RMP 平台作业完整文档

1.产品需求分析

随着市场情况的不断变化，银行时常需要推出新的存款产品，并提供平台支持外部基金的代销业务。然而，传统开发模式需要大量定制化工作和较长的开发周期，常常导致产品上市时机错失，也不利于基金销售。因此，开发一款面向银行、基金公司和用户的三方综合性一站式平台有着极其重要的战略意义。平台应该能够快速响应市场变化，支持多样化产品，并提供用户友好界面，保障安全可靠性，从而促进基金销售和业务发展。针对市场需求，本小组成员进行了本产品的开发，本平台主要面向公众个人用户（平台用户）、银行平台的配置管理用户（平台管理者）和理财金公司用户（服务提供者）三类用户，主要功能需求如下：

1. 公众个人用户

- 产品搜索
- 分类查看
- 按关键词搜索存款产品
- 查看和购买产品

（上述功能包含手机端和网页端两种入口）

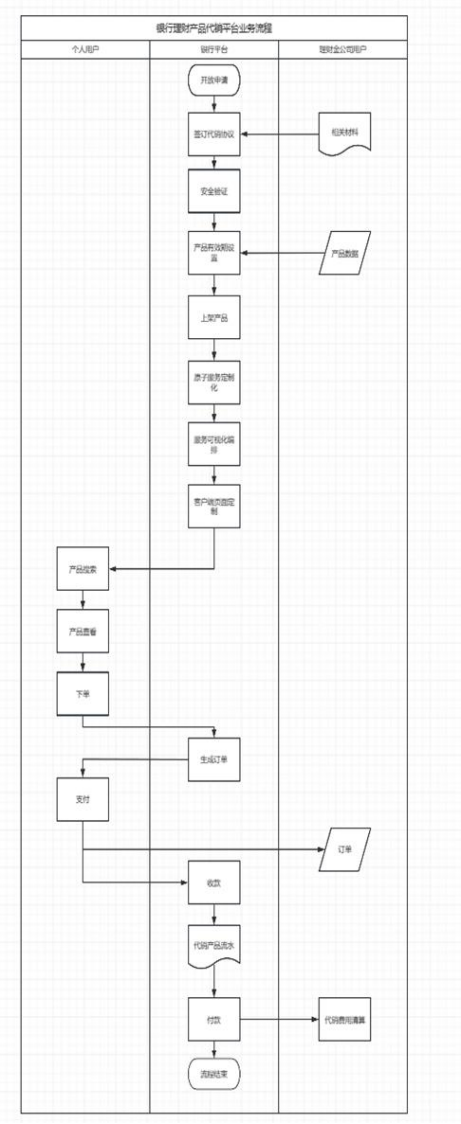
2. 银行平台的配置管理用户

- 产品管理（CURD）
- 原子服务定制化（对用户管理，交易系统等模块进行微服务管理）
- 服务可视化编排（以图形操作对后端返回的数据进行裁剪、聚合等操作）
- 客户端页面定制化（决定各个产品展示页面的展示方式）
- 对相关理财金公司的各种验证和产品有效期设置

3. 理财金公司用户

- 上传并定义各种理财产品
- 和银行签订代销协议
- 支持与用户的交易和赎回
- 和银行平台的代销费用清算

根据以上功能需求分析结果，我们可以画出如下的业务流程泳道图：



本产品流程中主要涉及银行平台（管理用户）、理财金公司用户和公众个人用户三类参与者。

主要包含的用例有：

1.银行方的客户端定制功能；

2.银行方的服务可视化功能；

3.银行方的产品管理功能；

4.银行方的设置公司验证与产品有效期功能；

5.在银行方与理财金公司用户方间的签订代销协议功能；

6.在银行方与理财金公司用户方间的代销费用结算功能；

7.理财金公司用户方的上传理财产品功能；

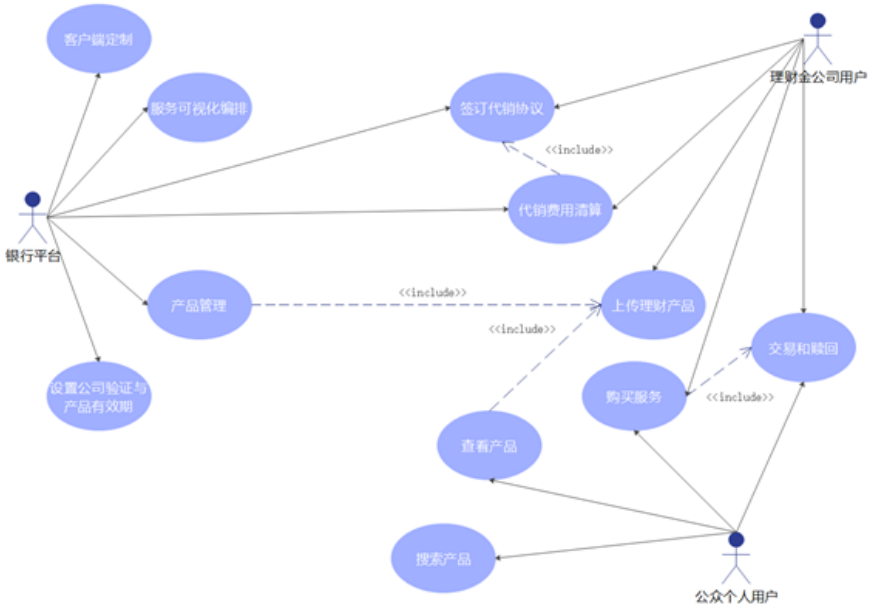
8.在理财金公司用户方与公众个人用户方间的交易和赎回功能；

9.在理财金公司用户方与公众个人用户方间的购买服务功能

10.公众个人用户方的搜索产品功能；

11.公众个人用户方的查看产品功能；

结合以上分析，可绘制如下用例图：



2. 产品架构设计

1. 核心算法

在推荐基金产品的时候，我们准备使用基于物品的协同过滤对基金产品进行排序，我们目前计划里，存储基金产品的格式如下：

```
{
  "name": "Stable Growth Fund",
  "description": "This fund aims to provide stable growth opportunities through a diversified investment portfolio for long-term returns.",
  "type": "1", // 1 表示基金，2 表示存款?
```

```
  "risk": "1", // 1 低风险，2 中风险，3 高风险
  "rate": "8", // 8%
  "minInvest": "500", // 购买最低额度
  "term": "12", // 以 month 计
  "manageFee": "1.5", // 平台收取 1.5%
  "status": "1", // 1 开放购买，2 停止购买，3 清算完毕
  "createDate": "2023-01-15"
}
```

我们首先根据基金的相关 attribute，例如 type, risk, rate, minInvest, term 来构建基金产品之间的相似度矩阵，然后根据用户的交易记录来判断出用户对某个基金产品的偏好，并推荐与该基金产品相似的其他基金产品。例如，对于一个低风险偏好的用户，可以推荐与已选基金风险水平相似、预期收益率较高、投资期限和最低额度适合的其他基金产品。通过这种方式，我们可以为用户提供与其投资偏好相匹配的基金产品推荐，以帮助其做出更好的投资决策。

输入与输出

输入：

1.“userTxData”：包含用户历史交易信息的数组，其中主要包含历史产品及其相关 属性（“risk”、“rate”、“minInvest”）。

2. “productData”：包含产品仓库中的产品及其相关属性。

输出：更新后的“productData”，根据计算出的相似度从高到低排列。

算法步骤

1. 提取用户购买的产品列表“userPurchase”；

2. 计算所有产品属性的最大值和最小值；

3. 定义归一化函数，将产品属性归一化到[0，1]范围；

4. 对产品仓库中的每个产品，计算其与每个用户历史购买产品的欧氏距离（相似度）并相加，以此作为该产品与用户购买习惯的符合程度度量；

5. 根据相似度总和对产品仓库中的产品进行排序；

6. 更新“productData”，呈现更新排序后的产品列表。

复杂度分析

1. 时间复杂度：

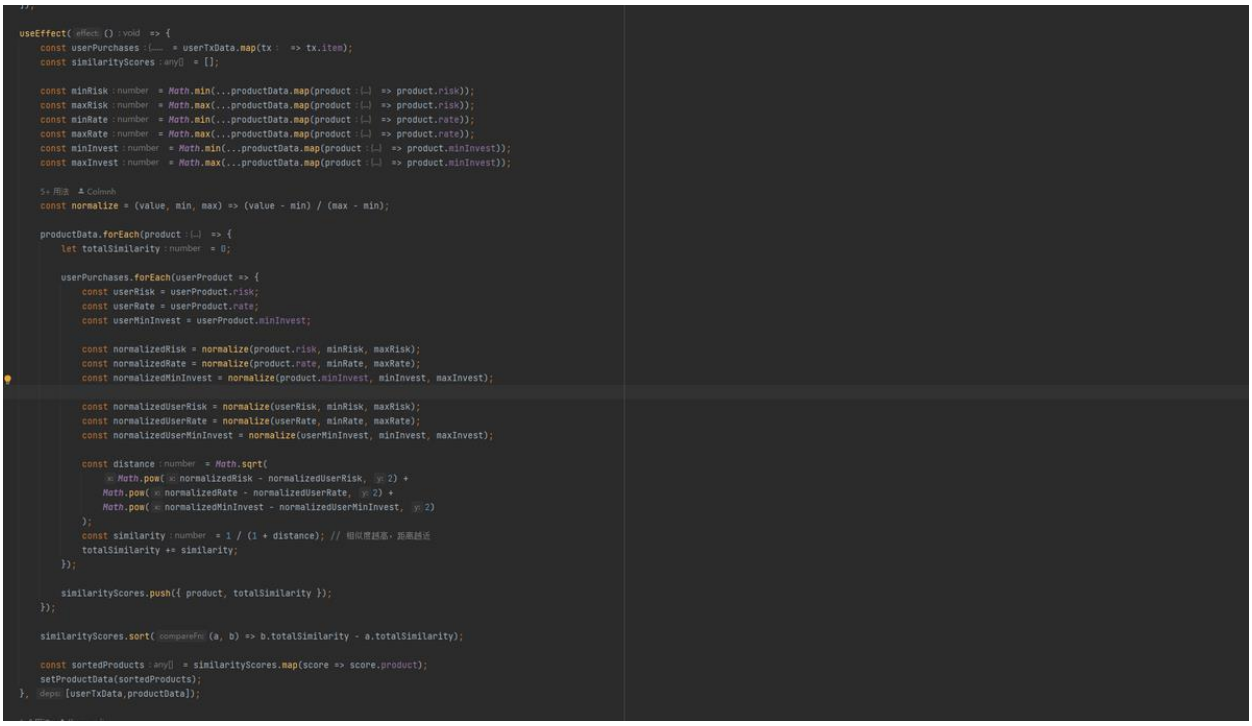
a. 归一化最小值和最大值：O(m)（m 为“productData”的大小）。

b. 计算相似度：O(m*n)（n 为“userPurchase”的大小）。

c. 总时间复杂度：O(m+m*n)。

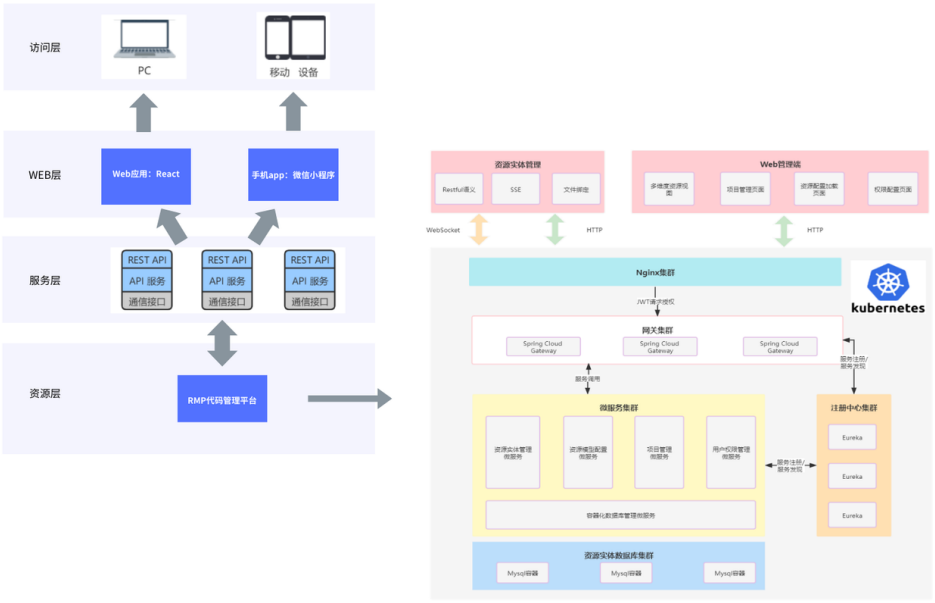
2.空间复杂度：O(m+n)。

算法实现



2. 技术架构

根据分析，本产品的技术架构图绘制如下：



本产品主要分为访问层、WEB 层、服务层、资源层四个结构层，其中：

访问层包括 PC 与移动设备；

WEB 层包括针对 Web 应用开发的 React 框架与针对手机开发的微信小程序；

服务层包括通信接口、API 服务与 REST API；

资源层主要为 RMP 代码管理平台。

在 RMP 平台内部，主要结构如下：

资源实体管理与 Web 管理端主要通过 HTTP 服务与集群内部进行通信；

集群内部主要通过 Nginx 集群向网关集群发送 JWT 请求授权；

网管集群调用微服务集群中的微服务模块；

可以通过注册中心集群增加网关集群与微服务集群可调用的微服务内容；

集群中的内容存储于 Mysql 容器中。

实现内容

姓名	分工	贡献度%
王乐峰	个人用户web开发、原子服务实现方式设计、沟通协调分工	25
刘津铭	公司用户web开发、业务角色设计	25
王璟昊	管理员用户web开发、技术架构设计、业务流程设计	25
杨俊杰	实现协同过滤算法推荐产品、用例建模	25