



软件项目管理 立项报告



题 目 股票数据监测系统

学 院 软件学院

专 业 软件工程

组 长 杨淳葳 2020141470305

成 员 周备 20201414660110

文昱涵 2020141240056

王超弘 2020141660104

解佳慧 2020141461172

晏扬 2020141461072

石佳 2020141461140

指导教师 杨波

二 0 二 三 年 三 月 二 十 二 日

目录

第一章 项目背景及意义	5
第二章 主要开发内容	5
第三章 应用技术	6
3.1 框架: Spring Boot	6
3.2 持久层框架: MyBatis	6
3.3 数据库: MySQL	7
3.4 JWT	7
3.5 前端: React+Axios	7
3.6 Swagger	8
3.7 Python	8
第四章 开发模式	9
第五章 组员分工	10
第六章 编码规范	11
6.1 命名	11
6.1.1 类命名	11
6.1.2 方法命名	11
6.1.3 变量命名	12
6.1.4 图片命名	12
6.1.5 常量	12
6.2 格式	12
6.2.1 每一行的字数限制	12
6.2.2 注释	13
6.2.3 代码优化	13
6.2.4 系统函数和自定义函数	13
6.2.5 大括号	13
6.2.6 运算符	14
6.2.6.2 for 语句	14
第七章 管理工具	15
7.1 版本管理工具: Github	15
7.2 华为云 Devcloud	15

第八章 运行环境.....	16
第九章 开发计划.....	16
第十章 风险分析.....	17
10.1 技术问题.....	17
10.2 时间问题.....	17
10.3 分工问题.....	18

第一章 项目背景及意义

随着金融业的发展，股票的历史成交信息和实时交易数据量日益庞大。为了更容易地理解和利用股票数据，把握市场变化，提高投资决策的效率和准确性，从而提高投资效果和竞争力，股票数据分析可视化工具对金融业从业者和投资者而言作用重大。

随着科学技术的不断提高，计算机科学日渐成熟，其强大的功能已为人们深刻认识。使用计算机对股票交易信息进行管理，有着手工管理所无法比拟的优点，例如：实时性高、检索迅速、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点使股票信息处理工作规范化、系统化、程序化，提高信息处理的速度、准确性和及时性。因此，建立一个股票数据监测系统，可以极大地简化了金融从业者的工作，为企业在瞬息万变的金融市场立于不败之地提供重要保障。

第二章 主要开发内容

该项目主要开发工作分为四个模块：用户管理、历史数据、用户收藏、实时数据。

	功能	描述
用户管理	登录	用户通过输入正确的账号和密码进入系统。
	注册	用户可以通过点击邮箱注册账号。
	注销账号	用户可以通过点击注销账号按钮来退出系统。

	修改密码	用户可以修改密码，要求输入原密码，若原密码错误则无法修改。
历史数据	查看股票	可查看股票的历史交易数据，包括开盘价、收盘价等。
	创建股票	用户可以自行创建股票并录入相关信息。
	搜索股票	通过股票的 ID 或名字搜索到具体的某只股票的信息。
用户收藏	收藏	用户可以将某只具体的股票收藏并添加到收藏界面。
	查看	用户可以在收藏界面查看已收藏股票的交易数据。
实时数据	查看	用户能够查看收藏股票在一天之内的波动情况
	详情	用户在详情界面可以查看收藏股票历史数据 k 线图。
	导入/导出	能够以 csv 格式导入/导出数据。

第三章 应用技术

3.1 框架：Spring Boot

Spring Boot 是由 Pivotal 团队提供的一套开源框架，可以简化 spring 应用的创建及部署。它提供了丰富的 Spring 模块化支持，可以帮助开发者更轻松快捷地构建出企业级应用。Spring Boot 通过自动配置功能，降低了复杂性，同时支持基于 JVM 的多种开源框架，可以缩短开发时间，使开发更加简单和高效。

3.2 持久层框架：MyBatis

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动

设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs (Plain Ordinary Java Object, 普通的 Java 对象) 映射成数据库中的记录。

3.3 数据库：MySQL

Mysql 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System, 关系数据库管理系统) 应用软件之一。

3.4JWT

Json web token (JWT), 是为了在网络应用环境间传递声明而执行的一种基于 JSON 的开放标准 (RFC 7519). 该 token 被设计为紧凑且安全的，特别适用于分布式站点的单点登录 (SSO) 场景。JWT 的声明一般被用来在身份提供者和服务提供者间传递被认证的用户身份信息，以便于从资源服务器获取资源，也可以增加一些额外的其它业务逻辑所必须的声明信息，该 token 也可直接被用于认证，也可被加密。

3.5 前端：React+Axios

React 是用于构建用户界面的 JavaScript 库，起源于 Facebook 的内部项目，该公司对市场上所有 JavaScript MVC 框架都不满意，决定自行开发一套，用于架设 Instagram 的网站。在 React 里传递多

种类型的参数，如声明代码，帮助开发者渲染出 UI、也可以是静态的 HTML DOM 元素、也可以传递动态变量、甚至是可交互的应用组件。

react 的特点包括 1. 声明式设计 2. 组件化 3. 高效 4. 灵活

Axios 是一个基于 promise 的网络请求库，作用于 node.js 和浏览器中，它是 isomorphic 的(即同一套代码可以运行在浏览器和 node.js 中)。在服务端它使用原生 node.js http 模块，而在客户端(浏览端)则使用 XMLHttpRequest。

3.6 Swagger

Swagger 是一款 RESTFUL 接口的文档在线自动生成+功能测试功能软件。Swagger 是一个规范和完整的框架,用于生成、描述、调用和可视化 RESTful 风格的 Web 服务。目标是使客户端和文件系统作为服务器以同样的速度来更新文件的方法,参数和模型紧密集成到服务器。

3.7 Python

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。Python 的设计具有很强的可读性，相比其他语言经常使用英文关键字，其他语言的一些标点符号，它具有比其他语言更有特色语法结构。

使用 Python 调用第三方 Api 获取实时数据。

第四章 开发模式

本项目使用敏捷开发作为开发模型，敏捷开发是一种迭代、增量的软件开发方法，它强调通过快速交付、灵活性和合作来满足不断变化的需求。它强调团队协作、快速响应变化、提供高质量的软件产品，并注重持续的反馈和改进。

敏捷开发核心理念是将软件开发过程分解为一系列小的、可重复的周期，每个周期称为“迭代”，每个迭代都会产生一个可用的软件版本，该版本包括新的或改进的功能和特性。这种方法的优点在于，它可以快速响应需求变化，缩短交付周期，提高客户满意度，同时也可以降低项目风险。

敏捷开发方法有多种，如 Scrum、Kanban、XP 等，它们都有着共同的原则，例如注重个人和互动、可工作的软件、合作、响应变化等，这些原则旨在帮助团队更好地协作、交付高质量的软件，并不断地改进。

对于本团队中的不同角色来说，使用敏捷开发有以下优点：

产品经理可以通过敏捷开发模型更好地管理需求。敏捷开发模型注重持续的用户反馈和需求变化，可以让产品经理快速了解用户需求和反馈，及时进行调整和更新。同时，敏捷开发模型也能够帮助产品经理更好地管理产品需求，不断优化和完善产品功能和特性。

敏捷开发模型可以让项目经理更好地管理项目进度和团队协作。敏捷开发模型注重迭代周期和团队协作，可以让项目经理更好地掌控

项目进度和成员工作状态，及时调整项目计划，保证项目顺利进行。同时，敏捷开发模型也可以让项目经理更好地协调团队成员之间的合作，提高团队协作效率。

敏捷开发模型可以让运营更好地了解用户需求和反馈，及时进行调整和更新，提高用户满意度。同时，敏捷开发模型也可以让运营更好地掌握产品功能和特性，提高产品运营效率。

敏捷开发模型可以让程序员更好地理解用户需求和反馈，及时进行调整和更新，提高产品质量。同时，敏捷开发模型也可以让程序员更好地协作和沟通，提高团队协作效率。

敏捷开发模型可以让测试更好地理解用户需求和反馈，及时进行测试和反馈，提高产品质量。同时，敏捷开发模型也可以让测试更好地协作和沟通，提高团队协作效率。

敏捷开发模型可以让维护更好地了解用户需求和反馈，及时进行更新和维护，提高产品质量和用户满意度。同时，敏捷开发模型也可以让维护更好地协作和沟通，提高团队协作效率。

第五章 组员分工

本项目的开发按照软件项目开发周期的各个阶段（配置管理、需求分析、系统设计、代码编写、软件测试和部署）进行组员分工。

成员（七人）	分工	完成情况
杨淳葳（项目	项目管理	

经理)	需求分析 概要设计 配置管理	
晏扬(产品经理)	需求分析 界面设计	
王超弘(程序员)	代码编写(前端) 项目部署	
解佳慧(程序员)	代码编写(后端) 项目部署	
文昱涵(维护)	项目日常维护	
周备(测试)	软件测试(文档) 软件测试(功能)	
石佳(运营)	产品运营(文档)	

第六章 编码规范

6.1 命名

6.1.1 类命名

类名是个一名词, 采用大小写混合的方式, 每个单词的首字母大写(大驼峰命名法)。尽量使类名简洁而富于描述。使用完整单

词, 避免缩写词(除非该缩写词被更广泛使用, 像 URL, HTML)

eg: UserManage, UserData 等。

6.1.2 方法命名

方法名是一个动词, 采用大小写混合的方式, 第一个单词的首字母小写, 其后单词的首字母大写(小驼峰法命名法)。

eg: run(), runFast(), getBackground() 等

6.1.3 变量命名

除了变量名外，所有实例，包括类，类常量，均采用大小写混合的方式，第一个单词的首字母小写，其后单词的首字母大写(小驼峰法命名法)。变量名不应以下划线或美元符号开头，尽管这在语法上是允许的。变量名应简短且富于描述。变量名的选用应该易于记忆，即能够指出其用途。尽量避免单个字符的变量名，除非是一次性的临时变量。临时变量通常被取名为 `i`, `j`, `k`, `m` 和 `n`，它们一般用于整型；`c`, `d`, `e`，它们一般用于字符型。

eg: `char c`, `int i`, `float myWidth` 等

6.1.4 图片命名

使用英文，全部小写，单词中间"_"隔开添加 UI 模块名作为前缀，避免冲突

图片存放在 `/static/images` 下 eg: `bar_title`, `icon_coin`

6.1.5 常量

类常量和 ANSI 常量的声明，应该全部大写，单词间用下划线隔开。(尽量避免 ANSI 常量，容易引起错误)

eg: `static final int MIN_WIDTH = 4`; `static final int MAX_WIDTH = 999`;
`static final int GET_THE_CPU = 1`;

6.2 格式

6.2.1 每一行的字数限制

每一行的代码尽量不要超出 80 个字的长度，超出的回车排版，方法名的冒号对齐

6.2.2 注释

以下三种情况需要写注释：

- (1) 公共接口(注释要告诉阅读代码的人, 当前类能实现什么功能)。
- (2) 涉及到比较深层专业知识的代码（注释要体现出实现原理和思想）。
- (3) 容易产生歧义的代码（但是严格来说，容易让人产生歧义的代码是不允许存在的）。

6.2.3 代码优化

避免相同的代码段在多个地方出现相同的代码，必须归纳出来并且用一个类封装起来

语句嵌套层次不得超过 3 层，超出的必须抽取出中间函数 g: for, while 循环 if, do 等

及时删除或注释掉无用的代码确定不使用的代码应该删除

6.2.4 系统函数和自定义函数

系统函数放在上方，自定义函数放在下方

6.2.5 大括号

控制语句(if, for, while, switch)中，大括号开始于行尾函数中，大括号开始于行首

6.2.6 运算符

6.2.6.1 运算符与变量之间的间隔

6.2.6.1.1 一元运算符与变量之间没有空格：

例如：!bValue、~iValue、++iCount、*strSource、&fSum

6.2.6.1.2 二元运算符与变量之间必须有空格

fWidth = 5 + 5; fLength = fWidth * 2;

fHeight = fWidth + fLength; for(int i = 0; i < 6; i++)

6.2.6.1.3 多个不同的运算符同时存在时应该使用括号来明确优先级；

在多个不同的运算符同时存在的时候应该合理使用括号，不要盲目依赖操作符优先级。

6.2.6.2 if 语句

必须列出所有分支（穷举所有的情况），而且每个分支都必须给出明确的结果。

不要使用过多的分支，要善于使用 return 来提前返回错误的情况

首先判断出各种错误的情况然后提前返回，把最正确的情况放到最后返回。条件过多，过长的時候应该换行

条件语句的判断应该是变量在左，常量在右每个分支的实现代码都必须被大括号包围

6.2.6.3 for 语句

不可在 for 循环内修改循环变量，防止 for 循环失去控制。避免使

用 `continue` 和 `break`。

第七章 管理工具

7.1 版本管理工具：Github

作为开源代码库以及版本控制系统，Github 拥有超过 900 万开发者用户。随着越来越多的应用程序转移到了云上，Github 已经成为了管理软件开发以及发现已有代码的首选方法。GitHub 可以托管各种 git 库，并提供一个 web 界面，但它与外国或中国的服务不同，GitHub 的独特卖点 在于从另外一个项目进行分支的简易性。为一个项目贡献代码非常简单：首先点击项目站点的“fork”的按钮，然后将代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。已经有人将 GitHub 称为代码玩家的 MySpace。

7.2 华为云 Devcloud

一站式、全流程、安全可信的云原生 DevSecOps 平台，开箱即用，预置最佳实践，助力质量提升和效率倍增：

- 1.华为研发实践加持：集华为多年研发实践，前沿研发理念，领先研发工程能力于一体；
- 2.一站式 DevSecOps 平台：提供端到端的 10+ 子服务，覆盖软件开发全生命周期，开箱即用；

3.支持丰富的技术栈：20+主流编程语言、开发框架和构建环境，应用无缝上云；

4.安全可信：Built-In 研发安全能力，5000+代码检查规则，Top10 语言广泛支持。

第八章 运行环境

操作系统：WINDOWS、MAC OS

第九章 开发计划

计划	时间
需求分析	5-6 周（32h）
具体设计	7-9 周（48h）
基本界面与功能	10-13 周（64h）
测试与修改	14-15 周（32h）
答辩	16 周
开发人员：四人	总工时：176 小时

第十章 风险分析

10.1 技术问题

该项目对于并发量的要求较高，在极端情况下可能会出现系统崩溃的情况，这在实际使用中是不能容忍的；该项目对于实时性的要求较高，在使用过程中需要做到低延迟的数据查询和数据录入。如果遇到技术难题，尽量尝试查阅资料，观看视频等方式学习，如若学不会，可以与组员协商，尝试简化功能。

为了避免这些风险，可以以下措施：

1. 技术选型要合理，选择成熟稳定的技术方案；
2. 建立有效的技术沟通机制，及时发现和解决问题；
3. 建立完善的技术文档体系，便于团队成员之间的知识共享；
4. 培训和提高团队成员的技能水平，以便更好地应对技术问题；

10.2 时间问题

根据工时估计功能的数量和难度，以防时间不足，如果确实出现时间不足的状况，则删减优先级较低的功能。

可以采取以下措施：

1. 制定详细的项目计划，包括时间节点和任务分解；
2. 建立有效的沟通机制，及时发现和解决问题；

3. 严格执行项目计划，确保按时完成任务；
4. 建立完善的风险管理机制，及时应对风险事件；

10.3 分工问题

在软件开发过程中，人员分工的合理性和科学性对于项目的成功至关重要。如果人员分工不合理，可能会导致以下问题：任务分配不均，导致某些成员负担过重，而另一些成员则没有得到充分利用；团队协作效率低下，沟通不畅，影响项目进度；技术水平不匹配，导致某些成员无法胜任自己的工作。

为了避免这些问题，可以采取以下措施：

1. 根据团队成员的技能和经验进行合理的人员分工；
2. 制定详细的任务计划和时间节点，并明确每个成员的职责；
3. 建立有效的沟通机制，及时发现和解决问题；
4. 建立完善的培训机制，提高团队成员的技能水平。