成绩	（采用四级记分制）

西北大学

本科毕业论文（设计）

题目：流浪动物救助平台的设计与实现

学生姓名 俞家宝

学　　号 2021117338

指导教师 刘晓霞

院　　系 信息科学与技术学院（软件学院）

专　　业 软件工程

年　　级 2021 级

教务处制

二〇二五年五月

诚信声明

本人郑重声明：本人所呈交的毕业论文（设计），是在导师的指导下独立进行研究所取得的成果。毕业论文（设计）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或在网上发表的论文。

特此声明。

论文作者签名：

日　期：　2025 年 4 月 6 日

摘　　要

随着我国宠物行业的迅猛发展，流浪动物数量日益增多，传统救助模式暴露出诸多问题，如管理缺乏科学性、领养渠道不畅、公众认知不足等，这些问题致使 “救助 - 弃养” 的恶性循环不断加剧。为有效解决这些问题，本论文设计并实现了流浪动物救助平台。该平台基于微服务架构和层次结构风格，运用 Uni-App、Spring Boot、Dubbo 等技术，具备宠物领养、社区互动、活动管理以及后台管理等核心功能。

该平台通过构建完整的数字化管理系统，详细登记动物信息、更新健康档案、记录喂养情况，并设置多级领养审核流程，申请人需提交证明材料并接受背景核查，最终由管理员完成终审匹配领养对象，这一流程保障了动物福利，规范了领养责任，有效解决了传统领养中存在的不规范、随意性大的问题。社区互动模块通过设置多层交流系统，为不同用户提供差异化功能，普通访客可查看公开信息，注册用户能参与基础互动，实名认证用户可发布专业帖子，还具备实时通讯功能，极大地促进了信息共享与用户交流，打破了传统救助模式下信息不对称的壁垒，让救助信息能够更广泛、更快速地传播。

关键词：流浪动物救助；微服务架构；层次结构风格；Uni-App；Spring Boot

ABSTRACT

With the rapid development of China's pet industry, the number of stray animals has been increasing, and traditional rescue models have exposed many problems, such as lack of scientific management, 不畅 (unsmooth) adoption channels, insufficient public awareness, etc. These problems have led to the continuous intensification of the vicious cycle of "rescue-abandonment". To effectively solve these problems, this paper designs and implements a stray animal rescue platform. Based on the microservice architecture and hierarchical structure style, and using technologies such as Uni-App, Spring Boot, and Dubbo, the platform has core functions such as pet adoption, community interaction, activity management, and background management.

The platform builds a complete digital management system to detail animal information registration, health file updates, feeding situation records, and sets up a multi-level adoption review process. Applicants are required to submit proof materials and undergo background checks, and finally, administrators complete the final review to match adoption objects. This process ensures animal welfare, standardizes adoption responsibilities, and effectively solves the problems of irregularities and great randomness in traditional adoptions. The community interaction module provides differentiated functions for different users through a multi-layer communication system. Ordinary visitors can view public information, registered users can participate in basic interactions, real-name authenticated users can publish professional posts, and it also has a real-time communication function, which greatly promotes information sharing and user communication, breaks the information asymmetry barrier in the traditional rescue model, and enables rescue information to spread more widely and quickly.

Keywords： stray animal rescue; microservice architecture; hierarchical structure style; Uni-App; Spring Boot

目　　录

1 绪论

1.1 应用背景及意义

随着城市化进程加快，流浪动物问题愈发严重，本文将着手设计并实现流浪动物救助平台，借助科技力量来应对挑战。这个平台将软件工程和层次结构风格[1]结合在一起，采用Spring Boot[2],[3]和Uni-App框架实现，旨在打造一个全面、高效的救助平台。

该平台实现过程中，前端用Vue.js，保证用户体验，后端依赖Spring Boot，确保业务逻辑高效运行，数据库选MySQL[4]数据库用来存储和管理重要信息，保证数据的安全，系统架构按照模块化和层次化的原则设计，让每个功能模块既能独立工作又能相互配合，满足高并发的需求，同时确保数据的一致性和完整性。另外，流浪动物救助平台在设计时特别关注安全性，通过用户认证和授权来保护平台的安全运行。

1.2 国内外应用现状

1.2.1 国外应用现状

Kim等人（2015）[5]在《An Evaluation of the Role the Internet Site Petfinder Plays in Cat Adoptions》中提到，可以用Redis提升用户访问日志的处理速度，进而增强Web服务的整体性能，采用NoSQL数据库进行缓存的方式，对系统优化很有帮助，尤其是面对海量信息和用户访问数据时。

Workman和Hoffman（2015）[6]在《An Evaluation of the Role the Internet Site Petfinder Plays in Cat Adoptions》中发现猫咪在网站上每天被点击的次数越多，在收容所等待被领养的时间就越短。猫咪的年龄越大，等待领养的时间就越长，同时每天被点击的次数也越少，这说明在线平台对流浪动物领养起到了重要作用。

Zhang等人（2020）[7]在《Research on Uni-app Based Cross-platform Digital Textbook System》中提到用Uni-App技术解决移动数字教材的问题，这种技术能减少应用开发的花费，还能扩大覆盖范围。

Liu（2024）[8]在《Design and Implementation of Online Ordering System Based on SpringBoot》里用了Spring Boot做后端，Vue.js做前端，MySQL当数据库，通过这些技术，完成了模块化的设计，包括学生管理、教师管理、班级和课程管理等功能模块。

1.2.2 国内应用现状

张晓梅（2020）[9]在《图书馆微信小程序应用研究》中提到，微信小程序依托微信运行，

给用户带来多种服务，文章分析了公共图书馆使用微信小程序的重要意义和当前情况，根据用户的个性化需求改进了图书馆微信小程序的运营方式。

姜苏（2020）[10]在《基于互联网平台解决社会流浪动物问题的可行性研究》里归纳了流浪动物的现状、传统方法解决流浪动物问题的好坏之处以及互联网模式的长处，还尝试搭建了一套依靠互联网平台的流浪动物救助系统。研究借助验证实验确认了这套系统解决社会流浪动物问题的可行性，为处理社会流浪动物问题给出了有效办法。

王亚丽等人（2024）[11]在《基于 uniapp 搭建的助推非遗平台的设计与实现》里，选择了 vue 的 uni-app 前端框架，搭配 jquery 和 ajax 技术达成用户交互。同时，利用云数据库技术完成平台界面效果和实际功能，跨平台技术的使用让平台能覆盖更多移动设备，让用户访问更加方便。

李亮和舒畅（2024）[12]在《微服务架构与容器化技术的软件开发实践》中提到，微服务和容器化技术结合起来，能让企业更轻松应对业务需求的变化，同时，这种结合还能提升系统的可维护性，减少开发和运维中的复杂程度。

1.3 论文结构安排

这篇文章按照软件工程中的开发步骤，讲述《流浪动物救助平台》的设计与实现，先用问卷、访谈和实地观察收集用户需求，再分析出用户特点、使用场景和用例，接着，根据需求分析的结果设计系统平台的整体框架和具体功能，然后，实现系统功能并展示出来，还用开源测试工具生成数据报告，文章内容分成七个部分，安排如下：

第一章是绪论，先说应用背景和意义，再讲流浪动物救助的现状[13],[14],[15]，接着说明平台开发的重要性，还提了国内外相关研究的情况，为后面的内容做铺垫。

第二章主要讲流浪动物救助平台的需求分析，先说明软件工程里需求分析的理论，再看用户需求，这些需求通过问卷调查、用户访谈和现场观察得到，接着构建用户画像，确定典型场景，整理用例，最后弄清系统的功能边界和非功能需求。

第三章讲的是流浪动物救助平台的设计，先说明系统的设计目标，再根据需求分析结果确定整体架构、模块划分以及它们之间的交互关系。接着具体设计每个模块的功能、界面布局和数据库表结构，重点放在确保设计的合理性和未来扩展性上。

第四章讲的是流浪动物救助平台的实现，根据系统设计方案，用选定的编程语言和开发工具来写代码，写代码时要遵守规范，把系统的各项功能做出来，重视代码的质量和以后的维护，还要把核心功能的实现过程和关键代码片段展示出来。

第五章讲的是流浪系统救助平台的测试，用开源测试工具全面测试系统，包括功能、性能和兼容性等方面，把测试的数据和结果记下来，找出系统的问题和缺陷，给出改进的办法，做出详细的数据报告，保证系统的质量和稳定性。

第六章是总结和展望，回顾项目开发的过程和成果，整理系统的优势与不足，提出未来研究的方向，比如功能扩展、性能提升等，为后续平台的改进提供参考。

2　流浪动物救助平台需求分析

2.1　流浪动物救助平台功能性需求

通过对流浪动物救助相关利益方的系统性调研，采用问卷调查、深度访谈和实地观察等方法，明确流浪动物救助平台的主要角色为普通用户、志愿者和管理员，各角色在平台上拥有不同的权限和功能。基于调研结果，整理了平台用例图（见图 2.1），并对各类角色的功能需求进行了如下分析。

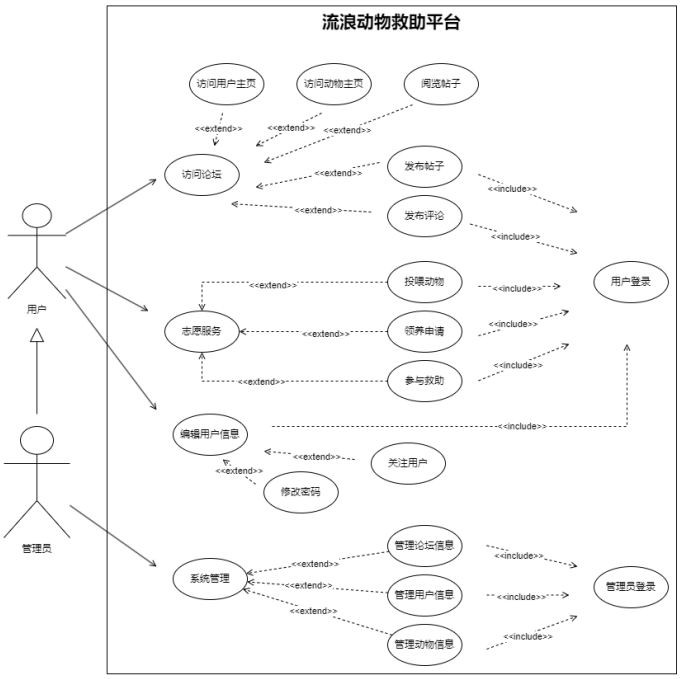

图 2.1 用例图

2.1.1 普通用户功能

普通用户可浏览平台内的动物信息、救助动态及相关知识，并参与论坛互动。平台应支持普通用户浏览动物主页、用户主页和各类帖子，发布图文或视频内容，及在帖子下发表评论。

普通用户可对个人资料进行管理，包括修改个人信息、更新密码、关注其他用户等。

普通用户可记录并分享自身关注或参与的救助案例，包括发现动物的地点、救助过程和动物后续状态。

普通用户可上传动物行为视频或描述，平台通过人工智能技术对动物健康状况进行初步分析，并提供相应建议，辅助用户科学判断动物状况。

2.1.2 志愿者功能

志愿者除具备普通用户的全部功能权限外，还可主动申请参与动物救助活动，包括动物

投喂、现场救助、医疗协助等。平台应支持志愿者报名各类救助活动，并对其参与情况进行记录和统计。

志愿者可发起动物领养申请，并跟踪申请进度，平台应为志愿者提供详细的领养流程指引及反馈机制，确保领养工作的规范性和透明度。

志愿者可在论坛或知识分享模块中发布专业救助经验、科普文章等内容。

2.1.3 管理员功能

管理员负责平台的整体运营和系统管理，包括论坛内容审核、用户与志愿者信息管理、动物信息维护等。管理员需确保信息合规、权限合理分配，并及时更新动物资料，保障平台秩序和数据准确性。

管理员可对用户、志愿者、动物及救助活动等数据进行统计分析，并通过可视化手段（如趋势图、热力图等）展示，为平台决策和资源配置提供数据支持。

管理员可设定各类数据阈值，实现对异常情况（如求助信息激增、违规操作频发等）的自动预警。对于常见违规行为，系统可自动处理（如账号屏蔽、内容删除），提升管理效率并减轻管理员工作负担。

2.2 流浪动物救助平台非功能性需求

2.2.1 性能需求

普通用户代表表示系统要保证具有良好的性能。在日常使用场景下，如查看动物信息、浏览帖子等操作，系统平均响应时间应控制在 0.5 秒以内，最长不超过 1 秒，确保不会感受到明显延迟。即使在高峰时段，系统在用户的操作下，仍应保持在 2 秒以内的响应时间。

2.2.2 安全性需求

管理员用户代表认为平台涉及大量用户个人信息和动物相关数据，数据安全至关重要。要防止数据被窃取或篡改。同时需要建立严格的身份验证机制，限制不同角色对功能和数据的访问，防止越权访问。

2.2.3 易用性需求

志愿者用户代表认为平台界面设计应遵循简洁直观的原则，布局合理、颜色搭配协调，避免信息过于繁杂，确保他们能够快速找到所需功能入口。另外对于复杂操作（如领养申请）提供清晰的步骤提示和引导信息，告知所需填写的信息及操作方法，减少出错概率。

2.2.4 可靠性需求

相关技术人员表示希望选用成熟稳定的技术框架和中间件，建立完善的容错机制。当系

统部分组件发生故障时，能够自动进行故障转移和恢复，确保核心功能不受影响。同时，建立定期的数据备份和恢复机制，将备份数据存储在异地，防止数据丢失。在发生数据灾难时，能够快速恢复数据。

2.3 流浪动物救助平台的可行性

经济可行性。系统开发主要依靠开源技术，成本相对较低。后期维护成本也在可控范围内。

技术可行性。选用的技术栈成熟稳定，能够满足系统开发需求。研究者具备相关技术背景，能够胜任系统开发工作。

社会可行性。随着人们对动物福利的关注度提高，该系统有望得到社会各界的支持和认可。

操作可行性。系统设计注重用户体验，界面友好，操作简单，易于被救助组织和普通用户接受和使用。

3　流浪动物救助平台的设计

3.1　流浪动物救助平台架构设计

为满足系统应对高并发、具备高扩展性和易于维护的需求，本平台采用微服务架构与层次结构风格[16],[17]理念，并结合前后端分离技术，将系统架构划分为用户接口层、应用层、领域层和基础设施层。

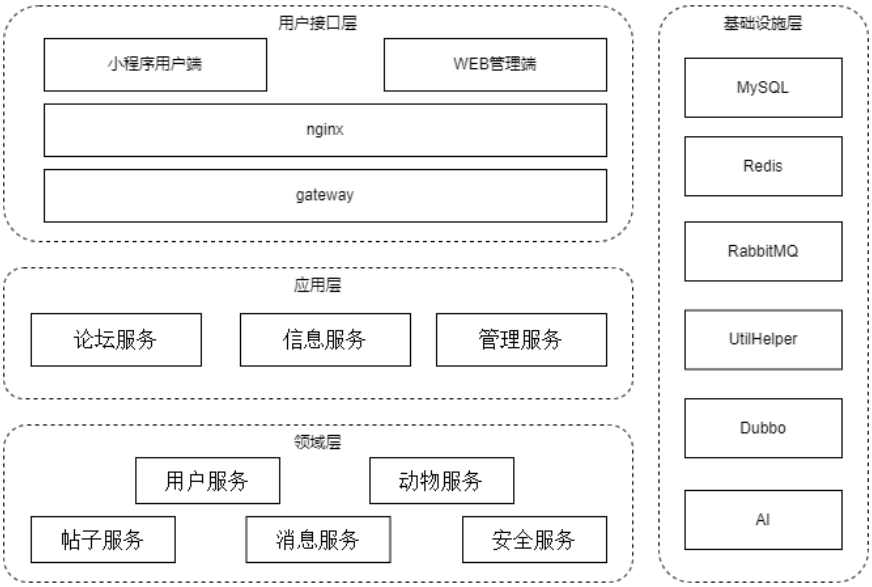

图 3.1 系统架构图

3.1.1　用户接口层

为满足用户在不同终端便捷访问平台功能，以及管理员高效管理平台数据的需求，设计了用户接口层。

在用户访问方面，小程序用户端选用 Uni - App 框架开发。其一次开发、多端运行的特性，能兼容多种平台，满足用户在不同设备上使用的需求。借助 axios 与后端 API 交互，实现数据快速传输与接收，确保用户在浏览动物信息、参与社区互动等操作时流畅无阻。

对于管理员管理需求，后台管理采用 Vue.js 框架搭配 Element UI 库搭建。Vue.js 的组件化开发提高搭建效率，Element UI 提供丰富组件方便界面设计。管理员通过 axios 调用后端接口，可对用户、动物、帖子等数据进行管理与审核，满足对平台数据进行集中管理和维护的需求。

此外，前端服务器使用 Nginx 托管静态文件，通过反向代理转发用户请求，提升系统安全性和性能。网关服务进行权限鉴权和微服务路由，确保只有合法用户能访问资源，保障系统稳定运行，满足系统安全稳定运行的需求。

3.1.2 应用层

为满足平台业务逻辑的协调处理和功能实现的需求，设立应用层。应用层虽不包含具体业务规则，但明确了软件的应用逻辑，负责协调领域层对象完成任务。

论坛服务整合领域层的帖子服务和消息服务，满足用户在社区内进行信息共享、互动交流和活动管理的需求，为用户提供活跃的交流场所。

信息服务与用户服务、动物服务协作，满足对用户、动物和关注等核心数据进行管理，以及跟踪领养流程的需求，确保数据的准确存储和高效查询。

管理服务针对 Web 管理端功能，满足管理员对用户提交内容审核、用户数据管理，以及全面掌握平台运行状态、处理违规信息的需求，保障平台稳定运营。

3.1.3 领域层

为满足平台核心业务逻辑的实现和管理需求，设计领域层来封装关键业务功能，并通过接口定义服务边界。

用户服务处理用户权限、认证和角色管理，满足在用户注册或登录时进行身份验证、权限分配的需求，确保不同用户角色只能访问和使用被授权的功能与数据，保障系统操作的安全性和规范性。

动物服务管理动物信息、投喂记录和领养申请，满足对流浪动物信息全面记录、喂养情况跟踪，以及规范领养流程、保障动物福利的需求。

帖子服务处理社区内的帖子、评论和活动报名信息，满足用户发布内容、参与互动，以及活动报名管理的需求，确保社区互动的有序进行。

消息服务负责聊天记录和通知推送，满足用户之间私信交流、接收系统通知的需求，实现信息的快速传递和精准送达。

安全服务进行用户登录验证和敏感词过滤，满足保障用户登录安全和维护平台健康环境的需求，防止非法登录和不良信息传播。

3.1.4 基础设施层

基础设施层为系统其他层提供技术支撑、保障系统稳定运行和高效处理数据。

多个独立服务基于 Spring Boot[18]搭建，利用 Nacos 进行服务注册和发现，Dubbo[19]负责服务间的 RPC 通信，满足系统服务管理和高效通信的需求，确保服务的稳定运行和调用的便捷性。

选用 MySQL 存储结构化数据，Redis 缓存高频访问数据[20]，满足系统对数据存储和快速

读取的需求，减轻数据库压力，提高系统响应速度。Redis 实现分布式锁，满足多线程或多服务环境下数据一致性的需求。

采用 RabbitMQ 消息队列中间件处理异步任务，满足系统异步任务排队处理的需求，避免因任务处理不及时影响系统性能。

这种架构设计将服务模块按业务领域划分，各部分独立性强，便于部署和扩展，减少耦合。Dubbo 降低服务间通信紧密程度，提高系统可靠性。Nginx 结合 Docker 容器部署，满足系统根据业务需求灵活扩展的需求，提升并发处理能力。Redis 缓存热点数据，加快系统反应速度。领域模型明确业务范围，便于复杂业务逻辑的维护。将通用功能置于基础服务层，提高开发效率，满足快速开发和持续优化的需求。

3.2 流浪动物救助平台数据库设计

3.2.1 数据库模型

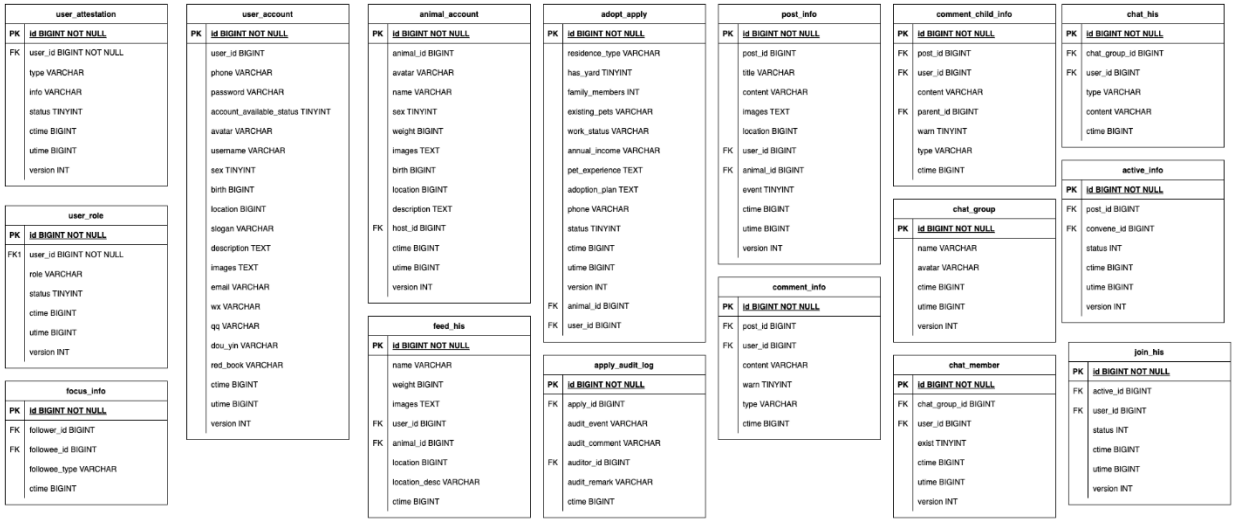

图 3.2 数据库表结构图

为满足流浪动物救助平台多元化业务需求，平台数据库采用 MySQL 作为主要存储系统，具体表结构（如图 3.2）说明如下：

用户账户表（user_account）。该表主要字段包括用户唯一标识、手机号、密码、头像、用户名、性别、出生日期、所在地、个性签名、简介、图片、邮箱及第三方账号（如微信、QQ、抖音、小红书）等。

用户角色表（user_role）。该表实现用户权限分级和多角色管理，支持同一用户拥有多种角色（如普通用户、志愿者、管理员等）。

用户认证表（user_attestation）。存储用户的实名认证、机构认证等信息，保障平台实名制和合规要求。字段涵盖认证类型、证件信息、认证状态等，通过索引优化认证信息的快

速检索。

动物账户表（animal_account）。记录救助动物的详细档案，包括动物编号、头像、名称、性别、体重、图片、生日、所在地、描述、归属人等。

领养申请表（adopt_apply）。该表用于管理用户对动物的领养申请，涵盖居住类型、家庭成员、现有宠物、工作状态、年收入、养宠经验、领养计划、联系电话等详细信息。

领养审核日志表（apply_audit_log）。记录领养申请的每一次审核过程，包括审核事件、意见、审核员、备注及时间戳，实现全流程可追溯，保障领养流程的规范性与透明性。

投喂记录表（feed_his）。该表保存用户对动物进行投喂的详细记录，包括食物名称、克重、图片、投喂人、动物、地点、详细地址及时间。

帖子信息表（post_info）。存储社区内用户发布的帖子，字段包括帖子 ID、标题、内容、图片、发布人、关联动物、事件标识等。

评论信息表（comment_info, comment_child_info）。支持主评论与子评论的分层结构，分别记录评论内容、所属帖子、评论人、评论类型、敏感标记等，满足社区互动的多样化需求。

群聊与成员表（chat_group, chat_member, chat_his）。支持用户组建群聊、加入群聊及消息历史管理，实现平台内的实时及异步通讯。

通过群聊 ID、用户 ID 等索引提升消息分发与历史检索效率。

关注关系表（focus_info）。记录用户之间的关注关系，支持多类型关注（如用户对用户、用户对动物等），为平台社交网络和信息流推荐提供数据支撑。

事件活动与参与表（active_info, join_his）。管理平台内的志愿活动、救助事件及用户参与记录。

3.2.1 实体关系与 ER 图设计

本平台数据库采用面向业务领域的实体关系建模，核心业务实体涵盖用户、动物、领养、互动、活动、社交等模块，实体关系如图 3.2 所示。

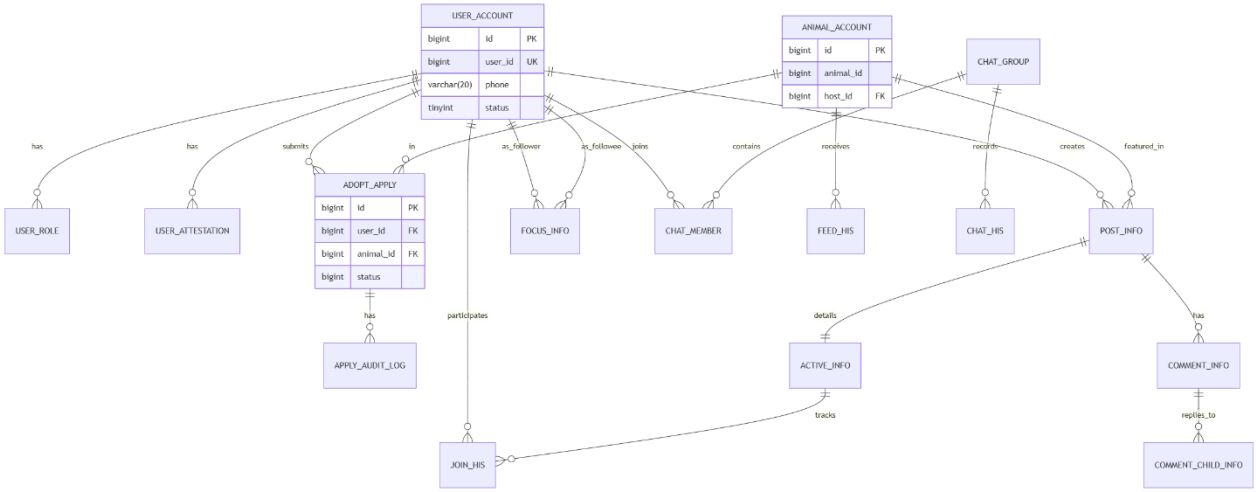

图 3.3 数据库实体关系图

用户与角色、认证、领养申请、帖子、关注、群聊成员、活动参与等实体均为一对多或多对多关系，支持灵活的权限和社交体系；

动物与领养申请、投喂记录、帖子等一对多关联，便于管理动物全生命周期数据；

领养申请与审核日志为一对多，确保流程可追溯；

帖子与评论、活动详情形成内容互动与活动组织的复合关系；

评论支持主子层级，满足深度社区互动；

群聊与成员、消息之间的关系支持实时与异步通讯；

活动与参与历史、用户与参与历史的多对多关系，便于统计和服务时长管理

3.2.2 数据表结构设计示例

以用户账户表为例，简要说明其结构设计，如图 3.4 所示。

```
CREATE TABLE user_account (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY COMMENT '主键id',
    `version` INT UNSIGNED DEFAULT 0 COMMENT '版本号',
    user_id BIGINT UNSIGNED UNIQUE NOT NULL COMMENT '用户id',
    phone VARCHAR(20) NOT NULL COMMENT '手机号',
    `password` VARCHAR(255) default '' COMMENT '密码',
    account_available_status TINYINT UNSIGNED DEFAULT 1 COMMENT '账户可用状态 0-禁用 1-启用',
    avatar VARCHAR(255) NOT NULL COMMENT '用户头像路径',
    username VARCHAR(50) NOT NULL COMMENT '用户名',
    sex TINYINT UNSIGNED DEFAULT 0 COMMENT '性别 0-未知 1-男 2-女',
    birth DATE COMMENT '出生日期',
    location BIGINT COMMENT '地区',
    slogan VARCHAR(255) COMMENT '格言',
    description TEXT COMMENT '简介',
    images TEXT COMMENT '用户图片',
    email VARCHAR(100) COMMENT '邮箱',
    wx VARCHAR(50) COMMENT '微信',
    qq VARCHAR(20) COMMENT 'QQ',
    dou_yin VARCHAR(50) COMMENT '抖音',
    red_book VARCHAR(50) COMMENT '小红书',
    ctime BIGINT NOT NULL COMMENT '创建时间',
    utime BIGINT NOT NULL COMMENT '更新时间',
    INDEX idx_phone (phone),
    INDEX idx_user_id (user_id),
    INDEX idx_email (email),
    INDEX idx_ctime (ctime),
    INDEX idx_username(username),
    INDEX idx_account_status(account_available_status),
    INDEX idx_user_location(location)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='用户账户信息表';
```

图 3.4 用户账户表 DDL

各表均设置自增主键和版本号（version）以支持乐观锁，保障并发场景下的数据一致性。所有时间字

段采用秒级时间戳，便于数据统计与查询。通过合理设计联合索引及唯一约束，提升数据检索效率与完整性约束能力。

3.3 流浪动物救助平台功能设计

本节根据前期需求分析，结合平台实际业务，对流浪动物救助平台功能进行设计。

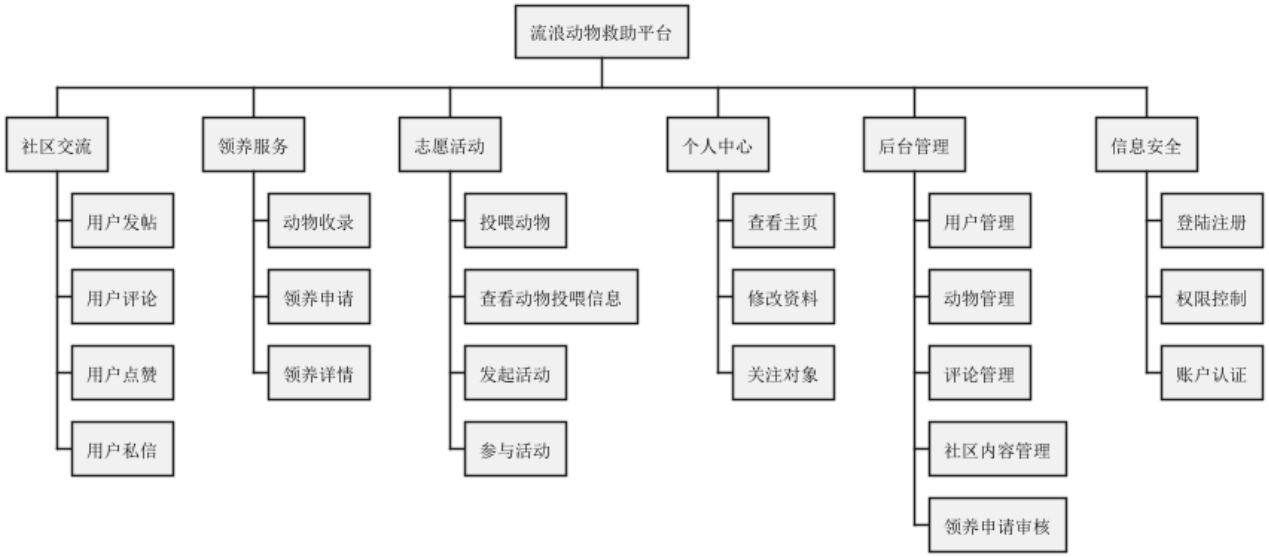

图 3.5 流浪动物救助平台功能树

3.3.1 社区交流模块

社区交流模块主要用于用户发帖、评论、点赞和私信。未注册用户仅能浏览公开内容，注册用户可以发帖、评论和点赞。完成实名认证的用户可发布救援求助、经验分享等更专业的内容。平台内设有私信和群聊，方便志愿者、救助组织等多方沟通。所有发帖和评论内容需经过系统审核，敏感内容自动屏蔽，确保社区环境健康。

3.3.2 领养服务模块

领养服务模块为每只待救助动物建立详细档案，包括照片、基础信息、健康状况、救助记录等。用户可浏览动物信息，认证用户可提交领养申请。系统自动校验申请信息，管理员审核后决定是否通过。平台记录每一次领养的申请、审核和后续回访，方便追踪动物去向。

3.3.3 志愿活动模块

志愿活动模块支持查看动物每日进食信息并上传投喂记录。机构或个人发起志愿活动，填写活动主题、时间、地点和要求，系统自动生成活动页面。志愿者可报名参加，活动组织方可分配任务、统计服务时长。活动结束后，志愿者可对活动进行评价，平台自动记录每位志愿者的服务记录。

3.3.4 个人中心模块

个人中心模块集中展示用户的基本信息、认证状态、发帖记录、领养申请进度和志愿服务历史。用户可在此修改资料、查看消息、管理关注对象。涉及隐私和重要操作时，系统要求二次验证，保障数据安全。

3.3.5 后台管理模块

后台模块为管理员提供用户、动物信息、帖子、活动等内容的统一管理。管理员可进行内容审核、违规处理、数据统计等操作。

3.3.6 信息安全模块

平台采用多种安全措施，包括用户登陆注册、身份认证、权限控制等。所有操作均有日志备份，防止数据丢失和泄露，确保用户信息和平台数据安全。

3.4 流浪动物救助平台原型设计

系统用了模块化的设计思路搭建原型，原型包含了五个核心界面，分别是社区论坛、领养中心、发布中心、消息中心和个人中心。通过简单直接的方式，展现了平台的基本架构和主要功能，为接下来的开发提供了明确的框架和方向。

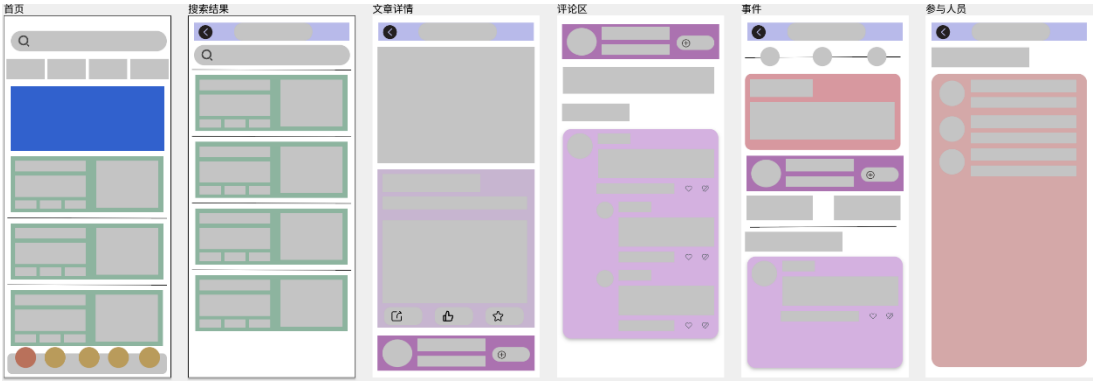

图 3.6 社区论坛低保真原型图

社区论坛页面设计见图 3.6，该界面以列表形式呈现各类帖子，包含显著的标题、发帖人信息、发帖时间及内容摘要等要素。未注册访客仅可浏览公开帖子，注册用户则可点击进入帖子详情页进行评论与点赞操作。页面底部设有发帖入口，此功能仅对注册用户可见，且完成实名认证的用户可发布救援求助、经验分享等更多类型的帖子。此外，页面设置搜索框，用户可通过关键词检索快速定位目标帖子。

图 3.7 领养中心低保真原型图

领养中心页面设计见图 3.7，该界面主要展示待领养动物信息，以卡片形式呈现个体档案，包含动物照片、基础属性（如品种、年龄、健康状况等）及领养条件等核心要素。用户点击卡片可查看包含喂养记录在内的详细资料，经平台认证的用户可在详情页提交领养申请。页面顶部设置筛选功能模块，支持按推荐优先级、地域范围等条件对展示内容进行过滤，以提升用户获取目标信息的效率。

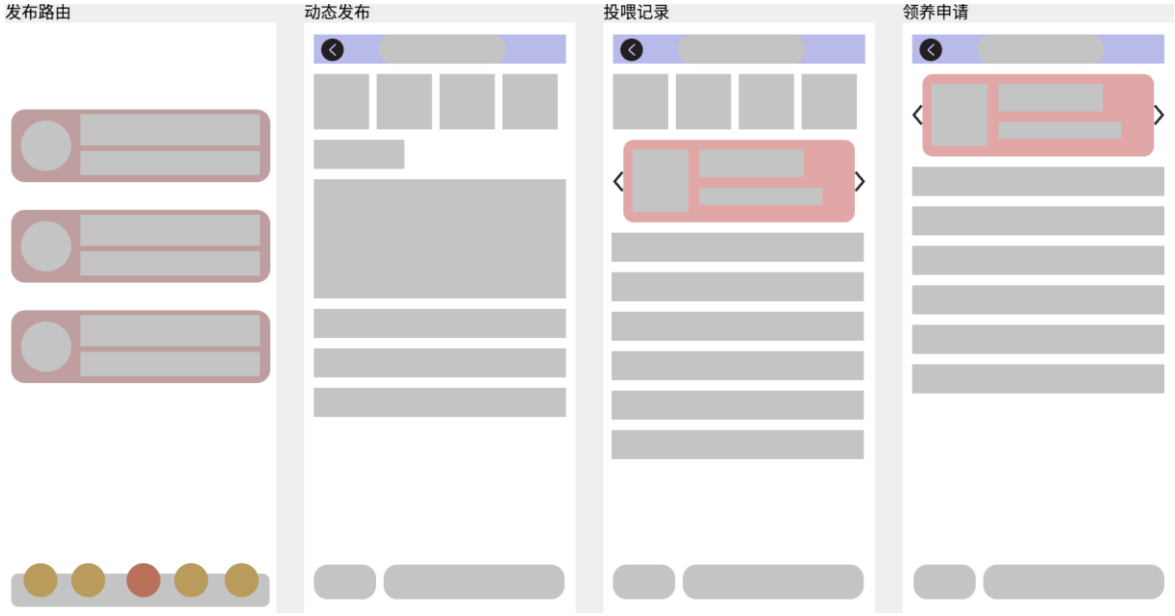

图 3.8 发布中心低保真原型图

发布中心界面设计见图 3.8，该模块仅向完成认证的机构及个人开放，用于发布志愿活动或救援求助信息。发布页面设置多组输入字段，涵盖活动/求助标题、具体内容、时间要求、地点信息、参与条件等核心要素，支持上传图片或文件作为辅助说明材料。信息填写完成后，

用户点击“发布”按钮提交内容，提交后需经平台管理员审核方可公开展示。

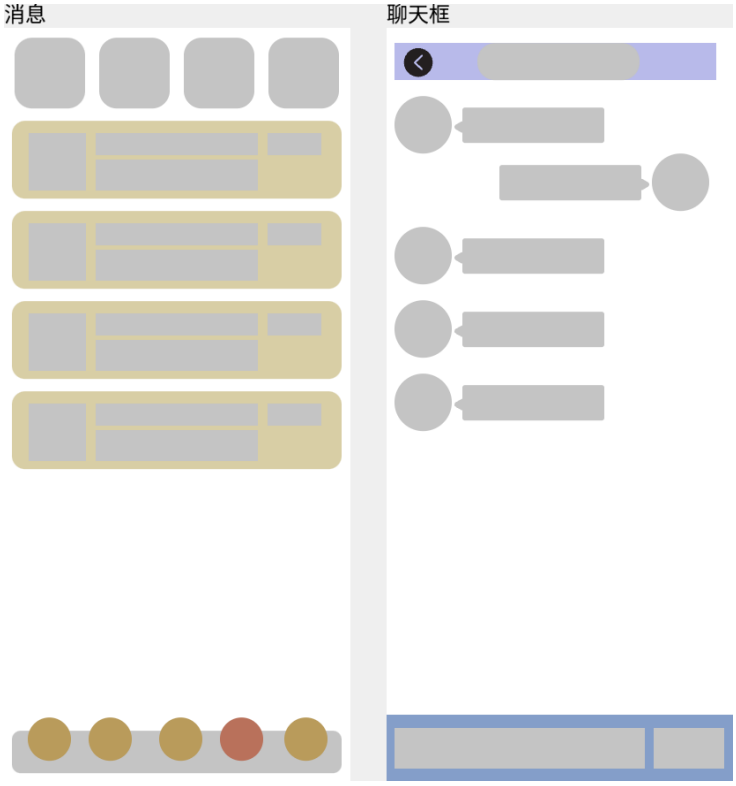

图 3.9 消息中心低保真原型图

消息中心界面设计见图 3.9，该模块以列表形式集中展示系统通知、用户私信及评论回复等信息，每条消息均标注发送者身份、内容摘要及接收时间。用户点击消息条目可查看具体内容，针对私信消息，可在详情页直接进行回复操作，以此实现用户对多类型信息的高效管理与快速响应。

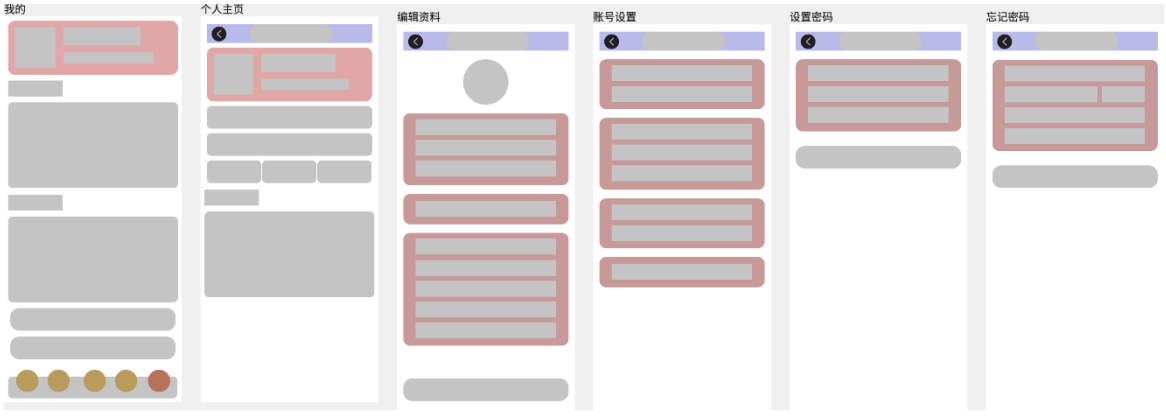

图 3.10 个人中心低保真原型图

个人中心页面设计见图 3.10，该模块集成用户资料管理、发帖记录追溯、志愿活动参与情况统计及领养申请状态跟踪等功能。在用户资料展示区域，系统默认显示头像、昵称、认证状态等核心信息，并提供编辑入口以便用户实时更新个人资料。发帖记录以列表形式呈现，包含帖子标题、发布时间等索引信息，点击单条记录可跳转至详情页进行内容查看或管理操

作（如删除、修改等）。志愿活动模块详细记录用户参与的活动名称、累计服务时长及平台评价反馈，形成可视化的公益服务档案。领养申请区域则动态展示申请项目的审核进度、处理状态等关键节点信息，便于用户及时掌握流程动态。

4　流浪动物救助平台的实现

4.1　用户身份认证与权限管理

为系统具有良好的易用性，使用户操作更加便捷，本系统实现多模态身份认证体系，交互流程见图 4.1。包含短信动态验证码、传统密码认证及第三方 OAuth2.0 联合认证三重机制。

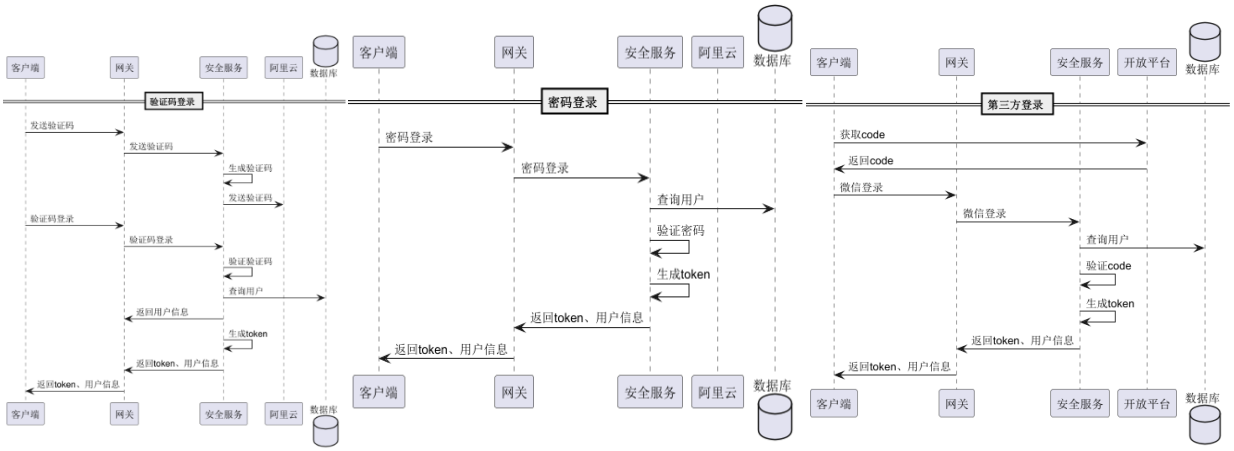

图 4.1 流浪动物救助平台登录交互流程

4.1.1　手机验证码生成

为保证每次验证码都具备唯一性，该系统采用 TOTP(Time-based One-Time Password)算法生成 6 位数字验证码，核心计算过程如式(4.1)所示：

$$CODE = (Hash(K_{seed} \oplus (T_{epoch} \gg 30) \oplus N_{nonce}))mod10^6 \quad (4.1)$$

其中K_{seed}为服务器预置密钥，T_{epoch}为当前 UNIX 时间戳（秒级精度），N_{nonce}为安全随机数。系统通过阿里云 SMS 服务实现端到端 TLS 加密传输，并采用 Redis 键值存储实现验证码状态管理（TTL=300s）。

针对暴力破解风险，实施滑动窗口频控策略（阈值：3 次/300s），超出阈值后触发 Geetest 行为验证，通过人机识别模型区分正常用户与自动化脚本。

4.1.2　第三方联合认证

鉴于小程序依赖于微信平台，该系统实现通过微信、QQ 账号等第三方联合认证的方式进行登录。针对微信与 QQ 的 OpenID 差异问题，设计双路径处理机制进行实现，映射关系如下：

$$ID_{union} = SHA256(OpenID||PlatformCode) \quad (4.2)$$

微信平台：直接获取ID_{union}作为唯一标识。

QQ 平台：通过 OpenID 和平台编号 PlatformCode 生成复合标识

4.1.3　会话管理

为保障系统的安全性进行会话管理，系统基于采用 JWT(JSON Web Token)实现无状态鉴权。

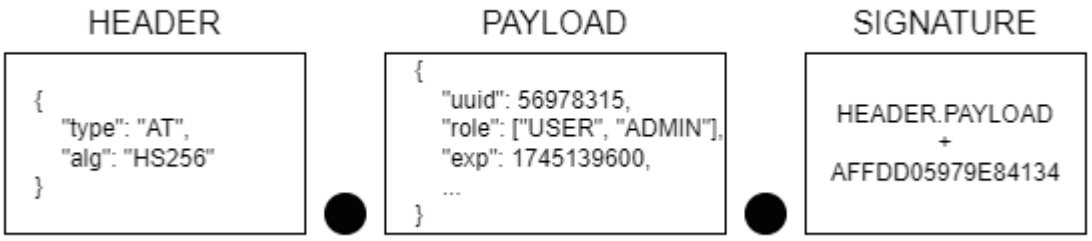

图 4.2 JWT 组成示意图

JWT 由三个部分组成(如图 4.2)：头部(HEADER)，载荷(PAYLOAD)，签证(SIGNATURE)。本系统在 JWT 头部声明 token 存储类型以及加密算法；载荷部分存储用户的唯一识别码、用户角色、过期时间；签证部分由头部(HEADER)，载荷(PAYLOAD)以及密钥(SECRET)组成，生成算法如（）所示：

$$token = Base64(HEADER).Base64(PAYLOAD).Base64(\text{SIGNATURE}) \quad (4.3)$$

$$\text{SIGNATURE} = SHA256(HEADER.PAYLOAD + SECRET) \quad (4.4)$$

为提高用户的交互体验，避免反复进行登录认证，该系统采用双 token 机制（如图 4.3）保障 OAuth 流程安全性。

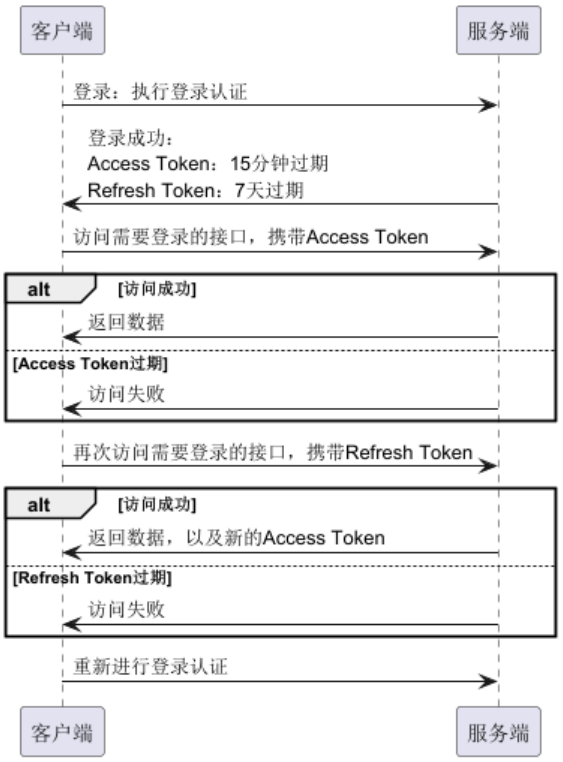

图 4.3 AT-RT 双 token 流程图

AccessToken(AT)短期有效（15min），用于业务请求

RefreshToken(RT)长期有效（7d），通过$uni.setStorageSync("rToken",rtoken)$进行存储

当 AT 过期时，客户端提交 RT 换取新 AT，直至 RT 过期需重新登录认证。

4.2 宠物信息管理与投喂记录

对宠物的投喂信息共享是本系统核心功能之一，为确保上传的投喂信息具备较高的可信性，防止对真实数据的污染，本系统基于多模态数据融合技术构建可信投喂验证体系，通过数据采集约束、服务端验证、管理员复检机制保障数据真实性。

4.2.1 数据采集约束

在移动端数据采集阶段，系统采用双重验证策略确保数据源可信性。

空间维度上，通过微信小程序$uni.getLocation()$接口获取 WGS-84 坐标系定位，应用地理围栏(Geo-fencing)技术限制操作范围，定义有效区域以宠物等级坐标为中心、半径 100 米的球冠区域，数学函数如（4.5）所示，运算表达式如（4.6）所示：

$$Havrsine(\theta) = \sin\frac{\theta}{2} \tag{4.5}$$

$$Havrsine(\Delta\phi) + \cos\phi_1 \cdot \cos\phi_2 \cdot Havrsine(\lambda_2 - \lambda_1) \leq \sin^2\frac{0.0015\pi}{2} \tag{4.6}$$

其中ϕ_1,ϕ_2为两点纬度，λ_1,λ_2为两点经度，0.0015π对应 100 米弧长(地球半径取 6371km)。

时间维度上，系统建立 ARIMA(0,1,1)时序模型检测异常行为，设置滑动窗口（窗口大小 W=1800s，滑动步长=300s），当窗口内投喂次数超过阈值 N=5 时触发验证流程并标记投喂数据。

图像采集环节采用设备硬件级验证方案，通过$uni.chooseImage()$API 强制获取实时影像数据，保证上传的图像都是实时拍摄。

4.2.2 服务端验证

服务端会对投喂记录的请求进行二次验证，包含空间一致性验证和资源冲突检测。

空间一致性验证基于改进 Haversine 公式（如 xx 所示）计算投喂点与宠物活动中心的球面距离，引入高度补偿因子(δ)

$$\delta = 2R \cdot \arcsin\left(\sqrt{Haversine(\theta) + \frac{(h_1-h_2)^2}{4R^2}}\right) \tag{4.7}$$

其中 R 为地球半径(6371km)，h_1,h_2为海拔高度(通过 Google Elevation API 获取)。

资源冲突检测采用 Redlock 分布式锁机制实现投喂记录排他性写入，锁过期时间 T=10s，避免并发场景下的数据覆盖问题，同时采用 CAS(Compare-And-Swap)机制确保原子性写入。

并对存在验证风险的记录生成告警事件，通过消息队列（RabbitMQ）向管理员推送弹窗

提醒。

4.2.3 管理员复检

针对存在风险并已上传的投喂记录，本系统实现了数据屏蔽功能。管理员通过后台管理系统对投喂信息进行复检，若存在问题则操作使信息不可见，并对投喂用户发出警告消息或直接冻结该用户。

4.3 社区互动与实时通信

用户参与志愿活动时要确保志愿者之间消息沟通的实时性及有序性，针对这一问题本系统基于分布式消息中间件架构设计，采用发布-订阅模式实现用户间实时通信，通过消息持久化与状态同步机制实现。

4.3.1 实时通信机制

用户在线时，系统通过 WebSocket 协议建立长连接，确保消息能够实时推送至在线用户。消息帧采用 STOMP（Simple Text Oriented Messaging Protocol）子协议进行封装，使消息格式标准化、易于解析。设用户集合为 U，当前在线用户为U_{on}，则有：

$$U_{on} = \{u_i | WebSocket_i = connected, u_i \in U\} \tag{4.8}$$

客户端周期性地（每 30 秒）发送心跳包（PING），服务端接收心跳并实时更新用户的在线状态。在线状态维护采用 Redis Cluster，用户状态 S 的更新表达式如下（Δt为心跳间隔 30s）：

$$S(u_i, t) = \begin{cases} 1, 若在[t - \Delta t，\ t]区间内收到心跳 \\ 0, 否则 \end{cases} \tag{4.9}$$

4.3.2 离线消息处理

用户离线时，系统自动维护消息内容，在 Redis 缓存中维护用户拉取到的最新一条消息的 id，同时将消息进行数据库持久化，保障用户消息时序一致性。消息持久化流程：

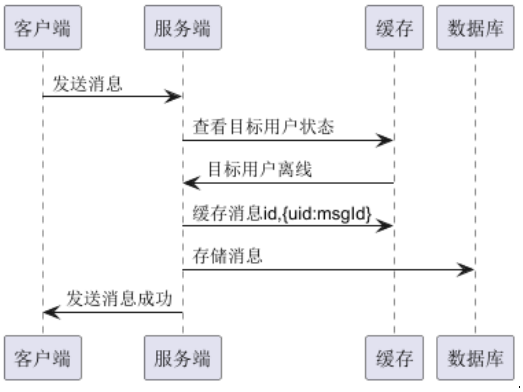

图 4.4 离线消息发送交互图

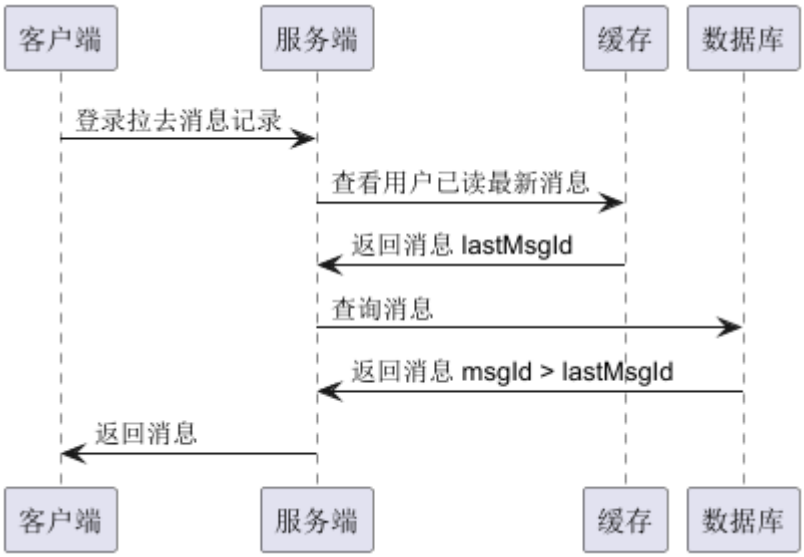

图 4.5 离线消息更新交互图

由于数据库中消息的存储已经按序存储，用户重新登录进行消息同步时直接从数据库读取即可。

4.4 领养申请与审核流程

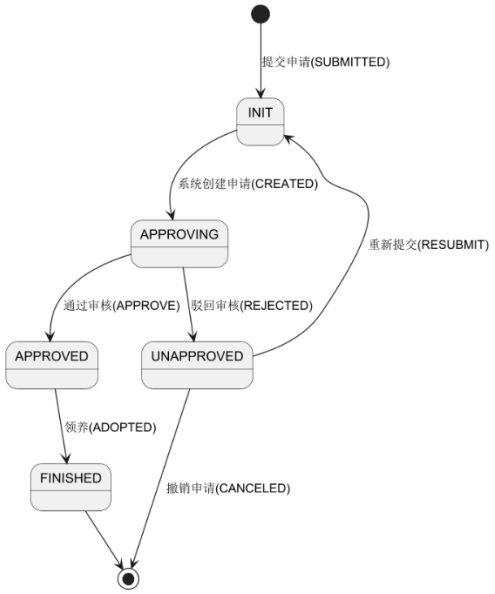

图 4.6 领养申请状态图

基于有限状态自动机(FSM)理论构建领养流程状态模型(如所示),定义申请初始化(INIT)、审核中(APPROVING)、审核通过(APPROVED)、审核未通过(UNAPPROVED)、申请完成(FINISHED)五种状态。状态迁移由事件触发，定义提交申请(SUBMITTED)、创建申请(CREATED)、通过审核(APPROVE)、驳回申请(REJECTED)、重新提交(RESUBMIT)、撤销申请 (CANCELED)、领养(ADOPTED)六种事件。

同时，系统采用责任链模式设计通知服务，当状态发生变更时，依次触发短信（阿里云

SMS)、站内信、邮件（JavaMail API）通知。在用户端提供可视化进度条，展示当前审核阶段。

4.5 后台管理

后台管理的实现包括用户与权限管理、动物与救助数据管理、社区内容审核、数据统计与可视化。

4.5.1 用户与权限管理

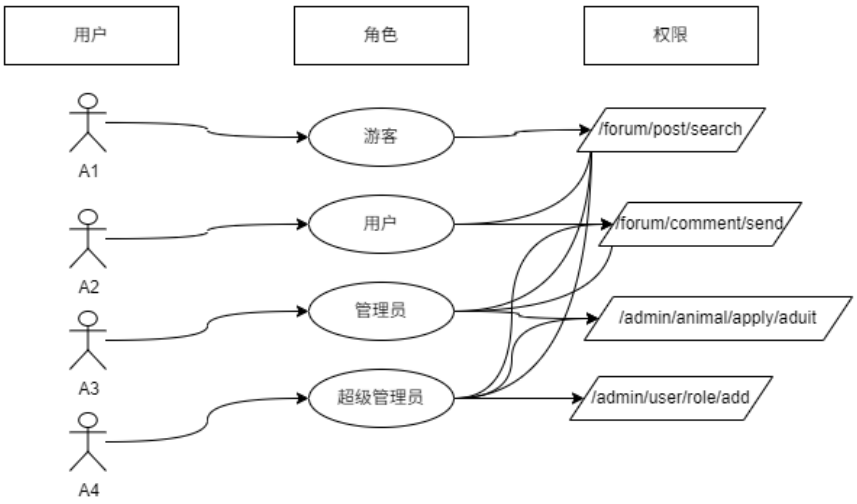

图 4.7 流浪动物救助平台 RBAC 模型示例图

系统采用基于角色的访问控制（RBAC）模型，定义四级角色体系：游客、普通用户、管理员、超级管理员。管理员具备用户信息审核、动物档案管理、论坛内容管控等权限；超级管理员可配置角色权限策略，实现接口级权限分配（如 /admin/audit 仅对超级管理员开放）。

基于领域层的 UserService，在用户登录时通过 JWT Payload 动态注入角色列表与权限范围，结合网关层的 Gateway 实现请求链路的权限拦截。

4.5.2 动物与救助数据管理

动物数据管理。管理员通过 Web 管理端录入动物基础信息（品种、健康状态、坐标位置等），调用/admin/animal/add 接口完成数据持久化。采用乐观锁机制（通过表字段 version 实现）解决并发修改冲突，确保数据一致性。

领养申请审核。管理员对领养申请（adopt_apply 表）执行三级审核：系统自动校验材料完整性(如身份证、居住信息)、志愿者实地家访评估、管理员终审匹配。状态变更通过 RabbitMQ 消息队列通知申请人，更新结果同步至 adopt_audit_log 表。

监管投喂记录。针对志愿者提交的投喂数据（feed_his 表），管理员可触发二次复检流

程：通过后台管理系统查看投喂数据详细信息，对风险数据执行屏蔽或标记处理，确保救助记录的真实性。

4.5.3 社区内容审核

为维护平台健康生态，系统设计“自动化过滤+二次复检+人工复核”三层审核机制实现高效内容管控：

预过滤阶段：用户发布帖子或评论时，前端通过 axios 访问/peace/prechek 接口，基于 Redis 缓存的一级敏感词库进行实时过滤，拦截违规内容并返回提示信息。

系统复检：对于通过预过滤阶段的内容，系统自动进行二级敏感词库的匹配，对于包含二级敏感词语的内容进行标记并通过 RabbitMQ 通知管理员。

人工复核：管理员通过后台管理端执行人工复核，对于争议内容，支持处理为“无影响”或 “驳回”，结果通过 WebSocket 实时推送至用户端。

违规处理：针对重复发布垃圾信息、恶意评论等行为，系统自动触发梯度处罚机制：首次警告、二次封禁 7 天、三次永久冻结，相关操作同步触发站内信通知。

5　流浪动物救助平台的测试

5.1　功能测试

5.1.1　用户注册与登录功能测试

测试平台支持的多模态身份认证方式，包括短信验证码、传统密码、第三方 OAuth2.0 联合认证。通过模拟正常及异常输入情况，验证系统对用户信息的有效性校验、验证码发送与过期处理、第三方登录流程的正确性，以及会话鉴权机制（JWT 双 token 机制）的安全性和稳定性。

5.1.2　宠物信息管理与投喂功能测试

重点测试动物信息的录入、展示、修改、删除等操作流程，确保管理员与志愿者能够高效录入和维护动物档案。针对投喂功能，验证地理围栏、时间窗口约束、实时影像上传、数据一致性校验等机制的有效性。通过模拟多用户同时投喂场景，检验系统对并发写入的处理能力及数据的准确性。

5.1.3　社区互动与论坛交流功能测试

测试用户在社区论坛中的发帖、评论、点赞、私信、群聊等操作流程，涵盖普通访客、注册用户和实名认证用户的不同权限。重点关注内容发布的合规性校验、敏感词过滤、实时消息推送与离线消息同步机制。

5.1.4 领养申请与审核流程测试

对领养中心的动物信息浏览、领养申请提交、材料上传与审核结果反馈等功能进行全流程测试，模拟不同类型用户的操作路径。测试平台对申请材料的完整性校验、审核环节的权限分配、状态跟踪及通知推送功能。

5.1.5　后台管理功能测试

针对管理员端的用户管理、动物信息管理、内容审核等功能进行全面测试。验证敏感操作的二次确认机制、违规内容的自动处理与人工复核流程、数据统计报表的准确性与实时性。

5.2　系统测试

系统测试要从整体上评估系统的性能、安全性以及易用性等方面，保证系统在实际运行环境里稳定又可靠。

5.2.1　响应时间测试

图 5.1 响应测试结果

采用 ApiFox 工具进行压力测试，模拟用户从少量逐步增加到 200 个并发量的场景，测试内容包括资料查询、发帖和领养申请等核心功能。测试结果表明：系统在常规使用下平均响应时间为 0.2 秒；在 200 并发的高负载情况下，响应时间仍能保持在 1 秒以内，完全满足性能需求，体现了良好的负载能力。

5.2.2 吞吐量测试

图 5.2 吞吐量测试结果

本系统设定 100 并发用户进行压力测试，模拟浏览页面、提交表单等典型操作。测试结果显示，系统每分钟可处理约 4000 次有效请求，表明当前处理能力完全满足预期业务需求，并具备良好的扩展潜力。。

5.2.3 易用性测试

系统易用性评估采用问卷调查和用户访谈相结合的方法，选取不同年龄、技术水平和需求的用户群体作为样本。调查结果表明：系统界面设计合理、导航清晰、操作逻辑直观，新用户经过简单引导即可完成浏览帖子、查询动物信息等基本操作，验证了系统具有良好的易用性和用户适应性。

5.2.4 安全性测试

本系统主要通过 SQL 注入和 XSS 跨站脚本进行攻击的模拟。使用 SQLMap 输入恶意 SQL 语句，验证系统对数据库的防护能力；在用户输入框中插入恶意脚本，检测系统对输入内容的过滤和转义效果。测试结果显示，系统未发现明显安全漏洞，具备较强的安全防护能力。

5.3 测试结论

平台各项核心功能均实现了设计目标。用户注册、登录、动物信息管理、社区互动、领养申请等模块功能完善，操作流程顺畅，能够满足流浪动物救助的实际业务需求。系统采用多级审核、权限分级、内容过滤等机制，保障了数据的准确性和平台运行的规范性。后台管理端支持高效数据管理和操作日志追踪，便于平台的日常维护和决策支持。

平台在性能、易用性和安全性方面表现良好。高并发测试下系统响应迅速，具备良好的扩展性和稳定性。界面设计简洁，用户易于上手，操作体验友好。安全测试未发现严重漏洞，数据加密、权限控制等措施有效保障了用户和平台数据的安全。整体来看，系统稳定可靠，能够适应实际应用环境的多样化需求。

6 总结与期望

6.1 项目总结

本研究围绕流浪动物救助场景的数字化需求，设计并实现了基于微服务架构的流浪动物救助平台系统。平台采用分层架构设计（用户接口层 / 应用层 / 领域层 / 基础设施层），集成 Uni-App 跨平台开发、Spring Boot 业务逻辑处理、Dubbo 服务通信及 MySQL+Redis 数据存储等技术栈，构建了宠物领养管理、社区互动、志愿活动组织、后台数据管控等核心功能模块。通过多级领养审核流程自动化、投喂记录地理围栏验证、RBAC 角色权限控制等机制，实现了救助业务流程的标准化与数据管理的安全性。系统测试表明，平台在 200 并发场景下响应时间保持在 1 秒以内，功能流程完整，数据校验机制有效，满足实际业务负载需求。

6.2 未来展望

未来平台优化将聚焦技术迭代与功能扩展：一方面，引入计算机视觉技术实现动物影像智能识别、自然语言处理技术优化内容审核效率，提升平台自动化处理能力；另一方面，推动与政府监管系统、宠物医疗平台的数据接口对接，构建“救助-医疗-领养-监管”全链路生态闭环。同时，通过区块链技术实现救助记录上链存证、VR 技术打造沉浸式领养体验场景，进一步强化平台的技术赋能价值，使其成为流浪动物救助[21],[22]领域高效、可靠的数字化基础设施。

参考文献

[1] 高祝宇,韩颂雨,杨明,等.基于 Spring MVC 的气象预警信息 Web 系统设计与实现[J].计算机与网络,2020,46(12):61-63.

[2] Wenjuan Shao, Kun Liu.Design and Implementation of Online Ordering System Based on SpringBoot[J].Journal of Big Data and Computing,2024,2(3):

[3] Yixuan Liu.Design and Implementation of a Student Attendance Management System based on Springboot and Vue Technology[J].Frontiers in Computing and Intelligent Systems,2024,8(1):91-97.

[4] 兰旭辉,熊家军,张海燕.基于 MySQL 的应用程序开发[J].空军雷达学院学报,2003,(02):59-61.

[5] Chul-Ho Kim,Kyeong-Won Park,Yong-Lak Choi.Web Service Performance Improvement with the Redis[J].Journal of the Korea Institute of Information and Communication Engineering,2015,19(9):2064-2072.

[6] Workman Miranda K,Hoffman Christy L.An Evaluation of the Role the Internet Site Petfinder Plays in Cat Adoptions.[J].Journal of applied animal welfare science : JAAWS,2015,18(4):388-97.

[7] Qi Zhang, Shulin Yang, Ruoyu Ren.Research on Uni-app Based Cross-platform Digital Textbook System[A]Proceedings of the 2020 3rd International Conference on Computer Science and Software Engineering (CCSE 2020)[C]. International Association of Applied Science and Engineering, 成都青恒景逸会务服务有限公司, 2020: 6.

[8] Wenjuan Shao, Kun Liu.Design and Implementation of Online Ordering System Based on SpringBoot[J].Journal of Big Data and Computing,2024,2(3):

[9] 张晓梅.图书馆微信小程序应用研究[J].传媒论坛,2020,3(03):93-94.

[10] 姜苏.基于互联网平台解决社会流浪动物问题的可行性研究[D]. 山东农业大学, 2020.

[11] 王亚丽,黄一格,吴琦琦,赵文涛,顾峻瑄,游嘉靖.基于uniapp搭建的助推非遗平台的设计与实现[J].科学技术创新,2024,(20):101-104.

[12] 李亮,舒畅.微服务架构与容器化技术的软件开发实践[J].物联网技术,2024,14(05):64-

67.

[13] 雷思雨, 武佳雪, 胡月馨.流浪动物救助站的现状及对策研究[A]“劳动保障研究”2024研讨会论文集（上册）[C]. 成都信息工程大学管理学院, 成都信息工程大学管理学院, 2024: 4.

[14] 邢月,喻德荣.简析流浪动物救助实践困境与路径优化[J].大众标准化,2021,(04):71-73.

[15] PetRescue UK and Ireland aiming to protect and rehome UK 'Easter Bunnies' with launch of new adoption platform[J].M2 Presswire,2021,

[16] 刘权.MVC 架构下高校餐饮采购管理系统的设计与开发[J].网络安全和信息化,2025,(03):85-87.

[17] 孙成,刘海燕.基于 Web Service 分层架构的实践教学检查管理系统研究[J].中国教育信息化,2019,(21):44-47.

[18] 李忠毅.基于 SpringBoot 的小型日常交流论坛的设计与实现[J].现代计算机,2020,(25):105-108.

[19] 赵子晨,朱志祥,蒋来好.构建基于Dubbo框架的Spring Boot微服务[J].计算机与数字工程,2018,46(12):2539-2543+2551.

[20] 曾超宇,李金香.Redis在高速缓存系统中的应用[J].微型机与应用,2013,32(12):11-13.

[21] 张秋雨.流浪动物救助实践困境与路径优化——基于四川省宜宾市的实证分析[J].法制与社会,2017,(15):188-190.

[22] 纪好.流浪动物救助公益服务设计研究[D]. 武汉理工大学, 2018.