**系统设计说明书**

|  |  |
| --- | --- |
| 学 院 | 软件学院 |
| 专 业 | 软件工程（金融科技特色方向） |
| 班 级 | 软金特2301 |
| 学 号 | 20236664 |
| 姓 名 | 郑圆丞 |
| 指导教师 | 李思垚、刘益先 |

目录

[1. 引言 - 1 -](#_Toc18057)

[1.1. 目的 - 1 -](#_Toc18002)

[1.2. 读者对象 - 1 -](#_Toc23791)

[1.3. 术语与缩写解释 - 1 -](#_Toc18142)

[2. 系统总体设计 - 1 -](#_Toc28842)

[3. 功能设计 - 1 -](#_Toc28842)

[4. 工程文件组织结构 - 1 -](#_Toc10665)

[5. 数据库设计 - 1 -](#_Toc21491)

[6. 模块设计 - 1 -](#_Toc11253)

[6.1. 模块内设计类的交互模型 - 1 -](#_Toc23363)

[6.2. 设计类说明 - 1 -](#_Toc13193)

[7. 交互原型设计 - 2 -](#_Toc5016)

[7.1. 总体风格 - 2 -](#_Toc9851)

[7.2. 用户界面清单 - 2 -](#_Toc19342)

[7.3. 页面原型 - 2 -](#_Toc7758)

[8. 选做内容说明 - 2 -](#_Toc5016)

# **引言**

## **目的**

本系统意在为基金投顾业务提供完善的中台的软件系统，整合基金公司原有业务和券商新的投顾业务。改变公募基金长久以来以"卖方代理"模式为主导的市场格局，同时基金销售机构将从过去的"以产品为中心"的销售导向，转向“以客户为中心"的服务导向。对于承载产品销售职能的证券公司来说，可以极大改变现有的基金销售困局，助力其财富管理行业的发展。

编写本文档的用途主要有以下几个方面：

1.详细阐述需求分析与功能设计：该文档详细记录了基金投顾系统的具体需求、功能模块、用户角色与权限等内容，为开发团队提供清晰的功能实现指导。

2.规划技术架构与实现方案：设计说明书中将描述系统的技术架构、数据库设计、接口规范、安全设计等技术细节，确保系统能够满足性能、可扩展性和安全性的需求。

3.作为开发过程中的参考与交流工具：在开发过程中，设计说明书作为开发团队、测试人员、项目经理等不同角色之间沟通的桥梁，确保各方对系统功能、架构和流程的理解一致，避免开发过程中出现偏差。

4.作为后期维护和迭代的基础文档：随着系统的迭代更新，设计说明书为后期维护提供了重要的参考，帮助开发人员快速理解系统结构、功能模块以及业务逻辑。

本文档的主要作用包括：

1.确保系统满足业务需求

设计说明书详细列出了系统需要实现的各项功能，确保开发团队可以根据客户需求和市场趋势设计出符合要求的系统。通过对基金投顾业务的深入分析，该文档帮助开发团队理解业务逻辑，从而设计出高效、符合市场需求的系统。

2.规范化系统开发与实施

该说明书为系统的开发过程提供了规范，涵盖了系统的架构设计、功能模块设计、数据流程、接口设计等关键内容。它确保开发人员能够按照统一的规范进行开发，避免出现不同开发人员之间理解偏差或技术实现不一致的情况。

3.促进多部门协作与沟通

基金业务涉及前台、中台、后台多个层面的功能设计，并且需要与基金公司和券商的现有业务系统进行整合。设计说明书通过详细描述每个模块和功能的实现方式，促进了不同部门（如业务、开发、测试、运营等）之间的沟通与协作，确保系统按期、高质量地交付。

4.支持系统优化与扩展

通过描述系统的技术架构与模块设计，设计说明书为后续的系统优化与扩展提供了基础。随着市场需求变化，证券公司可能需要对基金投顾系统进行功能扩展或优化，设计说明书提供了系统的整体框架和数据流向，为后期的维护和迭代开发提供了参考依据。

5.增强系统的可操作性与易用性

详细的设计说明书可以帮助开发人员在实现系统时充分考虑用户体验和可操作性。例如，在功能设计时，明确不同角色的操作权限和使用场景，使得系统在实施时能够更好地满足最终用户的需求。

## **读者对象**

1. 基金投资顾问：作为系统的核心用户，使用本文档是为了确保工具可靠、高效且符合业务需求。说明书能帮助投顾理解系统的底层逻辑（如因子计算规则、组合优化算法）、数据来源以及功能边界，从而更准确地运用工具构建策略、评估风险，并避免因系统设计缺陷导致投资偏差或合规问题。
2. 投顾管理员：投顾管理员需要查看该软件的系统设计说明书，以确保系统权限设置合理、安全且符合合规要求。说明书会明确不同角色的权限范围，避免越权操作或数据泄露风险。同时，管理员需确认权限分配是否支持业务需求，并能灵活调整以适应组织架构变化，从而保障系统安全、高效运行。
3. 开发团队：开发人员是本文档的重要读者群体，特别是系统架构师、前端开发人员、后端开发人员、数据库设计师等。设计说明书详细描述了系统的技术架构、各功能模块的设计、数据流向、接口规范等，帮助开发团队了解每个功能模块的实现方式，从而准确地进行系统开发。
4. 测试团队：测试人员需要详细了解系统的功能需求、设计和业务逻辑，以便编写相应的测试用例、进行功能验证和系统测试。该文档中对系统各个模块的功能描述，以及风险评估、合规性检查等要求，有助于测试人员全面理解系统行为，并确保系统在不同条件下正常工作。
5. 系统运维人员：运维人员需要了解系统的架构、各模块的协作关系、数据存储结构及安全设计等。本文档将帮助运维团队理解系统的运行机制，从而在系统上线后提供及时的运维支持，确保系统的稳定性、可用性及安全性。
6. 合规与法务团队：合规团队与法务团队需要详细了解系统中的风险控制和合规性设计，以确保基金投顾系统符合相关法律法规和行业标准。文档中对于合规性要求、风险控制流程等方面的描述，有助于合规团队审查系统是否符合法律规定，并及时进行必要的改进。
7. 高层管理人员：高层管理人员，特别是公司高层和决策者，需要从战略角度了解基金投顾系统的整体设计与开发方向。本文档为他们提供了系统的总体架构、业务流程及系统实施带来的潜在收益，帮助其评估系统对公司整体战略目标的支持和推动作用。

## **术语与缩写解释**

金融业务术语

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| 基金 | 基金是一种投资工具，它集合了众多投资者的资金，并由专业的基金经理或管理团队进行管理和运作。这些资金可以被投资于股票、债券、货币市场工具等多种资产，以分散风险并寻求资本增值或稳定收益。基金的种类繁多，包括股票型基金、债券型基金、混合型基金、货币市场基金等，每种基金都有其特定的投资目标和风险收益特征。 |
| 基金公司 | 基金公司，即基金管理公司，是指依法设立，并从事基金募集、基金份额销售、基金财产管理等业务的金融机构。基金公司为投资者提供资产管理服务，投资者通过购买基金份额成为基金的持有人。基金公司需要定期向投资者披露基金的运作情况和业绩表现，并与投资者保持沟通，解答投资者的疑问和关注。 |
| 基金经理 | 基金经理是金融行业中的一个关键职位，在基金公司工作，主要负责管理和运作基金。 |
| 基金画像 | 基金画像是指通过多维度数据分析，对一只基金的投资风格、风险收益特征、持仓结构、运作模式等进行系统性刻画，帮助投资者快速理解基金的本质特征 |
| 投资顾问 | 投资顾问是指为个人或机构投资者提供专业投资建议、资产配置方案和财富管理服务的持牌专业人士或机构。其核心职责是帮助客户实现财务目标，同时平衡风险与收益。 |
| 投顾因子 | 投顾因子是指投资顾问（或智能投顾系统）在提供资产配置建议时，所依赖的关键决策变量或量化指标。这些因子通常结合了传统金融理论、行为金融学以及客户个性化需求，用于优化投资组合的构建和调整。 |
| FOF | FOF（Fund of Funds，基金中的基金）是一种以其他基金（而非直接股票、债券等底层资产）作为主要投资标的的特殊基金类型。其核心逻辑是通过专业化的基金筛选和组合管理，实现风险的二次分散和资产配置优化。 |

软件技术术语

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| Spring Boot | Spring Boot 是一个基于 Java 的开源框架，用于快速构建企业级应用。它简化了 Spring 应用的初始配置，提供自动装配、内嵌服务器（如 Tomcat）和丰富的 Starter 依赖，让开发者能专注于业务逻辑而非繁琐的配置。支持 RESTful API、微服务架构，并整合了 Spring Security、JPA 等生态组件，适合后端开发。 |
| Vue 3​ | Vue 3 是前端流行的 JavaScript 框架，用于构建用户界面和单页应用。它采用组合式 API提升代码组织灵活性，性能优化如虚拟 DOM 重写、Tree-shaking 支持。Vue 3 响应式系统基于 Proxy，支持 TypeScript，生态包含 Vue Router、Pinia（状态管理），适合开发高效、模块化的前端项目。 |
| MySQL​ | MySQL 是广泛使用的关系型数据库，以稳定性、易用性和开源特性著称。支持 SQL 语法、ACID 事务、索引优化和主从复制，适用于结构化数据存储。常与 Spring Boot 搭配（通过 JDBC 或 JPA），或作为 Vue 前端应用的后端数据源。社区版免费，适合中小型项目，也有高性能集群方案。 |

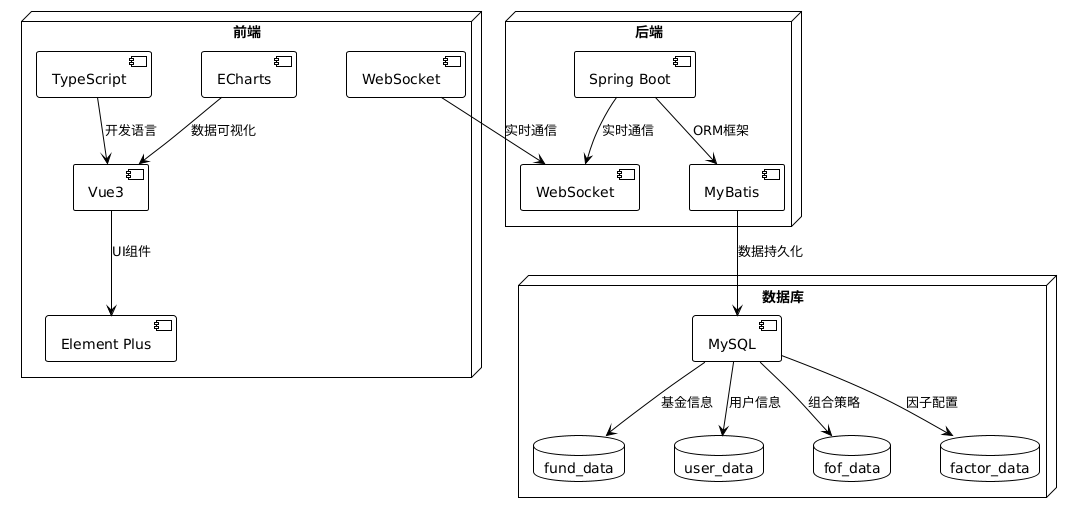
# 系统总体设计

本系统采用基于Vue3+SpringBoot+mySQL前后端分离的核心技术框架开发，兼用WebSocket和RPA技术完善系统功能。

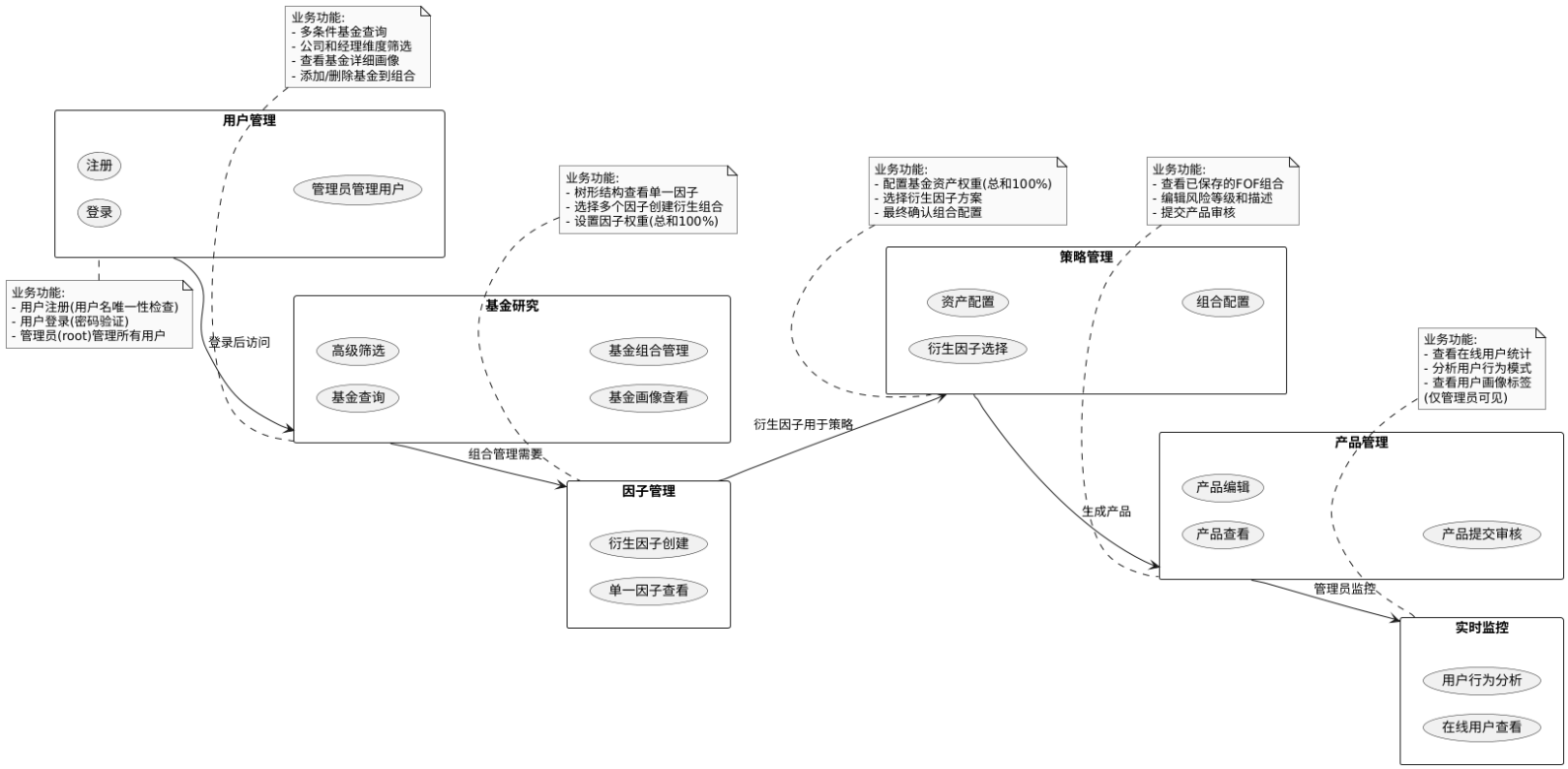
前端：采用Vue3＋element-plus实现与后端的交互及页面的美化。

后端：采用SpringBoot为系统框架，mySQL数据库存储数据，使用RPA技术爬取大量真实业务数据，使用WebSocket实现低延迟检测系统在线用户以及获取用户操作数据。

技术架构图：



业务架构图：



# 功能设计

1、注册，登录：

首先进入系统时需要注册成为用户。注册时系统要求用户名不能重复，会检测注册用户名是否存在，只有检测成功后才能注册成功。登录时需输入正确的用户名和密码即可进入系统。系统初始化了唯一管理员root，密码是123456。

2、基金研究：

在基金研究模块用户可以查看基金代码、基金名称、所属公司名称、托管基金经理名称、单位净值、年化收益率、累计收益率和基金画像，其中点击基金画像可以查看详细的基金描述以及基金持仓明细构成，包括股票、债券和现金。页面支持通过基金代码、基金名称、公司名称和基金经理名称的模糊查询和组合查询，也支持点击筛选功能根据公司和经理信息进行筛选，包括公司规模、公司评级、风险等级、基金经理管理规模、年化回报率、从业经验。它们分别具有不同标签以供筛选，支持组合筛选。用户选中心仪的基金后点击基金选项的添加按钮即可将基金添加到自己的基金组合中，点击删除按钮将基金从组合中删除。页面还支持分页查看基金数目，可以选择一页呈现几条数据以及跳转页面。

3、因子管理：

模块支持查看树形结构的单一因子，用户可通过父类因子选择多个单一因子创建为新的衍生因子组合方案。用户需要填写因子组合名称，并为各个因子设置权重，权重要求加和为100%。

4、策略管理：

该模块细分为三个子模块，首先进入资产配置模块，对本次构建组合的全部基金进行分配权重，要求权重总和必须为100%。点击保存配置后进入选择衍生因子方案模块，从之前所选多组衍生因子中选择一个作为本产品的衍生衍生因子方案。点击保存后进入基金组合配置模块，用户可以查看之前对基金组合配置所有选项，确认无误后可起基金组合名称，随后点击保存即可保存为FOF基金组合方案。

5、组合产品管理：

模块呈现用户保存的所有FOF基金组合方案，信息包括FOF方案名称、DF名称、用户id、状态、风险等级和描述。初始状态为未审核，初始风险等级和描述为空。用户点击操作中的编辑按钮可以选择产品风险等级和编辑产品描述，点击确定后填充页面空白的信息并将本方案上交给审核平台，状态变为审核中。

6、用户管理：

本模块只对系统管理员root开放，可以对注册系统所有用户的用户名和密码进行更改。

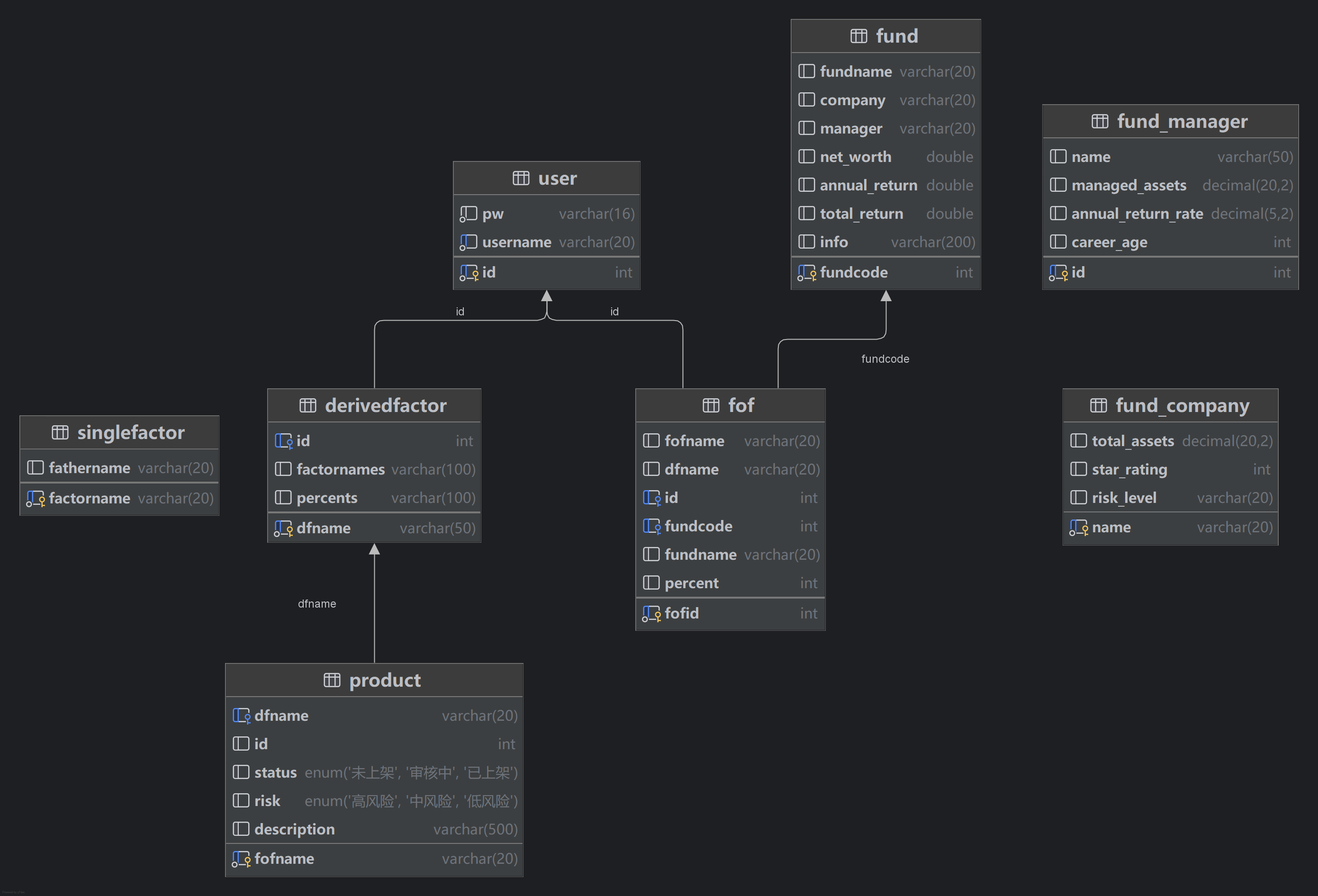
7、（拓展功能——基于大模型的开发）实时监控：

本模块只对系统管理员root开放。管理员可以查看在线用户数量，对用户点击查看详情即可查看该用户的用户行为和用户画像信息，包括：用户id、用户名、角色、在线时长、活跃度、最后操作时间、行为类型分布、时段活跃度、基金选择分布、因子使用分布和画像标签。

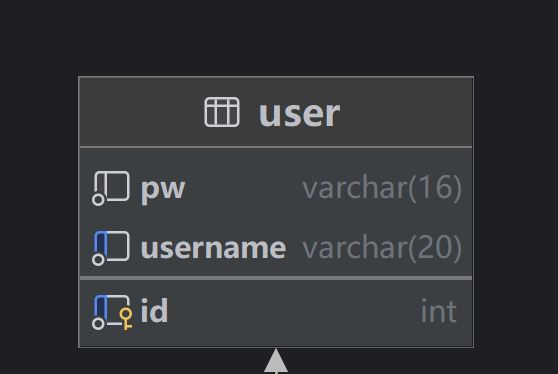
# 工程文件组织结构

# 数据库设计

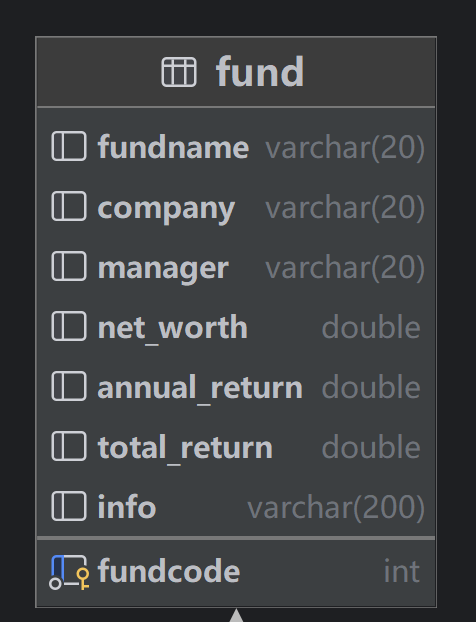
本项目使用了MySQL数据库，总共使用了8个数据库表。表内容与结构如下：



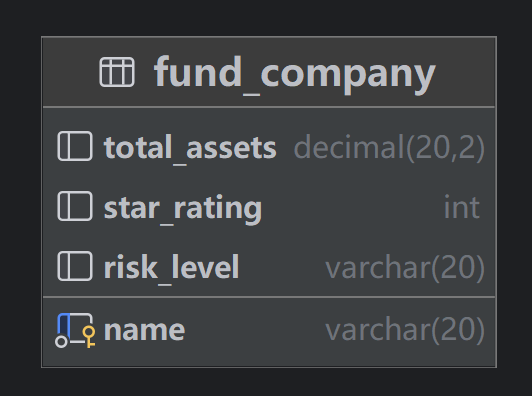
1. user表：用于储存系统用户（即投顾人员和系统管理员）的账号数据。主码为自动生成、自动增加的id，类型为整数，不可修改，作为区分用户的唯一标识。username用于储存用户名称，类型为字符串，可以修改，但是不能重复。pw用于储存用户密码，类型为字符串。



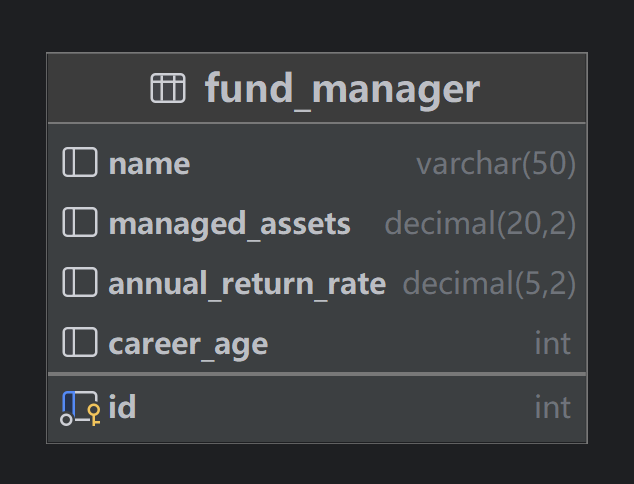
1. fund表：用于储存市场上的基金数据。主码为fundcode，基金代码，类型为int，作为识别基金的唯一标识。fundname为基金名称，company储存基金公司名称，manager储存基金经理名称，其类型均为字符串。net\_worth储存基金净值（单位：亿元），annual\_return储存基金年收益率，total\_return储存基金总收益率，类型均为浮点数。info储存基金信息（基金画像），类型为字符串。



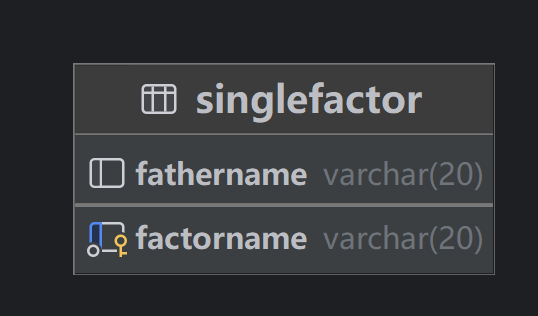
1. fund\_company表：用于储存市场上的基金公司数据。主码为name，储存基金公司名称，类型为字符串。risk\_level储存基金公司风险等级，类型为字符串。star\_rating储存基金公司星级，类型为整数。total\_assets储存基金公司资产（单位：亿元），类型为定点数。



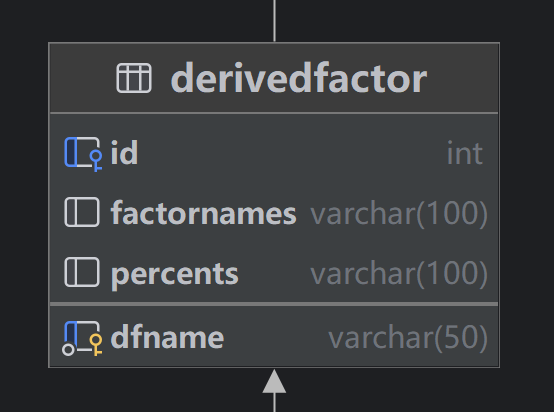
1. fund\_manager表：用于储存基金经理数据。主码为基金经理id，类型为整数。name储存基金经理姓名，类型为字符串。managed\_assets储存基金经理管理的资产金额（单位：亿元），annual\_return\_rate储存基金经理的评价年回报率，二者类型均为定点数。career\_age储存基金经理从业年限，类型为整数。



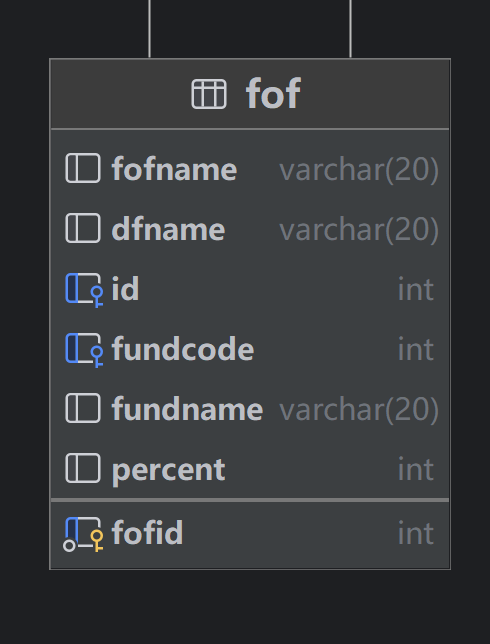
1. singlefactor表：用于储存单一因子与因子树数据。主码为factorname，为单一因子的名称，类型为字符串。fathername储存单一因子的父因子，类型为字符串，多个单一因子有共同的父因子，实现了树形结构。



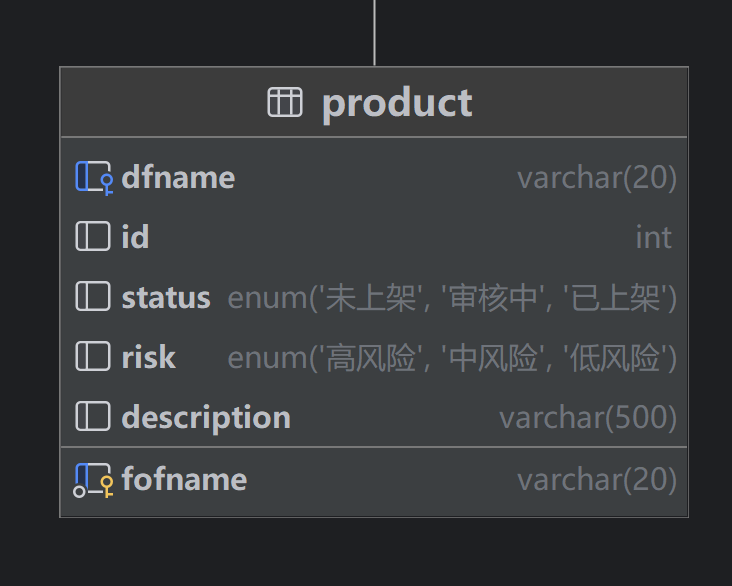
1. derivedfactor表：用于储存投资顾问创建的衍生因子数据。主码为dfname，储存衍生因子的名称，类型为字符串。外码id储存创建该因子的用户id，类型为整数。factornames储存该衍生因子包含的所有单一因子的名称，percents储存所有单一因子的比例，不同的因子间以逗号分隔，二者类型均为字符串。



1. fof表：用于储存投资顾问选择的FOF基金组合、其中每个基金的比例，与对应的衍生因子，基金组合中的每一个基金都作为一条数据单独存储。主码为自动增加的fofid，表示单个基金在基金组合中，类型为整数。外码id储存创建该基金组合的用户id，类型为整数。外码fundcode储存该单个基金的基金代码，类型为整数。fundname储存单个基金的名称，类型为字符串。percent储存该单个基金在基金组合中的占比，类型为整数。fofname储存FOF组合名称，由投资顾问设置，类型为字符串。dfname储存FOF应用的衍生因子名称，类型为字符串。



1. product表：用于储存投资顾问创建的基金组合产品，其名称、衍生因子、产品状态（是否上架与审核）、风险、描述。主码fofname储存该基金组合产品的名称，类型为字符串。外码dfname储存该产品应用的衍生因子名称，类型为字符串。id储存创建该产品的用户id，类型为整数。status储存产品状态，分为‘未上架’、‘审核中’、‘已上架’。risk储存产品风险等级，分为‘高风险’、‘中风险’、‘低风险’。deacription储存产品描述，类型为字符串。



# 模块设计

## **6.1.模块内设计类的交互模型**

*基于模块职责（模块对外接口），识别设计类，创建设计类之间的交互模型。*

## **6.2.设计类说明**

*基于模块内设计类的交互模型创建的模块设计类图*

*<类图详细说明模板（类或接口说明>*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 类名 |  | 所属包 | | | |  | |
| 继承 | 表格内文字，宋体，五号 | | | | | | |
| 实现 |  | | | | | | |
| 属性 | | | | | | | |
| 名称 | 类型 | 默认值 | | | Pub/Prv/Pro | | |
|  |  |  | | |  | | |
|  |  |  | | |  | | |
| 方法 | | | | | | | |
| 名称 | 参数 | | 返回值 | 异常 | | | 描述 |
|  |  | |  |  | | |  |
|  |  | |  |  | | |  |
| 事件 | | | | | | | |
| 名称 | 条件 | | 参数 | 目的 | | | |
|  |  | |  |  | | | |
|  |  | |  |  | | | |

# 交互原型设计

## **7.1.总体风格**

*本节描述本产品所有界面必须统一采用的风格。对应于总体设计阶段的用户界面设计。*

* *以下以Web界面的典型风格说明怎样进行总体风格设计。*

## **7.2.用户界面清单**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **代码** | **名称** | **所属子系统** | **相关组件/模块** | **描述** |
| *UI1.1* | *logon.htm* | */* | */* |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## **3.3.页面原型**

*此处给出页面原型截图*