
股票数据分析系统

(Stock Data Analysis System)

软件设计说明书

(Software Design Description)

编 撰 人：全组人员

审 核 人：杨波

文档版本：V1

版本历史

序号	版本	更改时间	更改人	状态	更改描述
1	V1. 0	2023. 4. 20	全组人员	新建	
2	V1. 1	2023. 5. 1	全组人员	修改	<p>晏扬：第一章引言、第二章项目概述编写 石佳：第三章数据设计编写 杨淳葳：第四章功能设计编写，包括【SDD-1】、【SDD-2】、【SDD-3】和【SDD-4】 王超弘：第四章功能设计编写，包括【SDD-5】、【SDD-6】、【SDD-7】和【SDD-8】 文昱涵：第四章功能设计编写，包括【SDD-9】、【SDD-10】、【SDD-11】、【SDD-12】和【SDD-13】 解佳慧：第五章接口设计编写（包括，包括【SDD-14】、【SDD-15】、【SDD-16】），第六章性能设计编写（包括【SDD-17】、【SDD-18】），文档整合 周备：第八章约束限制编写，第九章 SDD 汇总表编写</p>
3	V1. 2	2023. 5. 10	全组人员	修改	<p>石佳：第二章项目概述、第四章数据设计修改 杨淳葳：第五章功能设计修改，包括【SDD-1】、【SDD-2】、【SDD-3】和【SDD-4】 王超弘：第五章功能设计修改，包括【SDD-5】、【SDD-6】、【SDD-7】和【SDD-8】 文昱涵：第五章功能设计修改，包括【SDD-9】、【SDD-10】、【SDD-11】、【SDD-12】和【SDD-13】 解佳慧：第六章接口设计修改（包括，包括【SDD-14】、【SDD-15】、【SDD-16】），格式调整 周备：第九章性能设计修改、第三章架构设计补充，包括【SDD-17】、【SDD-18】 晏扬：第九章 SDD 和 SRS 对应表修改</p>

状态：新建、增加、修改、删除。

目录

1.引言	1
1.1 编制目的	1
1.2 范围	1
1.3 预期的读者	1
1.4 术语和缩略语	1
2.项目概述	2
2.1 背景	2
2.2 范围	2
2.3 系统用例图	3
2.4 用户的特点	3
2.5 假定条件和约束限制	3
2.6 运行环境	4
3.架构设计	4
4.数据设计	6
4.1 E-R 图及表汇总	6
4.2 表结构设计	7
5.功能设计	9
5.1 用户管理	9
5.1.1 【SDD-001】登录	9
5.1.2 【SDD-002】注册	10
5.1.3 【SDD-003】注销账号	12
5.1.4 【SDD-004】修改密码	13
5.2 历史数据	15
5.2.1 【SDD-005】查看股票	15
5.2.2 【SDD-006】创建股票	16
5.2.3 【SDD-007】搜索股票	18
5.3 用户收藏	19
5.3.1 【SDD-008】收藏股票	20
5.3.2 【SDD-009】查看收藏	21
5.4 实时数据	22
5.4.1 【SDD-010】查看实时数据	22
5.4.2 【SDD-011】查看详情数据	24

5.5 数据的导入和导出	26
5.5.1 【SDD-012】股票数据导入	26
5.5.2 【SDD-013】股票数据导出	27
6. 接口设计	29
6.1 【SDD-014】内部接口设计	29
6.2 【SDD-015】外部接口设计	36
6.3 【SDD-016】用户界面设计	38
7. 性能设计	43
7.1 【SDD-017】响应时间性能设计	43
7.2 【SDD-018】并发设计	43
8. 约束限制	44
8.1 图形要求	44
8.2 报表格式	44
9. SDD 和 SRS 对应表	45

1. 引言

1.1 编制目的

本说明书目的在于明确说明“股票数据分析系统”的需求、界定系统实现功能的范围、指导系统设计以及编码。作为“股票数据分析系统”开发和验收的参考依据、作为测试人员设计测试用例的依据。

1.2 范围

- 1.产品名称：股票数据分析系统
- 2.产品功能：用户管理、历史数据、用户收藏、实时数据
- 3.产品应用：作为股票数据分析可视化工具利用计算机对股票交易信息进行管理，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。

1.3 预期的读者

开发人员、项目经理、产品经理、测试人员、文档的编写人员

1.4 术语和缩略语

术语、缩略语	解释
SQL	Structured Query Language（结构化查询语言），一种用于访问、查询数据库的语言
MySQL	系统所使用的数据库管理系统（DBMS）

2. 项目概述

2. 1 背景

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。

2. 2 范围

- 1) 用户管理
 - a. 登录
 - b. 注册
 - c. 注销账号
 - d. 修改密码
- 2) 历史数据
 - a. 查看股票
 - b. 创建股票
 - c. 搜索股票
- 3) 用户收藏
 - a. 收藏
 - b. 查看
- 4) 实时数据
 - a. 查看
 - b. 详情
 - c. 导入/导出

2.3 系统用例图

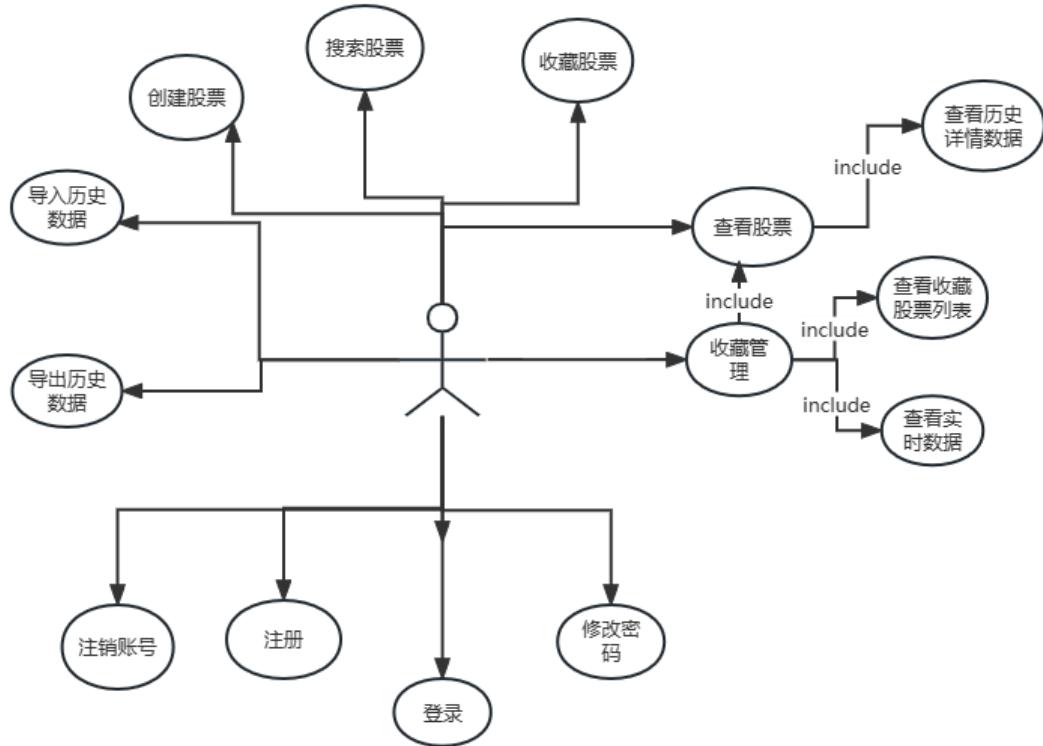


图 2-1 系统用例图

2.4 用户的特点

表 2-1 用户特点

用户	主要业务描述	用户特点
普通用户	登录、注册账号, 进行密码管理, 对实时数据进行操作, 对股票信息进行收藏	熟悉本系统的功能, 需要通过该系统了解股票信息

2.5 假定条件和约束限制

- 1) 对软件环境的要求: 个人 PC, 系统基于 Windows/MacOS
- 2) 对硬件环境的要求: 具有 IO 设备 (显示器、键鼠)、网络连接
- 3) 对软件使用者的要求: 熟悉基本的电脑操作

2.6 运行环境

1) 客户端

操作系统: Windows 10/ Windows 11/MacOS

服务器的通讯: TCP/IP 协议, HTTP 协议

2) 应用服务器端

操作系统: 腾讯云-CentOS7

应用服务器: Tomcat 9

3) 数据库

管理系统: MySQL5.7

可视化工具: Navicat

设计工具: PowerDesigner

访问工具: MyBatis

4) 后端编程

Java 运行环境: JDK8

IDE 工具: IntelliJ IDEA

5) 前端编程

IDE 工具: Visual Studio

3. 架构设计

为了实现系统的可靠性、可扩展性、可维护性、可重用性等优良特性，以满足系统的功能需求和性能要求，对本系统进行如下图的架构设计。系统架构设计是将系统分解成组成部分，定义它们之间的交互和依赖关系，以及每个组件的职责和功能。这有助于确保系统的整体性能和质量，以及在日后的开发、维护和扩展中更加容易。

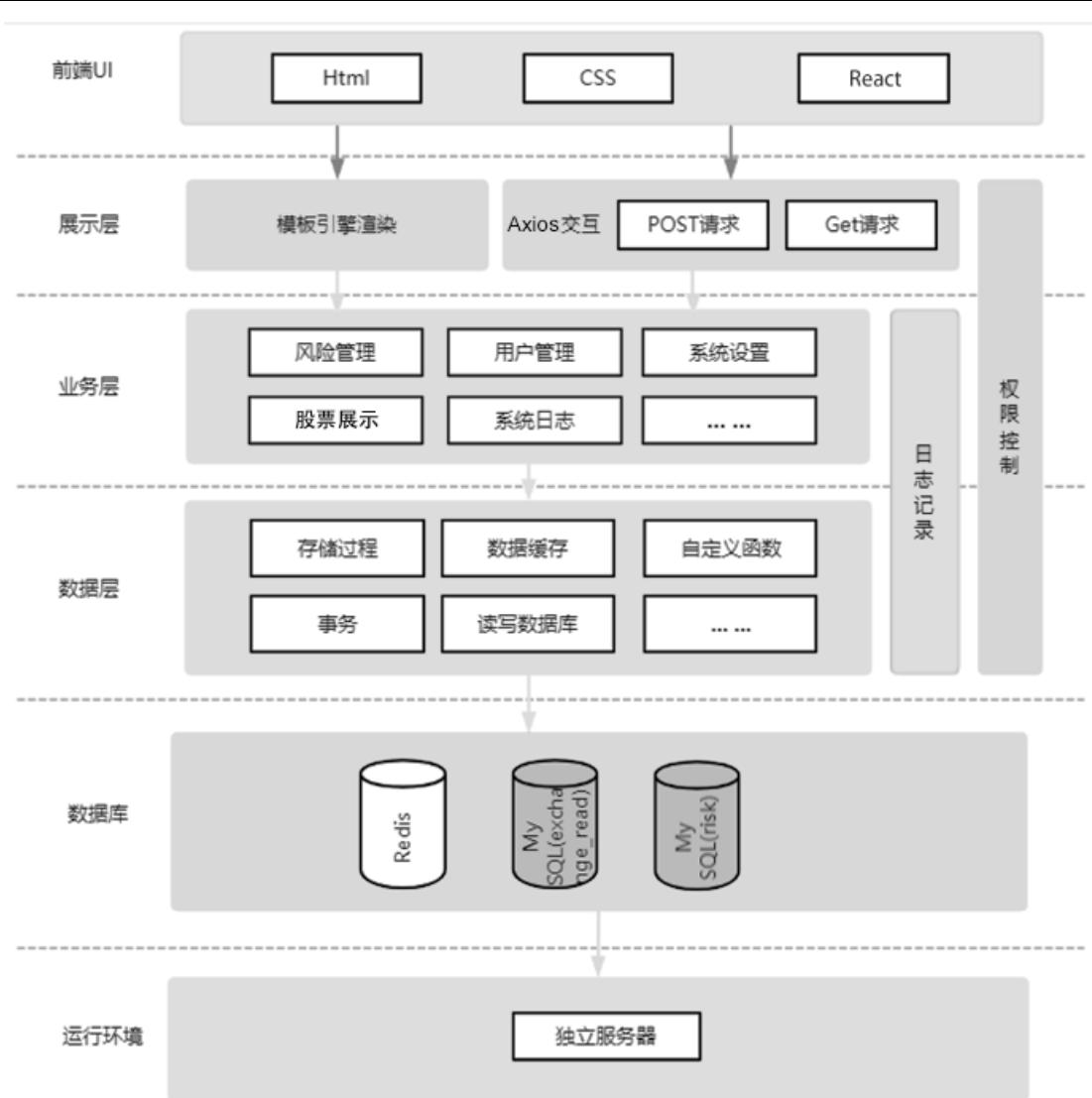


图 3-1 系统架构图

4. 数据设计

4.1 E-R 图及表汇总

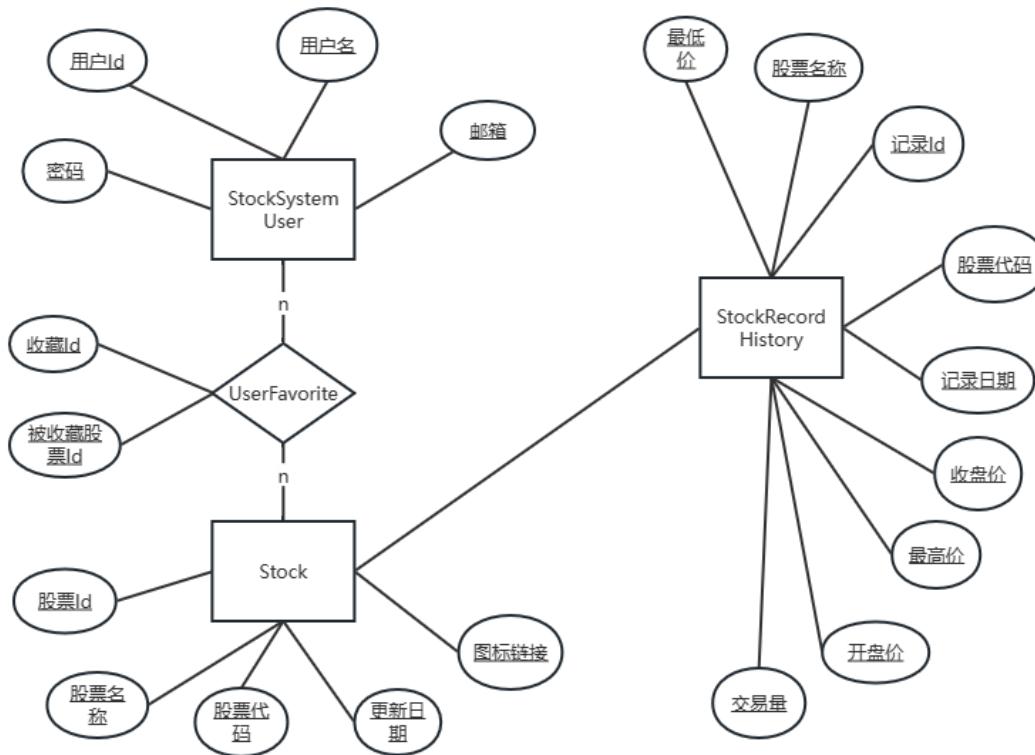


图 3-1 E-R 图

表 3-1 数据库表

表名	中文名称	功能说明
user	用户表	包括基本用户信息及登录相关信息和用户角色
stock	股票表	存储股票的股票 id 等基本信息
stockRecordHistory	股票历史数据表	存储股票历史数据，包括开盘收盘价，历史最低，最高价等
userFavorite	用户收藏表	存储用户收藏的股票数据

4.2 表结构设计

表 4-2 用户表

表名	user			
列名	描述	数据类型(精度范围)	空/非空	约束条件
user_id	用户 id	int	非空	主键, 自增
password	用户登录密码	String		
user_name	用户名	String		
email	用户邮箱地址	String		
补充说明				

表 4-3 股票表

表名	stock			
列名	描述	数据类型(精度范围)	空/非空	约束条件
stock_id	股票 id	int	非空	主键, 自增
stock_code	股票代码	String		
stock_name	股票名	String		
update_date	数据更新日期	Date		
stock_avg	股票图标链接	String		
补充说明				

表 4-3 股票历史数据表

表名	stockRecordHistory			
列名	描述	数据类型(精度范围)	空/非空	约束条件

stockRecord_id	股票历史数据 id	int	非空	主键, 自增
stock_code	股票代码	String		
Stock_name	股票名称	String		
Record_date	记录日期	Date		
Open_price	开盘价	BigDemical		
Close_price	收盘价	BigDemical		
Lowest_price	最低价	BigDemical		
Highest_price	最高价	BigDemical		
volume	交易量	double		
补充说明				

表 4-3 用户收藏表

表名	userFavorite			
列名	描述	数据类型（精度范围）	空/非空	约束条件
fav_id	收藏关系 id	int	非空	主键, 自增
stock_code	收藏股票代码	String		
补充说明				

5. 功能设计

5.1 用户管理

5.1.1 【SDD-001】登录

5.1.1.1 功能概述

用户可以通过系统进行登录，用户在对应输入框中输入账号（用户名）和密码，确定后将数据提交到服务器验证，验证成功后进入到对应角色的系统管理界面，验证失败提示用户重新输入或找回密码，并清空输入框内的内容。

5.1.1.2 输入、处理和输出

输入：用户的用户名和密码。

处理：

1) 前端数据验证模块：系统检查用户的输入是否合法，如是否为空、是否符合格式等。

2) 后端登录处理模块：系统查询数据库，比对用户的用户名和密码，如匹配则登录成功，如不匹配则登录失败。

3) 前端页面模块：负责设计和实现用户登录页面，包括数据验证和页面提示等功能。

输出：用户成功登录系统，可以进行各种操作。

5.1.1.3 功能设计

1) 前端页面设计：我们需要设计一个用户登录页面，其中包含两个输入框（账号、密码）和一个登录按钮。同时，需要加入一些数据验证，确保用户输入的账号和密码符合要求。如果验证失败，需要在页面上提示用户重新输入或找回密码。

2) 后端处理设计：后端处理需要完成以下几个功能：
a) 接收前端传来的账号和密码；
b) 验证账号和密码是否匹配，如果匹配则登录成功，否则登录失败；

c) 系统应有一定的安全防护机制，如密码强度限制、登录次数限制、异常登录检测等，以确保用户信息安全。

3) 数据传输设计：前端通过 HTTP 协议将数据传输到后端进行处理。具体来说，前端通过 Ajax 技术向后端发送一个 POST 请求，请求中包含用户名和密码。后端接收到请求后，对用户名和密码进行验证，如果验证通过则返回一个包含用户信息的 JSON 对象。

4) 模块设计：

- a) 数据验证模块：负责验证用户输入的账号和密码是否符合要求；
- b) 登录处理模块：负责接收前端传来的数据，验证账号和密码是否匹配，并在验证通过后将用户信息保存到数据库中；
- c) 前端页面模块：负责设计和实现用户登录页面，包括数据验证和页面提示等功能。

5.1.1.4 业务约束条件

- 1) 用户名和密码均为必填项，系统不允许空输入。
- 2) 用户名和密码应符合预设的格式，如长度、字符类型等，不符合要求的输入将被拒绝。
- 3) 系统应保证用户的用户名和密码信息存储在安全可靠的数据库中，且数据库应具有一定的备份机制。
- 4) 登录失败的错误信息应明确具体，方便用户了解原因并进行修正。
- 5) 登录成功后，用户应能够顺利地进入系统主界面进行相关操作。
- 6) 在用户未登录或登录过程中，系统应严格保护用户的隐私信息，防止信息泄露或被他人非法使用。
- 7) 系统应有一定的安全防护机制，如密码强度限制、登录次数限制、异常登录检测等，以确保用户信息安全。

5.1.2 【SDD-002】注册

5.1.2.1 功能概述

用户可以通过邮箱并且填写相关信息注册账号。

5.1.2.2 输入、处理和输出

输入：用户的注册信息，如用户名、密码、邮箱、手机号等。

处理：

- 1) 前端输入检查模块：将用户输入的数据以 JSON 格式发送给后端校验模块。
- 2) 后端校验模块：将用户注册信息存储到数据库中，并返回生成的验证码和注册结果给发送邮件模块和前端反馈模块。
- 3) 发送邮件模块将生成的验证码和用户注册邮箱发送到用户邮箱中。用户在邮箱中点击激活链接后，邮件验证模块接收到请求，并进行验证码验证，验证结果返回给前端反馈模块。
- 4) 前端反馈模块：根据后端返回的结果提示用户进行下一步操作。

输出：用户可以激活账号并返回登录界面。

5.1.2.3 功能设计

- 1) 前端输入检查模块：检查用户输入是否为空、是否符合格式要求等，并在本地进行基本验证，避免不必要的网络请求。
- 2) 后端校验模块：接收前端发送的注册请求，进行用户是否已经存在的验证，如果已经存在则返回错误信息，如果不存在则将用户信息存储到数据库中，生成随机的验证码并发送给用户的注册邮箱。
- 3) 发送邮件模块：接收后端生成的验证码和用户的注册邮箱，发送验证邮件给用户。
- 4) 邮件验证模块：接收用户点击邮件中的激活链接后的请求，验证验证码是否正确，如果正确则将用户的状态改为已激活，否则返回错误信息。
- 5) 前端反馈模块：接收后端返回的注册结果，如果注册成功则提示用户前往邮箱激活账号，如果失败则提示错误信息。

5.1.1.4 业务约束条件

- 1) 用户名、密码、邮箱、手机号等必须符合系统规定的格式和长度要求，否则系统会提示用户重新填写正确的信息；
- 2) 用户名、邮箱和手机号必须是唯一的，不能与已存在的用户信息重复，

否则系统会提示用户更换不同的用户名、邮箱或手机号；

3) 用户完成验证后，可以通过用户名和密码登录系统，否则系统会提示用户账号未激活；

4) 用户注册失败时，系统会提示用户失败原因，并提供相应的解决方案，如更换用户名或联系客服；

5) 系统会对用户注册信息进行加密存储和传输，保障用户信息的安全性；

5.1.3 【SDD-003】注销账号

5.1.3.1 功能概述

用户可以通过系统进行注销账号，输入用户名和密码，系统验证用户的身份，如正确则允许用户注销账号，如错误则提示用户重新输入或找回密码。注销账号后，用户的所有信息将被删除，无法再次登录系统。

5.1.3.2 输入、处理和输出

输入：用户的用户名和密码

处理：

1) 前端输入检查模块：将用户输入的数据以 JSON 格式发送给后端校验模块。

2) 后端校验模块：将用户注册信息存储到数据库中，并返回生成的验证码和注册结果给发送邮件模块和前端反馈模块。

3) 发送邮件模块：将生成的验证码和用户注册邮箱发送到用户邮箱中。用户在邮箱中点击激活链接后，邮件验证模块接收到请求，并进行验证码验证，验证结果返回给前端反馈模块。

4) 前端反馈模块根据后端返回的结果提示用户进行下一步操作。

输出：用户无法再次使用该账号登录系统。

5.1.3.3 功能设计

1) 前端输入检查模块：检查用户输入是否为空、是否符合格式要求等，并在本地进行基本验证，避免不必要的网络请求。

-
- 2) 后端注销校验模块：接收前端发送的注销请求，进行用户身份验证，比对用户名和密码，如匹配则进行注销操作，如不匹配则返回错误信息。
 - 3) 前端反馈模块：接收后端返回的注销结果，如果注销成功则提示用户已注销账号并跳转到登录界面，如果失败则提示错误信息。

5.1.3.4 业务约束条件

- 1) 用户名和密码是必填项，系统需要对用户输入进行验证，确保其不为空且符合格式要求；
- 2) 用户名和密码需要与系统中保存的数据匹配，否则无法进行注销操作；
- 3) 注销操作成功后，用户无法再次使用该账号登录系统；
- 4) 系统需要向用户及时反馈注销结果，包括注销成功或失败等信息；
- 5) 用户在注销前需要确保自己的账户状态正常，否则可能导致注销失败；
- 6) 系统需要保证用户的隐私信息不泄露，包括用户的用户名和密码等敏感信息；
- 7) 系统需要保证注销操作的安全性，防止非法用户通过该操作恶意攻击系统

5.1.4 【SDD-004】修改密码

5.1.4.1 功能概述

用户在账号管理界面中可修改密码。输入当前密码、新密码和确认密码。若原密码错误则无法修改。若当前密码和确认密码不匹配，会提示输入错误，系统即时显示。而不是等到填写完新密码后提交表单时才显示错误。新密码的长度要求 6 位及以上的数字或字母。

5.1.4.2 输入、处理和输出

输入： 用户输入当前密码、新密码、确认密码，新密码的长度要求 6 位及以上的数字或字母。

处理：

-
- 1) 前端输入检查模块：接收用户输入的当前密码、新密码和确认密码，并进行基本的输入格式验证和密码规则验证，如果验证通过则将数据以 JSON 格式发送给后端处理模块，否则提示用户错误信息。
 - 2) 后端处理模块：接收前端发送的密码修改请求，查询数据库比对用户当前密码是否正确，如正确则更新用户密码为新密码，如不正确则返回错误信息。并将处理结果返回给前端反馈模块。
 - 3) 前端反馈模块：接收后端返回的修改结果，如果修改成功则提示用户修改成功并跳转到登录页面，如果失败则提示错误信息。

输出：

- 1) 弹出修改成功提示框
- 2) 前端系统即时显示错误提示

5.1.4.3 功能设计

- 1) 前端输入检查模块：检查用户输入的数据是否合法，如是否为空、是否符合格式要求等，并在本地进行基本验证，避免不必要的网络请求。
- 2) 后端处理模块：接收前端发送的密码修改请求，查询数据库比对用户当前密码是否正确，如正确则更新用户密码为新密码，如不正确则返回错误信息。
- 3) 前端反馈模块：接收后端返回的修改结果，如果修改成功则提示用户修改成功并跳转到登录页面，如果失败则提示错误信息。

模块数据交互：

前端输入检查模块将用户输入的数据以 JSON 格式发送给后端处理模块。
后端处理模块将用户密码更新到数据库中，并返回修改结果给前端反馈模块。
前端反馈模块根据后端返回的结果提示用户进行下一步操作。

5.1.4.4 业务约束条件

- 1) 当前密码输入框、新密码输入框、确认密码输入框必须存在，并且不能为空。
- 2) 新密码必须是 6 位及以上的数字或字母组成。
- 3) 当前密码输入框中输入的密码必须与数据库中该用户的密码匹配，否则无法修改密码。
- 4) 确认密码输入框中输入的密码必须与新密码输入框中输入的密码相同，

6) 否则会提示错误信息并且系统即时显示错误。

5.2 历史数据

可查询和查看股票的每日历史交易数据，包括开盘价、收盘价等，还可自行创建股票录入数据。

5.2.1 【SDD-005】查看股票

5.2.1.1 功能概述

用户可以查看股票的历史交易数据，包括开盘价、收盘价等。用户在输入框中输入股票代码或股票名字，确定后将数据提交到服务器进行搜索，将搜索出的结果返回给用户。

5.2.1.2 输入、处理和输出

输入：用户的股票代码或股票名字。

处理：

- 1) 前端数据验证模块：系统检查用户的输入是否合法，如是否为空、是否符合格式等。
- 2) 后端搜索处理模块：系统查询数据库或第三方数据源，获取用户输入的股票的历史交易数据，如有则返回给前端，如无则提示用户无此股票。
- 3) 前端页面模块：负责设计和实现用户查看股票数据的页面，包括数据展示和页面提示等功能。

输出：用户成功查看到股票的历史交易数据，可以进行分析或其他操作。

5.2.1.3 功能设计

- 1) 前端页面设计：设计用户查看股票数据的页面，包含股票代码或名字的输入框和搜索按钮。加入数据验证，确保输入符合要求。若验证失败，需提示用户重新输入或选择其他股票。
- 2) 后端处理设计：完成以下功能：

-
- a)接收前端传来的股票代码或名字;
 - b)查询数据库或第三方数据源，获取股票的历史交易数据。若查无此股票，提示用户；
 - c)系统应有缓存机制，以提高查询效率和减少压力。
- 3) 数据传输设计：前端通过 HTTP 协议将数据传输到后端进行处理。前端通过 Ajax 技术向后端发送 GET 请求，请求中包含股票代码或名字。后端返回包含股票历史交易数据的 JSON 对象。
- 4) 模块设计：
- a)数据验证模块：验证用户输入的股票代码或名字是否符合要求；
 - b)搜索处理模块：接收前端传来的数据，查询数据库或第三方数据源，获取股票的历史交易数据，并返回给前端；
 - c)前端页面模块：设计实现用户查看股票数据的页面，包括数据展示和页面提示等功能。

5.2.1.4 业务约束条件

- 1) 股票代码或名字为必填项，不允许空输入。
- 2) 股票代码或名字应符合预设格式，如长度、字符类型等。不符合要求输入将被拒绝。
- 3) 系统应确保从数据库或第三方数据源获取股票历史交易数据准确、及时、与实际市场情况一致。
- 4) 搜索失败的错误信息应明确具体，方便用户了解原因和修正。
- 5) 搜索成功后，用户应能清晰地查看股票历史交易数据，包括开盘价、收盘价等。用户可进行排序、筛选、导出等操作。
- 6) 系统应严格保护用户隐私信息，防止信息泄露或被他人非法使用，无论用户是否处于登录状态。

输出：用户成功查看到股票的历史交易数据，可以进行分析或其他操作。

5.2.2 【SDD-006】创建股票

5.2.2.1 功能概述

用户可自行创建股票并录入相关信息。输入股票代码、名字、类型、发行价

等信息后，确认数据并提交到服务器进行保存。创建成功后，系统提示用户并显示新创建的股票信息。

5.2.2.2 输入、处理和输出

输入： 用户股票代码、名字、类型、发行价等信息。

处理：

1) 前端数据验证模块：验证用户输入是否合法，如是否为空、是否符合格式等。

2) 后端保存处理模块：检查用户输入的股票代码是否已存在于数据库中。若不存在，则将新创建的股票信息保存到数据库中并返回成功信息给前端；若已存在，则提示用户该股票已存在并拒绝保存。

3) 前端页面模块：设计和实现用户创建股票信息的页面，包括数据验证和页面提示等功能。

输出： 用户成功创建新的股票，并可以查看新创建的股票信息。

5.2.2.3 功能设计

1) 前端页面设计：设计用户创建股票信息的页面，包含多个输入框（股票代码、名字、类型、发行价等）和一个确定按钮。加入数据验证，确保用户输入符合要求。若验证失败，需提示用户重新输入或修改信息。

2) 后端处理设计：完成以下功能：

a)接收前端传来的各项信息；

b)检查数据库中是否已不存在相同的股票代码。若不存在，则将新创建的股票信息保存到数据库中并返回成功信息给前端。若已存在，则提示用户该股票已存在并拒绝保存；

c)系统应具备一定的安全机制，如防止恶意创建大量无效或重复的股票等。

3) 数据传输设计：前端通过 HTTP 协议将数据传输到后端进行处理。前端通过 Ajax 技术向后端发送 POST 请求，请求中包含各项信息。后端返回一个 JSON 对象，包含成功或失败的信息。

4) 模块设计：

a)数据验证模块：验证用户输入的各项信息是否符合要求；

b)保存处理模块：接收前端传来的数据，检查数据库中是否已不存在相同的股票代码，并根据情况保存或拒绝新创建的股票信息；

c)前端页面模块：设计和实现用户创建股票信息的页面，包括数据验证和页面提示等功能。

5.2.2.4 业务约束条件

- 1) 各项信息为必填项，不允许空输入。
- 2) 各项信息应符合预设格式，如长度、字符类型等。不符合要求的输入将被拒绝。
- 3) 系统应确保新创建的股票信息存储在安全可靠的数据库中，并且不与已有的其他股票冲突。
- 4) 保存失败的错误信息应明确具体，方便用户了解原因并进行修正。
- 5) 保存成功后，用户应能够顺利地查看新创建的股票信息，并可对其进行修改或删除等操作。

5.2.3 【SDD-007】搜索股票：

5.2.3.2 输入、处理和输出

输入：用户的股票 ID 或名字。

处理：

- 1) 前端数据验证模块：系统检查用户的输入是否合法，如是否为空、是否符合格式等。
- 2) 后端搜索处理模块：系统查询数据库，根据用户输入的股票 ID 或名字，匹配相应的股票信息，如有则返回给前端，如无则提示用户无此股票。
- 3) 前端页面模块：负责设计和实现用户搜索股票信息的页面，包括数据验证和页面提示等功能。

输出：用户成功搜索到具体的某只股票的信息，可以进行查看或其他操作。

5.2.3.3 功能设计

- 1) 前端页面设计：我们需要设计一个用户搜索股票信息的页面，其中包含一个输入框（股票 ID 或名字）和一个搜索按钮。同时，需要加入一些数据验证，确保用户输入的股票 ID 或名字符合要求。如果验证失败，需要在页面上提示用

户重新输入或选择其他股票。

2) 后端处理设计:

- a) 接收前端传来的股票 ID 或名字;
- b) 查询数据库, 根据用户输入的股票 ID 或名字, 匹配相应的股票信息, 如果有则返回给前端, 如果无则提示用户无此股票;
- c) 系统应有一定的缓存机制, 以提高查询效率和减少对数据库的压力。

3) 数据传输设计: 前端通过 HTTP 协议将数据传输到后端进行处理。具体来说, 前端通过 Ajax 技术向后端发送一个 GET 请求, 请求中包含股票 ID 或名字。后端接收到请求后, 返回一个包含股票信息的 JSON 对象。

4) 模块设计:

- a) 数据验证模块: 负责验证用户输入的股票 ID 或名字是否符合要求;
- b) 搜索处理模块: 负责接收前端传来的数据, 查询数据库, 匹配相应的股票信息, 并返回给前端;
- c) 前端页面模块: 负责设计和实现用户搜索股票信息的页面, 包括数据验证和页面提示等功能。

5.2.3.4 业务约束条件

- 1) 股票 ID 或名字均为必填项, 系统不允许空输入。
- 2) 股票 ID 或名字应符合预设的格式, 如长度、字符类型等, 不符合要求的输入将被拒绝。
- 3) 系统应保证从数据库获取的股票信息是准确和及时的, 并且与实际情况一致。
- 4) 搜索失败的错误信息应明确具体, 方便用户了解原因并进行修正。
- 5) 搜索成功后, 用户应能够清晰地查看到具体的某只股票的信息, 并且可以对其进行修改或删除等操作。
- 6) 在用户未登录或登录过程中, 系统应严格保护用户的隐私信息, 防止信息泄露或被他人非法使用。

5.3 用户收藏

用户可以收藏股票, 在收藏页面查看已收藏股票的交易数据。

5.3.1 【SDD-008】收藏股票

5.3.1.1 功能概述

用户可以将某只具体的股票收藏并添加到收藏界面。用户在查看股票信息的页面，点击收藏按钮，将股票信息提交到服务器进行保存，保存成功后提示用户收藏成功，并显示新收藏的股票信息。

5.3.1.2 输入、处理和输出

输入：用户的股票信息和收藏按钮。

处理：

- 1) 前端数据验证模块：系统检查用户是否已经登录，如未登录则提示用户先登录，如已登录则继续操作。
- 2) 后端保存处理模块：系统检查用户是否已经收藏过该股票，如未收藏则将该股票信息保存到数据库中，并返回成功信息给前端；如已收藏则提示用户该股票已收藏，并拒绝保存。
- 3) 前端页面模块：负责设计和实现用户收藏股票信息的页面，包括数据验证和页面提示等功能。

输出：用户成功收藏某只具体的股票，并可以查看新收藏的股票信息。

5.3.1.3 功能设计

- 1) 前端页面设计：设计一个用户收藏股票信息的页面，其中包含一个收藏按钮，以及一个显示已收藏的股票信息的列表，并加入数据验证功能。
- 2) 后端处理设计：接收前端传来的股票信息和收藏按钮，检查数据库中是否存在相同的用户和股票的关联记录，如果不存在则将新创建的关联记录保存到数据库中，并返回成功信息给前端；如果已存在则提示用户该股票已收藏，并拒绝保存。
- 3) 数据传输设计：前端通过 HTTP 协议将数据传输到后端处理。具体来说，前端通过 Ajax 技术向后端发送一个 POST 请求，请求中包含股票信息和收藏按钮。后端接收到请求后，返回一个包含成功或失败信息的 JSON 对象。
- 4) 模块设计：设计数据验证模块、保存处理模块和前端页面模块。

5.3.1.4 业务约束条件

- 1) 用户必须先登录才能进行收藏操作，系统不允许未登录的用户进行收藏操作。
- 2) 用户不能重复收藏同一只股票，系统不允许重复的关联记录存在于数据库中。
- 3) 系统应保证新创建的关联记录存储在安全、可靠的数据仓库中，并且不与已有的其他关联记录冲突。
- 4) 出现保存失败的错误信息时应明确具体，方便用户进行修正。
- 5) 用户成功收藏某只股票后应能够顺利地查看新收藏的股票信息，并能够对其进行取消或删除等操作。

5.3.2 【SDD-009】查看收藏

5.3.2.1 功能概述

用户可以在收藏界面查看已收藏股票的交易数据。用户在收藏界面，点击某只股票，将股票信息提交到服务器进行搜索，将搜索出的结果返回给用户。

5.3.2.2 输入、处理和输出

输入：用户的股票信息和查看按钮。

处理：

- 1) 前端数据验证模块：系统检查用户是否已经登录，如未登录则提示用户先登录，如已登录则继续操作。
- 2) 后端搜索处理模块：系统查询数据库或第三方数据源，获取用户点击的股票的交易数据，如有则返回给前端，如无则提示用户无此股票。
- 3) 前端页面模块：负责设计和实现用户查看收藏股票数据的页面，包括数据展示和页面提示等功能。

输出：用户成功查看到收藏股票的交易数据，可以进行分析或其他操作。

5.3.2.3 功能设计

- 1) 前端页面设计：设计一个用户查看收藏股票数据的页面，其中包含一个显示已收藏的股票信息的列表，以及一个查看按钮，并加入数据验证功能。
- 2) 后端处理设计：接收前端传来的股票信息和查看按钮，查询数据库或第三方数据源，获取股票的交易数据，如果有则返回给前端，如果无则提示用户无此股票，并应有一定的缓存机制提高查询效率。
- 3) 数据传输设计：前端通过 HTTP 协议将数据传输到后端进行处理。具体来说，前端通过 Ajax 技术向后端发送一个 GET 请求，请求中包含股票信息和查看按钮。后端接收到请求后，返回一个包含股票交易数据的 JSON 对象。
- 4) 模块设计：设计数据验证模块、搜索处理模块和前端页面模块。

5.3.2.4 业务约束条件

- 1) 用户必须先登录才能进行查看操作，系统不允许未登录的用户进行查看操作。
- 2) 用户必须有收藏的股票才能进行查看操作，系统不允许没有收藏的用户进行查看操作。
- 3) 系统应保证从数据库或第三方数据源获取的股票交易数据是准确和及时的，并且与实际市场情况一致。
- 4) 搜索失败的错误信息应明确具体，方便用户进行修正。
- 5) 用户成功查看到收藏股票的交易数据后，应能够对其进行排序、筛选、导出等操作，以满足不同的分析需求。

5.4 实时数据

用户查看所收藏的股票在一天之内的波动情况与历史数据 k 线图，能够以 csv 的格式导入与导出数据。

5.4.1 【SDD-010】查看实时数据

5.4.1.1 功能概述

在用户收藏界面，用户可以通过点击查看按钮，查看到自己收藏的股票在一

天之内的波动情况。用户在收藏界面，点击某只股票的查看按钮，将股票信息提交到服务器进行搜索，将搜索出的结果返回给用户。

5.4.1.2 输入、处理和输出

输入：用户的股票信息和查看按钮。

处理：

- 1) 前端数据验证模块：系统检查用户是否已经登录，如未登录则提示用户先登录，如已登录则继续操作。
- 2) 后端搜索处理模块：系统查询数据库或第三方数据源，获取用户点击的股票在一天之内的波动情况，如有则返回给前端，如无则提示用户无此股票。
- 3) 前端页面模块：负责设计和实现用户查看实时数据的页面，包括数据展示和页面提示等功能。

输出：用户成功查看到自己收藏的股票在一天之内的波动情况，并可以进行分析或其他操作。

5.4.1.3 功能设计

1) 前端页面设计：我们需要设计一个用户查看实时数据的页面，其中包含一个显示已收藏的股票信息的列表，以及一个查看按钮。同时，需要加入一些数据验证，确保用户已经登录并且有收藏的股票。如果验证失败，需要在页面上提示用户先登录或收藏股票。

- 2) 后端处理设计：后端处理需要完成以下几个功能：
 - a) 接收前端传来的股票信息和查看按钮；
 - b) 查询数据库或第三方数据源，获取股票在一天之内的波动情况，并返回给前端；如果无则提示用户无此股票；
 - c) 系统应有一定的缓存机制，以提高查询效率和减少对数据库或第三方数据源的压力。

3) 数据传输设计：前端通过 HTTP 协议将数据传输到后端进行处理。具体来说，前端通过 Ajax 技术向后端发送一个 GET 请求，请求中包含股票信息和查看按钮。后端接收到请求后，返回一个包含实时波动情况的 JSON 对象。

4) 模块设计：

- a) 数据验证模块：负责验证用户是否已经登录并且有收藏的股票；
- b) 搜索处理模块：负责接收前端传来的数据，查询数据库或第三方数据源，

获取实时波动情况，并返回给前端；

c) 前端页面模块：负责设计和实现用户查看实时波动情况的页面，包括数据展示和页面提示等功能。

5.4.1.4 业务约束条件

1) 用户必须先登录才能进行查看操作，系统不允许未登录的用户进行查看操作。

2) 用户必须有收藏的股票才能进行查看操作，系统不允许没有收藏的用户进行查看操作。

3) 系统应保证从数据库或第三方数据源获取实时波动情况是准确和及时，并且与实际市场情况一致。

4) 搜索失败错误信息应明确具体，方便了解原因并进行修正。

5) 搜索成功后，清晰地展示波动情况，并且可以对其进行排序、筛选、导出等操作。

5.4.2 【SDD-011】查看详情数据

5.4.2.2 输入、处理和输出

输入： 用户选择某只具体的股票和查看按钮。

处理：

1) 前端数据验证模块：系统检查用户是否已经登录，如未登录则提示用户先登录，如已登录则继续操作。

2) 后端搜索处理模块：系统查询数据库或第三方数据源，获取用户选择的股票的历史 k 线图，如有则返回给前端，如无则提示用户无此股票。

3) 前端页面模块：负责设计和实现用户查看详情数据的页面，包括数据展示和页面提示等功能。

输出： 用户成功查看到某只具体的股票的历史 k 线图，并可以进行分析或其他操作。

5.4.2.3 功能设计

1) 前端页面设计：我们需要设计一个用户查看详情数据的页面，其中包含一个显示股票信息的列表，以及一个查看按钮。同时，需要加入一些数据验证，确保用户已经登录并且有选择的股票。如果验证失败，需要在页面上提示用户先登录或选择股票。

2) 后端处理设计：后端处理需要完成以下几个功能：

- a) 接收前端传来的股票信息和查看按钮；
- b) 查询数据库或第三方数据源，获取股票的历史 k 线图，并返回给前端；如果无则提示用户无此股票；
- c) 系统应有一定的缓存机制，以提高查询效率和减少对数据库或第三方数据源的压力。

3) 数据传输设计：前端通过 HTTP 协议将数据传输到后端进行处理。具体来说，前端通过 Ajax 技术向后端发送一个 GET 请求，请求中包含股票信息和查看按钮。后端接收到请求后，返回一个包含历史 k 线图的 JSON 对象。

4) 模块设计：

- a) 数据验证模块：负责验证用户是否已经登录并且有选择的股票；
- b) 搜索处理模块：负责接收前端传来的数据，查询数据库或第三方数据源，获取股票的历史 k 线图，并返回给前端；
- c) 前端页面模块：负责设计和实现用户查看详情数据的页面，包括数据展示和页面提示等功能。

5.4.2.4 业务约束条件

1) 用户必须先登录才能进行查看操作，系统不允许未登录的用户进行查看操作。

2) 用户必须有选择的股票才能进行查看操作，系统不允许没有选择的用户进行查看操作。

3) 系统应保证从数据库或第三方数据源获取的历史 k 线图是准确和及时的，并且与实际市场情况一致。

4) 搜索失败的错误信息应明确具体，方便用户了解原因并进行修正。

5) 搜索成功后，用户应能够清晰地查看到某只具体的股票的历史 k 线图，并且可以进行放大、缩小、导出等操作。

5.5 数据的导入和导出

5.5.1 【SDD-012】股票数据导入

5.5.1.1 功能描述

在用户详情界面，用户可以通过点击上传按钮，将数据以 csv 格式进行导入。

5.5.1.2 输入、处理和输出

输入：用户的股票信息和上传按钮。

处理：

- 1) 前端数据验证模块：系统检查用户是否已登录，如未登录则提示用户先登录，如已登录则继续操作。
- 2) 后端处理模块：根据用户上传的股票信息 csv 文件，加入后端数据库；
- 3) 前端页面模块：负责设计和实现用户上传股票信息的页面，包括数据验证和页面提示等功能。

输出：用户成功上传某只具体的股票信息，并可在云端查看或使用。

5.5.1.3 功能设计

1) 前端页面设计：

设计一个用户导入股票信息的页面，包含上传按钮和显示已导入股票信息的列表。数据验证确保用户已登录并上传有效文件。验证失败将在页面上提示用户登录或重新上传文件。

2) 后端处理设计：

- a) 接收前端传来的文件信息和上传按钮；
- b) 检查文件中是否包含有效股票信息。有效信息将被保存到数据库，并返回成功信息给前端；无效信息将提示用户文件无效拒绝保存；
- c) 系统应有一定的安全防护机制，防止恶意导入大量无效或重复的股票信息等。

3) 数据传输设计：

前端通过 HTTP 协议将数据传输到后端进行处理。前端通过 Ajax 向后端发送 POST 请求，请求中包含文件信息和上传按钮。后端返回 JSON 对象，包含成功或失败信息。

- 4) 模块设计：
 - a) 数据验证模块：验证用户是否已登录并上传有效文件；
 - b) 保存处理模块：接收前端传来的数据，检查文件中是否包含有效股票信息，保存或拒绝新导入的股票信息；
 - c) 前端页面模块：设计和实现用户导入股票信息的页面，包括数据验证和页面提示等功能。

5.5.1.4 业务约束条件

- 1) 用户必须先登录才能进行导入操作，系统不允许未登录用户导入。
- 2) 用户必须上传符合要求的文件，不允许上传空文件或非 csv 格式文件。
- 3) 系统应保证新导入的股票信息存储在安全可靠的数据库中，不与已有其他股票冲突。
- 4) 导入失败错误信息应明确具体，方便用户了解原因并进行修正。
- 5) 导入成功后，用户可查看新导入的股票信息，进行修改或删除等操作。

5.5.2 【SDD-013】股票数据导出

5.5.2.1 功能概述

在股票详情界面，用户可以通过点击下载按钮，将数据以 csv 格式进行导出。用户在详情界面，点击某只具体的股票的下载按钮，将股票信息提交到服务器进行处理，将处理后的结果返回给用户，并自动下载到本地。

5.5.2.2 输入、处理和输出

输入：用户的股票信息和下载按钮。

处理：

- 1) 前端数据验证模块：系统检查用户是否已登录，如未登录则提示用户先登录，如已登录则继续操作。

2) 后端处理模块：根据用户选择的股票信息，生成相应的 csv 文件，并返回给前端；

3) 前端页面模块：负责设计和实现用户导出股票信息的页面，包括数据验证和页面提示等功能。

输出： 用户成功导出某只具体的股票信息，并可在本地查看或使用。

5.5.2.3 功能设计

1) 前端页面设计：

设计一个用户导出股票信息的页面，包含一个显示股票详情的列表和下载按钮。数据验证确保用户已登录并选择了股票。验证失败将在页面上提示用户登录或选择股票。

2) 后端处理设计：

a) 接收前端传来的股票信息和下载按钮；

b) 根据用户选择的股票信息，生成相应的 csv 文件，并返回给前端；

c) 系统应有一定的安全防护机制，防止恶意导出大量无效或重复的股票信息等。

3) 数据传输设计：

前端通过 HTTP 协议将数据传输到后端进行处理。前端通过 Ajax 向后端发送 GET 请求，请求中包含股票信息和下载按钮。后端返回 JSON 对象，包含 csv 文件链接。

4) 模块设计：

a) 数据验证模块：验证用户是否已登录并选择了股票。

b) 处理模块：接收前端传来的数据，根据用户选择生成相应 csv 文件，并返回给前端。

c) 前端页面模块：设计和实现用户导出股票信息的页面，包括数据验证和页面提示等功能。

5.5.2.4 业务约束条件

1) 用户必须先登录才能进行导出操作，系统不允许未登录用户进行导出。

2) 用户必须选择股票才能进行导出操作，系统不允许未选择的用户进行导出。

3) 系统应保证生成的 csv 文件准确及时，并与实际市场情况一致。

-
- 4) 导出失败错误信息应明确具体，方便用户了解原因并对照修正。
 - 5) 导出成功后，用户可自动下载 csv 文件到本地，并可在本地查看或使用。

6. 接口设计

6.1 【SDD-014】内部接口设计

内部接口采用 RESTful 风格编写，由前端子系统发起请求，后端子系统提供功能。以下为内部接口的详细设计说明。

表 6-1 登录接口

接口名称	登录			
接口描述	用于校验用户输入的用户名与密码是否正确、是否允许登录 使用 GET 方法 访问地址为 /api/users/login			
请求参数	字段名称	字段来源	示例	逻辑验证规则
	用户账号	用户在前端页面输入	XiejiahuiSCU@163.com	需符合邮箱地址命名的基本规则
返回参数	字段名称	字段来源	示例	逻辑验证规则
	用户名	后端根据输入值查询数据库所得	Root	用户账号与密码校验失败则为空
	用户 ID	后端根据输入值查询数据库所得	00001	用户账号与密码校验失败则为空
	用户账号	后端根据输入值查询数据库所得	XiejiahuiSCU@163.com	用户账号与密码校验失败则为空

表 6-2 注册接口

接口名称	注册			
接口描述	用于用户注册 使用 POST 方法 访问地址为 api/users/reg			
请求参数	字段名称	字段来源	示例	逻辑验证规则
	用户账号	用户在前端页面输入	XiejiahuiSCU@163.com	需符合邮箱地址命名的基本规则
	密码	用户在前端页面输入，并通过 base64 算法加密	Gfdyw34m4n2mck432(xjh123@\$)	不可为空
	昵称	用户在前端页面输入	root	不可为空
返回参数	字段名称	字段来源	示例	逻辑验证规则
	注册是否成功	后端判断用户输入的信息能否成功注册账号	True/false	注册成功则返回 true，失败则返回 false

表 6-3 修改密码接口

接口名称	修改密码			
接口描述	用于用户修改密码 使用 PUT 方法 访问地址为 api/users/changepwa			
请求参数	字段名称	字段来源	示例	逻辑验证规则
	用户账号	用户在前端页面输入	XiejiahuiSCU@163.com	需符合邮箱地址命名的基本规则
	用户旧密码	用户在前端页面	Gfdyw34m4n2mck432	不可为空

		输入，并通过 base64 算法加密	(xjh123@\$)	
	用户新密码	用户在前端页面输入，并通过 base64 算法加密	2351fd34ssmfremsnf (abcHh&&&)	不可为空
返回参数	字段名称	字段来源	示例	逻辑验证规则
	修改是否成功	后端判断用户输入的信息能否成功修改密码	True/false	修改密码成功则返回 true，失败则返回 false

表 6-4 获取收藏列表接口

接口名称	获取收藏列表			
接口描述	用于获取用户的收藏列表 使用 GET 方法 访问地址为/api/favorites			
请求参数	字段名称	字段来源	示例	逻辑验证规则
	用户账号	从前端的 local Storage 中获取	XiejiahuiSCU@163.com	不可为空
返回参数	字段名称	字段来源	示例	逻辑验证规则
	股票 ID	后端查询得用户收藏的股票的 ID	AAPL	如无记录则返回 null

表 6-5 添加收藏接口

接口名称	添加收藏			
接口描述	用于用户收藏某支股票 使用 PUT 方法			

	访问地址为/api/favorites/{stockCode}			
请求参数	字段名称	字段来源	示例	逻辑验证规则
	用户账号	从前端的 local Storage 中获取	XiejiahuiSCU@163.com	不可为空
	股票 ID	用户点选并由前端获取	AAPL	不可为空
返回参数	字段名称	字段来源	示例	逻辑验证规则
	添加是否成功	后端判断用户是否成功添加收藏该支股票	True/false	添加收藏成功则返回 true, 失败则返回 false

表 6-6 取消收藏接口

接口名称	取消收藏			
接口描述	用于用户取消收藏某支股票 使用 DELETE 方法 访问地址为/api/favorites/{stockCode}			
请求参数	字段名称	字段来源	示例	逻辑验证规则
	用户账号	从前端的 local Storage 中获取	XiejiahuiSCU@163.com	不可为空
	股票 ID	用户点选并由前端获取	AAPL	不可为空
返回参数	字段名称	字段来源	示例	逻辑验证规则
	取消收藏是否成功	后端判断用户是否成功取消收藏该支股票	True/false	取消收藏成功则返回 true, 失败则返回 false

表 6-7 查询股票历史数据接口

接口名称	查询股票历史数据			
接口描述	<p>用于查询股票历史数据</p> <p>使用 GET 方法</p> <p>访问地址为/api/stockhistories/{stockCode}</p>			
请求参数	字段名称	字段来源及描述	示例	逻辑验证规则
	股票 ID	由前端获取	AAPL	不可为空
返回参数	字段名称	字段来源及描述	示例	逻辑验证规则
	股票 ID	后端查询所得	AAPL	
	股票名称	后端查询所得	Apple Inc	
	股票历史交易数据	后端查询所得，包括日期、开盘价、收盘价、最高值、最低值、成交量	[{date:2020-01-01, low:11.99, high:13.02, open:12.45, close:12.25, volume:132434}, {date:2020-01-02, low:11.98, high:13.05, open:12.25, close:12.66, volume:132434},]	

表 6-8 创建股票接口

接口名称	创建股票			
接口描述	<p>用于创建股票</p> <p>使用 POST 方法</p> <p>访问地址为/api/stockhistories</p>			
请求参数	字段名称	字段来源及描述	示例	逻辑验证规则
	股票 ID	由前端获取	AAPL	不可为空

返回参数	字段名称	字段来源及描述	示例	逻辑验证规则
	创建是否成功	后端判断股票 I D 是否合法, 是否能获取其历史数据	True/false	创建成功则返回 true, 失败则返回 false

表 6-9 上传历史数据接口

接口名称	上传历史数据			
接口描述	用于上传某支股票的历史数据 使用 POST 方法 访问地址为/api/stockhistories/upload			
请求参数	字段名称	字段来源及描述	示例	逻辑验证规则
	包含股票数据的 csv 文件	用户上传, 由前端获取		不可为空
返回参数	字段名称	字段来源及描述	示例	逻辑验证规则
	添加是否成功	后端添加数据后, 返回是否添加成功信息	True/false	添加成功则返回 true, 失败则返回 false

表 6-10 查询所有股票实时数据接口

接口名称	查询所有股票实时数据			
接口描述	用于查询所有股票实时数据 使用 GET 方法 返回值为一数组, 以下仅展示单个数组内的元素所包含的字段信息 访问地址为/api/stock			
请求参数	字段名称	字段来源及描述	示例	逻辑验证规则

	无			
返回参数	字段名称	字段来源及描述	示例	逻辑验证规则
	股票 ID	后端查询所得的股票 ID	AAPL	
	股票价格	后端查询所得的对应股票在该时间点的价格	12.02	
	当前时间	后端获取到的股票价格的更新时间	2020-01-01 15:00:00	

表 6-11 查询某支股票实时数据接口

接口名称	查询某支股票实时数据			
接口描述	用于查询某支股票实时数据 使用 GET 方法访问地址为/api/stocks/{stockCode}			
请求参数	字段名称	字段来源及描述	示例	逻辑验证规则
	股票 ID	用户输入，由前端获取	AAPL	不可为空
返回参数	字段名称	字段来源及描述	示例	逻辑验证规则
	股票价格	后端查询所得的对应股票在该时间点的价格	12.02	
	当前时间	后端获取到的股票价格的更新时间	2020-01-01 15:00:00	

表 6-12 查询满足条件的几支股票实时数据接口

接口名称	查询满足条件的几支股票实时数据			
接口描述	用于查询满足条件的几支股票实时数据 使用 GET 方法 返回值为一数组，以下仅展示单个数组内的元素所包含的字段信息 访问地址为 /api/stock/q			
请求参数	字段名称	字段来源及描述	示例	逻辑验证规则
	股票 ID	用户输入，由前端获取	AAPL	
	股票名称	用户输入，由前端获取	Apple	
返回参数	字段名称	字段来源及描述	示例	逻辑验证规则
	股票 ID	后端查询所得的股票 ID	AAPL	
	股票价格	后端查询所得的对应股票在该时间点的价格	12.02	
	当前时间	后端获取到的股票价格的更新时间	2020-01-01 15:00:00	

6.2 【SDD-015】外部接口设计

本系统的股票交易数据依赖于外部接口 Finnhub Stock API 提供的股票价格数据；股票的 logo 依赖于 tradingview API 提供的图片文件。具体需求见表 4-2 与表 4-3

表 6-13 系统外部接口 tradingview API

接口名称	从 tradingview 系统请求股票 logo 图片			
接口描述	用于获取股票所代表的公司的 logo 访问地址为 https://s3-symbol-logo.tradingview.com/			
请求参数	字段名称	字段来源	示例	逻辑验证规则
	股票 ID	所需查询股票的股票代码	AAPL	不可为空
返回参数	状态	数据获取成功与否的状态值	Ok / no_data	若获取成功则返回“ok”，失败则返回“no_data”
	股票 logo	股票所代表的公司的 logo	股票 Logo 的 png 文件	

表 6-14 系统外部接口 tradingview API

接口名称	从 Finnhub 系统请求股票价格数据			
接口描述	用于获取股票历史交易数据与实时交易数据，包括开盘价、收盘价、最高价、最低价等 访问地址为 https://finnhub.io/api/v1/stock/candle			
请求参数	字段名称	字段来源	示例	逻辑验证规则
	起始时间点	根据所需查询的股票数据时间范围，选择当前时间点或此前一段时间的时间点，需转换为 UNIX 时间戳格式	1679476980	不可为空
	结束时间点	根据所需查询的股票数据时间范围，选择当前时间点或此前一段时间的时间点，需转换为 UNIX 时间戳格式	1679649780	不可为空

	股票 ID	所需查询股票的股票代码	AAPL	不可为空
	分辨率	股价信息的时间粒度	1, 5, 15, 30, 60, D, W, M	不可为空
返回参数	状态	数据获取成功与否的状态值	Ok / no_data	若获取成功则返回“ok”，失败则返回“no_data”
	开盘价	所查询的股票在所查询的时间点的开盘价	12.02	
	收盘价	所查询的股票在所查询的时间点的收盘价	12.45	
	最高价	所查询的股票在所查询的时间点的最高价	13.02	
	最低价	所查询的股票在所查询的时间点的最低价	11.99	

6.3 【SDD-016】用户界面设计

使用 antDesign 组件库，实现扁平简约风格，以蓝色为主题色，突出重要元素。

有效地区分视觉层级，根据功能的优先级来确定元素的视觉层级的高低。

6.3.1 屏幕布局

屏幕布局采用响应式布局，能够在不同宽度的电脑屏幕上完美显示，兼容性更强。

下图为响应式布局示意图，左边与右边分别为在电脑屏幕上与手机屏幕上的呈现效果。

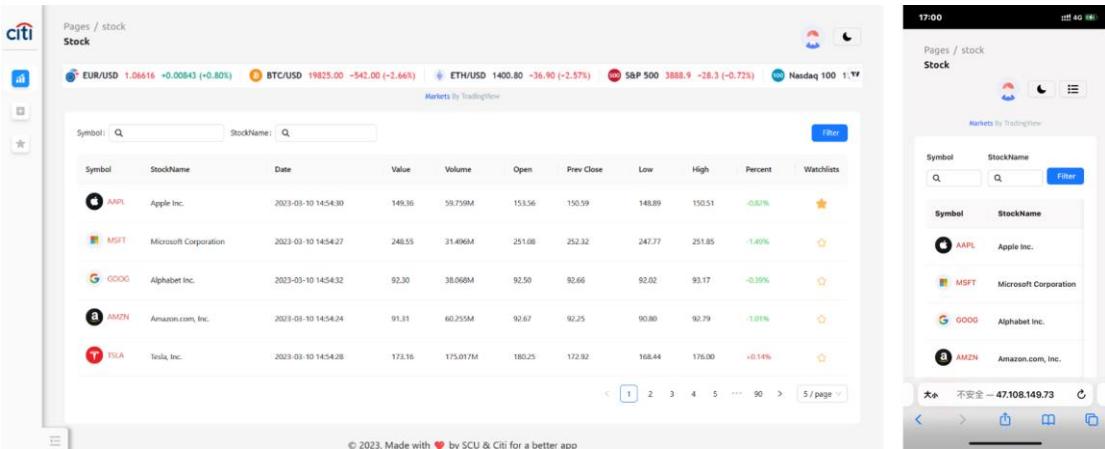


图 6-1 响应式界面设计示意图

6.3.2 显示格式设计

输入框中文字选择左对齐，提示文字为灰色，输入的文字为黑色。

日期格式统一为“yyyy-MM-dd”，具体到天；时间格式统一为“yyyy-MM-dd HH:mm:SS”

输入框中存在不符合规定的内容时，给出错误提示。

当文字过长时，最大宽度为容器的 80%，并且将其余文字显示为省略号或者进行换行。

6.3.3 错误提示设计

系统出现错误时，采取在屏幕上端弹出提示文字的方式进行错误提示。

当页面无法找到时，将链接重定向至 404 页面，为用户提供良好的使用体验

6.3.4 主要功能界面原型图

为明确本系统的前端界面设计，本章节展示了主要功能界面的原型图。

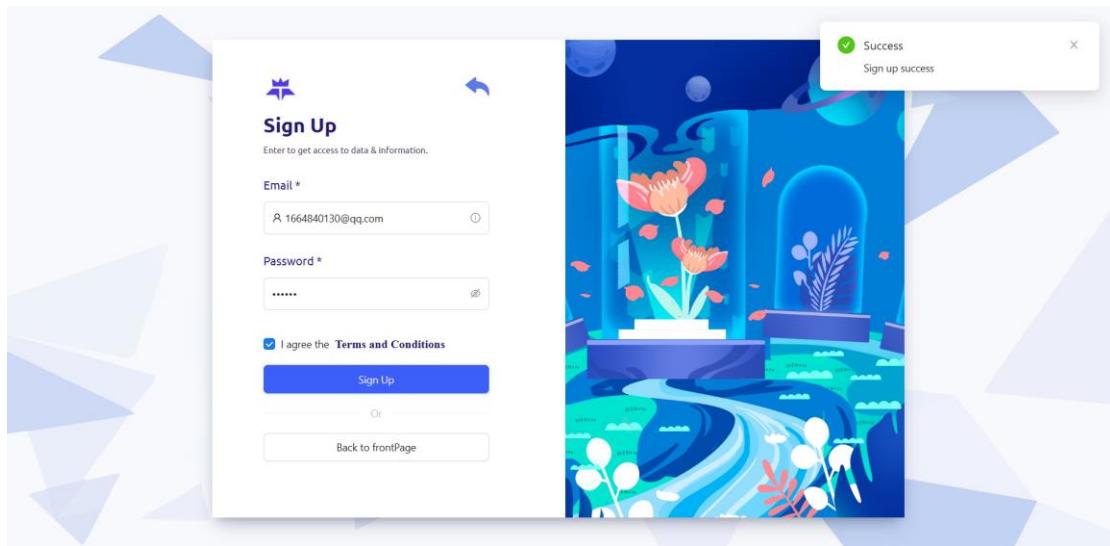


图 6-2 注册界面原型图

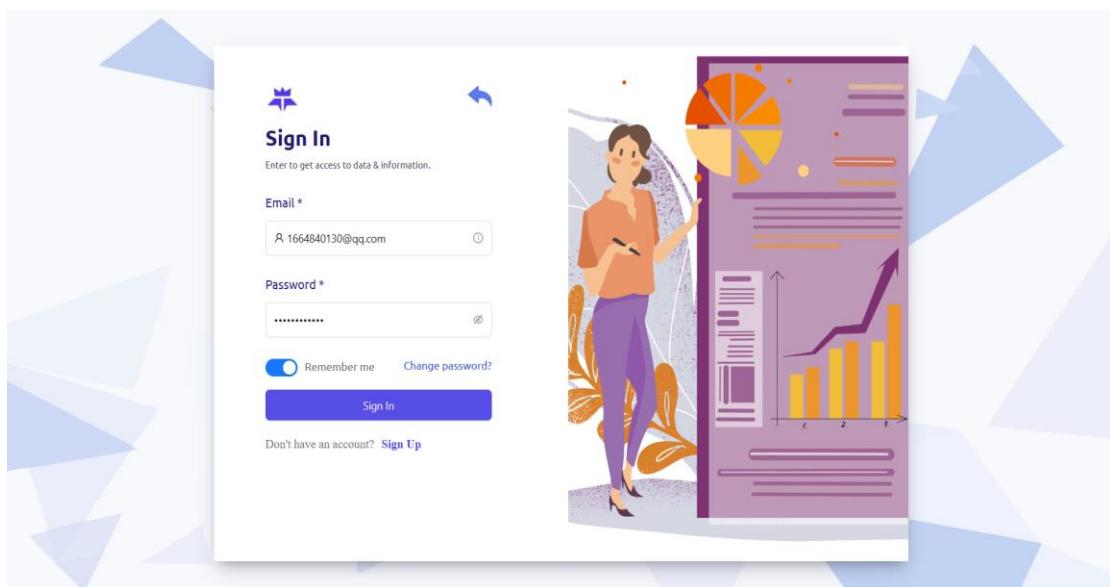


图 6-3 登录界面原型图

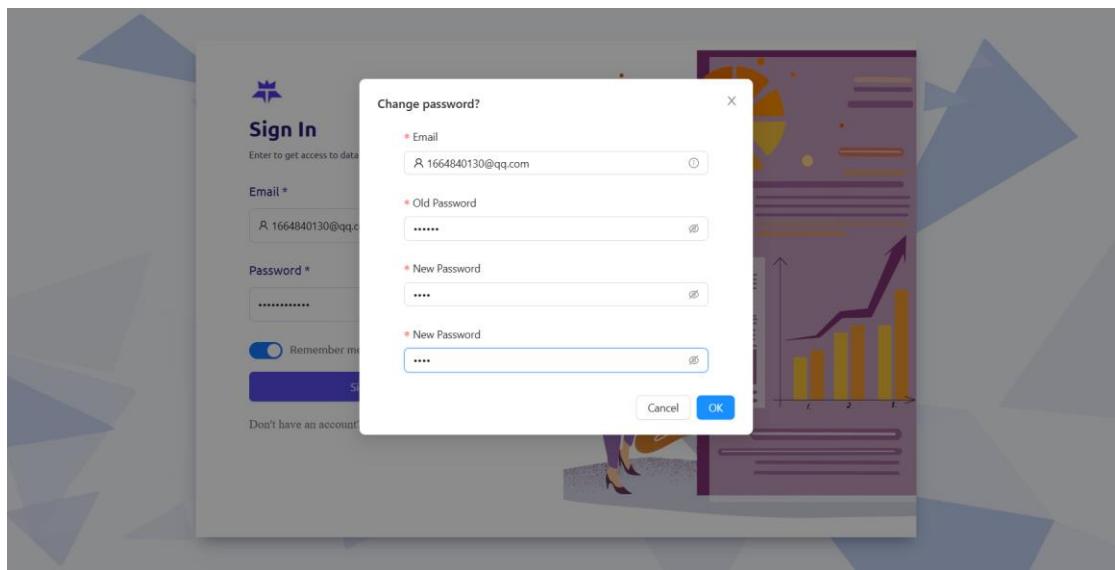


图 6-4 修改密码界面原型图

A wireframe prototype of a stock market interface. The top navigation bar includes the Citi logo, a 'Stocks' button, and 'Create' and 'Favourites' buttons. The main header 'Pages / stock Stock' is followed by a search bar and a 'Symbol' dropdown. The main content area displays a table of stock data:

Symbol	StockName	Date	Value	Volume	Open	Prev Close	Low	High	Percent	Watchlists
AAPL	Apple Inc.	2023-03-10 21:00:05	148.50	59.759M	150.21	150.59	147.61	150.94	-1.39%	
MSFT	Microsoft Corporation	2023-03-10 21:00:05	248.59	31.496M	251.08	252.32	247.60	252.78	-1.48%	
GOOG	Alphabet Inc.	2023-03-10 21:00:04	91.01	38.068M	92.50	92.66	90.80	93.18	-1.78%	
AMZN	Amazon.com, Inc.	2023-03-10 21:00:05	90.73	60.255M	92.67	92.25	90.25	93.57	-1.65%	
TSLA	Tesla, Inc.	2023-03-10 21:00:05	173.44	175.017M	175.13	172.92	168.44	178.29	+0.30%	

The right side of the interface features a sidebar with 'User Info' (Hello, dear user 1664840130@qq.com) and 'Sign out' buttons. At the bottom, there are navigation arrows and a '5 / page' dropdown.

图 6-5 查看股票界面原型图

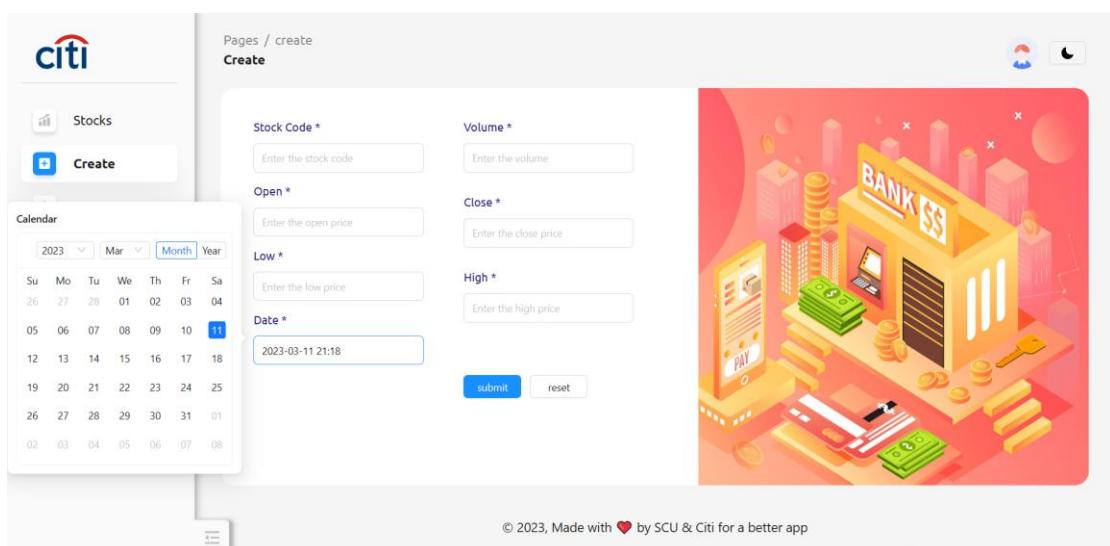


图 6-6 添加股票记录界面原型图



图 6-7 股票历史数据展示界面原型图

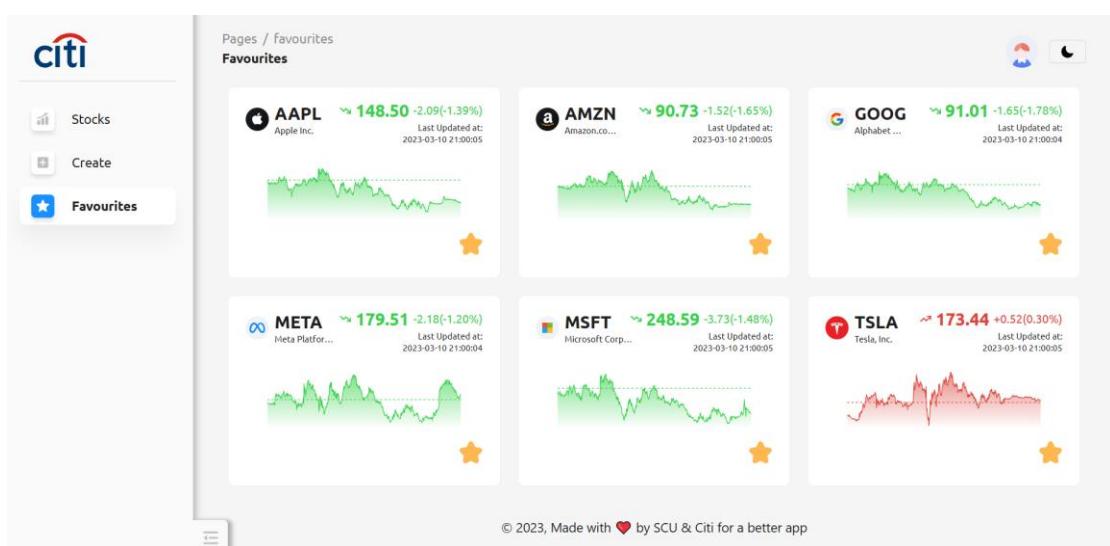


图 6-8 股票实时数据界面原型图

7. 性能设计

7.1 【SDD-017】响应时间性能设计

使用前端构建工具，如 Webpack，对静态资源进行压缩、懒加载和打包，减少不必要的网络请求和提高页面响应速度。

优化前端组件访问后端 API 的方式，尽量减少请求次数，避免不必要的字符串拼接操作，避免在通过 AJAX 请求获取数据时渲染频繁。

利用缓存实现数据存储和读取，如使用 Redis 框架，可以有效提升数据库访问效率，减少不必要的数据库请求和响应时间。

使用前端性能监测工具，如 Google Analytics，可以帮助发现并解决前端性能问题，提高用户体验。

使用前端性能优化技术，如图片懒加载、CDN 加速、减少 HTTP 请求等，可以进一步提高前端性能，加速页面加载速度，提高用户满意度。

7.2 【SDD-018】并发设计

可以使用浏览器缓存和服务器端缓存，减少每次重复请求的流量；利用一些本地存储如 Local Storage 及 Session Storage 来缓存应用数据。

使用可重入锁、读写锁等等，并发安全的数据结构如 ConcurrentHashMap，以减少并发难题产生的因素。

借助 MQ (Message Queue) 技术，将请求发送到队列中，各个程序独立进行计算并返回结果，提高系统运行效率。

使用非阻塞 I/O 方式，降低线程阻塞的风险，优化 IO 效率。

通过不同的压力测试工具进行精细化的压力测试，并发量大的时候观察代码是否存在瓶颈、数据库读写效率是否过低等，发现问题后及时调整优化。

8. 约束限制

8.1 图形要求

- 1) 界面中的文本标签应使用通用词汇，含义准确清晰；
- 2) 界面中的跳转功能应主要采用左侧边栏选择方式，同时配合顶部和按钮切换，直观易懂，以降低人为操作失误、减少手工工作量和规范信息的一致性；
- 3) 数值类型输入部分应使用可编辑的输入框，对内容进行正则校验，以避免不合规则的数值输入；
- 4) 数据过滤应使用分类别过滤选择，以降低查找的难度，减少操作者查找不方便的情况；
- 5) 当数据量较大时，应采用分页浏览的形式。

8.2 报表格式

- 6) 网格区的顶部有报表列名
- 7) 网格区域中显示数据内容
- 8) 网格区底部可以有分页（页数）按钮/信息

9. SDD 和 SRS 对应表

SDD 编号	SRS 编号	名称
SDD-001	SRS-001	登录
SDD-002	SRS-002	注册
SDD-003	SRS-003	注销账号
SDD-004	SRS-004	修改密码
SDD-005	SRS-005	查看股票
SDD-006	SRS-006	创建股票
SDD-007	SRS-007	搜索股票
SDD-008	SRS-008	收藏股票
SDD-009	SRS-009	查看收藏
SDD-010	SRS-010	查看实时数据
SDD-011	SRS-011	查看详情数据
SDD-012	SRS-012	股票数据导入
SDD-013	SRS-013	股票数据导出
SDD-014	SRS-016	内部接口设计
SDD-015		外部接口设计
SDD-016		用户界面设计
SDD-017	SRS-14	响应时间性能设计
SDD-018	SRS-15	并发设计

SDD-019	SRS-017	可靠性设计
SDD-020	SRS-19	安全性设计