

本科毕业设计（论文）

**租赁业务风险控制安全中台的设计与实现**

**Design and Implementation of Security Platform for Risk Control of Leasing Business**

学 院： 软件学院

专 业： 软件工程

学生姓名：

学 号：

指导教师：

**北京交通大学**

2022年11月

学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名： 指导教师签名：

签字日期： 年 月 日 签字日期： 年 月 日

中文摘要

随着互联网的发展，网络黑产也随之兴起。他们遍布在互联网的各个领域，以互联网为媒介，通过恶意注册账号开展网络诈骗、利用平台交易渠道进行虚假交易、薅取营销活动和新人礼包优惠等方式对平台进行攻击，从中获益。为对抗网络黑产、减少公司损失、保障用户利益，风险控制产品应运而生。风险控制安全中台作为企业级风险控制产品，帮助企业构建风险控制规则引擎。风险控制策略师可针对不同业务场景在本系统内自由配置风控规则，系统根据风控规则和用户操作数据对所有用户风险情况进行分析，并给出相应防控措施，同时对规则执行情况和整体风险情况进行监控。

在系统设计与实现过程中，本人独立完成系统的需求分析和功能设计，将系统分为敏感数据审计、风险情况分析、规则引擎构建、规则执行监控四个模块，并完成所有模块的后端开发工作。各模块功能如下：

（1）敏感数据审计：与公司其余内部系统配合使用，实现对公司内部员工查看敏感数据的监控功能。提供敏感数据审计规则配置与管理功能，并使用定时任务检查员工是否违反规则并触发告警，同时提供告警内容及敏感数据点击日志搜索查看。

（2）风险情况分析：实现对平台外部整体风险情况分析展示功能。将用户各业务线风险分数通过统计学方法聚合，以此为依据评估并展示平台风险程度。同时与公司其余用户行为分析系统联动，提供用户操作查看和黑白名单库管理功能。

（3）规则引擎构建：该模块主要实现规则引擎构建、运行及调整功能。风控策略师建立特征库，并构建由事件、业务线、规则、规则组组成的决策流，在运行阶段根据用户行为数据和规则做出风险控制决策，并实现规则、规则组参数手动配置调整功能。

（4）规则执行监控：该模块主要实现规则执行效果监控功能。利用算法对规则阈值预测并自动调整，同时对规则执行结果进行统计，对规则准确性、有效性进行评估，规则异常立刻触发告警。除此之外，根据规则执行情况提供异常用户名单，以便进行后续操作。

目前，该风险控制安全中台已在公司内部正式部署上线。系统共接入登陆、注册、反爬、内部员工审计四个业务场景，并针对上述场景设计4条决策流共7个规则。系统日流入数据量约20万条，执行规则10万余次，实时监控公司1700万用户及1万余名员工风险情况。

**关键词：**风险控制；规则引擎；大数据处理

ABSTRACT

With the development of the Internet, online black production has also emerged. They are all over the Internet, relying on the Internet as a medium to attack the platform and benefit from it. In order to fight against online black production, reduce company losses, and protect the interests of users, risk control products came into being. The risk control security center helps enterprises build a risk control rule engine, customize personalized risk control rules, and the system analyzes the risk control rules and user operation data, gives corresponding prevention and control measures, and monitors the overall risk situation.

In the process of system design and implementation, the author independently completed the demand analysis and function design of the system, divided the system into four modules, and completed the back-end development of all modules. The functions of each module are as follows:

(1) Sensitive data audit: Realize the monitoring function for company employees to view sensitive data. Provide rule configuration and management. Violation of rules will trigger alarms, and realize alarm content and click log search to view.

(2) Risk situation analysis: Realize the analysis and display of the overall risk situation outside the platform. Aggregate the risk scores of each business line of the user, and use this as the basis to evaluate the risk level of the platform.

(3) Rule engine construction: realize the function of rule engine construction and adjustment. Build a feature library, and build a decision flow composed of events, business lines, rules, and rule groups, use user behavior data to make risk control decisions, and realize automatic configuration.

(4) Rule execution monitoring: including rule execution result statistics, rule accuracy assessment, rule threshold prediction and automatic adjustment functions. At the same time, a list of abnormal users is provided.

At present, the system has been officially deployed and launched within the company, with access to four business scenarios of login, registration, anti-climbing, and auditing, and 7 rules for 4 decision flows are designed. The system has a daily inflow data volume of about 200,000, executes rules more than 100,000 times, and monitors the risk situation of the company's 17 million users and more than 10,000 employees in real time.

**KEYWORDS：**risk control; rules engine; big data processing

目 录

[中文摘要 i](#_Toc101791287)

[ABSTRACT ii](#_Toc101791288)

[目 录 iii](#_Toc101791289)

[1 引言 1](#_Toc101791290)

[1.1 项目背景 1](#_Toc101791291)

[1.2 风险控制产品发展现状研究 2](#_Toc101791292)

[1.2.1 企业级风险控制产品演进历程 2](#_Toc101791293)

[1.2.2 所在公司风险控制现状 3](#_Toc101791294)

[1.3 项目主要内容和论文的组织结构 3](#_Toc101791295)

[2 相关理论及技术综述 5](#_Toc101791296)

[2.1 SpringBoot系统框架 5](#_Toc101791297)

[2.2 Redis缓存机制 5](#_Toc101791298)

[2.3 Kafka消息订阅发布系统 6](#_Toc101791299)

[2.4 Flink流式计算引擎 6](#_Toc101791300)

[2.5 三次指数平滑法 7](#_Toc101791301)

[2.6 规则引擎相关理论 8](#_Toc101791302)

[2.6.1 规则 8](#_Toc101791303)

[2.6.2 规则组 9](#_Toc101791304)

[2.6.3 决策树 10](#_Toc101791305)

[2.7 本章小结 10](#_Toc101791306)

[3 租赁业务风险控制安全中台需求分析 11](#_Toc101791307)

[3.1 需求分析综述 11](#_Toc101791308)

[3.2 功能性需求分析 11](#_Toc101791309)

[3.2.1 敏感数据审计功能 12](#_Toc101791310)

[3.2.2 风险情况分析功能 14](#_Toc101791311)

[3.2.3 规则引擎构建功能 17](#_Toc101791312)

[3.2.4 规则执行监控功能 20](#_Toc101791313)

[3.3 非功能性需求分析 22](#_Toc101791314)

[3.4 本章小结 22](#_Toc101791315)

[4 租赁业务风险控制安全中台概要设计 23](#_Toc101791316)

[4.1 系统体系结构 23](#_Toc101791317)

[4.2 系统功能结构 24](#_Toc101791318)

[4.3 系统数据库设计 25](#_Toc101791319)

[4.4 本章小结 30](#_Toc101791320)

[5 租赁业务风险控制安全中台详细设计与实现 32](#_Toc101791321)

[5.1 敏感数据审计模块 32](#_Toc101791322)

[5.1.1 敏感数据审计规则管理 33](#_Toc101791323)

[5.1.2 敏感数据规则执行及结果展示 35](#_Toc101791324)

[5.2 风险情况分析模块 38](#_Toc101791325)

[5.3 规则引擎构建模块 42](#_Toc101791326)

[5.3.1 用户行为数据预处理 43](#_Toc101791327)

[5.3.2 规则匹配计算 45](#_Toc101791328)

[5.3.3 防控措施实施 47](#_Toc101791329)

[5.4 规则执行监控模块 47](#_Toc101791330)

[5.4.1 规则阈值预测及自动调整 47](#_Toc101791331)

[5.4.2 规则评价 49](#_Toc101791332)

[5.5 本章小结 52](#_Toc101791333)

[6 系统测试 53](#_Toc101791334)

[6.1 测试目标及准备 53](#_Toc101791335)

[6.2 功能测试 53](#_Toc101791336)

[6.3 非功能测试 59](#_Toc101791337)

[6.4 本章小结 60](#_Toc101791338)

[7 总结与展望 61](#_Toc101791339)

[7.1 全文总结 61](#_Toc101791340)

[7.2 系统展望 61](#_Toc101791341)

[参考文献 63](#_Toc101791342)

[致 谢 65](#_Toc101791343)

1 引言

随着互联网的迅速发展以及大数据时代的到来，互联网企业不仅要面临激烈的行业竞争，还要应对黑色产业带来的风险，各大企业都需要一套灵活高效的风险控制系统和工作机制来防控这些风险。本章着重介绍该系统的应用背景、发展现状和实用价值，同时介绍本人的工作内容、实现的方法技术以及系统实现效果。

1.1 项目背景

随着互联网的飞速发展，由于互联网便捷、高效等特点，许多传统行业逐渐向互联网方向发展，原先的线下业务逐渐向线上进行迁移。以租赁行业为例，本人所在公司为用户提供房屋租赁服务，公司从业主手中收取房屋，通过装修改造等方式对房屋进行包装升级，再出租给租客，并为租客提供入住后的物业、保修、网络等服务。随着互联网的发展，公司将签约、预约看房、保修、转租等一些列业务迁移至APP及小程序中，不再通过线下方式进行，同时加入搬家、保洁、电商等一系列租赁服务衍生业务。总结来说，公司业务存在如下特点：

（1）业务多、覆盖面广：包含看房、装修、签约、搬家、保洁、维修、家具购置、转租等业务，为用户打造租赁一条龙服务，其中不乏容易被销赃的业务。

（2）用户多：公司目前总用户量约1700万，覆盖了北京、上海、杭州等十座城市。

（3）用户数据敏感：由于房屋租赁需要用户完整的个人信息，因此公司内部存储了大量用户的敏感数据，包括身份信息、财产信息等，一旦泄露将造成严重后果，危害用户个人安全及公共安全。

（4）单笔交易额大：房屋租赁、家具购买等业务单笔交易额少则五六千，多则数十万，借贷情况频发，一旦出现问题会给公司及用户带来巨大损失。

（5）营销活动多：一、二线城市房价居高不下，租房依旧是主流趋势。但目前租赁行业不存在独角兽企业存在，因此各公司之间竞争激烈。公司目前仍处于拉新阶段，时常有各类营销活动，给黑产提供了很多可乘之机。

基于以上业务特点，公司对黑产有着巨大的吸引力，在以下方面都有被黑产攻击的风险：

（1）恶意注册：黑产通过多种途径取得的手机卡号作为注册材料，或者使用非法取得的身份信息，以手动或程序的方式批量创造账号并进行养号，从而利用虚假账号从平台牟利[1]。

（2）用户作弊：也名为“薅羊毛”，黑产骗取营销活动中为正常用户准备的奖品、优惠券等福利。

（3）虚假交易：黑产通过借贷的方式在平台进行交易，随即私下将房屋转租给其余租客并一次性收取房租，以类似于无息借贷的方式进行牟利。

（4）广告营销：黑产使用虚假账号，在公司平台内部社区中发布广告或其他网络禁止内容，并使用其他账号进行评论、转发，以获得较高流量。

（5）数据泄露：公司内部员工可以看到大量客户个人信息，将客户个人信息卖给非法机构从而从中获利。或员工不通过公司渠道，和客户进行私下交易，将交易所得全部据为己有。

以上行为一旦发生，会严重侵害平台和用户利益，扰乱正常交易秩序，或造成平台生态紊乱。本项目所设计的业务安全风险控制中台，以公司业务为基础，通过建立公司范围的风险控制体系，设定一系列风险控制策略和规则，在黑灰产攻击平台时迅速识别，进行拦截或者反攻击，从而保障平台安全。

1.2 风险控制产品发展现状研究

据2018年5月发布的《数字金融反欺诈白皮书》显示，2017年我国黑产从业人员超过150万人，年产值达千亿元级别[2]。因此，风险控制产品并不是一个新概念，各大公司也早就着手研发自己的风控产品，如美团的宙斯系统，阿里的风险识别产品等。本部分将介绍风险控制产品的演进历程，及租赁业务风险控制安全中台研发前所在公司的风控现状。

### 1.2.1 企业级风险控制产品演进历程

企业级风险控制产品演进历程主要分为4个阶段，包括硬编码阶段、初级产品化阶段、成熟产品阶段和中台化产品阶段[3]。

硬编码阶段主要存在于业务发展初期，由于用户量小、业务简单，黑产关注度也较低。企业资源也重点投入到业务开发和扩大用户量中，因此这个阶段风控策略比较简单，主要通过将风控策略写死在业务代码中的方式，实现简单的风控目的。该阶段存在策略开发周期长、策略上线前效果不可预估、上线后策略准确性和有效性难以评估、策略难以调整的缺点，风控效果较差。

初级产品化阶段主要存在于业务发展的成长期，随着用户量开始变多，业务对黑产的价值开始体现，企业通过产品设计的手段将原有的策略体系从业务代码中迁移出来，构建单独的风险控制产品，并且实现简单的策略参数修改[4]。该阶段风控策略通常单独执行互相独立，没有形成体系。除此之外，不断的参数调整使得策略管理的资源投入超过风险对抗效率提升。

成熟产品阶段主要存在于业务发展的成熟期，用户量达到顶峰，业务也变得十分庞大。此时，简单的策略参数配置已经无法满足风控需求，丰富风控策略的逻辑关系、提前预知风险、预测策略效果、构建体系化解决方案是这一阶段的主要目标[5]。大部分公司的风险控制产品停留在该阶段不再变化。

中台化产品阶段主要存在于业务发展的成熟期后期，企业开始寻求其余方向发展的可能性，衍生出其余的新业务[6]。原有的风控体系与原业务耦合性较大，很难迁移至新的业务中，因此，需要覆盖绝大部分业务需求，实现风控策略的灵活复用、高度抽象，追求服用的成本最小化。

### 1.2.2 所在公司风险控制现状

租赁业务风险控制安全中台研发前，公司内部的风险控制系统建设刚刚起步，介于硬编码阶段和初级产品化阶段之间。曾经设置过公司层面的风险控制规则，小部分业务领域有自己的风险控制策略，但是存在以下问题：

（1）虽然小部分业务领域有自己的风险控制策略，但是较为简单，如设置黑名单、限制同一用户某一操作单天次数上限等，易被绕过，且各个业务领域之间的风险控制策略和结果无法共享。

（2）曾经的公司层面风险控制规则，仅每天针对当天的用户操作数据，按照特定的规则进行统计，将统计数据导出为excel进行人工筛查。这种方式十分机械，筛查为纯手工操作，效率低下。规则简单、单一，所有规则为硬编码方式实现，不方便进行修改。并且规则和结果均受到时间限制，针对具有时间跨度的数据无法进行分析，不同时间产生的结果也不方便进行横向比较。由于这种方式过于机械，目前已不再使用。

（3）公司层面没有针对风险控制形成体系，即公司整体风险情况以及每个用户的风险情况无法进行评估和可视化展示。

## 1.3 项目主要内容和论文的组织结构

租赁业务风险控制安全中台是搭建在用户和公司各个子系统之间的的风险控制平台，以租赁业务和其余衍生业务为前提，接收用户操作，根据中台内部的规则体系对用户操作进行分析，向子系统提供防控建议。

系统主要内容包括：

（1）规则配置。系统通过脚本对常见特征、运算符、结果进行预定义，业务人员通过规则模板，针对不同事件和业务，使用可视化前端界面进行规则配置，将规则按一定分支及流程编排形成决策流和规则组。

（2）规则解析、执行与展示。当有事件触发时，系统对决策树进行解析，加载用户数据进行执行，使用大数据处理框架减少规则执行时间，并根据执行结果向业务方反馈用户风险系数和防控建议。系统根据规则模型执行情况动态调整用户风险分数，对规则命中情况和平台风险情况进行实时展示。

（3）规则评价和阈值自动调整。系统对前一段时间的用户操作数据进行分析，通过模型对未来一段时间内的规则阈值进行预测和动态调整。系统通过规则评价体系分析规则的有效性和合理性，若指标超出正常范围，向业务人员发送告警信息。

（4）敏感数据审计。实现公司内部涉及敏感数据系统埋点日志接入。系统可根据员工职级配置一定时间段内查看敏感数据次数阈值，及配置告警格式。使用定时任务将敏感数据点击日志按照员工职级和时间段进行聚合，与规则进行匹配，超过阈值触发告警，并实现告警记录查看。

本文组织结构如下：

第一章引言，介绍租赁业务风险控制安全中台背景和意义，以及风险控制产品发展现状，同时阐述该项目的主要内容。

第二章相关理论与技术概述，介绍项目中主要涉及到的理论和技术，主要包括规则及决策流含义、SpringBoot框架、Redis缓存、Flink大数据框架、三次指数平滑法时间序列预测算法。

第三章租赁业务风险控制安全中台需求分析，主要介绍该系统的模块划分，描述该系统的功能性需求和非功能性需求。

第四章租赁业务风险控制安全中台概要设计，主要描述系统总体架构、各模块设计方案和系统数据库设计方案。

第五章租赁业务风险控制安全中台详细设计，在概要设计基础上，对各模块进行了详细描述，展示模块设计的详细类图、顺序图。

第六章系统测试，以需求分析为基础，对系统功能进行验证。

第七章总结与展望，对本文工作进行总结，并对该系统未来发展进行展望。

2 **风险控制安全中台**相关理论及技术综述

本章对租赁业务风险控制安全中台中使用到的理论和技术基础进行简要介绍。系统后端采用SpringBoot框架进行搭建，前端采用vue框架，数据库主要使用mysql进行搭建，配合Redis进行缓存提高读写速度。数据处理部分采用flink大数据框架。系统采用以规则为核心的规则引擎对用户行为进行分析，最终判断该用户是否为风险用户，并对风险用户进行处置。在规则阈值调整过程中，系统采用三次指数平滑法，利用已有数据对未来阈值进行预测并进行调整。

2.1 SpringBoot系统框架

Spring框架是一个开放源代码的JavaEE应用程序框架，提供了功能强大IOC、AOP及Web MVC等功能，可以单独使用搭建应用程序，也可以和其余Web框架或Swing等桌面应用程序组合[7]。Spring框架具有方便解耦、简化开发、可测试强的优点。但在使用Spring框架开发过程中，往往需要一系列的配置操作，耗费开发人员大量时间。

SpringBoot是在Spring基础上提供的一套全新的开源框架，去除了大量的XML配置文件，简化了复杂的依赖关系。SpringBoot具有Spring的所有优点，并且操作更加简单、功能更加丰富、性能更加稳定。除此之外，SpringBoot框架集成了大量常用的第三方库配置，可在应用中零配置使用，配置代码的减少使得开发者可以更加专注于业务逻辑。目前，SpringBoot已经成为主流开发技术之一[8]。

2.2 Redis缓存机制

Redis是一个高性能的、开源的非关系型数据库，底层采用C语言进行编写，存储结构采用Key-Value形式。它是一种基于内存并可提供持久化选项的日志型、高性能Key-Value数据库，在大数据量应用场景下有很大优势[9]。Redis具有以下优点：

（1）性能高，操作在内存中进行，相比于MySQL之类的数据库存取速度快，并发能力强。Redis读的速度可达110000次/s，写的速度可达81000次/s。

（2）支持多种类型的value存储，包括string（字符串）、list（链表）、set（集合）、zset（有序集合）、hash（哈希）等。

（3）提供了Java、C、C++、PHP等多种客户端，使用便捷

（4）支持集群操作，包括主从同步、负载均衡，数据可以从主服务器向任意数量的从服务器同步，从服务器也可以关联其他从服务器的主服务器[10]。

（5）支持持久化，可将数据保存在硬盘中。

（6）支持事务，操作均为原子操作，对数据的更改要么全部执行，要么全部不执行。

基于上述优点，Redis常常被应用于缓存场景中，可将经常查询或很少修改的数据存放到缓存中，减少对数据库的访问操作，降低数据库压力，减少数据丢失，提高数据的访问速度[11]。

2.3 Kafka消息订阅发布系统

Kafka是Apache软件基金会下的一款开源分布式发布订阅消息系统，在系统中，可以作为消息队列处理数据生产和数据消费速度不一致的问题，帮助服务器减轻并发压力。

对于Kafka来说，客户端包括两种类型：生产者（Producer）和消费者（Consumer）。生产者，也称为发布者，负责创建消息，而消费者，也称为订阅者，负责消费或读取消息。在 Kafka 中，消息以主题（Topic）来区分，每一个主题都对应一个消息队列，同时为了解决吞吐量不足的问题，Kafka引入分区的概念进行解决，对主题进行水平扩展。

Kafka 服务器也称为 Broker，接受生产者发送的消息并存入磁盘，一个 Broker 每秒可以处理成千上万的分区和百万量级的消息[12]。一个集群（Cluster）由若干个 Broker 组成，由其中一个Broker作为集群控制器，负责管理集群，包括分配分区到 Broker、监控 Broker 故障等[13]。

本系统中，Kafka作为消息队列，接收埋点日志进行后续处理。

2.4 Flink流式计算引擎

Apache Flink是一个分布式处理框架，可在无边界和有边界数据流上进行计算[14]。Flink架构如图2-2所示。

Flink具备时间概念，在Flink中存在三种时间概念，包括事件时间（Event Time）、处理时间（Processing Time）和进入系统时间（Ingestion Time）[15]。由于不同业务风控场景中关注的时间不同，因此可以在数据处理时自由选择需要的时间。

Flink提出了窗口的概念，无界的数据可通过窗口机制被划分为有限的数据区间，在数据流处理过程中可根据时间窗口对数据进行分析。Flink中的窗口分为3种：滑动窗口、滚动窗口和会话窗口[16]，其中滑动窗口只窗口长度固定，窗口间有重叠；滚动窗口与滑动窗口类似，但窗口间可以重叠；会话窗口设定时间长度，若该窗口内没有接收到数据则开启新的窗口。

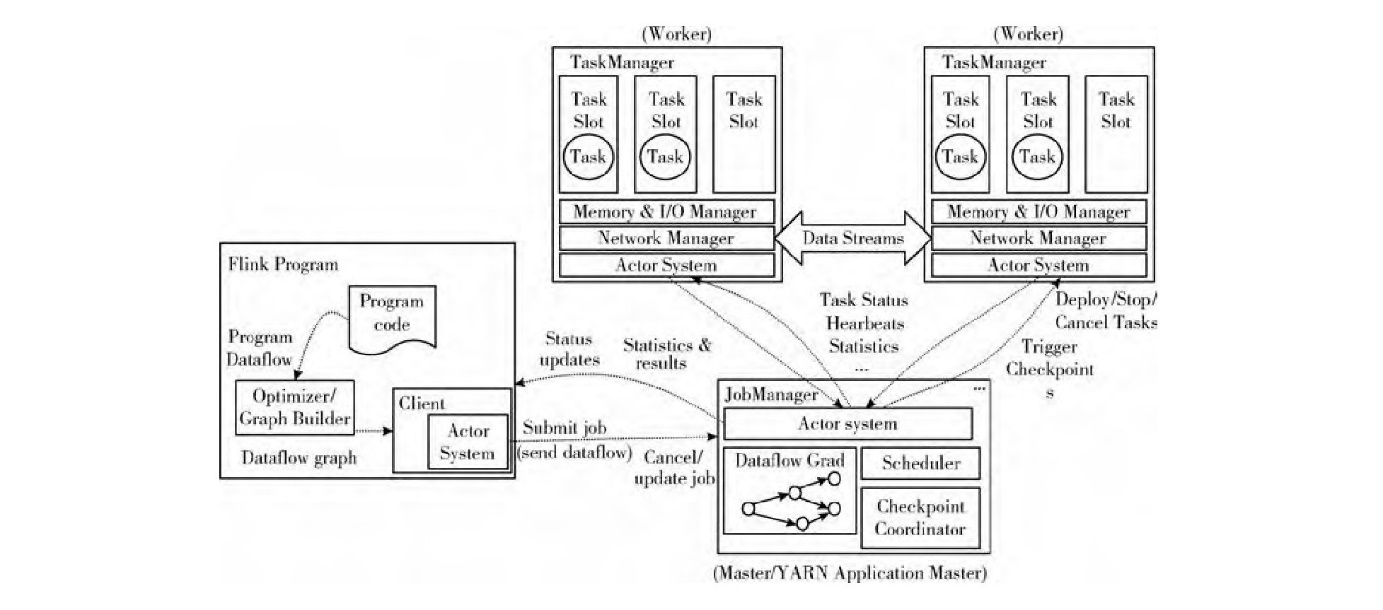


图2-1 Flink架构图

2.5 三次指数平滑法

指数平滑法是时间序列分析预测的方法之一，适应性强且容易操作。三次指数平滑法是指数平滑法的一个分支，主要用于有趋势性且呈非线性的时间序列预测，具有理论简单、预测效果好的优点[17]。三次指数平滑法模型分为3步：

（1）对历史数据进行平滑。计算公式为：

(2-1)

式中，t是时间序号，Xt是时间序列中第t个历史数据，β是静态平滑系数，取值范围为(0,1)，St(1),St(2),St(3)分别是第一、二、三次指数平滑值，

（2）对未来数据进行预测。令为从t时刻起往前第T个数据的预测值，有：

(2-2)

其中，

(2-3)

(2-4)

(2-5)

（3）参数确定。在使用该模型进行预测时，需确定静态平滑系数β和平滑初始值。

其中，平滑初始值通常为最初3个历史数据的平均值[18]，即

(2-6)

静态平滑系数β可以利用历史数据，通过循环遍历的方式进行求解。已知0＜β＜1，用步长0.01遍历该区间，计算所有历史数据的真实值与预测值之间的误差平方和并找到最小值，该值即为静态平滑系数β的取值[19]。

2.6 规则引擎相关理论

业务风险控制安全中台的核心是规则引擎，规则引擎可完成接受用户数据、编写业务规则、做出风控决策的一些列动作。规则引擎由规则、规则组、决策树组成，下面将依次对这三个概念进行介绍

2.6.1 规则

规则是由特征、运算符、阈值这三要素构成条件表达式[20]，如“年龄大于50”。每个规则对应一个惩罚措施，在规则执行过程中，若条件表达式计算结果为真，称之为规则命中，施行惩罚措施。

特征为规则执行的依据，可为输入数据或数据通过某种计算方式聚合后的结果，如“手机号”、“设备号”、“一段时间内同一设备号登录次数”均为特征。特征的计算方式通过脚本进行定义，计算结果可为布尔、数值、字符、字符串、集合共5种数据类型。

运算符包括＞、≥、=、≤、＜、∈、!∈共七种。

阈值须根据特征计算结果的数据类型和运算符类型决定，阈值的数据类型与运算符和特征的数据类型对应关系如下：

表2-1 阈值数据类型与运算符和特征的数据类型对应关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **运算符\特征数据类型** | **布尔** | **数值** | **字符** | **字符串** | **集合** |
| **＞** | - | 数值、字符 | 字符、数值 | 字符串 | - |
| **≥** | - | 数值、字符 | 字符、数值 | 字符串 | - |
| **=** | 布尔、数值 | 数值、字符 | 字符、数值 | 字符串 | 集合 |
| **≤** | - | 数值、字符 | 字符、数值 | 字符串 | - |
| **＜** | - | 数值、字符 | 字符、数值 | 字符串 | - |
| **∈** | - | 集合 | 集合、字符串 | 集合、字符串 | 集合 |
| **!∈** | - | 集合 | 集合、字符串 | 集合、字符串 | 集合 |

惩罚措施为规则命中后的后续操作，可为加入黑名单、添加标签、拒绝本次操作、发送验证码等。一个规则可对应0或多个惩罚措施。

下面给出规则示例：

表2-2 单一规则示例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **编号** | **规则名称** | **特征** | **特征计算结果数据类型** | **运算符** | **阈值** | **惩罚措施** |
| 001 | 年龄准入规则 | 用户年龄 | 数值 | ≥ | 18 | 拒绝本次操作 |

2.6.2 规则组

在单一规则执行过程中，由于缺少规则间逻辑（与、或、非）关系和执行顺序的处理，导致整体规则体系不够完善[21]。因此引入规则组的概念解决缺少逻辑关系的问题。规则组由规则、规则组命中逻辑、惩罚措施组成。

每个规则组内需包含至少一条规则，每条规则需属于至少一个规则组。

规则组命中逻辑由组内规则执行结果组成的条件表达式构成，若带入用户数据执行规则后，该表达式为真，称之为规则组命中，可实施该规则组的惩罚措施。如某一规则组内包含3条规则，分别为规则1、规则2、规则3，则规则组命中逻辑可为：（规则1执行结果 或 规则2执行结果）且 规则3执行结果。

规则组惩罚措施同单一规则惩罚措施，一个规则组可对应0或多个惩罚措施。

下面给出规则组示例：

表2-3 规则组示例

|  |  |  |  |
| --- | --- | --- | --- |
| **规则组名称** | **规则组命中逻辑**  **（用规则编号代替）** | **惩罚措施1** | **惩罚措施2** |
| 登陆IP限制 | 001 或 002 或 003 | 增加用户标签：  登陆IP异常 | 扣除用户安全分：5分 |

规则组内规则列表如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **编号** | **规则名称** | **特征** | **特征计算结果数据类型** | **运算符** | **阈值** | **惩罚措施** |
| 001 | 国外IP登陆 | 用户IP所在城市 | 字符串 | !∈ | 国内城市集合 | 拒绝本次操作 |
| 002 | 高危IP登陆 | 用户IP所在城市 | 字符串 | ∈ | 国内犯罪高发城市集合 | - |
| 003 | 非业务范围IP登陆 | 用户IP所在城市 | 字符串 | !∈ | 公司业务范围城市集合 | - |

2.6.3 决策树

规则组概念的引入解决了单一规则缺少逻辑关系的问题，决策树的引入则规定了规则间的执行顺序。

风险控制的规则和决策需要在特定业务场景下方可生效[22]，因此企业级风险控制产品需要将企业业务抽象为事件，再根据事件场景设置相应的风控规则。抽象出的事件如登陆、注册、下单、浏览、评论等等。

决策树针对事件，以类似二叉树的模式，定义了该事件的完整决策过程。决策树所有节点均为规则组，左子树和右子树分别为该规则组命中和未命中之后的下一待处理规则组，左子树和右子树均可为空。在规则执行过程中，通过事件定位到对应决策树，依次执行规则组，直至待处理规则组为空，规则执行结束。

下面给出决策树示例：

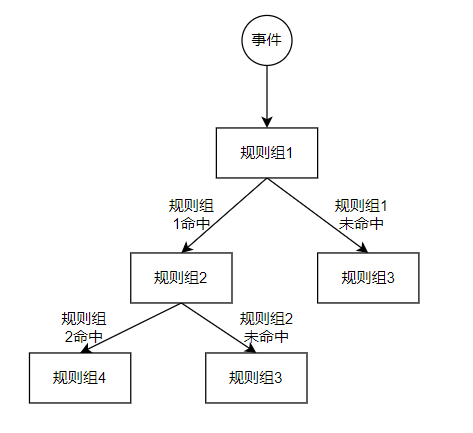


图2-2 决策树示例

2.7 本章小结

本章主要介绍了本系统开发过程中涉及到的理论基础和技术基础，包括系统后端搭建过程中使用到的SpringBoot框架、用于缓存加快数据读写速度的Redis、数据处理过程中使用到的Flink流式数据处理引擎、阈值预测过程中使用的三次指数平滑法，最后介绍了规则引擎的基本理论和构成规则引擎的几个单位规则、规则组、决策树。

3 租赁业务风险控制安全中台需求分析

本章主要进行租赁业务风险控制安全的需求分析，从用户角度梳理基本需求，并按照功能模块对产品进行划分，针对每一个模块，通过用例图和用例描述的方式对该模块具体功能进行阐述。同时从产品使用和用户需求角度，对系统非功能性需求进行梳理。

3.1 需求分析综述

产品的诞生是为了解决用户需求，因此若想要设计出一款好的产品，需要了解用户是谁，用户有什么样的需求，才可以设计出正确、易用的产品。

租赁业务风险控制安全中台的目标是抵抗黑产攻击以及防止内部人员泄露敏感数据，系统使用者主要分为风控策略师、业务经理、安全部门负责人三类，其需求如下：

（1）风控策略师：需要系统提供规则配置功能。由于公司业务复杂、风险点多，且黑产的攻击手段变化快、无法预知，因此要求规则配置功能需方便修改、易复用、易迁移。并且系统需提供规则评价功能，以便于了解规则的准确性和有效性，对规则进行调整迭代。由于手动调整过于繁琐，需要系统提供简单的规则自动调整功能。

（2）业务经理：需要借助系统了解所在业务线的风险情况，要求风险情况展现较为直观，易于查看。除此之外，需要系统提供高危用户列表，以便于进行进一步的防控。最后，为了减少系统误判的可能性，系统内部需设立黑白名单机制。

（3）安全部门负责人：需要借助系统了解公司整体风险情况，且展现方式较为直观。需要了解高危用户列表，包括公司内部员工和外部用户。需要审查规则的合理性和有效性，当规则不再适用时督促风控策略师进行调整。

3.2 功能性需求分析

租赁业务风险控制安全中台是搭建在用户和公司各个子系统之间的的风险控制平台，以租赁业务和其余衍生业务为前提，接收用户操作，根据中台内部的规则体系对用户操作进行分析，向子系统提供防控建议。主要功能包括查看平台整体风险状况、检索用户在平台内具体操作、规则、规则组、决策树配置、监控规则执行情况、对规则进行评价、配置敏感数据告警规则等。

按照功能划分，将该系统分为四个主要功能，分别为敏感数据审计、风险情况分析、规则引擎构建、规则执行监控。下面将详述四个功能的具体功能性需求。

3.2.1 敏感数据审计功能

该功能主要实现对公司内部员工查看敏感数据的监控。该部分需与公司其余内部系统配合使用，按照数据分类分级原则，对公司内部系统所展示的所有数据进行分级，并将员工点击查看敏感数据的操作日志传入风险控制安全中台。风控策略师可在系统中针对员工角色配置一段时间内点击查看敏感次数上限阈值。系统使用定时任务，定期根据数据点击日志检查是否超过阈值，若超过则触发告警，向员工经理及安全部份发送告警信息。业务经理及安全部门负责人可在系统中查看具体告警内容及敏感数据点击日志。该功能用例图如图3-1所示。

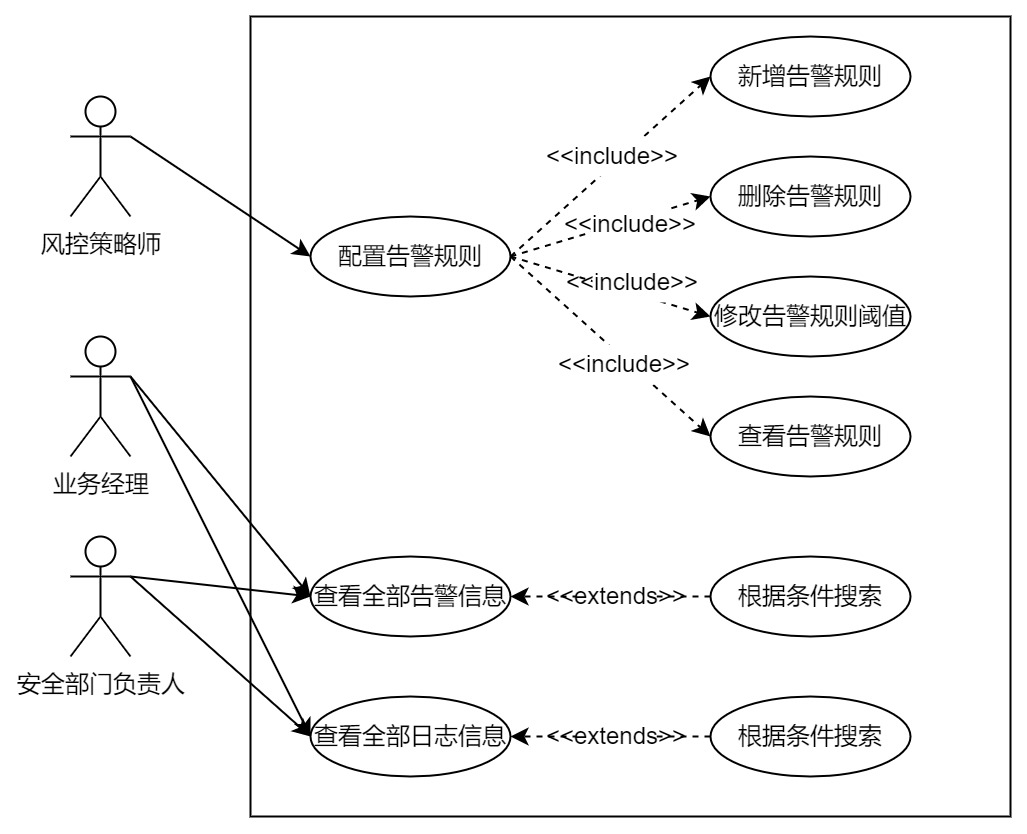


图3-1 敏感数据审计用例图

下面对于该部分中的三个主要用例配置告警规则、查看全部告警信息、查看全部日志信息进行详细描述。

表3-1 配置告警规则用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC1 | **用例名称** | 配置告警规则 |
| **活动者** | 风控策略师 | **优先级** | 高 |
| **描述** | 本用例描述风控策略师在配置告警规则时的一系列操作，包括对告警规则的增删改查 | | |
| **前置条件** | 风控策略师进入系统 | | |
| **基本流程** | 1. 风控策略师点击导航栏“告警-告警规则” 2. 界面上显示目前所有告警规则，包括职位名称、告警时间范围、一级告警阈值、二级告警阈值 | | |

续表3-1 配置告警规则用例说明

|  |  |
| --- | --- |
| **基本流程** | 1. 风控策略师点击下方“添加”按钮 2. 界面新增一行空规则，包含4个待输入选项：职位名称、时间范围、一级阈值、二级阈值 3. 风控策略师从职位名称下拉框中进行选择，并输入时间范围、一级和二级阈值 4. 风控策略师点击保存 5. 系统将该规则加入规则库中，提示保存成功 6. 风控策略师点击某一规则右侧“删除”按钮 7. 系统将该规则从规则库中删除，提示保存成功 8. 风控策略师修改任意规则时间范围、一级和二级阈值 9. 风控策略师点击保存按钮 10. 系统更新规则库，提示保存成功 |
| **扩展流程** | 7a. 如果风控策略师输入的时间范围不为正一位小数，或输入阈值不为正整数，或一级阈值小于二级阈值，提示输入错误  7b 如果风控策略师未点击“保存”按钮，直接点击其余位置，则系统不更新规则库，界面返回至2中界面 |
| **异常流程** | 无 |
| **后置条件** | 无 |
| **补充说明** | 无 |

表3-2 查看全部告警信息用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC2 | **用例名称** | 查看全部告警信息 |
| **活动者** | 业务经理、安全部门负责人 | **优先级** | 高 |
| **描述** | 本用例描述业务经理和安全部门负责人查看告警信息的一系列操作，包括根据条件进行筛选 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“告警-告警记录” 2. 界面上显示目前所有告警记录，包括时间、触发告警员工姓名、工号、所属部门、触发告警级别、规则时间内敏感数据点击次数 3. 用户在界面上方筛选框内，根据时间、员工姓名、工号、所属部门、触发告警级别进行筛选，点击“搜索” 4. 系统根据条件对所有告警记录进行过滤，界面显示所有满足条件的告警记录 5. 用户点击某一记录右侧“查看详情”按钮 6. 系统跳转至敏感数据点击日志查看界面，显示该员工全部点击日志 | | |

续表3-2 查看全部告警信息用例说明

|  |  |
| --- | --- |
| **扩展流程** | 4a 若没有满足筛选条件的告警记录，系统提示“无满足条件记录” |
| **异常流程** | 无 |
| **后置条件** | 无 |
| **补充说明** | 无 |

表3-3 查看全部日志信息用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC3 | **用例名称** | 查看全部日志信息 |
| **活动者** | 业务经理、安全部门负责人 | **优先级** | 中 |
| **描述** | 本用例描述业务经理和安全部门负责人查看日志信息的一系列操作，包括根据条件进行筛选 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“告警-日志详情” 2. 界面上显示目前所有日志信息，按照时间由近到远排序，显示内容包括时间、员工姓名、工号、所属部门、点击系统、字段名称、字段描述 3. 用户在界面上方筛选框内，根据时间、员工姓名、工号、所属部门、点击系统进行筛选，点击“搜索” 4. 系统根据条件对所有告警记录进行过滤，界面显示所有满足条件的告警记录 | | |
| **扩展流程** | 4a 若没有满足筛选条件的日志信息，系统提示“无满足条件记录” | | |
| **异常流程** | 无 | | |
| **后置条件** | 无 | | |
| **补充说明** | 无 | | |

3.2.2 风险情况分析功能

该功能主要实现对平台外部整体风险情况分析展示。在该功能中，实现对所有用户各业务线风险分数聚合及展示，以及根据用户风险分数评估平台风险程度。同时提供用户操作查看功能，可根据用户uid、手机号、设备号搜索该用户全部操作。提供黑白名单库，在该模块中可对黑白名单库进行手动维护。该模块用例图如图3-2所示。

下面对于该功能中的三个主要用例平台整体风险情况查看、用户具体操作查看、黑白名单库管理进行详细描述。

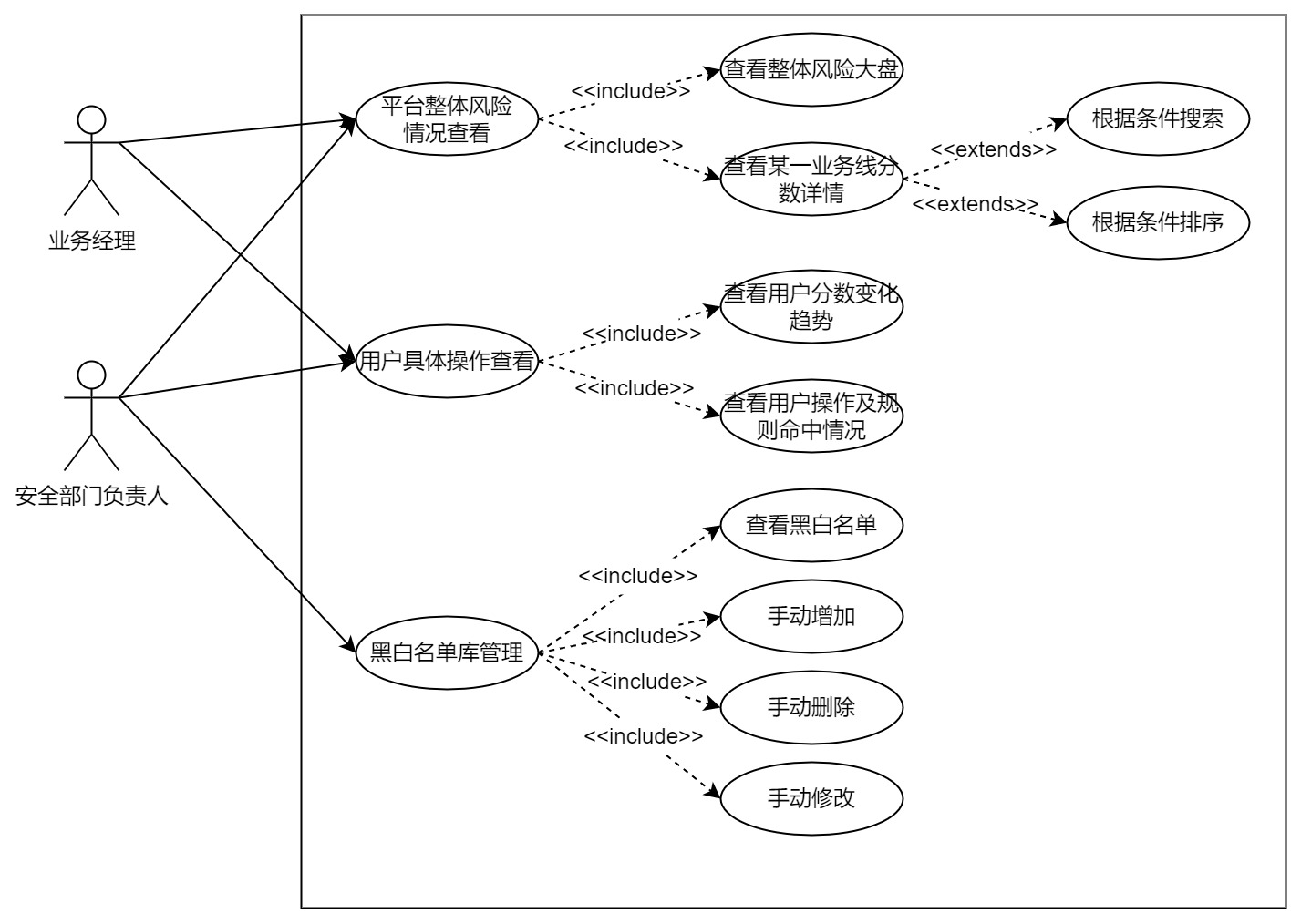


图3-2 风险情况分析模块用例图

表3-4 平台整体风险情况查看用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC4 | **用例名称** | 平台整体风险情况查看 |
| **活动者** | 业务经理、安全部门负责人 | **优先级** | 高 |
| **描述** | 本用例描述业务经理和安全部门负责人查看平台整体风险情况的具体操作，包括查看整体风险大盘和查看某一业务线风险详情 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“风险总览” 2. 界面上利用饼状图显示平台内用户风险总分分布情况，以及各业务线用户风险分分布情况 3. 用户将鼠标移动至饼状图某个区域上，显示该分数该区域内用户总人数及占总用户人数百分比 4. 用户点击平台总体风险饼状图 5. 界面跳转至平台总体风险详情界面，界面上方使用折线图显示各分数段用户人数，界面下方显示全部用户分数详情，包括用户id、最后一次登陆时间、风险总分、各业务线风险分、用户标签、用户风险级别 6. 用户按照风险总分、各业务线风险分、风险级别进行筛选排序 7. 界面按照筛选条件显示对应内容 8. 点击用户id字段 | | |

续表3-4 平台整体风险情况查看用例说明

|  |  |
| --- | --- |
| **基本流程** | 1. 界面跳转至用户操作详情界面 2. 用户返回风险总览界面，点击任意业务线风险饼状图，或整体风险展示饼状图 3. 界面跳转至该业务线风险详情界面，界面上方使用折线图显示各分数段用户人数，界面下方显示全部用户分数详情，包括用户id、最后一次登陆时间、风险总分、该业务线风险分、用户标签、用户风险级别 |
| **扩展流程** | 无 |
| **异常流程** | 无 |
| **后置条件** | 无 |
| **补充说明** | 用户各业务线风险分数为实时结果，用户总风险分数根据各业务线风险分数计算得出，该分数每1小时更新一次。 |

表3-5 用户具体操作查看用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC5 | **用例名称** | 用户具体操作查看 |
| **活动者** | 业务经理、安全部门负责人 | **优先级** | 高 |
| **描述** | 本用例描述业务经理和安全部门负责人查看某一用户具体操作的一系列操作，包括查看用户分数变化趋势和查看用户具体操作及规则命中情况 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“用户操作详情” 2. 跳转至操作详情查看界面，顶端显示用户id和手机号输入框以及搜索按钮 3. 用户输入至少一个搜索条件，点击搜索 4. 界面上显示满足条件的用户详情，上方显示近一周用户各项风险分数变化情况，下方显示用户所有操作，包括时间、触发事件名称、操作所属业务线、操作触发规则名称、操作前后分数变化详情 | | |
| **扩展流程** | 3a 若用户没有输入任何条件，直接点击搜索，系统提示请输入搜索条件  3b 若经查询，系统中没有同时满足输入用户id和手机号条件的用户，系统提示“无该用户” | | |
| **异常流程** | 无 | | |
| **后置条件** | 无 | | |
| **补充说明** | 无 | | |

表3-6 黑白名单库管理用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC6 | **用例名称** | 黑白名单库管理 |
| **活动者** | 安全部门负责人 | **优先级** | 高 |
| **描述** | 本用例描述安全部门负责人管理黑白名单一系列操作，包括根据条件进行筛选 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“黑白名单库-黑名单” 2. 界面上显示黑名单库中四张表的全部内容，四张表分别为uid表、IP表、手机号表、设备号表，内容包括账号、加入时间、加入原因、最后修改人、修改时间 3. 用户点击“添加”按钮，在弹窗中选择要添加的表名，填写具体内容和添加理由，点击“确定” 4. 系统更新黑名单库，并刷新界面 5. 用户点击某一黑名单右侧“修改”按钮，可对加入原因进行修改 6. 系统更新黑名单库，同时修改最后修改人、修改时间，并刷新界面 7. 用户点击某一黑名单右侧“删除”按钮 8. 系统更新黑名单库，并刷新界面   用户点击导航栏“黑白名单库-白名单”，重复流程2-8 | | |
| **扩展流程** | 4a 若用户添加黑名单中的uid表，但输入的uid不存在于用户库中，系统提示该用户不存在 | | |
| **异常流程** | 无 | | |
| **后置条件** | 无 | | |
| **补充说明** | 无 | | |

3.2.3 规则引擎构建功能

该功能主要实现规则引擎构建及调整。风控策略师建立特征库，对不同特征接收参数进行解析，定义特征计算规则；建立防控措施库，定义不同措施处置规则；构建由事件、业务线、规则、规则组组成的决策流，并实现规则、规则组执行逻辑、参数配置调整功能。

该模块用例图如图3-3所示，包括两个主要用例特征库建立，决策树建立，两个主要用例中又包含5个小用例。

下面对于该功能中的两个主要用例进行详细描述。

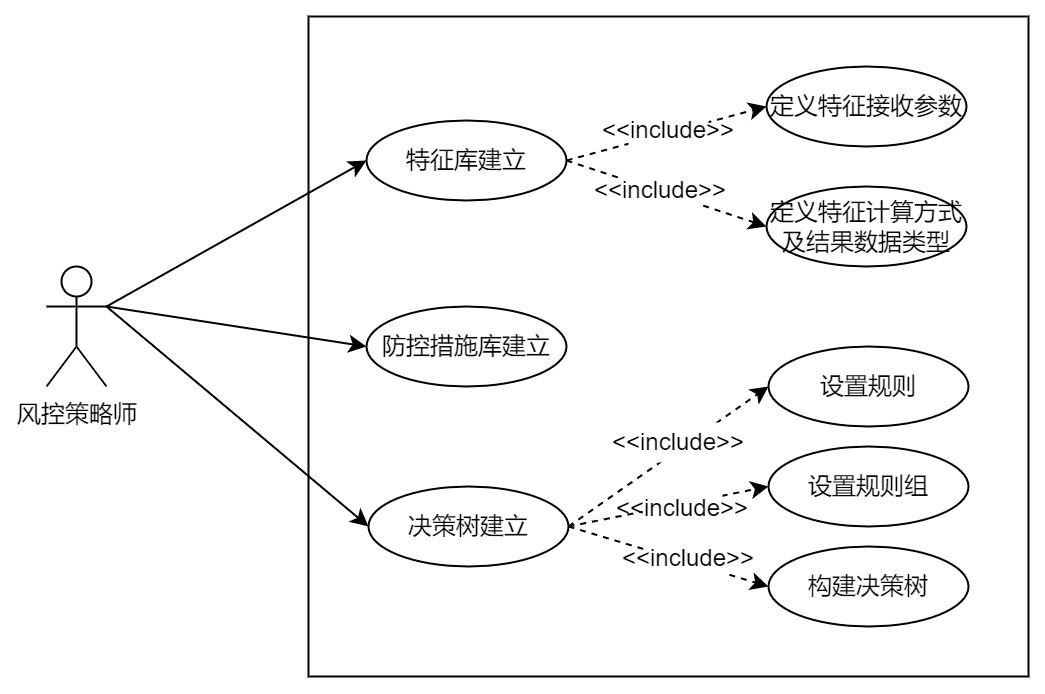


图3-3 风险情况分析模块用例图

表3-7 特征库建立用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC7 | **用例名称** | 特征库建立 |
| **活动者** | 风控策略师 | **优先级** | 高 |
| **描述** | 本用例描述风控策略师建立特征库的具体操作，包括定义特征接收参数、计算方式及结果数据类型 | | |
| **前置条件** | 无 | | |
| **基本流程** | 1. 用户在系统目录下，建立java文件featureLibrary，作为特征定义文件，后续若增加特征，直接在该文件中添加即可。 2. 定义特征函数，设定接收参数 3. 撰写特征计算方法，其中需要数据清洗的操作可使用Flink定义job完成，将结果存放至redis中。 4. 定义特征计算结果数据类型 5. 将特征名称、特征函数、结果数据类型存入数据库中，待建立规则时使用。 | | |
| **扩展流程** | 5a 更新数据库时，检查特征名称是否重复，若重复需修改featureLibrary中的特征名称。 | | |
| **异常流程** | 无 | | |
| **后置条件** | 无 | | |
| **补充说明** | 无 | | |

表3-8 决策树建立用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC8 | **用例名称** | 决策树建立 |
| **活动者** | 风控策略师 | **优先级** | 高 |
| **描述** | 本用例描述风控策略师建立决策树的一系列操作，包括规则、规则组设置及决策流构建 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“规则引擎-规则建立” 2. 跳转至规则建立界面，显示所有现有规则，包括规则名称、特征、操作符、阈值、惩罚措施 3. 点击“新增规则”，在弹窗中设置规则名称、特征、操作符、阈值、惩罚措施，点击“确认” 4. 系统更新规则库，并刷新前端界面 5. 点击对应规则后的“编辑”按钮，可编辑规则名称、特征、操作符、阈值、惩罚措施，点击“保存” 6. 系统更新规则库，并刷新前端界面 7. 用户点击导航栏“规则引擎-规则组建立” 8. 跳转至规则组建立界面，显示所有现有规则组，包括规则组名称、包含规则列表、规则组命中逻辑、惩罚措施 9. 点击“新增规则组”，在弹窗中设置规则组名称、选择组内规则，填写规则组命中逻辑及惩罚措施，点击“确认” 10. 系统更新规则库，并刷新前端界面 11. 用户点击导航栏“规则引擎-决策树建立” 12. 跳转至决策树界面，根据不同事件显示决策树，包括规则组名称、规则组组合关系 13. 点击“新增规则组”，在弹窗中设置决策树对应事件，定义规则组排列方式，点击“确认” 14. 系统更新规则库，并刷新前端界面 | | |
| **扩展流程** | 4a 新增或修改规则时，系统根据操作符和特征计算结果数据类型检查阈值设置是否合法，若非法则提示阈值非法  5a 若修改规则后未点击保存，直接点击页面其余地方，则不更新规则库，视为无改动 | | |
| **异常流程** | 无 | | |
| **后置条件** | 无 | | |
| **补充说明** | 无 | | |

3.2.4 规则执行监控功能

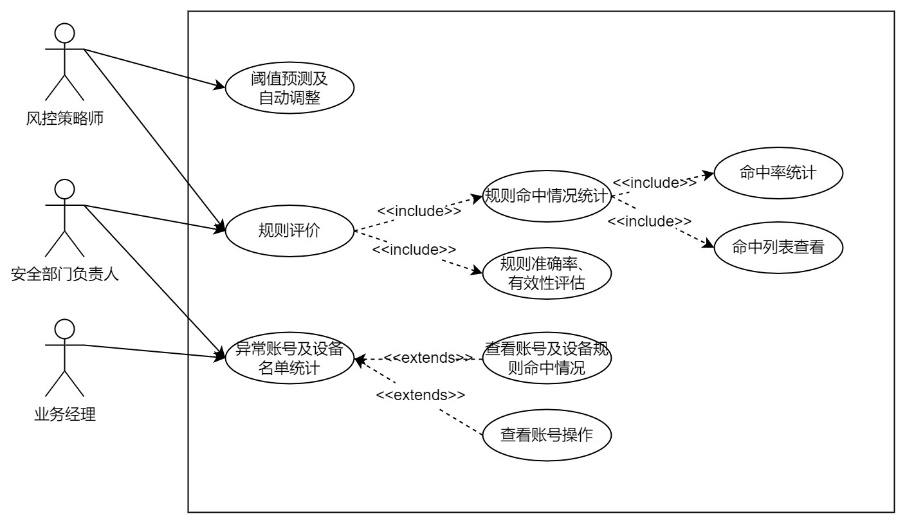
该功能主要实现规则执行监控，为安全部门负责人及风控策略师提供规则执行结果统计、评价规则准确性及合理性、规则阈值预测及自动调整功能，以及为安全部门负责人和业务经理提供异常账号及设备名单，以便进行后续操作。该功能用例图如图3-4所示。

图3-4 规则执行监控模块用例图

下面对于该功能中的三个主要用例特征库建立、决策树建立进行详细描述。

表3-9 规则评价用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC9 | **用例名称** | 规则评价 |
| **活动者** | 安全部门负责人、风控策略师 | **优先级** | 高 |
| **描述** | 本用例描述风控策略师和安全部门负责人对规则进行评价的具体操作，包括查看规则命中情况，以及对规则准确率和有效性进行评估。 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“规则评价-规则命中情况统计” 2. 系统跳转至规则命中情况统计界面，系统左侧菜单显示全部事件 3. 用户从菜单中选择事件 4. 系统界面上利用饼状图显示该事件中所有规则组的命中情况，每个饼状图中显示该规则组内所有规则的命中次数和命中率，以及规则组整体命中率 5. 用户点击界面上的任一规则组或任意规则 6. 系统界面上显示该规则组或该规则的全部命中列表 7. 用户点击导航栏“规则评价-有效性评估” 8. 系统跳转至规则有效性评估界面，系统左侧菜单显示全部事件 9. 用户从菜单中选择事件 | | |

续表3-9 规则评价用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC9 | **用例名称** | 规则评价 |
| **活动者** | 安全部门负责人、风控策略师 | **优先级** | 高 |
| **描述** | 本用例描述风控策略师和安全部门负责人对规则进行评价的具体操作，包括查看规则命中情况，以及对规则准确率和有效性进行评估。 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“规则评价-规则命中情况统计” 2. 系统跳转至规则命中情况统计界面，系统左侧菜单显示全部事件 3. 用户从菜单中选择事件 4. 界面上利用饼状图显示该事件中所有规则组的命中情况，每个饼状图中显示该规则组内所有规则的命中次数和命中率，以及规则组整体命中率 5. 用户点击界面上的任一规则组或任意规则 6. 系统界面上显示该规则组或该规则的全部命中列表 7. 用户点击导航栏“规则评价-有效性评估” 8. 系统跳转至规则有效性评估界面，系统左侧菜单显示全部事件 9. 用户从菜单中选择事件 10. 界面上利用饼状图显示该事件中所有规则及规则组的准确率和有效性 | | |
| **扩展流程** | 无 | | |
| **异常流程** | 无 | | |
| **后置条件** | 无 | | |
| **补充说明** | 规则及规则组的准确率和有效性每5分钟计算一次 | | |

表3-10 阈值预测用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC10 | **用例名称** | 阈值预测 |
| **活动者** | 风控策略师 | **优先级** | 中 |
| **描述** | 本用例描述系统根据风控策略师前几次设定的阈值，对未来一段时间内阈值自动调整的一系列操作 | | |
| **前置条件** | 风控策略师为规则设置初始阈值，并且规则部署上线超过3天 | | |
| **基本流程** | 1. 系统根据所有历史阈值，对未来24小时内阈值进行预测 2. 系统将规则阈值自动调整为预测结果 3. 系统每5分钟计算所有规则及规则组的准确率和有效性 | | |
| **扩展流程** | 3a 若在新阈值使用过程，任意规则及规则组准确率或有效性低于80%，系统向风控策略师发送告警，提示修改阈值 | | |
| **异常流程** | 无 | | |
| **后置条件** | 无 | | |
| **补充说明** | 无 | | |

表3-11 异常用户统计用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | UC11 | **用例名称** | 异常用户统计 |
| **活动者** | 业务经理、安全部门负责人 | **优先级** | 中 |
| **描述** | 本用例描述业务经理和安全部门负责人查看异常用户的一系列操作，包括查看账号及设备规则命中情况和查看账号操作 | | |
| **前置条件** | 用户进入系统 | | |
| **基本流程** | 1. 用户点击导航栏“规则评价-异常用户统计” 2. 系统跳转至异常用户统计界面，显示所有异常设备及异常账号列表，包括设备/账号名称、命中规则次数 3. 用户点击账号名称，跳转至用户操作统计界面 4. 用户点击规则命中次数，跳转至规则命中列表界面 | | |
| **扩展流程** | 无 | | |
| **异常流程** | 无 | | |
| **后置条件** | 无 | | |
| **补充说明** | 无 | | |

3.3 非功能性需求分析

租赁业务风险控制安全中台作为企业级风控系统，除上述功能性需求外，还需满足以下非功能性需求。

（1）易用性。系统界面需对用户友好，界面简洁，布局合理，交互操作符合用户习惯，交互体验感好。

（2）灵活性。风控系统涉及业务及事件多种多样，后续可能还会有新的业务线或者事件接入，因此在系统构建过程中需要做到高内聚、低耦合，使系统易扩展。

（3）正确性。由于风控决策会对用户造成或多或少的影响，有时甚至出现阻断性防控措施，因此需保证风控策略的正确性，不可对业务造成负面影响。

（4）高性能。安全中台是介于用户和业务系统中间的一环，若处理时间过长，会影响用户操作体验。因此需将数据处理时延降低至用户可接受范围内。

3.4 本章小结

本章首先对系统用户及用户根本需求进行分析，之后对系统各功能进行了详细的需求分析，包括敏感数据审计、风险情况分析、规则引擎构建、规则执行监控四大功能，最后对系统非功能性需求进行阐述。

4 租赁业务风险控制安全中台概要设计

本章以系统整体需求为基础，对系统的概要设计进行阐述。首先给出系统整体体系架构，然后按照功能模块，依次对每个模块的设计进行相关介绍，包括敏感数据审计、风险情况分析、规则引擎构建、规则执行监控，最后对系统的数据库设计进行阐述。

4.1 系统体系结构

租赁业务风险控制安全中台的总体架构图如图4-1所示。系统前端搭建使用了Vue框架，搭配了IView组件库，包含众多UI组件，达到界面美观且减少前端开发工作量的目的。使用Echarts组件库完成前端图表绘制，不仅可定制化实现各类可视化图表，且大大简化工作量。

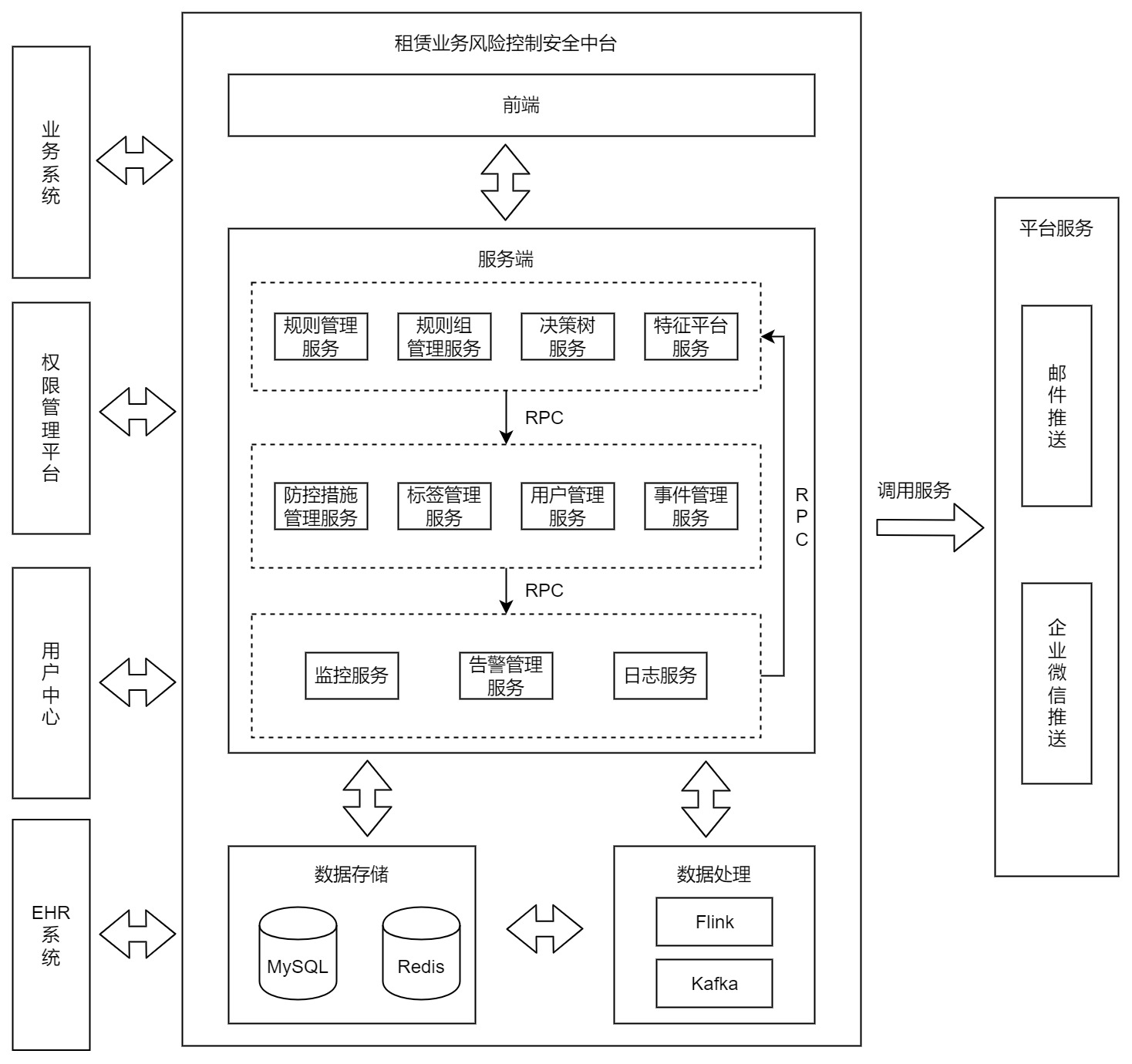


图4-1 系统架构图

系统后端采用java语言，使用SpringBoot框架简化开发工作量。系统采用微服务架构，将功能拆分为多个微服务，减少系统后端各模块之间的耦合度，易于修改、扩展和后期维护。各服务独立运行，使得系统部署更加便捷。不同服务之间的通信采用RPC。

系统存储的数据来源于三类：第一类为公司内部各系统敏感数据点击的埋点日志，通过Kafka发送至系统，经过数据清洗后存入MySQL数据库，等待敏感数据审计规则匹配；第二类为用户中心传来的用户登陆及注册数据，第三类为各业务系统传入的用户其余操作数据，这两类数据均通过Kafka发送至系统，持久化存储到MySQL数据库中，同时通过特征平台过滤，传入Flink进行数据清洗及特征聚合，将处理好的特征结果存入redis供规则计算使用。权限管理平台和EHR系统传来的数据接收即使用，不在系统内进行存储。

邮件通知和企业微信通知功能由公司基础服务平台提供，系统直接调用服务，不再进行实现。

4.2 系统功能结构

在第三章需求分析的过程中，本系统有4大模块11个用例。4大模块分别是敏感数据审计、风险情况分析、规则引擎构建、规则执行监控。系统总体功能模块结构图如图4-2所示。

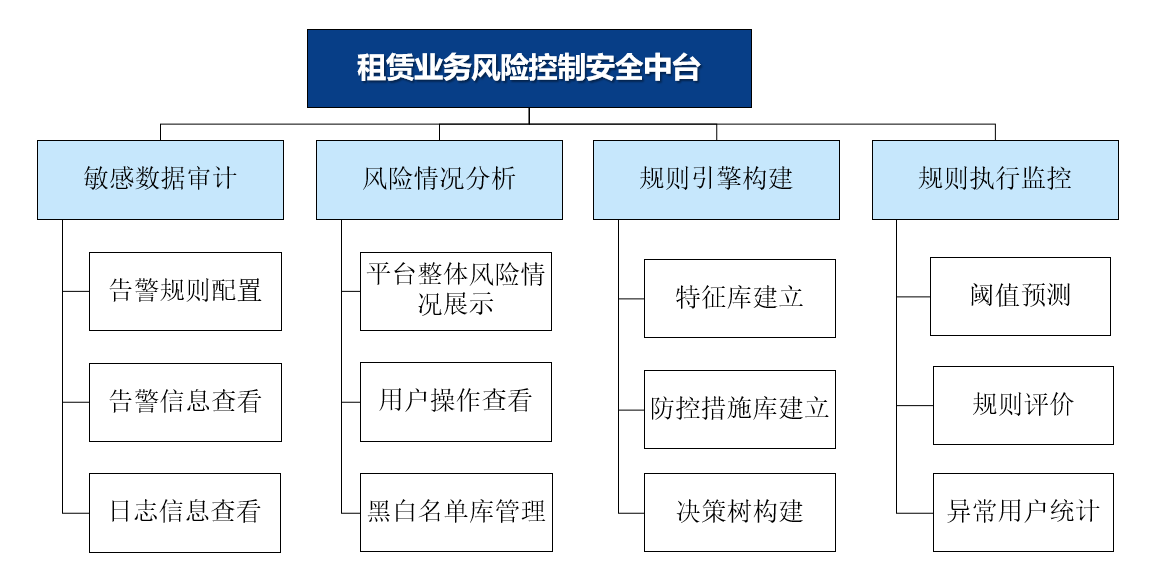


图4-2 租赁业务风险控制安全中台总体结构图

敏感数据审计模块主要是对员工点击内部系统上的敏感数据次数进行监控，每5分钟从MySQL数据库中按照规则限定的时间完成一次数据聚合，并与规则阈值进行比较。超出阈值调用HER系统数据，向对应管理层发送告警信息。

风险情况分析模块主要对平台外部用户风险进行分析展示，利用Flink框架处理用户各业务线风险分数，计算业务线加权参数得到用户在平台整体风险情况并进行展示，并提供黑白名单库管理即用户操作查看功能。

规则引擎构建模块主要实现风控规则构建及运行。风控策略师通过脚本定义特征，特征计算过程中利用Flink框架完成特征聚合，并将特征结果存入Redis。风控策略师利用定义好的特征构建规则、规则组及决策树。当有用户数据流入，该模块根据事件匹配决策树，并根据规则从Redis中取出对应特征进行计算，最后根据规则计算结果从防控措施库中匹配对应防控措施，完成对用户请求的风险判定和处理。

规则执行监控模块分为阈值预测、规则评价和异常用户统计。阈值预测中使用三次指数平滑法，利用以往规则阈值对未来24小时阈值进行预测，并自动调整。规则评价中，每5分钟根据规则命中情况，利用公式计算查准率和召回率，若低于80%发送告警。异常用户统计中，记录所有用户uid和设备号规则命中情况，并对异常用户进行统计。

4.3 系统数据库设计

本节主要介绍租赁业务风险控制安全中台的数据库设计，先给出系统的ER图，共包括16个实体和3个实体间关系，并针对每一张表给出详细设计，包括字段、类型、属性及字段描述。系统ER图如图4-3所示。

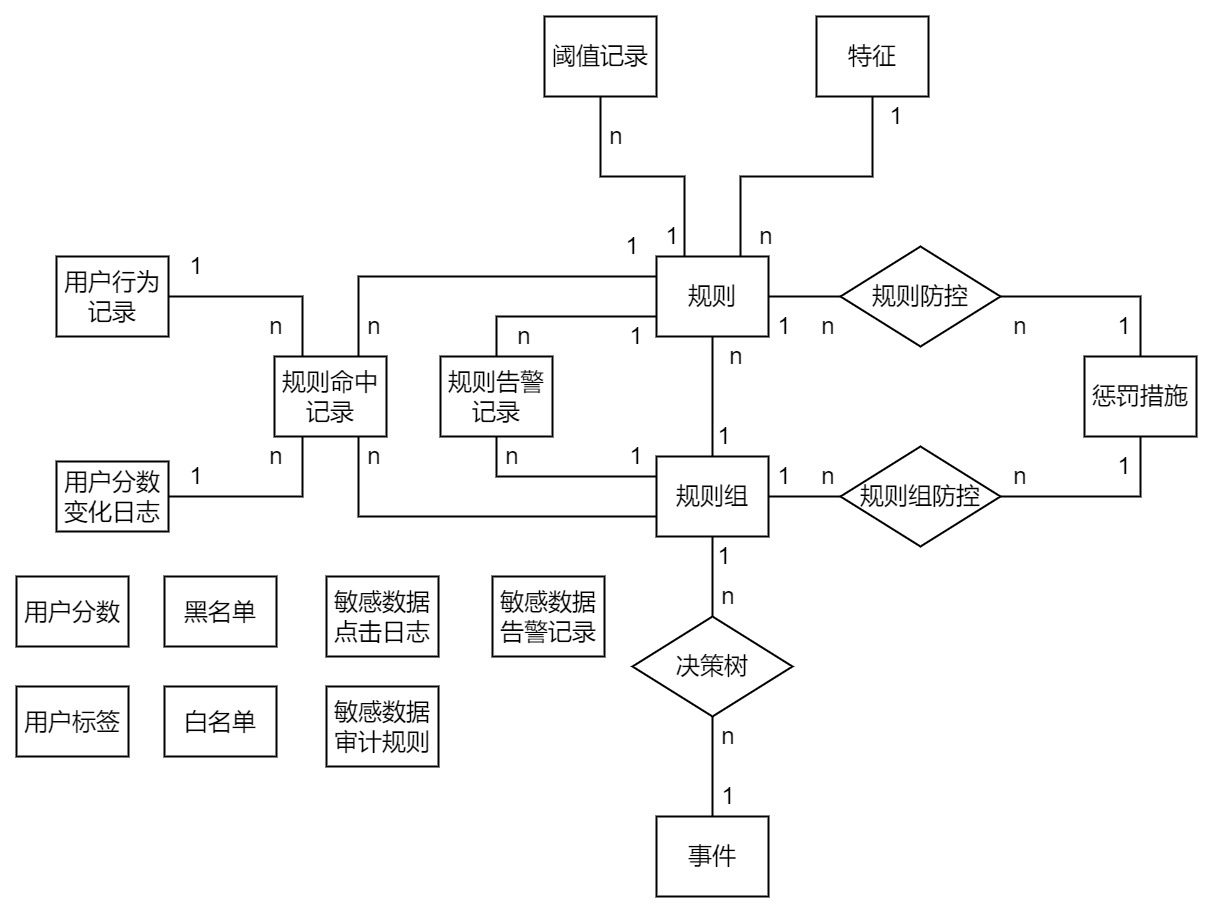


图4-3 系统ER图

表详细设计如表4-1至4-19所示。

表4-1 敏感数据点击日志表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：data\_click\_log** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | operation\_time | datetime | - | 点击时间 |
| 3 | operation\_username | varchar(50) | - | 操作人邮箱前缀 |
| 4 | operation\_uid | varchar(10) | - | 操作人工号 |
| 5 | operation\_name | varchar(10) | - | 操作人姓名 |
| 6 | operation\_field | varchar(255) | - | 操作字段名称 |
| 7 | system | varchar(50) | - | 操作系统名称 |
| 8 | operation\_status | varchar(10) | - | 操作人状态（在职/离职） |
| 9 | data\_origin | varchar(255) | - | 操作数据来源 |

表4-2 敏感数据审计规则表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：data\_click\_rules** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | role\_code | varchar(10) | - | 职位编码 |
| 3 | role\_name | varchar(50) | - | 职位名称 |
| 4 | role\_city | varchar(10) | - | 职位城市编码 |
| 5 | level1\_threshold | int | - | 一级阈值 |
| 6 | level2\_threshold | int | - | 二级阈值 |
| 7 | time\_window | int | - | 时间窗口 |
| 8 | create\_time | datetime | - | 创建时间 |
| 9 | create\_user | varchar(255) | - | 创建人 |

表4-3 敏感数据告警记录表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：data\_click\_alarm\_record** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | alarm\_level | varchar(10) | - | 告警级别 |
| 3 | operation\_username | varchar(50) | - | 操作人邮箱前缀 |
| 4 | operation\_uid | varchar(10) | - | 操作人工号 |
| 5 | operation\_name | varchar(10) | - | 操作人姓名 |
| 6 | operation\_time | int | - | 操作人查看次数 |
| 7 | leader\_uid | varchar(10) | - | 上级领导员工编号 |
| 8 | leader\_name | varchar(10) | - | 上级领导姓名 |
| 9 | alarm\_time | datetime | - | 告警时间 |

表4-4 特征表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：feture\_info** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | 特征id |
| 2 | feature\_name | varchar(50) | - | 特征名称 |
| 3 | feature\_format | varchar(10) | - | 特征计算结果格式 |
| 4 | feature\_field | varchar(255) | - | 参与特征计算的参数 |
| 5 | time\_windows | int(10) | - | 时间窗口（单位：秒） |
| 6 | description | varchar(255) | - | 特征描述 |

表4-5 规则表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：rules** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | 规则id |
| 2 | rule\_name | varchar(50) | - | 规则名称 |
| 3 | group\_id | int | FOREIGN KEY | 所属规则组id |
| 4 | feature\_id | varchar(255) | FOREIGN KEY | 参与特征计算的参数 |
| 5 | operator | varchar(4) | - | 操作符 |
| 6 | threshold | varchar(50) | - | 阈值 |
| 7 | description | varchar(255) | - | 规则描述 |

表4-6 规则阈值记录表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：rules\_threshold\_log** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | rule\_id | int | FOREIGN KEY | 规则id |
| 3 | threshold | varchar(50) | - | 阈值 |
| 4 | time | datetime | - | 时间 |
| 5 | isValid | bool | - | 该阈值是否有效 |

表4-7 规则组表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：rule\_group** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | 规则组id |
| 2 | group\_name | varchar(50) | - | 规则组名称 |
| 3 | business\_code | varchar(50) | - | 所属业务线 |
| 4 | hit\_logic | varchar(255) | - | 命中逻辑 |
| 5 | description | varchar(255) | - | 规则组描述 |

表4-8 惩罚措施表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：protection** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | 惩罚措施id |
| 2 | protection\_name | varchar(50) | - | 惩罚措施名称 |
| 3 | protection\_type | varchar(50) | - | 惩罚措施类型 |

表4-9 规则防控表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：rule\_protection** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | rule\_id | int | FOREIGN KEY | 规则id |
| 3 | protection\_id | varchar(50) | FOREIGN KEY | 惩罚措施id |

表4-10 规则组防控表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：rule\_group\_protection** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | rule\_group\_id | int | FOREIGN KEY | 规则组id |
| 3 | protection\_id | varchar(50) | FOREIGN KEY | 惩罚措施id |

表4-11 事件表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：events** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | 事件id |
| 2 | event\_name | varchar(100) | - | 事件名称 |
| 3 | enable\_status | tinyint(4) | - | 是否启用 |
| 4 | description | varchar(255) | - | 事件描述 |

表4-12 决策树表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：decision\_tree** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | group\_id | int | FOREIGN KEY | 规则组id |
| 3 | event\_id | int | FOREIGN KEY | 所属事件id |
| 4 | is\_first\_group | bool | - | 是否为该决策树第一个规则组 |
| 5 | hit\_decision | int | FOREIGN KEY | 命中待处理规则组id |
| 6 | unhit\_decision | int | FOREIGN KEY | 未命中待处理规则组id |

表4-13 用户行为记录表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：user\_behavior\_record** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | bigint | PRIMARY KEY | id |
| 2 | uid | varchar(50) | - | 用户uid |
| 3 | time | datetime | - | 用户操作时间 |
| 4 | phone | varchar(30) | - | 用户手机号 |
| 5 | imei | varchar(30) | - | 设备号 |
| 6 | ip | varchar(30) | - | 操作IP |
| 6 | event\_id | int | FOREIGN KEY | 事件id |
| 7 | business\_code | varchar(50) | - | 所属业务线 |
| 8 | sys | varchar(20) | - | 操作来源系统 |

表4-14 规则命中记录表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：rule\_hit\_record** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | uid | varchar(50) | - | 用户uid |
| 3 | time | datetime | - | 规则命中时间 |
| 4 | phone | varchar(30) | - | 用户手机号 |
| 5 | imei | varchar(30) | - | 设备号 |
| 6 | event\_id | int | FOREIGN KEY | 事件id |
| 7 | rule\_id | int | FOREIGN KEY | 命中规则id |
| 8 | rule\_group\_id | int | FOREIGN KEY | 命中规则组id |
| 9 | is\_blacklist | bool | - | 该请求是否来自黑名单 |
| 10 | is\_valid | bool | - | 是否验证通过 |
| 11 | user\_behavior\_record\_id | bigint | FOREIGN KEY | 对应用户行为id |

表4-15 用户分数变化日志表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：user\_score\_log** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | bigint | PRIMARY KEY | id |
| 2 | uid | varchar(50) | - | 用户uid |
| 3 | business\_code | varchar(50) | - | 业务线 |
| 4 | change\_score | int | - | 本次修改的分数 |
| 5 | before\_score | int | - | 修改前分数 |
| 6 | after\_score | int | - | 修改后分数 |
| 7 | change\_reason | varchar(50) | - | 修改原因 |
| 8 | business\_code | varchar(50) | - | 所属业务线 |
| 9 | rule\_hit\_record\_id | int | FOREIGN KEY | 对应规则命中记录id |

表4-16 用户分数表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：user\_score** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | bigint | PRIMARY KEY | id |
| 2 | uid | varchar(50) | - | 用户uid |
| 3 | business\_code | varchar(50) | - | 业务线 |
| 4 | current\_score | int | - | 当前分数 |

表4-17 用户标签表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：user\_tag** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | tag\_name | varchar(50) | - | 标签名称 |
| 3 | tag\_score | int | - | 标签权重 |

表4-18 黑名单表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：blacklist** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | black\_key | varchar(50) | - | 黑名单key值，可为手机号、设备号等 |
| 3 | type | varchar(20) | - | 黑名单类型（IP/phone/uid…） |
| 4 | business\_code | varchar(50) | - | 业务线 |
| 5 | add\_reason | varchar(255) | - | 加入原因 |
| 6 | add\_user | varchar(50) | - | 添加人邮箱前缀 |
| 7 | add\_time | datetime | - | 添加时间 |
| 8 | is\_del | bool | - | 是否删除 |

表4-19 白名单表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名：blacklist** | | | | |
| **序号** | **字段名** | **类型** | **属性** | **描述** |
| 1 | id | int | PRIMARY KEY | id |
| 2 | white\_key | varchar(50) | - | 白名单key值，可为手机号、设备号等 |
| 3 | type | varchar(20) | - | 白名单类型（IP/phone/uid…） |
| 4 | business\_code | varchar(50) | - | 业务线 |
| 5 | add\_reason | varchar(255) | - | 加入原因 |
| 6 | add\_user | varchar(50) | - | 添加人邮箱前缀 |
| 7 | add\_time | datetime | - | 添加时间 |
| 8 | is\_del | bool | - | 是否删除 |

4.4 本章小结

本章在第三章需求分析的基础上，介绍了系统的整体架构设计，并对系统功能模块进行划分，简要介绍了各个功能模块的实现方法。最后介绍系统的数据库设计，提供了系统整体ER图以及各个表的设计细节。

5 租赁业务风险控制安全中台详细设计与实现

本章将介绍租赁业务风险控制安全中台对的详细设计与实现，以第四章中概要设计为基础，分别针对敏感数据审计模块、风险情况分析模块、规则引擎构建模块和规则执行监控模块介绍实现的类图与时序图，并对实现效果进行展示。

5.1 敏感数据审计模块

该模块主要实现对公司内部员工查看敏感数据的监控功能。包括敏感数据审计规则管理、敏感数据审计规则执行、敏感数据审计结果展示三部分。业务流程如图5-1所示。

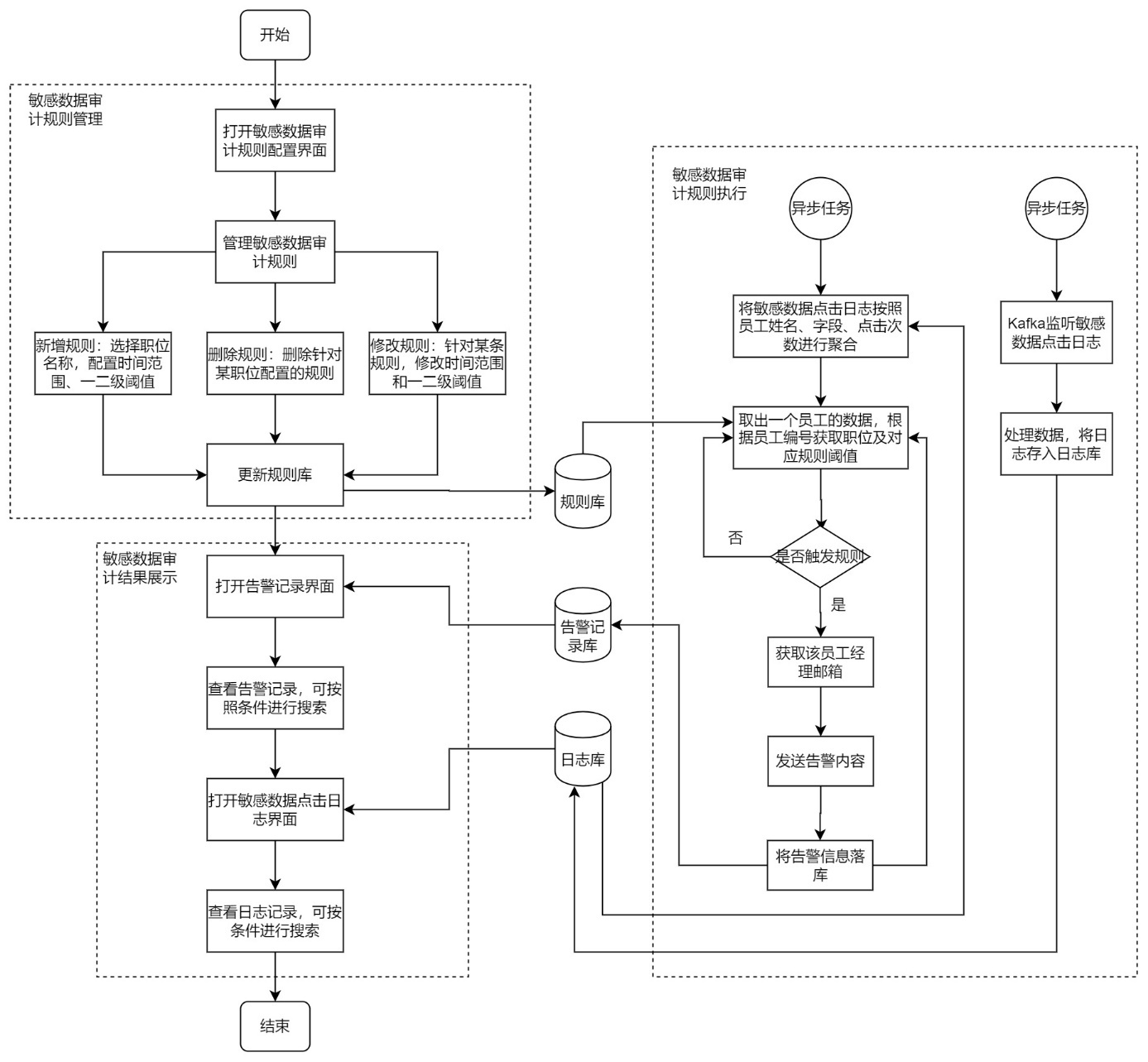


图5-1 敏感数据审计模块业务流程图

下面将分别介绍各部分的详细设计方案。

5.1.1 敏感数据审计规则管理

该部分主要实现敏感数据审计规则的添加、修改、删除、查看功能，类图如图5-2所示。该部分主要涉及6个类。dataClickRulesController类主要负责接收前端请求，dataClickRules类为敏感数据审计规则的实体类，dataClickRulesService接口接收dataClickRules实例，响应前端请求，并交由dataClickRulesImpl类进行实现，其中主要包括三个方法，完成对规则的修改和查询。dataClickRulesMapper类配合dataClickRulesMapper.xml文件，完成对数据库的操作。CityEnum类中保存规则管理过程中使用到的常量，包括城市和职位信息等。

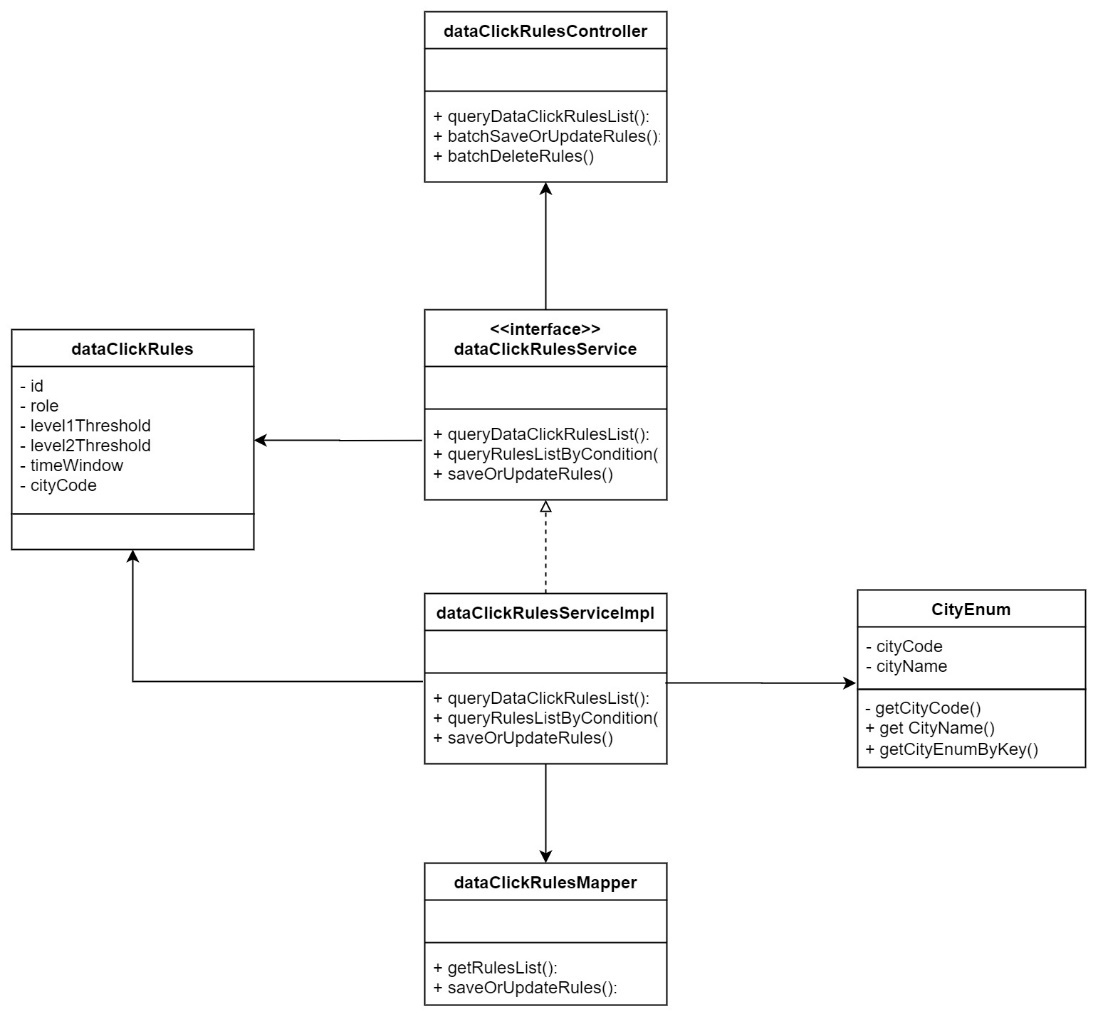


图5-2 敏感数据审计规则管理类图

下面以添加规则为例简单介绍规则管理流程，时序图如图5-3所示。dataClickRulesController类接收前端请求，当前端调用该类的queryDataClickRulesList方法发出查看规则列表请求时，controller调用dataClickRulesService的同名方法，并由dataClickRulesImpl类实现。Impl类调用dataClickRulesMapper类中的getRulesList方法，由mapper操作数据库，并将规则列表返回，由service将结果返回至controller，由controller发送至前端，在界面上进行展示。此时风控策略师通过前端界面添加规则，前端调用controller类中的batchSaveOrUpdateRules方法将请求发送至后端，controller调用service的saveOrUpdateRules方法进行处理。service先从cityEnum类中获取规则相应参数，包括城市、职位等，再调用mapper类操作数据库完成规则添加。



图5-3 敏感数据点击规则添加时序图

敏感数据审计规则管理实现效果如图5-4所示，其中部分涉密数据已遮挡。

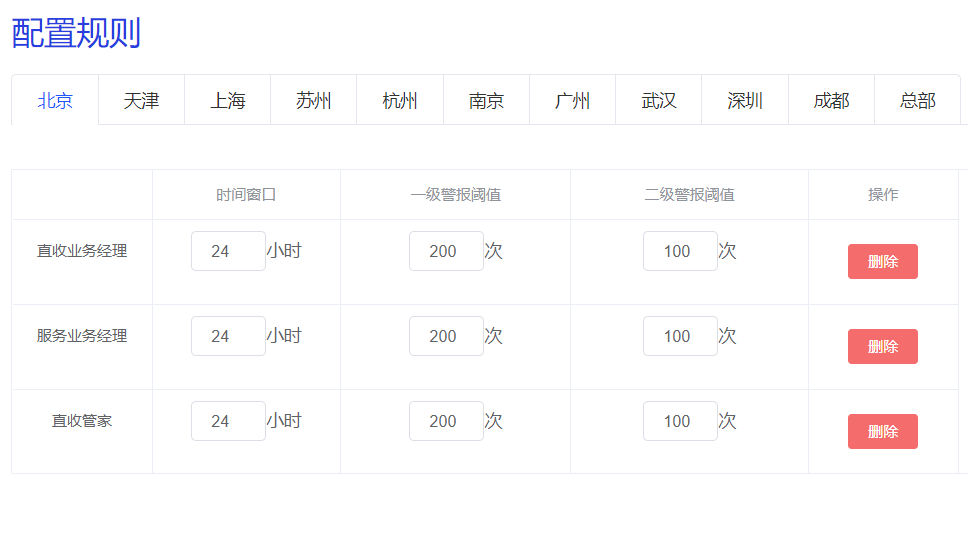


图5-4 敏感数据规则管理实现效果

5.1.2 敏感数据规则执行及结果展示

该部分主要实现执行已经配置好的敏感数据审计规则及发送告警功能，主要包括14个类，类图如图5-5所示。其中Service类均为接口，负责实现主要功能，Impl类为对应Service的实现类。Mapper类负责和数据库交互。

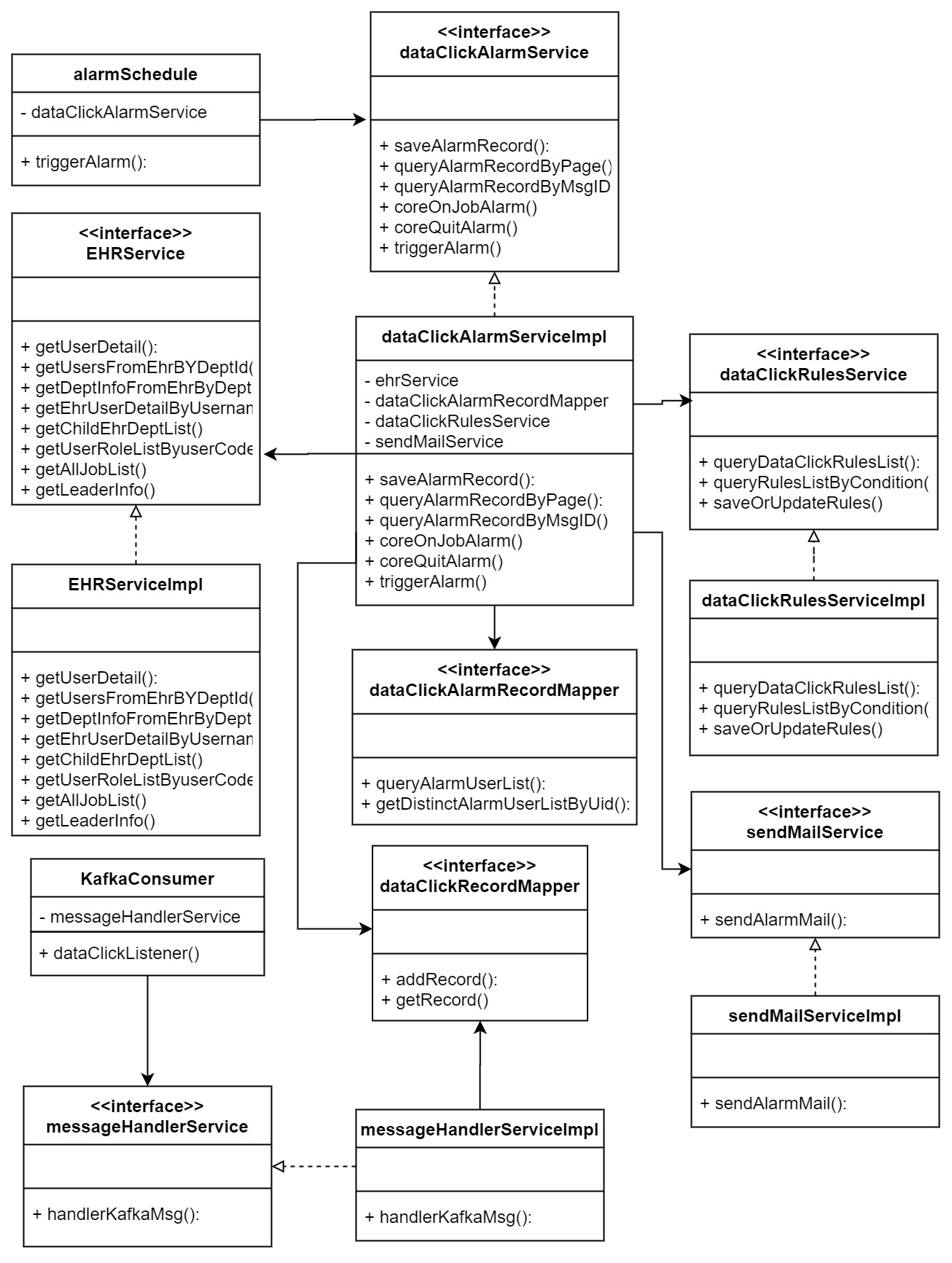


图5-5 敏感数据审计规则执行类图

其中messageHandlerService、messageHandlerServiceImpl、KafkaConsumer、dataClickRecordMapper这四个类主要完成敏感数据点击日志的接收与保存，KafkaConsumer负责监听Kafka中的日志消息，并交由messageHandlerService进行数据格式转化与保存，dataClickRecordMapper负责操作数据库，messageHandlerServiceImpl为messageHandlerService的实现类。

其余10个类负责规则执行与告警信息发布。由alarmSchedule类开启定时任务，调用triggerAlarm方法执行规则，dataClickAlarmService为规则执行核心类，负责聚合用户点击数据，并根据用户状态（在职/离职）和用户城市与职位编码与规则进行匹配。其中，用户点击数据的聚合由dataClickRecordMapper类完成，用户状态与职位可借助EHRService类调用EHR系统（公司内部人力资源系统）接口进行查询，规则数据可通过调用dataClickRulesService类中的query方法获得。若触发规则，则通过EHRService类获得员工领导及安全部门负责人信息，并通过sendMailService类完成邮件和企业微信发送，同时使用dataClickAlarmRecordMapper类更新告警记录数据库。四个Impl类为对应接口的具体实现类。

敏感数据点击规则运行及告警时序图如图5-6所示。

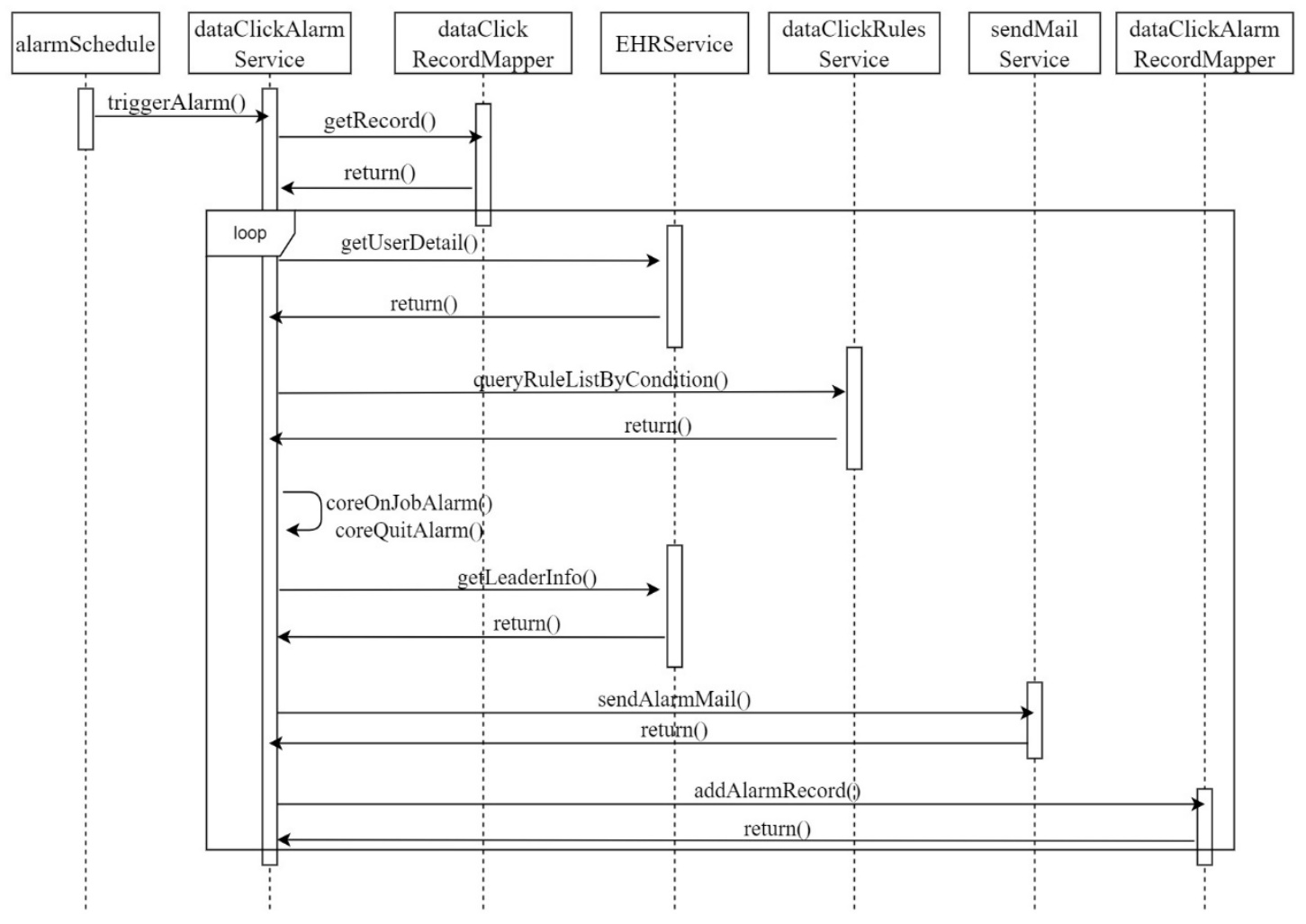


图5-6 敏感数据点击规则运行及告警时序图

其中，调用dataClickRecordMapper类中的getRecord方法返回的是一个list，里面包含每一个员工的工号以及点击次数，因此在dataClickAlarmService的triggerAlarm方法中，使用循环遍历list中的所有员工，依次判断是否触发规则。

敏感数据规则结果展示实现效果如图5-7和图5-8所示，其中部分涉密数据已遮挡。



图5-7 敏感数据点击日志查看及搜索实现效果

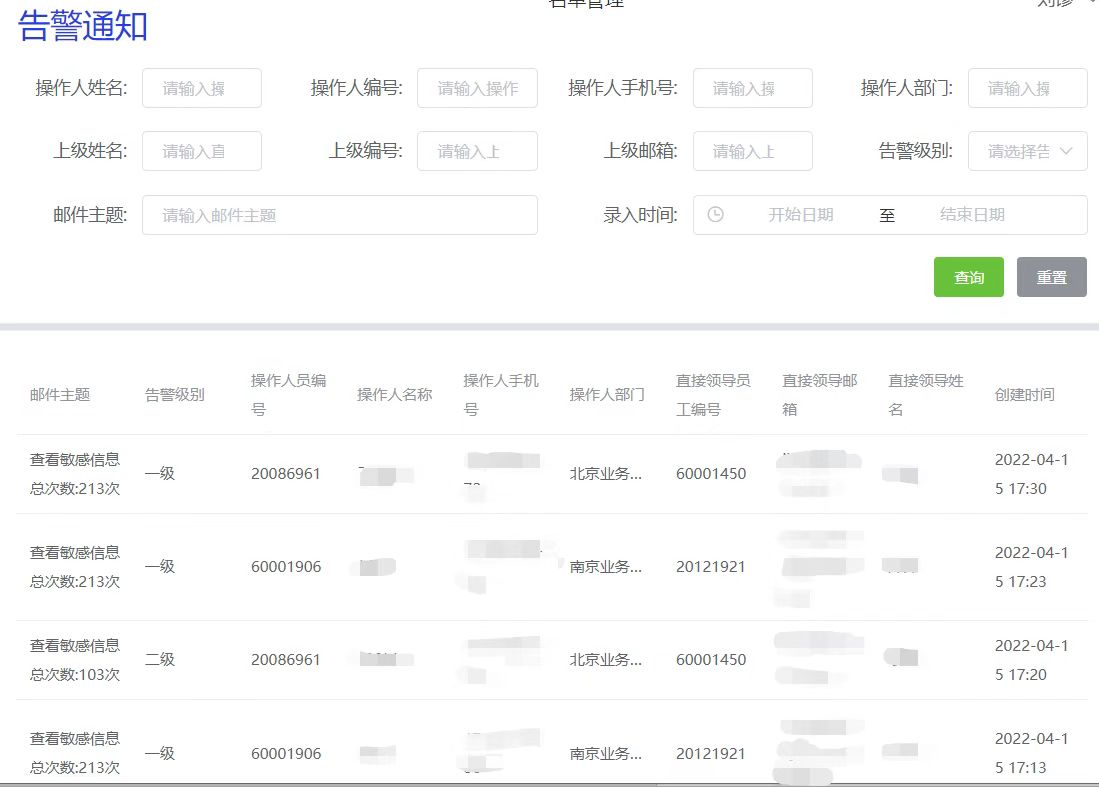


图5-8 敏感数据告警查看及搜索实现效果

5.2 风险情况分析模块

该模块主要实现平台整体风险分析及黑白名单库管理的功能，其中平台整体风险情况分析包括各业务线风险分数聚合、风险大盘展示、用户操作查看。业务流程图如图5-6所示。

平台整体风险情况分析类图如图5-9所示。共涉及11个类，其中riskScoreController类和userOperationController类接收前端请求，当前端需要获取用户分数及用户操作时，Controller类调用riskScoreService类中的方法获取数据，按照前端要求进行相应处理，如按照条件筛选、聚合等，并将结果返回至Controller类。其中，涉及到操作数据库的部分由userScoreMapper和userScoreLogMapper两个类进行处理，涉及到的数据传输借助UserScoreDto和userScoreLogDto完成。

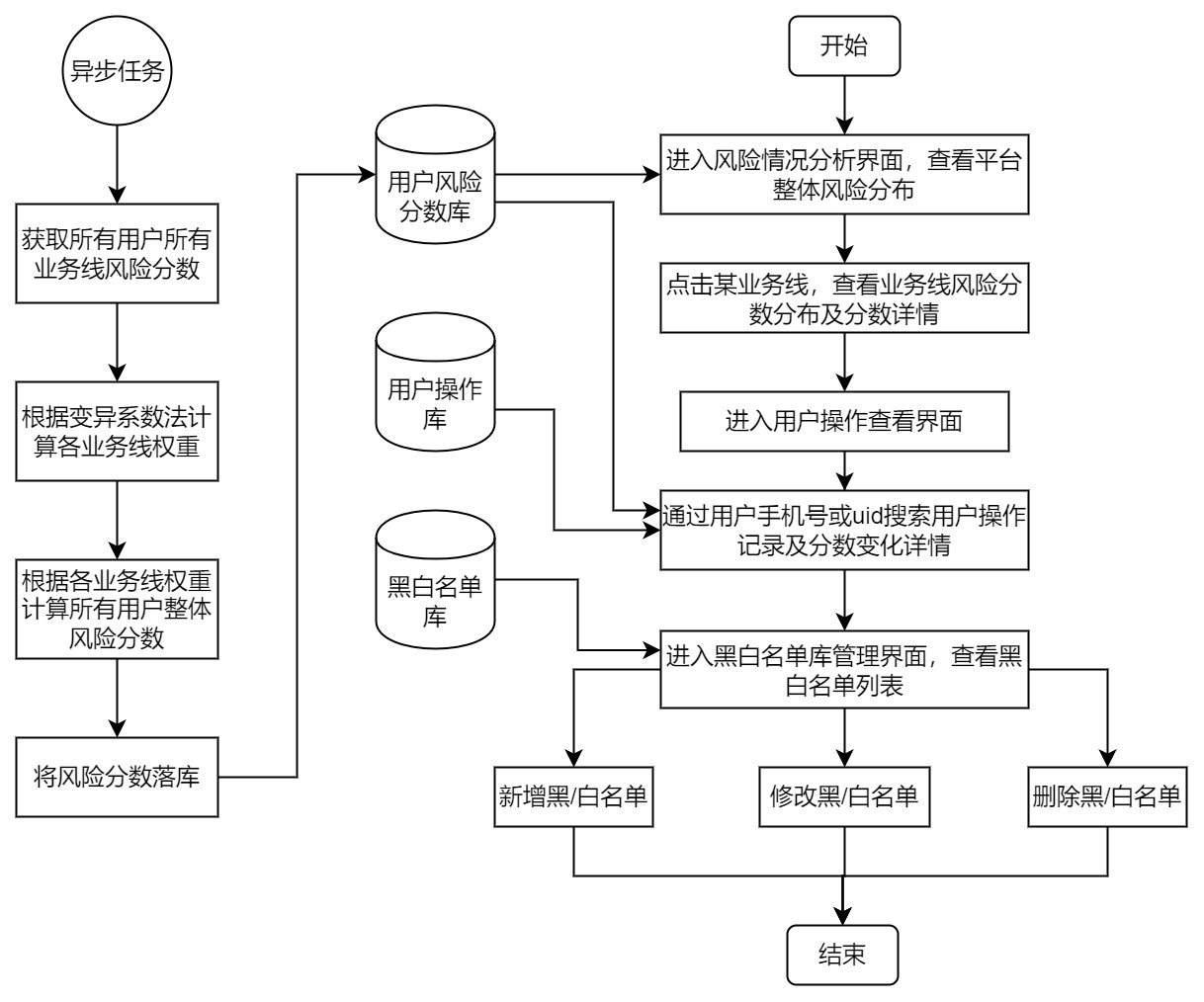


图5-9 平台整体风险情况分析业务流程图

下面以“查看某一业务线所有用户风险分数详情”这一场景简单介绍该模块的处理流程。时序图如5-10所示。riskScoreController类收到前端请求，调用自身getRiskDetailByBusiness函数，该函数内调用riskScoreService类中的同名函数，利用userScoreMapper访问数据库，通过getUserScoreByBusiness函数，从数据库中查找该业务线所有用户风险分数详情。riskScoreService类将结果表示为UserScoreDto组成的list返回给controller类，由controller类返回给前端。

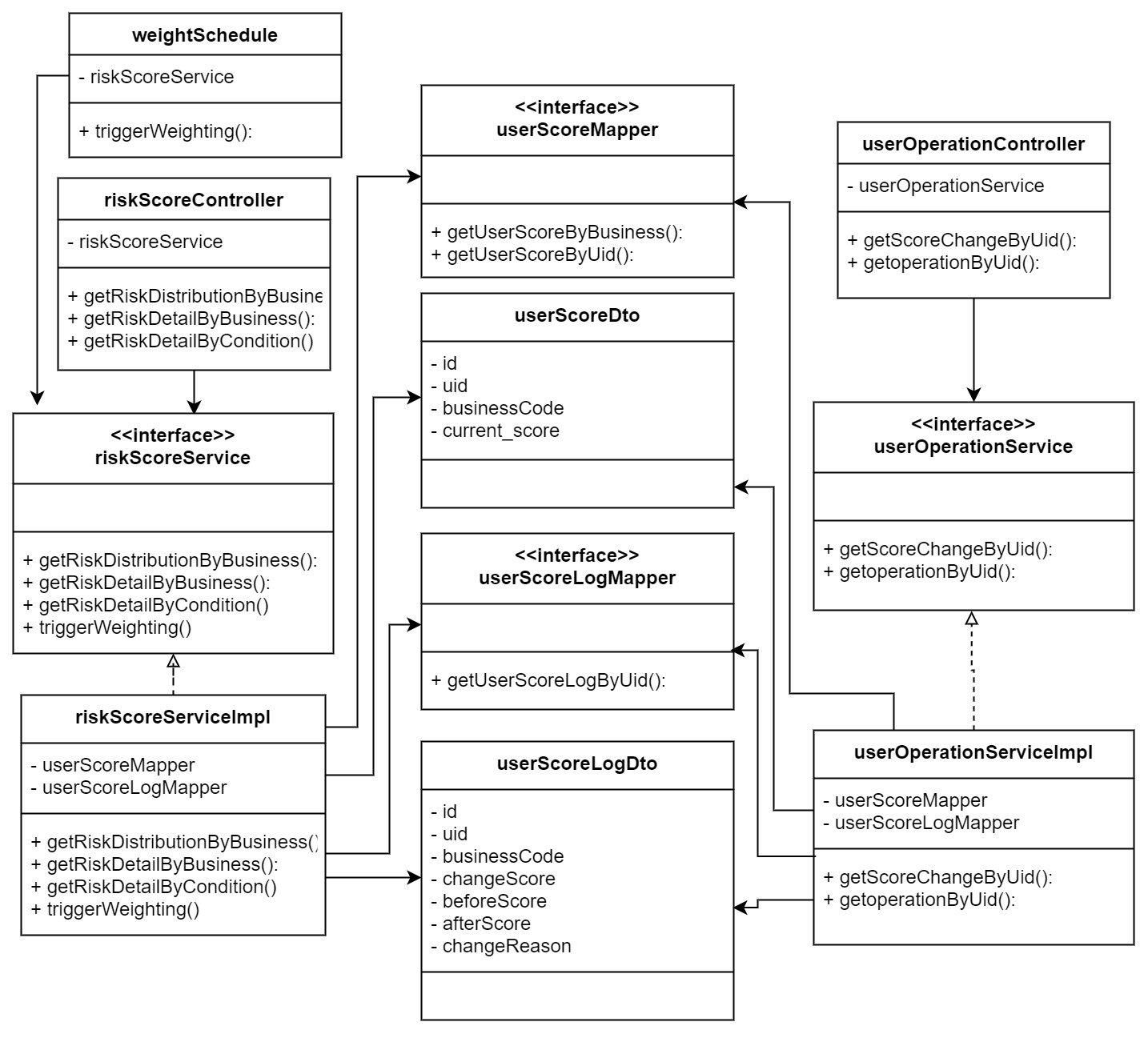


图5-10 平台整体风险情况分析类图

用户各业务线风险分数是通过规则执行情况自动计算实时更新，但用户在平台整体风险分数是利用各业务线分数，通过加权平均计算求得。且各业务线加权系数是利用变异系数法，通过全部用户全部业务线分数情况计算得到，该值并非固定值，每1小时更新一次。使用WeightSchedule类开启定时任务，每1小时调用一次triggerWeighting函数，利用riskScoreService类中的同名函数进行计算，并使用userScoreMapper类更新数据库。

黑白名单库管理部分类图如图5-11所示。

该部分共涉及8个类，其中BlackListController和WhiteListController类接收前端请求，交由BlackListService和WhiteListService处理，涉及数据库操作部分由blackListMapper类完成。

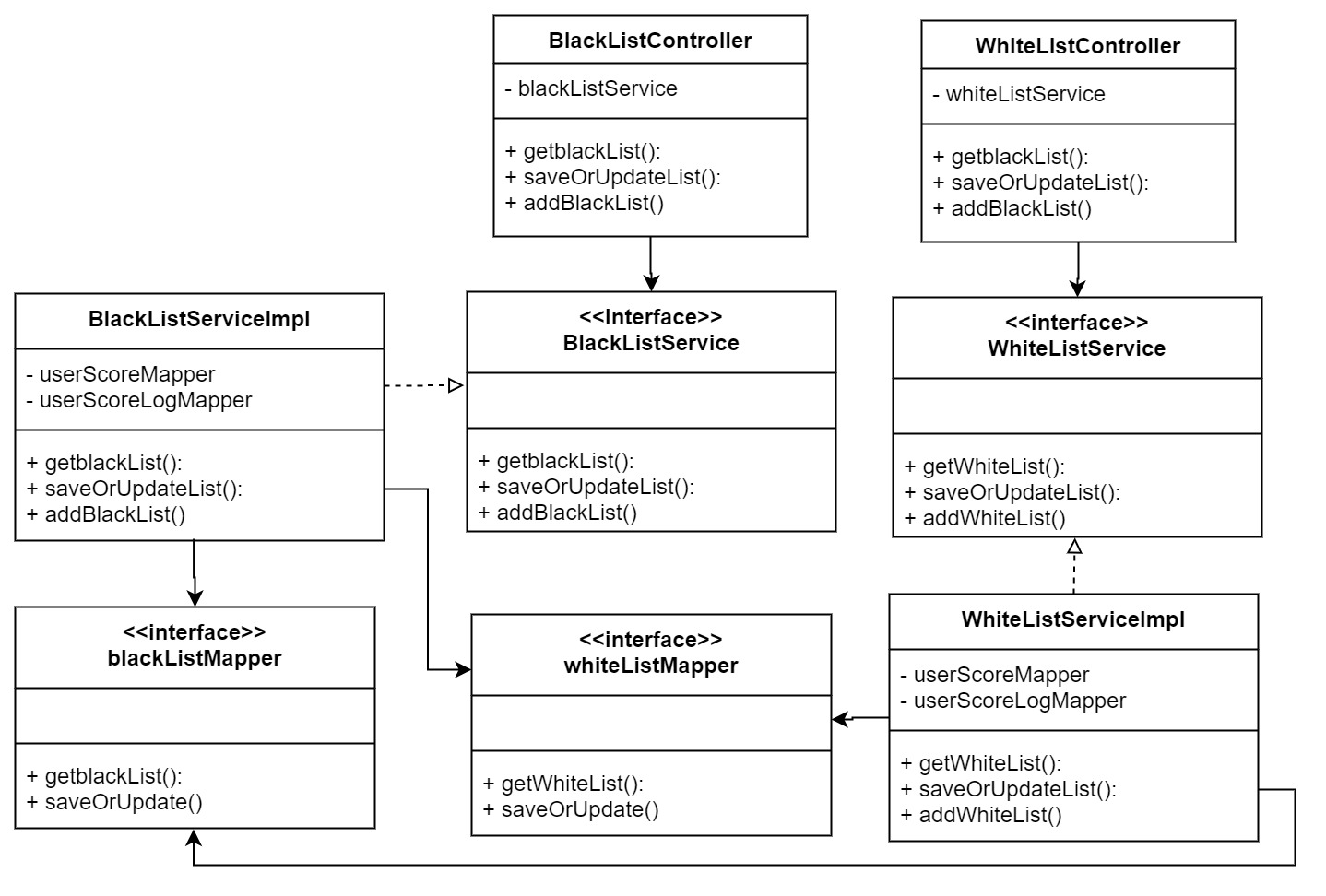


图5-11 黑白名单库管理类图

风险情况分析模块实现效果如图5-12—5-15所示，敏感数据已用测试数据代替或使用掩码遮挡。

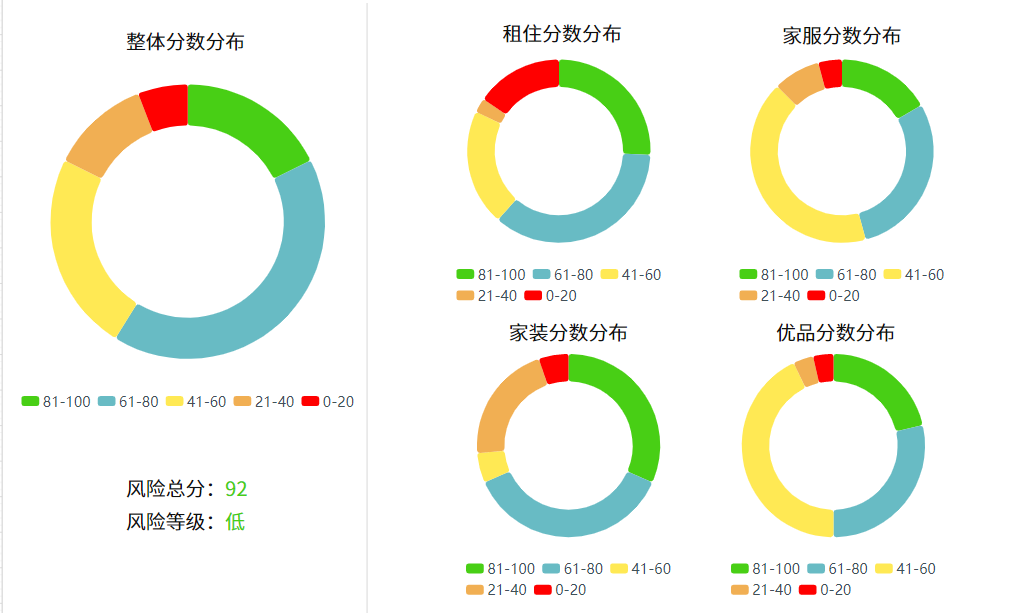


图5-12 平台整体风险情况分析实现效果



图5-13 业务线风险情况分析实现效果

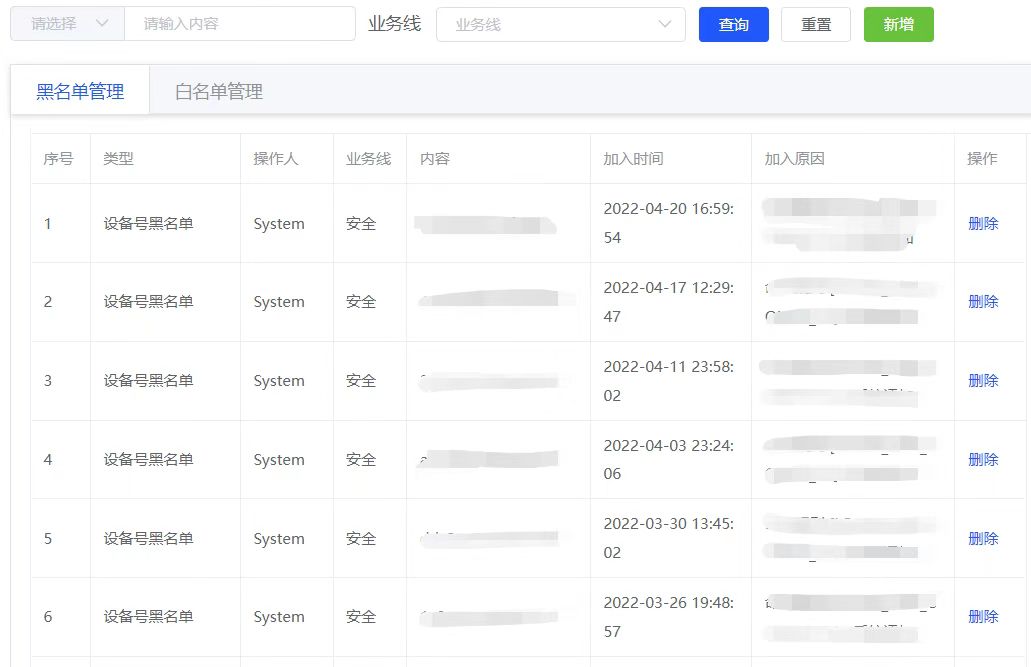


图5-14 黑白名单库管理实现效果



图5-15 用户操作查看实现效果

5.3 规则引擎构建模块

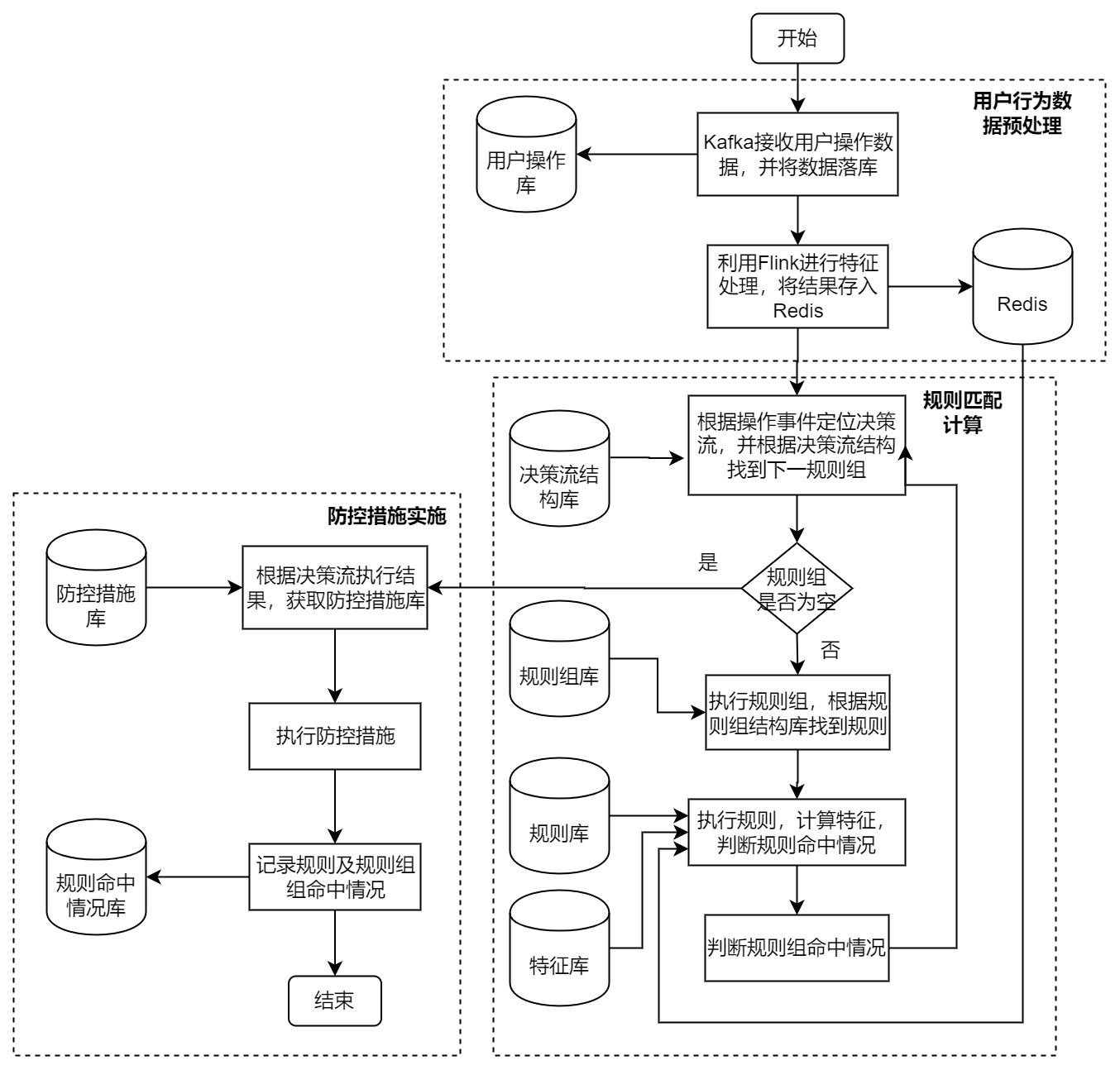
该模块业务流程图如图5-16所示。该模块主要实现规则、规则组、决策树三层规则引擎构建以及规则引擎运行功能，其中又分为3个部分：用户行为数据预处理、规则匹配计算、防控措施实施。

图5-16 规则引擎构建模块业务流程图

用户行为数据预处理部分负责接收用户行为数据，完成数据落库，并利用Flink对部分特征进行计算。规则匹配计算部分负责根据规则引擎构建结果，匹配规则进行计算，并返回结果。防控措施实施部分负责根据规则执行结果实施对应防控措施，并更新规则执行结果库。下面依次介绍这3部分的详细设计。

5.3.1 用户行为数据预处理

该部分类图如图5-17所示。该模块共涉及6个类，模块入口为KafkaConsumer类，负责监听Kafka中发送的事件，在该类中为每个事件设立一个Listener函数。此处为方便描述，仅在图中展示登陆、注册两个事件作为示例。EventHandlerService类负责对事件进行处理，实现用户操作落库，利用Flink完成对部分用户操作的处理，如计数等，并将处理结果存入Redis中。对Redis的操作由RedisUtil工具类完成，UserBehaviorRecordMapper和UserScoreInfoMapper类负责操作数据库user\_behavior\_record和user\_score表，记录用户操作数据。

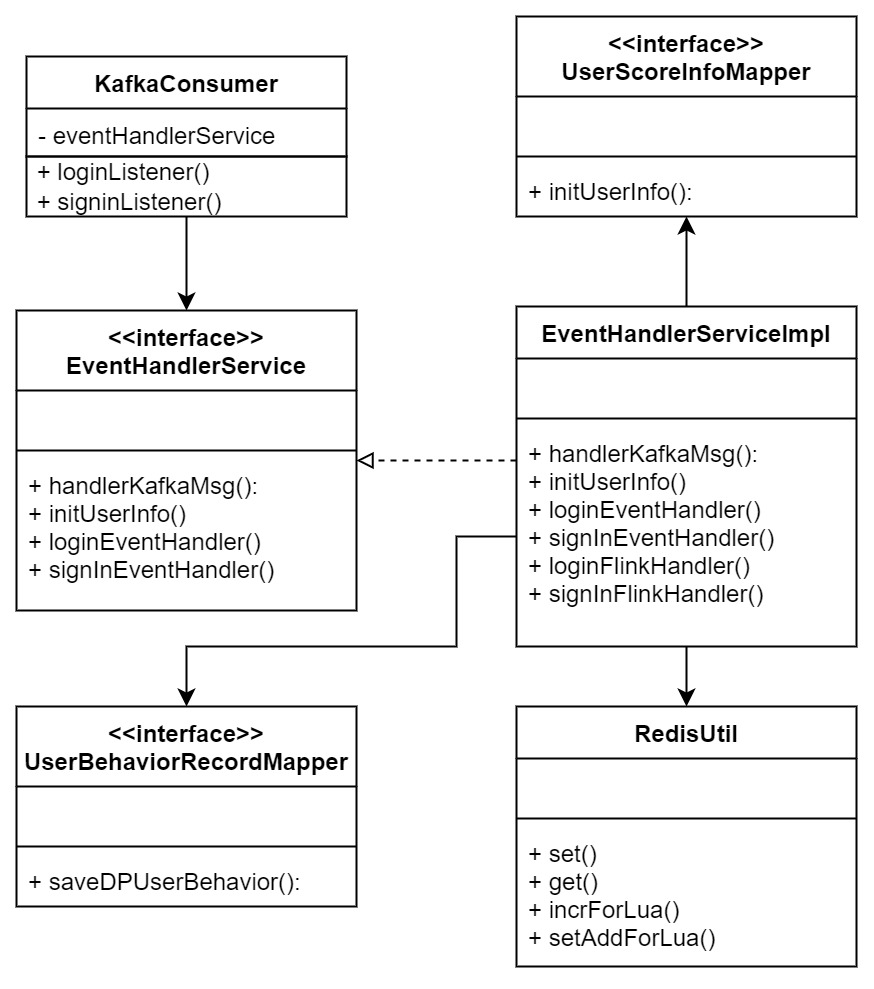


图5-17 用户行为数据预处理类图

用户行为数据预处理时序图如图5-18所示，以注册事件为例。

KafkaConsumer类通过调用自身signInListener方法，从Kafka中监听用户注册事件行为数据，当监听到事件发生后，调用EventHandlerService类中handlerKafaMsg方法，完成用户初始化和操作数据落库。EventHandlerService类调用自身initUserInfo函数，借助UserScoreInfoMapper类中的同名方法，完成对用户进行初始化操作，待Mapper类返回结果后，调用UserBehaviorRecordMapper类中的saveDPUserBehavior方法，将用户操作保存至Mysql数据库中。EventHandlerService向KafkaConsumer返回用户初始化及操作落库结果。之后，KafkaConsumer调用EventHandlerService中signInEventHandler方法，完成对用户注册这一事件中部分特征的计算。EventHandlerService调用自身signInFlinkHandler方法，利用Flink完成特征计算并将结果存储至Redis。在注册事件中，特征计算包括将该时间段内该IP、imei、phone的注册次数+1，并完成定义时间内注册次数的聚合。对Redis的操作借助RedisUtil类中set()、incrForLua()等方法实现。

其余事件实现时序图与注册事件类似，每个事件定义了不同的FlinkHandler函数，根据参数不同，对特征有不同的操作。

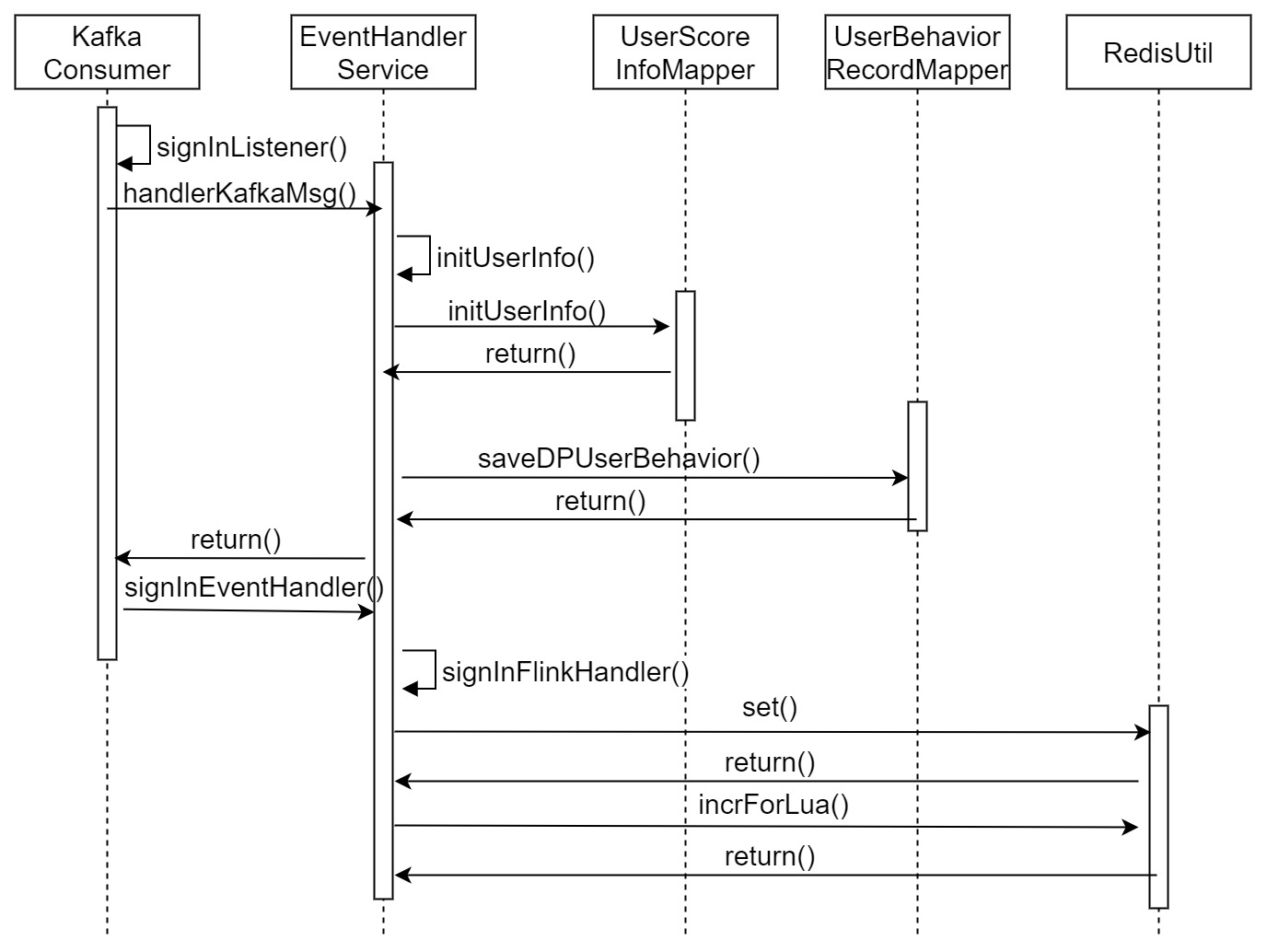


图5-18 注册事件用户行为数据预处理时序图

5.3.2 规则匹配计算

规则匹配计算类图如图5-19所示。该部分共涉及18个类，其中EventHandlerService为规则计算起始类，EngineService、RuleGroupService、RuleService、FeatureService四个类分别负责完成决策树、规则组、规则及特征的匹配与计算，FeatureMap中存储了特征名称及特征实体间的对应关系，Mapper类负责操作数据库。

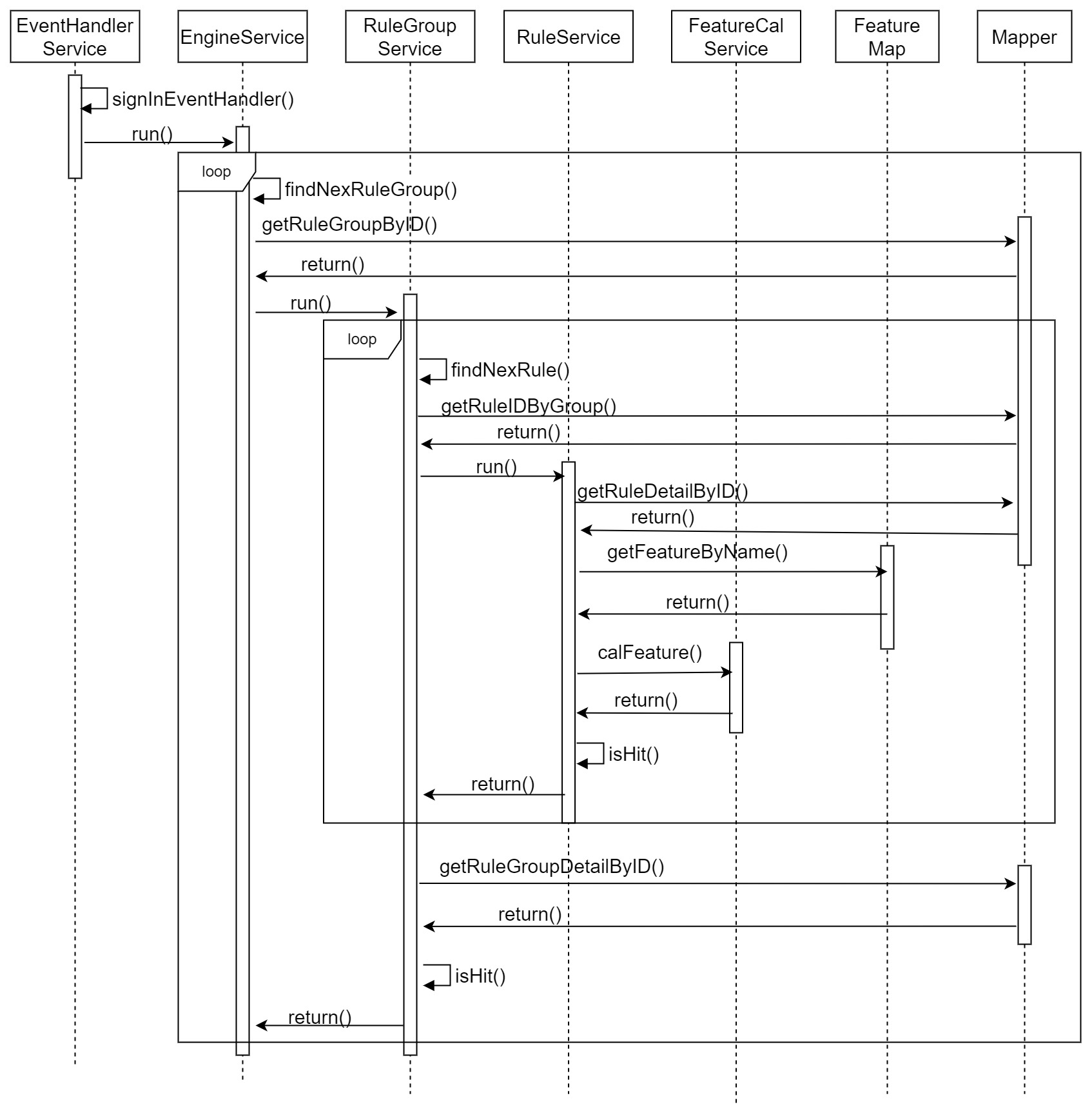


图5-19 规则匹配计算时序图

四个Service类通过run方法逐层调用计算，并逐层将结果返回，及FeatureService将特征计算结果返回给RuleService，RuleService将规则计算结果返回给RuleGroupService，RuleGroupService将规则组计算结果返回给EngineService。RuleGroup、Rule、Feature分别为规则组、规则和特征的实体类，不同特征通过继承Feature类，重写calFeature方法实现，该模块中涉及的5个Mapper类负责操作数据库中对应的表，Impl类为对应Service类接口的具体实现类。规则匹配计算时序图如图5-20所示。

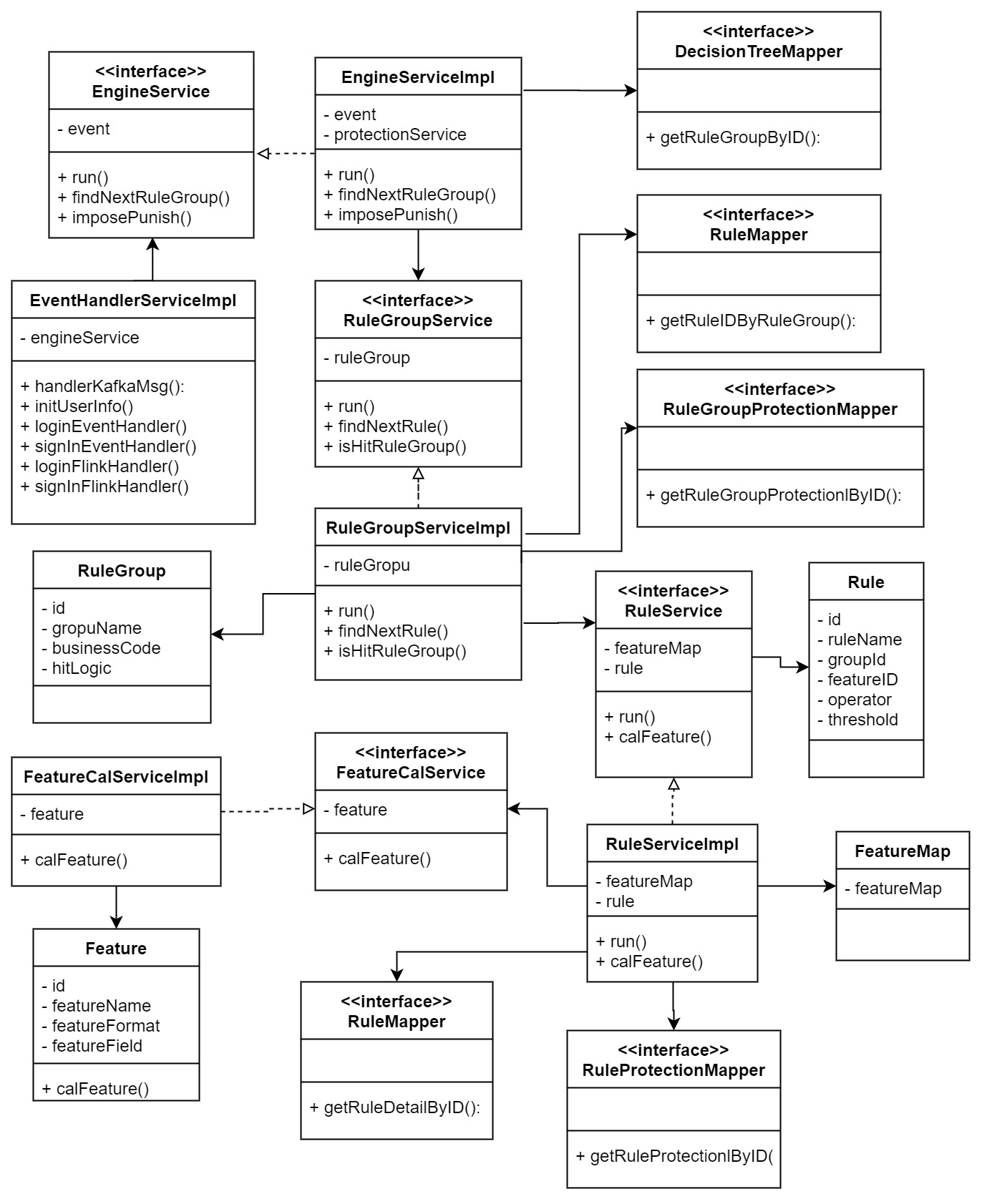


图5-20 规则匹配计算类图

5.3.3 防控措施实施

防控措施实施部分类图如图5-21所示，共涉及7个类。其中，EngineService为防控措施实施逻辑控制类，通过RuleGroupProtectionMapper类访问数据库获取防控措施详情，并使用imposePunish方法调用ProtectionService类中的exeProtection方法实施防控措施。其中Protection类为防控措施实体。最后，通过RuleHitRecordMapper类修改数据库中rule\_hit\_mapper表，将规则命中情况落库。

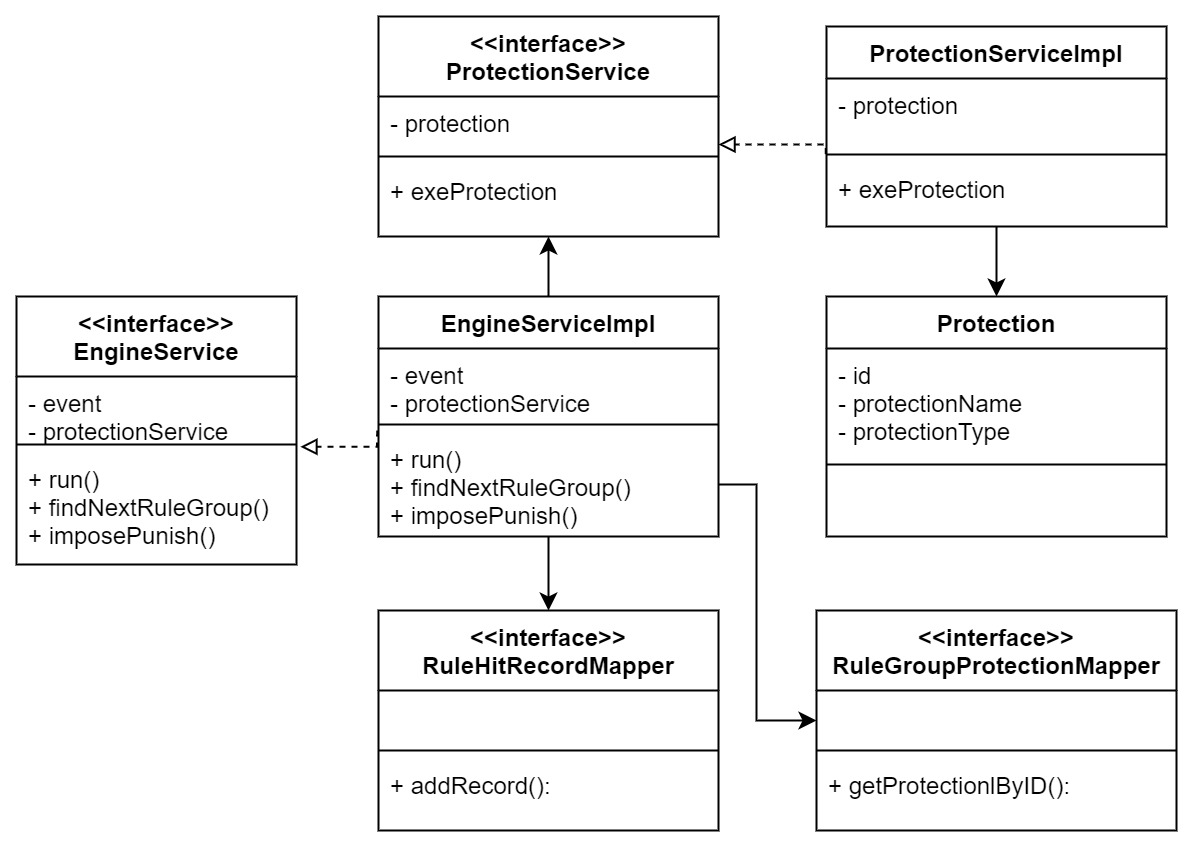


图5-21 防控措施实施类图

5.4 规则执行监控模块

该模块主要实现规则阈值预测及自动调整和规则评价，其中规则评价部分包括规则有效性分析、规则命中情况统计、异常用户名单分析功能。下面分别介绍规则阈值预测和规则评价两部分的详细设计。

5.4.1 规则阈值预测及自动调整

规则阈值预测及自动调整业务逻辑如图5-22所示。每日零点执行一次该任务，从规则历史阈值库中获取所有规则历史数据，循环遍历所有规则，将该规则历史数据传入三次指数平滑法计算函数中，得到当日阈值的预测值，并修改规则库中该规则阈值。

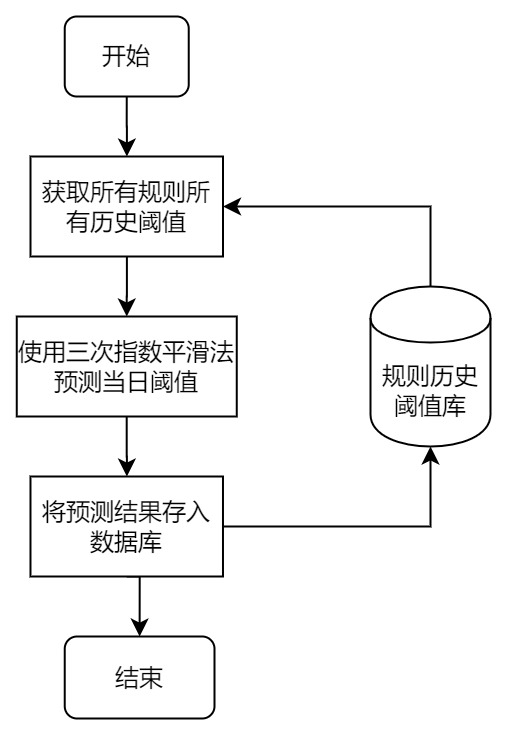


图5-22 规则阈值预测及自动调整业务流程图

类图如图5-23所示，共涉及5个类。ForecastSchedule类通过triggerForecast方法启动定时任务，每天零点进行阈值预测与调整。ThresholdForecastService类为阈值预测核心类，通过coreForecast方法，利用三次指数平滑法，以同一规则历史阈值为基础进行预测，并通过setThreshold方法将预测结果配置进数据库中。ThresholdForecastServiceImpl类为Service接口的具体实现类。涉及对数据库的操作，通过RuleThresholdLogMapper和RulesMapper两个类完成。

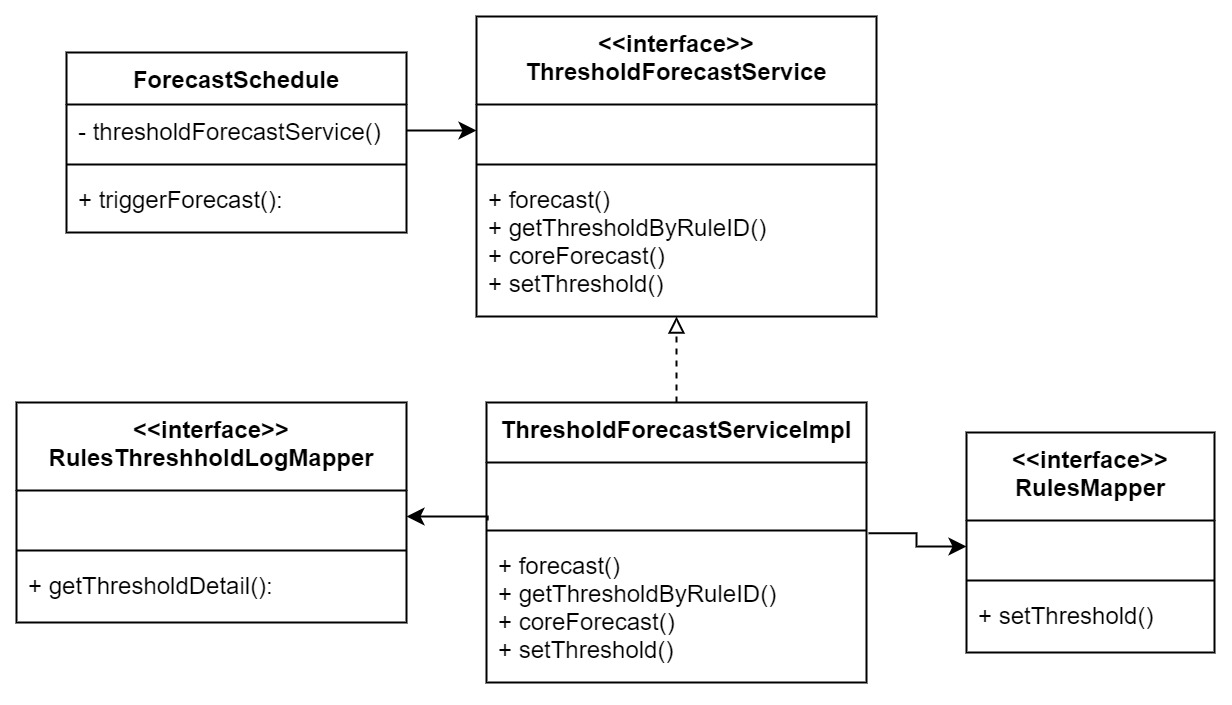


图5-23 规则阈值预测类图

5.4.2 规则评价

规则评价业务流程如图5-24所示。主任务为规则及规则组、准确性、有效性评估查看，及异常用户列表查看，包括异常账号异常设备两种异常用户。规则评价中还包含异步任务，每5分钟执行一次，从规则库中获取所有规则列表，并根据规则命中情况计算规则的准确率和有效性，若低于80%触发告警，向风控策略师发送告警信息提示修改规则。

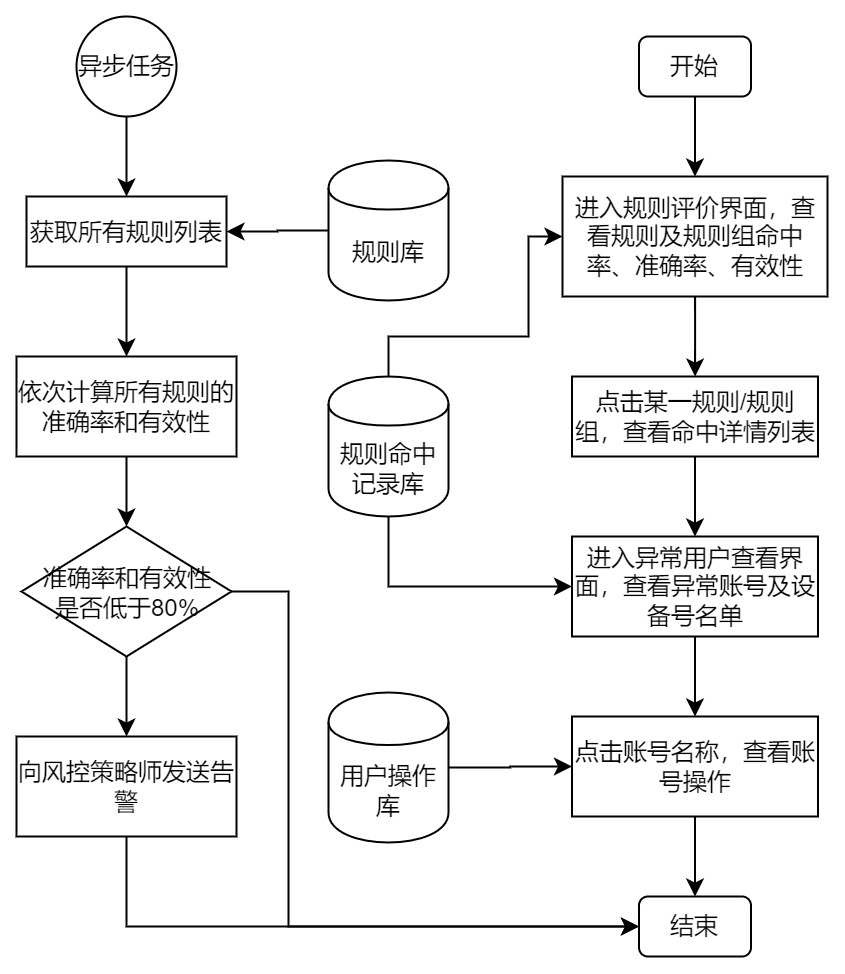


图5-24 规则评价业务流程图

规则评价部分类图如图5-25所示，共涉及10个类。RuleHitController和RuleHitService两个类负责实现规则命中率查看、具体命中列表查看以及异常用户分析展示功能。Controller类负责接收前端请求，并调用Service中的方法进行实现，RuleHitServiceImpl是Service的具体实现类。ForecastSchedule和RuleEvaluationService两个类负责规则有效性和准确性评估，由ForecastSchedule负责调度，每5分钟执行一次定时任务，由RuleEvaluationService类完成准确率和有效性的计算，若某一规则准确率和有效性低于80%，则通过sendMailService类发送告警。其中，涉及对数据库的操作由RuleHitRecordMapper完成，数据传输借助RuleHitRecordDto数据传输对象完成。

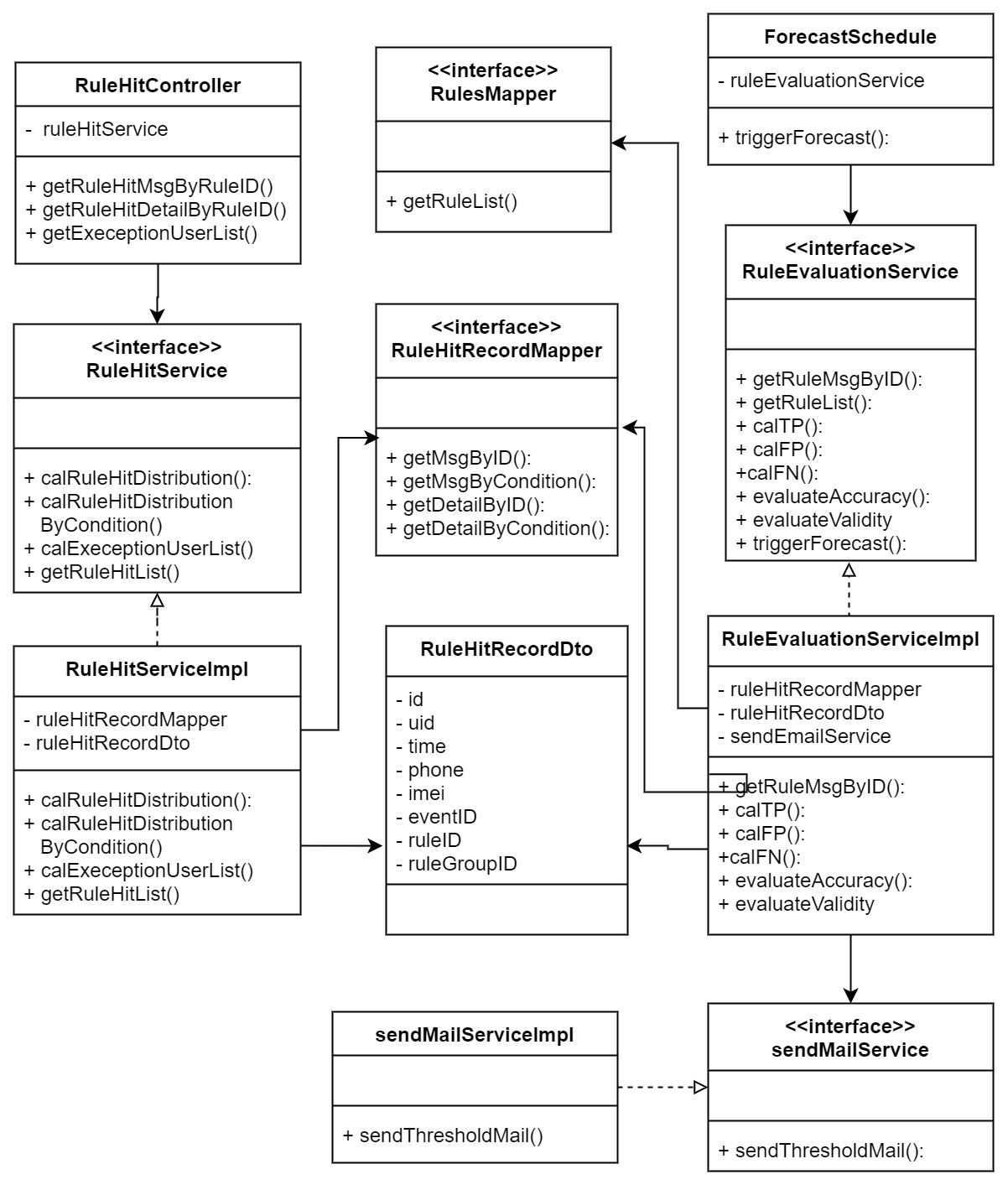


图5-25 规则评价类图

规则准确性和有效性评估部分时序图如图5-25所示。ForecastSchedule类调用RuleEvaluationService类的triggerForecast方法开启定时任务，RuleEvaluationService类调用自身getRuleList方法获得规则列表，其中访问数据库的操作，通过调用RuleMapper类中的getRuleList方法实现，循环计算规则列表中所有规则的有效性和准确性，通过RuleHitRecordMapper类访问数据库可以获得规则命中记录。若准确率和有效性低于80%，利用SendMailService类的sendThresholdMail方法发送告警邮件，其中风控策略师的邮箱通过参数传入。

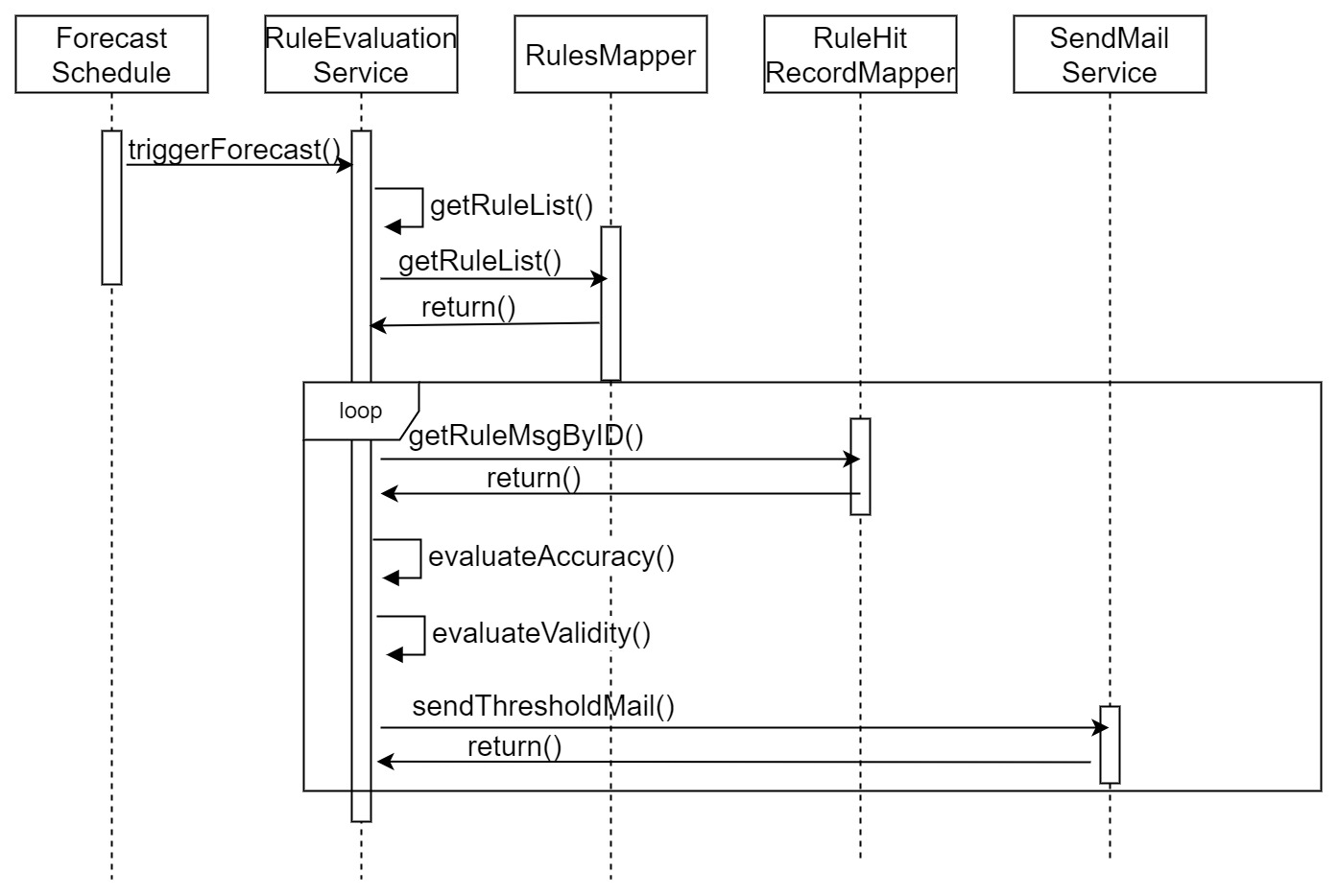


图5-26 规则准确性和有效性评估时序图

规则评价实现效果如图5-27和5-28所示。其中敏感信息已采用测试数据或使用掩码进行遮挡。图5-27展示了登陆事件两个规则组中各个规则命中情况的饼状图，以及规则组的准确率和有效性，图表采用Echart组件绘制，图5-28展示了规则命中具体列表。

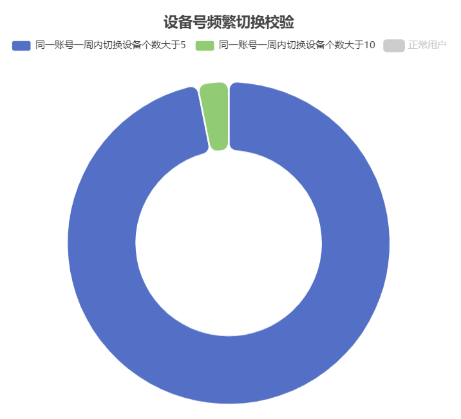
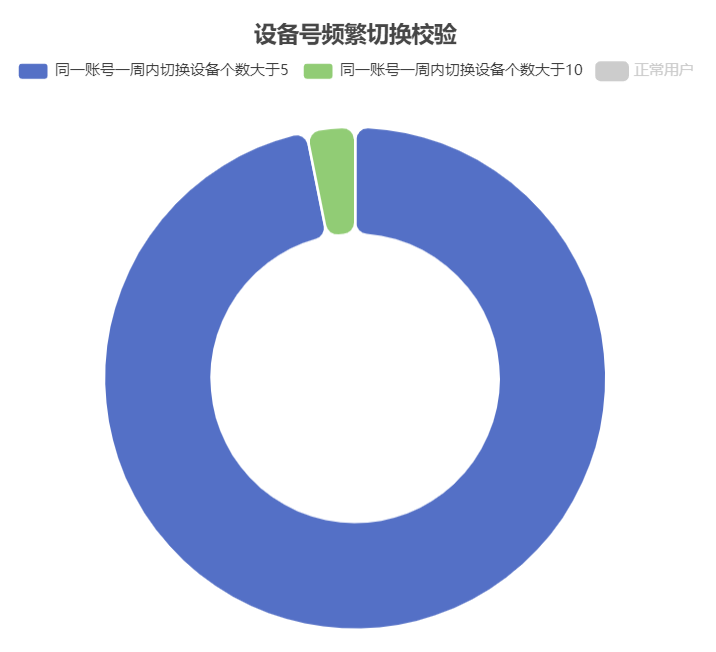
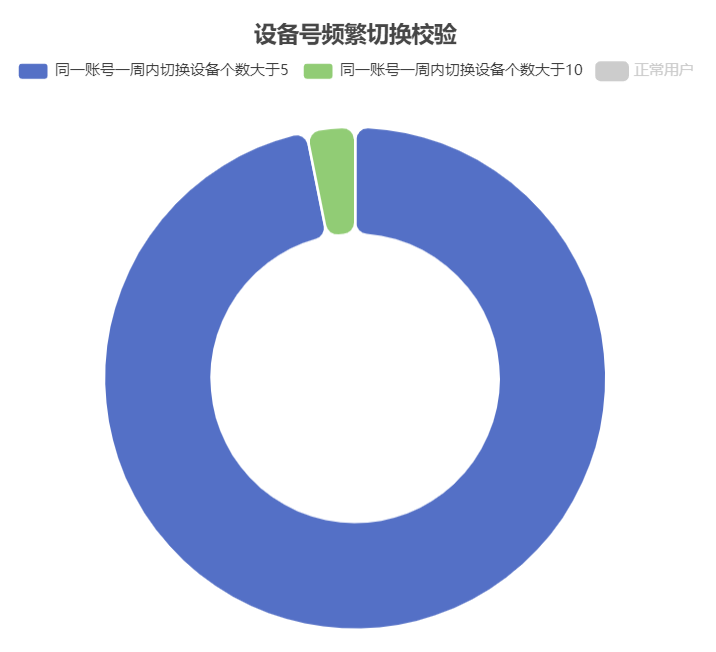
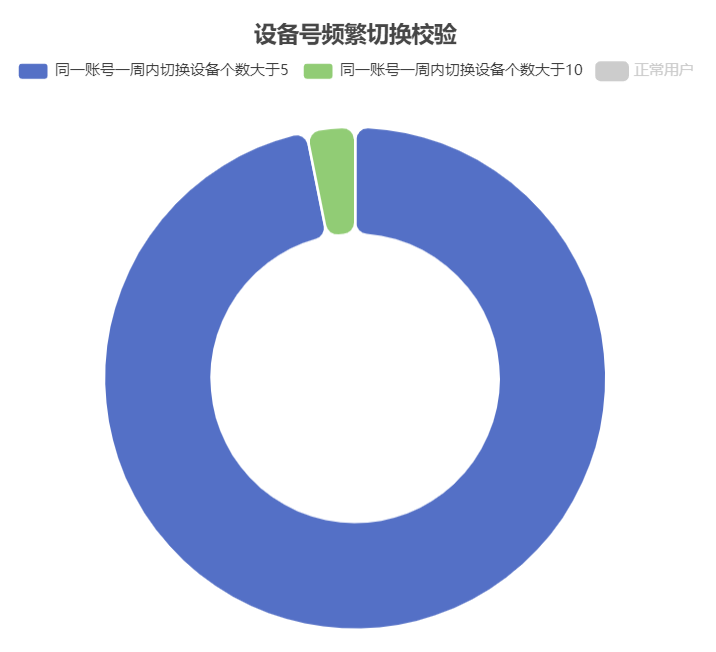
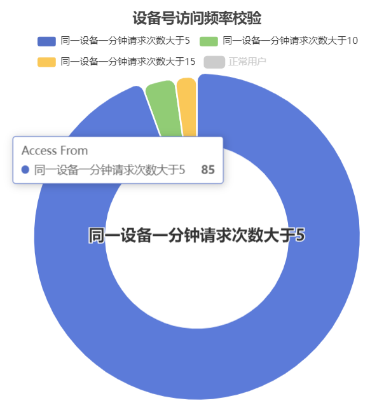
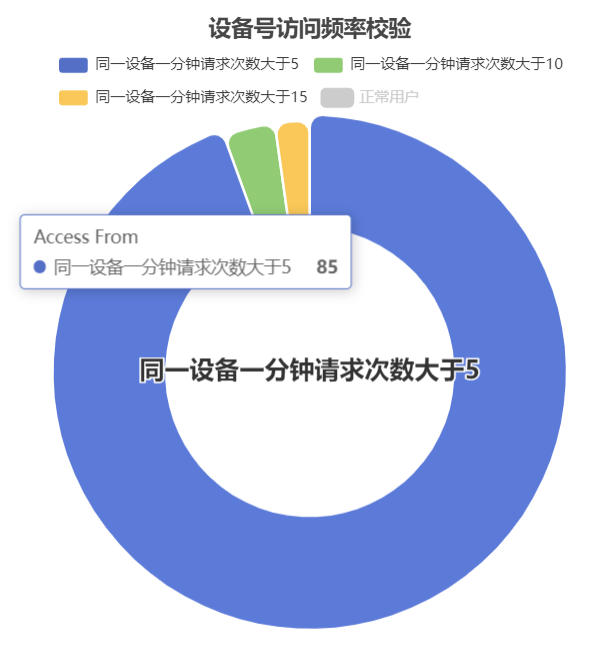
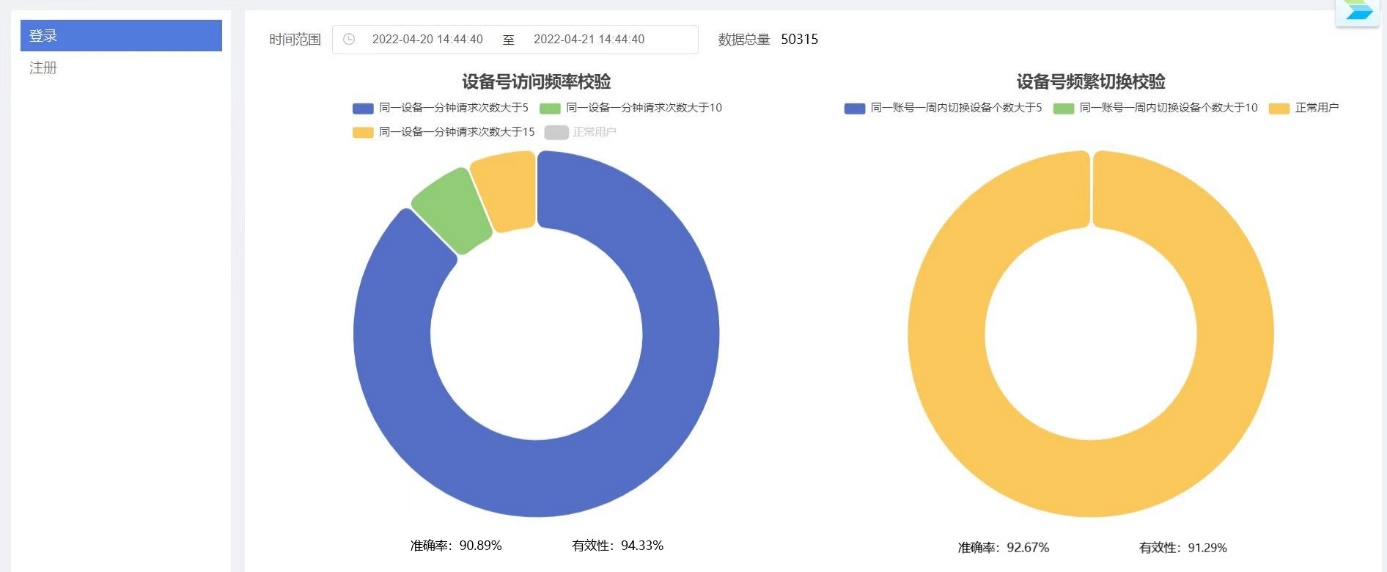
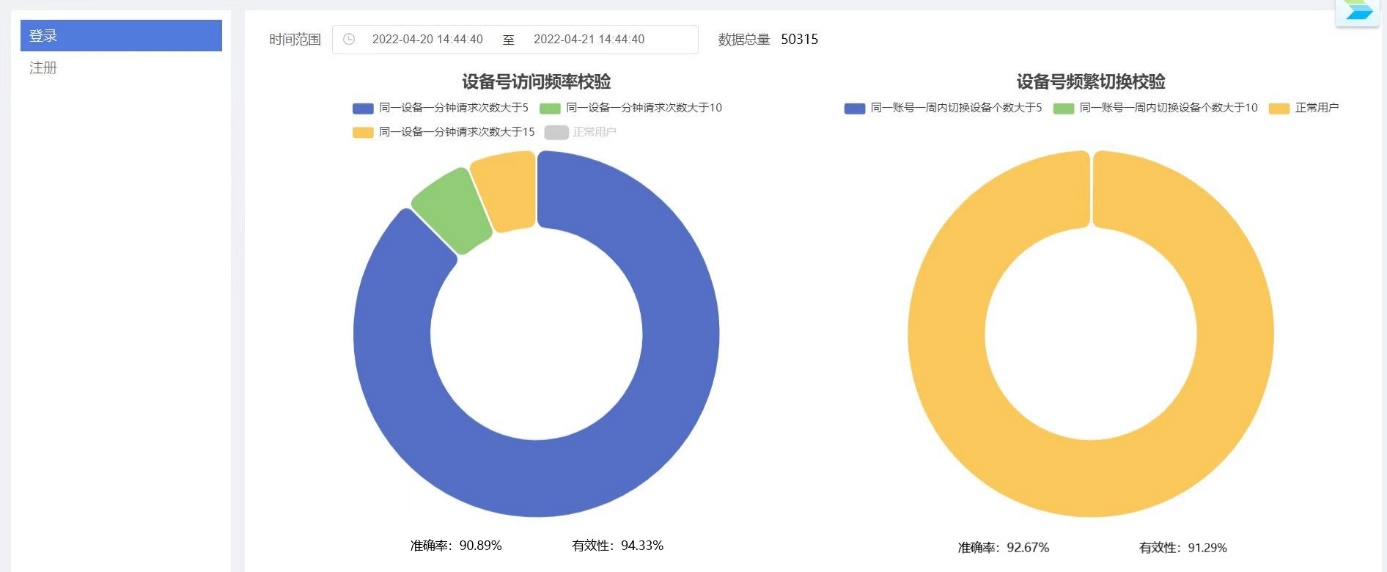
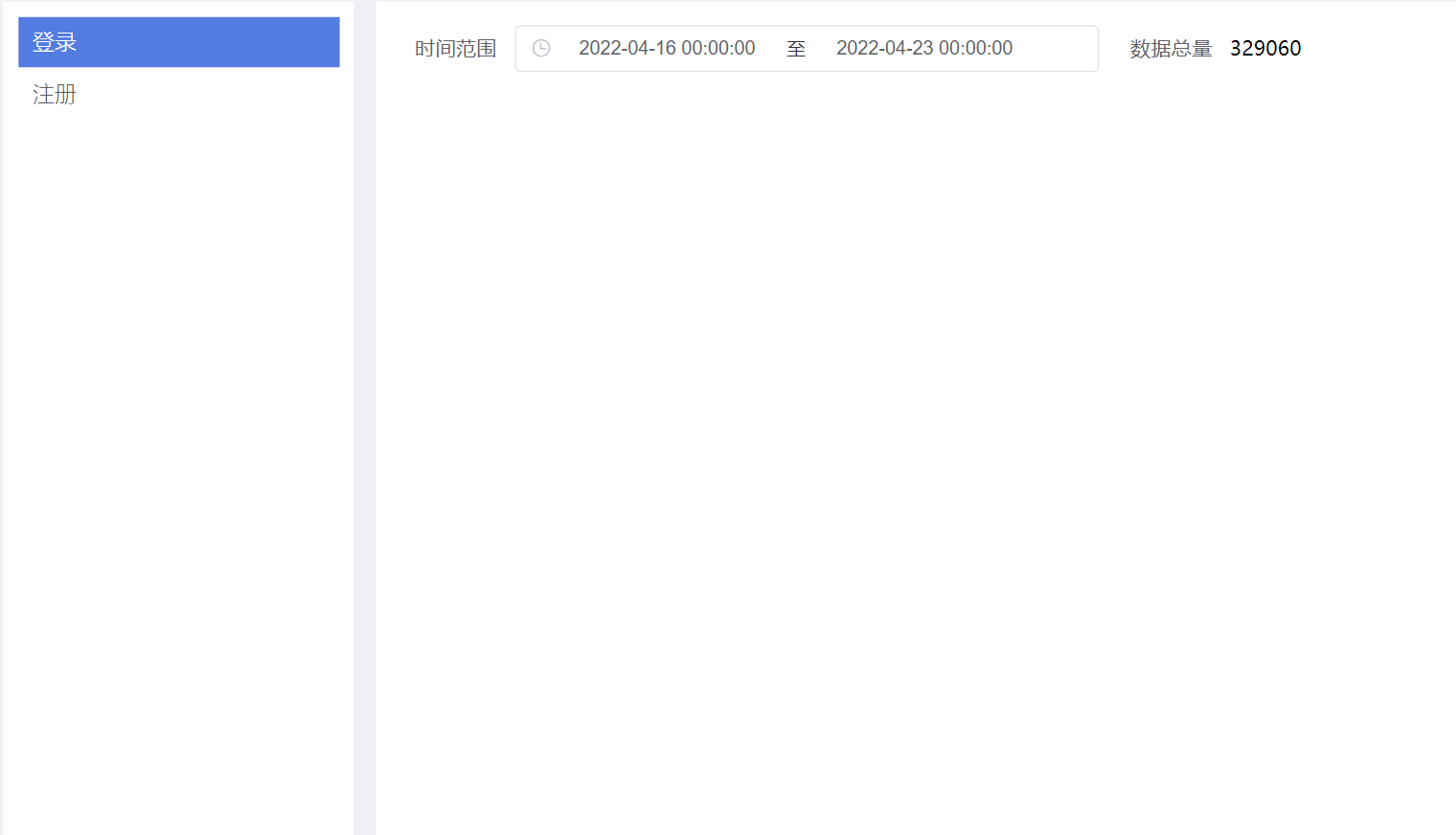


图5-27 规则命中率查看实现效果



图5-28 规则命中详情查看实现效果

5.5 本章小结

本章对系统敏感数据审计、风险情况分析、规则引擎构建和规则执行监控四大模块的具体设计与实现方法进行了阐述，先使用流程图描述了模块的具体业务逻辑，之后使用类图展示了各个类之间的关系，并使用时序图展示了功能实现的具体流程。最后展示了系统最终实现的界面效果。

6 系统测试

本章节中，以第三章需求分析为基础，对租赁业务风险控制安全中台进行功能性和非功能性。功能性测试主要针对11个用例进行，非功能性测试主要针对系统的正确性和响应时间进行测试。

6.1 测试目标及准备

本系统的测试主要分为两方面，功能测试和非功能测试。功能测试主要验证系统是否满足提出的需求，所有交互是否按照预期结果进行响应，以及响应内容是否正确。非功能测试针对系统的响应时间开展。

测试环境所需的硬件与软件版本信息如表6-1所示。

表6-1 系统测试环境

|  |  |
| --- | --- |
| 服务 | 测试环境 |
| 硬件详情 | CPU：i7 8550U，内存：8G |
| 数据库服务 | MySQL：5.7，Redis：5.0.6 |
| 软件详情 | Flink：1.9.2，Kafka：2.4.0 |
| 操作系统 | Windows 10 |

6.2 功能测试

功能测试按照第三章中用例描述划分的功能模块，使用黑盒测试的方法，按照用例描述中的使用流程进行测试，通过比较预期结果与实际结果，判断测试是否通过。

表6-2 配置告警规则测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC1 | **用例名称** | 配置告警规则 |
| **测试日期** | 2022-3-10 | **优先级** | 高 |
| **测试目的** | 是否可以查看、配置、修改告警规则 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 点击导航栏“告警-告警规则” 2. 点击下方“添加”按钮 3. 从职位名称下拉框中进行选择，并输入时间范围、一级和二级阈值，点击保存。其中，输入的时间范围不为正一位小数，一级阈值小于二级阈值 4. 修改规则，输入正确的时间和阈值 5. 点击某一规则右侧“删除”按钮 | | |

续表6-2 配置告警规则测试用例

|  |  |
| --- | --- |
| **测试流程** | 1. 修改任意规则时间范围、一级和二级阈值，点击保存按钮 |
| **预期结果** | 1. 界面上显示目前所有告警规则，包括职位名称、告警时间范围、一级告警阈值、二级告警阈值 2. 界面新增一行空规则，包含4个待输入选项：职位名称、时间范围、一级阈值、二级阈值 3. 系统提示输入错误 4. 系统提示保存成功，查看数据库规则已新增 5. 系统提示保存成功，查看数据库规则已删除 6. 系统提示保存成功，查看数据库规则已更新 |
| **实际结果** | 测试结果符合预期，测试用例通过 |

表6-3 查看全部告警信息测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC2 | **用例名称** | 查看全部告警信息 |
| **测试日期** | 2022-3-10 | **优先级** | 高 |
| **测试目的** | 违背规则时是否触发告警，并且告警内容是否正确 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 为职务编码为101071的员工配置告警规则，时间范围为1小时，一级阈值10，二级阈值5 2. 使用该员工账号登陆某系统，点击某敏感字段5次，记录点击时间、点击字段 3. 进入系统，点击“告警记录” 4. 点击记录右侧“查看详情”按钮 5. 使用该员工账号登陆某系统，再点击某敏感字段5次，记录点击时间、点击字段 | | |
| **预期结果** | 1. 系统显示保存成功 2. 5分钟后，触发二级告警 3. 界面上显示所有告警记录，告警记录具体内容包括时间、触发告警员工姓名、工号、所属部门、触发告警级别、规则时间内敏感数据点击次数显示正确 4. 系统跳转至敏感数据点击日志查看界面，显示该员工全部敏感数据点击日志 5. 5分钟后，触发一级告警 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-4 查看全部日志测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC3 | **用例名称** | 查看全部日志 |
| **测试日期** | 2022-3-10 | **优先级** | 高 |
| **测试目的** | 点击敏感数据是否发送日志，且日志是否正确 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 使用职务编码为101071的员工账号登陆某系统，点击某敏感字段 2. 进入系统，点击“告警日志” 3. 在界面上方筛选框内，根据工号进行筛选，点击“搜索” | | |
| **预期结果** | 1. 界面上显示全部日志信息，按照时间由近到远排序，显示内容包括时间、员工姓名、工号、所属部门、点击系统、字段名称、字段描述 2. 界面显示该员工的点击日志，内容正确 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-5 平台整体风险情况查看测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC4 | **用例名称** | 平台整体风险情况查看 |
| **测试日期** | 2022-3-30 | **优先级** | 高 |
| **测试目的** | 平台整体风险情况展示是否正确 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 用户点击导航栏“风险总览” 2. 用户将鼠标移动至饼状图某个区域上 3. 用户点击平台总体风险饼状图 4. 用户按照风险总分、各业务线风险分、风险级别进行筛选排序 5. 点击用户id字段 6. 用户返回风险总览界面，点击任意业务线风险饼状图 | | |
| **预期结果** | 1. 界面上利用饼状图显示平台内用户风险总分分布情况，以及各业务线用户风险分分布情况 2. 显示该分数该区域内用户总人数及占总用户人数百分比 3. 界面跳转至平台总体风险详情界面，界面上方使用折线图显示各分数段用户人数，界面下方显示全部用户分数详情，包括用户id、最后一次登陆时间、风险总分、各业务线风险分、用户标签、用户风险级别 4. 界面按照筛选条件显示对应内容 5. 界面跳转至用户操作详情界面 6. 界面跳转至该业务线风险详情界面，界面上方使用折线图显示各分数段用户人数，界面下方显示全部用户分数详情，包括用户id、最后一次登陆时间、风险总分、该业务线风险分、用户标签、用户风险级别 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-6 用户具体操作查看测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC5 | **用例名称** | 用户具体操作查看 |
| **测试日期** | 2022-3-30 | **优先级** | 高 |
| **测试目的** | 用户操作是否展示正确 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 使用平台某账号进行一系列操作 2. 点击导航栏“用户操作详情” 3. 输入账号的手机号，点击搜索 | | |
| **预期结果** | 1. 跳转至操作详情查看界面，顶端显示用户id和手机号输入框 2. 界面上显示该用户详情，上方显示近一周用户各项风险分数变化情况，下方显示用户所有操作，包括时间、触发事件名称、操作所属业务线、操作触发规则名称、操作前后分数变化详情 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-7 黑白名单库管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC6 | **用例名称** | 黑白名单库管理 |
| **测试日期** | 2022-3-30 | **优先级** | 高 |
| **测试目的** | 黑白名单库管理部分功能是否正确 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 点击导航栏“黑白名单库-黑名单” 2. 用户点击“添加”按钮，在弹窗中选择要添加的表名，填写具体内容和添加理由，点击“确定” 3. 用户点击某一黑名单右侧“修改”按钮，对加入原因进行修改 4. 用户点击某一黑名单右侧“删除”按钮 | | |
| **预期结果** | 1. 界面上显示黑名单库中四张表的全部内容，四张表分别为uid表、IP表、手机号表、设备号表，内容包括账号、加入时间、加入原因、最后修改人、修改时间 2. 界面上显示新添加黑名单，检查数据库已更新，检查添加时间、添加人是否正确。 3. 界面上更新该黑名单原因、修改时间，检查数据库已更新，且最后修改人、最后修改时间均正确， 4. 界面上删除该黑名单，检查数据库已更新 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-8 特征库建立测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC7 | **用例名称** | 特征库建立 |
| **测试日期** | 2022-2-23 | **优先级** | 高 |
| **测试目的** | 检测根据现有机制，是否可以添加特征 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 新建TestFeature类，继承Feature类 2. 设定接收参数、操作符、阈值，重载calFeature方法定义计算方式 3. 将特征名称、特征函数、结果数据类型存入数据库中 | | |
| **预期结果** | 1. 检查数据库，已新增特征 2. 新建规则，在选择特征时可使用该特征 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-9 决策树运行测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC8 | **用例名称** | 决策树运行 |
| **测试日期** | 2022-2-23 | **优先级** | 高 |
| **测试目的** | 检测是否可以配置规则，以及规则是否生效 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 用户点击导航栏“规则引擎-规则建立” 2. 点击“新增规则”，在弹窗中设置规则参数，点击“确认” 3. 用户点击导航栏“规则引擎-规则组建立” 4. 点击“新增规则组”，在弹窗中设置规则组名称、选择组内规则，填写规则组命中逻辑及惩罚措施，点击“确认” 5. 用户点击导航栏“规则引擎-决策树建立” 6. 点击“新增决策树”，在弹窗中设置决策树，点击“确认” 7. 使用某用户账号，做出违反规则操作 | | |
| **预期结果** | 1. 跳转至规则建立界面，显示所有现有规则，包括规则名称、特征、操作符、阈值、惩罚措施 2. 系统更新规则库，并刷新前端界面 3. 跳转至规则组建立界面，显示所有现有规则组 4. 系统更新规则组库，并刷新前端界面 5. 跳转至决策树界面，根据不同事件显示决策树 6. 系统更新规则库，并刷新前端界面 7. 系统执行规则，在规则命中表中增加用户本次操作，并实施配置的防控措施。 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-10 规则评价用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC9 | **用例名称** | 规则评价 |
| **测试日期** | 2022-3-25 | **优先级** | 高 |
| **测试目的** | 检测规则评价机制是否生效，是否可以查看规则命中情况，以及对规则准确率和有效性进行评估。 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 点击导航栏“规则评价-规则命中情况统计” 2. 从菜单中选择事件 3. 点击界面上的任一规则组或任意规则 4. 点击导航栏“规则评价-有效性评估” 5. 从菜单中选择事件 | | |
| **预期结果** | 1. 系统跳转至规则命中情况统计界面，系统左侧菜单显示全部事件 2. 系统界面上利用饼状图显示该事件中所有规则组的命中情况，每个饼状图中显示该规则组内所有规则的命中次数和命中率，以及规则组整体命中率 3. 系统界面上显示该规则组或该规则的全部命中列表 4. 系统跳转至规则有效性评估界面，系统左侧菜单显示全部事件 5. 系统界面上利用饼状图显示该事件中所有规则及规则组的准确率和有效性 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-11 阈值预测用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC10 | **用例名称** | 阈值预测 |
| **测试日期** | 2022-3-25 | **优先级** | 高 |
| **测试目的** | 检测规则阈值是否可以进行自动调整。 | | |
| **测试准备** | 为规则设置初始阈值 | | |
| **测试流程** | 1. 连续3天根据情况对阈值做出调整 2. 第四天凌晨检查阈值是否自动进行调整 3. 第四天检查阈值有效性和合理性是否低于80% 4. 检查风控策略师是否收到邮件 | | |
| **预期结果** | 1. 第四天0：00系统调用算法，对阈值进行预测，检查数据库中阈值已经自动修改为预测结果，并且规则运行时按照新阈值进行判定。 2. 若有效性和合理性低于80%，向风控策略师发送告警 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

表6-12 异常用户统计用例说明

|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | TC11 | **用例名称** | 异常用户统计 |
| **测试日期** | 2022-3-25 | **优先级** | 中 |
| **测试目的** | 检测系统是否可以展示异常用户名单，并可查看账号及设备规则命中情况 | | |
| **测试准备** | 用户进入系统 | | |
| **测试流程** | 1. 点击导航栏“规则评价-异常用户统计” 2. 点击账号名称 3. 点击规则命中次数 | | |
| **预期结果** | 1. 系统跳转至异常用户统计界面，显示所有异常设备及异常账号列表，包括设备/账号名称、命中规则次数 2. 系统跳转至用户操作统计界面 3. 系统跳转至规则命中列表界面 | | |
| **实际结果** | 测试结果符合预期，测试用例通过 | | |

6.3 非功能测试

非功能测试的目的是保证系统在满足功能性需求基础上，在易用性、正确性及性能方面也满足非功能性需求。由于系统易用性、灵活性不方便进行评估，正确性保障不是本人重点工作内容，因此本次非功能测试主要以性能测试为主，以响应时间为衡量标准来判断是否满足需求。

本次非功能测试主要分为两部分，第一部分是站在本系统使用者角度，对系统的部分数据展示功能进行点击，测试系统响应时间，具体测试结果如下：

表6-13 系统部分数据展示功能响应时间测试结果

|  |  |  |
| --- | --- | --- |
| **界面** | **操作** | **响应时间** |
| 平台整体风险分数查看界面 | 点击某一业务线，展示业务线分数详情 | 84ms |
| 某一业务线分数详情界面 | 设置筛选条件，进行搜索 | 76ms |
| 用户具体操作查看界面 | 输入用户uid进行搜索 | 65ms |
| 黑白名单库管理 | 查看黑名单 | 61ms |
| 黑白名单库管理 | 添加黑名单 | 63ms |
| 规则命中情况统计界面 | 查看某一规则组命中率 | 91ms |
| 规则命中情况统计界面 | 查看某一规则命中列表 | 82ms |
| 异常用户统计界面 | 查看异常账号及设备名单 | 79ms |
| 异常用户统计界面 | 查看某一账号操作列表 | 66ms |
| 敏感数据审计告警查看界面 | 查看敏感数据告警记录 | 71ms |
| 敏感数据审计日志查看界面 | 查看敏感数据点击日志 | 69ms |

第二部分是站在平台用户角度，测试风控系统的加入对于平台用户操作的处理时延影响情况。本次测试主要关注从系统接收用户操作，到所有防控措施实施完成的时间。具体测试结果如下：

表6-14 系统部分数据展示功能响应时间测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **事件** | **用户** | **是否触发规则** | **响应时间** |
| 注册 | 用户1 | 否 | 118ms |
| 注册 | 用户1 | 是 | 130ms |
| 注册 | 用户2 | 是 | 127ms |
| 注册 | 用户2 | 是 | 128ms |
| 登陆 | 用户1 | 否 | 102ms |
| 登陆 | 用户1 | 否 | 114ms |
| 登陆 | 用户1 | 是 | 167ms |
| 登陆 | 用户2 | 否 | 108ms |
| 登陆 | 用户2 | 是 | 124ms |
| 登陆 | 用户2 | 是 | 129ms |

测试结果表明，系统响应时间符合要求，非功能测试通过。

6.4 本章小结

本章主要对系统进行了功能测试和非功能测试，首先介绍了测试目标以及测试使用的硬件、软件版本信息，之后对于第三章中需求分析获得的11个用例依次进行黑盒测试，最后从系统使用者角度及平台用户角度，对系统性能进行了测试。功能测试和非功能测试均全部通过。

7 总结与展望

本章主要对项目的完整开发周期、研究意义及论文整体工作进行总结，同时结合企业风控实际应用，分析目前系统设计及实现过程中存在的不足，提出改进方案，并对系统未来发展方向进行展望。

7.1 全文总结

首先，本文介绍了风控系统产生的背景，并以租赁业务为例，分析了公司业务特点及易被黑产攻击的风险点所在，以及公司现有风控系统存在的问题，以此本系统的意义，以公司业务为基础，通过建立公司范围的风险控制体系，设定一系列风险控制策略和规则，在黑灰产攻击平台时迅速识别，进行拦截或者反攻击，从而保障平台安全。

接下来，从用户角度出发，分析该系统的使用者及其原始需求，将用户需求进行提炼整合，将系统按照业务逻辑拆分为4个模块：敏感数据审计、风险情况分析、规则引擎构建、规则执行监控，并针对四个模块进行了需求细化与用例描述。同时对系统非功能性需求进行了分析。

然后，学习了系统搭建过程中涉及到的技术与理论，包括SpringBoot框架、Redis缓存机制、Kafka、Flink流式计算引擎、三次指数平滑法、变异系数法以及规则引擎相关理论，并对系统进行架构设计，按照微服务架构对系统服务进行拆分。同时完成了数据库的设计，梳理各个实体之间的关系，并设计表结构。

在概要设计基础上，对4大模块分别进行详细设计，按照SpringBoot框架和RPC通信方式，将类分为实体类、前端控制器类（Controller）、数据服务层及实现类（Service和ServiceImpl）、数据库操作类（Mapper）、数据传输对象类（Dto）、工具类（Util）、常量接口类（Constant）及相关配置信息（Config），用类图说明类与类之间的关系，并用时序图阐述类之间的交互机制。

最后，根据详细设计对系统进行编码实现，并以需求分析为依据，对系统进行功能测试和非功能测试。

该系统的实现，不仅是帮助企业构建了专属的、可定制化的业务风控系统，更是保障了平台内上千万用户的信息与财产安全。

7.2 系统展望

目前该系统已经完成一期内容，交由公司上线使用。但是在系统的设计与实现过程中，依然存在可以改进的地方，在二期中可以对系统加以完善。

首先，在规则引擎构建方面，目前仅仅是基于配置化规则进行风险防控，虽然可以借助算法对规则阈值进行调整，但在系统应用过程中，依旧需要耗费较大人力成本用于规则调整。在后续发展中，可以加入模型、评分卡等机制，或通过机器学习的相关方法丰富规则引擎的防控机制，提高防控效果，且减少人力成本。

其次，是在规则执行方面。目前的规则状态仅包括生效和未生效两种，缺少灰度机制，即在规则生效之前，应存在试运行阶段，使用部分真实用户数据对规则有效性和准确性进行评估，评估通过后再正式生效。灰度机制可减少规则正式执行过程中造成的错杀或漏杀情况。

最后，风控系统永远不是一成不变的。随着业务的发展以及黑灰产攻击手段的更改，风控场景、规则、防控措施都应随之做出调整，以适应新的应用场景。风控之路任重而道远，但随着技术发展，风控系统也将日益完善，最终在企业和用户与不法分子之间构建起严密的安全防线。

参考文献

[1] 任彬,王梓林.基于Drools规则引擎的柔性化钣金产线优化排程[J].高技术通讯,2022,32(01):57-65.

[2] Meilian Ge. Construction of Intelligent Risk Control System for Commercial Banks in the Era of Big Data[J]. International Journal of Frontiers in Sociology,2021,3.0(11.0).

[3] 池炜成,史立学,刘智琼,朱明英.微服务架构下规则平台方案与规则迁移方法[J].现代计算机,2021(18):142-145.

[4] Luo Xi,Fu Ye,Yin Lihua,Xun Hao,Li Yixin. A scalable rule engine system for trigger-action application in large-Scale IoT environment[J]. Computer Communications,2021(prepublish).

[5] 陈逸聪. 基于Drools规则引擎的业务规则管理平台的设计与实现[D].西安电子科技大学,2021.DOI:10.27389/d.cnki.gxadu.2021.002535.

[6] 樊一航. 基于流处理引擎的业务安全平台的设计与实现[D].北京交通大学,2021.DOI:10.26944/d.cnki.gbfju.2021.002733.

[7] 李唯.基于SpringBoot+Mybatis的驾校预约系统设计与实现[J].电脑编程技巧与维护,2022(03):10-12.DOI:10.16184/j.cnki.comprg.2022.03.003.

[8] Digital Food ordering system based on Spring Framework[J]. International Journal of Recent Technology and Engineering,2020,8(6).

[9] 何小林,文永胜,李月茜.风控信息WEB系统的架构设计探讨[J].福建电脑,2013,29(12):43+57.

[10] Guanhong Chen,Jiangming Xu. Design and implementation of efficient Learning platform based on SpringBoot Framework[J]. Journal of Electronics and Information Science,2020,6(1).

[11] 陈开,陈柯辰.基于Spring框架的气象探测设备监控系统设计与实现[J].软件导刊,2020,19(07):112-116.

[12] 冯冼. 基于Kafka的高并发消息机制优化研究[D].湘潭大学,2021.DOI:10.27426/d.cnki.gxtdu.2021.001882.

[13] XiaoZhen Jiang. Design and Research on the Financial Approval Risk Control System[J]. Journal of Physics: Conference Series, 2020, 1533(2).

[14] Martin Heller. What is Apache Kafka? Scalable event streaming[J]. InfoWorld.com,2022.

[15] Varga Balázs,Balassi Márton,Kiss Attila. Towards autoscaling of Apache Flink jobs[J]. Acta Universitatis Sapientiae, Informatica, 2021,13(1).

[16] N. Deshai,B.V.D.S. Sekhar,S. Venkataramana. Processing Big Data with Apache Flink[J]. International Journal of Recent Technology and Engineering (IJRTE),2019,8(1s3).

[17] 史永胜,王文琪.基于改进三次指数平滑法的航材需求预测[J].计算机工程与设计,2020,41(11):3118-3122.DOI:10.16208/j.issn1000-7024.2020.11.019.

[18] 张一帆. 改进的动态三次指数平滑法火电厂发电量预测研究[D].河北工程大学,2020.DOI:10.27104/d.cnki.ghbjy.2020.000298.

[19]. 廖明阳,刘兴伟,马宏亮. 智能网联汽车信息安全风控系统研究[J].计算机时代,2019(10):19-23.DOI:10.16644/j.cnki.cn33-1094/tp.2019.10.006.

[20] Jianmei Ma. Design and Implementation on Credit Card Risk Control System[C]//.Proceedings of 2016 3rd International Conference on Materials Engineering, Manufacturing Technology and Control(ICMEMTC 2016).,2016:1850-1856.

[21] 常昊. 互联网金融风控系统贷中风控和预警功能的设计与实现[D].南京大学, 2020. DOI:10.27235/d.cnki.gnjiu.2020.001604.

[22] David Carral, Irina Dragoste,Markus Krötzsch. Reasoner = Logical Calculus + Rule Engine[J]. KI - Künstliche Intelligenz, 2020(prepublish).

致 谢

在本篇论文完成的同时，我要感谢我的导师王方石老师，在毕业论文的全过程中对我悉心指导，从选题到开题、中期检查，再到最后的论文撰写都提出了很多建设性的指导意见。通过老师的指导，我不仅在技术方面有所提高，更学会怎样的表达方式更能准确的向读者传达我的想法，在此向导师表示衷心的感谢。

其次我要感谢我的父母一直以来对我的照顾、包容、理解、支持，因为有他们的存在，我才可以无后顾之忧地做自己想做的事，将最真实的自己展现给这个世界。

我还要感谢四年来对我提供帮助的老师们，他们教会我知识，教会我表达，教会我为人处世。

我要感谢我实习单位的同事们，感谢他们包容第一次正式迈向社会的我的无知，包容我犯下的小错误，包容我因为冬奥和疫情导致的长时间远程办公，并且无私的向我提供帮助。因为他们的存在，让我的实习期充实、快乐、有所收获。

我要感谢我的朋友们，感谢你们长久以来对我的信任、理解、陪伴，我会珍惜和你们每个人的情谊，愿友谊长存。

最后我要感谢我自己，感谢我4年来的努力，感谢我低谷时的坚持，感谢我得意时的冷静。

本科毕业，是一段旅程的结束，也是下一段旅程的开始。我会带着所有人的爱与鼓励，奔赴下一场山海。长路漫漫，愿你我在未来的某一天，顶峰相见。