****



**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)

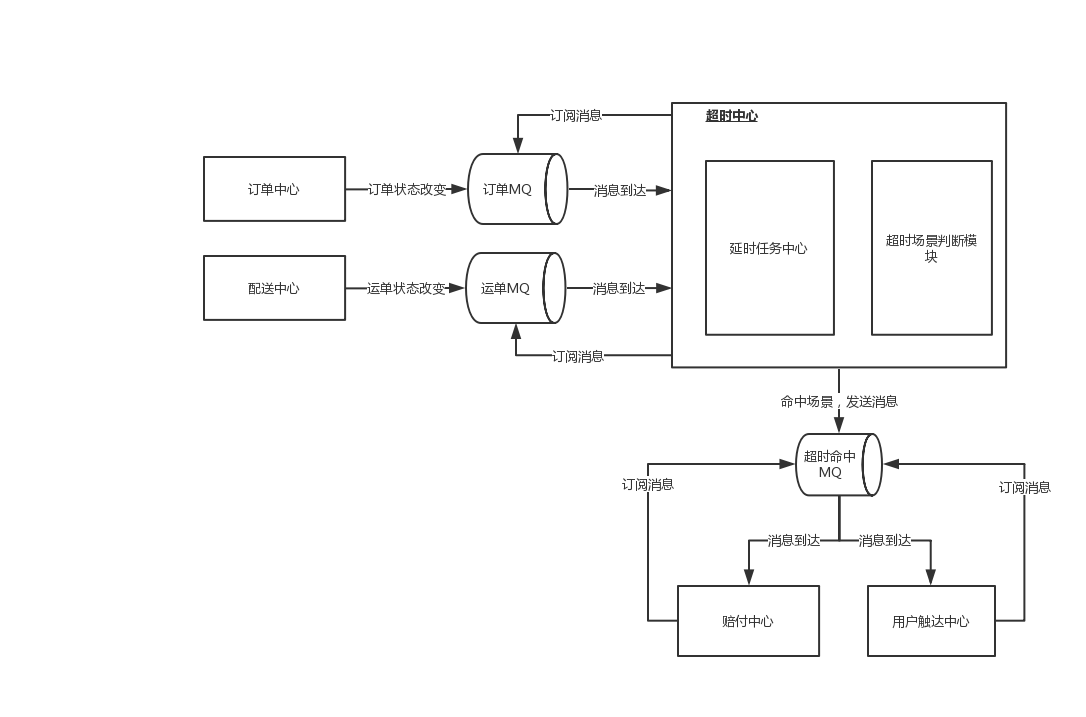
[致 谢 18](#_Toc25947946)

# 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“饿了么体验保障平台”。

饿了么目前拥有一个超时系统，内部定义了超时场景，当命中超时场景时，系统就会去做相应的处理。例如，用户下单后，商户超时未接单，此时系统会允许用户无责取消，同时会发送消息提醒商家；骑手取餐后超时未送达，此时系统会给用户发放一些优惠券等。



（上图为原有系统）

然而，随着市场的不断发展，产品的不断迭代，当前订单超时场景中，除了物流/骑手本身原因导致订单履约超时外，还涉及到商户以及用户的原因。例如：商户没能够即时准备好餐品导致超时、商户原因中断配送导致超时、用户联系不上导致超时等。超时涉及的场景越来越丰富，这也直接导致了运营人员需要及时更新超时场景，并且根据场景制定对应的处理策略。当前超时场景的判断是硬编码在超时中心中，每次需求的变更都需要进行需求评估，方案评估，然后再去修改源码，不但耗费人力，而且策略不能够及时生效。除此之外，为了将补偿策略制定的更加精细化，运营人员需要根据不同地区、不同城市制定不同的补偿策略。目前的系统不支持细粒度的策略制定。因此，研发一个能够对超时场景精细化运营配置，提高运营效率的平台很有必要。

为了解决上述问题，本文设计和实现了“饿了么体验保障平台”，提供了规则配置能力，包括配置超时发生的场景，超时发生的条件，生效的城市、订单来源、物流类型，以及命中超时场景后的补偿操作，很好的满足了对超时场景精细化运营的需求，同时提高了运营效率，节省了开发人力。

## 1.2 国内外发展现状

## 1.3 本文主要工作

## 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

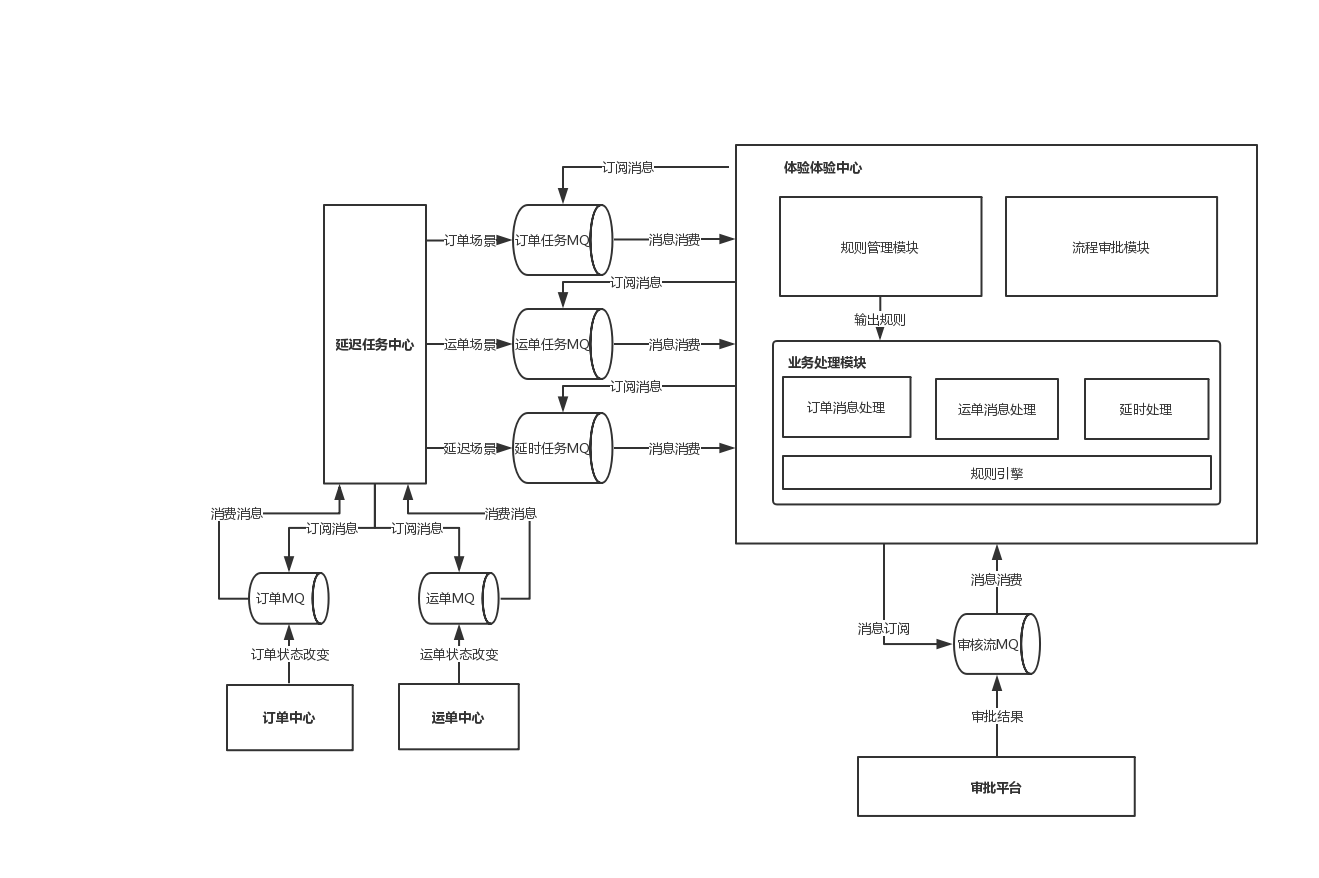
缓存

## 2.6 本章小结

##

# 第三章 系统分析与设计

## 3.1 系统总体规划



## 3.2 系统需求分析

### 3.2.1 规则管理模块

规则管理模块主要用于运营同学进行规则的管理，包括规则的增删改查。



需求:

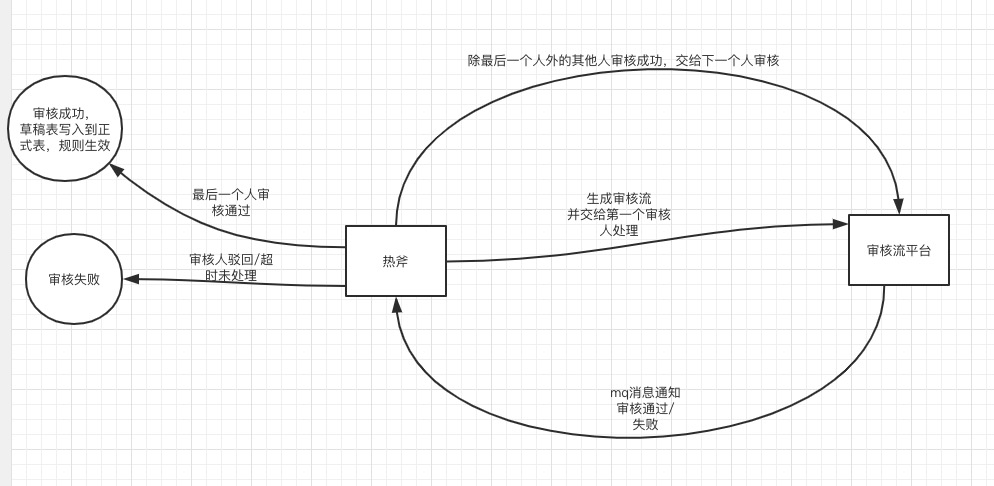
1. 运营新增规则
2. 运营修改规则
3. 运营查看规则
4. 运营删除规则

### 审核流模块

审核流模块负责的任务如下：

1. 运营同学查看审核进度
2. 运营同学取消审核流程
3. 监听审核平台发送的mq消息，并根据mq消息进行审核流程流转
4. 保证mq消息的不丢失，在workflow上配置定时任务，并根据状态进行流程流转。

审核流程图：



需求：

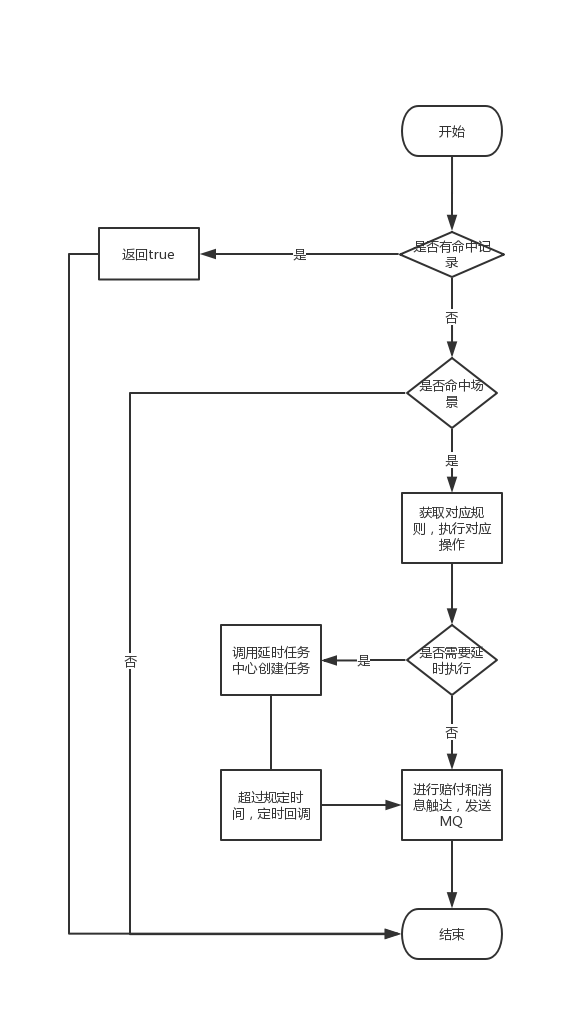
1. 运营新增/更新规则后，需要触发审核流程的开始
2. 运营取消流程审批
3. 监听MQ消息，进行流程的流转
4. 处理Mq消息丢失情况
5. 调用量较大，需要缓存，且需要及时更新缓存

### 业务处理模块

业务处理模块主要负责进行命中场景的判断，命中后的操作。按照目前的业务来说，业务模块需要监听三个消息队列：

* order.message。主要负责订单相关的task
* order.message.action。主要负责配送相关的task
* deliver.order.message。主要负责被延时的task

当监听到消息后，业务模块会分别对他们进行相应的处理。



需求：

1. 处理订单MQ消息
2. 处理配送MQ消息
3. 处理延时任务消息
4. Actions的编写
5. Actions执行失败后的任务兜底
6. 缓存的及时更新

## 3.3 非功能性需求

##

## 3.5 数据库设计

##

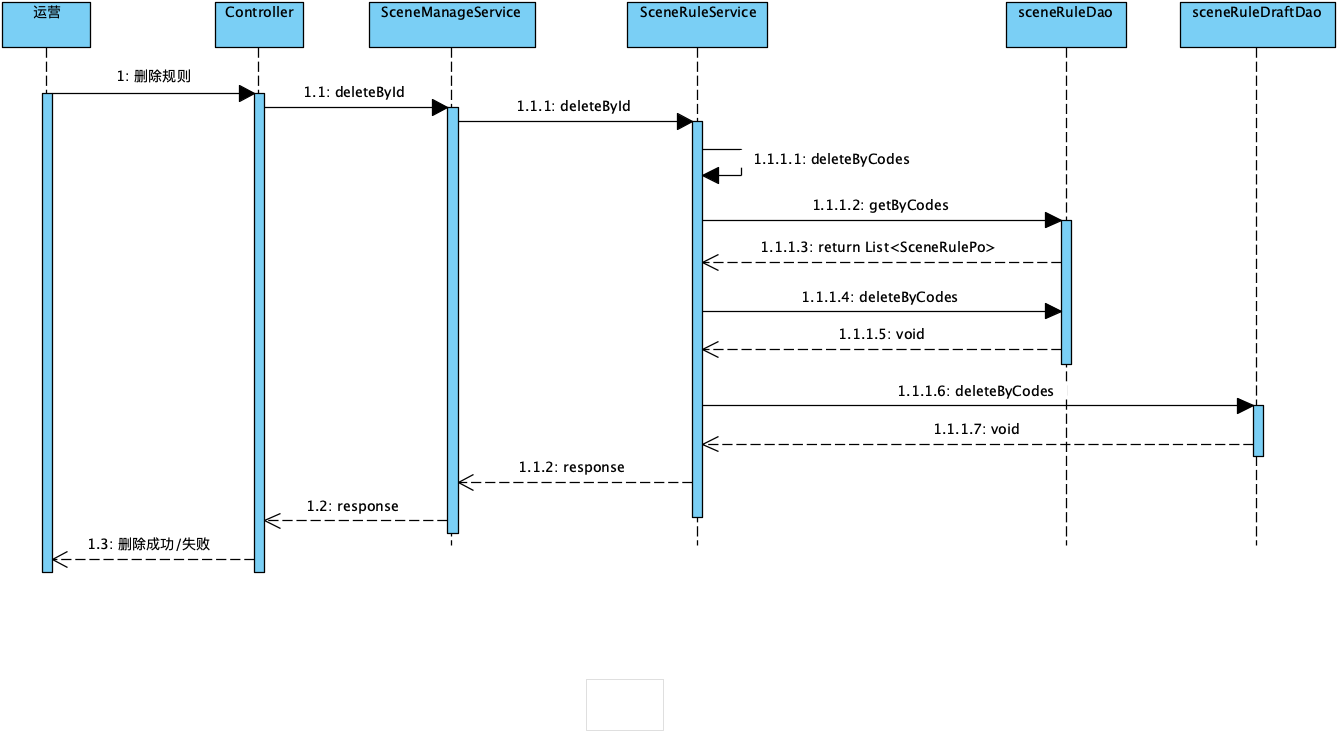
## 3.6 本章小结

##

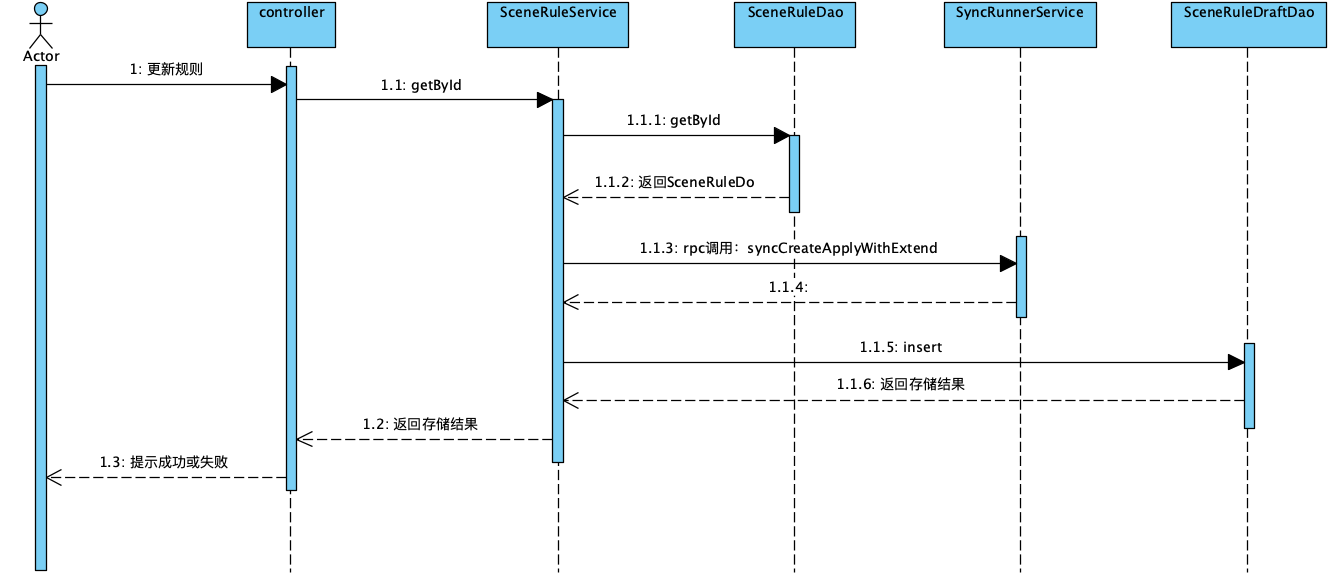
# 第四章 系统实现

## 4.1 规则配置模块

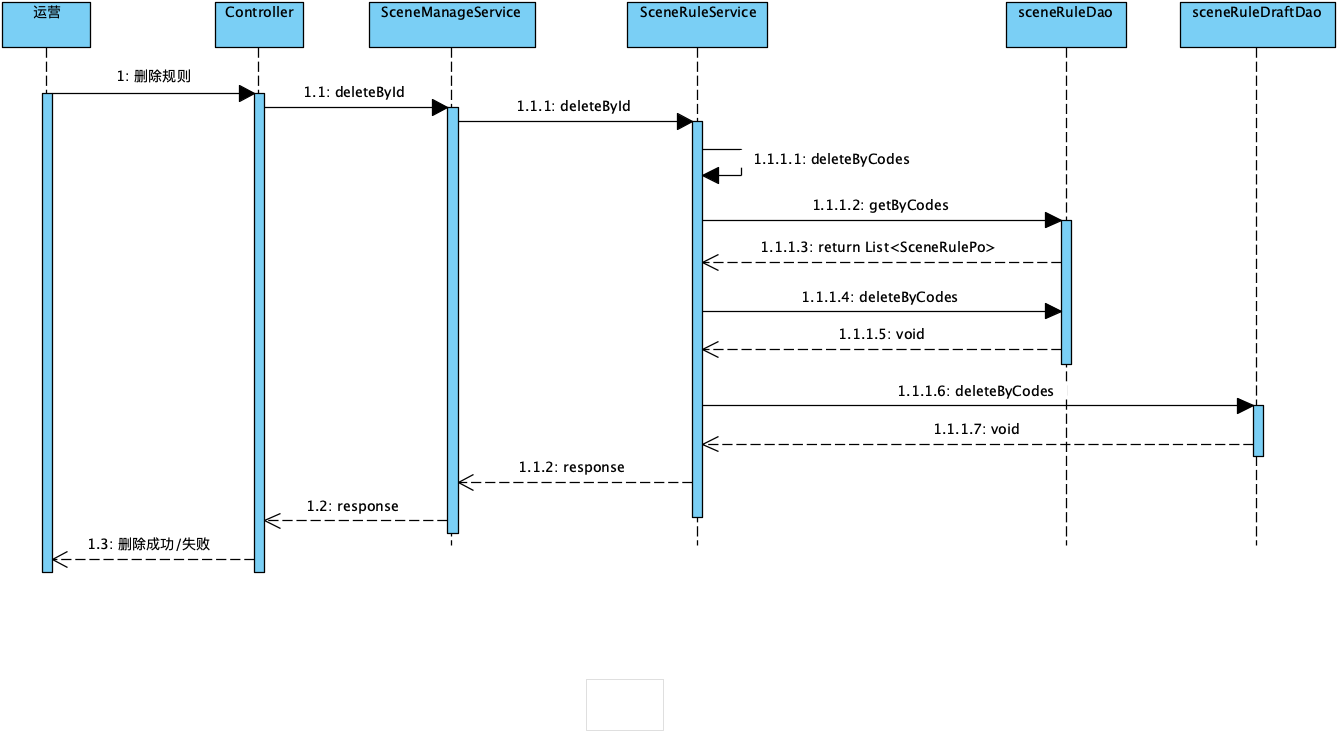
1. 创建规则



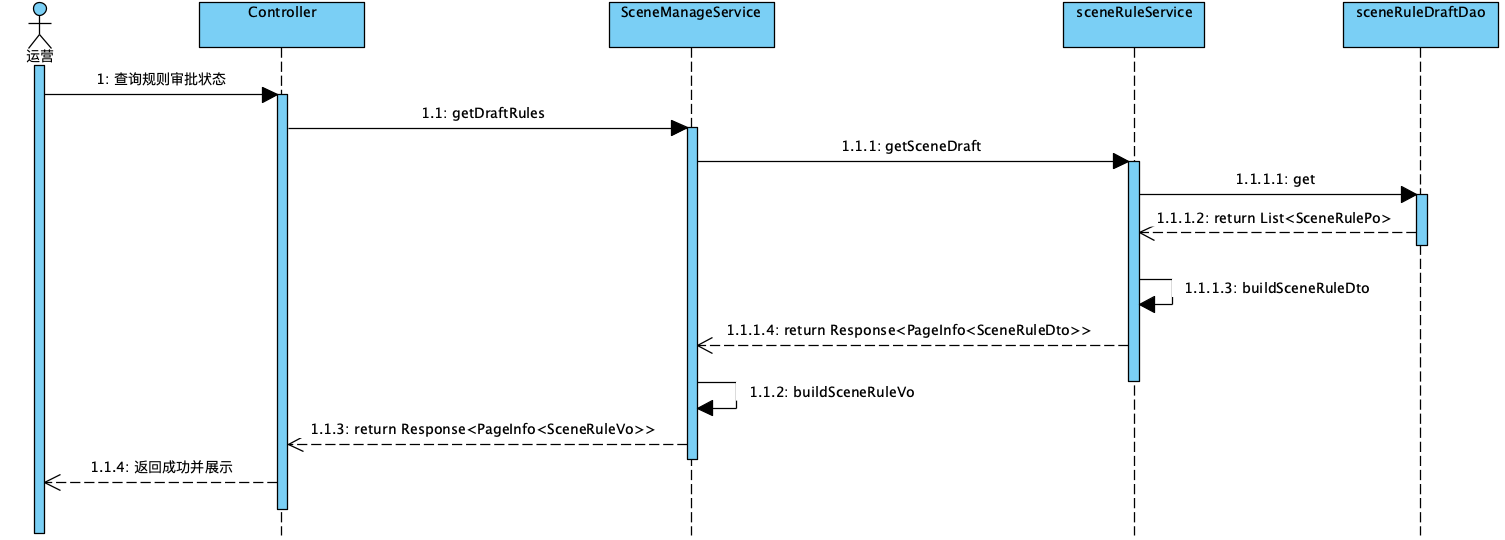
1. 修改规则



1. 删除规则



1. 查找规则



## 4.2 审核流模块

目前审核流已经完成。

## 4.3 业务处理模块

目前业务处理模块已经完成，业务处理模块所涉及到的规则引擎还不太了解，需要深入进行了解。

## 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

##