了极大的发展，这又对云计算等互联网基础行业产生了强力的正向推动作用^[2]。在互联网行业实现飞跃式发展的同时，越来越多的企业与相关行业收益。其中，论坛、社区等具有社交属性的软件受益匪浅，用户数量不断增加，越来越多的人选择使用网络社交代替线下社交。

正是在这两种因素的影响下，越来越多的人选择通过线上手段进行股票交易，而因为论坛式社区的极大发展，传统在线股票交易平台也纷纷加入社区功能，使得越来越多的人加入炒股，而股民又聚集在这些社区中讨论，形成了一种良性循环。仅新兴股票社区“雪球”目前就拥有 4500 万注册用户，月活数更是达到了夸张的 150 万。

众所周知，股票市场交易是较容易受到信息与情绪影响的。类似于挤兑，当市场信息趋向于负面，市场情绪偏向于悲观的时候，极易造成挤压性恐慌，2015 年中国股灾正是一个典型的体现。同样，当情绪趋向于正向，也会对相关股票产生正向影响，2021 年年初发生的黑天鹅事件“游戏驿站轧空事件”正验证了这样的规律，随着民众情绪波动，公司“游戏驿站”股价涨幅一度超过 190 倍，从最低每股 2.57 美元飙升至最高 483 美元。

因此，较为清晰准确地把握市场整体情绪有助于交易者更加深刻地理解市场运作，帮助我们制定下一步交易决策^[3]。

1.2 设计内容

基于以上背景，本项目设计一个综合股票系统。该系统由三大模块构成，即信息收集、信息分析与交互系统。

1.2.1 数据收集

为进行市场情感倾向分析，需要针对能够展现市场情绪的信息进行收集。这其中，用户在各个社区中的发言与交流作为重点进行采集，这些交流与发言信息称之为“网络股评”，整个系统均围绕着其构建。此外，还要对由官方发布的具有政策倾向性的信息进行收集，例如“降息降准”等“救市”政策。同时，还要对客观数据进行采集，其包括股票分时股价，公司信息等。在信息收集的过程中，要对被采集数据的用户保证隐私安全性，保证数据可用性。

1.2.2 数据分析

需要针对上述收集到的信息进行分析与研究。其中包括了以下三大部分。首先，需要针对获取到的信息进行简单处理，构建相应的模型，例如分词、标注等。第二，需要将初步处理后的数据进行分析，主要利用相关算法模型判断其信息的情感倾向性。最后，通过大量的股评数据与分析得出的情感倾向数据，构建市场整体情绪数据，为进一步系统构建打好基础^[4]。

1.2.3 交互系统

通过上述对信息进行的收集与分析，构建一个可供交互的网站系统。该系统对上述信息进行展示，其包括展示相应股票信息，展示收集到的相关评价与新闻，同时针对评价与新闻进行情感倾向的展示。最后，展示其整体倾向分布。除此之外，该系统还应拥有用户登录注册，查询相关股票，自选等功能以满足日常交易参考需要。同时，应预留二期开发接口，为后期添加论坛等其他功能在架构上打入基础。

1.3 方案介绍与技术选型

1.3.1 方案介绍

(1) 微服务架构

微服务架构是一种性能极高的分布式服务架构思想，其核心是将一个大型的整体架构程序进行拆分，拆分后的每个部分都拥有独立运行与横向拓展的能力。例如传统软件中某一功能需要与其他功能共同运行，传统分布式结构也是将整体系统在多台机器上同时运行。而在微服务架构中，登录功能可以只是整体服务中的一个子服务，该服务可同时运行多个，并且由负载中心决定一次请求由谁进行负责，同时其他服务不会因为某个特定的单独功能出现问题而整体崩溃。由于其没有限制的横向拓展能力与低耦合，微服务有着更高的稳定性和相应速度^[5]。因此，目前大多数公司在大型项目开发中更加倾向于使用微服务架构进行开发。

(2) 敏捷开发

敏捷开发是迭代开发技术的一种，迭代开发并不会事先通过完整文档约束整个项目的每个细节，只会指定大致方向，将总体计划拆分成不同小部分，通过不断对小部分进行开发并发布得到反馈后再进行下一部分开发，不断重复，迭代，最终形成完整产品。而敏捷开发在迭代开发中又加入了增量开发思想，即对小部分进行的划分是针对功能进行的，使得每次迭代后都会产生新的功能可供使用。

因此，在敏捷开发模式下，项目将会以最快的速度完成交付上线，而后通过快速迭代加入更多功能，可以极大降低成本，提高效率^[6]。

1.3.2 技术选型

(1) Laravel 框架

Laravel 框架是一个基于 PHP 语言开发的免费开源的后端框架，其旨在通过 PHP 语言实现 Web 软件的 MVC 框架，以编码优雅著称。

(2) Vue 框架

Vue 是一款基于 JavaScript 的用于创建前端界面的 MVVM 框架，是目前与 React 一起认为是最受欢迎的 JavaScript 框架。其在前后端分离开发，单页应用上有无法比拟的优势

(3) Python 与数据爬虫

爬虫技术是一种在万维网中自动获取信息的机器人技术，通过模拟人的访问行为获取相应数据。而 Python 作为一门语法简单、开源库众多的脚本语言，在数据抓取领域有着先天优势，因此大多数数据爬虫选择采用 Python 编写。

(4) TensorFlow 与 Keras

Keras 是一款使用 Python 语言进行编写的开源神经网络库，可以提供包括 CNN、RNN、LSTM 等神经网络的构建。TensorFlow 是一款由谷歌推出的开源机器学习库，提供了 Keras 相关操作接口。

(5) MySQL 与 Redis

MySQL 是由 MySQL AB 公司开发的一款免费开源的关系型数据库系统，后被甲骨文收购并开发成为商用数据库，目前是世界上使用最广泛的数据库系统之一。Redis 是用 C 语言进行开发的一款内存行数据库，一般用于对关系型数据库等进行数据缓存，常被与 MySQL 一同使用。

(6) Docker

容器技术是一种基于虚拟化思想设计的小颗粒度虚拟系统，容器之间不会产生互相影响导致全面崩溃，同时也能保证虚拟化宿主机的安全。Docker 是一款基于容器技术开发的开源软件，用户可以通过 Docker 打包自己的程序并运行。

1.4 论文组织结构

本文组织结构共分为五个大章。第一章主要介绍了本系统设计的初衷，背景以及其意义，同时简单介绍了系统的主要功能与技术方案。第二章对系统整体进行了需求分析与可行性分析，明确界定了系统需求，即每部分应该具有什么功能。可行性分析则针对功能进行了相关技术与经济方向上的分析，证明了本设计的可行性。第三章介绍了本设计的总体设计，其包括系统架构设计，运行环境设计，实际应用技术等内容，同时对其他系统整体不可拆分的部分进行了描述。第四章则更加详细地介绍了系统整体之下各个功能与模块间的联系与设计。本章与第三章共同构成了本篇论文的核心。最后一章第五章则介绍了部分开发与测试流程，展示了系统的健壮性和可控性，此外，还针对系统安全，网络设计等进行了更加进一步的讲解与演示。

2 系统需求与可行性分析

2.1 需求分析

2.1.1 功能性需求

(1) 数据收集

① 利用爬虫技术，实时地对中国大陆的上海证券交易所与深圳证券交易所的股票实时交易数据进行实时获取并进行分类保存。

② 利用爬虫技术，实时地对上述范围的股票的相应网络社区讨论进行实时与定时收集并进行存储。

③ 利用爬虫技术，定期将客观信息进行抓取，其包括每天新股上市信息，股票发行与其公司信息，相关财报、公告、新闻等，并对其进行存储。

④ 进行信息筛选，过滤掉明显无效的垃圾信息，同时对违反相关法律法规的信息进行筛选以减少相关风险。

(2) 数据分析

① 对信息手机系统收集到的网络股评数据利用词向量神经网络模型进行分词与词向量分析，以满足后续分析的需求。

② 对以上生成的词向量信息进行汇总整理，除去明显异常数据，将其他数据利用 LSTM 等模型进行训练，生成情感倾向分析。

③ 对大量数据进行情感倾向分析后，将大量数据进行汇总整理，通过统计图表的形式进行保存，以便后续系统使用。

(3) 交互系统

① 用户系统：其提供互联网用户的注册功能，注册应通过邮箱实现唯一验证。注册后应具有登录、找回密码、修改个人信息等用户系统功能。同时，用户系统应保证用户在使用期间的不间断登录。

② 股票系统：交互系统的核心系统，用户向用户展示股票信息。其应包括对当前股票热度的排行，对股票的搜索功能，对股票信息详情的展示功能，对股票讨论即股评的展示功能，对由信息分析系统分析出的数据进行展示的功能等。

③ 管理系统：提供管理员功能，管理员可以通过管理系统对用户进行管理，对日志进行分析等。

④ 文件系统：为整体系统提供文件服务，其包括处理用户上传的头像等图片文件，文档、附件等其他文件。

2.1.2 非功能性需求

(1) 开发模式

① 由于系统功能众多且庞杂，互相之间联系并不甚紧密，因此开发模式应该选择可以支持独立模块开发的形式。

② 开发周期短，交付部署速度快。

③ 由于系统整体性能消耗庞大，因此开发模式应满足便于对整体系统进行实时评估并拓展。

④ 应为后续持续长期开发打下基础。

(2) 架构需求

① 对于各个子系统模块间不应产生大量耦合，单独子系统模块下线维护不应导致系统整体崩溃。

② 各个子系统间通讯应该简单，不应过分依赖某单一系统进行系统间交互，子系统间应为分布式网状结构。

③ 架构需要满足较高的横向拓展性能以满足整体系统易于扩展，避免由于已有系统运行环境性能不满足导致系统整体响应缓慢。

④ 系统应便于部署，易于扩展开发，预留相关前置接口，以满足不断新加入的开发需求与后期功能的补全。

(3) 安全需求

① 系统应保证网络安全，杜绝包括 SQL 注入，XSS 攻击等常见基础攻击，同时应保证减少渗透攻击的可能，避免内网系统向外暴露，避免系统关键模块产生信息泄露，对外只应使用单一入口且对入口进行伪装与防护。

② 系统应保证数据安全，包括密码在内的数据必须进行满足国家信息安全标准的加密形式^[7]。

③ 系统应保证用户隐私安全，不对包括用户地理位置，浏览行为等信息进行收集，严格禁止能对用户信息产生暴露的分析行为。

④ 系统应声明爬虫协议，明确哪些信息可以进行抓取，哪些信息应被严格禁止抓取，同时应对被禁止的资源进行反抓取处理。

(4) 健壮性需求

① 系统应保证基础的异常处理体系，出现的异常不应对用户进行暴露，异常不应迷惑用户使用，应明确对用户进行提醒。

② 系统整体结构要求保证不能当单一服务发生故障时导致系统整体无法响应，系统关键模块出现故障时不应导致系统无法访问。

③ 服务应满足性能需求，当性能不足时应保证可以临时添加计算资源以满足系统性能要求。

2.1.3 设计约束

(1) 部署约束

① 系统部署环境应满足常规网站服务器需求，系统环境应保证系统运行稳定性安全性。

② 环境不应该过于依赖某一种或某一类特定程序，尽可能使用跨平台程序，降低开发成本与部署难度。

(2) 经济约束

① 要求系统尽可能使用开源代码进行开发，尽量减少商业代码或软件的使用，避免经济纠纷。

② 系统应最大限度利用现有资源，减少资源浪费，降低系统开发与运行成本，减少运维投入。

(3) 法律约束

① 要求系统不能包含任何违反相关法律法规的内容，尽可能规避可能产生的法律纠纷。

② 系统应受到法律保护，保证系统运行安全。

2.2 可行性分析

基于以上需求，设计提出以下可行性分析报告，其内容分为三大部分，即技术可行性，经济可行性与风险控制可行性。

2.2.1 技术可行性

(1) 开发技术

由于非功能性需求要求设计满足开发周期短，交付部署速度快，同时应满足足够扩展开发性能与后续兼容升级迭代方案，因此采用迭代开发形式。而设计整体应满足工作量需求，同时保证功能完整性，因此最终决定采用迭代开发中的敏捷开发进行开发。

(2) 架构技术

① 架构选择

本设计系统分为三大模块，每个模块下有拥有多个子系统，因此整体系统规模将会十分庞大且复杂，因此系统应该满足不同子系统与模块间不存在大规模耦合，应该减低各个子模块间通信难度，同时应该满足同时对不同模块开发的需求，因此最终架构设计技术采用微服务架构技术，能最大限度保证程序开发稳定与快速，同时其横向拓展性能也能保证系统在大规模需求性能的时候及时进行拓展。也为后期预留的迭代留下接口。

② 注册与发现中心

注册与发现中心时微服务架构的核心之一，所有构成系统的服务必须在服务注册与发现中心进行注册，通过注册中心互相发现，保证服务互相间可以定位。本设计采用 Nacos 作为注册与发现中心，通过私有化部署 Nacos 并通过相应 SDK 与 API 进行开发。

③ 通信框架

由于 Laravel 框架并未提供直接可用的微服务框架，因此需要利用 RPC 通讯框架自行进行微服务的开发。因此，选择 Thrift PRC 框架对 Laravel 进行二次开发，形成微服务框架。

④ 容器服务

各个服务应满足易于部署，方便扩展的要求，因此选用容器服务进行部署。由于服务器主要运行环境采用 Linux，因此采用其 Docker 开发相关容器服务。

(3) 数据收集技术

① 程序语言

数据收集技术选择利用爬虫对网络信息进行抓取，因此采用在网络爬虫领域较为成熟的 Python 语言进行程序的编写。

② 工具

数据采集工具采用 Python 语言的开源库程序 Requests 作为核心，利用其进行网页原始数据采集。同时，采用 BeautifulSoup4 与正则表达式对抓取到的原始数据进行解析并处理。在数据获取后，使用 Pandas 等工具进行进一步处理。

③ 系统结构

由于采用微服务架构，数据收集程序也会作为系统的一个服务运行。考虑到交互限制，采用 Flask 服务框架进行 RESTful API 接口的编写^[8]，构建结构调用结构，通过 HTTP 协议将数据收集功能向外部暴露。

(4) 数据分析技术

① 程序语言

由于 Python 语言拥有众多机器学习库程序，同时其语法简单，性能高，因此采用 Python 语言进行数据分析模型的开发。

② 算法技术

首先，需要对获取到的数据进行分词，因此采用 jieba 作为分词程序。在分词后，利用 word2vec 对词向量进行浅层神经网络的搭建，以方便后续深入的使用数据。

由于系统主要对股票评价的情感进行定性分析，因此采用在情感分析领域较为常用的 LSTM 长短期记忆神经网络算法进行模型的训练^[4]。采用 Python 语言的 TensorFlow 机器学习训练库调用 Keras 神经网络训练程序接口。

③ 模型使用

将上述流程训练得出的模型分通过 pickle 文件与 TensorFlow 的模型存储对模型进行保存。当使用时通过相应加载程序载入模型对输入数据进行计算得出分析结果。

同样需要将数据分析作为系统整体的一个子服务，通过 Flask 框架对 API 进行编写，通过编码调用模型，完成数据的输入与分析结果的返回。

(5) 交互系统技术

① 开发模式

全栈系统采用前后端分离模式进行开发，前后端相互之间通过 JSON 格式数据接口进行通信。

② 前端

前端采用 JavaScript 语言的 Vue 框架进行开发。

③ 后端

后端采用 PHP 语言的 Laravel 框架进行开发。

④ 数据库

数据库采用 MySQL，内存型数据库采用 Redis。

⑤ 拦截器与中间件

使用网关中间件对用户身份等进行验证^[10]，通过拦截器对非法请求与鉴权失败请求进行拦截并重定向。

(6) 网络技术

① 云计算

系统采用云计算服务器进行部署。

② CDN

系统将通过 CDN 对静态文件进行缓存，同时利用泛拨与 WAF 隐藏 IP 保证安全。

2.2.2 经济可行性

(1) 研发成本

研发部分成本来自算法设计与模型训练，集中在算法是否收费与训练环境搭建价格。算法采用开源框架，不收取费用。训练环境为支持 CUDA 的显卡运算环境，此前已购入英伟达 RTX3080 显卡用于训练，因此研发成本在可控范围内。

(2) 开发成本

开发成本来自于开发工具及相关框架，主要采用 JetBrains 出品的相关 IDE 进行开发，已拥有相关授权，成本可控。

(3) 运营成本

运营成本主要来自于服务器环境，已有服务器集群可供部署，成本可控。

2.2.3 风险控制可行性

(1) 经济风险

经济风险来自于收益成本比，本设计为开源项目无收益，而成本均在可控范围内，因此经济风险较低。

(2) 法律风险

设计符合包括民法典、知识产权保护法、个人隐私保护法等中华人民共和国相关法律法规。

(3) 社会风险

项目仅作为学习研究用途，不面向公众提供服务，无社会风险。

3 系统总体设计

3.1 系统架构设计

3.1.1 系统支撑结构

在大型规模的网站设计中，底层架构是决定系统上限的关键因素。一个好的底层架构设计可以保证后续开发过程中更少地受到复杂架构设计导致的混乱。同时，一个好的架构也是系统保持稳定运行，相应速度的关键。

在前文中，我们已经介绍了微服务架构的基本概念与为什么选择微服务架构。本节将更加深入地讲解本设计中包括微服务架构在内的系统整体设计原理与方案，如图 3.1 所示。

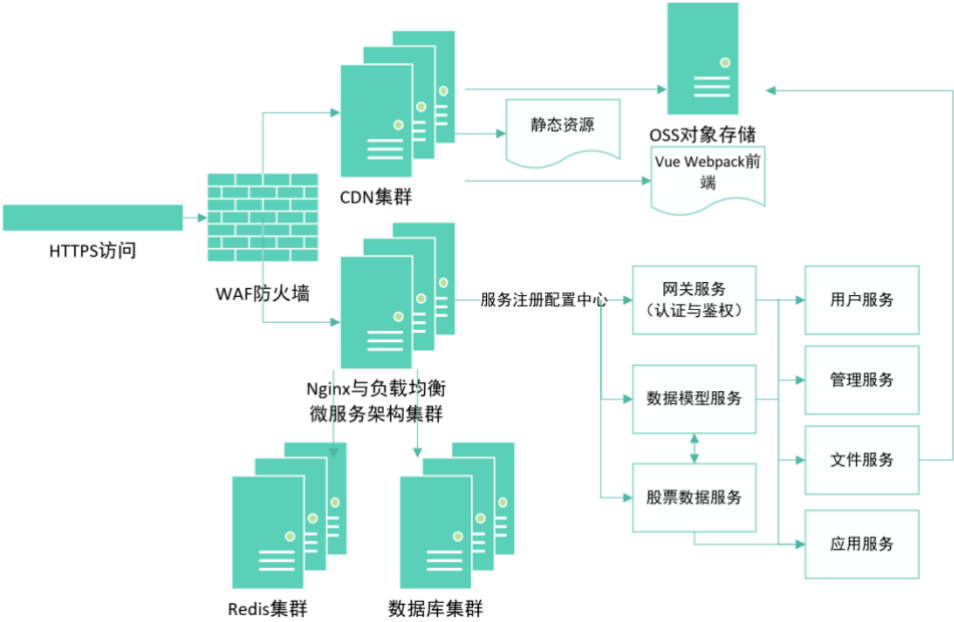


图 3.1 设计整体架构

图 3.1 展示了本设计的总体框架结构，其中，用户通过浏览器使用 HTTP 协议进行网站的访问，为了保证系统安全，所有请求均以 SSL 证书进行加密。在请求发生后，首先经由 WAF 服务将异常请求、攻击请求、机器人等请求进行拦截，保证后续系统集群安全。

而后，以前端、图片等为主的静态资源通过 CDN 进行存储，用户请求时通过泛拨域名就近访问 CDN 边缘节点获取对应资源。前端发出向后端的请求，后端请求由 Nginx 服务器进行接收。通过负载均衡顺序请求网关服务。

后端系统采用微服务架构，通过发现与注册中心将后端全部服务进行注册，Nginx 请求转发给网关服务后，由网关请求发现中心对相关服务进行发现并转发请求^[10]。其他服务事先已经进行注册，在使用时由发现中心随机选择健康实例进行返回。整体流程如图 3.2 所示。

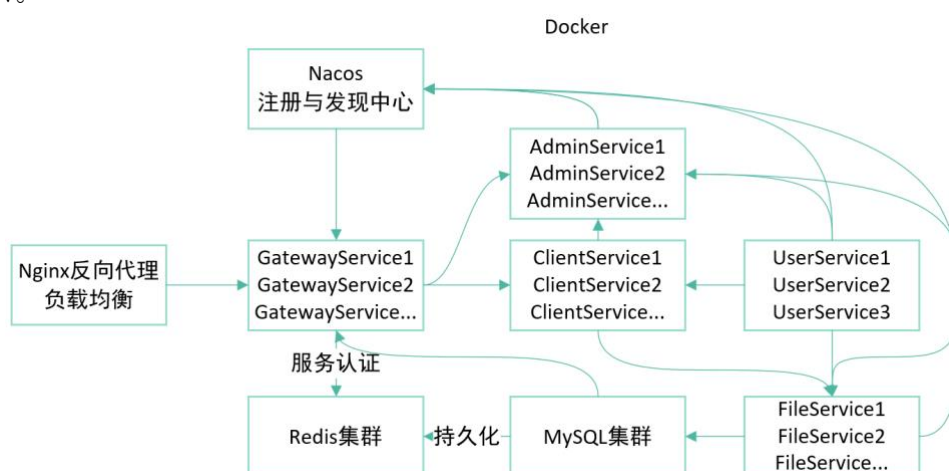


图 3.2 Nginx 与微服务访问流程

根据需求分析，服务主要分为三部分，即用于对数据进行采集的数据服务，对数据进行分析的模型服务，支撑交互系统的后端接口服务。

数据库采用 MySQL 云数据库服务集群，保证了数据安全性与可靠。同时，在系统中，采用 Redis 与文件驱动缓存对数据库数据进行本地缓存，降低数据库并发请求压力。

(1) 注册与发现中心

本设计采用本地化部署方式使用由阿里巴巴出品的微服务注册与发现中心 Nacos 对服务进行管理。私有化 Nacos 已部署于云服务器上可供访问，所有服务已完成注册，配置文件已经写入完毕。

(2) Nginx 负载均衡与反向代理

服务端采用 Nginx 作为 Web 服务器，利用其反向代理与负载均衡特性可以均衡向网关服务发送请求。同时，通过反向代理可以将不同部分服务通过同一域名进行访问，避免了 CORS 跨域资源共享问题。

(3) 服务间通信

服务间通信分为两部分，第一部分为数据采集与模型服务和后端服务间的通信，另一部分为后端各个服务间的通信。

数据采集与模型服务和后端服务间的通信由 RESTful API 接口实现，本部分将在第四章中进行介绍。而后端各个服务间的通信则由 Thrift RPC 框架构建，RPC 是一种服务端到客户端的模式协议，我们可以通过网关 API 对其他服务通过 RPC 请求进行分配调用，起到服务拆分的效果。

3.1.2 系统部署方案

由于系统采用多种语言与框架进行开发，为方便进行管理，全部服务与模块采用 Docker 进行部署。由于 Docker 以 Linux 系统作为虚拟底层，因此系统集群采用 Linux 系统集群。服务将分散部署在如图 3.3 所示的全部服务器上。

运行状态	节点名	类型	服务器位置	在线时间	负载	网络(B/s) ↓↑	流量(B) ↓↑	CPU	内存	硬盘
运行中	阿里云	KVM	中国 北京	414 天	0.3	12K 6K	180.89G 122.99G	5%	47%	41%
运行中	腾讯云	KVM	中国 北京	132 天	0	6K 1K	115.34G 69.21G	1%	68%	29%
运行中	天翼云	KVM	中国 北京	33 天	0.1	15K 4K	25.18G 4.69G	4%	25%	95%
运行中	家庭服务器	vSphere	中国 大连	26 天	0	134B 312B	27.95G 25.14G	5%	29%	7%
运行中	阿里云	KVM	中国 香港	145 天	0.8	1K 13K	161.44G 333.98G	5%	67%	63%
运行中	AWS	HVM	日本 东京	01:47:23	0	58B 300B	101.1M 3.2M	0%	2%	1%
运行中	搬瓦工	KVM	美国 洛杉矶	39 天	0.1	13K 14K	75.06G 75.64G	1%	49%	2%
运行中	BuyVM	OpenVZ	美国 拉斯维加斯	98 天	0	167B 435B	57.00G 68.10G	6%	47%	5%

图 3.3 服务器集群情况

服务器上对 Docker 环境进行部署，通过 Nginx 反向代理 Docker 服务对集群进行整合。

同时，为了保证开发效率与管理，所有代码使用 Git 进行管理，通过私有化部署 GitLab 进行保管与 CI 自动化流水线的设计。

3.2 数据库设计

进行大型架构系统的设计，数据库设计的优劣是系统后期开发与运行中稳定的关键，在本设计数据库设计过程中，分为以下步骤：确定数据来源与用途，绘制数据流图。进行约束与模型设计^[12]。通过物理模型逆向为数据库。通过现有数据库设计迁移脚本。

3.2.1 数据库设计分析

在设计中，数据库主要保存两类数据，即数据与模型服务所用到的股评与股票相关数据，和交互系统需要用到的包括用户、文件在内的多种数据。根据需求分析，我们可以绘制出顶层数据流图，如图 3.4 所示。

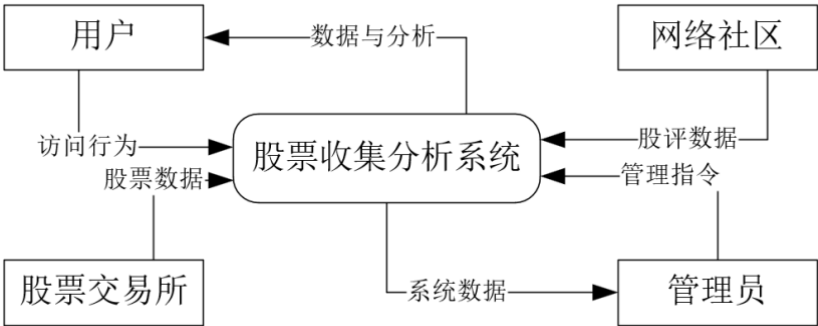


图 3.4 顶层数据流图

根据顶层数据流图，接下来继续进行对包括零层数据流图在内的各层数据流图的设计，如图 3.5 所示。

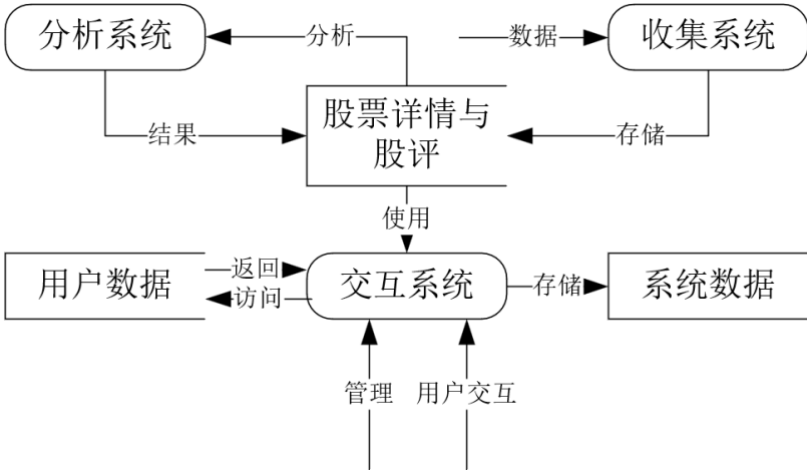


图 3.5 零层数据流图

3.2.2 数据库物理模型

在完成数据流图设计后，对数据库进行数据库建模。首先，根据数据流图进行数据表的概念模型设计，明确所需要的所有数据库表。然后，将概念模型进行补充，完善字段信息，设计字段间约束。最后，将所有字段的包括类型，长度等物理信息进行补全，形成数据库物理模型，如图 3.6 所示。

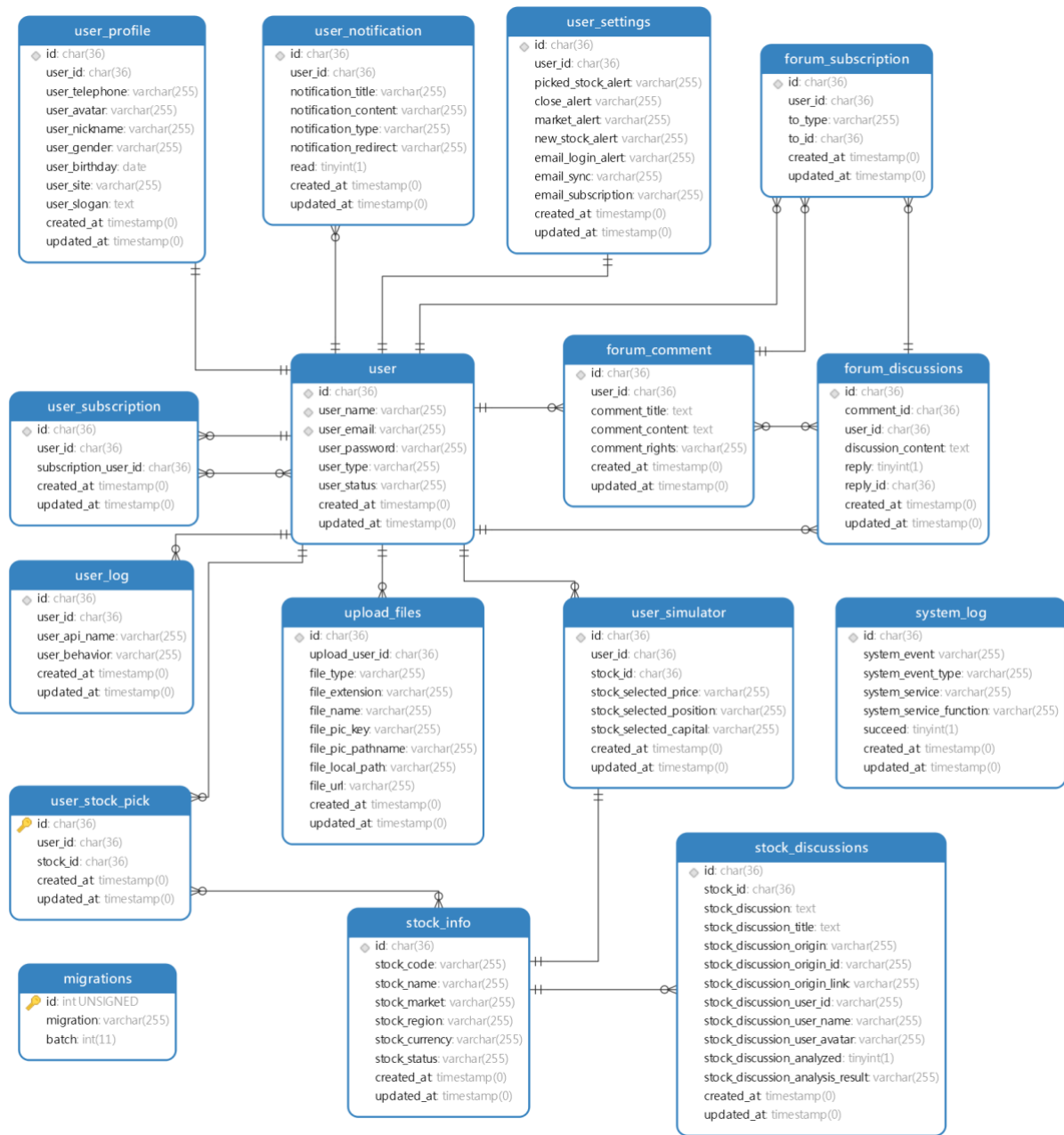


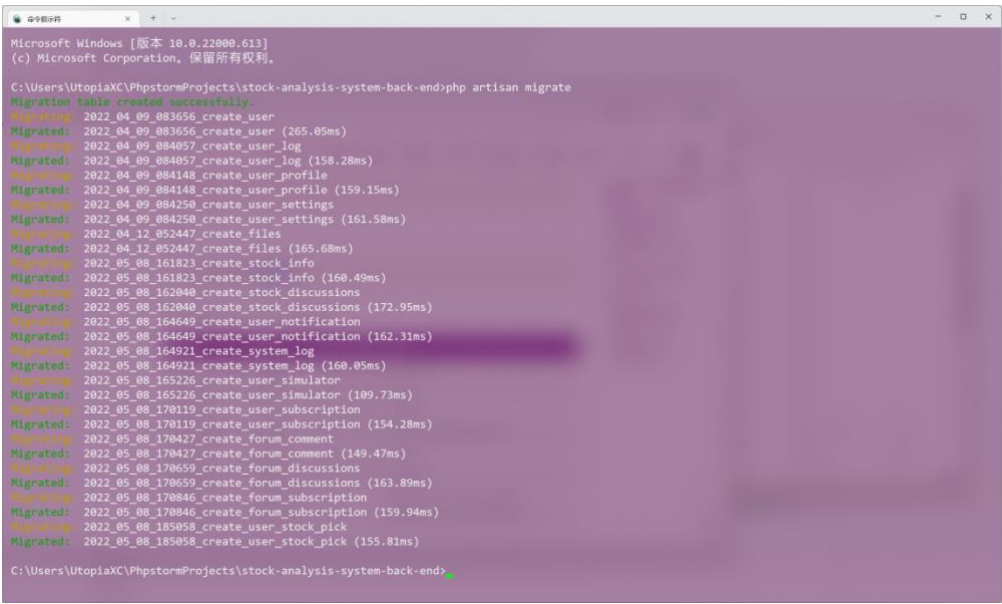
图 3.6 数据库物理模型

最后，通过物理模型对数据库进行构建，完成数据库设计与部署工作。

3.2.3 数据库迁移

完成了数据库部署工作后，为了方便后续开发重建测试与迁移，利用 Laravel 提供的数据库管理工具进行数据库迁移脚本的设计与开发。

通过迁移脚本，Laravel 可以通过脚手架程序自动创建数据表信息，并通过 seed 预先向数据库填充数据，配合 mock 数据生成使用，可为后续自动化测试打下基础。同时也是数据库在快照外的另一份结构安全保障。脚本运行如图 3.7 所示。



```
Microsoft Windows [版本 10.0.22000.613]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\UtopiaXC\PhppstormProjects\stock-analysis-system-back-end>php artisan migrate
Migration table created successfully.
Migrating: 2022_04_09_083656_create_user
Migrated: 2022_04_09_083656_create_user (265.05ms)
Migrating: 2022_04_09_084057_create_user_log
Migrated: 2022_04_09_084057_create_user_log (158.28ms)
Migrating: 2022_04_09_084148_create_user_profile
Migrated: 2022_04_09_084148_create_user_profile (159.15ms)
Migrating: 2022_04_09_084250_create_user_settings
Migrated: 2022_04_09_084250_create_user_settings (161.58ms)
Migrating: 2022_04_12_052447_create_files
Migrated: 2022_04_12_052447_create_files (165.68ms)
Migrating: 2022_05_08_161823_create_stock_info
Migrated: 2022_05_08_161823_create_stock_info (160.49ms)
Migrating: 2022_05_08_162040_create_stock_discussions
Migrated: 2022_05_08_162040_create_stock_discussions (172.95ms)
Migrating: 2022_05_08_164649_create_user_notification
Migrated: 2022_05_08_164649_create_user_notification (162.31ms)
Migrating: 2022_05_08_164921_create_system_log
Migrated: 2022_05_08_164921_create_system_log (160.05ms)
Migrating: 2022_05_08_165226_create_user_simulator
Migrated: 2022_05_08_165226_create_user_simulator (109.73ms)
Migrating: 2022_05_08_170119_create_user_subscription
Migrated: 2022_05_08_170119_create_user_subscription (154.28ms)
Migrating: 2022_05_08_170427_create_forum_comment
Migrated: 2022_05_08_170427_create_forum_comment (149.47ms)
Migrating: 2022_05_08_170659_create_forum_discussions
Migrated: 2022_05_08_170659_create_forum_discussions (163.89ms)
Migrating: 2022_05_08_170846_create_forum_subscription
Migrated: 2022_05_08_170846_create_forum_subscription (159.94ms)
Migrating: 2022_05_08_185058_create_user_stock_pick
Migrated: 2022_05_08_185058_create_user_stock_pick (155.81ms)

C:\Users\UtopiaXC\PhppstormProjects\stock-analysis-system-back-end>
```

图 3.7 进行数据库的迁移

4 系统详细设计

4.1 数据收集

4.1.1 数据来源

数据分为两大部分，一是用于对模型进行训练的训练集数据，二是用于分析展示的数据。首先，模型训练数据来自于微博数十万条评价数据，通过 `pandas` 工具进行管理。其中正向数据 50145 条，负向数据 49916 条。

股票数据来自于上海股票交易所与深圳股票交易所，股票社区讨论数据来自以雪球为主的多个网络社区平台，股票新闻数据与政策数据来自各公司信息披露与证监会。

4.1.2 数据获取

(1) 股票详情数据

通过 PHP 进行编写，在 Laravel 框架提供的 `seed` 数据库填充工具内对数据爬虫进行编写，通过脚手架指令进行运行。同时，脚手架支持计划任务，通过计划任务定时对全部数据进行抓取以获得实时股票数据与状态信息等。

(2) 股票评价数据

通过 Python 进行编写，通过 `Requests` 库向雪球、微博等论坛发送伪造请求并利用 `BeautifulSoup4` 框架对 HTML 档案进行解析^[13]，通过标签或 JSON 接口定位内容位置并解析为字典，通过 `MySQL` 进行存储并通过框架返回，如图 4.1 所示。

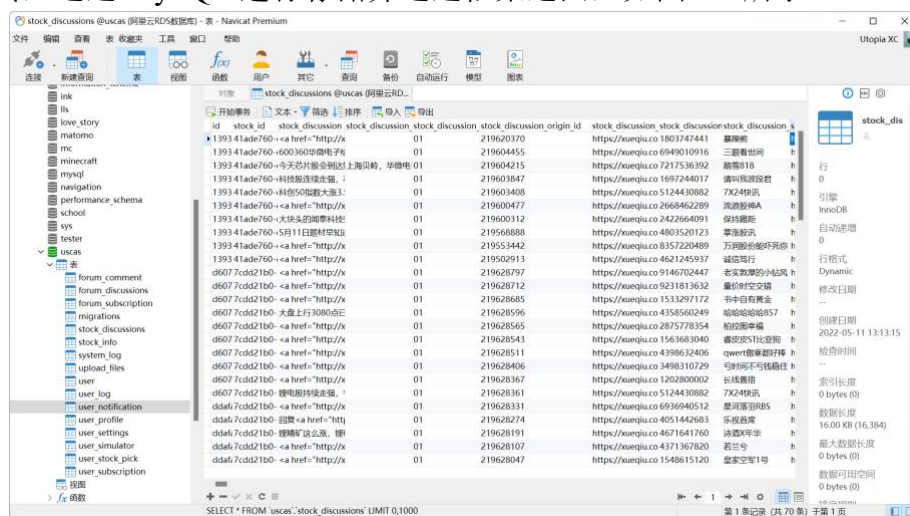


图 4.1 实时数据采集存储

4.1.3 框架设计

框架使用轻量级网站服务库 Flask 进行实现，利用其进行路由导航与 HTTP 服务，自行构建 MVC 模型结构，如图 4.2 所示。

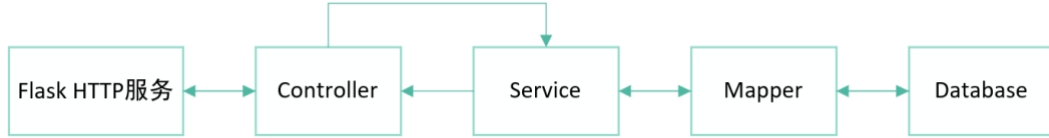


图 4.2 数据采集服务架构

通过服务架构，对每次用户请求的股评数据进行数据库存档，并通过 Flask 的 RESTful API 将数据返回，供其他系统使用。

4.1.4 系统运行

目前，数据服务提供接口包括股票详细信息、股票评论信息、实时每秒股价接口等，通过 Postman 进行测试验证系统运行可靠性。

4.2 数据分析

4.2.1 分析算法

系统主要目标是通过用户对用户情绪的定向分析进行市场概况展示，因此选用对情感向量分析较为友好的 LSTM 算法进行模型设计。

LSTM 全称为长短时记忆神经网络，是一种时间循环神经网络，也即在循环神经网络的基础上，在隐层的各神经单元上加入记忆单元，使时间序列上的记忆信息可以控制，使其更适用于处理和预测时间序列问题^[14]。

首先，LSTM 模型通过遗忘门忘记无用信息，其表述为

$$f_t = \sigma(W_f x_t + V_t h_{t-1} + b_t) \quad (4.1)$$

然后，输入门根据输入信息通过记忆细胞对长期记忆进行更新。

$$i_t = \sigma(W_i x_t + V_i h_{t-1} + b_i) \quad (4.2)$$

$$\hat{c}_t = \tan h(W_c x_t + V_c h_{t-1} + b_c) \quad (4.3)$$

$$c_t = i_t \hat{c}_t + f_t c_{t-1} \quad (4.4)$$

最后，通过输出门进行信息的输出，如图 4.3 所示。

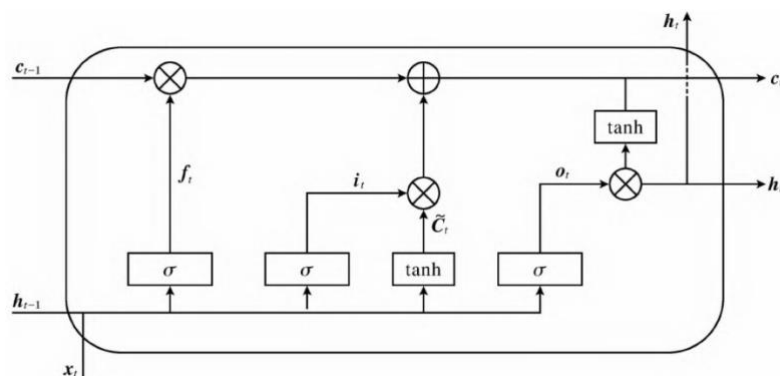


图 4.3 LSTM 算法工作流程

4.2.2 模型训练

(1) 数据处理与分词

首先，通过 Pandas 库将保存与 csv 格式的训练集进行载入，同时利用正则表达式等工具对数据进行预处理，将 HTML 格式与无实意词汇清理。然后将所有原始数据通过 jieba 库搜索引擎模式进行分词，以便后续进行词向量的而构建。

(2) 词向量模型

通过 word2vec 库构建浅层神经网络作为 LSTM 的输入层，通过训练集构建一个简单的词向量模型，如图 4.4 所示。将通过 jieba 分词库进行过分词的长句子转化为词向量，训练后通过 pkl 文件对模型进行保存^[15]。

① vector_size

即输出词向量空间的维数，维数越多向量包含信息越丰富，但是需要更大的数据集，在训练中，本设计将其设置为 300。

② min_count

即最低词频。当训练模型时，模型将会把低于最低词频的数据忽略，降低偶然性。本设计中，该参数设置为 10。

③ epochs

epochs 是一个超参数，其规定了整个网络对于训练集的训练次数，即每一个 epochs 都会利用模型将所有训练集训练一次，由于使用的是有限数据集，且采用梯度下降优化算法过程，因此数据在神经网络中传递一次并不足以支撑整个模型，我们需要将数据在

网络中多次传递，因此设置 epochs 参数。由于词向量模型为浅层网络，在此 epochs 设置为 20。

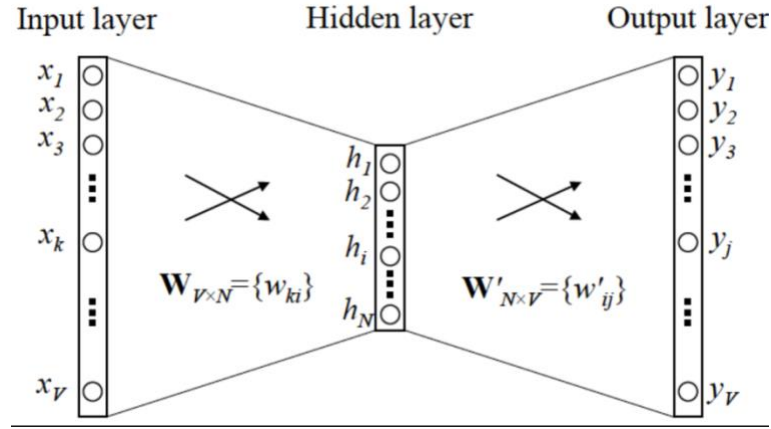


图 4.4 word2vec 网络原理

(3) LSTM 情感模型

完成词向量模型训练后，利用 sklearn，将数据集转换为可供训练的格式，并通过其工具划分为 80% 的训练集和 20% 的测试集。然后通过 Keras 构建神经网络。

① 嵌入层

Embedding，即嵌入层，一种特殊的全连接层。它是神经网络中的第一层，用来将词索引表示的窗口向量矩阵转换为各窗口向量矩阵，即将一个二维张量展开为三维张量进行输出。

其中，关注两个参数，input_dim 与 output_dim，分别表示输入张量维度与输出张量维度，本设计中，输入张量维度设置为 100000，输出维度设置为 100。

② 全连接层

Dasen 层是常用全连接层，全连接层每个神经节与上一层全部节点相连，用来将上层提取的特征进行整合。

因为本设计只对情感正负向进行分析，因此该层输出维度 units 设置为 2，激活函数采用在分类领域常用的归一指数函数 softmax。

③ LSTM 层

LSTM 层即长短期记忆神经网络层，是本模型的核心。本设计采用单层 LSTM 进行训练。

其中，本设计关注 `units` 与 `dropout`。`units` 参数为 LSTM 层的输出维度，本设计将其设置为与嵌入层一样的 100。`dropout` 为丢弃概率，LSTM 层中，为了防止训练数据与参数冲突，导致模型过拟合，过拟合体现在训练集中损失函数较小、准确率较高，而在测试集中损失函数大、准确率低。因此过拟合的模型基本不可用，为了避免这种情况发生，通过 `dropout` 对网络中的神经节进行随机删除，使整体模型不过分依赖局部特征。在本设计中，`dropout` 设置为 0.2，即训练中有 20% 的神经节会被删除。

表 4.1 模型训练参数

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 100)	10000000
lstm (LSTM)	(None, 100)	80400
dense (Dense)	(None, 2)	202

④ 损失函数

`loss`，即损失函数，用于评估模型预测值与真实值的不一致性，一般，损失函数越小，模型的稳定性就越好。一般，我们还会对 `val_loss`，即测试集损失进行分析，以保证前文提到的过拟合问题。

在本设计中，因为对情感进行二分类分析，因此采用最适合二分类的二元交叉熵函数作为损失函数。

⑤ 评价指标

本设计将准确率 `accuracy` 作为评价指标，准确率越高，模型拟程度越好。同样，本设计也关注测试集准确性，以保证模型不会过拟合。

⑥ 批尺寸

`batch_size`，即批尺寸。一般，当数据集较小时，进行全数据集训练，全数据集能更好的代表总体，而且由于不同权值梯度差别大，而全训练集可以通过 `Rprop` 只基于梯度符号且针对性地单独更新权值，有着更好的学习率。

但是，当训练集数据量十分庞大时，由于数据量的增加，对内存的开销也水涨船高，一次性载入全部数据变得不可行。而且通过 `Rprop` 方式进行迭代，会因为采样差异性，各梯度正直相互抵消，无法修正。

因此在大量数据的情况下，我们可以通过批梯度下降法选取一个合适的批次大小，通过分批训练保证训练成功。一般增大批次大小，内存利用率提高，完成单次 epoch 的迭代次数减小，处理速度更快。且其越大，其确定梯度下降方向越准，引起的训练震荡越小，如表 4.2 所示。但是盲目增加批大小也并不可取，会导致内存产生溢出，当迭代次数减少后希望达到相同精度所需的工作轮次增加，其参数修正更加缓慢^[16]。

⑦ epochs

本设计中，根据以上批大小与输入数据量进行分析，为达到较高的准确率与值相对可用的损失函数，将工作 epochs 参数设置为 40。批大小设置为 128，训练时间十小时。最后得出损失函数曲线如图 4.5 所示，最终损失值为 0.2795。

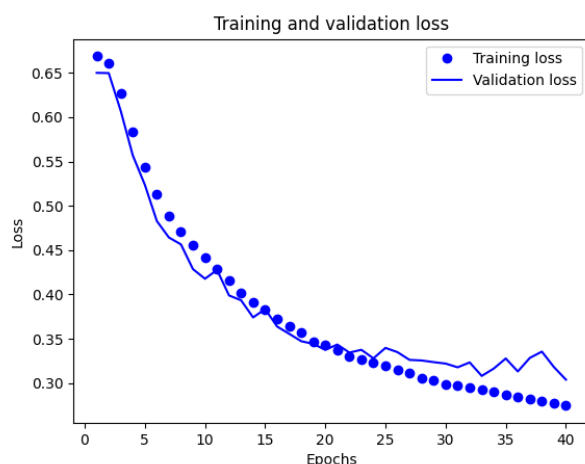


图 4.5 损失曲线

准确率指标为 0.8914，准确率变化曲线如图 4.6 所示。

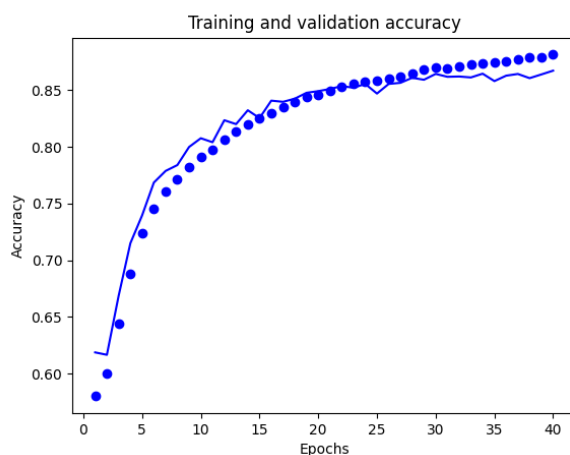


图 4.6 准确率曲线

4.2.3 模型使用

与数据采集相同，将训练后的模型通过 Flask 框搭建 RESTful API，通过 HTTP 接口向外暴露使用。

4.3 交互系统

4.3.1 前端框架

系统前端采用 Vue.js 框架进行开发，通过 Vue Route 路由进行单页应用开发，单页应用可以在不刷新浏览器的情况下对页面进行实时响应，减少网络请求占用的时间，同时对于 HTML 与 CSS 样式的编写更加友好，通过组件的形式对页面内容物进行开发，可以最大限度地进行复用，整体单页结构如图 4.7 所示。

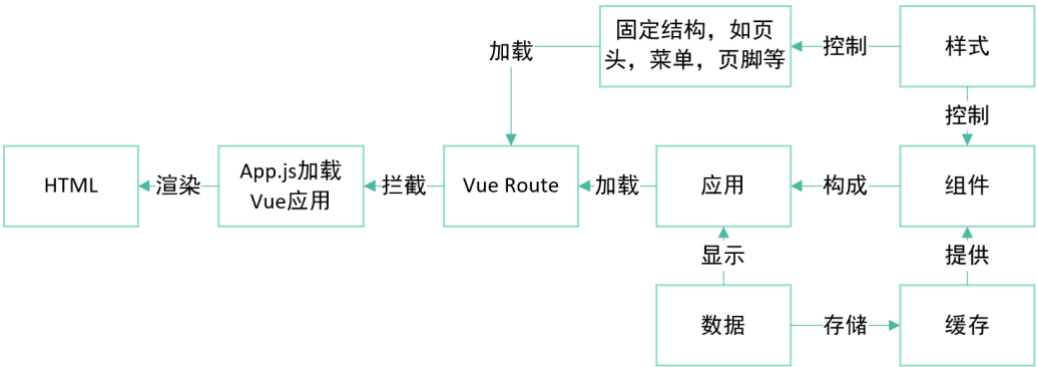


图 4.7 Vue 单页应用结构

同时，静态资源由对象存储，CDN 进行，保证前端相应反应迅速与可靠性，减少资源对源站的依赖性，其访问结构如图 4.8 所示。除此之外，前端完全实现响应式设计，保证用户访问在电脑与手机上效果一致。

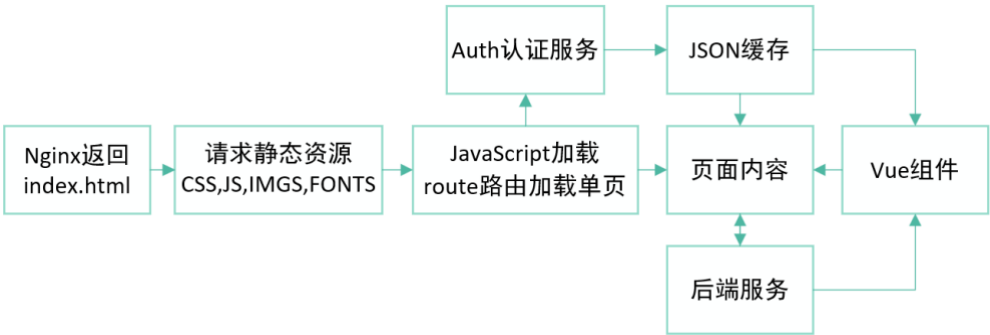


图 4.8 Vue 网络加载过程

4.3.2 后端框架

系统后端框架采用 Laravel 框架进行开发,通过 HTTP 协议接口响应系统前端请求,通过 ORM 数据库持久层框架 Eloquent 对数据库进行信息管理^[17],通过 Redis 对数据库数据进行响应缓存,利用定时任务完成系统级服务,整体框架如图 4.9 所示。

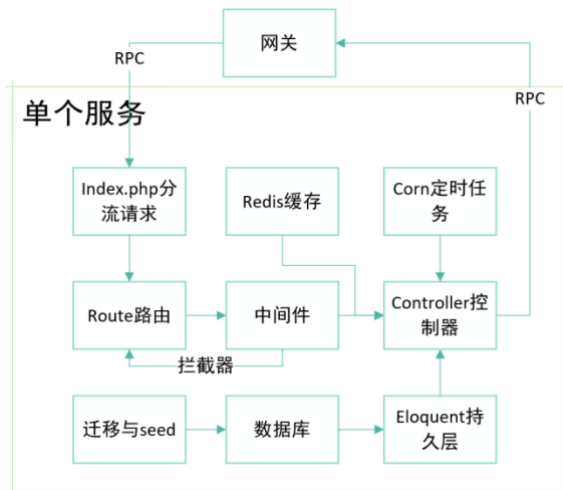


图 4.9 Laravel 框架原理

4.3.3 用户系统

用户系统主要由用户登陆与注册的验证模块、用户设置相关的配置模块、用户自选股等数据模块与用户个人主页等社区模块构成。

用户登录注册采用多种方式进行加密验证,保护用户安全同时保证用户登录后的使用体验,以下为部分页面截图。

(1) 用户登录

用户登录与注册界面主要提供用户填写信息的表单框,同时采用实时验证显示表单信息正确性,保证用户体验,登陆界面如图 4.10 所示。

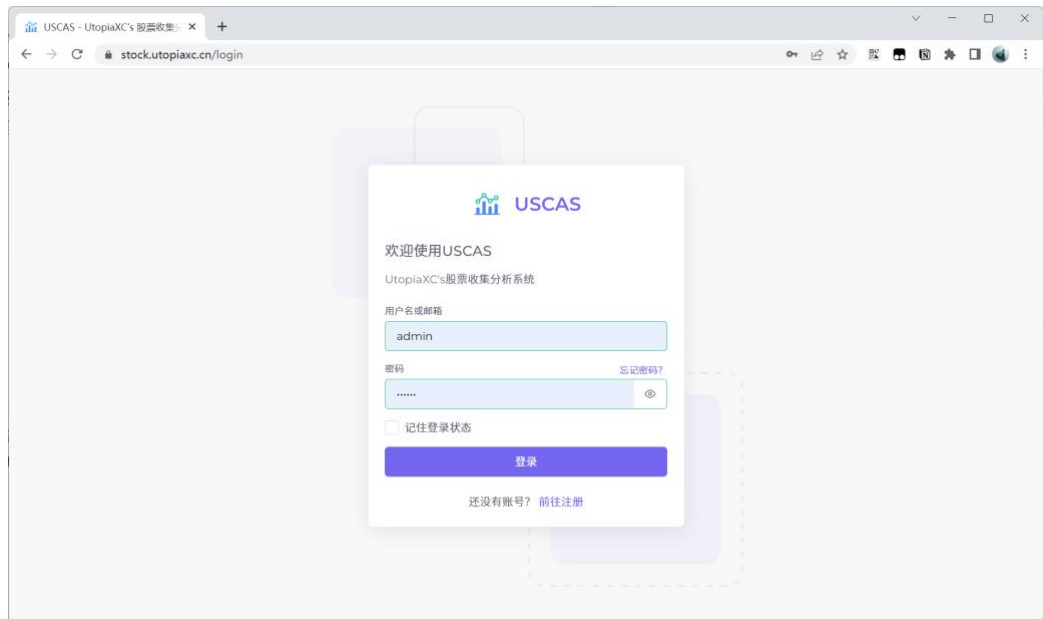


图 4.10 登录界面

(2) 验证邮件

用户注册或重置密码等敏感操作时，系统向用户发出验证码邮件验证身份，通过 HTML 模板规定邮件样式，如图 4.11 所示。



图 4.11 验证码邮件

(3) 用户设置

用户设置提供包括头像，信息，密码，通知与查看登录记录等功能，如图 4.12 所示。

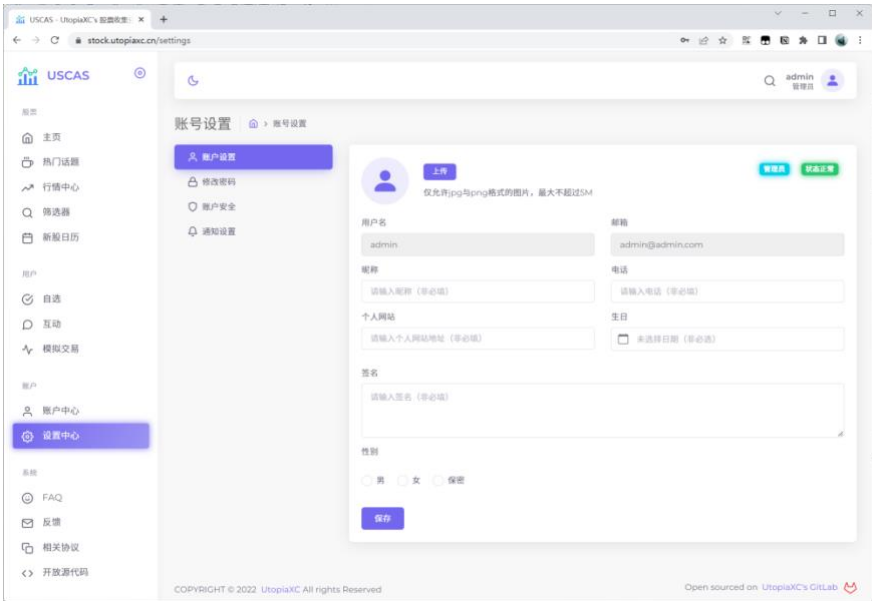


图 4.12 用户设置

4.3.4 股票系统

股票系统是交互系统的核心，其将数据采集、数据分析、用户服务融合为一体，为用户展现其希望获取的信息与数据，通过分析帮助用户制定下一步交易决策，因此此部分的设计与使用是本设计中最重要的一部分，以下展示部分实际系统运行界面。

(1) 主页

主页主要提供当前大盘整体信息，用户热评与排行榜信息，如图 4.13 所示。

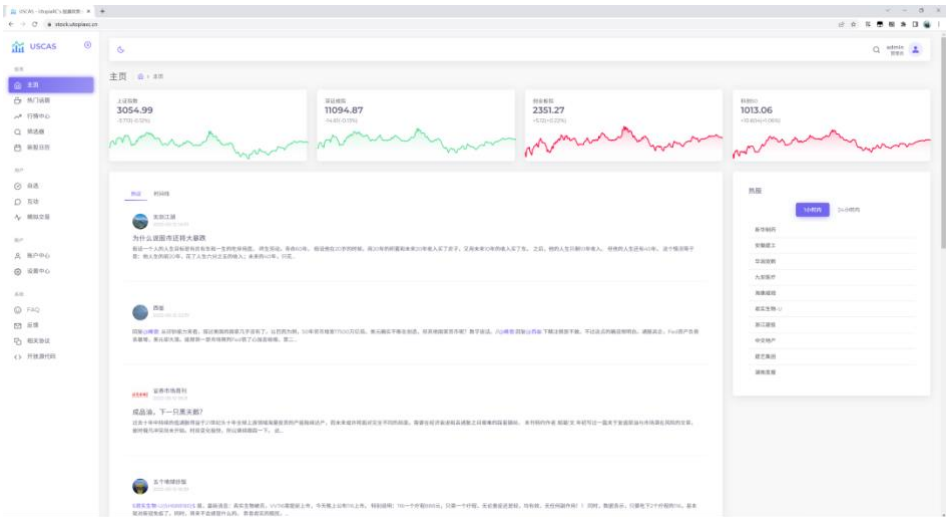


图 4.13 主页

(2) 股票详情页

股票详情页主要提供股票的相关信息、文件、K 线、实时动态、讨论及其分析与总体分析图表等信息，是整个股票系统的核心，如图 4.14 所示。

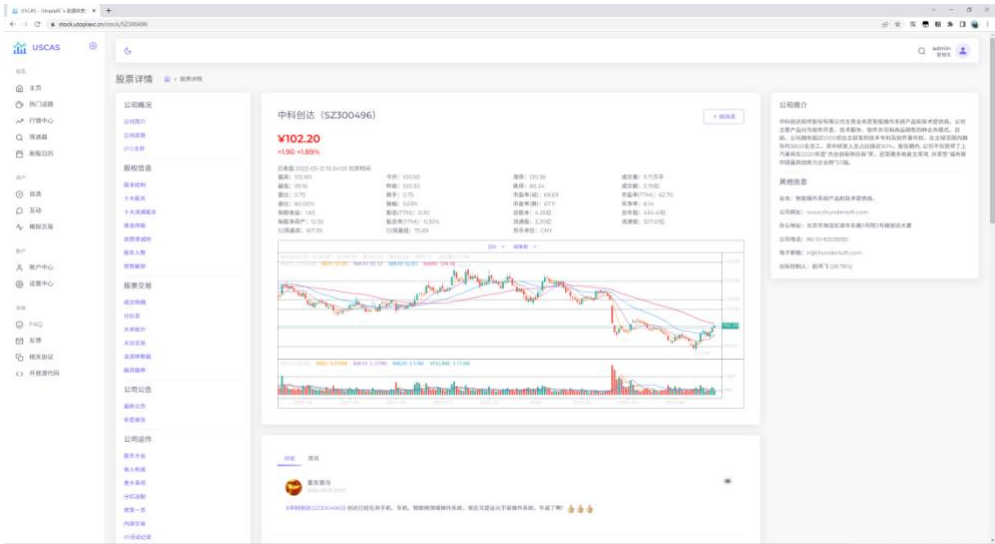


图 4.14 详情页

(3) 股评分析

股评分析通过调用模型获得单条评论的情感方向，如图 4.15 所示。



图 4.15 股评分析

(4) 分析汇总

分析汇总将近期多种数据进行总结，通过图表的方式展示该支股票的对应分析信息，折线图与饼状图展示了相关评价比例，购入评级展示了基于好评率的购买意见。如图 4.16 所示。



图 4.16 分析汇总

(5) 筛选器

用于通过特定条件进行相关股票的筛选, 用户可以根据交易市场、股票类型等信息, 通过关键词搜索等方式对所希望查找的股票进行相关的检索与筛选, 如图 4.17 所示。

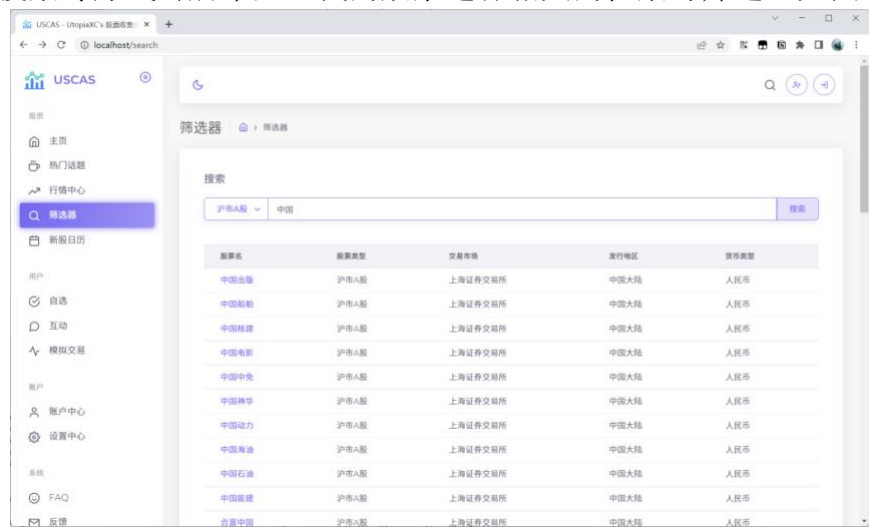


图 4.17 筛选器

(6) 新股日历

新股日历提供了新发行股票的相关信息，如图 4.18 所示。

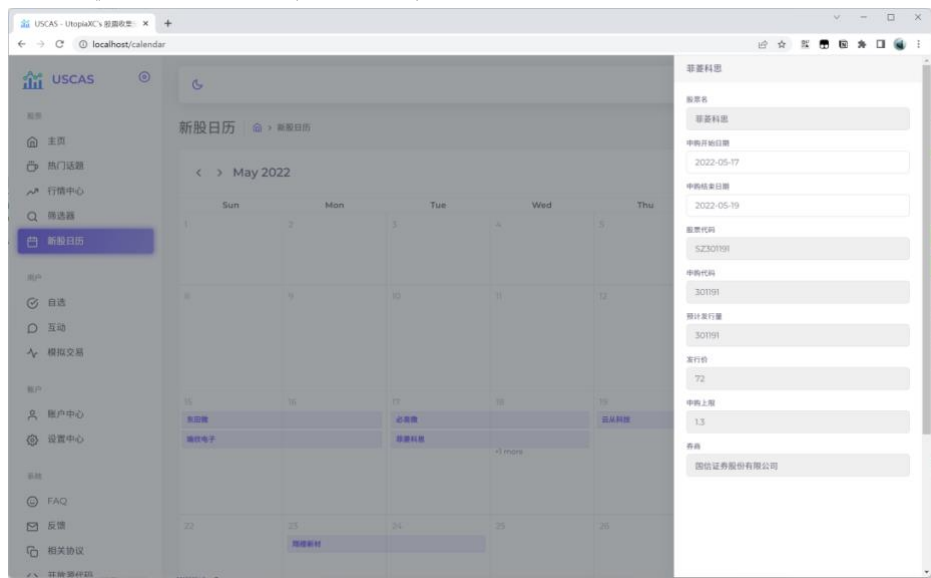


图 4.18 新股日历

4.3.5 文件系统

系统文件分为两个部分，一是用于存储由用户上传的图片，包括头像、发出的带图讨论等，二是用于存储类似持久化的信息文件，股票相关附件等信息。

图片系统采用 Laravel 开发文件系统进行管理，通过阿里云 OSS 进行保存，如图 4.19 所示。其他类型文件则由本地存储策略进行托管，由阿里云 DCDN 进行 CDN 缓存。

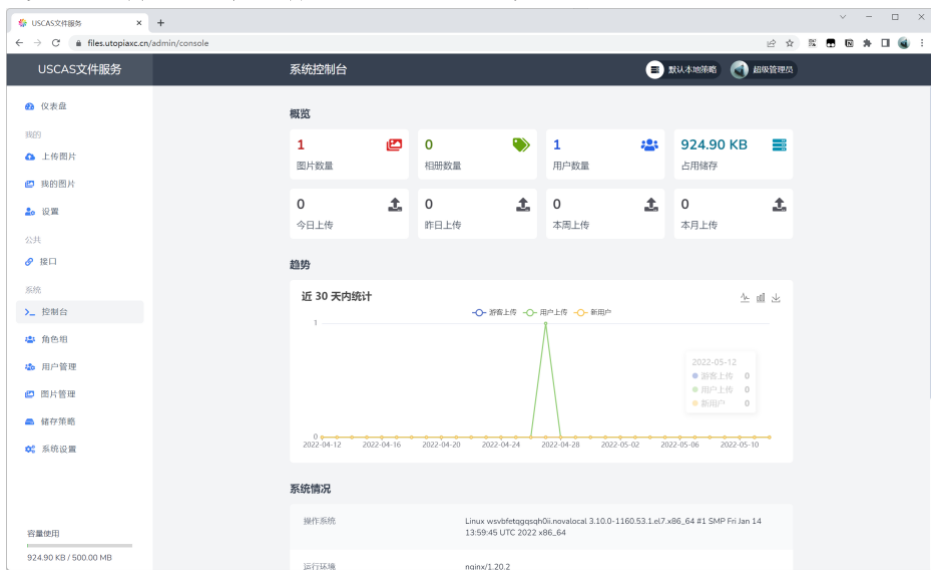


图 4.19 图片文件管理

5 安全与测试

5.1 网络安全

5.1.1 用户安全

安全性是一个计算机系统最为重要的一点，一个不安全的系统造成的危害十分巨大。比如此前的“华住集团用户信息泄露事件”，不仅对其本身造成了巨大的声誉损害，还对用户的隐私造成了侵害，因此，系统的安全性是保护双方的重要依仗。

本设计注重用户安全设计，所有敏感信息都将会被加密处理。如密码单向验证信息，设计并不对用户的密码进行原文保存，密码首先将被前端动态加盐 MD5 进行加密，通过 HTTPS 安全传输至服务器，然后采用 Bcrypt 散列算法对密码进行二次不可逆加密，保证了密码安全，哪怕数据库泄露也不会倒推出密码原文。

此外，包括 Cookie、Token 等信息均采用服务端 AES 加密后通过 JWT 二次加密保存，保证了交互数据的安全，同时通过动态生成的系统密钥实时对数据进行更新，保证了其泄露也无法被利用。

5.1.2 攻击抵御

除了设计本身的安全，系统也需要又一定的攻击防御能力，保证系统在大规模攻击下的稳定运行。

(1) 中间件拦截器

所有的请求都需要经过后端网关服务中间件，对其中可能存在的问题请求进行拦截。首先，中间件会检查每个请求所带有的 Token 信息，对用户身份进行验证，如果验证失败，则通知客户端清除 Token 数据并记录问题请求。

然后，检视请求的请求头，确认请求头是由真实客户端发出而并非伪造请求，如果发现请求存在伪造情况，则拒绝访问。

最后，中间件将检视所有提交的表单和跨站信息，如果表单中存在包括 XSS 攻击代码、php 远程执行代码、数据库注入等问题表单，进行表单清理并过滤，同时记录问题请求。

(2) CDN 泛拨

一般，通过域名对服务器 IP 进行解析后，服务器 IP 可以轻松通过 DNS 服务获取。然而，IP 暴露可能会产生一些问题，比如被端口扫描后爆破攻击。因此。隐藏 IP 对系统的安全性有一定的意义。

本设计中，采用阿里云 DCDN 技术，通过泛拨隐匿自身 IP，通过 CDN 中转进行源站访问，杜绝了 IP 外露问题。如图 5.1 所示。

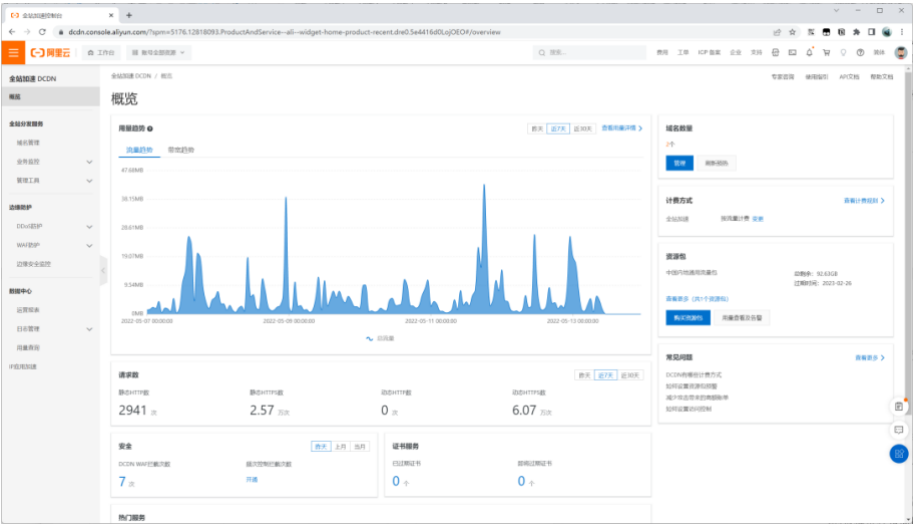


图 5.1 本设计 DCDN 配置

(3) WAF 防火墙

在网络安全中，除了系统进行攻击抵御外，同样需要在网络架构上对攻击进行防御。本设计采用阿里云 WAF 服务，过滤非法请求，保证系统安全性，如图 5.2 所示。

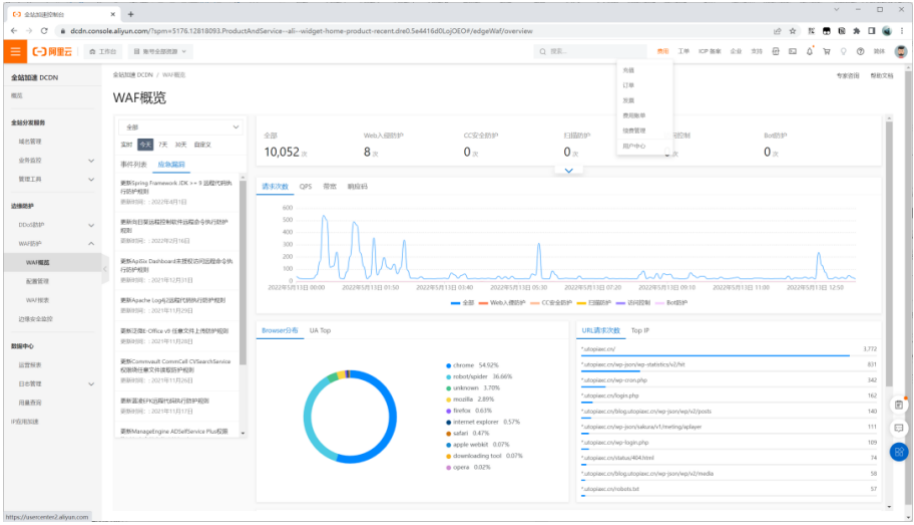


图 5.2 本设计 WAF 设置

5.1.3 计算稳定性

系统采用微服务分布式架构，保证了单一服务宕机不会影响整体服务。同时将来将引入服务治理机制，保证问题服务降级与熔断，在出现问题时第一时间切断出现问题的服务来源，保证系统稳定性。同时通过增加集群数量的方式横向对系统进行扩展。

5.2 系统健壮性

5.2.1 异常处理

(1) 异常拦截器

可能由于用户请求参数或其他某些原因，导致系统出现异常。一般，用户遇到系统异常的情况表现为页面无法加载或数据不正常等，这种情况对理解当下发生的情况有着极大的难度，对用户来说十分不友好，因此设计通过一个统一的拦截方式将异常抓取，当出现异常时，弹出 Toast 框向用户说明异常原因，为维护提供方便，同时提升用户访问体验。

(2) 内部异常的解决

在编码过程中，可能由于用户操作不当触发了某些限制条件，比如输入了错误的密码，系统在发生此类异常时，通过统一的返回方法类对异常进行返回并解释，本设计通过一个 R 类与 error 方法和一个枚举类对所有异常代码进行定义，保证用户访问时对此类异常能够清晰理解。

5.2.2 数据库容灾

对于一个大型系统来说，数据库就是大脑的存储神经中枢。因此，为了保证系统保存的信息安全，数据库必须进行容灾处理。数据库采用独立的云服务，不与其他物理服务器部署在一起，保证在其他系统故障时导致连带数据库一同宕机。

同时数据库指定定时任务，规定按照一定时间周期进行全量备份，并保存至其他位置，如图 5.3 所示。

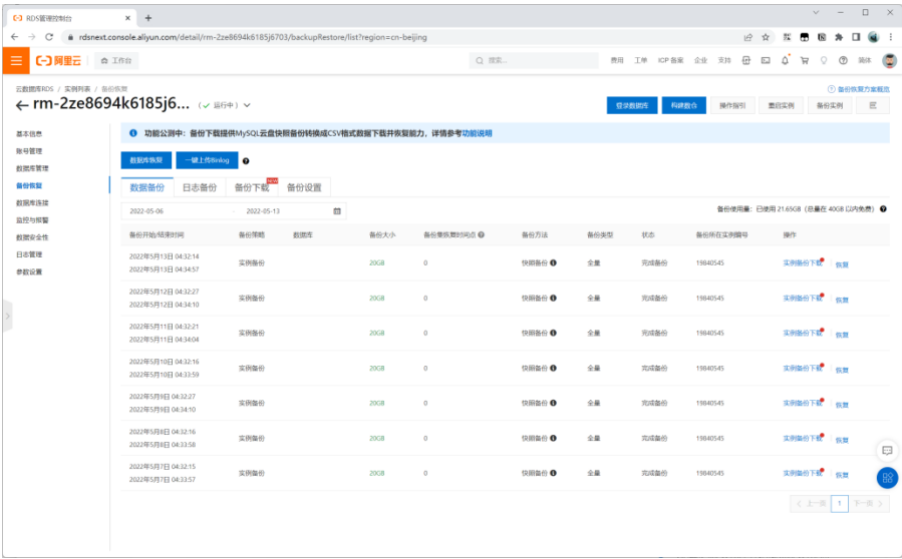


图 5.3 数据库备份

5.3 系统测试

5.3.1 单元测试

单元测试是一种模块化的测试手段，它的最小颗粒度可以以某个单独的方法作为标准，对其进行测试。在开发过程中，单元测试可以仅通过测试框架进行测试而不需要通过联合其他系统一同进行测试。因此，在开发过程中，好的单元测试可以有效降低系统开发过程中产生的漏洞^[18]。

在 Laravel 框架中，采用 Testing 框架对模块进行单元测试，帮助在开发过程中减少漏洞。

5.3.2 集成测试

在所有系统正式开发结束后，将对所有模块进行集成测试，即替换掉开发前期采用的 Mock 假数据接口，采用真是环境进行互相调用，此时一般也被称作为联合调试。当互相之间完成调用后，系统稳定，不再有新的漏洞出现，即认为集成测试成功。

5.3.3 测试自动化

测试中，测试数据的生成与接口的调用采用自动化测试的方式，利用 Python 的 Faker 库对数据进行生成，同时采用 Selenium 通过浏览器对接口进行访问，填充生成的数据，简化测试流程，保证测试进度。

结 论

本文介绍了一个基于 Laravel 设计的网络股评分析及应用系统。其系统包括数据收集、数据分析与交互系统三大部分。采用微服务架构设计，通过 Docker 在 Linux 系统进行部署。

数据服务通过 Python 爬虫技术提供了系统分析与展示需要的数据，数据分析服务通过包括 LSTM 在内的算法提供了对收集到的数据进行定向情感分析与汇总功能，最后通过基于 Laravel 后端、用户可注册的交互系统完成对信息的展示与交互。

通过本设计，用户可以更加准确地对当下股票交易市场进行把握，通过市场情绪分析了解当前市场倾向，以辅助用户进行下一步的交易决策。

此外，系统还有包括 K 线，时事新闻等功能，方便用户及时的获取股票相关信息，同时，因为网站架构稳定，响应速度较快，相比于竞品更加拥有优势。同时，系统还通过敏捷开发预留了包括论坛功能在内的大量后续迭代接口，方便更加进一步的后续开发。

系统的开发过程发现中国股市具有着高度的稳定性，制度优越性，和投资风险安全性。充分证明了我国经济发展的迅速，希望本设计今后能为中国市场经济贡献一份力。

参 考 文 献

- [1] 金彤.2021 年金融市场评述[J].中国金融,2022(03):83-84.
- [2] 蔡婷婷. 互联网行业新冠疫情前后经济市场分析 [J]. 商场现代化 ,2021(15):115-117.DOI:10.14013/j.cnki.scxdh.2021.15.043.
- [3] Wang H, Lu S, Zhao J. Aggregating multiple types of complex data in stock market prediction: A model-independent framework[J]. Knowledge-Based Systems, 2019, 164: 193-204.
- [4] Y. Yu, X. Si, C. Hu and J. Zhang, A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures, in Neural Computation, vol. 31, no. 7, pp[J]. 1235-1270, 2019, 31: 1235-1270.
- [5] O. Al-Debagy and P. Martinek, A Comparative Review of Microservices and Monolithic Architectures, 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), 2018, pp[C]. 000149-000154, doi: 10.1109/CINTI.2018.8928192.
- [6] Sharma, Sheetal, Darothi Sarkar and Divya Gupta. Agile Processes and Methodologies: A Conceptual Study[Z]. (2012).
- [7] ISO/IEC 19790:2012. Information technology — Security techniques — Security requirements for cryptographic modules[Z]. International Organization for Standardization.2015
- [8] 陈显军,耿强,纪洲鹏,湛永松.基于 Laravel 框架的 RESTful API 构建[J].信息与电脑(理论版),2017(15):42-44.
- [9] 余晓帆,朱丽青.基于 Flask 框架的社交网站数据爬取及分析[J].微型电脑应用,2022,38(03):9-12.
- [10] J T Zhao and S Y Jing and L Z Jiang. Management of API Gateway Based on Micro-service Architecture[J]. Journal of Physics: Conference Series, 2018,1087(3):032032-032032.
- [11] Ren Yu He. design and implementation of web based on laravel framework.Proceedings of the 2014 International Conference on Computer Science and Electronic Technology[C]. Amstelkade.Atlantis Press.2015.
- [12] 张坤,张云霞,孙全建.计算机软件数据库设计的原则及问题研究[J].电子技术与软件工程,2022(01):168-171.
- [13] 王钧. 基于深度学习的金融数据分析系统设计与实现[D].宁夏大学,2019.
- [14] 梁宇佳,宋东峰.基于 LSTM 和情感分析的股票预测 [J]. 科技与创新 ,2021(21):126-127.DOI:10.15913/j.cnki.kjycx.2021.21.053.
- [15] 柴源. 基于 Word2vec 和 SVM 的在线图书评论情感识别系统实现 [J]. 电子设计工程,2022,30(06):179-183.DOI:10.14022/j.issn1674-6236.2022.06.039.
- [16] 梁毅,丁振兴,赵昱,刘明洁,潘勇,金翊.一种面向分布式深度学习系统的资源及批尺寸协同配置方法[J].计算机学报,2022,45(02):302-316.
- [17] 杜佳. 对象/关系映射技术在信息系统开发中的应用研究[D].西安电子科技大学,2011.
- [18] 杨晨. 软件自动化测试方法的分析及应用 [J]. 现代工业经济和信息化 ,2022,12(01):167-168+171.DOI:10.16525/j.cnki.14-1362/n.2022.01.058.

致 谢

岁月不居，时节如流。毕业季春风一夜拂过人生，大学生涯如期落下帷幕。感怀时光之不羁，岁月之如光，值此之际，沉吟感惜。

其首，需感激孟佳娜教授，自本设计开题，迄论文截稿，孟教授为我与我的设计提供了至关重要的帮助。无论设计选题、文献翻译抑或论文撰写，孟教授一向给予中肯之建议，教导技术之原理。感之不弃，最终得以完成毕业设计，摘取学士之位。

次之，需感谢王巍老师在中期检查中给予本设计的建议，耐心教导，不胜其烦。

此外，向吴佩泽同学表示感谢，感谢其在本设计进行时对算法与流程上的帮助。

大学时光已然逝去，只增恋嫀。风景犹美，但路途不会止于脚下。抖擞精神，迈步前行，愿终有一天重聚首，意气仍风发。