

恒生第三小组-概要设计说明书

“

金融科技创新训练营

<项目名称>

简易基金理财销售管理系统

概要设计说明书

小组：第三小组

2024年7月

说 明

本文档中所包含的信息属于商业机密信息，如无恒生电子股份有限公司的书面许可，任何人都无权复制或
利用。

目录

1. 引言	
• 1.1 编写目的	
• 1.2 背景	
• 1.3 定义	
• 1.4 法律法规和参考资料	
• 法律法规	
• 需求规格说明书	
• 软件开发标准	
2. 总体设计	
• 2.1 需求概述	
• 2.2 系统分析	
• 系统模块间关系	
• 2.3 模块的层次结构图	
• 2.4 模块与需求对应表	
• 2.5 各模块概要说明	
• 模块概要说明	
• 2.6 系统角色与功能对应关系	
3. 性能设计	
4. 运行设计	
• 4.1 运行模块组合	
• 4.1.1 开户模块组合	
• 4.1.2 申购模块组合	
• 4.1.3 赎回模块组合	
• 4.1.4 清算模块组合	
• 4.1.5 业务查询模块组合	
• 4.1.6 产品管理模块组合	
• 4.2 运行控制	

- 4.2.1 开户模块组合 -----
- 4.2.2 申购模块组合 -----
- 4.2.3 赎回模块组合 -----
- 4.2.4 清算模块组合 -----
- 4.2.5 业务查询模块组合 -----
- 4.2.6 产品管理模块组合 -----

5. 错误处理 -----

- 5.1 错误发现 -----
- 5.2 错误解决 -----

6. 接口设计 -----

- 6.1 用户界面 -----
 - 6.1.1 产品管理模块 -----
 - 6.1.2 开户模块 -----
 - 6.1.3 申购模块 -----
 - 6.1.4 赎回模块 -----
 - 6.1.5 清算模块 -----
 - 6.1.6 业务查询模块 -----
 - 6.1.7 登录界面 -----
- 6.2 外部接口 -----
 - 6.2.1 第三方支付网关接口 -----
 - 6.2.2 身份验证服务接口 -----
 - 6.2.3 市场数据接口 -----
 - 6.2.4 风险评估服务接口 -----
 - 6.2.5 法规遵从性报告接口 -----
- 6.3 内部接口 -----
 - 6.3.1 产品管理模块接口 -----
 - 6.3.1.1 新增产品 -----
 - 6.3.1.2 删除产品 -----
 - 6.3.1.3 查询产品 -----
 - 6.3.1.4 修改产品 -----
 - -----

6.3.2 开户模块接口

- 6.3.2.1 开户 -----

• 6.3.3 申购模块接口 -----

- 6.3.3.1 申购 -----

• 6.3.4 赎回模块接口 -----

- 6.3.4.1 赎回 -----

• 6.3.5 清算模块接口 -----

- 6.3.5.1 清算操作 -----

• 6.3.6 业务查询模块接口 -----

- 6.3.6.1 交易查询 -----

• 6.3.7 登录模块接口 -----

- 6.3.7.1 用户登录 -----

7. 约束条件 -----

8. 系统数据结构设计 -----

- 8.1 设计要点 -----

- 8.2 详细设计 -----

- 8.3 数据结构和模块的关系 -----

9. 功能模块设计 -----

- 9.1 模块A：产品管理 -----

- 9.1.1 A1: 产品信息呈现 -----

- 9.1.2 A2: 产品新增/删除 -----

- 9.1.3 A3: 产品编辑 -----

- 9.1.4 A4: 产品详情查看 -----

- 9.1.5 A5: 产品净值走势图 -----

- 9.2 模块B：开户 -----

- 9.2.1 B1: 客户列表 -----

- 9.2.2 B2: 查询客户信息 -----

- 9.2.3 B3: 录入客户基本信息 -----

- 9.2.4 B4: 录入客户银行卡信息 -----

- 9.2.5 B5: 客户风险承受能力 -----

- 9.2.6 B6: 客户银行卡余额 -----
- 9.2.7 B7: 开户进程显示 -----
- 9.3 模块C: 申购 -----
 - 9.3.1 C1: 申购申请 -----
 - 9.3.2 C2: 搜索 -----
 - 9.3.3 C3: 风险等级确认 -----
 - 9.3.4 C4: 下单成功界面 -----
- 9.4 模块D: 赎回 -----
 - 9.4.1 D1: 赎回申请 -----
 - 9.4.2 D2: 搜索 -----
 - 9.4.3 D3: 下单成功 -----
- 9.5 模块E: 清算 -----
 - 9.5.1 E1: 日初初始化 -----
 - 9.5.2 E2: 交易确认功能 -----
 - 9.5.3 E3: 导出申请数据 -----
 - 9.5.4 E4: 清算日志 -----
- 9.6 模块F: 业务查询 -----
 - 9.6.1 F1: 交易查询 -----
 - 9.6.2 F2: 撤单 -----
- 9.7 模块G: 登录注册 -----
 - 9.7.1 G1: 登录界面 -----
 - 9.7.2 G2: 注册界面 -----

1. 引言

1.1 编写目的

本需求规格文档旨在详细描述和定义我们正在开发的理财销售管理系统的功能和需求。这份文档将作为项目开发团队的指导手册，确保所有参与者对项目需求有一致的理解，并提供明确的参考，以便在设计、开发、测试和维护过程中遵循。具体来说，编写目的包括：

- 1. 明确系统的功能和非功能需求：**详细列出理财销售管理系统应具备的各项功能，以及系统的性能、可靠性、可维护性等非功能需求。功能需求具体包括支持基金产品基本信息维护、客户开户、客户申购基金、客户赎回基金、清算、业务查询等。
- 2. 定义系统的用户接口和外部接口：**描述系统将如何与用户和其他系统交互，确保接口设计的清晰性和一致性。例如，系统需要提供用户友好的界面，支持与银行、支付平台等外部系统的集成。
- 3. 指导系统的设计和实现：**为开发团队提供详细的需求说明，使他们能够依据这些需求进行系统的设计和实现。这包括系统的架构设计、数据库设计、模块划分等。
- 4. 提供测试和验证的基础：**为测试团队提供明确的测试用例和标准，以验证系统是否符合需求。测试内容将涵盖功能测试、性能测试、安全测试等，确保系统的稳定性和安全性。
- 5. 形成项目管理的依据：**为项目经理和其他利益相关者提供需求的清晰描述，以便进行项目规划、进度控制和资源分配。明确的需求描述有助于项目各方在项目各阶段进行有效的沟通和协调。

通过该文档的编写，我们希望能够确保系统开发的高效性和准确性，最终交付一个符合预期、满足用户需求的高质量理财销售管理系统。

该系统的目标包括：

- **提高销售效率：**通过优化的客户管理和销售流程，帮助理财顾问提高工作效率。
- **增强客户满意度：**提供个性化的理财建议和服务，提升客户体验。
- **确保数据安全：**采用先进的安全技术，确保客户数据和财务信息的安全。
- **提供决策支持：**通过数据分析和报表生成功能，为管理层提供决策支持。

本需求规格文档将作为理财销售管理系统开发过程中的重要参考资料，确保项目的顺利进行和成功交付。

1.2 背景

项目名称	简易基金理财销售管理系统		
项目简称	基金管理系统	英文缩写	FSMS
项目编号	001	版本号	无
任务提出者	恒生电子		

开发者	训练营第三小组成员：侯青山、陈天乐、唐巍祚、柏云迪、陈语妍、朱开
用户	各基金公司柜台人员
同其它系统的关系	本系统为独立系统

1.3 定义

- **客户 (Client/Customer)**：指购买理财产品或使用理财服务的个人或机构。
- **账户 (Account)**：客户在理财系统中创建的用于管理其资金和投资的虚拟账户。
- **基金 (Fund)**：由投资公司或金融机构管理的集合投资工具，通过投资多种资产组合以实现预期收益。
- **申购 (Subscription/Purchase)**：客户购买基金或其他理财产品的行为。
- **赎回 (Redemption)**：客户卖出其持有的基金或其他理财产品的行为，以获取资金。
- **净值 (Net Asset Value, NAV)**：基金单位的价格，由基金总资产减去总负债后除以总单位数得出。
- **清算 (Settlement)**：交易完成后，资金和证券在买卖双方之间的交割过程。
- **风险评估 (Risk Assessment)**：对客户投资风险承受能力进行评估，以匹配适合的理财产品。
- **收益 (Return)**：投资所获得的利润或亏损，通常以百分比表示。
- **费用 (Fee/Charge)**：客户在进行理财交易时需支付的各类费用，包括管理费、申购费、赎回费等。
- **报表 (Report)**：系统生成的各类数据汇总和分析报告，用于展示客户账户的财务状况和交易记录。
- **交易 (Transaction)**：客户在理财系统中进行的买卖、转账等操作。
- **利率 (Interest Rate)**：客户存款或贷款的利息百分比。
- **资产配置 (Asset Allocation)**：根据客户风险承受能力和投资目标，将资金分配到不同类型资产的策略。

1.4 法律法规和参考资料

法律法规

1. 《中华人民共和国证券法》：规范证券市场活动，保护投资者合法权益。

2. 《中华人民共和国基金法》：规定基金管理、运作及监督管理的基本制度。
3. 《中华人民共和国反洗钱法》：防范和打击洗钱活动，维护金融秩序。
4. 《个人信息保护法》：保护个人信息安全，规范个人信息处理活动。

需求规格说明书

理财销售交易系统需求规格说明书 (FSMS-SRS)：详细描述系统的功能需求、性能需求、用户界面需求等，为系统设计和开发提供依据。

软件开发标准

1. **IEEE标准**（如**IEEE 830-1998**, **IEEE 12207**）：软件工程的标准，包括需求规格说明书编写标准和软件生命周期过程标准。
2. **ISO/IEC 27001**：信息安全管理标准，规定了建立、实施、维护和持续改进信息安全管理体系的要求。
3. **ISO/IEC 12207**：软件生命周期过程标准，规定了软件开发、运行和维护的标准过程。

2. 总体设计

2.1 需求概述

本理财销售管理系统旨在提供一个全面的理财产品管理平台，涵盖客户开户、资金存入和查询、基金购买、基金赎回、清算、业务查询等功能。系统将支持基金产品的基本信息维护、客户账户管理、交易记录和查询、资金计算和清算等关键业务需求，确保用户能够高效、安全地进行理财操作。

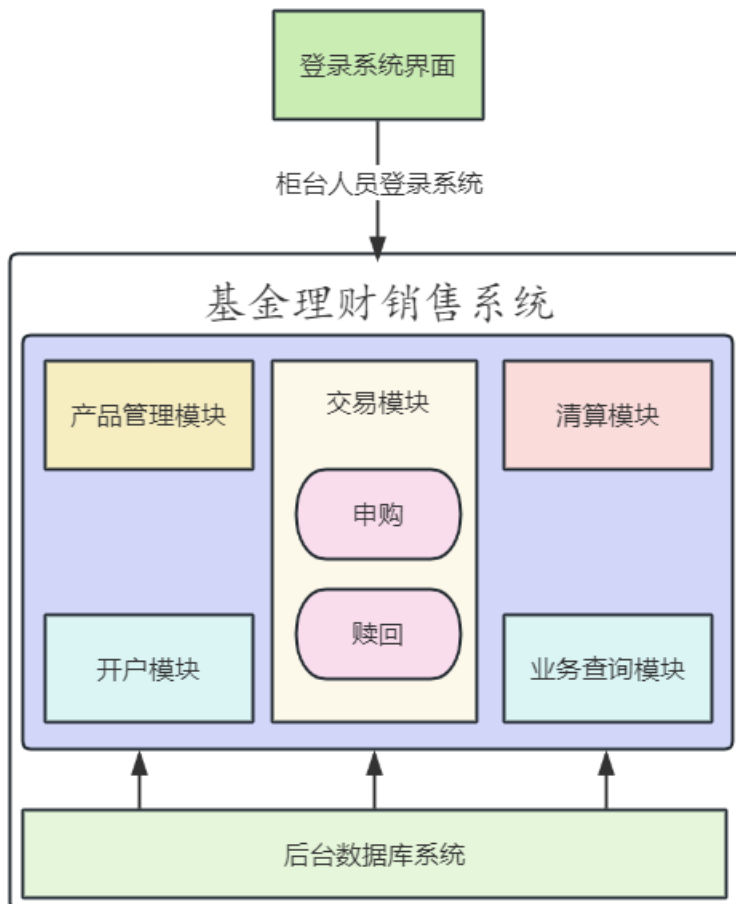
2.2 系统分析

系统模块间关系

- **系统登录模块 (G)**：是系统的入口，柜台人员通过此模块登录后，可以访问其他功能模块。
- **产品管理模块 (A)**：提供产品信息支持给交易模块的申购 (C) 和赎回 (D) 操作，以便用户选择和交易基金产品。

- **开户模块 (B)**：在用户成功开户后，用户的信息会存储到后台数据库系统中，交易模块和业务查询模块可以访问这些用户信息来处理相关请求。
- **交易模块**：
 - **申购模块 (C)** 和 **赎回模块 (D)** 分别处理基金的申购和赎回操作。
 - 申购和赎回的交易数据在完成后会传递给清算模块 (E) 进行资金结算。
- **清算模块 (E)**：接收交易模块传来的数据，进行资金的清算和结算，并将结果存储到后台数据库系统中。
- **业务查询模块 (F)**：可以访问后台数据库系统，提供柜台人员查询用户账户和交易信息的功能。
- **后台数据库系统**：是整个系统的数据存储中心，所有模块都可以从中读取和写入数据，确保数据的集中管理和一致性。

2.3 模块的层次结构图



2.4 模块与需求对应表

说明各项功能需求的实现同各模块的对应关系：

模块编号	模块名/子系统	子模块名	重要模块	功能说明	功
01	产品管理	产品管理	✓	管理和维护各种基金理财产品信息，例如产品名称、类型、收益率等	
02	开户	开户	✓	处理用户开户的相关业务，包括用户信息的录入和验证	
03	申购	申购	✓	处理用户购买基金请求	
04	赎回	赎回	✓	处理用户赎回基金请求	
05	清算	清算	✓	负责交易的结算工作，确保交易资金的正确转移和记录	
06	业务查询	业务查询	✓	提供对各类业务信息的查询功能，包括用户交易记录等	
07	登录注册	登录注册	✓	负责柜台人员的登录注册操作	

2.5 各模块概要说明

模块概要说明

1. 系统登录模块 (G)
 - 负责柜台人员的登录操作。
2. 产品管理模块 (A)

- 管理和维护各种基金理财产品的信息，例如产品名称、类型、收益率等。
3. 开户模块 (B)
- 处理用户开户的相关业务，包括用户信息的录入和验证。
4. 交易模块
- 包含申购模块 (C) 和赎回模块 (D)。
 - 申购模块 (C)：处理用户购买基金的请求。
 - 赎回模块 (D)：处理用户赎回基金的请求。
5. 清算模块 (E)
- 负责交易的结算工作，确保交易资金的正确转移和记录。
6. 业务查询模块 (F)
- 提供对各类业务信息的查询功能，包括用户交易记录等。

2.6 系统角色与功能对应关系

定义系统的主要功能清单，同时定义使用者对功能的权限。

子模块名	功能	角色1	角色2	角色N
产品管理	产品信息呈现	银行柜员	普通用户	
	产品新增/删除	银行柜员		
	产品编辑	银行柜员		
	产品详情查看	银行柜员	普通用户	
	产品净值走势图	银行柜员	普通用户	
开户	客户列表	银行柜员		
	查询客户信息	银行柜员		
	录入客户基本信息	银行柜员		
	录入客户银行卡信息	银行柜员		

	客户风险承受能力	银行柜员		
	客户银行卡余额	银行柜员		
	开户进程显示	银行柜员		
申购	申购申请	银行柜员	普通用户	
	搜索	银行柜员	普通用户	
	风险等级确认	银行柜员	普通用户	
	下单成功界面	银行柜员	普通用户	
赎回	赎回申请	银行柜员	普通用户	
	搜索	银行柜员	普通用户	
	下单成功	银行柜员	普通用户	
清算	日初初始化	银行柜员		
	交易确认功能	银行柜员		
	导出申请数据	银行柜员		
	清算日志	银行柜员		
业务查询	交易查询	银行柜员	普通用户	
	撤单	银行柜员	普通用户	
系统登陆	登录界面	银行柜员		
系统注册	注册界面	银行柜员		

3. 性能设计

对于该项目的性能设计，首先我们进行了性能目标设定：首先需要确定性能指标，根据理财销售管理系统的业务需求和用户期望，明确性能指标，如响应时间、吞吐量和并发用户数等。这些指标将成为性能设计的参考和准入标准；接下来需要量化性能目标，我们设定了具体的性能目标：在高峰期保持平均响应时间在5秒以内，或者能够支持每分钟500笔交易等。这样有助于评估和追踪系统在不同负载下的性能表现。

接下来为具体设计方面的描述：

1. 架构设计：

分布式架构：采用分布式架构将系统功能模块分散到多个服务器上，降低单点故障的风险，并提升系统的可扩展性和可靠性。

微服务架构：将系统功能模块化，采用微服务架构，使得每个服务独立部署和扩展，方便维护和优化。

2. 数据库优化：

读写分离：采用主从数据库架构，将读操作和写操作分离，提高数据库的并发处理能力。

索引优化：针对常用查询建立合适的索引，提高查询效率，减少数据库响应时间。

分区分表：根据业务需求对数据进行分区和分表，降低单表的数据量，提升数据库操作性能。

3. 缓存技术：

内存缓存：使用内存缓存技术，将频繁访问的数据缓存到内存中，减少数据库查询压力，提高响应速度。

页面缓存：对于静态页面或不频繁变化的数据进行页面缓存，减少服务器生成动态页面的次数，提高系统响应速度。

4. 异步处理：

消息队列：采用消息队列，将耗时的操作异步化处理，减少用户等待时间，提高系统的吞吐量。

任务调度：对于定时任务或批处理任务，使用任务调度系统进行异步处理，避免对实时请求的影响。

5. 负载均衡：

硬件负载均衡：使用硬件负载均衡器分配请求流量，确保服务器之间的负载均衡，提升系统的整体处理能力。

软件负载均衡：使用软件负载均衡器，将用户请求均匀分配到不同的应用服务器，提高系统的响应速度和稳定性。

6. 性能监控与优化：

性能监控：建立完善性能监控系统，实时监控系统的各项性能指标（如CPU、内存、响应时间等），及时发现性能瓶颈。

压力测试：定期进行压力测试和负载测试，模拟实际业务场景，评估系统在不同负载下的性能表现，并根据测试结果进行优化。

7. 代码优化：

算法优化：针对核心算法进行优化，减少时间复杂度和空间复杂度，提高代码执行效率。

代码重构：定期进行代码重构，去除冗余代码，提升代码质量和可维护性。

4. 运行设计

4.1 运行模块组合

4.1.1 开户模块组合

运行模块组合： 开户模块 (B) + 后台数据库系统 + 风险评估支持软件

运行历经的内部模块和支持软件：

- **B1:** 客户列表
- **B2:** 查询客户信息
- **B3:** 录入客户基本信息
- **B4:** 录入客户银行卡信息
- **B5:** 客户风险承受能力
- **B6:** 客户银行卡余额
- **B7:** 开户进程显示
- **后台数据库系统：** 用于存储和检索客户数据。
- **风险评估支持软件：** 用于进行客户风险承受能力的评估。

4.1.2 申购模块组合

运行模块组合： 申购模块 (C) + 产品管理模块 (A) + 风险评估支持软件 + 后台数据库系统 + 支付系统

运行历经的内部模块和支持软件：

- **C1:** 申购申请

- C2: 搜索
- C3: 风险等级确认
- C4: 下单成功界面
- A1: 产品信息呈现
- A4: 产品详情查看
- A5: 产品净值走势图
- 风险评估支持软件：用于确认客户的风险承受能力。
- 后台数据库系统：用于存储和检索交易数据。
- 支付系统：用于处理支付操作。

4.1.3 赎回模块组合

运行模块组合：赎回模块 (D) + 产品管理模块 (A) + 后台数据库系统 + 支付系统

运行历经的内部模块和支持软件：

- D1: 赎回申请
- D2: 搜索
- D3: 下单成功
- A1: 产品信息呈现
- A4: 产品详情查看
- A5: 产品净值走势图
- 后台数据库系统：用于存储和检索交易数据。
- 支付系统：用于处理支付操作。

4.1.4 清算模块组合

运行模块组合：清算模块 (E) + 后台数据库系统 + 日常结算软件

运行历经的内部模块和支持软件：

- E1: 日初初始化
- E2: 交易确认功能
- E3: 导出申请数据
- E4: 清算日志
- 后台数据库系统：用于存储和检索交易数据。
- 日常结算软件：用于处理和确认每日交易数据。

4.1.5 业务查询模块组合

运行模块组合： 业务查询模块 (F) + 后台数据库系统

运行历经的内部模块和支持软件：

- F1: 交易查询
- F2: 撤单
- 后台数据库系统： 用于存储和检索交易数据。

4.1.6 产品管理模块组合

运行模块组合： 产品管理模块 (A) + 后台数据库系统

运行历经的内部模块和支持软件：

- A1: 产品信息呈现
- A2: 产品新增/删除
- A3: 产品编辑
- A4: 产品详情查看
- A5: 产品净值走势图
- 后台数据库系统： 用于存储和检索产品数据。

4.2 运行控制

4.2.1 开户模块组合

外界运行控制的方式方法和操作步骤：

1. 银行柜员登录系统。
2. 访问开户模块。
3. 在客户列表中查看现有客户，或查询特定客户信息。
4. 使用录入客户基本信息和银行卡信息表单，添加新客户。
5. 进行客户风险承受能力评估。
6. 保存客户信息并完成开户操作。
7. 显示客户的开户进程状态。

4.2.2 申购模块组合

外界运行控制的方式方法和操作步骤：

1. 用户或银行柜员登录系统。
2. 访问申购模块。
3. 搜索可供申购的基金产品。
4. 选择产品并查看详细信息和净值走势图。
5. 确认产品的风险等级适合客户。
6. 填写申购申请表单并提交。
7. 支付系统处理支付操作。
8. 显示下单成功界面和订单详情。

4.2.3 赎回模块组合

外界运行控制的方式方法和操作步骤：

1. 用户或银行柜员登录系统。
2. 访问赎回模块。
3. 搜索可供赎回的基金产品。
4. 选择产品并查看详细信息和净值走势图。
5. 填写赎回申请表单并提交。
6. 支付系统处理支付操作。
7. 显示下单成功界面和订单详情。

4.2.4 清算模块组合

外界运行控制的方式方法和操作步骤：

1. 银行柜员登录系统。
2. 访问清算模块。
3. 日初系统启动时进行初始化操作。
4. 确认每日交易信息。
5. 导出交易申请数据。
6. 记录并显示清算操作日志。

4.2.5 业务查询模块组合

外界运行控制的方式方法和操作步骤：

1. 用户或银行柜员登录系统。
2. 访问业务查询模块。
3. 输入查询条件并查询交易记录。
4. 查看交易记录列表。
5. 如果需要，撤销某笔交易。

4.2.6 产品管理模块组合

外界运行控制的方式方法和操作步骤：

1. 银行柜员登录系统。
2. 访问产品管理模块。
3. 在产品信息呈现界面查看现有产品。
4. 使用产品新增功能添加新产品，填写产品信息输入表单并保存。
5. 使用产品编辑功能修改现有产品信息，并保存修改。
6. 使用产品删除功能移除不需要的产品。
7. 查看产品的详细信息和净值走势图。

5. 错误处理

5.1 错误发现

为了确保理财销售交易系统的稳定性和可靠性，系统需要具备自动识别和处理各种错误的能力。具体要求如下：

1. **错误识别机制：**系统必须能够准确识别多种错误，包括但不限于系统错误（如服务器故障）、用户输入错误（如无效数据输入）、网络错误（如连接超时）等。
2. **错误分类机制：**系统应当对识别到的错误进行分类，区分不同类型的错误，例如将其分为用户输入错误、网络错误、系统错误等，以便更好地分析和处理。
3. **错误处理方式：**系统应根据错误类型和严重程度，采取相应的处理措施。对于轻微错误，系统可以提示用户纠正；对于严重错误，系统应记录日志并尝试自动修复；在必要时，还需通知系统管理员。
4. **错误信息反馈：**系统应向用户提供详细且易于理解的错误信息，帮助用户识别错误原因，并提供相应的解决方案或建议，确保用户能够自行解决常见问题。

5. **错误日志记录：**系统需要将所有错误详细记录到错误日志中，包括错误类型、发生时间、相关用户信息等，以便系统管理员进行后续分析和排查。

5.2 错误解决

为了更好地管理和处理系统运行过程中可能出现的错误，需制定详细的错误处理流程：

1. **监测错误：**系统在运行过程中，应实时监测并检测各种潜在错误情况，包括用户操作、系统性能、网络状况等。
2. **识别错误：**当系统检测到错误时，首先需要准确识别错误类型，并根据其严重程度判断处理优先级。
3. **处理用户输入错误：**对于一般的用户输入错误或网络错误，系统应及时向用户提供友好的提示信息，指导用户如何修正错误。例如，当用户输入无效数据时，系统可以显示具体的错误提示并建议用户重新输入。
4. **记录严重错误：**对于严重的系统错误或运行时异常，系统应及时记录错误日志。日志应包含错误类型、发生时间、用户信息等详细信息，以便管理员进行分析和排查。
5. **自动修复和恢复：**系统在记录错误日志的同时，应尝试自动修复错误或恢复系统状态，确保系统能继续正常运行。例如，对于常见的网络中断问题，系统可以自动重试连接或切换备用线路。
6. **通知管理员：**对于无法自动修复的严重错误或需要管理员介入的情况，系统应及时向系统管理员发送警报或通知。通知内容应包括错误详细信息及建议的处理步骤，以便管理员能够迅速响应和解决问题。
7. **错误分析与优化：**系统管理员应定期分析错误日志，找出常见错误的根本原因，并进行系统优化和改进，预防类似错误再次发生。

6. 接口设计

6.1 用户界面

6.1.1 产品管理模块

U.D.U. 理财销售柜台系统

投资者问卷调查 风险承受能力查询 资金存入 印鉴录入 银行资料维护

请输入关键词

首页 个人开户

个人开户/登记

1 个人开户信息 2 反洗钱信息 3 CRS信息 4 风险测评信息

业务信息

委托方式 4/传真委托 X

其他委托方式 4/传真委托 X

TAR代码 请选择 (多选)

基金账号

客户信息

客户姓名

证件类型 请选择

证件号码

证件有效期 选择日期 固 长期有效

重复证件号码

客户性别 请选择

性别 请选择

下一步

业务信息
客户信息
适当性信息
银行账户信息
联系方式
经纪人
其他信息
经办人
附件信息

展开附件

6.1.3 申购模块

U.D.U. 理财销售柜台系统

投资者问卷调查 风险承受能力查询 资金存入 印鉴录入 银行资料维护

请输入关键词

首页 个人开户 产品购买

产品购买

委托方式 4/传真委托

账号输入 000096

基金账号 *110000000096 交易账号 000096

客户信息

客户名称:

客户类型:

证件类型:

证件号码:

反洗钱备注:

业务管理: 查看记录

业务信息

产品代码 请选择

购买金额 请输入

可用余额 0.00

货币币种 156/人民币

基金方式 1/普通方式 X

份额限制 请选择

6.1.4 赎回模块



6.1.5 清算模块



6.1.6 业务查询模块



6.1.7 登录界面



6.2 外部接口

6.2.1 第三方支付网关接口

- 目的：实现用户购买基金时的资金划转，以及赎回基金后的资金回笼。
- 接口描述：

- 接口名称： FundPurchasePayment
- 请求方式： POST
- 请求参数：

- `userId` (String): 用户唯一标识。
- `fundCode` (String): 基金代码。
- `amount` (Decimal): 交易金额。
- `paymentMethod` (Enum): 支付方式（如信用卡、借记卡、第三方支付等）。

○ 响应内容：

- `transactionId` (String): 交易唯一标识。
- `status` (Enum): 交易状态（成功、失败、处理中）。
- `message` (String, optional): 错误信息（交易失败时返回）。

6.2.2 身份验证服务接口

- 目的：确保用户身份的真实性，提供安全登录和账户管理功能。

● 接口描述：

- 接口名称： `AuthenticateUser`

- 请求方式：POST

○ 请求参数：

- `username` (String): 用户名或邮箱。
- `password` (String): 密码。
- 或
- `accessToken` (String): 通过OAuth2获取的访问令牌。

○ 响应内容：

- `userId` (String): 验证成功的用户唯一标识。
- `token` (String, optional): 新生成的访问令牌（若使用密码登录成功）。
- `refreshToken` (String, optional): 刷新令牌（若使用密码登录成功）。
- `expiresIn` (Int, optional): 令牌有效期（秒）。
- `status` (Enum): 验证状态（成功、失败）。
- `message` (String, optional): 验证失败原因。

6.2.3 市场数据接口

- 目的：获取最新的基金市场数据，包括但不限于基金净值、指数行情等。

● 接口描述：

- 接口名称： `FetchMarketData`

- 请求方式: GET
- 请求参数:
 - fundCodes (List): 需要查询的基金代码列表。
 - asOfDate (DateTime, optional): 查询的日期, 默认为当前日期。
- 响应内容:
 - marketData (List): 每个对象包含基金代码、净值、日增长率等关键数据。

6.2.4 风险评估服务接口

- 目的: 根据用户的投资偏好、财务状况等信息, 评估其风险承受能力。
- 接口描述:
 - 接口名称: EvaluateRiskProfile
 - 请求方式: POST
 - 请求参数:
 - userId (String): 用户唯一标识。
 - investmentExperience (Enum): 投资经验等级。
 - incomeLevel (Enum): 收入水平。
 - assetAllocation (Map<String, Decimal>): 资产配置比例 (股票、债券、现金等)。
 - 响应内容:
 - riskLevel (Enum): 用户的风险承受等级 (低、中、高)。
 - suggestedInvestmentStrategy (String): 建议的投资策略概述。

6.2.5 法规遵从性报告接口

- 目的: 定期或按需向监管机构提交交易记录、客户信息等合规报告。
- 接口描述:
 - 接口名称: SubmitComplianceReport
 - 请求方式: POST
 - 请求参数:
 - reportType (Enum): 报告类型 (交易报告、客户尽职调查报告等)。
 - data (Object): 报告具体内容, 格式根据报告类型而定。
 - 响应内容:

- `submissionId` (String): 报告提交记录的唯一标识。
- `status` (Enum): 提交状态（成功、失败）。
- `message` (String, optional): 提交失败的原因。

6.3 内部接口

6.3.1 产品管理模块接口

6.3.1.1 获取所有产品信息

接口信息

接口地址	/fund/fundinfo
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	系统遍历所有产品

请求示例

复制代码

```
{
  "Fund_id": "",
  "Fund_name": "",
  "Fund_type": "",
  "Fund_nav": "",
  "Fund_est_date": "",
  "Fund_code": ""
}
```

请求参数：

参数名	参数说明	类型	备注
-----	------	----	----

Fund_id	产品id	Int	非空
Fund_name	产品名称	String	非空
Fund_type	产品类型	String	非空
Fund_nav	产品净值	Decimal(10,2)	非空
Fund_est_date	上架时间	datetime	非空
Fund_code	产品代码	String	非空

6.3.1.2 获取某个产品详细信息

接口地址	/fund/fullinfo
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	根据id获取产品详细信息

请求示例

复制代码

```
{
  "Fund_id": "",
  "Fund_name": "",
  "Fund_type": "",
  "Fund_nav": "",
  "Fund_est_date": "",
  "Fund_code": ""
}
```

请求参数

参数名	参数说明	类型	备注
Fund_id	产品id	Int	非空
Fund_name	产品名称	String	非空
Fund_type	产品类型	String	非空
Fund_nav	产品净值	Decimal(10,2)	非空
Fund_est_date	上架时间	datetime	非空
Fund_code	产品代码	String	非空

6.3.1.3 删除产品

接口地址	/fund/modify_fund
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	根据id删除产品

请求示例

⌵

复制代码

```
{
  "Fund_id": "",
  "Fund_name": "",
  "Fund_type": "",
  "Fund_nav": "",
  "Fund_est_date": "",
  "Fund_code": ""
}
```

```
}
```

请求参数：

参数名	参数说明	类型	备注
Fund_id	产品id	Int	非空
Fund_name	产品名称	String	非空
Fund_type	产品类型	String	非空
Fund_nav	产品净值	Decimal(10,2)	非空
Fund_est_date	上架时间	datetime	非空
Fund_code	产品代码	String	非空

6.3.1.4 修改产品

接口地址	/fund/modify_fund
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	根据id修改产品详细信息

请求示例

∨

复制代码

```
{
  "Fund_id": "",
  "Fund_name": "",
  "Fund_type": "",
```

```
"Fund_nav": "",
"Fund_est_date": "",
"Fund_code": ""
}
```

请求参数：

参数名	参数说明	类型	备注
Fund_id	产品id	Int	非空
Fund_name	产品名称	String	非空
Fund_type	产品类型	String	非空
Fund_nav	产品净值	Decimal(10,2)	非空
Fund_est_date	上架时间	datetime	非空
Fund_code	产品代码	String	非空

6.3.1.5 增加产品

接口地址	/fund/modify_fund
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	新增产品信息

请求示例

∨

复制代码

```
{
```



```
"Fund_id":"","  
"Fund_name":"","  
"Fund_type":"","  
"Fund_nav":"","  
"Fund_est_date":"","  
"Fund_code":"","  
}
```

请求参数：

参数名	参数说明	类型	备注
Fund_id	产品id	Int	非空
Fund_name	产品名称	String	非空
Fund_type	产品类型	String	非空
Fund_nav	产品净值	Decimal(10,2)	非空
Fund_est_date	上架时间	datetime	非空
Fund_code	产品代码	String	非空

6.3.2 开户模块接口

6.3.2.1 查看客户列表

接口地址	/customer/customerinfo
请求方式	Get
请求数据类型	json
响应数据类型	json

接口描述	获取所有的客户信息
------	-----------

请求示例

∨

复制代码

```
{
  "customer_id": "",
  "customer_name": "",
  "customer_idcard_type": ""
}
```

请求参数:

参数名	参数说明	类型	备注
customer_id	客户id	Int	非空
customer_name	客户姓名	String	非空
customer_idcard_type	客户类型	int	非空

6.3.2.2 查看客户详细信息

接口地址	/customer/customerfullinfo
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	获取某个客户信息

请求示例

∨

复制代码

```
{
  "customer_id": "",
  "customer_name": "",
  "customer_idcard_type": "",
  "customer_idcard": "",
  "customer_phone": ""
}
```

请求参数：

参数名	参数说明	类型	备注
customer_id	客户id	Int	非空
customer_name	客户姓名	String	非空
customer_idcard_type	客户类型	int	非空
customer_idcard	客户认证号码	int	非空
customer_phone	客户联系反射方式	String	非空

6.3.2.3 搜索客户

接口地址	/customer/search_customer
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	根据客户认证号码查询客户

请求示例

∨

复制代码

```
{
  "customer_id":"","
  "customer_name":"","
  "customer_idcard_type":"","
  "customer_idcard":"","
  "customer_phone":""
}
```

请求参数：

参数名	参数说明	类型	备注
customer_id	客户id	Int	非空
customer_name	客户姓名	String	非空
customer_idcard_type	客户类型	int	非空
customer_idcard	客户认证号码	int	非空
customer_phone	客户联系反射方式	String	非空

6.3.2.4 开户操作

接口地址	/customer/create_customer
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	为客户创建新账户

请求示例

∨

复制代码

```
{
  "customer_id": "",
  "customer_name": "",
  "customer_idcard_type": "",
  "customer_idcard": "",
  "customer_phone": ""
}
```

请求参数:

参数名	参数说明	类型	备注
customer_name	客户姓名	String	非空
customer_idcard_type	客户类型	int	非空
customer_idcard	客户认证号码	int	非空
customer_phone	客户联系反射方式	String	非空

6.3.2.5 客户信息管理

接口地址	/customer/customer_management
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	进行账户的管理

请求示例

▼

复制代码

 复制代码

```
{
  "customer_name": "",
  "customer_idcard_type": "",
  "customer_idcard": "",

```

```
"customer_phone":""
}
```

请求参数：

参数名	参数说明	类型	备注
customer_name	客户姓名	String	非空
customer_idcard_type	客户类型	int	非空
customer_idcard	客户认证号码	int	非空
customer_phone	客户联系反射方式	String	非空

6.3.3 申购模块接口

6.3.3.1 查询用户信息

接口地址	/user/query
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	查询所有的用户信息

请求示例

⌵复制代码

```
{
  "account_id":""
}
```

请求参数：

参数名	参数说明	类型	备注
account_id	账户id	int	非空

6.3.3.2 查询基金信息

接口地址	/fund/query
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	查询所有的基金信息

请求示例

▼

复制代码

```
{
  "fund_id":""
}
```

请求参数：

参数名	参数说明	类型	备注
fund_id	基金id	int	非空

6.3.3.3 获取客户所有银行卡号

接口地址	/card/query
------	-------------

请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	查询所有的银行卡

请求示例

▼

复制代码

```
{
  "account_id":""
}
```

参数名	参数说明	类型	备注
account_id	银行卡号	int	非空

6.3.3.4 获取客户指定银行卡金额

接口地址	/card/balance
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	查询指定银行卡金额

请求示例

▼

复制代码

```
{
  "account_id": "",
  "card_number": ""
}
```

参数名	参数说明	类型	备注
account_id	银行卡号	int	非空
card_number	银行卡号	int	非空

6.3.3.5 发起申购申请

接口地址	/purchase/create
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	发起申购申请

请求示例

▼

复制代码

```
{
  "account_id": "",
  "card_number": "",
  "fund_id": "",
  "purchase_amount": ""
}
```

参数名	参数说明	类型	备注
-----	------	----	----

account_id	客户账户	int	非空
card_number	银行卡号	int	非空
fund_id	产品id	int	非空
purchase_amount	申购金额	decimal(10,2)	非空

6.3.4 赎回模块接口

6.3.4.1 查询用户信息

接口地址	/user/query
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	查询所有的用户信息

请求示例

∨

复制代码

```
{
  "account_id":""
}
```

请求参数:

参数名	参数说明	类型	备注
account_id	账户id	int	非空

6.3.4.2 查询基金信息

接口地址	/fund/query
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	查询所有的基金信息

请求示例

∨

复制代码

```
{
  "fund_id":""
}
```

请求参数：

参数名	参数说明	类型	备注
fund_id	基金id	int	非空

6.3.4.3 获取客户在该基金的剩余份额

接口地址	/hold/query
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	获取客户在该基金的剩余份额

请求示例

▼复制代码

```
{
  "account_id": "",
  "fund_id": ""
}
```

请求参数：

参数名	参数说明	类型	备注
account_id	客户账户	int	非空
fund_id	基金id	int	非空

6.3.4.4 发起赎回申请

接口地址	/hold/query
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	发起赎回申请

请求示例

▼复制代码

```
{
  "account_id": "",
  "fund_id": "",
  "redemption": ""
}
```

请求参数：

参数名	参数说明	类型	备注
accound_id	客户账户	int	非空
fund_id	基金id	int	非空
redemption	赎回份额	decimal(10,2)	非空

6.3.5 清算模块接口

6.3.5.1 生成产品净值

接口地址	/fund/create_dailynav
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	生成产品净值

6.3.5.2 获取产品净值

接口地址	/fund/get_dailynav
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	获取产品净值

6.3.5.3 处理申购

接口地址	/settlement/settle_purchase
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	处理申购

6.3.5.4 更新持仓信息

接口地址	/hold/update
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	更新持仓信息

6.3.5.5 更新份额记录

接口地址	/share/update
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	更新份额记录

6.3.5.6 处理赎回

接口地址	/settlement/settle_redemption
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	处理赎回

6.3.5.7 更新银行卡记录

接口地址	/cardrecord/update
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	更新银行卡记录

6.3.5.8 更新银行卡余额

接口地址	/card/update
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	更新银行余额

6.3.6 业务查询模块接口

6.3.6.1 交易查询

接口地址	/query/transaction_query
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	交易查询

6.3.6.2 撤单

接口地址	/transaction/delete_purchase
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	撤单

6.3.7 登录/注册模块接口

6.3.7.1 用户登录

接口地址	/user/login
请求方式	Get
请求数据类型	json
响应数据类型	json
接口描述	进行用户登录

请求示例

▼复制代码

```
{
  "user_id": "",
  "user_password": ""
}
```

请求参数：

参数名	参数说明	类型	备注
user_id	用户身份证号码	String	需要用户提供真实的
user_password	用户密码	String	

6.3.7.2 用户注册

接口地址	/user/register
请求方式	Post
请求数据类型	json
响应数据类型	json
接口描述	用户注册

请求示例

▼复制代码

```
{
  "user_id": "",
  "user_name": "",
  "user_gender": "",
  "user_nickname": "",
  "user_password": ""
}
```

请求参数：

参数名	参数说明	类型	备注
user_id	用户身份证号码	String	需要用户提供真实的
user_name	用户姓名	String	非空
user_gender	用户性别	String	非空
user_nickname	用户昵称	String	
user_password	用户密码	String	

7. 约束条件

1. 运行环境约束：

硬件要求：系统需在至少拥有4GB内存和双核处理器的服务器上运行，以保证基本的性能需求。若有更高的性能要求，服务器配置应适当提升。

操作系统兼容性：系统支持的操作系统包括Windows Server 2019及以上版本和Linux发行版（如Ubuntu 20.04及以上）。其他操作系统版本可能不被支持或需要额外的适配。

2. 网络带宽约束：

数据传输：为了保证实时交易数据的流畅传输，建议最低网络带宽为100 Mbps。对于高频交易系统，建议使用更高带宽的网络连接（例如1 Gbps或以上）。

3. 数据安全性约束：

加密：系统必须使用TLS 1.2或以上版本加密所有的数据传输，以保护用户数据和交易信息不被中途截获。

访问控制：系统需要实现多级用户权限管理，包括管理员、普通用户和只读用户等角色，并根据不同角色限制访问权限。

4. 数据存储约束：

数据库支持：系统应支持常用的关系型数据库，如MySQL 8.0或以上、PostgreSQL 13.0或以上。对于大规模数据存储，还需支持分布式数据库或数据仓库。

备份策略：系统需要实现定期自动备份机制（每日备份），并支持手动备份和恢复操作。备份数据需存储在异地备份服务器上。

5. 性能要求约束：

响应时间：系统的交易处理响应时间应不超过2秒，以确保用户能够在合理时间内完成交易操作。

并发用户：系统需支持至少500名并发用户同时进行交易操作，且在高峰期性能不显著下降。

6. 系统维护约束：

升级与补丁：系统需要具备自动化的升级和补丁管理机制，以便及时应用安全补丁和系统更新，保证系统的稳定性和安全性。

日志记录：系统必须记录所有关键操作和事件的日志，并支持对日志进行审计和分析。日志数据需至少保留6个月。

7. 用户界面约束：

兼容性：系统前端应支持主流浏览器（如Chrome、Firefox、Edge等）的最新两个主要版本，以及常见的移动设备浏览器。

无障碍支持：系统前端需符合WCAG 2.1 AA级别的无障碍标准，以确保所有用户，包括有视觉障碍的用户，都能顺利使用系统。

8. 法律与合规约束：

数据隐私：系统必须遵守相关的数据隐私法规，如《个人信息保护法》或《通用数据保护条例（GDPR）》等，确保用户数据的合法收集、存储和处理。

金融合规：系统应符合金融监管机构的相关规定，包括反洗钱法规（AML）和了解客户法规（KYC），以确保交易的合法性和合规性。

8. 系统数据结构设计

8.1 设计要点

名	自动递...	修改日期	数据长度	引擎
user	0		16 KB	InnoDB
transaction	0		16 KB	InnoDB
sharerecord	0		16 KB	InnoDB
settlement	0		16 KB	InnoDB
hold	0		16 KB	InnoDB
fund	0		16 KB	InnoDB
dailyfundnav	0		16 KB	InnoDB
customer	0		16 KB	InnoDB
creditcardrecord	0		16 KB	InnoDB
creditcard	0		16 KB	InnoDB
account	0		16 KB	InnoDB

1. **存储要求：** 每个数据项需满足特定的数据类型和长度要求，以保证数据的准确性和存储效率。

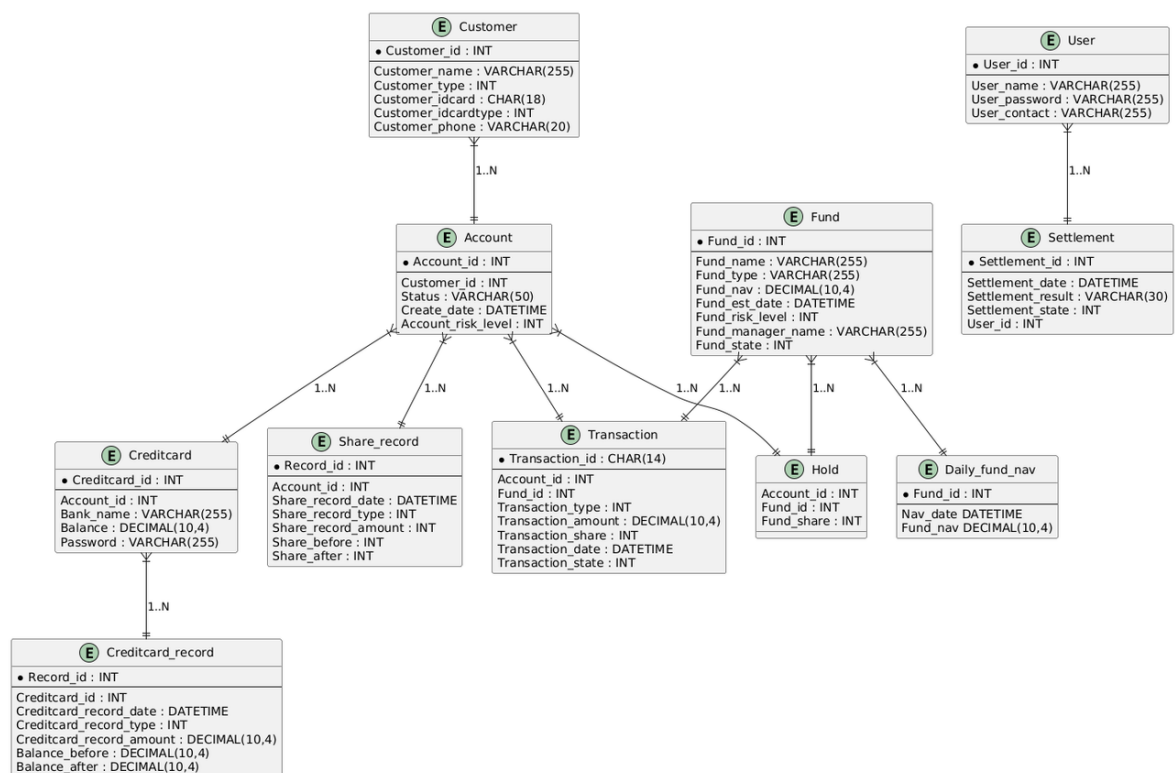
例如，账户余额和交易金额使用 `decimal(10,4)` 类型，确保财务数据的精确表示。

2. **物理关系：**

- **索引：** 为提高查询效率，对经常查询和排序的字段（如账户ID、客户ID、基金ID等）建立索引。

3. **保密处理：** 对于敏感的信息如账户密码、基金信息等，采用加密处理，并限制访问权限，确保数据安全。

8.2 详细设计



- 账户信息表：

名	类型	长度	小数点	不是 null	虚拟	键
account_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
customer_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
creditcard_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
account_balance	decimal	10	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
account_est_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
fund_share	decimal	10	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	


- 账户ID（主键，自增）
- 用户ID（int）
- 银行卡ID（int）
- 账户余额（decimal(10, 2)）
- 创建时间（datetime）
- 账户持有基金份额（decimal(10,2)）

- 客户信息表：

名	类型	长度	小数点	不是 null	虚拟	键
customer_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
customer_name	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
contact_info	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	
risk_level	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

- 客户ID（主键，自增）
- 客户姓名（varchar）
- 联系方式（varchar）
- 风险等级（int）

- 基金信息表：

名	类型	长度	小数点	不是 null	虚拟	键
fund_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
fund_name	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
fund_type	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
fund_nav	decimal	10	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
fund_est_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
fund_code	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

- 基金ID（主键）
- 基金名称（varchar）
- 基金类型（varchar）
- 基金净值（decimal(10,2)）

- 创建时间 (datetime)
- 基金代码 (varchar)

● 风险评估表：

名	类型	长度	小数点	不是 null	虚拟	键
riskassessment_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
customer_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
riskassessment_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
risk_level	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
riskassessment_result	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	


- 风险评估ID (主键)
- 客户ID (int)
- 风险评估日期 (datetime)
- 风险等级 (int)
- 风险评估结果 (varchar)

● 清算信息表：

名	类型	长度	小数点	不是 null	虚拟	键
settlement_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
settlement_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
settlement_result	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
settlement_status	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

- 清算ID (主键)
- 清算日期 (datetime)
- 清算结果 (varchar)
- 清算状态 (int)

● 交易信息表：

名	类型	长度	小数点	不是 null	虚拟	键
transaction_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
account_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
fund_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
transaction_type	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
transaction_amount	decimal	10	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
transaction_share	decimal	10	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
transaction_date	datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

- 交易ID (主键, 自增)
- 账户ID (int)
- 基金ID (int)
- 交易类型 (int)
- 交易金额 (decimal(10,2))

- 交易份额 (decimal(10,2))
- 交易时间 (datetime)
- 银行柜员信息表：

名	类型	长度	小数点	不是 null	虚拟	键
user_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
user_name	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
password	varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

- 银行柜员ID (主键, 自增)
- 银行柜员姓名 (varchar)
- 银行柜员密码 (varchar)

8.3 数据结构和模块的关系

说明各个数据结构访问这些数据结构的各个模块之间的对应关系，可采用如下图表形式：

	产品管理 模块	开户模块	申购模块	赎回模块	清算模块	业务查询 模块	登录模
账户信息表		a	a	a		a	
客户信息表		a	a	a		a	a
基金信息表			a	a			
风险评估表			a				
清算信息表					a		
交易信息表			a	a	a	a	
银行柜员信 息表	a	a	a	a	a	a	a

9. 功能模块设计

9.1 模块A：产品管理

功能概述：产品管理模块的主要功能是提供产品信息的维护和展示。包括产品的新增、删除、编辑，查看产品详情及其净值走势图等。该模块确保产品信息的准确性和及时性，支持基金产品的全面管理，便于柜台人员和管理人员操作和查询。

9.1.1 A1: 产品信息呈现

功能说明：系统所有角色的操作员采用一致的登录界面，登录提交后，产品管理模块需要提供产品列表展示功能，用于显示系统中的产品列表。

界面要素：产品列表表格，包括产品名称、风险等级、净值等信息。

9.1.2 A2: 产品新增/删除

功能说明：允许用户添加新产品或删除现有产品。

界面要素：新增按钮、删除按钮、产品信息输入表单（包括产品名称、风险等级、净值等）。

9.1.3 A3: 产品编辑

功能说明：用户可以编辑现有产品的信息。

界面要素：产品信息编辑表单（包括产品名称、风险等级、净值等）、保存按钮。

9.1.4 A4: 产品详情查看

功能说明：查看单个产品的详细信息。

界面要素：产品详情页面，显示产品的所有信息，包括历史数据和图表。

9.1.5 A5: 产品净值走势图

功能说明：显示产品净值的变化趋势。

界面要素：净值趋势图表，包括时间轴、净值数据点等。

9.2 模块B：开户

功能概述：开户管理模块的主要功能是为投资者提供开户服务，并管理和维护客户的基本信息、银行卡信息和风险承受能力。该模块确保客户信息的准确性和完整性，支持柜台人员为投资者快速办理开户业务。

9.2.1 B1: 客户列表

功能说明：显示所有客户的列表。

界面要素：客户列表表格，包括客户姓名、开户日期、状态等信息。

9.2.2 B2: 查询客户信息

功能说明：根据客户姓名或其他信息查询特定客户。

界面要素：查询输入框、查询按钮、查询结果表格。

9.2.3 B3: 录入客户基本信息

功能说明：录入新客户的基本信息。

界面要素：客户信息输入表单，包括姓名、身份证号、联系方式等。

9.2.4 B4: 录入客户银行卡信息

功能说明：录入客户的银行卡信息。

界面要素：银行卡信息输入表单，包括银行卡号、开户行等。

9.2.5 B5: 客户风险承受能力

功能说明：评估并记录客户的风险承受能力。

界面要素：风险评估问卷、评估结果显示。

9.2.6 B6: 客户银行卡余额

功能说明：显示客户银行卡的余额信息。

界面要素：余额查询功能、余额显示区域。

9.2.7 B7: 开户进程显示

功能说明：显示客户开户的进程状态。

界面要素：进程条或进度列表，显示开户各步骤的状态。

9.3 模块C：申购

功能概述：申购模块负责处理投资者的基金购买请求，从选择基金、确认投资细节到完成支付的全过程。它通过优化流程、提升操作便利性和保证交易安全，使得投资者能快速、高效地进行基金投资，同时确保所有操作符合相关法律法规要求，增强用户体验与服务效率。

9.3.1 C1: 申购申请

功能说明：用户可以提交产品的申购申请。

界面要素：申购申请表单，包括产品选择、申购金额输入等。

9.3.2 C2: 搜索

功能说明：搜索可供申购的产品。

界面要素：搜索输入框、搜索按钮、搜索结果列表。

9.3.3 C3: 风险等级确认

功能说明：在申购前确认产品的风险等级是否适合客户。

界面要素：风险等级显示、确认按钮。

9.3.4 C4: 下单成功界面

功能说明：申购申请成功后的确认界面。

界面要素：成功提示信息、订单详情显示。

9.4 模块D：赎回

功能概述：赎回模块主要负责帮助投资者将持有的基金转换为现金，实现投资收益的提现，同时提供资金的灵活性和投资组合的调整能力。主要包含的功能有：订单处理与验证、份额计算与定价。这一过程确保便捷、安全，并符合相关法规要求。

9.4.1 D1: 赎回申请

功能说明：用户可以提交产品的赎回申请。

界面要素：赎回申请表单，包括产品选择、赎回金额输入等。

9.4.2 D2: 搜索

功能说明：搜索可供赎回的产品。

界面要素：搜索输入框、搜索按钮、搜索结果列表。

9.4.3 D3: 下单成功

功能说明：赎回申请成功后的确认界面。

界面要素：成功提示信息、订单详情显示。

9.5 模块E：清算

功能概述：清算模块主要负责基金公司每日交易数据的确认、结算和记录。该模块包括日初始化、交易确认、数据导出和清算日志记录等关键功能，确保每日交易的准确处理和系统状态的实时更新。

9.5.1 E1: 日初初始化

功能说明：每天系统启动时进行初始化操作。

界面要素：初始化状态显示、进度条。

9.5.2 E2: 交易确认功能

功能说明：确认每日交易信息。

界面要素：交易列表、确认按钮、状态显示。

9.5.3 E3: 导出申请数据

功能说明：导出交易申请数据。

界面要素：导出按钮、文件格式选择、下载链接。

9.5.4 E4: 清算日志

功能说明：记录并显示清算操作日志。

界面要素：日志列表，包括操作时间、操作人、操作内容等。

9.6 模块F：业务查询

功能概述：业务查询模块主要提供交易申购、赎回的申请查询和未确认交易的撤单功能。该模块帮助柜台人员和投资人查询和管理其交易申请，确保交易数据的透明和撤单操作的有效执行。

9.6.1 F1: 交易查询

功能说明：查询交易记录。

界面要素：查询输入框、查询按钮、交易记录列表。

9.6.2 F2: 撤单

功能说明：允许用户撤销某笔交易。

界面要素：交易记录列表、撤单按钮、确认提示框。

9.7 模块G：登录注册

功能概述：登录管理模块主要提供系统的登录和身份验证功能，确保柜台人员和其他用户能够安全、快捷地访问系统功能。

9.7.1 G1: 登录界面

功能说明：系统的登录界面，提供用户登录功能。

界面要素：用户名输入框、密码输入框、登录按钮、忘记密码链接。

9.7.2 G2: 注册界面

功能说明：系统的注册界面，提供用户注册功能。

界面要素：用户名输入框、密码输入框、确认密码输入框、注册按钮。