****



**研 究 生 毕 业 论 文**

**（申请工程硕士学位）**

|  |  |
| --- | --- |
| **论文题目** | 饿了么体验保障平台的  设计与实现 |
| **作者姓名** | 金思晔 |
| **学科、专业名称** | 工程硕士(软件工程方向) |
| **研究方向** | 软件工程 |
| **指导教师** | 王金庆 高级工程师 |

**2019 年 4 月 日**

**学 号： MF1832073**

**论文答辩日期： 2020 年 月 日**

**指 导 教 师： （签字）**

**饿了么体验保障平台的**

**设计与实现**

|  |  |
| --- | --- |
| **作 者:** | **金思晔** |
| 指导教师: | **王金庆　高级工程师** |

|  |
| --- |
| **南京大学研究生毕业论文** |
| **(申请工程硕士学位)** |

|  |
| --- |
| **南京大学软件学院** |
| **2019年4月** |

**The Design and Implementation of**

**ELEME Experience Guarantee Platform**

**Jin, Si Ye**

**Submitted in partial fulfillment of the requirements for the degree of Master of Engineering**

Supervised by

Senior Engineer **Wang, Jinqing**

Software Institute

**NANJING UNIVERSITY**

Nanjing, China

April, 2019

# 摘 要

# Abstract

# 图目录

# 表目录

目录

[摘 要 V](#_Toc25947911)

[Abstract VI](#_Toc25947912)

[第一章 引言 8](#_Toc25947913)

[1.1 项目背景 8](#_Toc25947914)

[1.2 国内外发展现状 9](#_Toc25947915)

[1.3 本文主要工作 9](#_Toc25947916)

[1.4 本文的组织结构 9](#_Toc25947917)

[第二章 技术综述 9](#_Toc25947918)

[2.1 SpringBoot 9](#_Toc25947919)

[2.2 MaxQ 9](#_Toc25947920)

[2.3 Huskar 10](#_Toc25947921)

[2.4 WorkFlow 10](#_Toc25947922)

[2.5 Redis 10](#_Toc25947923)

[2.6 本章小结 10](#_Toc25947924)

[第三章 系统分析与设计 11](#_Toc25947925)

[3.1 系统总体规划 11](#_Toc25947926)

[3.2 系统需求分析 11](#_Toc25947927)

[3.2.1 规则管理模块 11](#_Toc25947928)

[3.2.2 审核流模块 12](#_Toc25947929)

[3.2.3 业务处理模块 13](#_Toc25947930)

[3.3 非功能性需求 15](#_Toc25947931)

[3.5 数据库设计 15](#_Toc25947932)

[3.6 本章小结 15](#_Toc25947933)

[第四章 系统实现 15](#_Toc25947934)

[4.1 规则配置模块 15](#_Toc25947935)

[4.2 审核流模块 16](#_Toc25947936)

[4.3 业务处理模块 17](#_Toc25947937)

[4.5 系统测试 17](#_Toc25947938)

[4.5.1 测试环境 17](#_Toc25947939)

[4.5.2 测试设计 17](#_Toc25947940)

[4.5.3 测试结果 17](#_Toc25947941)

[第五章 总结与展望 17](#_Toc25947942)

[5.1 总结 17](#_Toc25947943)

[5.2 工作展望 17](#_Toc25947944)

[参 考 文 献 18](#_Toc25947945)

[致 谢 18](#_Toc25947946)

# 第一章 引言

## 1.1 项目背景

论文选题源自于本人实习时参与的“饿了么热斧平台”。

对超时场景的处理是服务类平台不可或缺的能力，饿了么、美团、滴滴、淘宝等都有对超时场景的判定和相应的补偿操作。对于不同的超时场景，有不同的赔付策略，例如滴滴打车，司机没有在规定时间内到达指定地点，用户可以无责取消这笔订单；用户点外卖后，骑手没有在规定时间内将外卖送到指定地点，那么用户就会收到系统赔付的优惠券等。

饿了么超时场景的判定分散在各个服务当中，例如逆向系统、赔付系统、用户触达系统等。每次规则的修改，产品经理都需要各个系统依次去同步，额外的增加了产品经理的工作量。目前的规则是硬编码在代码中，如果需要修改规则，需要产品提紧急需求，开发人员出技术方案，各方评审方案，开发人员开发，测试人员测试，然后在发布窗口发布。对于开发人员来说，增加了高优先级的任务，需要优先完成紧急需求，增加了开发人员的工作量；对于业务方来说，需求得不到快速响应，延误了策略的实施。由于上述的这些问题，运营人员无法做到对规则的细粒度把控：例如，根据天气配置规则，根据城市配置规则，根据订单来源配置规则等，每次规则变动的需求都会造成牵一发而动全身的影响。因此开发一个能够支持细粒度配置规则，快速响应业务方需求的平台很有必要。

为了解决上述问题，本文设计和实现了“饿了么热斧系统”，系统通过网页配置规则的方式，自由把控规则的粒度；通过实时的解析规则，能够及时响应需求的变更。

## 1.2 国内外发展现状

规则的可配置化多应用于业务逻辑多变的情况。

网易考拉规则引擎平台是网易旗下的一款规则引擎系统，主要的业务对接方为敏感日志操作判定规则以及业务风控中风控判定规则。系统分为四个层面服务层、引擎层、计算层和存储层。服务层负责接入服务，引擎层主要负责执行规则，计算层负责指标计算和模型训练，存储层负责存储规则、规则执行结果、指标等。其中规则引擎是平台的核心，平台使用了groovy脚本进行规则引擎的开发。

同程艺龙机票自研实时规则引擎，基于groovy实现

阿里巴巴财务平台业务方接入平台，基于groovy实现

美团酒店实时规则引擎，基于aviator实现

饿了么营销平台，基于aviator实现。

## 1.3 本文主要工作

本文的目标是做一个能够快速响应业务方需求、对规则粒度自由把控、判定是否命中规则的系统。通过研究当前已有的规则引擎，阅读github上优秀的开源代码，结合生产过程中遇到的问题，提出相应的解决方案。

综上所诉，本文的主要工作有：

1. 分析了当前系统的现状和存在问题。
2. 详细分析了系统的各类需求，并确定系统的总体架构和数据库解构。
3. 根据需求分析和总体设计，完成了模块的划分，并实现各个模块。

## 1.4 本文的组织结构

# 第二章 技术综述

## 2.1 SpringBoot

Web框架

## 2.2 MaxQ

消息队列

## 2.3 Huskar

配置中心

## 2.4 WorkFlow

任务调度中心

## 2.5 Redis

缓存

## 2.6 本章小结

##

# 第三章 系统分析与设计

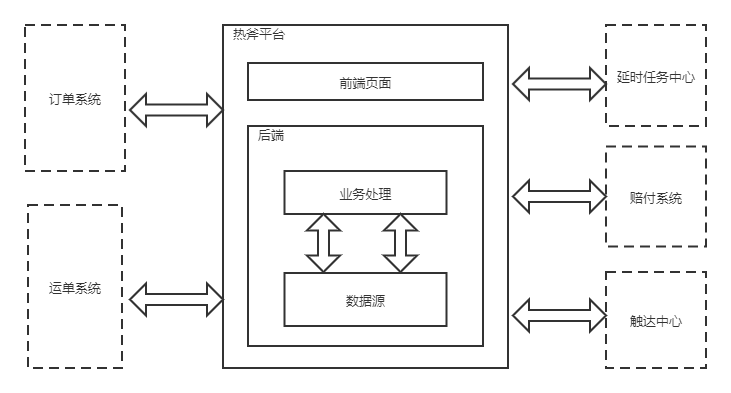
## 3.1 系统总体规划

饿了么热斧系统的用户角色有三种：运营人员和订单、运单系统和开发人员。

运营人员是规则的主要制定人，通过热斧平台，运营人员可以自由的编辑规则，规则在经过上级审核通过之后，既可以纳入规则库中，以供后续进行规则匹配。规则的启用时间可以设置，当规则快要超过过期时间的时候，系统需要能够给对应的运营人员发送邮件通知。规则提交审核之后，运营人员可以根据指定条件查看规则审核的状态，对于未到达审核终态（审核失败或审核通过）的规则，运营人员可以终止该条规则的审核。除此之外，运营人员还需要得到规则命中和未命中情况的反馈，因此平台需要提供根据订单号、场景名称、规则名称等维度查询命中记录的能力，以便运营人员进行统计，及时的调整规则。

订单、运单系统可以通过热斧平台根据已有的规则库匹配订单或运单是否命中超时场景，在命中超时场景后，平台根据规则中的补偿操作，完成后续对商户、用户的补偿。订单有这样的几个主节点：用户下单、用户已付款、订单待接单、订单已接单、订单退款中、订单已完结。对应的每次节点状态的变更，订单系统都会给订单消息队列中发送对应的消息，消费方在接受到消息后，会根据消息的类型，进行相应的处理。运单系统也是如此。因此，热斧平台需要监听订单和运单的消息队列，对于每次状态的改变，平台都需要在已有的规则库中进行匹配。在匹配的过程中，需要区分规则是立即匹配还是需要延迟指定时间进行匹配，如果是需要延迟指定时间进行匹配的规则，那么平台还需要创建定时任务，到达指定时间之后再进行匹配。

开发人员对热斧系统只有查看权限，在进行问题处理的时候，开发人员需要查看订单命中的规则和未命中原因。

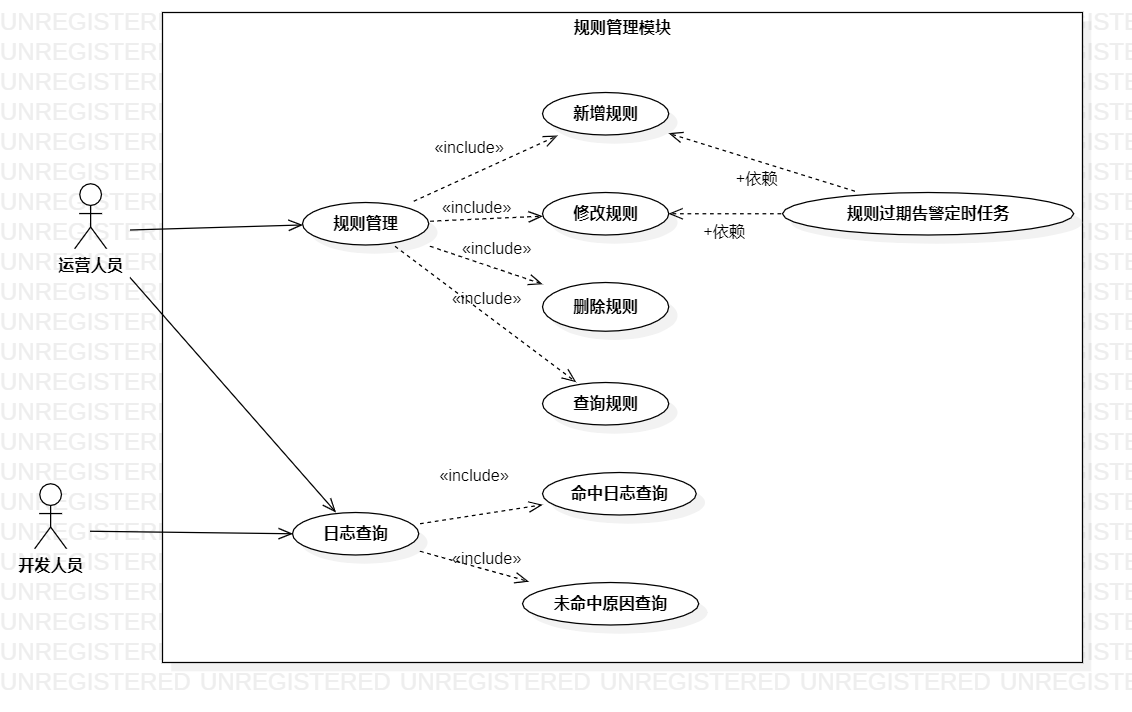


## 3.2 系统需求分析

基于3.1章节中系统的总体规划，结合热斧系统的实际使用场景，系统需要满足以下的需求：

1. 提供规则管理能力，运营人员在前端页面上能够对规则进行新增，修改，删除，查询等操作，并且能够看到规则命中和规则未命中的情况。
2. 提供规则预处理能力，将前端配置好的规则转换成系统能够执行的规则。
3. 提供规则审核流管理的能力，运营人员能够管理规则，包括查找，终止审核中的规则，系统需要在当前审核节点通过的时候，进行审核人的扭转，并且保证消息丢失的情况下能够继续审核。
4. 提供业务处理能力，系统需要监听订单、运单消息，消费这些消息的同时需要与规则库中的规则进行匹配，并且对匹配后的结果进行下一步的处理。

### 3.2.1 规则管理功能



规则管理用例图如图所示，规则管理模块包含了规则管理和日志查询两个模块。

规则管理子模块中，运营人员可以对规则进行操作，以快速响应业务需求，包括对规则的新增、删除、修改、查询。在新增和修改规则的时候，会创建告警定时任务，当规则有效期快过期的时候，告警任务就会发送邮件提醒用户规则快过期，让运营人员提前做处理。

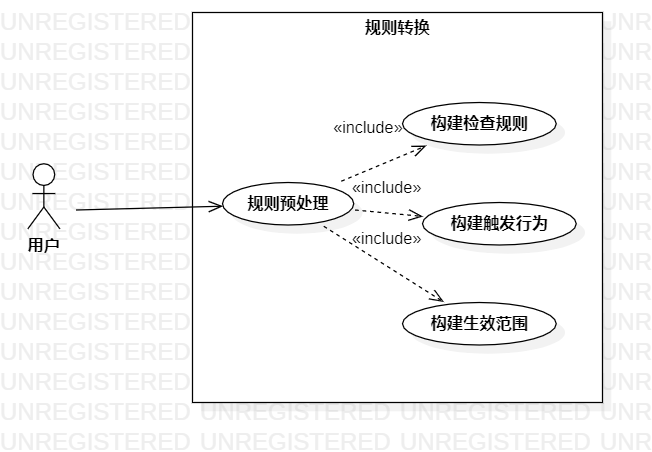
日志查询子模块中，运营人员根据命中日志和未命中原因可以优化当前的规则；开发人员可以查询未命中原因进行故障的排查。

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC1 | 名称 | 操作规则 |
| 描述 | 运用在配置页面上进行规则配置和查看 | | |
| 参与者 | 运营 | | |
| 前置条件 | 用户已经经过身份验证 | | |
| 后置条件 | 触发创建告警任务、触发审核流 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户点击新增规则。 2. 用户进行规则的配置。 3. 用户点击提交审核 4. 系统保存规则草稿，并触发审核流，触发创建告警任务。 | | |
| 拓展流程 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC2 | 名称 | 创建超时告警任务 |
| 描述 | 创建规则超时告警任务，定时发送告警邮件 | | |
| 参与者 | 系统 | | |
| 前置条件 | 用户修改/创建规则 | | |
| 后置条件 | 无 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户修改/创建规则。 2. 规则传送到后端 3. 后端生成邮件 4. 延时任务中心创建任务。 | | |
| 拓展流程 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC3 | 名称 | 查询日志 |
| 描述 | 运营和开发人员通过平台查看规则命中日志 | | |
| 参与者 | 运营人员、开发人员 | | |
| 前置条件 | 用户通过权限校验 | | |
| 后置条件 | 无 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户点击日志选项卡 2. 选择日志筛选条件 3. 点击查询按钮 | | |
| 拓展流程 | 无 | | |

### 规则预处理功能



规则按照功能可以分为三大类：超时判定规则、规则生效范围、和规则触发行为。用户配置完成的规则是以对象的方式传到后端系统，后端系统需要把存储在对象中的规则转换成能够执行的规则-以树的方式进行组织，树的每个节点代表着一个操作，其叶子节点是操作需要的参数：

如上图所示，AND节点代表一个操作，其叶子节点是AND节点的两个参数，

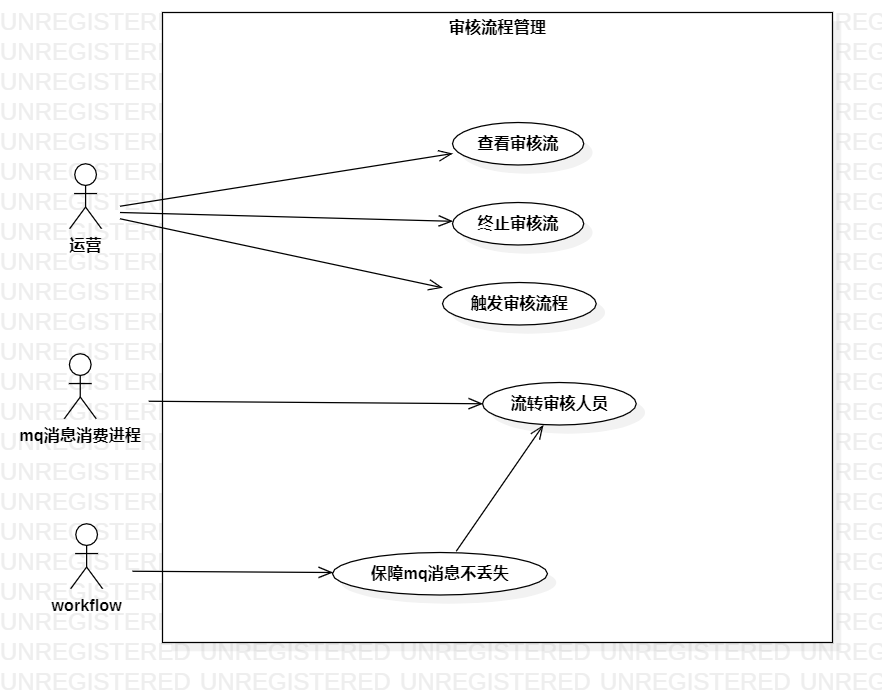
这个操作的含义就是输出AND节点两个值的逻辑与值。作为参数的节点可以是一个常量，也可以是另外的一个操作。因此将存储在对象中的规则按照这样的方式组织，并对该树进行后续遍历，这样就可以达到执行规则的效果。

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC4 | 名称 | 构建检查规则 |
| 描述 | 根据用户配置的检查规则构建能够真实执行的规则 | | |
| 参与者 | 热斧系统 | | |
| 前置条件 | 用户创建/修改规则 | | |
| 后置条件 |  | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户提交规则创建/修改 2. 后端根据检查规则构建规则执行树 | | |
| 拓展流程 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC5 | 名称 | 构建触发行为 |
| 描述 | 运营和开发人员通过平台查看规则命中日志 | | |
| 参与者 | 热斧系统 | | |
| 前置条件 | 用户创建/修改规则 | | |
| 后置条件 | 无 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户提交规则创建/修改 2. 后端根据触发行为构建动作执行树 | | |
| 拓展流程 | 无 | | |

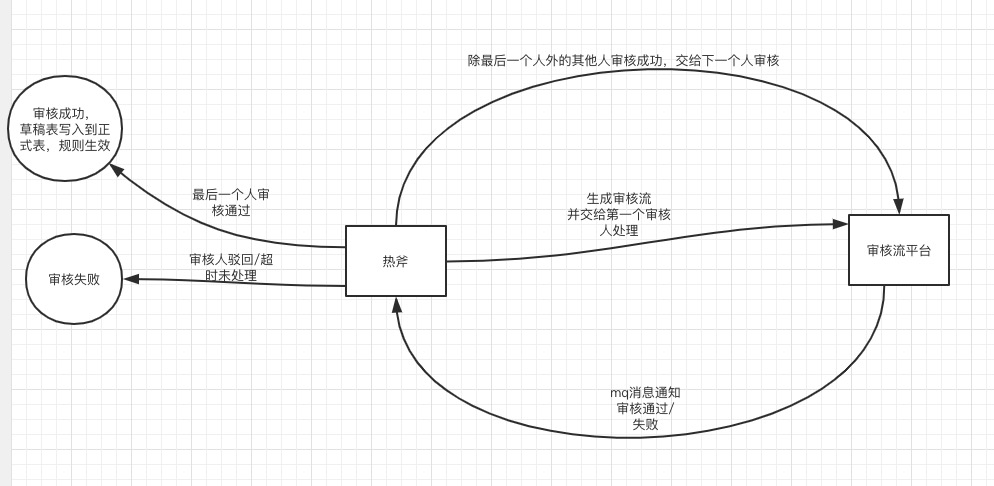
|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC6 | 名称 | 构建生效范围 |
| 描述 | 运营和开发人员通过平台查看规则命中日志 | | |
| 参与者 | 热斧系统 | | |
| 前置条件 | 用户创建/修改规则 | | |
| 后置条件 | 无 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户提交规则创建/修改 2. 后端根据生效范围构建生效范围执行树 | | |
| 拓展流程 | 无 | | |

### 审核流程管理功能



上图是审核流程管理的用例图。运营人员在创建、修改规则之后，就会触发审核流程，系统会在审核平台上创建一个审核流程。运营人员可以根据指定的条件查询审核流当前的状态，根据需求的变化，终止某一条规则的审核。

系统借助于审核平台完成规则的审核，因此系统需要监听审核平台的消息队列，审核流的某个节点审核通过之后，审核平台会向消息队列中发送审核通过的消息，系统接受到消息后，会将审核节点流转到下一个节点，直到审核通过或者审核被拒。



上图为审核流转图

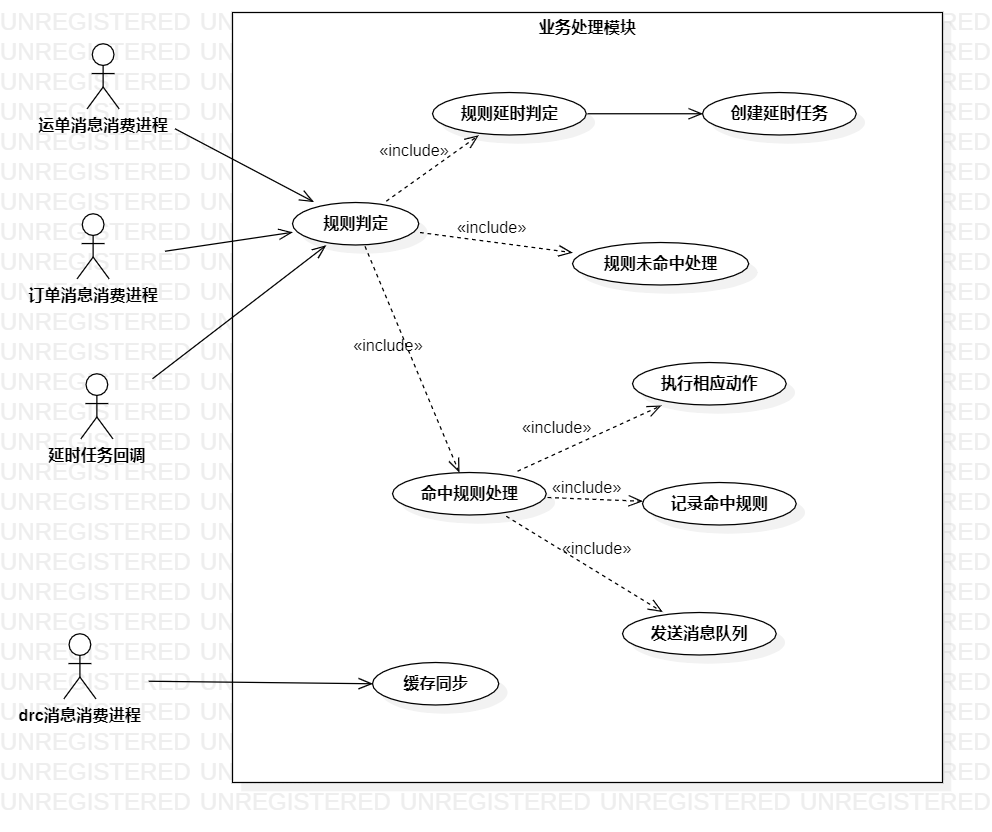
审核流程的流转监听消息队列来实现，消息队列的消息存在丢失的情况，如果审核人员审核通过了，但是通过的消息却丢失了，该审核流将无法流转，最终会停止在某个审核节点上。因此需要在workflow上配置任务，定时检测消息是否丢失。

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC7 | 名称 | 操作审核流 |
| 描述 | 对审核流进行操作，包括查看、终止审核流 | | |
| 参与者 | 运营 | | |
| 前置条件 | 运营配置好规则，并提交审核 | | |
| 后置条件 |  | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户切换至审核流选项卡 2. 选择审核流筛选规则 3. 点击查询 4. 点击终止审核六终止规则的审核 | | |
| 拓展流程 |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC8 | 名称 | 扭转审核流 |
| 描述 | 运营修改规则 | | |
| 参与者 | 热斧系统 | | |
| 前置条件 | 用户创建审核流 | | |
| 后置条件 | 无 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 监听审核流消息队列 2. 消息到达时解析审核流消息 3. 根据审核流消息终止、成功、扭转审核流 | | |
| 拓展流程 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC9 | 名称 | 保证审核流 |
| 描述 | 保证消息丢失之后审核流程能够扭转下去 | | |
| 参与者 | 热斧系统、workflow系统 | | |
| 前置条件 | 审核流消息丢失 | | |
| 后置条件 |  | | |
| 优先级 | 中 | | |
| 正常流程 | 1. workflow中配置定时任务和回调接口 2. 比对审核流平台和热斧平台审核记录 3. 将热斧平台审核记录更新为审核流平台数据 | | |
| 拓展流程 | 无 | | |

### 业务处理功能



上图为业务处理功能的用例图，当订单、运单状态改变的时候，就会往消息队列中发送状态改变消息，系统监听到消息后，会根据已有的规则进行判定，判定结果分为三种：命中规则、未命中规则和延迟判定。命中规则后需要执行后续的Action，对命中规则进行记录，同时发送命中消息到消息队列；未命中谷规则需要记录未命中的原因；延时处理需要在延时中心创建延时任务，等到达指定时候之后再去执行。

drc消息是异地多活数据同步消息。为了防止一个机房的机子全部宕机，系统一般会在不同的机房里部署，以两个机房为例，用户一次只会影响一个机房的数据，当用户影响a机房的数据的时候，a机房就会给b机房的消息队列发送一条数据同步消息，b机房接收到消息后，就会将b机房的数据就会和a机房的数据进行同步，保证最终一致性，b机房同理。机房之间发送的同步消息就是drc消息。为了避免频繁的访问数据库，数据库中的规则需要缓存到redis中，当数据库中的数据发生变化的时候，redis中的数据也需要同步，监听drc消息为了同步机房的redis消息。

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC10 | 名称 | 修改规则 |
| 描述 | 运营修改规则 | | |
| 参与者 | 运营 | | |
| 前置条件 | 用户已经经过身份验证 | | |
| 后置条件 | 触发创建告警任务、触发审核流 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户点击修改规则。 2. 用户进行规则的配置。 3. 用户点击提交审核 4. 系统保存规则草稿，并触发审核流，触发创建告警任务。 | | |
| 拓展流程 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC11 | 名称 | 修改规则 |
| 描述 | 运营修改规则 | | |
| 参与者 | 运营 | | |
| 前置条件 | 用户已经经过身份验证 | | |
| 后置条件 | 触发创建告警任务、触发审核流 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户点击修改规则。 2. 用户进行规则的配置。 3. 用户点击提交审核 4. 系统保存规则草稿，并触发审核流，触发创建告警任务。 | | |
| 拓展流程 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | UC12 | 名称 | 修改规则 |
| 描述 | 运营修改规则 | | |
| 参与者 | 运营 | | |
| 前置条件 | 用户已经经过身份验证 | | |
| 后置条件 | 触发创建告警任务、触发审核流 | | |
| 优先级 | 中 | | |
| 正常流程 | 1. 用户点击修改规则。 2. 用户进行规则的配置。 3. 用户点击提交审核 4. 系统保存规则草稿，并触发审核流，触发创建告警任务。 | | |
| 拓展流程 | 无 | | |

## 3.3 系统总体设计

### 3.3.1 规则设计

运营人员在进行规则配置的时候并非是随意的配置，而是需要在指定的规则集合下进行配置。

规则主要分为三大部分，超时判定规则、规则生效范围和规则触发动作。

在规则触发条件中，规则的定义如下：

|  |  |
| --- | --- |
| 规则名 | 可选值 |
| 订单类型 | 及时单 |
| 预定单 |
| 全部 |
| 主场景 | 提交订单 |
| 实际用户T |
| 预计时长 |
| 用户付款 |
| 商家接单 |
| 商家呼单 |
| 骑手接单 |
| 骑手取餐 |
| 骑手送达 |
| 业务阶段：开始节点/结束节点 | 提交订单 |
| 实际用户T |
| 预计时长 |
| 用户付款 |
| 商家接单 |
| 商家呼单 |
| 骑手接单 |
| 骑手取餐 |
| 骑手送达 |
| 未发生事件 | 商家接单 |
| 商家呼单 |
| 骑手接单 |
| 骑手取餐 |
| 骑手送达 |
| 时间比较符号 | + |
| - |
| > |
| < |
| = |

在规则生效范围中，规则定义如下

|  |  |
| --- | --- |
| 规则名 | 可选值 |
| 城市区域 | 代理商城市（全国2844个三级行政区） |
| 直营城市（（全国2844个三级行政区）） |
| 订单来源 | PC端web |
| 手机端web |
| Iphone App |
| Android App |
| openapi\_platform |
| 微信小程序 |
| 微信KA |
| 物流类型 | 蜂鸟专送 |
| 众包 |
| 择时达专送 |
| 新零售专送 |
| 蜂鸟快送 |
| 新零售 |
| E配送 |
| 蜂鸟混合送 |
| 蜂鸟质选 |
| 全城送 |
| 蜂鸟独家 |
| 业务类型 | 餐饮 |
| 新零售 |
| 全选 |

规则触发动作

|  |  |  |
| --- | --- | --- |
| 规则名 | 可选值 | |
| 用户触达 | 触达类型 | 短信 |
| 消息推送 |
| 接受者 | 用户 |
| 客户 |
| 短信模板 | |
| 备注说明 | |
| 生效范围 | |
| 用户取消 | 是否支持 | 支持 |
| 不支持 |
| 生效范围 | |
| 用户赔付 | 赔付类型 | 红包 |
| 优惠券 |
| 积分 |
| 触发时间点 | 订单取消或完成时 |
| 场景触发时 |
| 赔付规则名称 | |
| 渠道号 | |
| 生效范围 | |
| 商户索赔 | 是否支持 | 支持 |
| 不支持 |
| 生效范围 | |

函数设计

|  |  |  |
| --- | --- | --- |
| 函数名 | 函数介绍 | 参数介绍 |
| AND | 逻辑与 | 两个参数，返回两个参数的与值 |
| OR | 逻辑或 | 两个参数，返回两个参数的或值 |
| ENV | 从上下文中获取指定对象，并从对象中获取指定值。 | 两个参数，对象的key和值的key。 |
| IN | 数值在指定集合中 | N个参数，第一个参数为key，后续参数为指定结合 |
| LT | 小于 | 两个参数， |
| LE | 小于等于 | 两个参数 |
| GT | 大于 | 两个参数 |
| GE | 大于等于 | 两个参数 |
| ADD | 加 | 两个参数 |
| SUBTRACT | 减 | 两个参数 |
| MULTIPLICATION | 乘 | 两个参数 |
| TRIGGER\_AT | 触发时间 | 两个参数 |
| ORDER\_AND\_DELIVERY\_STATUS | 订单和运单状态 | 两个参数 |
| DELIVERY\_STATUS\_DOUBLE\_CHECK | 运单装填双重检测 | List集合 |

### 3.3.2 消息体设计

订单消息

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| Type | String | 消息代表的含义 |
| Description | String | 具体说明 |
| Paload.order\_id | Long | 订单id |
| Paload.role | Integer | 消息发起角色 |
| Paload.restaurant\_id | Integer | 餐馆id |
| Paload.user\_id | Integer | 下单用户id |
| Paload.come\_from | Integer | 订单来源 |
| Paload.is\_book | Integer | 预订单 |
| Paload.sent\_at\_millsec | Long | 发送时间戳 |
| Paload.order\_mode | Integer | 订单模型 |

运单消息

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| Type | String | 消息代表的含义 |
| Description | String | 具体说明 |
| Payload.carrier\_driver\_id | String | 骑手id |
| Payload.carrier\_id | String | 配送商id |
| Payload.deliverer\_name | String | 骑手名称 |
| Payload.deliverer\_phone | String | 骑手电话 |
| Payload.order\_id | Long | 订单id |
| Payload.send\_at | Long | 发送时间 |
| Payload.state | String | 运单主状态 |
| Payload.subState | String | 运单子状态 |
| Payload.tracking\_id | Long | 运单id |
| Payload.update\_time | Long | 更新时间 |

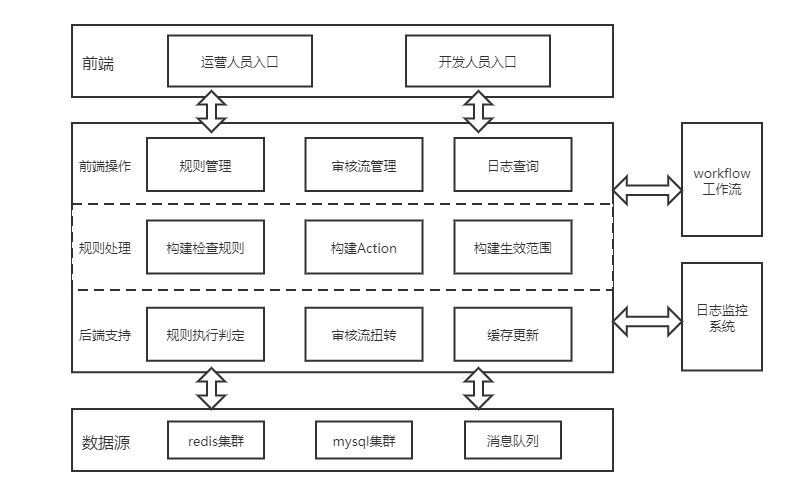
超时消息

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| Type | Integer | 消息代表的含义 |
| Event\_name | String | 事件名称 |
| Payload.order\_id | Long | 订单id |
| Payload.timeout | Long | 超时时间 |
| Payload.start\_time | Long | 开始时间 |

审批流消息

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| Apply\_instance\_id | Long | 审批实例id |
| Previous\_state | String | 上一个状态 |
| Current\_state | String | 当前状态 |
| Aaudit\_log\_id | Long | 审核操作ID |
| Event\_type | EventType | 事件类型 |
| Parameters | String | 参数 |
| Operator | Operator | 操作人 |

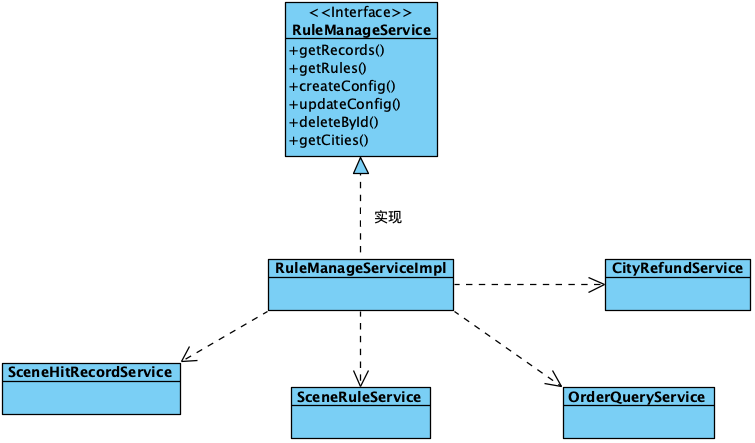
### 3.3.3 系统架构设计



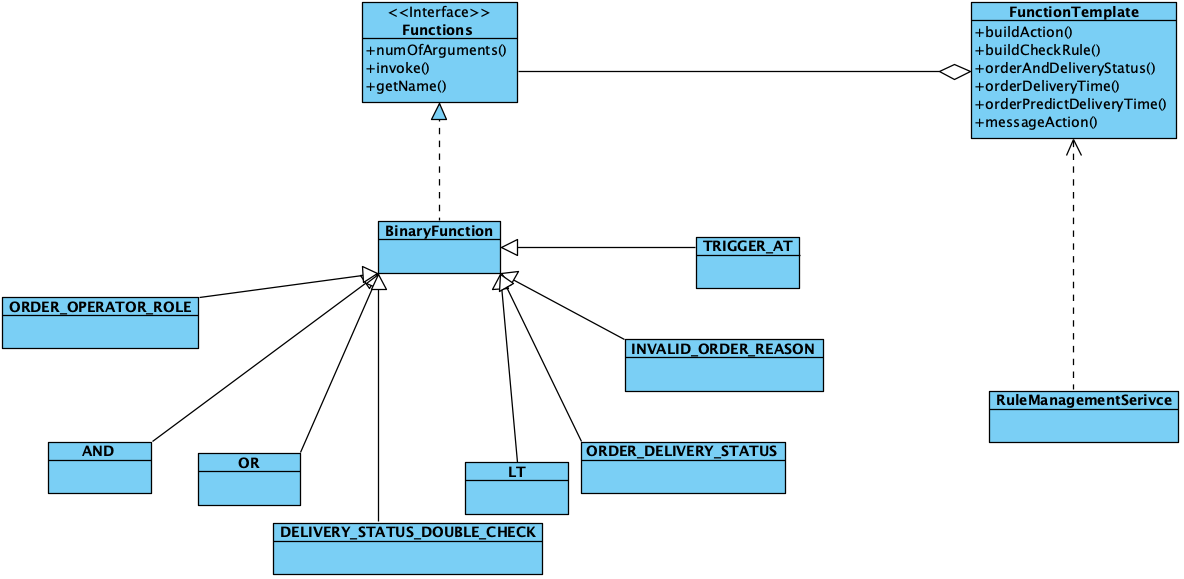
### 3.3.4 功能组成设计

## 3.4 系统模块设计

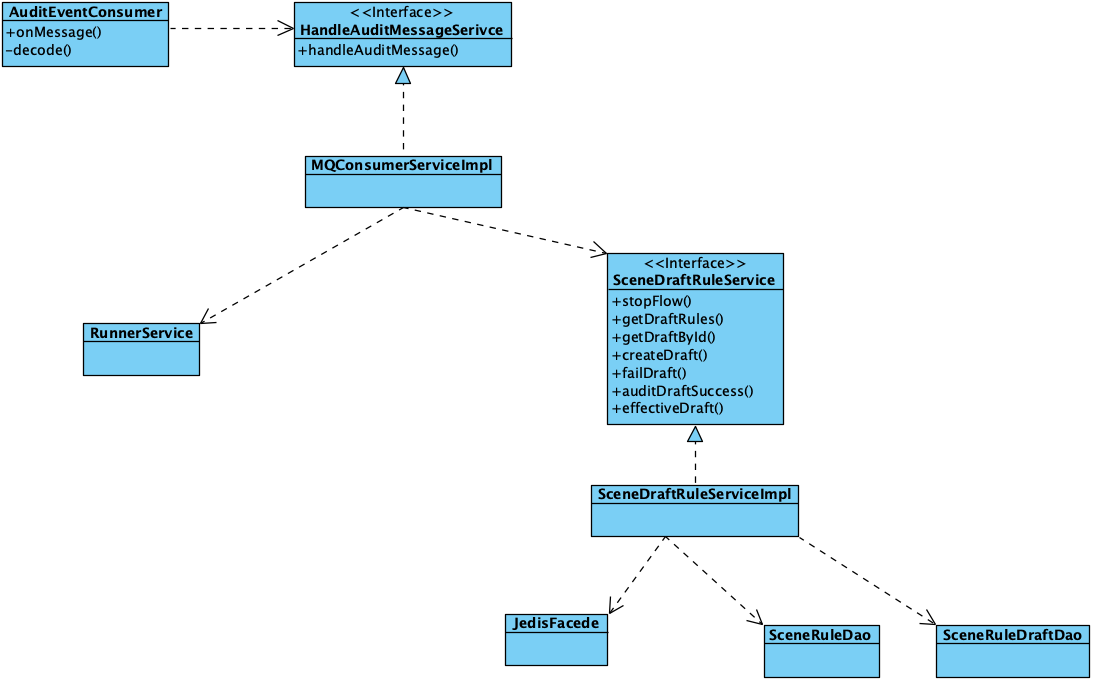
### 3.4.1 规则管理模块设计



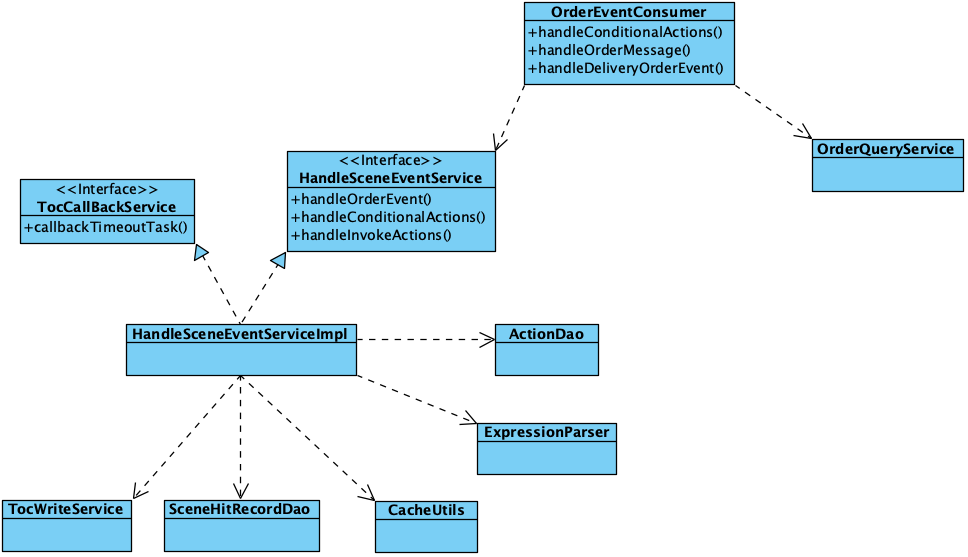
### 3.4.2 规则预处理模块设计



### 3.4.3 审核流程模块设计



### 3.4.4 业务处理功能



## 3.5 数据库设计

规则表：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| id | Long | 规则id |
| type | int | 场景类型 |
| description | int | 场景码 |
| Trade\_node | Varchar | 主场景 |
| Start\_trade\_node | Varchar | 业务开始节点 |
| End\_trade\_node | Varchar | 业务结束节点 |
| Trigger\_event | Varchar | 触发事件 |
| Status | Int | 规则状态 |
| Start\_time | Long | 开始时间 |
| End\_time | Long | 结束时间 |
| Check\_rules | Text | 校验规则 |
| Actions | Text | 触发行为 |

规则草稿表

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| id | Long | 规则id |
| type | int | 场景类型 |
| description | int | 场景码 |
| Trade\_node | Varchar | 主场景 |
| Start\_trade\_node | Varchar | 业务开始节点 |
| End\_trade\_node | Varchar | 业务结束节点 |
| Trigger\_event | Varchar | 触发事件 |
| Status | Int | 规则状态 |
| Start\_time | Long | 开始时间 |
| End\_time | Long | 结束时间 |
| Check\_rules | Text | 校验规则 |
| Actions | Text | 触发行为 |

订单命中规则表

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| Id | Long | Id |
| Order\_id | Long | 订单id |
| Scene\_id | Long | 场景id |
| Scene\_code | Int | 场景code |
| Occure\_time | Long | 出现时间 |
| Action\_result | Text | 场景触发的action结果 |
| timeout | Long | 超时时间 |
| Delay | Long | 延迟回调时间 |

场景规则动作表

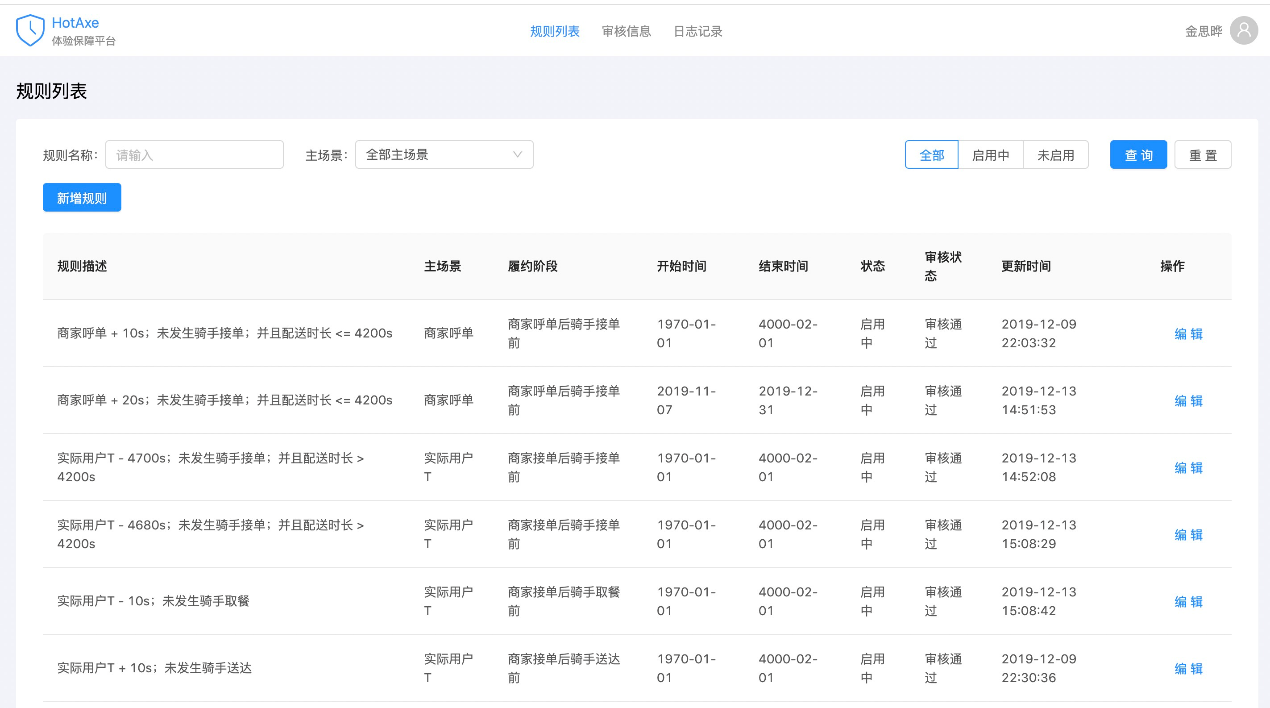
|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| id | Long | Id |
| Record\_id | Long | 触发记录id |
| Status | Long | 场景动作状态 |
| Params | Text | 场景动作参数 |
| Results | Text | 场景动作结果 |
| Creator | Varchar | 创建者 |
| updator | Varchar | 更新者 |
| Type | Int | 动作类型 |
| Order\_id | Long | 订单号 |
| trigger\_event | Varchar | 触发事件 |
| Rule\_id | Long | 规则id |
| Name | Varchar | 动作名称 |

## 3.5 本章小结

##

# 第四章 系统实现

## 4.1 规则管理模块







## 4.2 规则预处理模块

## 4.3 审核流模块



## 4.4 业务处理模块

##

## 4.5 系统测试

##

### 4.5.1 测试环境

##

### 4.5.2 测试设计

##

### 4.5.3 测试结果

##

# 第五章 总结与展望

## 5.1 总结

##

## 5.2 工作展望

##

# 参 考 文 献

##

# 致 谢

##