# 需求分析说明书

# 1 引言

## 1.1编写目的

本文将给出本系统系统的基本需求和基本的数据流程图，要实现整个系统，需要对用户的需求进行设计，概要设计主要是利用比较抽象爱你个的语言对整个需求进行概括，确定对系统的物理配置，确定整个系统的处理流程和系统的数据结构，接口设计，实现对系统的初步设计。本文的预期读者为对本项目有兴趣的人以及程序设计人员。

## 1.2背景介绍

如今社会十分注重投资，股票投资作为一种经典并且广泛的投资方式被大部分人所接受，其中一部分的股民希望通过更加规律性的事物对于股票的涨跌进行预判，并得到能够长期稳定盈利的方式，而本系统则是为该部分人群制作，希望能够通过本系统的提供的数据，对大部分股票进行正确的预判。另一功能则是为股民提供一些信息的整合与汇总，本功能为次要功能。

|  |  |
| --- | --- |
| 系统名称 |  |
| 系统提出者 |  |
| 开发者 |  |
| 主要用户 |  |
| 硬件平台 |  |

## 1.3定义

Spring Boot：Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。

MongoDB: 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似json的bson格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是他支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

Ubuntu：Ubuntu（友帮拓、优般图、乌班图）是一个以[桌面](https://baike.baidu.com/item/%E6%A1%8C%E9%9D%A2" \t "_blank)应用为主的开源GNU/Linux操作系统，Ubuntu 是基于[Debian](https://baike.baidu.com/item/Debian" \t "_blank)GNU/[Linux](https://baike.baidu.com/item/Linux)，支持x86、amd64（即x64）和[ppc](https://baike.baidu.com/item/ppc/150" \t "_blank)架构，由全球化的专业开发团队（Canonical Ltd）打造的。

Java：[Java](https://baike.baidu.com/item/Java/85979" \t "_blank)是一门[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1" \t "_blank)编程语言，不仅吸收了[C++](https://baike.baidu.com/item/C%2B%2B)语言的各种优点，还摒弃了C++里难以理解的[多继承](https://baike.baidu.com/item/%E5%A4%9A%E7%BB%A7%E6%89%BF" \t "_blank)、[指针](https://baike.baidu.com/item/%E6%8C%87%E9%92%88)等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。

Docker：Docker 是一个[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90/246339" \t "_blank)的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 [Linux](https://baike.baidu.com/item/Linux) 机器上，也可以实现[虚拟化](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E5%8C%96" \t "_blank)。容器是完全使用[沙箱](https://baike.baidu.com/item/%E6%B2%99%E7%AE%B1/393318)机制，相互之间不会有任何接口。

IDEA编译器：IDEA 全称IntelliJ IDEA，是java语言开发的集成环境，IntelliJ在业界被公认为最好的java开发工具之一， IDEA是JetBrains公司的产品，总部位于捷克共和国的首都布拉格

## 1.4参考资料

# 2平台需求分析

## 2.1社会需求

随着商场经济的逐渐深入，现代人对于资产的投资理念提高到了一个新的高度，而股票作为一种重要的投资手段，被人们所广泛接受和使用，电视节目与网络节目也都有大量的以“通过股票的形式盈利”的内容，这也从侧面反映出了，股民对于股市的期待以及对于能够通过一些途径获得盈利信息的心态。

下图为“上海证券交易所”2017年至2012年概况数据（数据来源为上交所<http://www.sse.com.cn/market/stockdata/overview/yearly/>）

我们可以明显的看到股票市场处2015年牛市以外均保持一定幅度的上升趋势，股票的总价值上升，可以从侧面体现股票市场的盈利存在一定的规律。

而希望知道或找出这个规律的人也自然而然的应运而生，再结合时代互联网经济的特征，我们处理数据的手段也不断提升，使用科学技术对于现有数据进行分析并找出规律，这是一个十分通用的方式。着同样也是本程序被设计的初衷与想法。

## 2.2系统目标与愿景

本系统的主要目的是开发一款能够比较准确预计股票走势，并且提供大部分股票数据的后台项目。

对于用户可以通过调用接口获取部分股票的历史数据，并且获得股票的预测信息。

根据本项目的计划，系统主要分成以下两部分

1. 数据爬取与展现
2. 数据分析与预测

主要目标有以下几点

1. 数据抓取：

股票历史信息

上市公司企业信息

股票指数信息

股票其他必要信息

1. 数据展示

调用接口展示JSON格式数据

1. 数据分析

使用部分技术对于已获得的数据进行分析，分析预测股票走势（调用接口）

## 2.3系统面向用户类别和特性

系统主要面向从事股票交易的个人投资者，即为股民，倾力投资国内股票市场，希望通过股票交易些许赢利以养家糊口，但不得不或长或短地担任股东，在交易大厅中曾经汗流浃背，曾经瑟瑟发抖，被社会另眼视为赌徒，被大股东视为小猪，且屡受绝望煎熬，无力自救，只能寄希望于政府那样的一个弱势人群。

其特点主要为：具有[分散性](https://baike.baidu.com/item/%E5%88%86%E6%95%A3%E6%80%A7/7479876)和流动性的特点。并且不论男女老少，不分文化职业，只要有购买能力都可以参与。因为门槛低，也导致了大部分用户不具备专业的投资水平和更多的投资时间，这也是大部分股民赔钱的根本性原因。所以，要想赚钱必须：1、需要花费更多的时间和精力；2、提高自身的水平；3、寻找外界的帮助。4。懂政策技术和指标。

对于这些人群希望通过所谓的技术手段来进行股票的盈利，然而还没有找到比较正确的途径与方法，本系统将服务该部分人群。

## 2.4运行环境

### 2.4.1硬件平台

|  |  |  |
| --- | --- | --- |
| 客户端 | | |
| 序号 | 硬件 | 相关组成描述 |
| 1 | PC端客户机 | 能够发送http请求，具备一定的json格式解析能力 |

|  |  |  |
| --- | --- | --- |
| 服务器端 | | |
| 序号 | 硬件 | 相关组成描述 |
| 1 | 路由器 | 可以稳定发送信息和连接网络的设备 |
| 2 | 服务器 | 暂时进行考虑中，使用云平台或直接使用家用电脑作为主要平台，最低配置要求为：英特尔至强系列或i3-8100或以上，内存8GB以上，硬盘空闲1GB以上，带宽1Mb/s以上 |

### 2.4.2 软件平台

1、Docker：容器（进阶目标）

2、MongoDB: 数据库

3、JDK1.8或OpenJDK1.8以上

## 2.5外部接口需求

### 2.5.1第三方股票查询接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口描述 | 所需服务 | 数据和控制信息 | 通信定义 | 需求来源 |
| 网易股票接口 | 查询股票信息 | 所有需要读写的数据 | 第三方接口 | 需要获取股票的历史数据与部分的参数 |

## 2.6功能需求

### 2.6.1查询股票历史数据

返回数据必须包含股票的当日基本信息，能够根据日期进行必要的查询，参考sina股票接口数据。

### 2.6.2查询上市公司的基本信息

返回数据必须包含该上市公司的基本信息，其中包含（行业板块信息，公司所在地等与法律相关信息，以及股票分析所需的基本信息，具体请参见雪球公司信息）

### 2.6.3查询其他技术指数信息

返回数据包含股民希望知道的一些指数技术信息以及一部分的网络信息（评论数，关注度等等）

### 2.6.4预测信息

返回数据需要包括提供的预测信息，和建议购买的股票编号与推荐程度，用户可以通过结合自身的经验与本程序提供的各种建议与推荐程度，进行一个有选择性的投资。

并且对于该接口希望能够获取不同算法下的推荐值，或者根据不同的算法进行的结果整合也是需要的。

### 2.6.5业务场景

假设现在有一名无丰富股票基础知识与投资经验的用户A，一位刚刚进入市场，懂得一些股票知识，但是并不熟悉股票的真正运作方式，并且没有过的多的股票投资经验的用户B，以及一位有多年股票投资经验并且对于股票有很深的了解无论从政策方面还是经济方面对于个股有着自身的理解的用户C。以上三位用户都希望通过股票来进行合理的投资是自己的财富或者生活更加美好或有趣。

对于A用户，一位对股市没有过多了解的人来说，希望得到一款系统，能够直接的向这名用户指明那些股票是值得购买的，而这些股票中最推荐购买的又是哪一种，对于这样的完全新手型的股票购买者来说程序需要极其的简单，简单到可以直接不需要通过任何思考就能获利，并提供一个比较便利的途径来达成目的，他只需要一个接口就是最终结果。

对于B用户，一位对于股票有一定的了解，但是无法总是做出正确的决定的投资者而言，他希望调用一个预测接口，并且对于股票添加一些简单的判断，他会调用多种算法实现的不同的股票预测结果，然后再对于每一个股票调用，指数型信息，来实现自己的简单判断，然后在自己的脑海或其他方式中总结出，哪一支股票可以帮助到他，也就是，他需要的简单指数信息和不同算法下的预测结果信息与推荐程度。

对于C用户而言，作为一个资深的股民，他需要一个程序来帮助他缩小范围，能够更加精准快速的找出具有良好潜力的股票，来节约大量的时间，同时他不需要太多的基础技术型指标来帮助他进行判断，他会打开其他比较熟悉的软件进行额外的判断。

### 2.6.6功能需求

优先级：5为最高，1为最低，数值越大将会越优先完成

|  |  |  |
| --- | --- | --- |
| 用户 | | |
| 需求名称 | 优先级 | 备注 |
| 查询股票历史数据 | 3 |  |
| 查询上市公司相关数据 | 3 |  |
| 查询其他技术指标信息 | 2 |  |
| 查询推荐购买的股票 | 5 | 需要使用多种模型进行推理啊 |

|  |  |  |
| --- | --- | --- |
| 管理员 | | |
| 需求名称 | 优先级 | 备注 |
| 爬去信息接口 | 5 | 需要爬去各种信息，并且完成信息的存储 |

### 2.6.7非功能性需求

#### A 性能需求

1、数据精准度

时间精准到天

由于部分股票价格剥夺比较薄弱，使用股票价格精准至分

2、并发特性

在网络稳定并且服务器达到标准时，可以承受大于1000的并发量并且不产生丢包或数据丢失等情况

3、故障处理特性

保证在每十五秒为周期检查接口活跃度，如果接口为非活跃状态或假死状态则停止服务并自动重新启动

#### B 安全措施需求

1、程序将会拥有一套完整的日志体系，可以打印出所有的错误发生和时间点，并且通过接口活跃度检查，将会进行邮件通知的报警行为。

2、 应用日志必须按日保存，每个文件保存应用一天的完整日志；

3、 历史应用日志必须压缩后按日保存，以减小历史日志文件对硬盘空间的占用；

4、 历史应用日志必须保留一个月（30天），以保证问题出现后，日志可以得到较好的保存。

#### C安全需求

系统要有足够的防御能力，防御普通黑客的攻击，尽量减少各种可能出现的漏洞，防止用户信息被不良分子窃取。

保证用户权限的正确设置，防止出现各种越权行为，保证系统的正常良好运行。  用户的信息保存采用国际流行的MD5加密，保证安全。

管理员接口将进行简单的鉴权（输入一串加密字符串，并进行解析鉴权）

#### D 操作需求

系统操作者多为非专业人士，所以必须提供直观的图形界面，使能够快速上手使用。

#### E 质量需求

易用性优于易学性，或者可移植性优于有效性。

## 2.7软件需求规格表

|  |  |  |  |
| --- | --- | --- | --- |
| 软件需求表 | | | |
| 子系统 | 模块内容 | 对应接口 | 功能说明 |
| 日志与测试系统 | 日志分析  自动化测试 | 日志分析接口  自动化测试检测接口 | 获取与分析日志内容，以最快的速度对日志进行分析，并且对错误日志进行汇总并且提交邮件，同时进行接口活跃度测试，实时监控接口活跃度达到良好的用户体验 |
| 安全系统 | 基本鉴权 | 包装型的鉴权接口 | 对于管理员的请求进行轻量级的鉴权服务，以此来抵挡误输入与部分的攻击行为，并提供给数据以基本的安全体系 |
| 核心算法预测系统 | 股票推荐与分数排名 | 各算法对应的推荐接口 | 本程序的核心算法与核心需求，对于本部分的接口我们使用不同的算法进行分析，以此来保证准确性。 |
| 爬虫系统 | 将存在于网络的上的数据接入本地或远程数据库 | 网易爬虫接口  百度爬虫接口  雪球爬虫接口  东方财富爬虫接口  新浪爬虫接口  官方爬虫接口 | 对于不同的数据我们使用不同的数据源来进行良好的爬去，对于一些特定的数据源我们也将使用特定的爬虫以此获取，这个系统也同样是本程序的核心算法之一，因为所有的数据分析都是建立在已有数据的基础上的 |
| 数据展现系统 | 将数据库包含的数据展示给用户 | 历史记录接口  技术指标接口  公司信息接口 | 主要提供给第三方或股民用户其他的评判条件 |

# 概要设计说明书

# 1、引言

本部分内容将会简介的告诉开发者与读者，我们的程序将如何运作，如何设计，层次以及包含的大致内容。在阅读完本章节的内容后，我们希望读者能够自行想象出一套大致的框架，在之后的详细设计中我们才会详细的告诉你这一套框架中的具体内容与实现方法，为了能够使读者更加清晰的了解到我们表达的概念，我们会使用图形，表格等方式进行说明。

在阅读本篇章之前，希望读者能够快速的或详细的阅读上文的需求分析，在阅读完本篇章之后，我们会对总体结构有一定的解释与说明，我们试着回答一下这些问题：我们将会以哪种形式完成这份程序；我们将会以哪种方法完成一些方法；哪些东西是必要的；程序的总体结构是什么以及哪些数据是必要的，并了解这些数据的含义。

当然还有很多东西是读者无法从本篇章了解的，例如具体实现逻辑，参与运算的具体参数，等等，这些东西将会在详细设计中说明，而本篇章将不会赘述，也不会纠缠。

同时我们将在详细设计的对本文档不断地进行细节更改，所以在阅读之前请确定，本篇章为最终版本。

# 2.总体设计

## 2.1需求

使用户获得股票信息与预测信息

## 2.2目标

通过程序预测使用户能够快速有效的盈利

## 2.3运行环境

OpenJDK1.8及以上，MySQL8.0.3及以上，Mongo3.0及以上

## 2.4局限

本程序只提供概率性与大致的预测，无法提供极为准确的预测，并且由于部分原因，无法提供详实的数据，所以无法满足一些特殊需求，也会发对特定条件进行针对性分析。

## 2.5结构

本程序因为功能简单，并没有使用时下最为流行的微服务架构，如果改为微服务架构，会降低耦合，但是会增大代码的冗余，并且增加运行成本，本程序暂时不考虑使用。

本程序使用经典的三层架构：dao层，service层，controller层，十分中庸的解决了现在问题，并且作为最经常被使用的架构，它有着与众不同的简单和稳定。

# 3.接口设计

## 3.1对外接口

外部接口均以utf-8作为标准编码，并且对于各个接口的返回数据均以JSON格式作为标准返回格式，其余规定将在一下表格中进行备注说明。同时也因为本项目暂时没有考虑安全性问题，所以并没有类似于authorization的鉴权手段。

### 3.1.1数据展示模块

一、K线图接口

a. 主要实现功能

本接口将返回某股票自1996年至今的所有走势，同时在调用者调用正确的日期时，也需要提供某一时间段内的价格走势图

b. 主要的传入参数

主要传入参数为开始日期，截止日期，以及正确的股票编号。

c. 主要的返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数含义 | 类型 | 示例数据 | 备注 |
| img | 图片 | base64 | - | - |

d. 额外的说明与备注

本接口为get方法传参，并且当用户出入非开盘日期或传入的该股票处于停牌阶段的我们将向前选择就近的日期并计算出相应的值，单用户只需要简单的json数据时，本接口也可以做一定的拓展。

二、上市公司信息接口

a. 主要实现功能

本接口主要返回用户想要知道的一些关于公司的基本信息，如所在地，注册资本等等，我们希望单用户使用本接口并获得数据后能够对该上市公司有一个比较大致的了解。

b. 主要的传入参数

传入参数主要为上市公司的股票代码。

c. 主要的返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数含义 | 类型 | 示例数据 | 备注 |
| 公司名称 | 公司的实际名称 | String | 浙江广厦股份有限公司 | - |
| 公司名称2 | 公司名称简称 | String | 浙江广厦 | - |
| 成立日期 | 公司的成立日期 | String | 19930713 | - |
| 注册资本 | 注册资本 | String | 87178.91 | - |
| 地区名称 | 公司所在地区的名称 | String | 浙江省 | - |
| 地区编号 | 公司所在地区的编号 | String | CN330000 | - |
| 一级板块名称 | 名称 | String | CN330000 | - |
| 一级板块编号 | 编号 | String | 1102 | - |
| 所属行业名称 | 二级行业所属的名称 | String | 房地产开发 | - |
| 所属行业编号 | 所属的行业编号 | String | 430100 | - |

d. 额外的说明与备注

本接口同样使用的是get方法进行参数的传递，也就是通过url传值。

三、股票单日价格查询接口

a. 主要实现功能

此接口为股票的单日价格接口，输入日期和股票代码后，我们将会从数据库中查找相对应的股票信息，同时提供多种数据的展示。

b. 主要的传入参数

比较重要的数据为正确的日期，与正确的股票代码，当为输入正确的参数时，我们将返回空值，或错误信息。

c. 主要的返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数含义 | 类型 | 示例数据 | 备注 |
| 日期 | 交易日当天的日期 | String | 20090313 | - |
| 开盘价 | 交易日当天的开盘价格 | Double | 3.95 | - |
| 收盘价 | 交易日当天的收盘价格 | Double | 3.87 | - |
| 浮动金额 | 单日最高价与最低价的差值 | Double | -0.08 | - |
| 浮动比例 | 浮动金额的比例 | Double | -2.03 | - |
| 单日最低价 | 交易日当天的最低价格 | Double | 4.03 | - |
| 单日最高价 | 交易日当天的最高价格 | Double | 3.86 | - |
| 成交量 | 交易日当天该股票的成交总量 | Double | 629.36 | - |
| 成交额 | 当天的成交总额 | Double | 2472.21 | - |
| 换手率 | 股票交换的比例 | Double | 0.82 | - |

d. 额外的说明与备注

本方法为get请求，通过url传值。

四、上市公司经济参数接口

a. 主要实现功能

返回和提供上市公司的主要经济参数，希望用户通过调用本接口，能够对于希望了解的上市公司的经济情况有一个大致的了解与研究，但是本程序只提供最为基础的数据信息，并不包含更加深入的类似于证券交易所的收费查询信息，希望注意。

b. 主要的传入参数

如上一接口，本接口同样需要上市公司的股票代码来进行查询，但是不需要时间。

c. 主要的返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数含义 | 类型 | 示例数据 | 备注 |
| 日期 | 报表日期 | String | 20171231 | - |
| 每股收益 | 每股的净收益 | Double | 0.860 | - |
| 每股净资产 | 每股所有拥有的净资产 | Double | 7.381 | - |
| 每股现金流 | 请参考数据词典 | Double | 0.309 | - |
| 净资产增长率 | 请参考数据词典 | Double | 9.732 | - |
| 净资产收益率（加权） | 请参考数据词典 | Double | 12.240 | - |
| 主营业务收入增长率 | 请参考数据词典 | Double | 17.481 | - |
| 净利润增长率 | 请参考数据词典 | Double | 111.222 | - |
| 总资产增长率 | 请参考数据词典 | Double | -2.46 | - |
| 销售毛利率 | 请参考数据词典 | Double | 14.067 | - |
| 主营业务收入 | 请参考数据词典 | Double | 289092900259.280 | - |
| 主营业务利润 | 请参考数据词典 | Double | 24924170426.650 | - |
| 利润总额 | 请参考数据词典 | Double | 24035130135.610 | - |
| 净利润 | 请参考数据词典 | Double | 19170337569.640 | - |
| 资产总额 | 请参考数据词典 | Double | 350234632615.17 | - |
| 负债总额 | 请参考数据词典 | Double | 175762228075.96 | - |

d. 额外的说明与备注

GET方法作为主要的方法，该接口只会返回最新的财务报表，其中财务报表中也会有一定的缺省值，有可能来自于上市公司的统计不足，并不是接口问题。

### 3.1.2数据预测模块

一、股票chart计算接口

a. 主要实现功能

作为本项目核心接口，该接口主要为用户表现了，但股票处于上涨与下跌是，经过一定的计算，他们对于某一相同的随机概率算法，但是展示出了不同的统计学表现与不同的概率分布，那么就能够说明这个曲线对于股票的上涨与下跌有一个指示性的作用。同时也就能够证明了理论的成立。

b. 主要的传入参数

要得出入参数只有一个运算量，也就是一个数字。

c. 主要的返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数含义 | 类型 | 示例数据 | 备注 |
| img | 以url或Base64作为资源的图片资源 | String | locahhost:9902/static/chartImg/… | 涨跌两条曲线分布 |

d. 额外的说明与备注

因为受限于计算机的性能，当作该接口设计时我们应该考虑到该接口的极限运算能力，与计算时间，所以显而易见的是该接口的运算量会有一个最大值，或者说有一个极限值，同时这个接口但运算次数过少的时候无法真实的体现出股票涨跌的规律，使用理论上应该也有一个最小值，使用希望在之后的实际代码编写过程中能够通过经验和实际数据，能够获取这个接口的最大值或最小值。故在此不做过多的说明与解释。

二、股票chart预测接口

a. 主要实现功能

作为本项目的核心接口之一，该接口主要的目的是上一接口运算所得的结果进行一个合成，也就是说本接口才是真正的预测接口。本接口我们将提供出一张拥有三条曲线的图片，分别为上涨曲线，下跌曲线，本股票的曲线，用户可以根据我们提供出的图标数据自己进行判断。

b. 主要的传入参数

与上一接口相同的运算量，以及他希望预测的股票编号。

c. 主要的返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 股票chart预测接口 | | | | |
| 参数名称 | 参数含义 | 类型 | 示例数据 | 备注 |
| img | 以url或Base64作为资源的图片资源 | String | locahhost:9902/static/chartImg/… | 希望三条曲线能够合成一张图 |
| 涨跌幅度 | 该股票的涨跌幅度 | Double | 17.334 | - |

d. 额外的说明与备注

本接口与上一接口拥有相同的特点，也就是拥有一个极限运算量，与最小运算量，同时在调用本接口之前我们建议用户先调用上一接口，因为本接口将调用上一接口的运算数据，如果没有上一接口的运算数据，我们将返回错误给用户，使用请注意。

三、数学公式预测接口

a. 主要实现功能

主要实现功能为给出一个股票未来上升或下降的数值。

b. 主要的传入参数

股票代码。

c. 主要的返回参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数含义 | 类型 | 示例数据 | 备注 |
| 股票上升幅度 | 股票在一定时间内的上升幅度 | Double | 1.924 | - |

d. 额外的说明与备注

与之前的数学公式图标预测相同，都是通过大量的测试数据进行推导得出的，使用单计算量越大越能符合真实情况。

## 3.2内部接口

本内容与上文部分内容重合，特别是体现在返回参数上，使用并且该部分内容需要开发者自行考虑，此说明文档主要起到一个提示作用，并不会具体详细的解释各个参数，而是将接口内容大致说明，具体参数请由开发者自身决定。

### 3.2.1爬虫模块

本模块均为入库操作，只存在success或fail两种返回值。在操作本模块的时候请一定注意，部分接口有可能存在访问限制，所以请谨慎使用，并且考虑到合理的共存，请不要为了追求性能使用过多的线程，以此给他人服务器参数大于预测的压力。

一、东方财富网爬虫接口

抓取东方财富网的股票列表信息，并且进行解析，并且将结果入库，在此我们希望获得两个数据，即股票代码，以及股票所属交易所，并且只抓取上证与深证的主要股票，不包含其他内地小众股票。

二、雪球爬虫接口

抓取雪球网的数据，公司数据与企业报表数据。该接口经过预研发现可以通过

三、搜狐爬虫接口

历史股价信息。

四、百度手机爬虫接口

股票基础数据。

### 3.2.2基础数据服务模块

一、原始数据转换一层数据接口

将上文中所提及的抓取获得的原始数据直接转化为一级数据格式，也就是进行一些处理，我们需要的是提炼出比较关键的参数，并且对这些参数进行格式化，具体的参数计算以及参数的选取我们会在详细设计中严格的说明，但是在这里我们这是提及我们需要这些东西，来为下一步的二级数据做一个准备。

二、一层数据转二层数据接口

当我们拥有一个一级数据时，我们会发现一级数据已经拥有了一个天然的权重，并且这样的现象是十分严重的，我们无法保证在进行股票比较时，这些权重是否会对最终的结果造成不必要的影响，使用二级数据我们需要将这个天然的权重淡化，甚至于抹平这个权重带来的不必要的影响，使用我们需要对以及数据进行所谓的平等化、去权重化的操作，也就是生成我们的二级数据，如上一接口一样，如何才能去除权重，我们将会在详细设计中较好的诠释出来，使用请阅读下面的篇章。

三、随机数生成接口

随机数生成模块是预测的重要接口，我们需要大量的假想数据，同时我们也需要大量的为结果提供概率分布的数据，而随机数生成接口必须完成这样的任务，也就是使预测结果可统计，并且使其符合统计学规律，所有的计算类的包含具体公式与算法的接口，我们都会在详细设计中给予说明。

### 3.2.3数据预测计算模块

一、公式预测接口

公式预测接口，这不是一个对外开放的接口，这个接口主要通过大量的预测性的运算，并通过一种比较简单的自我学习，自我调整的方式，来进行不断的比较，拟合，同时删除无法进行回归的数据，我们也不在这里讨论十分详细的运算法则，与最终的公式实现。

# 4.数据库设计

数据库这里使用的是MySQL与Mongo，日后还会考虑使用PostgreSQL，以及部分的newSQL，对于现有逻辑，MySQL主要存放大批量的结构化数据，或者说可转化为结构化的数据，这些数据需要包含以下特点，需要大量的读取写入，将会存放大量的数据（超过2GB，当mongo类似的内存型数据库超过一定量时，读写效率会大幅度下降），Mongo数据库将存放非结构化数据。

同时因为本项目不包含过多的数据库关系逻辑概念，所以我们将不会对于数据库概念设计E-R图有描述，仅仅只会告诉开发者哪些表是被需要的，以及这些表包含的字段所代表的意义（也就是我们所说的数据字典），同时我们也将对数据库的物理结构进行说明，但是因为是小型数据库，我们暂时不会采用主从数据库类型的分布式结构。

## 4.1数据库结构设计

### 4.1.1 MySQL部分

该部分主要存储结构化数据，即硬数据，不会轻易改变，适合大量读写。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbol | | | | |
| 股票代码与所属证券交易所 | | | | |
| 字段 | 描述 | 数据类型 | 约束 | 主键 |
| stock\_code | 股票代码 | varchar(30) | 不可为空 | 1 |
| exchange | 股票所属证券交易所 | varchar(30) | - | 0 |
|  |  |  |  |  |
| basic\_info | | | | |
| 上市公司基础信息 | | | | |
| 字段 | 描述 | 数据类型 | 约束 | 主键 |
| code | 股票编号 | varchar(30) | 不可为空 | 1 |
| boardcode | 板块编号 | varchar(100) | - | 0 |
| boardname | 地区代码(省级) | varchar(100) | - | 0 |
| comp\_name | 公司名称 | varchar(100) | - | 0 |
| compsname | 公司股票简称 | varchar(100) | - | 0 |
| found\_date | 成立日期 | varchar(100) | - | 0 |
| keycode | 地区编号 | varchar(100) | - | 0 |
| keyname | 地区名称 | varchar(100) | - | 0 |
| level2code | 二级行业编号 | varchar(100) | - | 0 |
| level2name | 二级行业名称 | varchar(100) | - | 0 |
| regcapital | 注册资本 | varchar(100) | - | 0 |
|  |  |  |  |  |
| fin\_info | | | | |
| 上市公司季度财务报表 | | | | |
| 字段 | 描述 | 数据类型 | 约束 | 主键 |
| id | id号 | varchar(100) | 不可为空 | 1 |
| basiceps | 基本每股收益 | double | 不可为空 | 0 |
| code | 股票编号 | varchar(100) | - | 0 |
| main\_businc\_grow\_rate | 主营业务收入增长率(%) | double | 不可为空 | 0 |
| mainbusiincome | 主营业务收入 | double | 不可为空 | 0 |
| mainbusiprofit | 主营业务利润 | double | 不可为空 | 0 |
| naps | 每股净资产 | double | 不可为空 | 0 |
| netassgrowrate | 净资产增长率(%) | double | 不可为空 | 0 |
| netinc\_grow\_rate | 净利润增长率(%) | double | 不可为空 | 0 |
| netprofit | 净利润 | double | 不可为空 | 0 |
| o\_per\_cash\_per\_share | 每股现金流 | double | 不可为空 | 0 |
| report\_date | 报表日期 | varchar(100) | - | 0 |
| salegrossprofitrto | 销售毛利率(%) | double | 不可为空 | 0 |
| totalassets | 资产总额 | double | 不可为空 | 0 |
| totalliab | 负债总额 | double | 不可为空 | 0 |
| totassgrowrate | 总资产增长率(%) | double | 不可为空 | 0 |
| totprofit | 利润总额 | double | 不可为空 | 0 |
| weightedroe | 净资产收益率(加权)(%) | double | 不可为空 | 0 |
|  |  |  |  |  |
| hq\_info | | | | |
| 股票历史价格 | | | | |
| 字段 | 描述 | 数据类型 | 约束 | 主键 |
| id | id号 | varchar(100) | 不可为空 | 1 |
| close | 收盘价 | double | 不可为空 | 0 |
| code | 股票编号 | varchar(100) | 索引 | 0 |
| data | 报表日期 | varchar(100) | - | 0 |
| exchange\_rate | 换手率（%） | double | 不可为空 | 0 |
| float\_pre | 浮动比例（百分比） | double | 不可为空 | 0 |
| floatw | 浮动金额 | double | 不可为空 | 0 |
| high\_price | 最高 | double | 不可为空 | 0 |
| low\_price | 最低 | double | 不可为空 | 0 |
| open | 开盘价 | double | 不可为空 | 0 |
| vol | 成交量（万） | double | 不可为空 | 0 |
| vol\_price | 成交额（万） | double | 不可为空 | 0 |
|  |  |  |  |  |
| snap\_shot | | | | |
| 股票实时简单经济指数 | | | | |
| 字段 | 描述 | 数据类型 | 约束 | 主键 |
| code | 股票编号 | varchar(100) | 不可为空 | 1 |
| bv\_ratio | 市净率 | double | 不可为空 | 0 |
| circulating\_capital | 流通股 | int | - | 0 |
| net\_assets\_per\_share | 每股净资产 | double | 不可为空 | 0 |
| per\_share\_earn | 每股收益 | double | 不可为空 | 0 |
| peratio | 市盈率 | double | 不可为空 | 0 |
| total\_share\_capital | 总股本 | int | - | 0 |
|  |  |  |  |  |
| first\_calculation\_unit | | | | |
| 一层计算单元 | | | | |
| 字段 | 描述 | 数据类型 | 约束 | 主键 |
| code | 股票编号 | varchar(100) | 不可为空 | 1 |
| bv\_ratio | 市净率 | double | 不可为空 | 0 |
| fifteen\_price | 15日均价 | double | 不可为空 | 0 |
| four\_hundred\_price | 400天均价 | double | 不可为空 | 0 |
| history\_price | 历史均价 | double | 不可为空 | 0 |
| low\_fifteen\_price | 15日最低价 | double | 不可为空 | 0 |
| low\_four\_hundred\_price | 400天最低价 | double | 不可为空 | 0 |
| low\_history\_price | 历史最低价 | double | 不可为空 | 0 |
| low\_one\_eighty\_price | 180天最低价 | double | 不可为空 | 0 |
| low\_sixty\_price | 60日最低价 | double | 不可为空 | 0 |
| low\_thirty\_price | 30日最低价 | double | 不可为空 | 0 |
| mainbusiincome | 主营业务收入 | double | 不可为空 | 0 |
| mainbusiincome\_avg | 历史主营业务收入平均值 | double | 不可为空 | 0 |
| netinc\_grow\_rate | 净利润增长率(%) | double | 不可为空 | 0 |
| netprofit | 净利润 | double | 不可为空 | 0 |
| netprofit\_avg | 历史净利润平均值 | double | 不可为空 | 0 |
| now\_price | 时价 | double | 不可为空 | 0 |
| one\_eighty\_price | 180天均价 | double | 不可为空 | 0 |
| peratio | 市盈率 | double | 不可为空 | 0 |
| sixty\_price | 60日均价 | double | 不可为空 | 0 |
| thirty\_price | 30日均价 | double | 不可为空 | 0 |
| totalassets | 资产总额 | double | 不可为空 | 0 |
| totalassets\_avg | 历史资产总额平均值 | double | 不可为空 | 0 |
| up\_rate | 60日涨幅(%)(1/4 \* 前30日 + 3/4 \* 后半段30日均价) | double | 不可为空 | 0 |
|  |  |  |  |  |
| second\_calculation\_unit | | | | |
| 二层计算单元 | | | | |
| 字段 | 描述 | 数据类型 | 约束 | 主键 |
| code | 股票编号 | varchar(100) | 不可为空 | 1 |
| bv\_ratio | 1 / ( 1 + e^( -市净率 / 市净率平均值)) | double | 不可为空 | 0 |
| fifteen\_price | 15日均价 / 时价 | double | 不可为空 | 0 |
| four\_hundred\_price | 400天均价 / 时价 | double | 不可为空 | 0 |
| history\_price | 历史均价 / 时价 | double | 不可为空 | 0 |
| low\_fifteen\_price | 15日最低价 / 时价 | double | 不可为空 | 0 |
| low\_four\_hundred\_price | 400天最低价 / 时价 | double | 不可为空 | 0 |
| low\_history\_price | 历史最低价 / 时价 | double | 不可为空 | 0 |
| low\_one\_eighty\_price | 180天最低价 / 时价 | double | 不可为空 | 0 |
| low\_sixty\_price | 60日最低价 / 时价 | double | 不可为空 | 0 |
| low\_thirty\_price | 30日最低价 / 时价 | double | 不可为空 | 0 |
| mainbusiincome | 1 / ( 1 + e^( -主营业务收入 / 历史主营业务收入平均值)) | double | 不可为空 | 0 |
| mainbusiincome\_avg | 历史主营业务收入平均值 | double | 不可为空 | 0 |
| netinc\_grow\_rate | 1 / ( 1 + e^( -净利润增长率 / 净利润增长率平均值)) | double | 不可为空 | 0 |
| netprofit | 1 / ( 1 + e^( -净利润 / 历史净利润平均值)) | double | 不可为空 | 0 |
| netprofit\_avg | 历史净利润平均值 | double | 不可为空 | 0 |
| now\_price | 时价 | double | 不可为空 | 0 |
| one\_eighty\_price | 180天均价 / 时价 | double | 不可为空 | 0 |
| peratio | 1 / ( 1 + e^( -市盈率 / 市盈率平均值)) | double | 不可为空 | 0 |
| sixty\_price | 60日均价 / 时价 | double | 不可为空 | 0 |
| thirty\_price | 30日均价 / 时价 | double | 不可为空 | 0 |
| totalassets | 1 / ( 1 + e^( -资产总额 / 历史资产总额平均值)) | double | 不可为空 | 0 |
| totalassets\_avg | 历史资产总额平均值 | double | 不可为空 | 0 |
| up\_rate | 60日涨幅(%)(1/4 \* 前30日 + 3/4 \* 后半段30日均价) | double | 不可为空 | 0 |

### 4.1.2 Mongo部分

此部分为非结构化数据，并且只会存放临时数据，在这里只会表达出关键词语的含义，并不会做过多的解释。

## 4.2数据库环境

Windows10操作系统，MySQl5.6或以上，Mongo3.3.x或以上

# 5 程序模型设计

## 5.1程序主流程

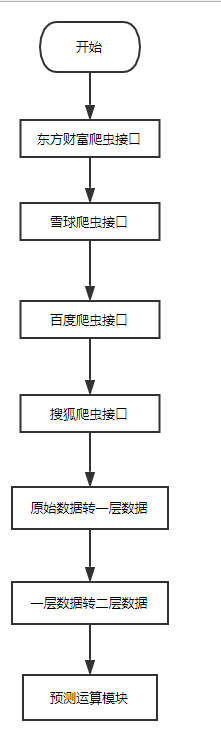


图5-1-1

## 5.2 运行预测与安全

### 5.2.1 运行时间与所占用资源

以下标准由具体环境变动而变动，此处估计时间为中上级家用台式机运行速度。

爬虫模块：总占用资源 磁盘 4GB

东方财富网爬虫：30秒以内

雪球爬虫：30分钟以内

百度爬虫：10分钟以内

搜狐爬虫：60分钟以内

数据处理模块：占用资源 磁盘50MB

原始数据转一级数据：10分钟以内

一级数据转二级数据：20分钟以内

数据计算模块：占用资源 磁盘20GB以下

图表计算：按计算量而定，一次计算应控制在8个小时以内

公式计算：同上

预测模块：无占用

图片预测：同上

公式预测：同上

总运行时间与占用资源估计

硬盘资源：25GB以下

主流程运行时间不超过24个小时

## 5.3安全与备灾

因为不存在