

《高级实践》 软件设计文档

UkraineBHelper 软件设计文档

文档编号：DOC-RM-220405



学	院	软件学院
专	业	软件工程
年	级	2019

2022年04月05日

目 录

第一章	简介	1
1.1	背景	1
1.2	文档概要	1
1.3	约束	2
1.4	参考资料	3
第二章	架构设计	4
2.1	总体架构	4
第三章	高层设计	7
3.1	模块图	7
3.2	包图	9
第四章	接口说明	18
4.1	Security模块	18
4.2	GateWay模块	19
4.3	Service模块	20
第五章	业务对象及相互关系	21
5.1	概述	21
5.2	业务概念一览	21
5.3	登录	21
5.4	发布房源	21
5.5	修改房源	21
5.6	删除房源	21

5.7	发布新闻	23
5.8	修改新闻	23
5.9	删除新闻	23
5.10	举报	23
5.11	举报审核	30
5.12	系统通知	30
5.13	用户管理	33
5.14	角色管理	33
5.15	权限管理	33
第六章	详细设计	35
6.1	数据库设计	35
6.2	重要流程设计	53
第七章	附录	62
第八章	版本修订记录	63

第一章 简介

战争冲突中，受伤的总是民众。本项目旨在从人道主义出发，搭建帮助难民的互助救助系统。本次主要关注的是俄乌冲突中的受伤民众，他们的安危是俄乌两国在谈判中的议题之一，也收到了国际社会的关注。

1.1 背景

战争的出现是大国之间的博弈和多方利益的考量，在战争中民众的生命安全一直是参战双方的谈判议题，以近期出现的俄乌冲突为例，早在俄乌之间的战争还未爆发的前一周，乌克兰的前一百名富豪们就有96名乘包机逃离了乌克兰，在战时状态，剩下的乌克兰老百姓不仅不被允许离境，还发放枪支给他们，希望他们抵抗。甚至连部分女性，也被迫成为士兵，派往战场。这些人中，很多人都是在两天的时间内才学会射击。

一个逃难去到波兰的乌克兰难民小男孩在被问道，为什么不去选一件为难民孩子们捐赠的玩具，反而一个人在这里孤独悲伤地坐着？他回答说：“小男孩？我现在是家里的大人了”。

在冲突的民众，主要面临以下问题：

- 缺少食物，淡水，住所等基本生存需求。
- 缺少基本应急药品。
- 与临近国家语言不通。
- 无法与家人团聚。

解决这些问题也是我们开发设计这套系统首要关注的内容。

1.2 文档概要

该文档是针对该难民救助系统的设计文档，包括有软件架构设计、高层设计、接口说明、业务对象及相互关系、详细设计。

- 软件架构设计

本系统采用了微服务架构，将业务拆分为了多个子系统独立部署，同时结合当前流行的微服务技术如网关、注册中心等。

- 高层设计

本系统采用传统MVC三层模式，将各个子模块分为视图层、业务层和持久层，其中视图层负责和前端进行数据交互，业务层负责核心逻辑处理，持久层负责和数据库进行数据交互。同时该系统还拆分为多个子系统，

以达到松耦合、易扩展的目的。

- 接口说明

将对系统中用到的interface和abstract class进行详细说明，包括他们的作用、public方法的用途以及参数的说明。

- 业务对象及相互关系

将对本系统涉及到的业务对象，如用户类、权限类、新闻类、帖子类等，进行详细的说明，并对其相互关系进行设计。

- 详细设计

将对本系统的数据库、数据表关系以及各个模块的业务流程进行详细设计。

1.3 约束

- 相关需要有身份认证以保证安全性。
- 系统无前置的软硬件基础。

1.4 参考资料

表 1-1 参考资料

资料名称	版本日期	说明
UML大战需求分析	2012年2月第一版	无
待补充		

第二章 架构设计

2.1 总体架构

该系统采用的是微服务架构，即将业务拆分为多个子业务独立部署，各个子业务之间彼此独立，互相协作。同时整合网关、注册中心、负载均衡等当下流行的微服务技术，使得各个服务之间的协作更加有合理、高效。同时为了应对集群部署，也为了提高处理速度，本系统采用Redis缓存技术，以便部分业务的处理。总体的架构图如图 2-1 所示。

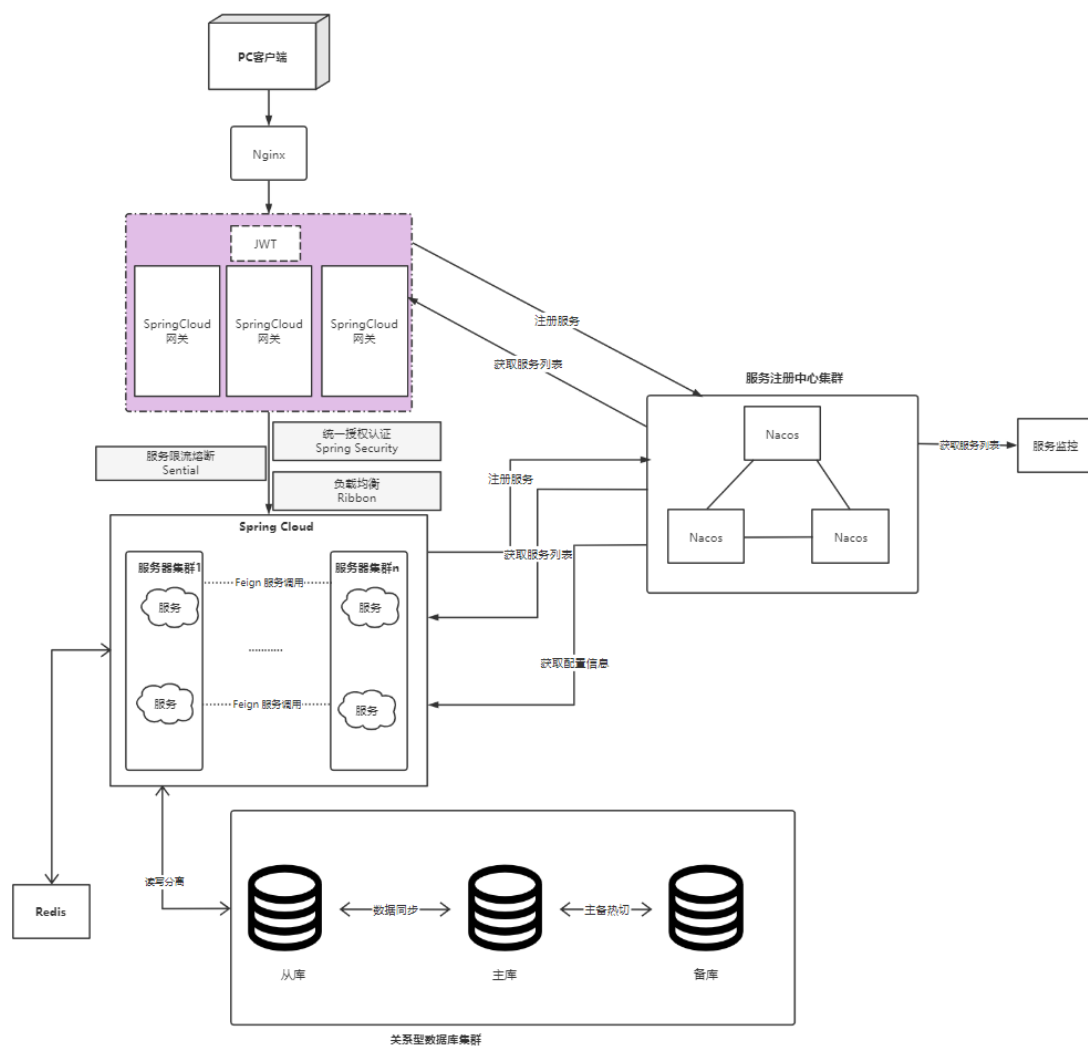


图 2-1 总体架构

2.1.1 具体微服务模块

具体的微服务模块即系统给用户提供的业务服务，包括注册认证、发帖浏览等，每个具体微服务都整合了SpringCloud框架，微服务之间通过OpenFeign进行相互调用。各个微服务集群部署，负载均衡技术采用ribbon，负载均衡策略为轮询。

2.1.2 中间件

本系统采用了许多微服务的中间件，以便更好地为微服务框架服务。

2.1.2.1 服务注册与发现

服务注册采用的是阿里巴巴开发的Nacos，部署方法为3个Nacos搭建服务注册集群。该集群可以实现各个微服务的注册，服务列表的获取和微服务的监听。

2.1.2.2 服务调用

本系统的服务调用技术采用的是OpenFeign，其为HTTP形式的REST API提供了简洁高效的RPC调用方式，同时OpenFeign也整合了负载均衡功能，便于集群之间的调用。

2.1.2.3 负载均衡

负载均衡技术使用的是Ribbon，结合OpenFeign实现微服务集群之间的互相调用，鉴于本系统的流量并不大，因此负载均衡策略采用轮询即可。

2.1.2.4 网关

网关技术使用的是Spring Gateway，他为众多服务模块提供了一个统一的访问路径，同时负责分发路由和认证授权，是所有服务调用的第一步。

2.1.2.5 配置中心

配置中心使用Nacos的配置中心，和Nacos服务注册中心紧密结合，将项目的配置文件进行统一的管理，同时也将项目的开发环境进行有效的隔离。

2.1.2.6 服务降级和服务熔断

本系统采用Sentinel技术实现服务降级和服务熔断，在流量超出服务器上限，或是出错率超过阈值时，会返回特定页面，以给客户更好的服务体验。

2.1.2.7 消息队列

本系统采用RabbitMQ实现消息队列，实现应用之间的异步调用，使得一些不需要及时反馈的操作可以异步进行，从而加快响应速度，提高用户体验。

2.1.3 存储与缓存

2.1.3.1 数据库

系统采用的为MySQL数据库，其数据模型的具体请详见后面的详细设计，

会将每张数据表的字段和约束具体描述。

2.1.3.2 缓存

系统采用的是Redis缓存，其是基于内存的非关系型数据库，用于为本系统提升查询速度，主要在认证授权模块等高频模块使用。

第三章 高层设计

通过全局，虽然我们对该系统采用的是微服务架构，但是纵向来说，对于某个具体的模块，我们依然采用传统的MVC模式，即区分视图层、业务层和持久层。视图层负责和前端进行数据交互，业务层负责业务逻辑处理，持久层负责和数据库进行数据交互，三层各司其职，实现系统的可控、可扩展。本章针对需求规格说明书提出的需求进行提炼，给出本系统的层次结构，包括包图、部署图等设计。

3.1 模块图

系统中我们分为了许多Modules，主要为4个模块，Common、Service、GateWay、NacosCore，其作用分别是通用工具模块，微服务业务模块，网管模块和服务注册中心模块，下面图 3-1 给出了系统的总体模块图。

其中四个模块还含有不同的子模块，我们对于四个模块的作用及其子模块做一个大致说明。

- Common通用工具模块，将其他模块通用的功能抽象成一个独立的模块，便于调用。
 - Base模块，包含了一些基本配置，比如Redis缓存的配置，返回类的配置等等。
 - Security模块，整合了SpringSecurity框架，用于对用户进行认证授权。
- Service微服务业务模块，即负责系统实际业务的模块，将所有业务统一到一个父类模块下，便于统一配置。
 - UserService模块，负责处理用户信息的模块，包括用户登录、用户注册、管理获取用户列表等。
 - HouseService模块，负责处理房源帖子的信息，包括发帖、改贴、帖子列表等。
 - ReportService模块，负责举报信息，包括发起举报，审核举报等。
 - NewsService模块，负责管理新闻信息，包括编辑新闻，获取新闻列表等。
 - AuditService模块，负责管理审核信息，包括举报审核、新闻审核等。
 - SystemService模块，负责处理系统的消息和系统业务对象，用于管理员给用户群发通知，业务对象管理等。
- OpenFeign服务调用模块，负责各个微服务之间进行参数封装和方法调

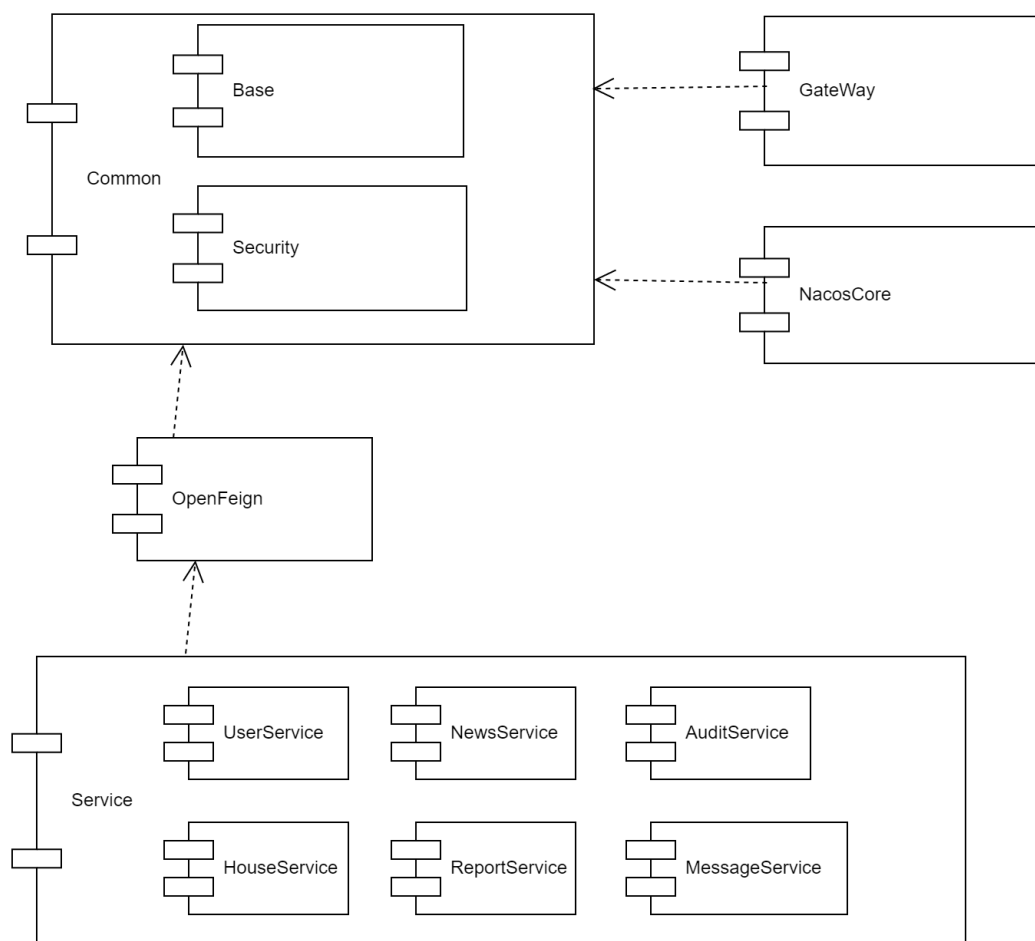


图 3-1 模块图

用。

- GateWay网管模块，负责分发路由，将请求转发到对应的微服务上，同时也负责初步校验JWT，检验用户身份。
- NacosCore微服务注册中心模块，负责管理系统中的各项微服务，包括服务注册、服务发现、流量监控等。

3.2 包图

接下来我们来设计各个模块中的包结构，不同功能的模块有着不同的包结构。大体来说，业务模块主要以MVC模式为主，即不同的层次对应不同的包，而配置相关的模块则会根据配置类别进行不同的分包。

3.2.1 Common通用工具模块

Common模块中包含两个子模块，为Base模块和Security模块。

3.2.1.1 Base

Base中主要包含两个包，分别是Config和Util，分别负责一些全局参数的配置。Config中包含了Redis、RestTemplate、Constant(全局常量)的配置。Util中包含了ResponseUtil(统一返回设置)、Result(统一返回类)等全局统一用的配置。Base模块的总体包图如图 3-2 所示。

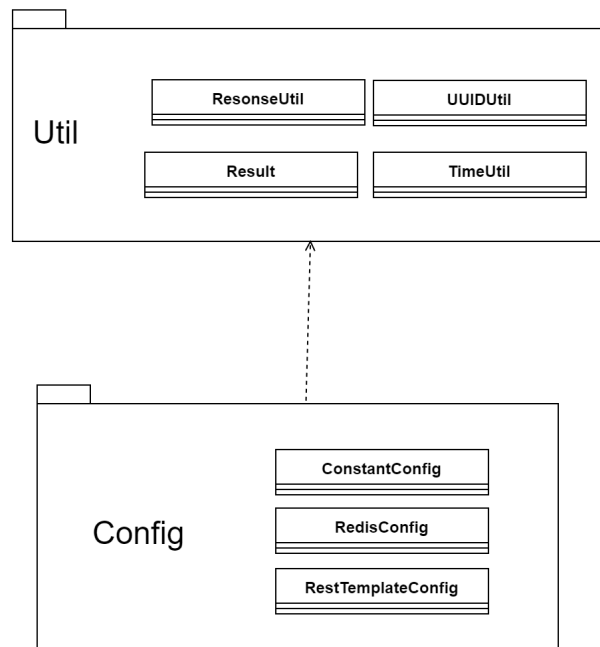


图 3-2 Base模块包图

3.2.1.2 Security

Security中主要包含4个包，为Config、Filter、Entity、Security。其中，Config为安全框架的具体配置，Filter中自定了一些过滤器，Entity中定义了一些实体类，Security中定义了具体的认证授权操作类。Security模块的总体包图如图 3-3 所示。

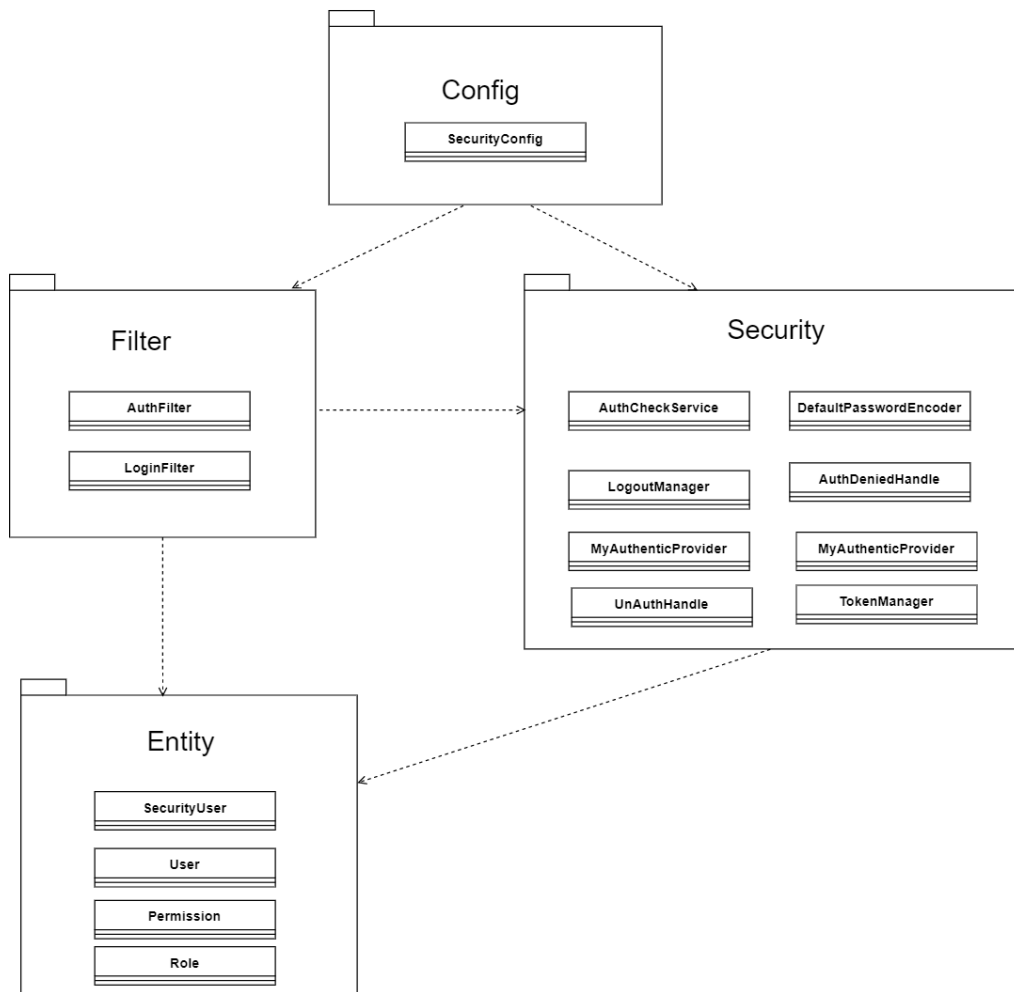


图 3-3 Security模块包图

3.2.2 OpenFeign服务调用模块

OpenFeign服务调用模块主要包含2个包，为FormGeneration和Client。其中FormGeneration负责封装不同模块的参数，Client负责声明各个模块暴露的接口。

3.2.3 Service微服务业务模块

Service模块包含6个子模块，为UserService模块、HouseService模块、NewsService模块、ReportService模块、AuditService模块、SystemService模块。

3.2.3.1 UserService

UserService模块主要包含6个包，主要是MVC模式下的几个包，包括Controller、Service、Impl、Dao、Entity、Vo。前5个包的功能在前文均有描述，而Vo包的主要功能是统一该模块的前端传来的数据模型，比如UserForm，以及后端返回给前端的数据模型，比如UserVo。UserService模块的总体包图如图 3-4 所示。

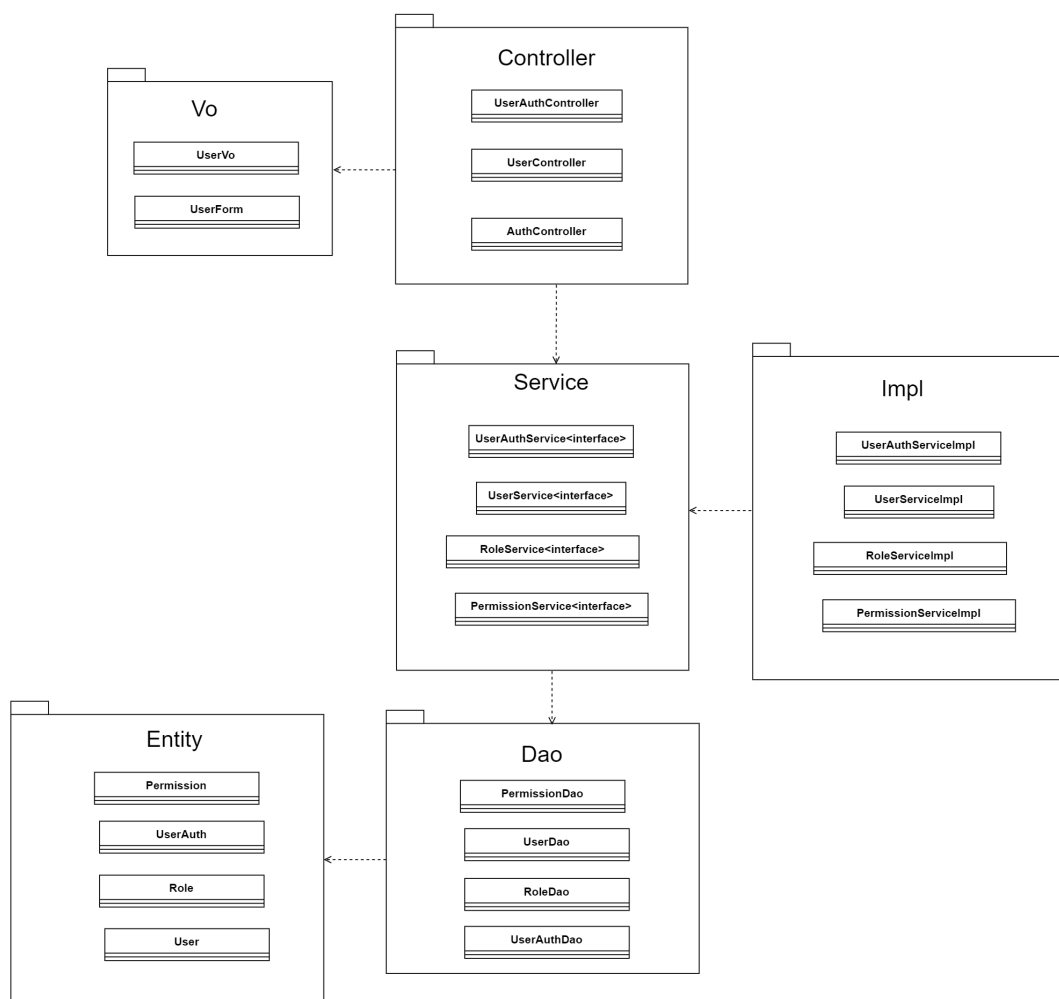


图 3-4 UserService模块包图

3.2.3.2 HouseService

HouseService模块和上文一样，也包含6个包。总体包图如图所示。

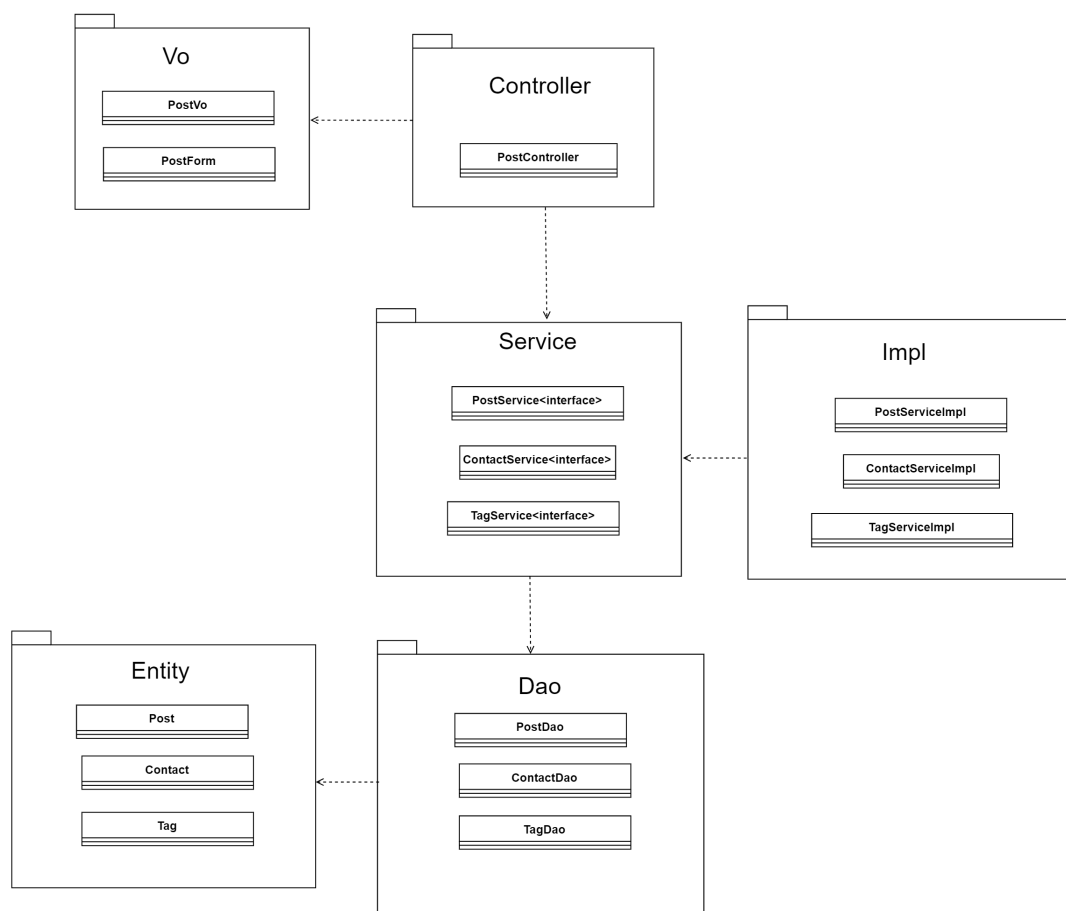


图 3-5 HouseService模块包图

3.2.3.3 NewsService

包含6个包，总体包图如图 3-6 所示。

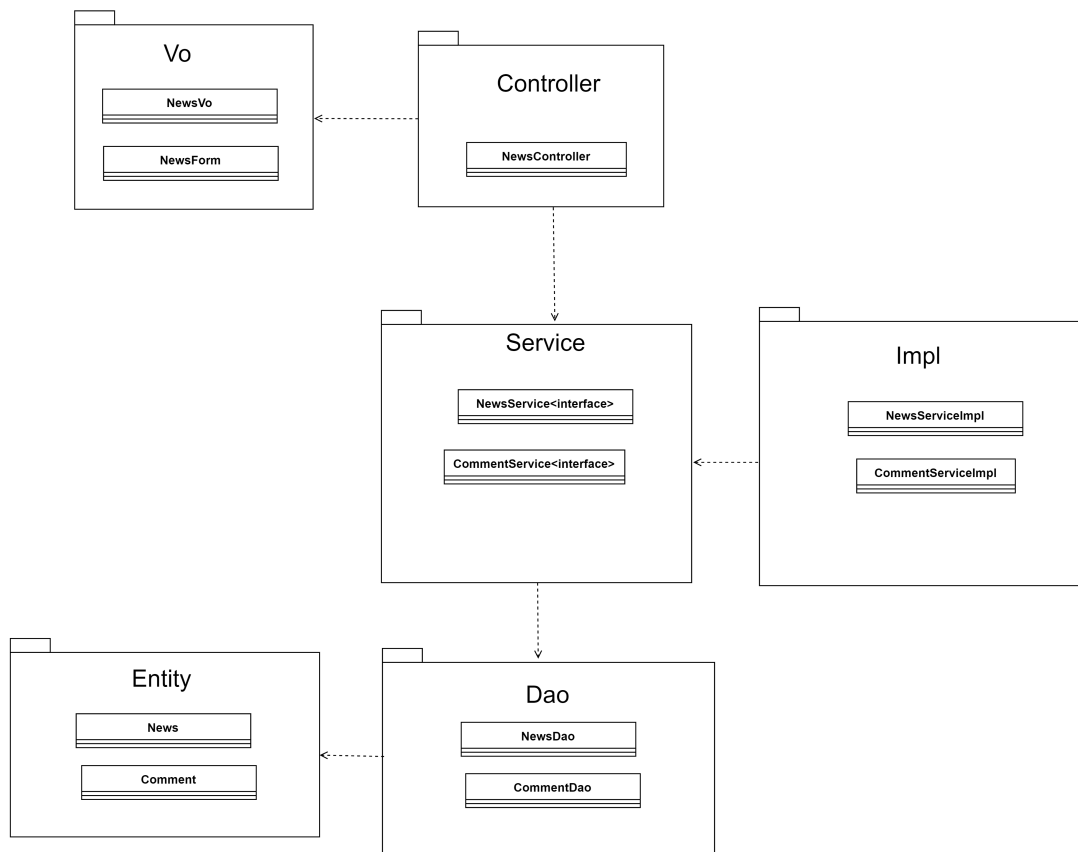


图 3-6 NewsService模块包图

3.2.3.4 ReportService

包含6个包，总体包图如图 3-7 所示。

3.2.3.5 AuditService

包含6个包，总体包图如图 5-10 所示。

3.2.3.6 SystemService

总体包图如图 3-9 所示。

3.2.4 GateWay网关模块

GateWay网关模块负责路由转发，同时还负责初步的用户权限检查。所谓的网关权限检查，即对于用户的登陆凭证token进行初步检验。由于路由转发功能在配置文件中已经实现，需要自定义的权限检查只需要一个包即可。下面为GateWay网关模块的包图 3-10。

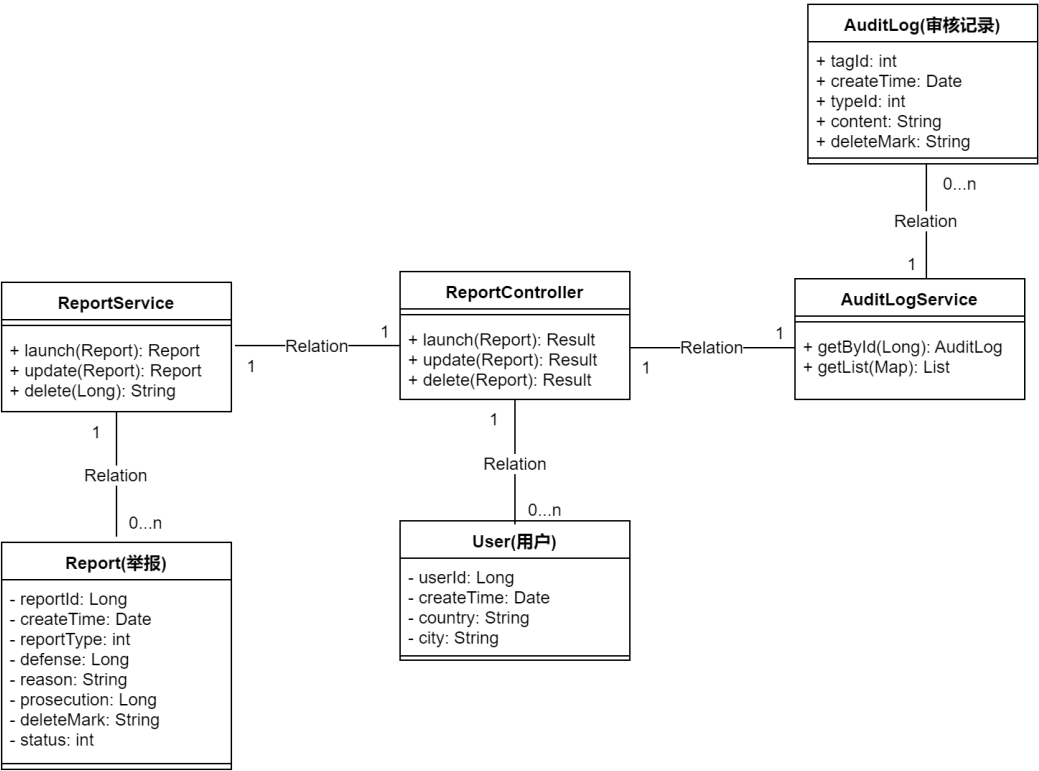


图 3-7 ReportService模块包图

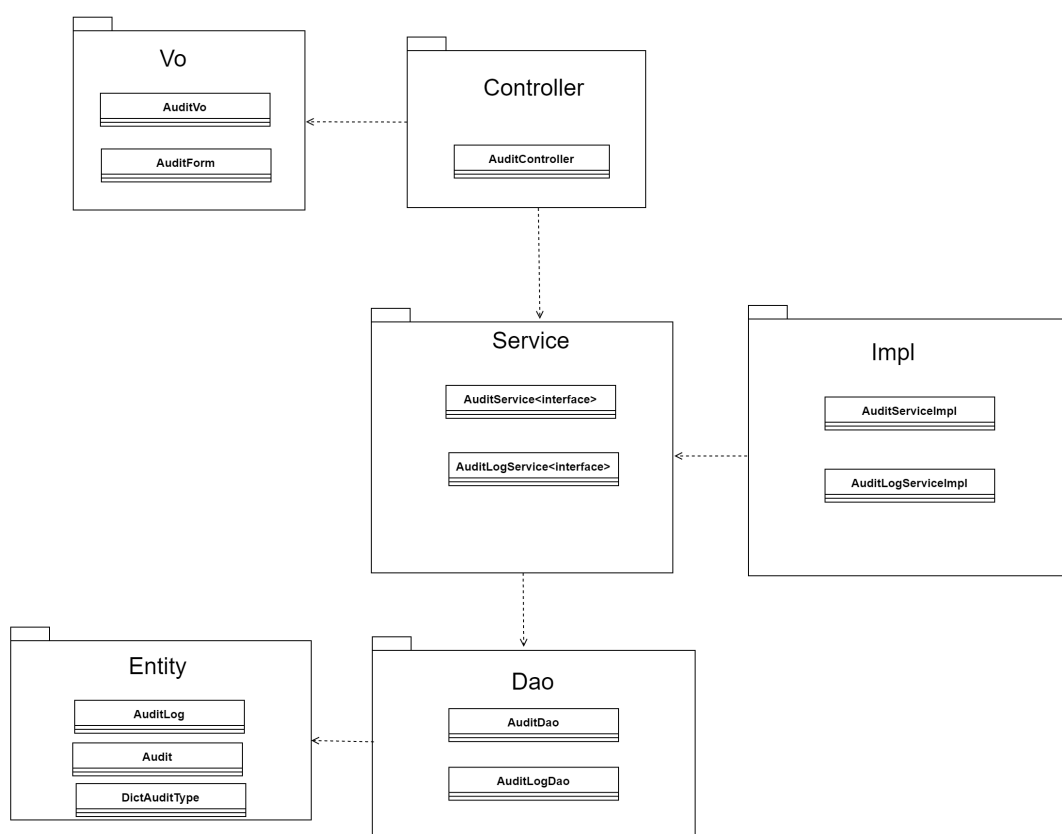


图 3-8 AuditService 模块包图

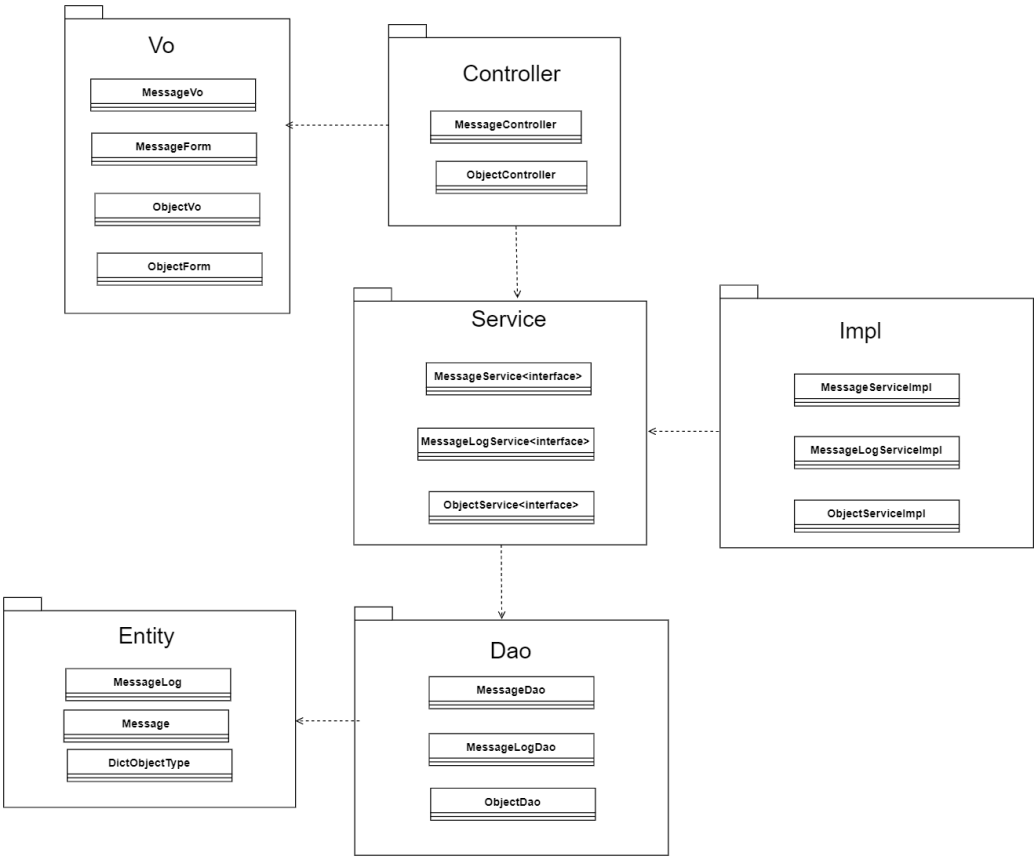


图 3-9 SystemService模块包图

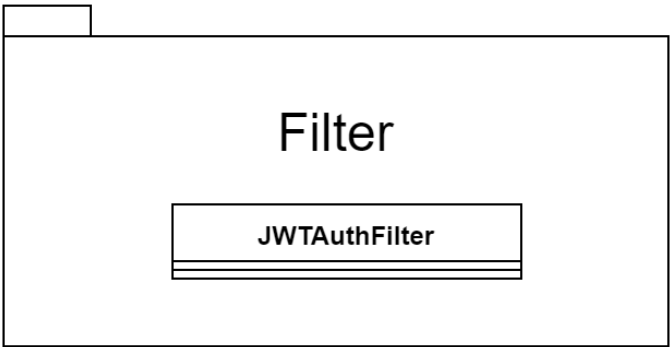


图 3-10 GateWay模块包图

3.2.5 系统部署图

本系统的部署主要涉及到服务端、客户端和数据库。客户端主要为Web浏览器，服务端需要部署多个微服务，同时启动Redis服务，数据库使用MySQL数据库。系统部署图如图 3-11 所示。

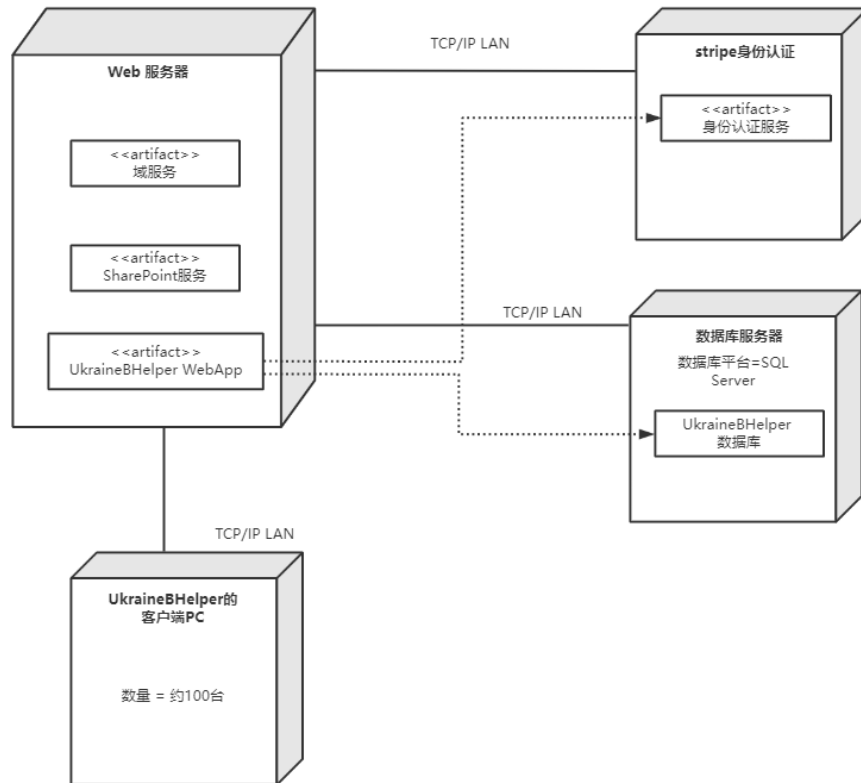


图 3-11 系统部署图

第四章 接口说明

本章将对系统中的应用的接口，包括interface，abstract class进行详细说明，主要在于说明他们的public方法，以及方法中参数的意义和返回值的意义。

4.1 Security模块

为了便于扩展不同的认证授权方案，Security模块定义了许多认证授权操作的接口，以及相关过滤器的抽象类，包括登录过滤器、授权过滤器、用户加载接口等。

4.1.1 UsernamePasswordAuthenticationFilter

此抽象类为登录时调用的过滤器类。类中共定义了3个主要方法，作用分别为用户认证、认证成功和认证失败。

- attemptAuthentication方法，返回类型为Authentication(用户认证类)，参数有2个，类型分别为HttpServletRequest、HttpServletResponse。方法用途为将request中的用户信息进行校验。
- successfulAuthentication方法，返回类型为void，参数有4个，类型分别为HttpServletRequest、HttpServletResponse、Authentication、FilterChain(过滤链)。方法用途为用户认证成功后进行相关操作，包括加载权限等，然后将用户认证信息通过过滤链传递给下个过滤器。
- unsuccessfulAuthentication方法，返回类型为void，参数有3个，类型分别为HttpServletRequest、HttpServletResponse、Authentication。方法用途为用户认证失败后进行相关操作，包括返回失败信息等。

4.1.2 BasicAuthenticationFilter

此抽象类为授权时调用的过滤器类，类中需要说明的方法主要有1个，作用是获取用户权限，然后交给鉴权过滤器去鉴权。

- doFilterInternal方法，返回类型为void，参数有3个，类型分别为HttpServletRequest、HttpServletResponse、FilterChain。方法用途为将request中的用户进行权限加载，然后把权限信息交给鉴权过滤器去鉴权。

4.1.3 AccessDeniedHandler

此接口定义了用户权限不够时所调用方法。主要的方法有1个，作用是设置用户权限不足的信息，然后将其返回给用户。

- handle方法，返回类型为void，参数有3个，类型分别为

为`HttpServletRequest`、`HttpServletResponse`、`AccessDeniedException`(权限异常类)。方法用途为在`HttpServletResponse`中设置用户权限不足的信息并返回给用户。

4.1.4 AuthenticationEntryPoint

此接口定义了用户认证失败时所调用方法。主要的方法有1个，作用是设置用户认证失败的信息，然后将其返回给用户。

- `commence`方法，返回类型为`void`，参数有3个，类型分别为`HttpServletRequest`、`HttpServletResponse`、`AuthenticationException`(认证异常类)。方法用途为在`HttpServletResponse`中设置用户认证失败的信息并返回给用户。

4.1.5 PasswordEncoder

此接口定义了用户密码加密和匹配的策略，主要有2个方法，作用分别是用户密码的加密算法和用户密码的匹配算法。

- `encode`方法，返回类型为`String`，参数有1个，类型为`CharSequence`(用户传来的原始密码)，作用在于对用户输入的密码进行加密计算，至于加密密码可以自定义，比如md5加密等。
- `matches`方法，返回类型为`Boolean`，参数有2个，类型都是`CharSequence`，分别表示用户输入的密码和数据库查询的密码，然后通过自定义的匹配算法进行匹配。

4.1.6 UserDetailsService

此接口定义了用户详细信息的加载方法，主要是1个方法，作用为从数据库加载用户的基本信息、角色信息和权限信息，并进行用户登录初始化工作。

- `loadUserByUsername`方法，返回类型为`UserDetail`，参数有1个，类型分别为`String`，表示登录用户的用户名。方法用途为根据传入的用户名从数据库加载用户基本信息、角色信息和权限信息，并进行初始化工作，比如权限信息存入Redis缓存等。

4.2 GateWay模块

因为GateWay网关是需要处理客户端发来的一切请求，我们需要自定义一些过滤器。为了可扩展性，需要一个全局的过滤器类，实现过滤器的扩展。

4.2.1 GlobalFilter

此接口定义了过滤器实际过滤采用的方法，主要方法有1个，作用是接收并预处理客户端发来的一切请求。

- `filter`方法，返回类型为`Mono<void>`，参数有2个，类型分别

为ServerWebExchange(客户端请求)、 GatewayFilterChain(网关过滤器链)。方法用途为将客户端请求进行预处理，如果符合要求则放行，否则返回错误信息。

4.3 Service模块

Service是业务模块，用的是MVC的框架，其中将每个具体的业务方法都进行了interface+impl的组合，即每个业务方法都被抽象成了一个接口，这样的话我们需要进行扩展就只需要新建impl即可，增加了系统的可扩展性。

第五章 业务对象及相互关系

5.1 概述

本系统的主要业务模块分为房源管理，新闻管理，用户权限管理，系统管理，技术模块分为前端技术和后端技术，技术栈为：

- 后端技术： SpringCloud + SpringSecurity + Nacos + OpenFeign + Spring Gateway + Mybatis-plus + MySQL
- 前端技术：

5.2 业务概念一览

本系统的模块可以详细分为用户、角色、权限、房源、新闻、信息、举报、审核、日志、通讯这十个模块。下图为这几个模块的关系类图。

5.3 登录

登录是个较为复杂的模块，因为他需要认证用户信息，生成用户登录凭证，还需要将用户的权限信息存入缓存中以便后续的鉴权工作。

登录模块使用了SpringSecurity架构，其中我们自定义了登陆用户信息，还有一些相关过滤器，同时我们使用了SpEL语言在AuthCheckService中重定义了系统的鉴权规则，使得系统动态匹配用户权限信息。同时用户密码的加密，以及登陆凭证的生成都是根据项目实际情况自行定义的，其类图关系如下。

5.4 发布房源

房源发布主要为屋主的操作，该模块要求屋主填写填写房源基本信息，包括房源地址，容纳人数，是否允许宠物等，还需要屋主留下联系方式，难民可以根据联系方式自行去联系屋主。涉及到的实体类有用户类（User），帖子类（Post），联系方式（Contact），标签（Tag）。具体的方法都在PostController中进行调用。

5.5 修改房源

屋主可以查看自己发布的房源，然后进行修改，比如添加删除标签，修改联系方式等等，具体的方法还是在PostController中调用。

5.6 删除房源

屋主可以查看自己发布的房源，然后进行删除操作，就可以将发布过的房源不再显示。

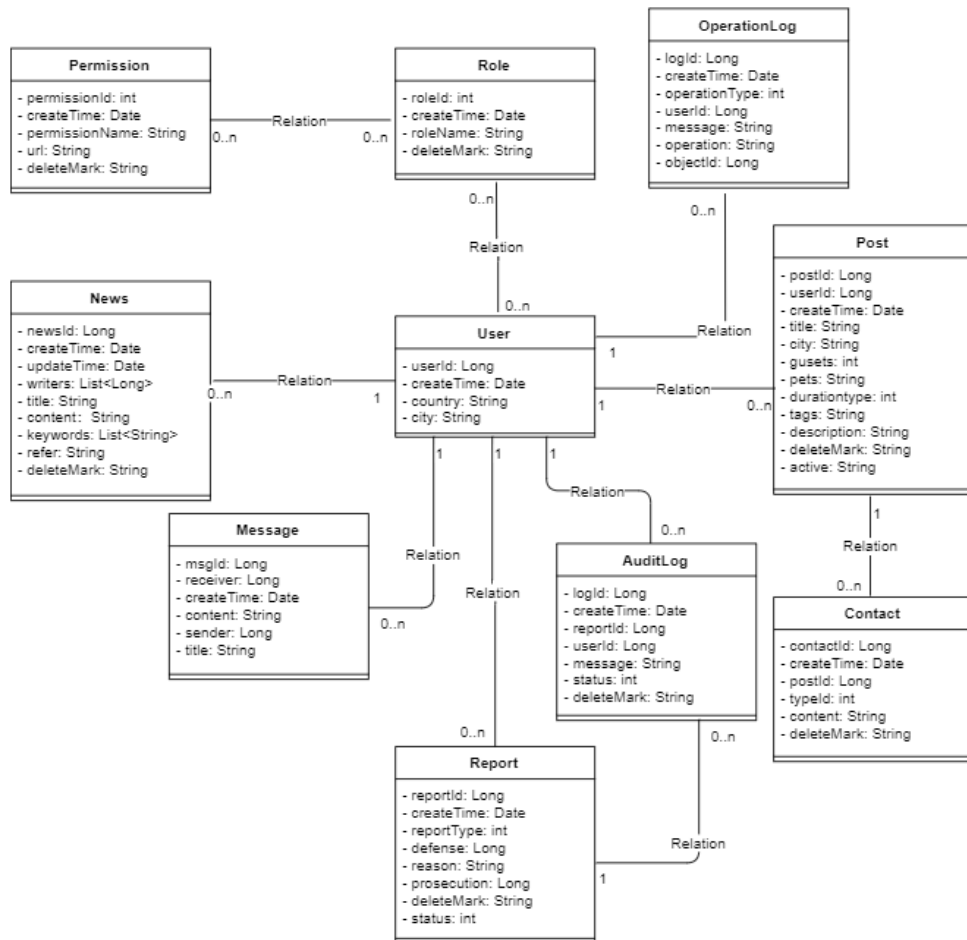


图 5-1 总类图

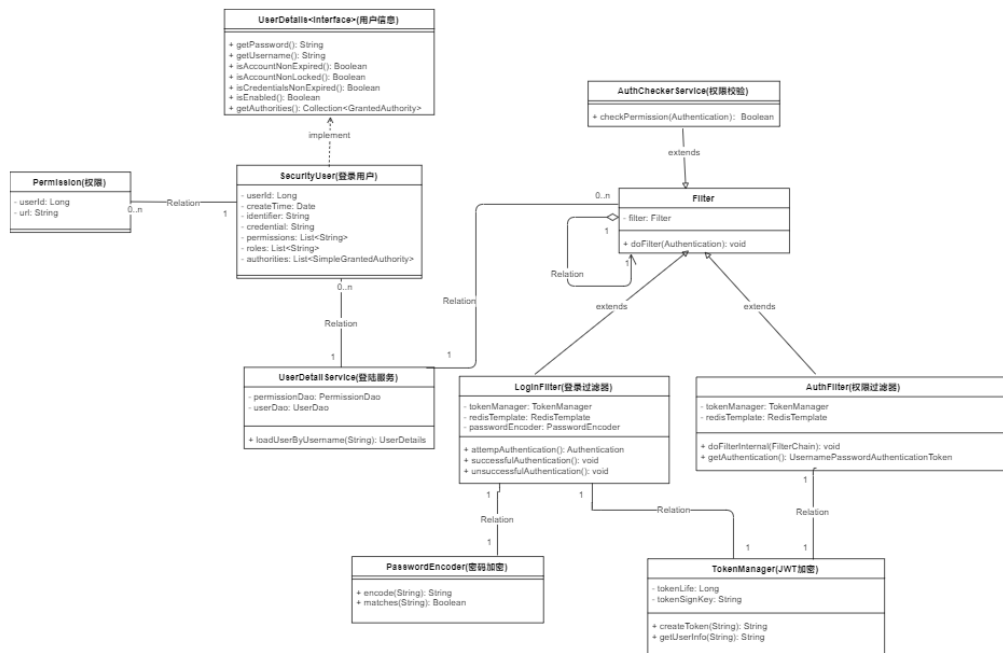


图 5-2 登录

5.7 发布新闻

发布新闻主要为编辑者的操作，该模块要求编辑者填写新闻的详细信息，包括新闻标题，新闻内容，关键词等，如果是引用的别的网站的新闻则需要注明引用链接，新闻模块是所有用户均能观看。新闻需要进行审核才能发布。涉及到的实体类有用户类（User），新闻类（News），审核类（AuditLog）。具体方法都在NewsController中进行调用。

5.8 修改新闻

编辑者可以查看自己发布的新闻，然后进行修改，比如修改内容，修改标题等等，修改后需要审核才能发布。具体的方法还是在NewsController中调用。

5.9 删除新闻

管理员可以查看发布的新闻，然后删除操作，具体的方法是在NewsController中调用。

5.10 举报

所有用户都可以对新闻或者房源进行举报，举报需要填写举报理由，然后会由管理员进行审核，管理员会和被举报的用户进行交涉，若举报信息属实则

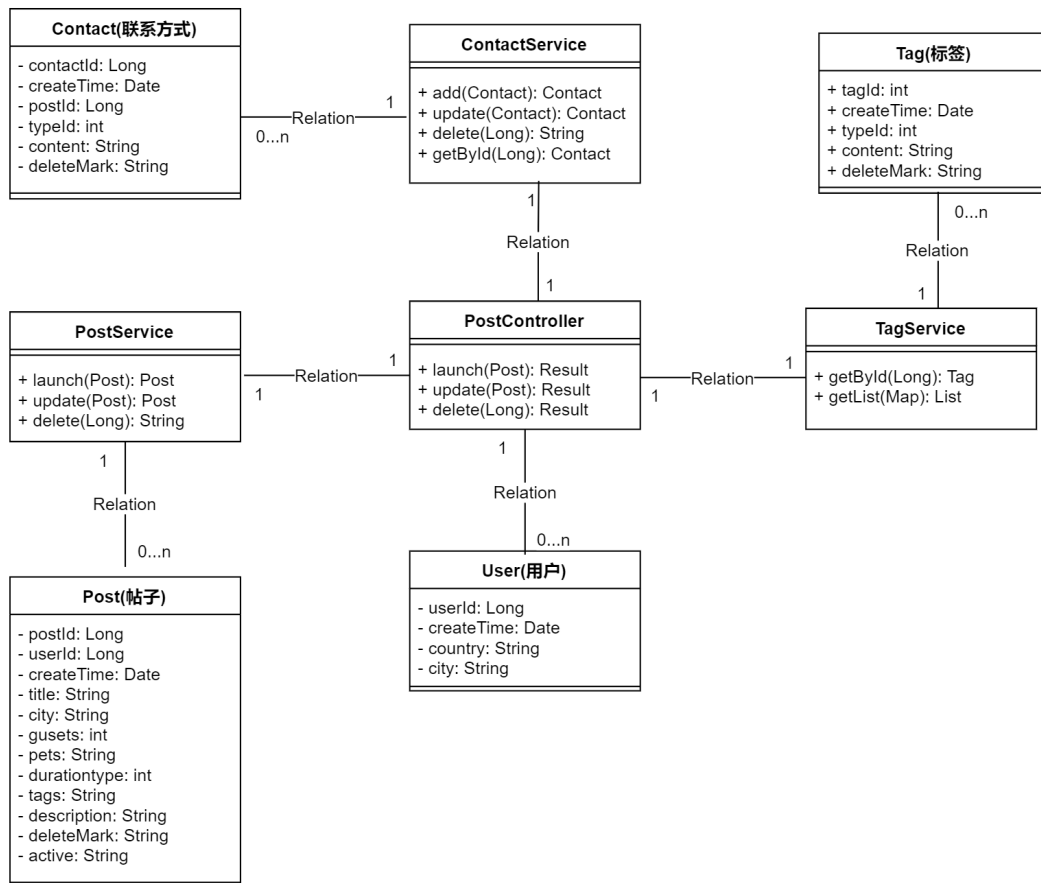


图 5-3 发布房源

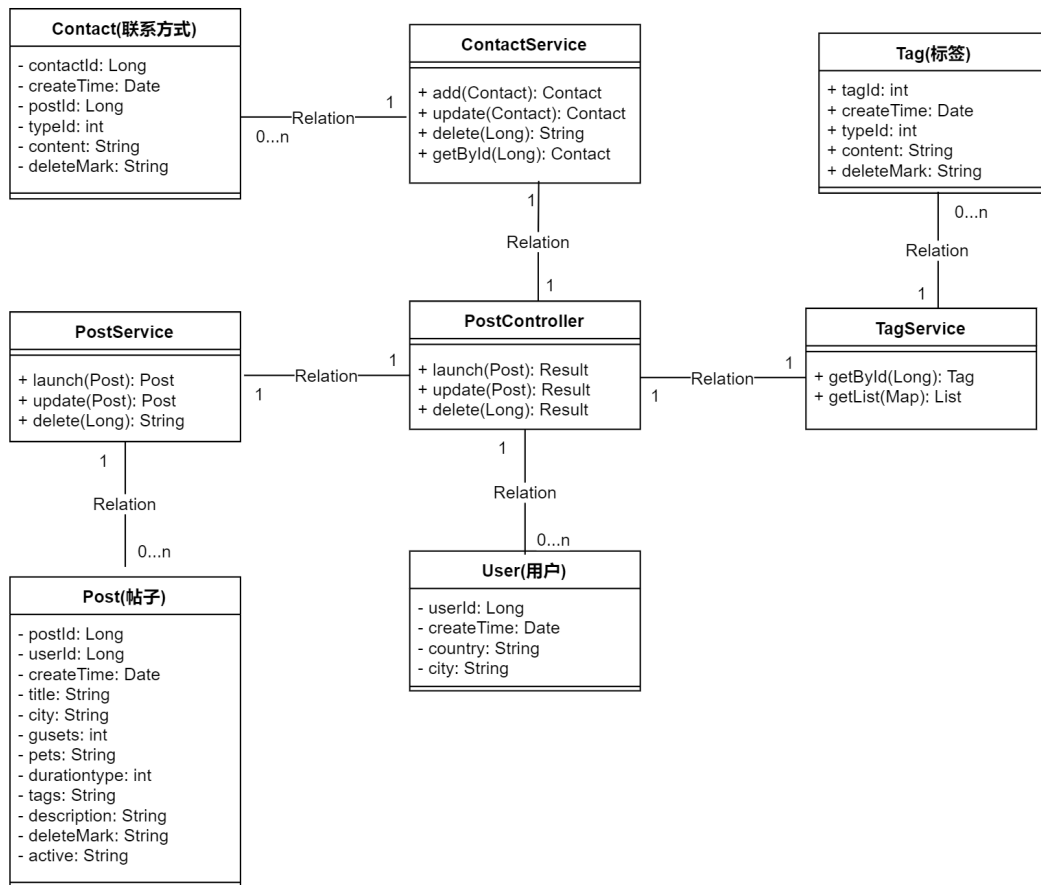


图 5-4 修改房源

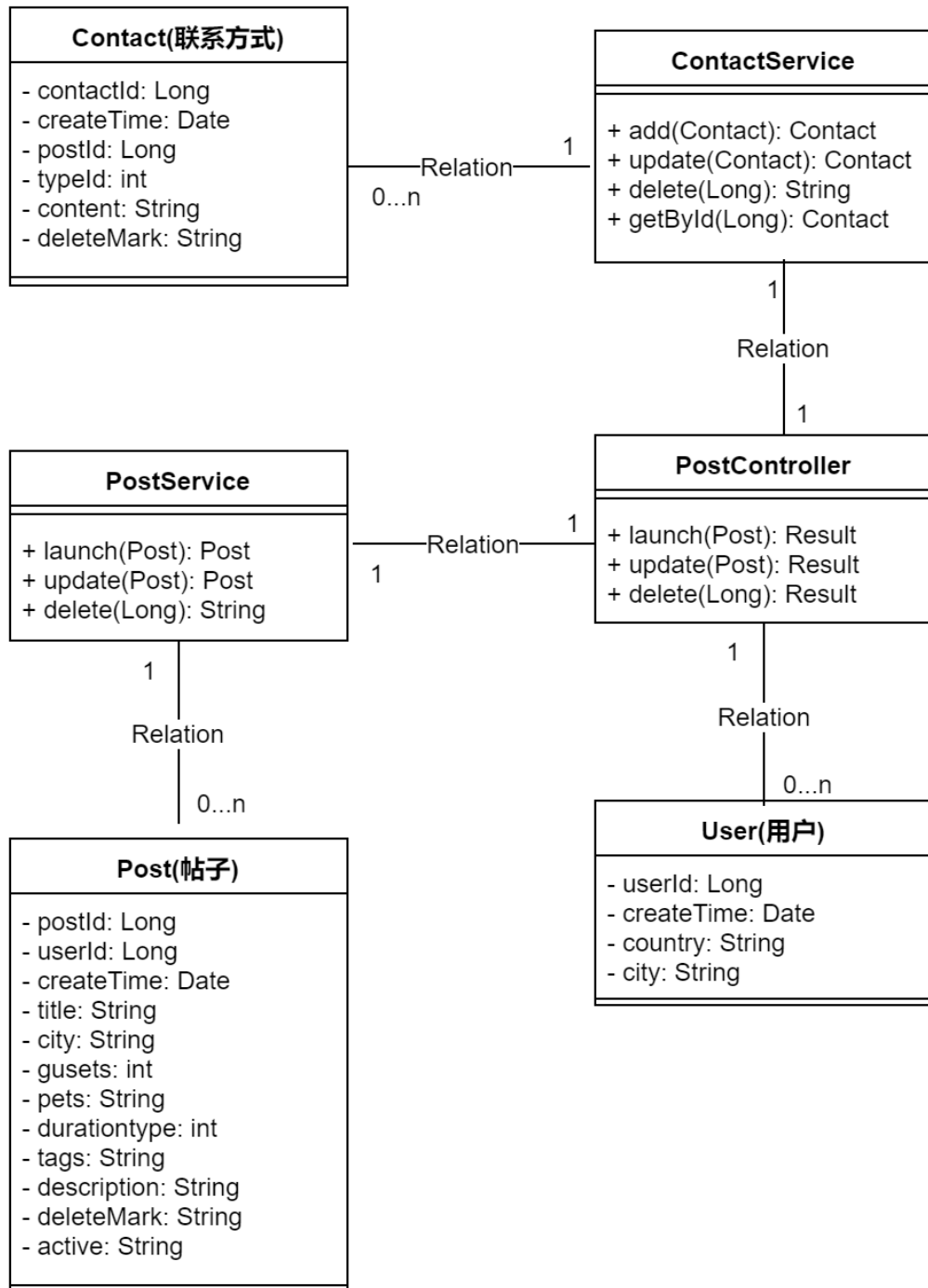


图 5-5 删除房源

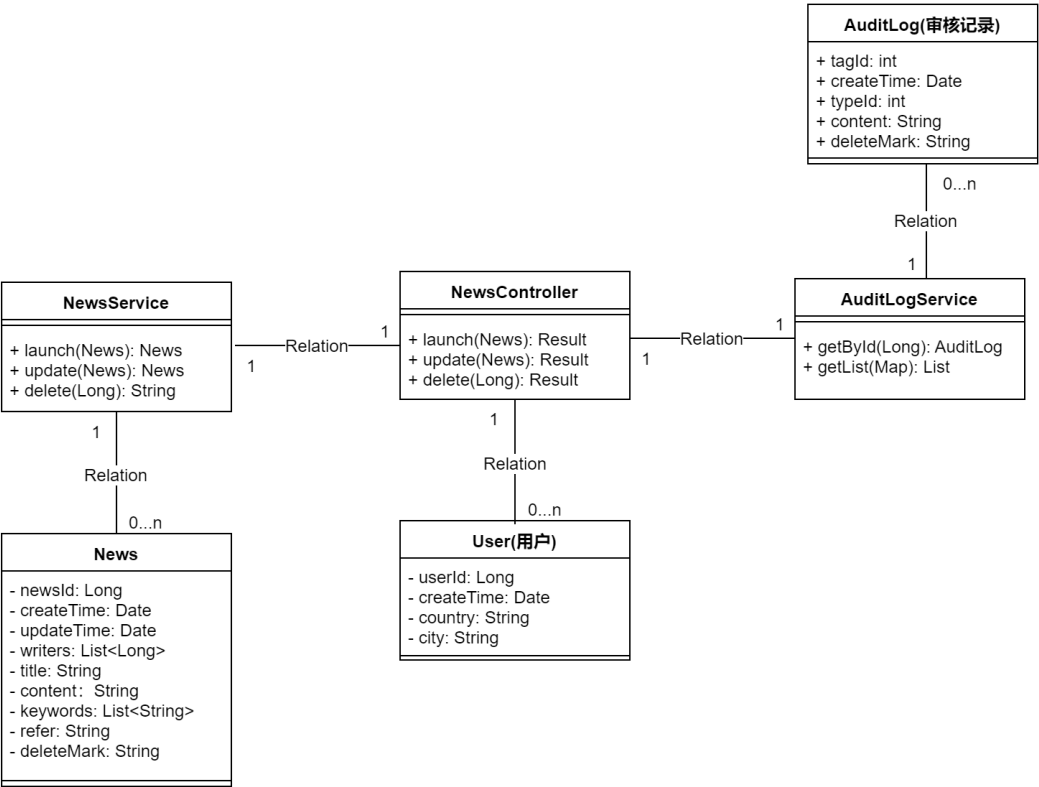


图 5-6 发布新闻

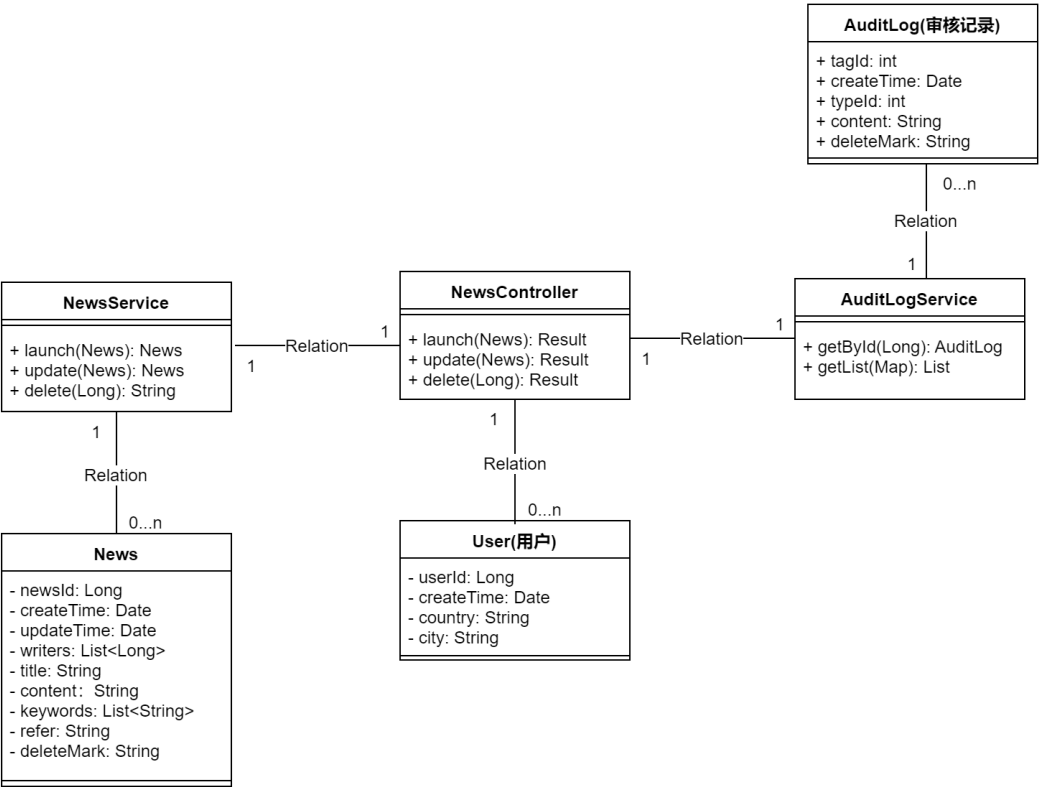


图 5-7 修改新闻

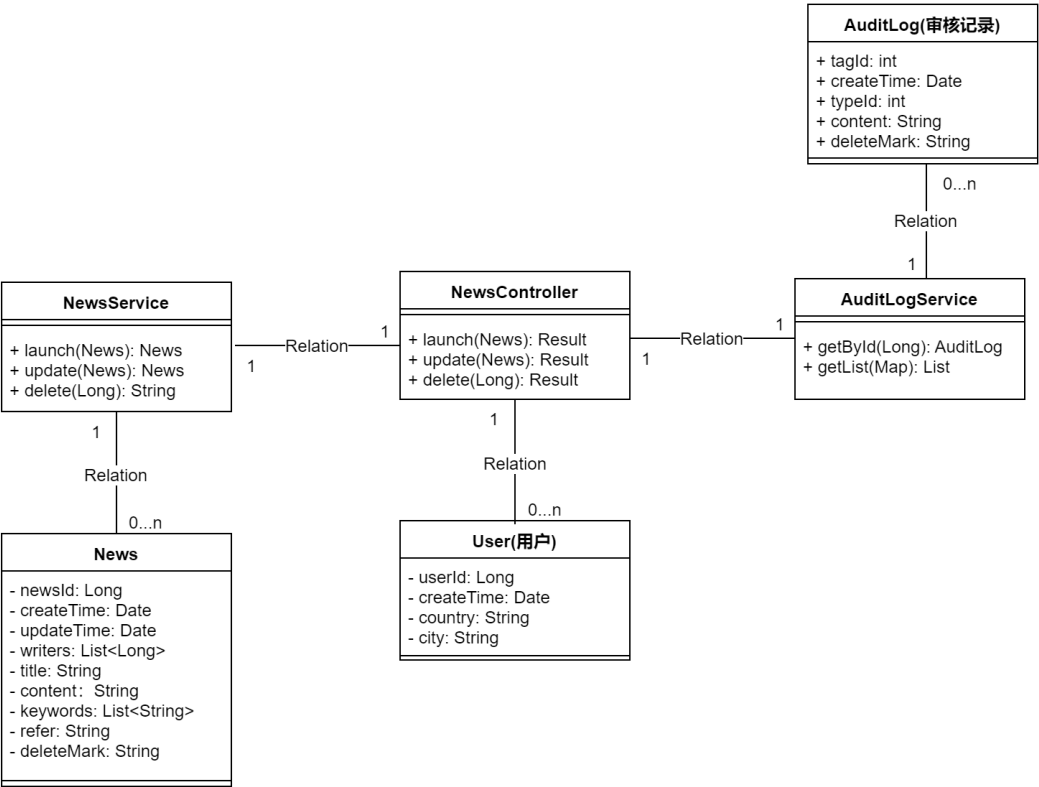


图 5-8 删除新闻

下架发布信息。涉及到的实体类有用户类（User），举报类（Report），审核类（AuditLog）。具体方法均在ReportController中进行调用。

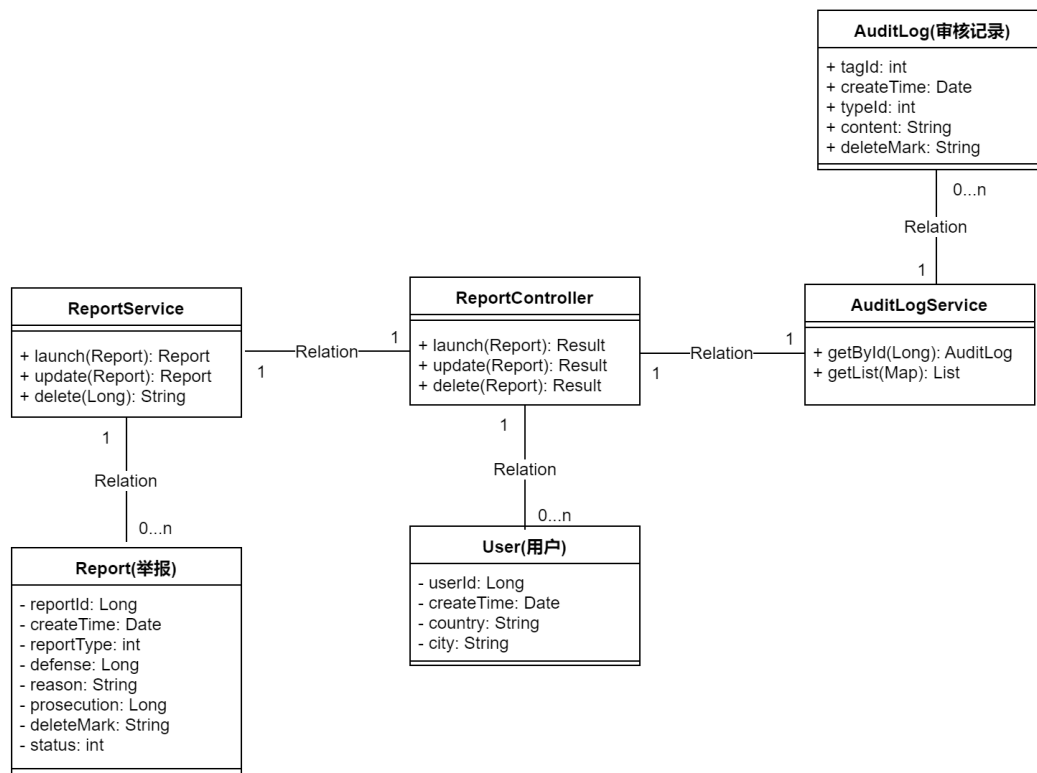


图 5-9 举报

5.11 举报审核

管理员对用户的举报进行审核，审核需要填写审核结果、审核信息，同时，还需要给涉及到的用户反馈审核结果，及通过系统通知反馈给举报人和被举报人，若举报人为游客，则不予反馈。涉及到的实体类有用户类（User），举报类（Report），审核类（AuditLog），消息类（Message）。具体方法在AuditLogController中调用。

5.12 系统通知

管理员通过消息的方式对用户进行系统通知，系统通知可以对全体用户广播，也可以针对于个别用户，管理员需要选择通知人群，填写通知内容。涉及到的实体类有用户类（User），消息类（Message），具体方法在MessageController中调用。

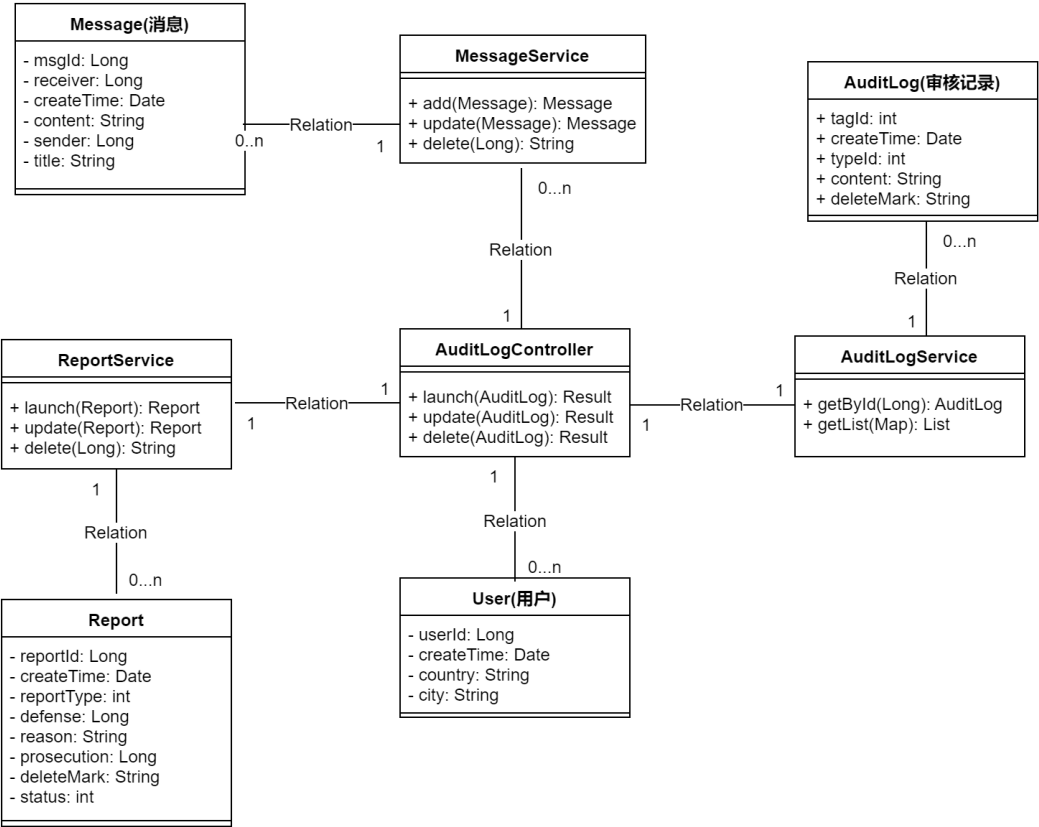


图 5-10 举报审核

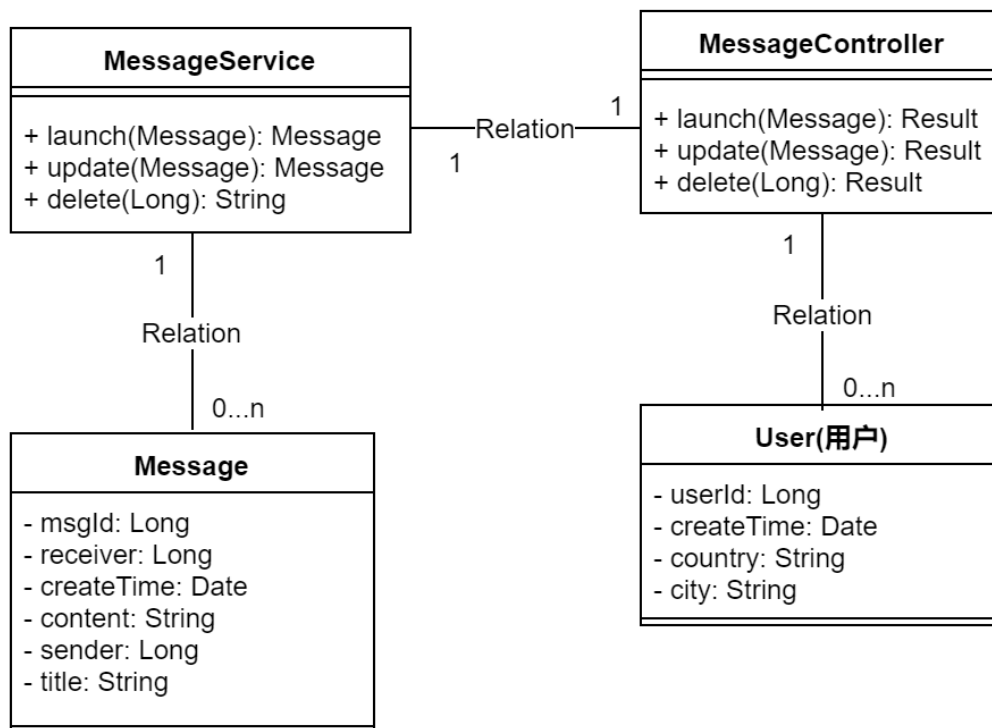


图 5-11 系统通知

5.13 用户管理

管理员可以通过后台对用户进行管理，包括用户封禁、用户查询、用户赋权、剥夺用户权限等操作，该模块的作用主要是对于用户权限的操作，涉及到的实体类有用户类（User），角色类（Role），权限类（Permission）。涉及到的方法在UserController中进行调用。

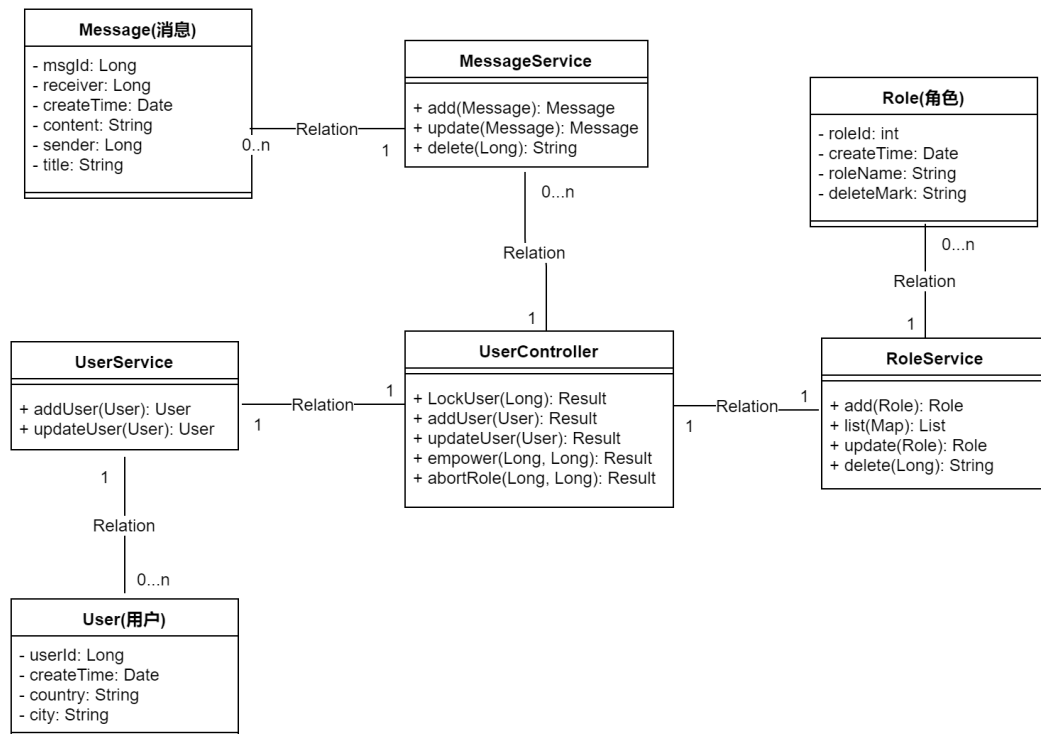


图 5-12 用户管理

5.14 角色管理

管理员可以通过后台对角色进行管理，包括删除角色，新增角色、角色赋权、剥夺角色权限等，该模块的作用主要是对于角色权限的操作，涉及到的实体类有角色类（Role），权限类（Permission）。涉及到的方法在RoleController中进行调用。

5.15 权限管理

管理员可以通过后台对权限进行管理，包括删除权限，新增权限等，涉及到的实体类为权限类（Permission），方法均在PermissionController中调用。

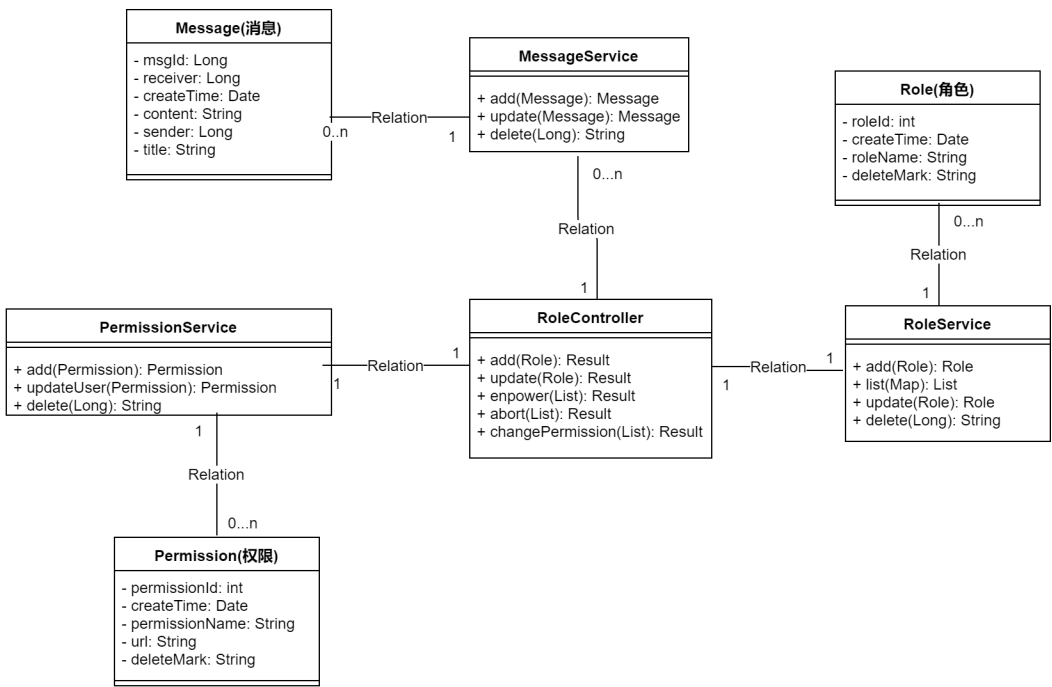


图 5-13 角色管理

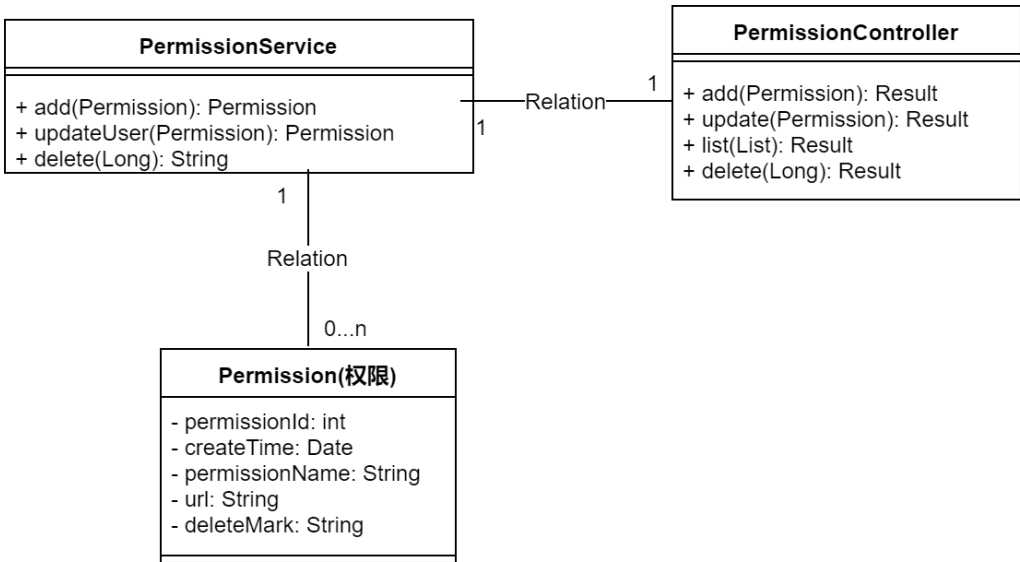


图 5-14 权限管理

第六章 详细设计

前几章设计了系统的框架、层次、业务对象等方面，本章将对系统进行详尽的设计，包括数据库的设计，主要流程的设计等。

6.1 数据库设计

本系统使用的是MySQL数据库，由于系统框架为微服务框架，所以本章将分模块为设计各自模块的数据库表。同时，为了提高查询速度，便于开发，所有的数据表均不会设置外键，也没有触发器，全靠后端上层逻辑保证数据的完整性。后期为提高查询速度会设计索引和视图。

6.1.1 UserService用户模块

用户模块的数据表主要设计用户基本信息、角色信息、权限信息等，一方面是为用户登录认证授权服务，另一方面是为管理员进行用户管理服务。权限管理方面采用的是RBAC的数据表设计，即将权限信息分为5个表，用户信息表、角色信息表、权限信息表、用户角色表和权限角色表。

6.1.1.1 User角色表

1. 表结构

字段名	字段类型	字段描述	初始值	约束
USER_ID	Long	用户唯一约束		PK,AUTO.INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
COUNTRY	VARCHAR(100)	国家		
CITY	VARCHAR(200)	城市		
NAME	VARHCAR(100)	姓名		
IDENTITY_NO	VARCHAR(30)	身份证		

2. 建表语句

```
CREATE TABLE(
  USER_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  COUNTRY VARCHAR(100),
  CITY VARCHAR(200),
  NAME VARCHAR(100),
  IDENTITY_NO VARCHAR(30)
)
```

6.1.1.2 USER_AUTH用户认证表

本系统支持第三方登录，所以每个用户对应多个登陆凭证和密码，比如邮箱、手机号等。本表记录的就是用户和对应的登陆方式，包括用户ID、登陆账号、登陆密码、登陆方式等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
AUTH_ID	Long	用户唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
USER_ID	Long	关联用户标识		NOT NULL
IDENTITY_TYPE	int	登录类别标识		NOT NULL
IDENTIFIER	VARCHAR(50)	身份唯一标识		NOT NULL
CREDENTIAL	VARCHAR(100)	登陆验证		NOT NULL

2. 建表语句

```
CREATE TABLE(
  AUTH_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
  USER_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  IDENTITY_TYPE INT NOT NULL,
  IDENTIFIER VARCHAR(50) NOT NULL,
  CREDENTIAL VARCHAR(100) NOT NULL
)
```

6.1.1.3 DICT_ROLE角色表

为了便于权限管理，根据RBAC设计，将多个权限抽象出来，封装在一个角色中。角色表 and 用户表是多对多的关系，即一个用户可以拥有多个角色，一个角色也可以被多个用户拥有。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
ROLE_ID	Long	角色唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
ROLE_NAME	VARCHAR(30)	角色名称		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```
CREATE TABLE(
  ROLE_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
```

```

ROLE_NAME VARCHAR(30) NOT NULL,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO'
)

```

6.1.1.4 DICT_PERMISSION权限表

权限表记载了系统中的所有权限，本系统中的权限值对应的是请求路径，即本系统的权限控制是基于URL的。角色表和权限表是多对多的关系，即一个角色拥有多个权限，一个权限也可以被多个角色拥有。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
PERMISSION_ID	Long	权限唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
PERMISSION	VARCHAR(30)	权限名称		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```

CREATE TABLE(
    PERMISSION_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
    CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
    PERMISSION_NAME VARCHAR(30) NOT NULL,
    DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO'
)

```

6.1.1.5 DICT_USER_ROLE用户角色表

此表为用户和角色的关联关系表，将用户id和角色id建立关联，从而可以知道用户对应的角色。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
ROLE_ID	INT	角色唯一约束		PK,AUTO_INCREMENT
USER_ID	Long	用户唯一约束		NOT NULL
GROUP_ID	INT	部门ID	10005	NULL DEFAULT 10005
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
LOCK_MARK	VARCHAR(3)	冻结标识	NO	NULL DEFAULT 'NO'
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```

CREATE TABLE(
    ROLE_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
    USER_ID INT(64) NOT NULL,

```



```

GROUP_ID INT NULL DEFAULT 10005,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
LOCK_MARK VARCHAR(3) NULL DEFAULT 'NO',
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO'
)

```

6.1.1.6 DICT_ROLE_PERMISSION角色权限表

此表为角色和权限的关联关系表，将角色id和权限id建立关联，从而可以获取角色对应的权限。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
ID	INT	唯一标识		PK,AUTO_INCREMENT
ROLE_ID	Long	角色唯一约束		NOT NULL
PERMISSION_ID	Long	权限唯一约束		NOT NULL
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
LOCK_MARK	VARCHAR(3)	冻结标识	NO	NULL DEFAULT 'NO'
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```

CREATE TABLE(
ROLE_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
PERMISSION_ID INT(64) NOT NULL,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
LOCK_MARK VARCHAR(3) NULL DEFAULT 'NO',
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO'
)

```

6.1.1.7 DICT_MENU栏目表

本表记录了该系统中当前登录用户可以操作的模块，用于前端动态获取并加载。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
MENU_ID	INT	唯一约束		PK,AUTO_INCREMENT
MENU	VARCHAR(100)	栏目名		NOT NULL
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
URL	VARCHAR(200)	路径	NO	NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```
CREATE TABLE(
MENU_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
MENU VARCHAR(100) NOT NULL,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
URL VARCHAR(200) NOT NULL
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO'
)
```

6.1.1.8 DICT_ROLE_MENU角色栏目表

此表记录了角色和栏目的对应关系，不同角色可以获取不同栏目。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
ID	INT	唯一标识		PK,AUTO_INCREMENT
ROLE_ID	INT	角色唯一约束		NOT NULL
MENU_ID	INT	栏目唯一约束		NOT NULL
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```
CREATE TABLE(
ROLE_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
PERMISSION_ID INT(64) NOT NULL,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
LOCK_MARK VARCHAR(3) NULL DEFAULT 'NO',
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO'
)
```

6.1.1.9 DICT_GROUP部门表

此表记录了系统中的所有部门，用于数据权限的管理，不同部门的人之间不能互相看到数据。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
GROUP_ID	INT	唯一标识		PK,AUTO_INCREMENT
GROUP_NAME	VARCHAR(200)	部门名		NOT NULL
PARENT_ID	INT	父级部门		NOT NULL
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```
CREATE TABLE(
  GROUP_ID INT PRIMARY KEY AUTO_INCREMENT,
  GROUP_NAME VARCHAR(200) NOT NULL,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  PARENT_ID INT NULL NOT NULL,
  DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO'
)
```

6.1.1.10 表格关系

该模块共涉及了9张表，下图 6-1 为9张表的关系。

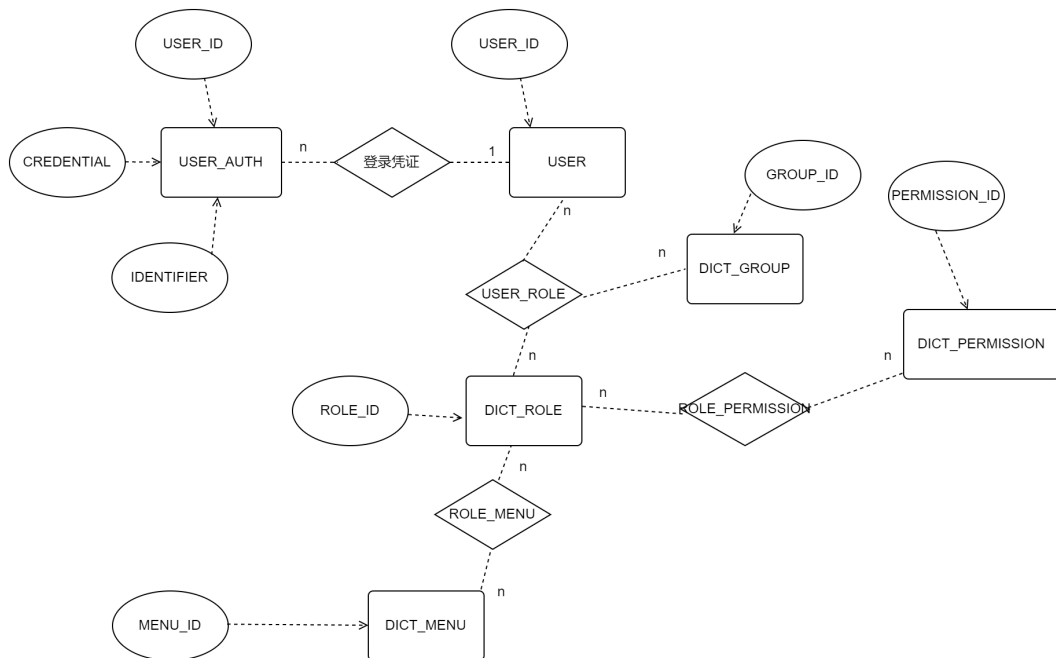


图 6-1 用户管理模块ER图

6.1.2 HouseService房源模块

房源模块主要涉及帖子基本信息、房主联系方式、帖子标签等等和房源帖子相关的数据表。为了功能的扩展性，我们会设置多个字典表，便于功能的增删改，比如标签表的设置是为了方便房源标签的及时增删。

6.1.2.1 POST帖子表

房源帖子包含了帖子的基本信息，比如房源地址、是否允许宠物、房源标签、房源描述等等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
POST_ID	Long	帖子唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
USER_ID	Long	关联用户标识		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
TITLE	VARCHAR(200)	标题	WE CAN OFFER HELP	NULL DEFAULT 'WE CAN OFFER
CITY	VARCHAR(200)	城市(市, 省, 国)		NOT NULL
GUESTS	int	人数	1	NULL DEFAULT 1
PETS	VARCHAR(3)	是否允许宠物	NO	NULL DEFAULT 'NO'
DURATIONTYPE	int	时长类型	1	NULL DEFAULT 1
TAGS	VARCHAR(200)	标签组		
ACTIVE	VARCHAR(3)	有效表示	YES	NULL DEFAULT 'YES'
DESCRIPTION	VARCHAR(500)	住所描述		

2. 建表语句

```

CREATE TABLE(
  POST_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
  USER_ID INT(64) NOT NULL,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
  TITLE VARCHAR(200) NULL DEFAULT 'WE CAN OFFER HEPL',
  CITY VARCHAR(200) NOT NULL,
  GUESTS INT NULL DEFAULT 1,
  PETS VARCHAR(3) NULL DEFAULT 'NO',
  DURATIONTYPE INT NULL DEFAULT 1,
  TAGS VARCHAR(200),
  DESCRIPTION VARCHAR(500),
  ACTIVE VARCHAR(3) NULL DEFAULT 'YES'
)

```

6.1.2.2 CONTACT联系方式表

此表记录了房源帖上具体的联系方式，为了便于联系方式的增删改，我们将联系方式独立出来形成一张数据表，以减少耦合度。表中主要记录了对应的帖子，以及基本的联系信息。

1. 表结构

2. 建表语句

```

CREATE TABLE(
  CONTACT_ID INT(64) PRIMARY KEY AUTO_INCREMENT,

```

字段名	字段类型	字段描述	初始值	约束
CONTACT_ID	Long	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
POST_ID	Long	关联关联标识		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
CONTENT	VARCHAR(200)	联系途径		NOT NULL
TYPE_ID	int	联系方式类别标识		NOT NULL

```

POST_ID INT(64) NOT NULL,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
CONTENT VARCHAR(200) NOT NULL,
TYPE_ID INT NOT NULL
)

```

6.1.2.3 DICT_CONTACT_TYPE联系方式类别表

此表记录了房源帖上那些联系方式的类型，为了后续可以便于直接增加联系方式类型，比如现阶段支持电话和邮箱，一段时间后想增加微信功能，独立出一个类别较容易扩展。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
TYPE_ID	int	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
CONTACT_NAME	VARCHAR(30)	联系方式		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```

CREATE TABLE(
TYPE_ID INT PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
CONTACT_NAME VARCHAR(30) NOT NULL
)

```

6.1.2.4 DICT_TAG标签表

此表记录了房源帖子上具体的标签，比如标签的类型，标签的内容等。比如说，一个房源帖子的标签为“拥有药物资源，类别为救助”等等。

1. 表结构

2. 建表语句

字段名	字段类型	字段描述	初始值	约束
TAG_ID	int	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
TYPE_ID	int	标签类别		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
CONTENT	VARCHAR(200)	标签内容		NOT NULL

```
CREATE TABLE(
TAG_ID INT PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
CONTENT VARCHAR(200) NOT NULL,
TYPE_ID INT NOT NULL
)
```

6.1.2.5 DICT_TAG_TYPE标签类别表

每个标签都有对应的类别，这是为了便于管理，也是为了便于扩展，此表记录了标签类别信息，包括类别id、类别内容等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
TYPE_ID	int	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
TYPE_NAME	VARCHAR(50)	类别内容		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```
CREATE TABLE(
TYPE_ID INT PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
TYPE_NAME VARCHAR(50) NOT NULL
)
```

6.1.2.6 DICT_DURATION时长类型表

此表记录了帖子里时长的信息，比如一周、一月、无限期等。

1. 表结构

2. 建表语句

```
CREATE TABLE(
DURATION_ID INT PRIMARY KEY AUTO_INCREMENT,
```

字段名	字段类型	字段描述	初始值	约束
DURATION_ID	int	唯一约束		PK,AUTO.INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DURATION	VARCHAR(50)	时长描述		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

```

CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
DURATION VARCHAR(50) NOT NULL
)

```

6.1.2.7 总体ER图

本模块的数据表均围绕房源帖子POST展开，下图 6-2 给出本模块的ER图。

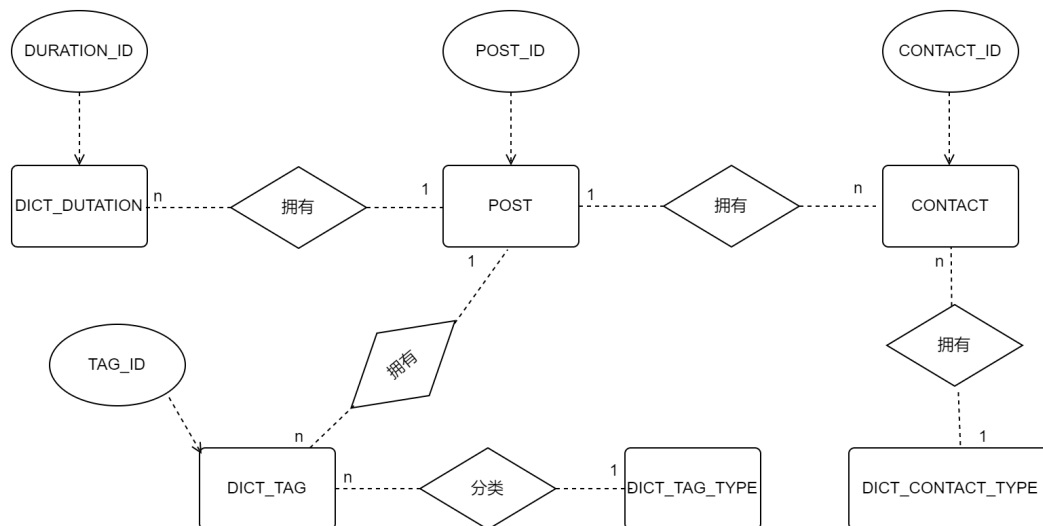


图 6-2 房源管理ER图

6.1.3 Report举报模块

举报模块主要涉及举报基本信息，同时会和其余的模块进行联通，比如用户表、房源帖子表等。

6.1.3.1 REPORT举报信息表

此表记录了举报的基本信息，包括举报内容、举报理由、举报类别、举报人等等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
REPORT_ID	Long	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
OBJTYPE_ID	int	举报类型		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
DEFENSE	Long	被举报者		NOT NULL
RESON	VARCHAR(500)	举报利用		NOT NULL
PROSECUTION	Long	举报者		NOT NULL
AUDIT_STATUS	int	审核状态	1	NULL DEFAULT 1

2. 建表语句

```
CREATE TABLE(
REPORT_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
OBJTYPE_ID INT NOT NULL,
DEFENSE INT(64) NOT NULL,
RESON VARCHAR(500) NOT NULL,
PROSECUTION LONG NOT NULL,
AUDIT_STATUS INT NULL DEFAULT 1
)
```

6.1.4 AUDIT模块

该模块为本系统的审核模块，记录的信息有审核的基本信息、审核类别等，目的是将系统的历次审核记录下来形成日志，便于管理。】

6.1.4.1 AUDIT审核表

此表记录了审核的基本信息，包括审核内容、操作人、操作类型等等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
AUDIT_ID	Long	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
OBJTYPE_ID	int	被审核类别		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
STATUS	int	审核状态	1	NULL DEFAULT 1
OPERATOR	Long	审核人		NOT NULL
OPER	int	操作		NOT NULL
MESSAGE	VARCHAR(200)			

2. 建表语句


```

CREATE TABLE(
  AUDIT_ID INT PRIMARY KEY AUTO_INCREMENT,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
  OBJTYPE_ID INT NOT NULL,
  STATUS INT NOT NULL,
  OPERATOR LONG NOT NULL,
  OPER INT NOT NULL,
  MESSAGE VARCHAR(200)
)

```

6.1.4.2 DICT_OPER_TYPE审核类型表

此表表示所有业务对象的审核类型，包括管理员通过、管理员驳回等，便于管理和扩展。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
OPER_ID	int	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
OPER	VARCHAR(200)	操作		NOT NULL

2. 建表语句

```

CREATE TABLE(
  OPER_ID INT PRIMARY KEY AUTO_INCREMENT,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
  OPER VARCHAR(200) NOT NULL
)

```

6.1.4.3 DICT_STATUS审核状态表

此表标识所有业务对象的审核状态，包括待审核、驳回、通过等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
STATUS_ID	int	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
STATUS	VARCHAR(30)	状态		NOT NULL
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'

2. 建表语句

```
CREATE TABLE(
  STATUS_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
  STATUS VARCHAR(30) NOT NULL
)
```

6.1.4.4 总体ER图

本模块的总体ER图如图 6-3所示

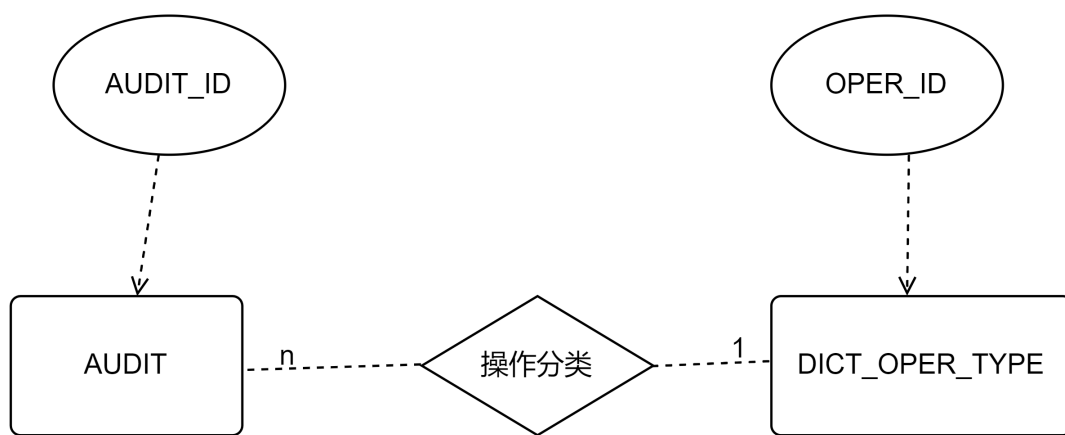


图 6-3 审核模块ER图

6.1.5 News模块

News是新闻模块，涉及到的数据表有新闻基本信息表，新闻内容表，新闻栏目表等。

6.1.5.1 NEWS新闻基本信息表

本表记录了新闻的基本信息，包括新闻的标题、栏目、阅读量等等。

1. 表结构

2. 建表语句

```
CREATE TABLE(
  NEWS_ID INT PRIMARY KEY AUTO_INCREMENT,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
  CATALOGUE INT NOT NULL ,
  TITLE VATCHAR(200) NOT NULL,
```

字段名	字段类型	字段描述	初始值	约束
NEWS_ID	Long	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
CATALOGUE	int	栏目		NOT NULL
TITLE	VARCHAR(200)	标题		NOT NULL
AUTHORS	VARCHAR(200)	作者		NOT NULL
LINK	VARCHAR(500)	引用链接		
UPDATE_TIME	Date	修改日期	CURRENTTIMESTAMP	NULL DEFAULT CURRENTTIMESTAMP
READ_NUM	int	阅读量	0	NULL DEFAULT 0
STATUS	int	状态	1	NULL DEFAULT 1

AUTHORS VARCHAR(200) NOT NULL,
 LINK VARCHAR(200) ,
 UPDATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
 READ_NUM INT NULL DEFAULT 0,
 STATUS INT NULL DEFAULT 1
)

6.1.5.2 NEWS_PIC新闻图片表

本表记录了新闻信息中的图片信息，表中保存了图片的链接地址、大小、文件名等信息。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
PIC_ID	Long	唯一约束		PK,AUTO_INCREMENT
NEWS_ID	Long	新闻标识		NOT NULL
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
PIC_NAME	VARCHAR(100)	图片名称		NOT NULL
FILE_NAME	VARCHAR(200)	文件名		NOT NULL
FILE_SIZE	int	大小		NOT NULL
PIC_DES	VARCHAR(500)	描述		
FILE_PATH	VARCHAR(200)	目录		NOT NULL

2. 建表语句

CREATE TABLE(
 PIC_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
 NEW_ID INT(64) NOT NULL,
 CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
 DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',

```

PIC_NAME VARCHAR(100) NOT NULL ,
FILE_NAME VARCHAR(200) NOT NULL,
FILE_SIZE INT NOT NULL,
PIC_DES VARCHAR(500) ,
FILE_PATH VARCHAR(200) NOT NULL
)

```

6.1.5.3 NEWS_CONTENT新闻文本表

此表记录了新闻的文本信息，本系统的文本系统拟定使用Markdown格式，表中记录了对应的新闻，文本文件的位置等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
CONTENT_ID	Long	唯一约束		PK,AUTO_INCREMENT
NEWS_ID	Long	新闻标识		NOT NULL
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
FILE_NAME	VARCHAR(200)	文件名		NOT NULL
FILE_SIZE	int	大小		NOT NULL
FILE_PATH	VARCHAR(200)	目录		NOT NULL

2. 建表语句

```

CREATE TABLE(
CONTENT_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
NEWS_ID INT(64) NOT NULL,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
FILE_NAME VARCHAR(200) NOT NULL,
FILE_SIZE INT NOT NULL,
FILE_PATH VARCHAR(200) NOT NULL
)

```

6.1.5.4 DICT_CATALOGUE新闻栏目类

此表记录了新闻的栏目集合，包括栏目名称、创建日期等。

1. 表结构

2. 建表语句

```

CREATE TABLE(
CATALOGUE_ID INT PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,

```

字段名	字段类型	字段描述	初始值	约束
CATALOGUE_ID	int	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
CATALOGUE	VARCHAR(30)	栏目		NOT NULL

```

DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
CATALOGUE VARCHAR(30) NOT NULL
)

```

6.1.5.5 总体ER图

如图 6-4 所示为NEWS模块的总体ER图。

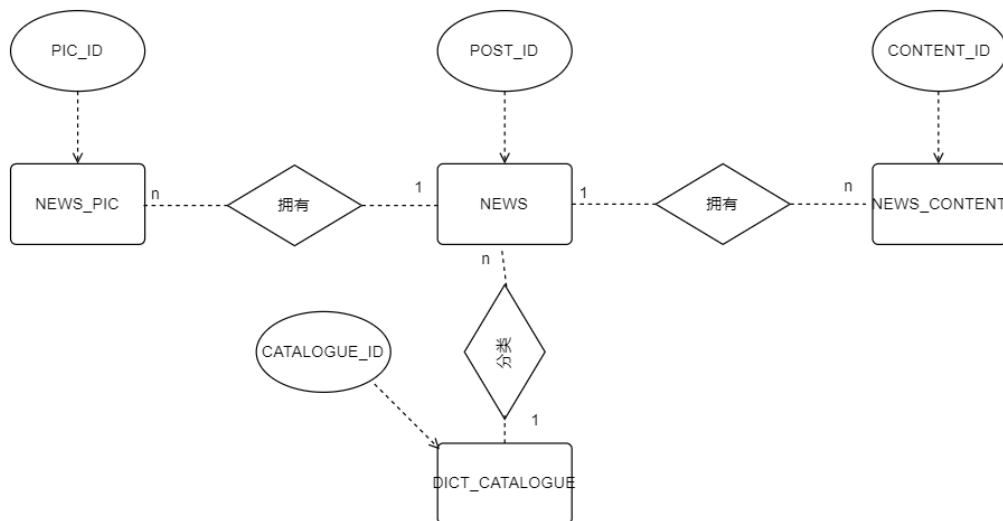


图 6-4 新闻模块ER图

6.1.6 SystemService模块

该模块为系统管理模块，保存系统相关的数据，比如业务对象类型、系统信息、系统日志等。

6.1.6.1 DICT_OBJTYPE业务对象表

本表记录了本系统中用到的所有业务对象，并将其进行分类，便于管理和业务对象的扩展。

1. 表结构
2. 建表语句

```
CREATE TABLE(
```

字段名	字段类型	字段描述	初始值	约束
OBJTYPE_ID	int	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
OBJTYPE	VARCHAR(100)	操作		NOT NULL

```

OBJTYPE_ID INT PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
OBJTYPE VARCHAR(200) NOT NULL
)

```

6.1.6.2 MESSAGE系统信息表

本表记录了系统消息类，具体包括消息类别、消息人群、操作人等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
MESSAGE_ID	Long	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
CONTENT	VARCHAR(1000)	内容		NOT NULL
SCOPE	int	范围		NOT NULL
SPECIFIC_USERS	VARCHAR(4000)	特定人群		
TITLE	VARCHAR(100)	标题		NOT NULL

2. 建表语句

```

CREATE TABLE(
MESSAGE_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
CONTENT VARCHAR(1000) NOT NULL,
SCOPE INT NOT NULL,
SPECIFIC_USERS VARCHAR(4000),
TITLE VARCHAR(100) NOT NULL
)

```

6.1.6.3 DICT_SCOPE消息范围表

本表记录了系统模块消息记录的发送范围，以便管理。

1. 表结构

2. 建表语句

字段名	字段类型	字段描述	初始值	约束
SCOPE_ID	Long	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
SCOPE	VARCHAR(100)	范围		NOT NULL

```

CREATE TABLE(
SCOPE_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
SCOPE VARCHAR(100) NOT NULL
)

```

6.1.6.4 SYSLOG系统日志表

本表记录了系统模块管理员所进行的一些重要操作，比如修改用户信息、修改新闻信息等等。本表信息包含了日志的操作类型、操作时间、板块类型等等。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
LOG_ID	Long	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
OBJTYPE_ID	int	业务对象类型		NOT NULL
OBJ_ID	Long	业务对象标识		NOT NULL
OPERATOR	Long	操作人		NOT NULL
OPERATION	int	操作		NOT NULL
MESSAGE	VARCHAR(200)	操作说明		NOT NULL

2. 建表语句

```

CREATE TABLE(
LOG_ID INT(64) PRIMARY KEY AUTO_INCREMENT,
CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
OBJTYPE_ID INT NOT NULL,
OBJ_ID INT(64) NOT NULL,
OPERATOR INT(64) NOT NULL,
OPERATION INT NOT NULL,
MESSAGE VARCHAR(200) NOT NULL
)

```

6.1.6.5 DICT_SYS_OPERATION

本表记录了系统管理模块对应着的一切操作类型，主要包含重要模块信息的增删改查。

1. 表结构

字段名	字段类型	字段描述	初始值	约束
OPER_ID	int	唯一约束		PK,AUTO_INCREMENT
CREATE_TIME	Date	创建时间	CURRENTTIMESTAMP	NULL DEFAULT CURRENTSTAMP
DELETE_MARK	VARCHAR(3)	删除标识	NO	NULL DEFAULT 'NO'
OPER	VARCHAR(200)	操作		NOT NULL

2. 建表语句

```
CREATE TABLE(
  OPER_ID INT PRIMARY KEY AUTO_INCREMENT,
  CREATE_TIME TIMESTAMP NULL DEFAULT CURRENTTIMESTAMP,
  DELETE_MARK VARCHAR(3) NULL DEFAULT 'NO',
  OPER VARCHAR(200) NOT NULL
)
```

6.2 重要流程设计

对于系统中的重要操作的后端逻辑流程，接下来进行详细的设计。重要逻辑流程，多为各个服务模块之间的先后调用，本节将使用时序图展示服务调用先后关系，比如用户登录、用户授权、用户发帖等等流程。

6.2.1 登录授权流程

登录是用户使用产品的开始，也是认证授权的开始，由于本系统采用的是微服务模块，各个模块之间独立部署，若是每个模块都集成登录功能，则太过冗余，若是在网关负责登录授权，一则网关负担过重，二则一旦网关攻破，内部会很危险，所以特意独立出一个认证授权服务专门负责此功能，同时采用JWT进行用户身份认证。

具体流程为：

- 用户请求登录接口，网关转发请求至用户认证模块，若认证成功则返回登录凭证token，并将用户权限存于Redis缓存，否则返回失败信息。
- 用户请求其余功能，网关检查是否携带token，若无token则返回登陆提示，若有将token发送至授权模块。
- 授权模块获取token，解析用户信息，获取用户权限，检验权限，并返回检验结果给网关。

- 若用户有权限，则网关转发请求至对应服务，若没有权限，则返回权限不足信息。

下图图 6-5 为时序图：

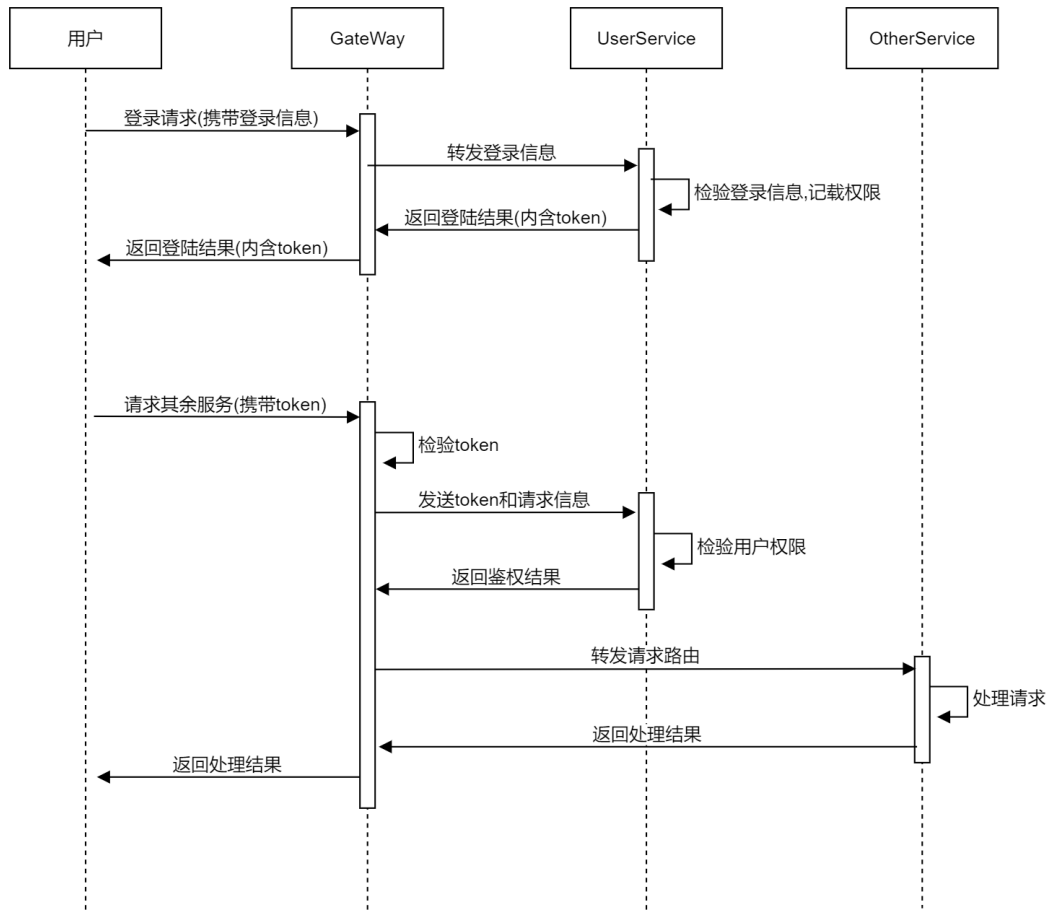


图 6-5 登录授权

6.2.2 用户邮箱注册

用户系统支持用户邮箱注册，邮箱注册需要用户输入自己的邮箱号，和预设的密码，以及基本的用户信息。提交注册申请后，后台会给用户发送检验邮件，用户需要打开自己的邮箱进行激活操作。

具体流程为：

- 用户请求注册接口，网关转发注册信息至用户管理模块。
- 用户管理模块检验参数合法性，若参数不合法则返回注册失败，若合法则生成密钥，存入Redis缓存，并将密钥嵌入邮件发送给注册邮箱。
- 用户打开邮箱点击激活链接进行激活，实际是一条携带密钥的激活请求，网关转发至用户管理模块。

- 用户管理检验密钥是否和缓存的匹配，若匹配则写入数据库，并返回注册成功，若不匹配或者是超过时限则返回注册失败。

下图图 6-6 为时序图：

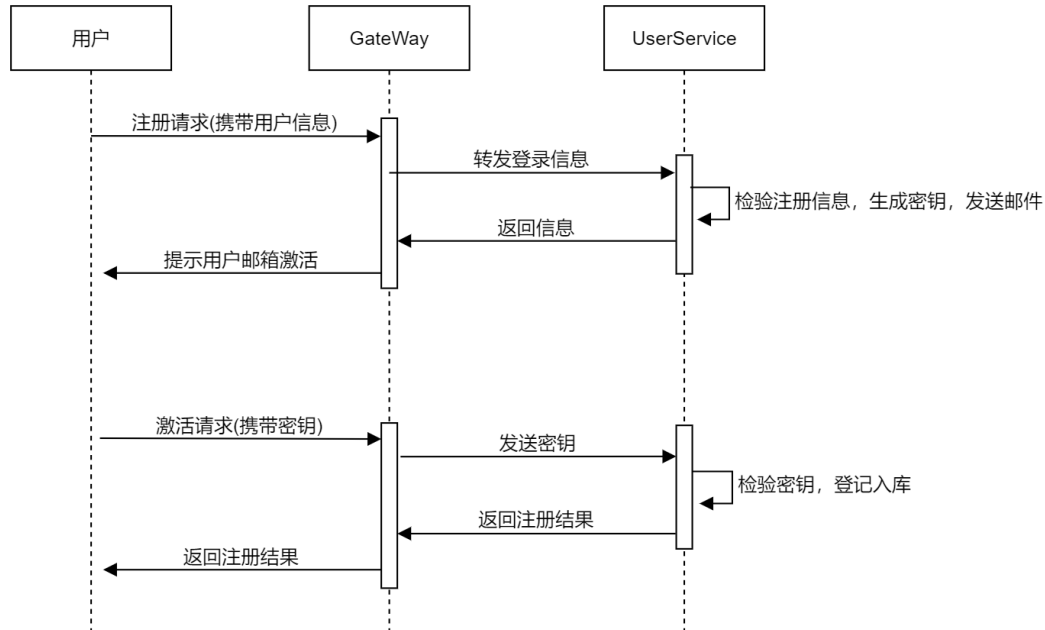


图 6-6 注册授权

6.2.3 用户发布房源贴

用户在房源系统中发布房源援助帖子，同时需要选择标签和联系方式。只有经过实名安全认证过的用户才能进行房源发布，实名安全认证服务有第三方提供。

具体流程为：

- 用户请求发帖接口，网关转发注册信息至房源管理模块。
- 房源接口检验参数合法性，不合法则返回发帖失败信息。
- 合法则将信息入库，并将发帖成功信息返回给客户端。

下图图 6-7 为时序图：

6.2.4 管理员进行用户管理

管理进行用户管理时，首先需要用户服务模块检验管理员的权限，权限通过管理员可以对用户进行操作。比如冻结或解冻用户，剥夺用户权限等等。同时，管理员的每次操作都需要载入日志。同时，系统会给帖子用户发送通知。

具体流程为：

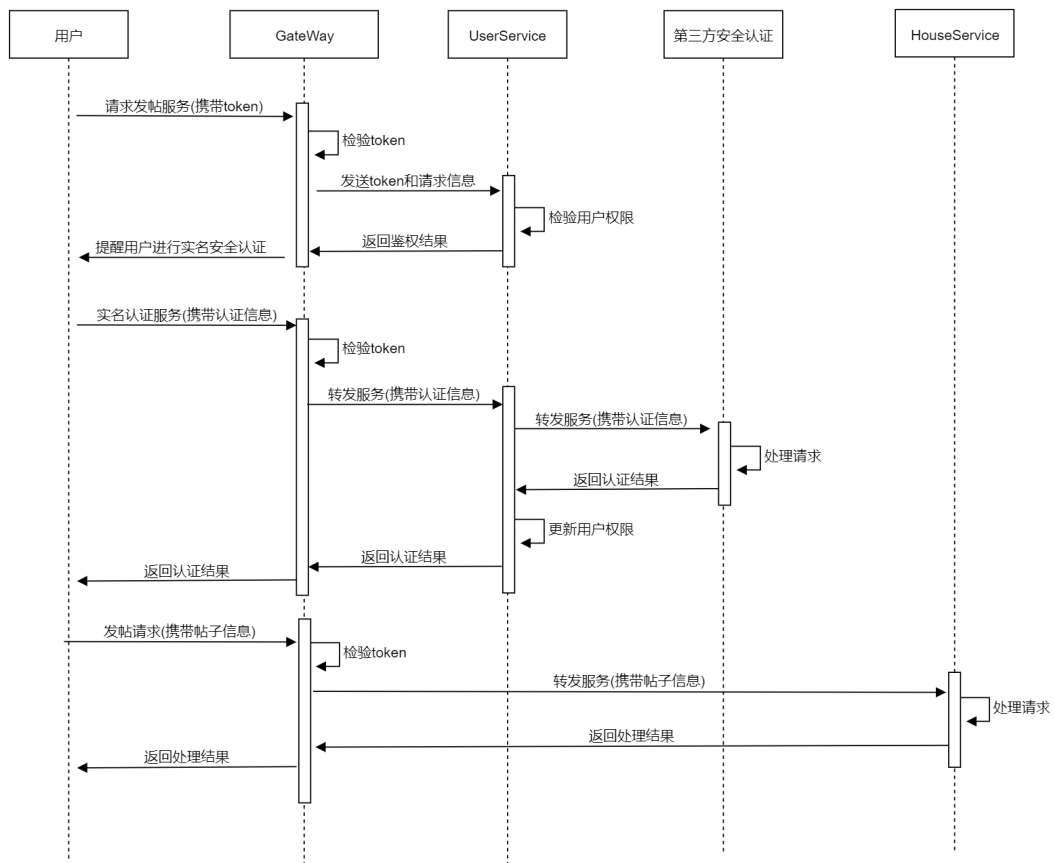


图 6-7 用户发布房源帖

- 管理员请求用户管理接口，网关转发请求信息至用户管理模块。
- 用户模块检验管理员身份和权限，通过则继续执行操作，否则返回权限不足。
- 用户模块执行用户管理操作，并生成日志信息，封装转发给系统服务模块。
- 系统服务接收服务信息，校验参数，写入日志数据库，同时发送通知给相关用户。

下图图 6-8 为时序图：

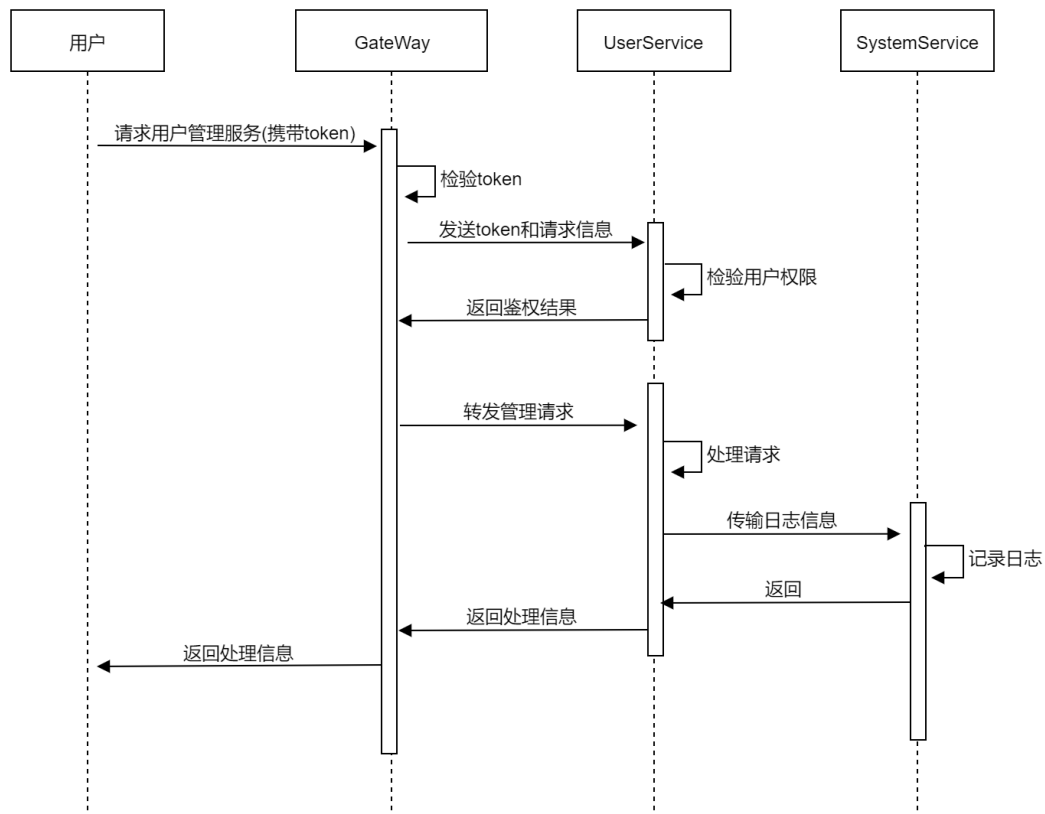


图 6-8 管理员用户管理

6.2.5 管理员进行房源管理

管理进行房源管理时，首先需要用户服务模块检验管理员的权限，权限通过后管理员可以对帖子进行操作。比如删除帖子等。同时，管理员的每次对帖子操作会生成日志文件，交由系统处理。同时，系统会给帖子用户发送通知。

具体流程为：

- 管理员请求帖子管理接口，网关转发请求信息至用户管理模块。

- 用户模块检验管理员身份和权限，通过则转发请求给房源系统，否则返回权限不足。
- 房源模块执行帖子管理操作，并生成日志信息，封装转发给系统服务模块。
- 系统服务接收服务信息，校验参数，写入日志数据库，同时发送通知给相关用户。

下图图 6-9 为时序图：

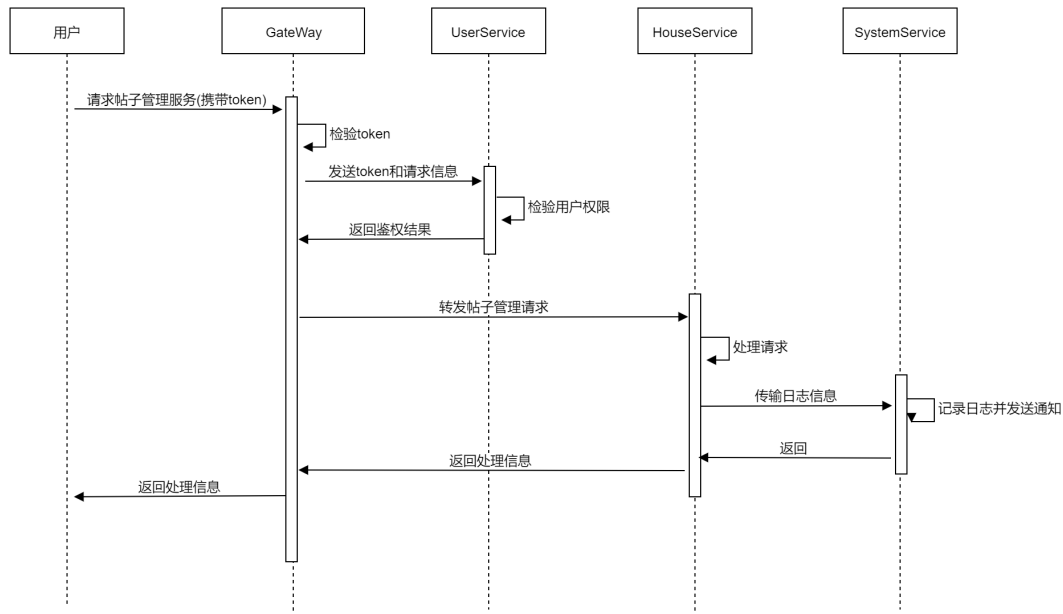


图 6-9 管理员帖子管理

6.2.6 管理员进行新闻管理

管理进行新闻管理时，首先需要用户服务模块检验管理员的权限，权限通过管理员可以对新闻进行操作。比如删除新闻等。同时，管理员的每次对新闻操作会生成日志文件，交由系统处理。同时，系统会给相关编辑者发送通知。

具体流程为：

- 管理员申请新闻管理接口，网关转发请求信息至用户管理模块。
- 用户模块检验管理员身份和权限，通过则继续转发给新闻模块，否则返回权限不足。
- 新闻模块执行新闻管理操作，并生成日志信息，封装转发给系统服务模块。
- 系统服务接收服务信息，校验参数，写入日志数据库，同时发送通知给相关用户。

下图图 6-10 为时序图：

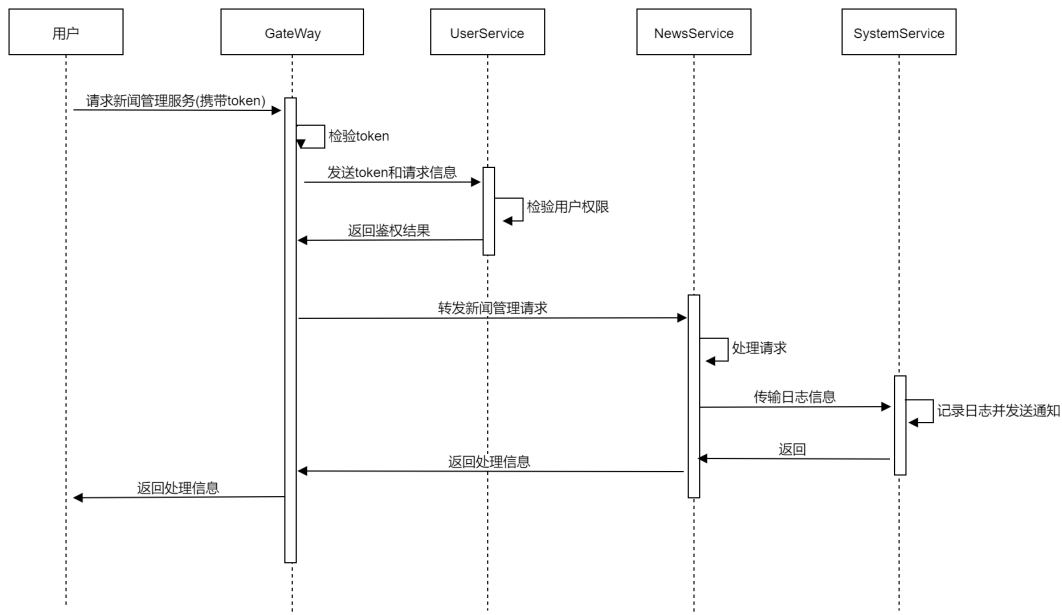


图 6-10 管理员新闻管理

6.2.7 管理员审核举报

管理员进行举报审核时，首先需要用户服务模块检验管理员的权限，权限通过后管理员可以对举报进行审核，比如通过和驳回等。同时，管理员的每次对审核操作会生成日志文件，交由系统处理。同时，系统会给相关编辑者发送通知。

具体流程为：

- 管理员申请举报审核接口，网关转发请求信息至用户管理模块。
- 用户模块检验管理员身份和权限，通过则继续转发给审核模块，否则返回权限不足。
- 审核模块处理后生成举报修改类，向举报模块发送修改举报状态的请求。
- 审核模块生成日志信息，封装转发给系统服务模块。
- 系统服务接收服务信息，校验参数，写入日志数据库，同时发送通知给相关用户。

下图图 6-11 为时序图：

6.2.8 管理员审核新闻

管理员进行审核新闻时，首先需要用户服务模块检验管理员的权限，权限

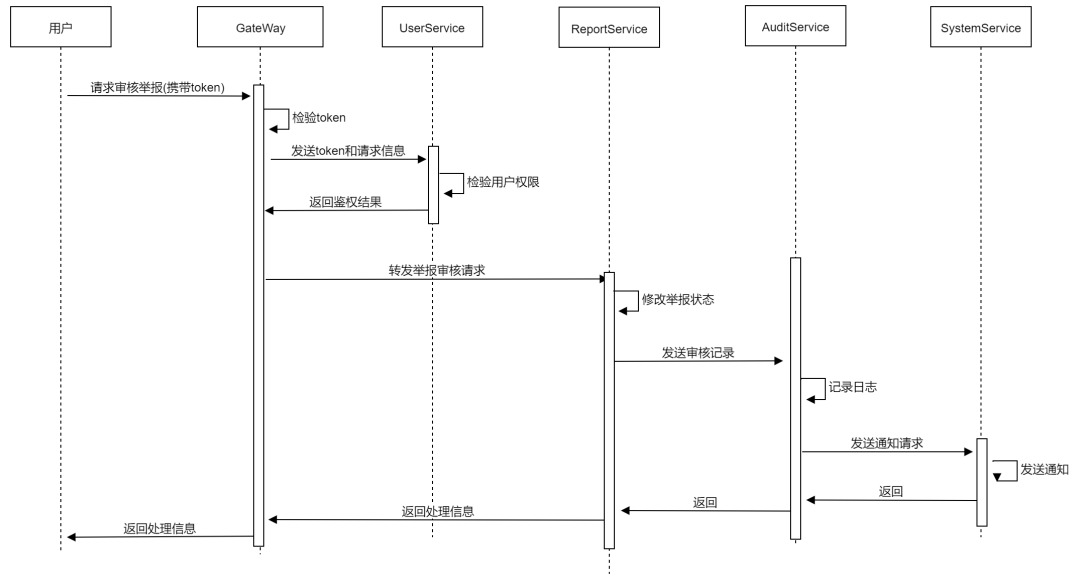


图 6-11 管理员审核举报

通过后管理员可以对新闻进行审核，比如通过和驳回等。同时，管理员的每次对审核操作会生成日志文件，交由系统处理。同时，系统会给相关编辑者发送通知。

具体流程为：

- 管理员申请新闻审核接口，网关转发请求信息至用户管理模块。
- 用户模块检验管理员身份和权限，通过则继续转发给审核模块，否则返回权限不足。
- 审核模块处理后生成新闻修改类，向新闻模块发送修改新闻状态的请求。
- 审核模块生成日志信息，封装转发给系统服务模块。
- 系统服务接收服务信息，校验参数，写入日志数据库，同时发送通知给相关用户。

下图图 6-12 为时序图：

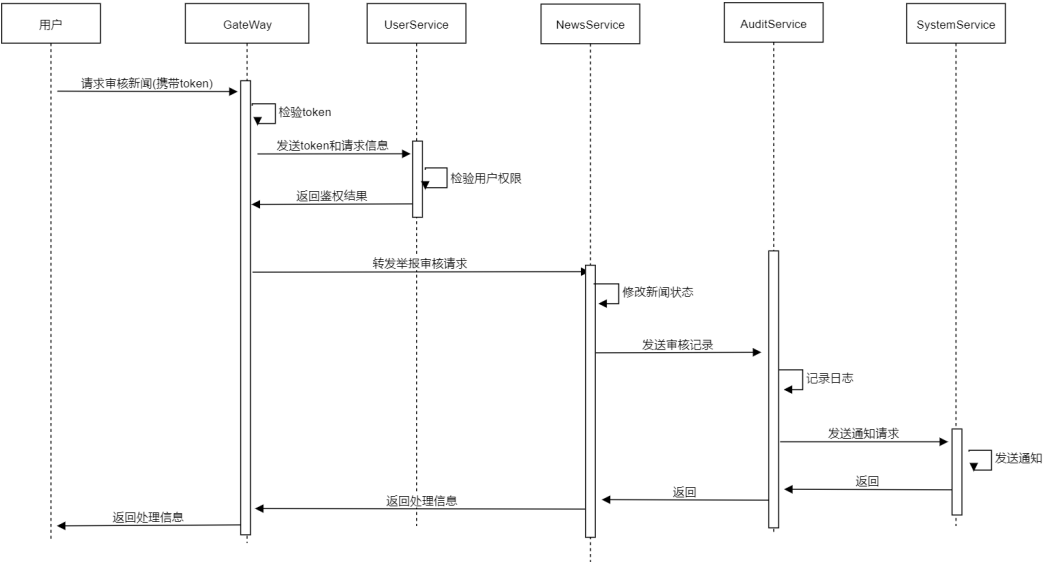


图 6-12 管理员审核新闻