****

|  |  |
| --- | --- |
| **成绩** | （采用四级记分制） |

**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**

目前我国流浪动物传统救助模式面临着诸多问题，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。

该平台采用Spring Boot为核心后端框架，结合Dubbo实现服务间解耦与高效通信，前端采用Uni-App实现多端适配，系统整体分为用户接口层、应用层、领域层和基础设施层，提供宠物信息登记、档案管理、投喂记录追踪、领养申请与审核、社区交流互动、志愿活动发布与管理、后台数据统计与内容审核等核心功能。平台引入JWT无状态鉴权、第三方联合认证、人机识别与频控策略用于保障系统安全性；采用WebSocket与STOMP协议实现用户间实时通信，Redis集群维护用户在线状态与消息缓存，RabbitMQ消息队列处理异步任务，提升系统性能与响应速度。领养流程基于有限状态自动机（FSM）理论设计，配合责任链模式实现多渠道通知，确保流程规范透明。投喂记录模块采用地理围栏校验、ARIMA时序模型异常检测、Redlock分布式锁与CAS机制，保证数据的真实性和一致性。后台管理系统支持分级权限控制、数据可视化与内容审核，提升平台运营效率。

**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**

[**摘 要** I](#_Toc196088589)

[**ABSTRACT** II](#_Toc196088590)

[1 绪论 1](#_Toc196088591)

[1.1 应用背景及意义 1](#_Toc196088592)

[1.2 国内外应用现状 1](#_Toc196088593)

[1.2.1 国外应用现状 1](#_Toc196088594)

[1.2.2 国内应用现状 1](#_Toc196088595)

[1.3 论文结构安排 2](#_Toc196088596)

[2 流浪动物救助平台需求分析 4](#_Toc196088597)

[2.1 流浪动物救助平台功能性需求 4](#_Toc196088598)

[2.1.1 普通用户功能 4](#_Toc196088599)

[2.1.2 志愿者功能 4](#_Toc196088600)

[2.1.3 管理员功能 5](#_Toc196088601)

[2.2 流浪动物救助平台非功能性需求 5](#_Toc196088602)

[2.2.1 性能需求 5](#_Toc196088603)

[2.2.2 安全性需求 5](#_Toc196088604)

[2.2.3 易用性需求 5](#_Toc196088605)

[2.2.4 可靠性需求 5](#_Toc196088606)

[2.3 流浪动物救助平台的可行性 6](#_Toc196088607)

[3 流浪动物救助平台的设计 7](#_Toc196088608)

[3.1 流浪动物救助平台架构设计 7](#_Toc196088609)

[3.1.1 用户接口层 7](#_Toc196088610)

[3.1.2 应用层 8](#_Toc196088611)

[3.1.3 领域层 8](#_Toc196088612)

[3.1.4 基础设施层 8](#_Toc196088613)

[3.2 流浪动物救助平台数据库设计 9](#_Toc196088614)

[3.2.1 数据库模型 9](#_Toc196088615)

[3.2.1 实体关系与ER图设计 10](#_Toc196088616)

[3.2.2 数据表结构设计示例 11](#_Toc196088617)

[3.3 流浪动物救助平台功能设计 12](#_Toc196088618)

[3.3.1 社区交流模块 12](#_Toc196088619)

[3.3.2 领养服务模块 12](#_Toc196088620)

[3.3.3 志愿活动模块 12](#_Toc196088621)

[3.3.4 个人中心模块 12](#_Toc196088622)

[3.3.5 后台管理模块 13](#_Toc196088623)

[3.3.6 信息安全模块 13](#_Toc196088624)

[3.4 流浪动物救助平台原型设计 13](#_Toc196088625)

[4 流浪动物救助平台的实现 17](#_Toc196088626)

[4.1 用户身份认证与权限管理 17](#_Toc196088627)

[4.1.1 手机验证码生成 17](#_Toc196088628)

[4.1.2 第三方联合认证 17](#_Toc196088629)

[4.1.3 会话管理 17](#_Toc196088630)

[4.2 宠物信息管理与投喂记录 19](#_Toc196088631)

[4.2.1 数据采集约束 19](#_Toc196088632)

[4.2.2 服务端验证 19](#_Toc196088633)

[4.2.3 管理员复检 20](#_Toc196088634)

[4.3 社区互动与实时通信 20](#_Toc196088635)

[4.3.1 实时通信机制 20](#_Toc196088636)

[4.3.2 离线消息处理 20](#_Toc196088637)

[4.4 领养申请与审核流程 21](#_Toc196088638)

[4.5 后台管理 22](#_Toc196088639)

[4.5.1 用户与权限管理 22](#_Toc196088640)

[4.5.2 动物与救助数据管理 22](#_Toc196088641)

[4.5.3 社区内容审核 23](#_Toc196088642)

[5.1 功能测试 24](#_Toc196088643)

[5.1.1 用户注册与登录功能测试 24](#_Toc196088644)

[5.1.2 宠物信息管理与投喂功能测试 24](#_Toc196088645)

[5.1.3 社区互动与论坛交流功能测试 24](#_Toc196088646)

[5.1.4 领养申请与审核流程测试 24](#_Toc196088647)

[5.1.5 后台管理功能测试 24](#_Toc196088648)

[5.2 系统测试 24](#_Toc196088649)

[5.2.1 响应时间测试 24](#_Toc196088650)

[5.2.2 吞吐量测试 25](#_Toc196088651)

[5.2.3 易用性测试 25](#_Toc196088652)

[5.2.4 安全性测试 26](#_Toc196088653)

[5.3 测试结论 26](#_Toc196088654)

[6 总结与期望 27](#_Toc196088655)

[6.1 项目总结 27](#_Toc196088656)

[6.2 未来展望 27](#_Toc196088657)

[参考文献 28](#_Toc196088658)

1 绪论

1.1 应用背景及意义

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目前我国流浪动物传统救助模式面临的管理缺乏科学性、领养渠道不畅、公众认知不足等诸多挑战。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。

该平台设计过程中，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、访谈等方式收集用户需求并分析，根据分析的结果设计系统平台的整体框架和具体功能，最后，实现系统功能并展示出来。文章分为六章，结构安排如下：

第一章为绪论，介绍流浪动物救助平台的应用背景、意义及国内外相关应用现状[13],14],15]，明确论文研究目标。

第二章为需求分析，基于问卷调查、用户访谈和实地观察，梳理平台主要用户角色，明确功能性和非功能性需求，分析系统可行性。

第三章为系统设计，依据需求分析结果，提出平台的整体架构，细化各功能模块，设计数据库结构，明确各模块间的交互关系。

第四章为系统实现，详细说明平台各核心模块的具体实现方法，包括安全模块、用户模块、投喂记录、社区互动、领养流程及后台管理等。

第五章为系统测试，采用功能测试和系统测试相结合的方法，对平台的各项功能、性能、安全性和易用性进行全面验证，并分析测试结果。

第六章为总结与展望，总结平台建设成果，分析存在的问题与不足，提出后续优化方向。

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，分析出各类角色的功能，用例图如图2.1所示。

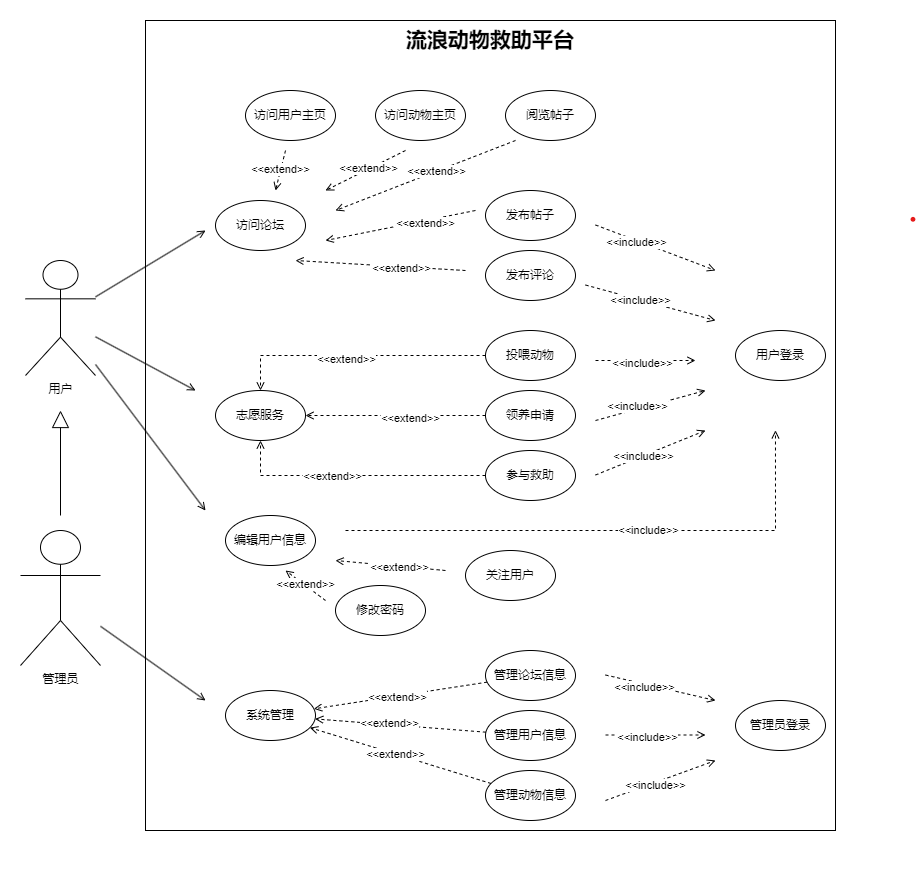


图2.1 用例图

2.1.1 普通用户功能

信息浏览与查询：支持用户浏览流浪动物基本信息、救助动态、领养公告等内容。

领养申请：用户可在线提交领养申请，上传相关证明材料，实时跟踪审核进度。

投喂记录上传：用户可上传动物投喂记录，系统通过地理围栏和实时影像验证确保信息真实性。

社区互动：注册用户可参与社区交流，发布帖子、评论和参与话题讨论，促进信息共享。

个人信息管理：支持用户对个人资料、联系方式等信息进行管理与维护。

2.1.2 志愿者功能

志愿活动报名与管理：志愿者可查看、报名参与平台发布的救助活动，管理个人活动记录。

动物健康档案维护：协助录入、维护救助动物的健康信息与救助进展。

救助反馈与沟通：通过平台与其他志愿者、管理员实时沟通，反馈救助进展与遇到的问题。

2.1.3 管理员功能

动物与用户信息管理：对动物档案、投喂记录、用户信息进行增删改查操作。

领养流程审核：对领养申请进行多级审核，确保流程规范与动物福利。

社区内容审核：对社区发布的内容进行审核、屏蔽或删除，维护平台健康环境。

权限与安全管理：分级管理用户权限，保障系统安全运行。

2.2 流浪动物救助平台非功能性需求

2.2.1 性能需求

支持高并发访问，保证在高并发用户场景下响应时间不超过2秒。

2.2.2 安全性需求

系统如果被攻击，应保证数据将不会发生丢失的现象。

2.2.3 易用性需求

用户界面应该支持多端适配，提供简洁明了的操作界面和流程指引。

2.2.4 可靠性需求

系统部分组件发生故障时，能够自动进行故障转移和恢复。

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。

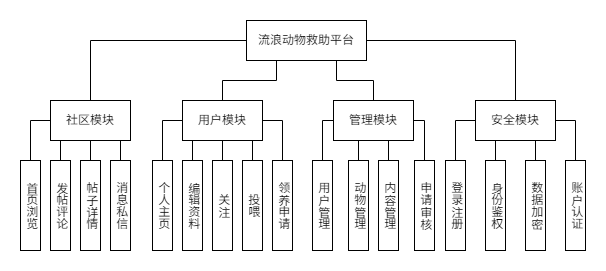


图3.5 流浪动物救助平台功能树

3.1.1 社区模块

社区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。

3.1.2 用户模块

用户模块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。

3.1.3 管理模块

管理模块主要为管理员提供用户管理、动管理、内容管理、申请审核功能。用户管理包括增删用户，更改用户权限等操作；动物管理包括编辑动物信息，增删动物等；内容管理包括帖子内容检查、评论内容检查、屏蔽内容等；申请审核包括领养申请审核、身份认证审核等。

3.1.4 安全模块

安全模块提供用户登录注册功能，负责对用户权限的鉴别，对向外提供的数据进行明文加密，处理实名认证等相关账户认证的请求。

3.2 流浪动物救助平台架构设计

根据需求分析结构及功能设计，平台采用微服务架构与层次结构风格设计[16,17]。整体架构如图所示，分为用户接口层、应用层、领域层和基础设施层。

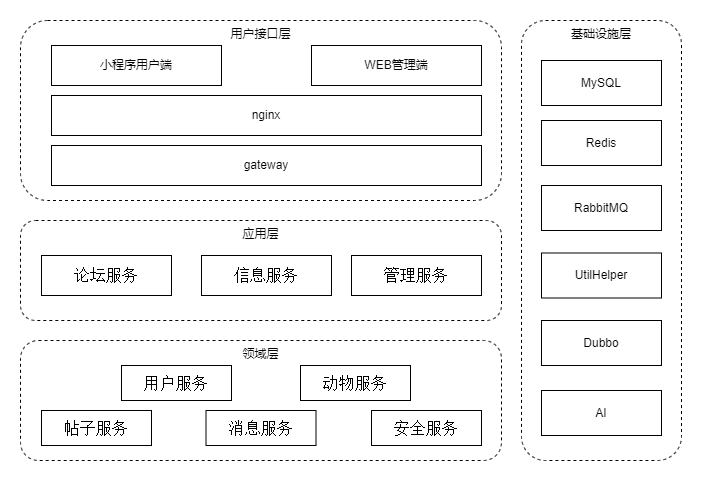


图3.1 系统架构图

3.1.1 用户接口层

用户接口层[18]主要负责为终端用户（普通用户、志愿者、管理员）提供交互界面。前端采用Uni-App框架，实现对Web端、移动端等多终端的适配，提升用户体验。接口层通过API网关（Nginx）实现统一入口，进行用户请求的转发、权限鉴权和流量控制，保障系统安全与高效。

3.1.2 应用层

应用层负责协调各业务模块的逻辑处理，实现系统功能的具体业务流程。各服务模块独立部署，采用Dubbo[19]进行服务间通信，实现服务解耦与高可用。平台核心服务包括：

用户服务：负责用户注册、登录、实名认证、权限分配等功能。

动物服务：实现流浪动物信息的登记、健康档案管理、救助记录更新等功能。

领养服务：支持领养申请提交、多级审核、状态跟踪及通知推送，确保领养流程规范透明。

社区服务：实现用户发帖、评论、活动组织、实时消息推送等社区互动功能。

管理服务：为管理员提供用户管理、内容审核、数据统计与可视化等后台管理能力。

3.1.3 领域层

领域层聚焦于平台核心业务逻辑的实现与抽象。根据需求分析结果，平台主要划分为以下功能模块：

动物管理模块：实现动物信息的录入、健康档案维护、救助记录管理等。

领养管理模块：支持领养申请、材料上传、多级审核、进度跟踪、责任链通知等功能。

社区互动模块：支持发帖、评论、活动报名、实时消息推送与离线消息持久化。

投喂记录模块：集成地理围栏校验、ARIMA时序模型异常检测、实时图像采集与验证。

后台管理模块：包含用户及权限管理、内容审核、数据统计与可视化、系统日志管理等。

3.1.4 基础设施层

基础设施层为平台提供底层支撑，包括数据库、缓存、消息队列等。平台采用MySQL作为关系型数据库，存储核心业务数据；Redis[20]集群负责缓存高频数据、维护用户在线状态及消息缓存；RabbitMQ实现异步消息处理。同时，结合Docker和Nginx，实现平台容器化部署。

3.2 流浪动物救助平台数据库设计

3.2.1 数据库概念结构设计

根据需求分析以及功能模块图，本平台需要如下数据实体及其相关属性。

1. 用户数据实体：用户编号，角色编号，认证编号，手机号，密码，账户状态，头像，用户名，性别，出生日期，地区，格言，简介，用户图片，邮箱，微信，QQ，抖音，小红书。

4. 动物数据实体：动物编号，头像，名称，性别，体重，图片，生日，地区，描述，用户编号。

5. 领养申请数据实体：申请编号，居住类型，有无院子，家庭成员数量，现有宠物情况，工作状态，年收入，养宠经验，领养计划，联系人电话，申请状态，用户编号，动物编号。

4. 领养审核日志数据实体：审核编号，申请编号，审核事件，审核意见，审核员编号，审核备注。

5. 帖子数据实体：帖子编号，标题，内容，图片，地区，发帖人编号，动物编号，是否事件。

6. 评论数据实体：评论编号，父评论编号，内容，帖子编号，评论人编号，敏感评论，评论类型。

7. 投喂记录数据实体：记录编号，食物名称，食物克重，食物图片，用户编号，动物编号，地点，详细地址。

8. 聊天数据实体：群聊编号，群名，群头像，消息编号，用户编号，消息类型，消息内容。

9. 活动数据实体：活动编号，帖子编号，事件流程状态，用户编号。

各数据实体之间的关联关系如图所示：

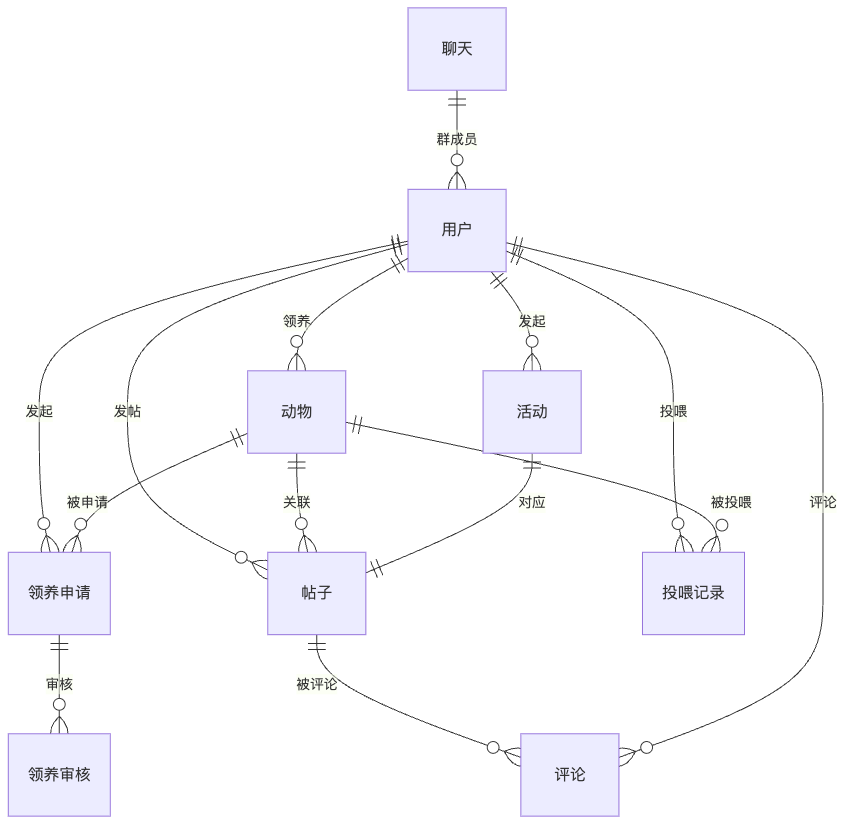


图3.3 数据库ER图

3.2.2 数据库逻辑结构设计

本平台数据库的逻辑结构设计主要包括以下内容：

1. 转换数据模型：将概念设计的每个实体转化为MySql数据库所支持的数据表，如用户账户表、动物账户表。

2. 明确数据完整性约束：为每个数据表都应设置id作为主键，保证实体完整性；通过外键约束关联其他表的主键，保证参照完整性如评论表的post\_id字段与帖子表的id字段关联；根据业务需求设置字段的唯一性、非空等约束保证用户自定义完整性，如用户表的user\_id字段。

3. 设计实体间关联关系：对于1:1或1:N的关系，使用属性记录另一个实体的主键，如动物表中的host\_id；对于N:N关系，通过创建中间表的方式实现，例如用户关注表。

4. 规范化设计：消除数据冗余、非主属性对码的部分函数依赖和传递函数依赖，保证数据存储符合第三范式（3NF）规范。

4. 性能优化：根据业务需求为每张表设置索引，提高数据查询速度。使用Redis为数据库提供缓存功能，减轻数据库压力，保证数据库的可用性。

3.2.1 数据库物理结构设计

根据MySQL数据库InnoDb引擎的特性，本平台数据库表结构如下所示。

表3.1 用户账户表(user\_account)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| user\_id | BIGINT UNSIGNED | 否 | 是 | 用户id（唯一） |
| phone | VARCHAR(20) | 否 | 是 | 手机号 |
| password | VARCHAR(255) | 否 | 否 | 密码 |
| account\_available\_status | TINYINT UNSIGNED | 否 | 否 | 账户可用状态 0-禁用 1-启用 |
| avatar | VARCHAR(255) | 否 | 是 | 用户头像路径 |
| username | VARCHAR(50) | 否 | 是 | 用户名 |
| sex | TINYINT UNSIGNED | 否 | 否 | 性别 0-未知 1-男 2-女 |
| birth | DATE | 否 | 否 | 出生日期 |
| location | BIGINT | 否 | 否 | 地区 |
| slogan | VARCHAR(255) | 否 | 否 | 格言 |
| description | TEXT | 否 | 否 | 简介 |
| images | TEXT | 否 | 否 | 用户图片 |
| email | VARCHAR(100) | 否 | 否 | 邮箱 |
| wx | VARCHAR(50) | 否 | 否 | 微信 |
| qq | VARCHAR(20) | 否 | 否 | QQ |
| dou\_yin | VARCHAR(50) | 否 | 否 | 抖音 |
| red\_book | VARCHAR(50) | 否 | 否 | 小红书 |
| ctime | BIGINT | 否 | 是 | 创建时间 |
| utime | BIGINT | 否 | 是 | 更新时间 |

表3.2 用户角色表(user\_role)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| user\_id | BIGINT UNSIGNED | 否 | 是 | 用户id |
| role | VARCHAR(50) | 否 | 是 | 角色名 |
| status | TINYINT UNSIGNED | 否 | 否 | 状态 0-禁用 1-启用 |
| ctime | BIGINT | 否 | 是 | 创建时间（秒级时间戳） |
| utime | BIGINT | 否 | 是 | 更新时间（秒级时间戳） |

表3.3 用户认证表(user\_attestation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| user\_id | BIGINT | 否 | 是 | 用户id |
| type | VARCHAR(20) | 否 | 是 | 认证类型 |
| info | VARCHAR(255) | 否 | 是 | 证件信息 |
| status | TINYINT | 否 | 否 | 认证状态 1-可用，0-过期 |
| ctime | BIGINT | 否 | 是 | 创建时间 |
| utime | BIGINT | 否 | 是 | 更新时间 |
| version | INT UNSIGNED | 否 | 否 | 版本号 |

表3.4 动物账户表(animal\_account)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| animal\_id | BIGINT | 否 | 是 | 动物id |
| avatar | VARCHAR(255) | 否 | 是 | 动物头像 |
| name | VARCHAR(50) | 否 | 是 | 动物名称 |
| sex | TINYINT | 否 | 否 | 动物性别 0-未知，1-公，2-母 |
| weight | BIGINT | 否 | 否 | 动物体重g |
| img | TEXT | 否 | 否 | 动物图片 |
| birth | BIGINT | 否 | 否 | 动物生日 |
| location | BIGINT | 否 | 否 | 三级地址码 |
| description | TEXT | 否 | 否 | 描述 |
| host\_id | BIGINT | 否 | 是 | 归属id |
| ctime | BIGINT | 否 | 是 | 创建时间（秒级时间戳） |
| utime | BIGINT | 否 | 是 | 更新时间（秒级时间戳） |

表3.5 领养申请表(adopt\_apply)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| residence\_type | VARCHAR(20) | 否 | 是 | 居住类型 |
| has\_yard | TINYINT | 否 | 否 | 1是0否有院子 |
| family\_members | INT UNSIGNED | 否 | 否 | 家庭成员数量 |
| existing\_pets | VARCHAR(200) | 否 | 否 | 现有宠物情况 |
| work\_status | VARCHAR(20) | 否 | 否 | 工作状态 |
| annual\_income | VARCHAR(20) UNSIGNED | 否 | 否 | 年收入万元 |
| pet\_experience | TEXT | 否 | 否 | 养宠经验描述 |
| adoption\_plan | TEXT | 否 | 是 | 领养后的饲养计划 |
| phone | VARCHAR(20) | 否 | 是 | 联系人电话 |
| status | BIGINT | 否 | 否 | 申请状态 |
| user\_id | BIGINT | 否 | 是 | 用户id |
| animal\_id | BIGINT | 否 | 是 | 动物id |
| ctime | BIGINT | 否 | 是 | 创建时间 |
| utime | BIGINT | 否 | 是 | 更新时间 |

表3.6 领养申请表(apply\_audit\_log)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| apply\_id | BIGINT UNSIGNED | 否 | 是 | 申请编号 |
| audit\_event | VARCHAR(63) | 否 | 是 | 审核事件 |
| audit\_comment | VARCHAR(255) | 否 | 是 | 审核意见 |
| auditor\_id | BIGINT UNSIGNED | 否 | 是 | 审核员id |
| audit\_remark | VARCHAR(500) | 否 | 否 | 审核备注 |
| ctime | BIGINT | 否 | 是 | 创建时间 |

表3.7 投喂记录表(feed\_his)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| name | VARCHAR(50) | 否 | 是 | 食物名称 |
| weight | BIGINT | 否 | 是 | 食物克重 |
| images | TEXT | 否 | 是 | 食物图片 |
| user\_id | BIGINT | 否 | 是 | 用户id |
| animal\_id | BIGINT | 否 | 是 | 动物id |
| location | BIGINT | 否 | 是 | 地点 |
| location\_desc | VARCHAR(255) | 否 | 是 | 详细地址 |
| ctime | BIGINT | 否 | 是 | 创建时间 |

表3.8 帖子信息表(post\_info)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| post\_id | BIGINT | 否 | 是 | 帖子id |
| title | VARCHAR(63) | 否 | 是 | 标题 |
| content | VARCHAR(4096) | 否 | 是 | 内容 |
| images | TEXT | 否 | 否 | 图片 |
| location | BIGINT | 否 | 是 | 三级地址 |
| user\_id | BIGINT | 否 | 是 | 发帖人id |
| animal\_id | BIGINT | 否 | 否 | 动物id |
| event | TINYINT | 否 | 否 | 是否是事件 |
| ctime | BIGINT | 否 | 是 | 创建时间 |
| utime | BIGINT | 否 | 是 | 更新时间 |

表3.9 评论信息表(comment\_info)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| content | VARCHAR(255) | 否 | 否 | 评论内容 |
| post\_id | BIGINT | 否 | 是 | 帖子id |
| user\_id | BIGINT | 否 | 是 | 评论人id |
| warn | TINYINT | 否 | 否 | 敏感评论 |
| type | VARCHAR(20) | 否 | 是 | 评论类型 |
| ctime | BIGINT | 否 | 是 | 创建时间 |

表3.10 评论信息子表(comment\_child\_info)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| content | VARCHAR(255) | 否 | 是 | 评论内容 |
| parent\_id | BIGINT | 否 | 是 | 父集id |
| post\_id | BIGINT | 否 | 是 | 帖子id |
| user\_id | BIGINT | 否 | 是 | 评论人id |
| warn | TINYINT | 否 | 否 | 敏感评论 |
| type | VARCHAR(20) | 否 | 是 | 评论类型 |
| ctime | BIGINT | 否 | 是 | 创建时间 |

表3.11 群聊表(chat\_group)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| name | VARCHAR(63) | 否 | 是 | 群名 |
| avatar | VARCHAR(255) | 否 | 是 | 群头像 |
| ctime | BIGINT | 否 | 否 | 创建时间 |
| utime | BIGINT | 否 | 否 | 更新时间 |

表3.12 群成员表(chat\_member)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| chat\_group\_id | BIGINT | 否 | 是 | 群id |
| user\_id | BIGINT | 否 | 是 | 用户id |
| exist | TINYINT | 否 | 否 | 是否退群 |
| ctime | BIGINT | 否 | 是 | 创建时间 |
| utime | BIGINT | 否 | 是 | 更新时间 |

表3.13 聊天记录表(chat\_his)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| chat\_group\_id | BIGINT | 否 | 是 | 群id |
| user\_id | BIGINT | 否 | 是 | 用户id |
| type | VARCHAR(31) | 否 | 是 | 消息类型 |
| content | TEXT | 否 | 是 | 消息内容 |
| ctime | BIGINT | 否 | 是 | 创建时间 |

表3.14 用户关注表(focus\_info)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| follower\_id | BIGINT | 否 | 是 | 关注者用户id |
| followee\_id | BIGINT | 否 | 是 | 被关注者id |
| followee\_type | VARCHAR(20) | 否 | 是 | 被关注者类型 |
| ctime | BIGINT | 否 | 是 | 创建时间 |

表3.15 事件活动信息表(active\_info)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| post\_id | BIGINT | 否 | 是 | 事件详情id |
| status | INT | 否 | 否 | 事件流程状态 |
| convene\_id | BIGINT | 否 | 是 | 发起者id |
| ctime | BIGINT | 否 | 是 | 创建时间 |
| utime | BIGINT | 否 | 是 | 更新时间 |

表3.16 事件活动参与表(join\_his)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 可空 | 主键 | 说明 |
| id | BIGINT UNSIGNED | 是 | 否 | 主键id |
| version | INT UNSIGNED | 否 | 否 | 版本号 |
| active\_id | BIGINT | 否 | 是 | 事件id |
| user\_id | BIGINT | 否 | 是 | 参与者id |
| status | INT | 否 | 否 | 是否参与 |
| ctime | BIGINT | 否 | 是 | 创建时间 |
| utime | BIGINT | 否 | 是 | 更新时间 |

4 流浪动物救助平台的实现

4.1 安全模块

4.1.1 登录注册

本平台实现登陆注册功能如图4.1所示，交互流程见图4.2。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。

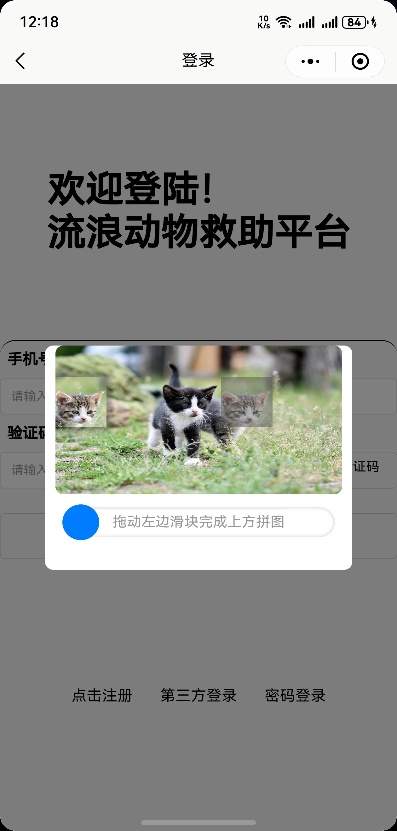


图4.1 流浪动物救助登录注册效果图

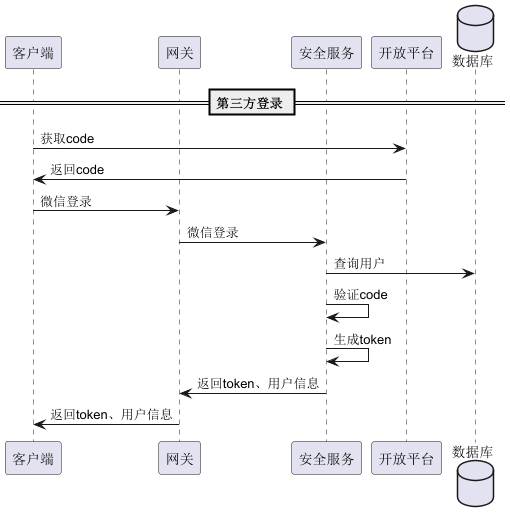
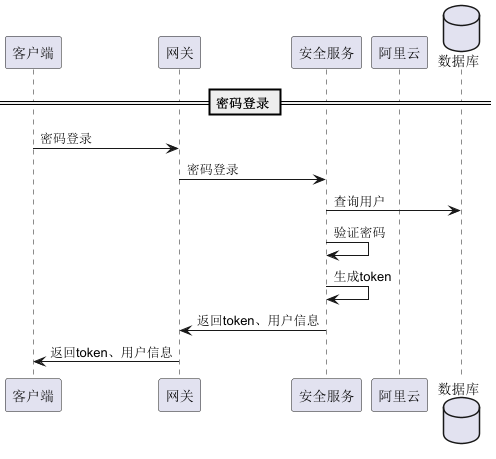
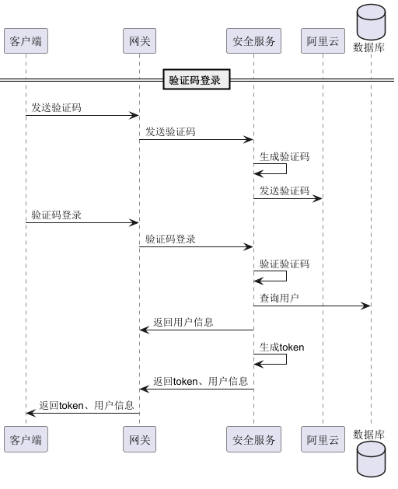


图4.2 流浪动物救助平台登录交互流程

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：

(4.1)

其中为服务器预置密钥，为当前UNIX时间戳（秒级精度），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。

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：

(4.2)

微信平台：直接获取作为唯一标识。

QQ平台：通过OpenID和平台编号PlatformCode生成复合标识。

4.1.2 身份鉴权

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。

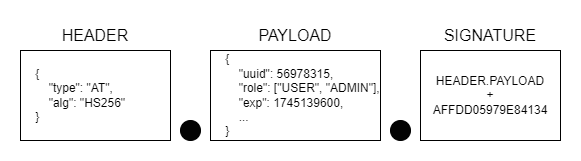


图4.3 JWT组成示意图

JWT由三个部分组成(如图4.3)：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：

(4.3)

(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4.4）保障OAuth流程安全性。

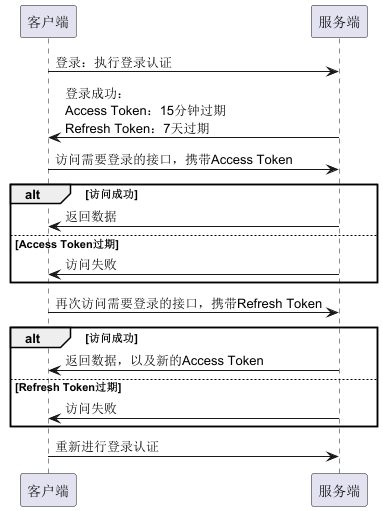


图4.4 AT-RT双token流程图

Access Token(AT)短期有效（15min），用于业务请求

Refresh Token(RT)长期有效（7d），通过进行存储

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。

4.1.3 数据加密

安全模块针对敏感数据（包括用户手机号、地址、聊天内容等）实现了分层加密机制，涵盖数据对外展示与持久化存储两个环节。

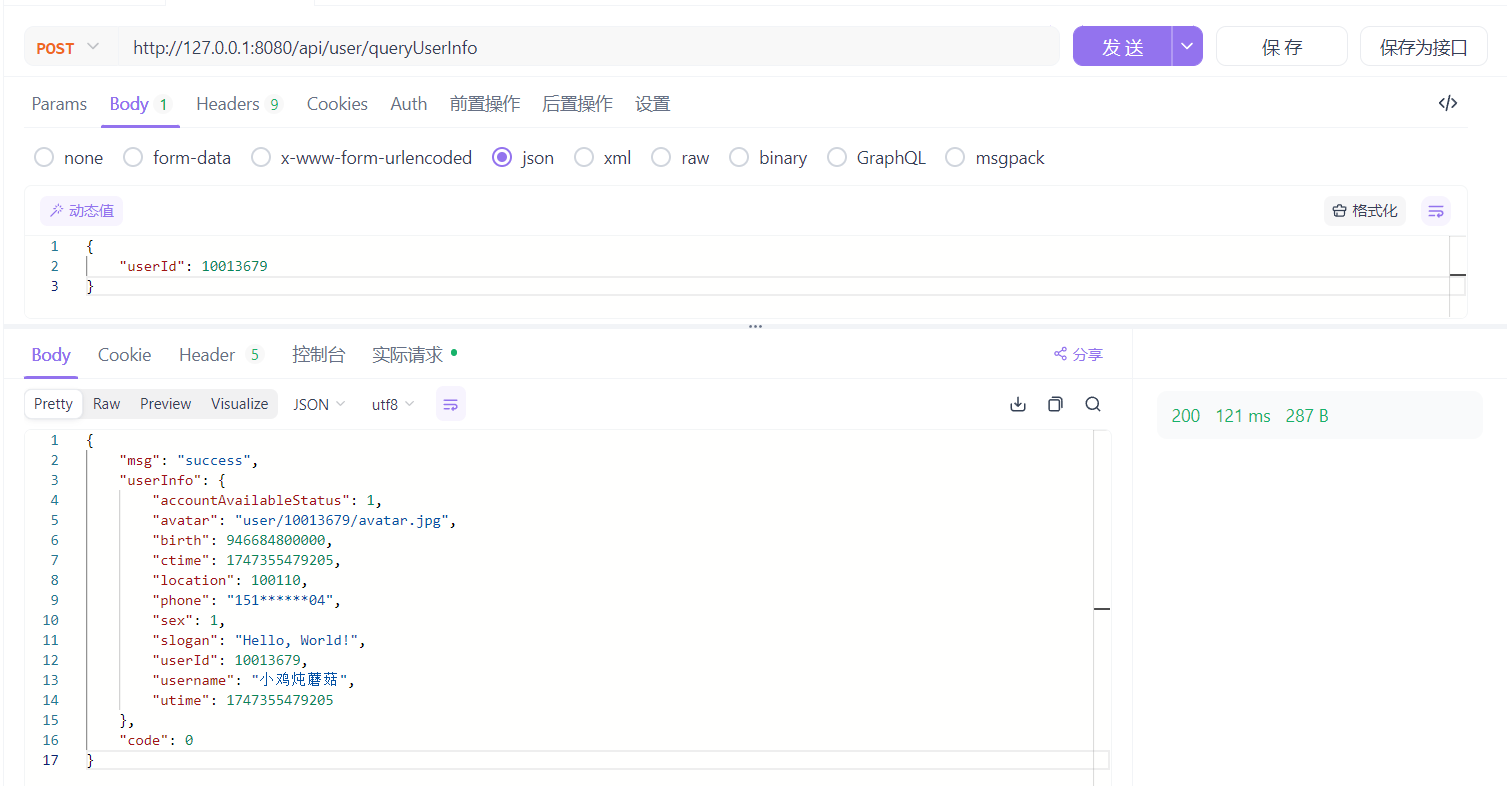


图4.5 敏感信息脱敏效果图

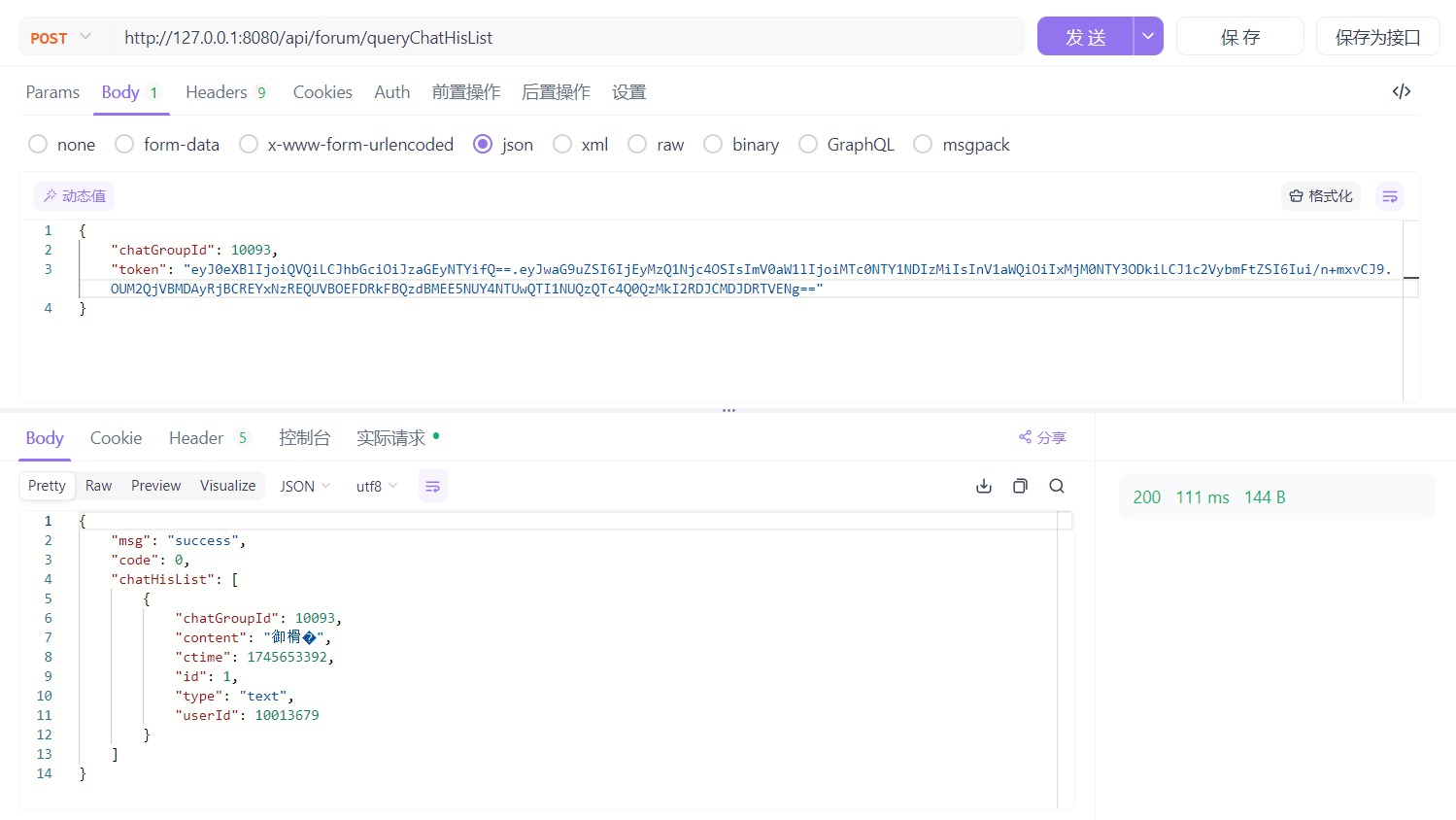


图4.6 敏感文本加密效果图

涉及手机号、真实姓名等个人信息的数据采用数据脱敏技术（如图4.5），隐藏掉关键信息。对于需传输的敏感文本（如聊天内容），系统引入字符偏移编码机制（如图4.6），对数据进行加密处理，前端接收后再进行解密。

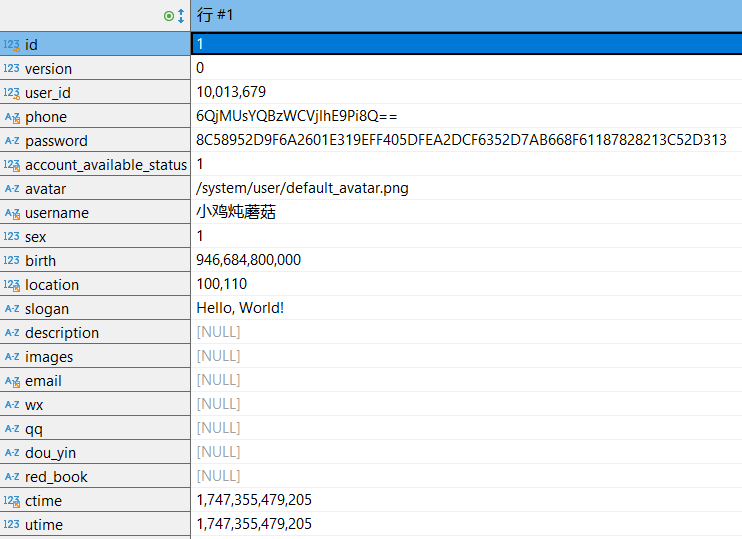


图4.6 持久化数据加密效果图

在持久化存储方面，针对密码等核心敏感数据，采用SHA256哈希算法实现不可逆加密，防止数据被还原。对于手机号等需定期解密使用的信息，采用AES对称加密算法进行加密存储。

4.1.4 账户认证

本平台账户认证包括官方认证与实名认证两种方式。官方认证需要用户提交相关信息及佐证材料，由平台管理员对材料的真实性和完整性进行人工核验。实名认证则通过阿里云进行身份二要素核验。

4.2 用户模块

用户模块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

4.2.1 个人主页

个人主页实现了对于用户基础信息、发布帖子、领养宠物、投喂记录的展示，如图4.7。并且在对于当前用户主页、其他已关注用户主页和其他未关注主页的操作功能上进行了区分。



图4.7 个人主页实现效果图

用户查看自己的主页时，页面会显示用户关注页面和编辑资料页面的入口按钮。查看其他用户主页时，无法查看其他用户的关注信息，并且编辑资料按钮不可见。当用户关注该用户后，其主页关注按钮会变为取消关注按钮，同时解锁发消息和打电话的页面入口按钮。

4.2.1 编辑资料

编辑资料页面如图4.8所示，可以对用户的基础信息和账户信息进行修改，同时为了避免用户误操，保障数据的安全，功能实现采用快照机制对数据进行预修改。当用户修改基础信息时，数据并不会立即进行存储，而是在点击提交后才会发起请求修改用户的数据。当用户修改账户信息时，都要通过校验手机验证码的方式确保用户本人操作。



图4.8 编辑资料实现效果图

4.2.1 关注

关注功能实现了用户对自己关注用户列表、对关注自己用户列表以及对于关注的动物账户列表的展示。在该页面用户同时也可以对目标用户进行取关和再次关注的操作。

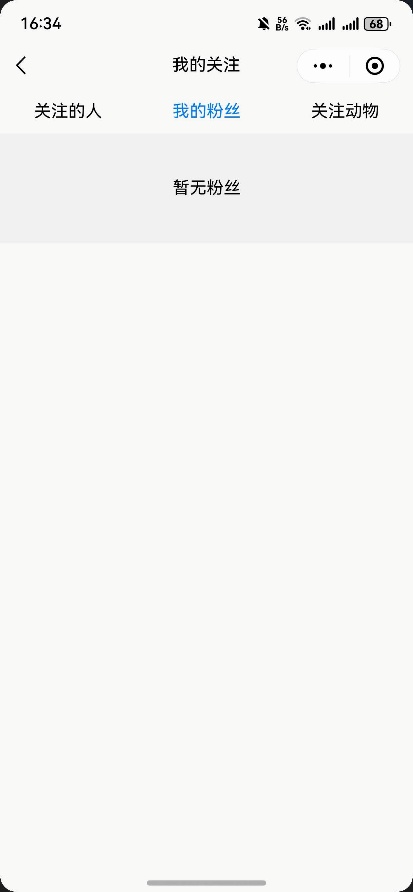


图4.6 关注列表实现效果图

4.2.1 投喂

本平台对投喂功能的实现如图4.9所示，通过数据采集约束、服务端验证和管理员复检机制，对上传的投喂信息进行真实性校验。



图4.9 动物投喂实现效果图

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：

(4.5)

(4.6)

其中为两点纬度，为两点经度，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API强制获取实时影像数据，保证上传的图像都是实时拍摄。

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()

(4.7)

其中R为地球半径(6371km)，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。

3. 管理员复检

管理员通过管理端查看被服务端标记的异常信息，进行人工审核。

4.3.1 领养申请

本平台基于有限状态自动机构建领养流程状态模型（如图4.10所示）。用户可在申请详情页面（如图4.11所示）查看申请审核状态。



图4.10 领养申请实现效果图



图4.11 申请详情实现效果图

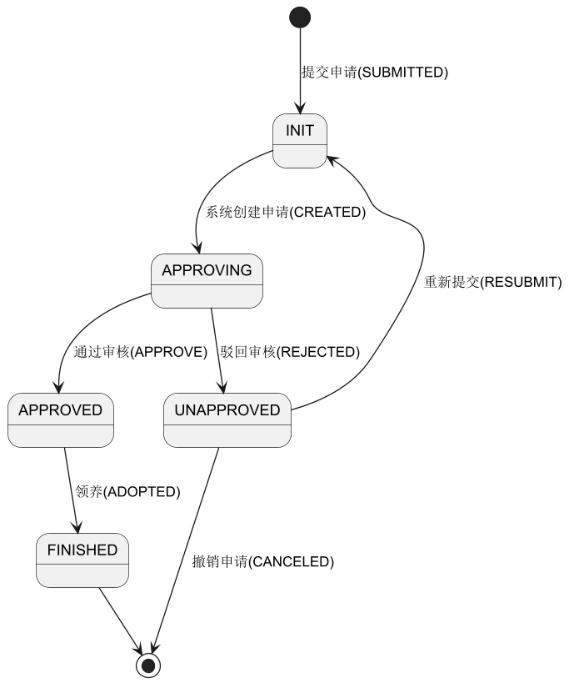


图4.6 领养申请状态图

定义申请初始化（INIT）、审核中（APPROVING）、审核通过（APPROVED）、审核未通过（UNAPPROVED）、领养完成(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。

4.3 社区模块

4.3.1 首页预览

本平台首页提供关键词模糊检索功能，支持用户通过输入部分关键词查询相关内容。系统对热点帖子进行轮播展示，并支持按照关注、推荐、同城、活动等不同维度对帖子进行聚合展示，实现效果如图4.12所示。



图4.12 社区首页实现效果图

4.3.1 帖子详情

帖子详情页面实现了对帖子信息的模块化展示，用户可在该页面浏览帖子内容及相关评论，并可对帖子执行点赞、收藏、分享等操作。此外，页面支持跳转至发帖人个人主页。



图4.13 帖子详情实现效果图

4.3.1 发帖评论

用户可在指定页面（如图4.14）发布新帖，填写相关内容并提交后，帖子将展示于平台相应版块。其他用户可对帖子进行评论（如图4.15），系统对评论内容进行展示和管理，实现用户间的信息交流与互动



图4.14 帖子发布实现效果图



图4.15 帖子详情实现效果图

4.3.1 消息私信



图4.16 消息私信实现效果图

本平台实现了消息中心页面（如图4.16），对系统消息与用户消息进行统一聚合展示，并支持用户间的即时聊天功能。

1. 实时消息处理

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，则有：

(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（为心跳间隔30s）：

(4.9)

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。交互流程如图4.17、4.18所示。

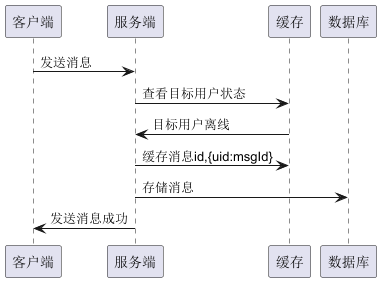


图4.17 离线消息发送交互图

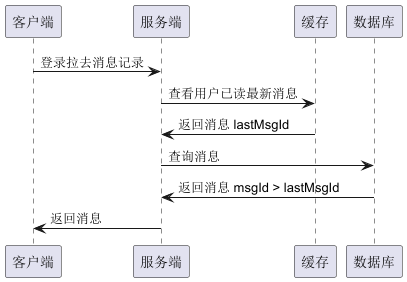


图4.18 离线消息更新交互图

4.5 管理模块

4.5.1 用户管理

用户管理实现了对用户账号状态管理、系统权限增删以及对用户权分配的功能。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）。



图4.19 用户列表实现效果图

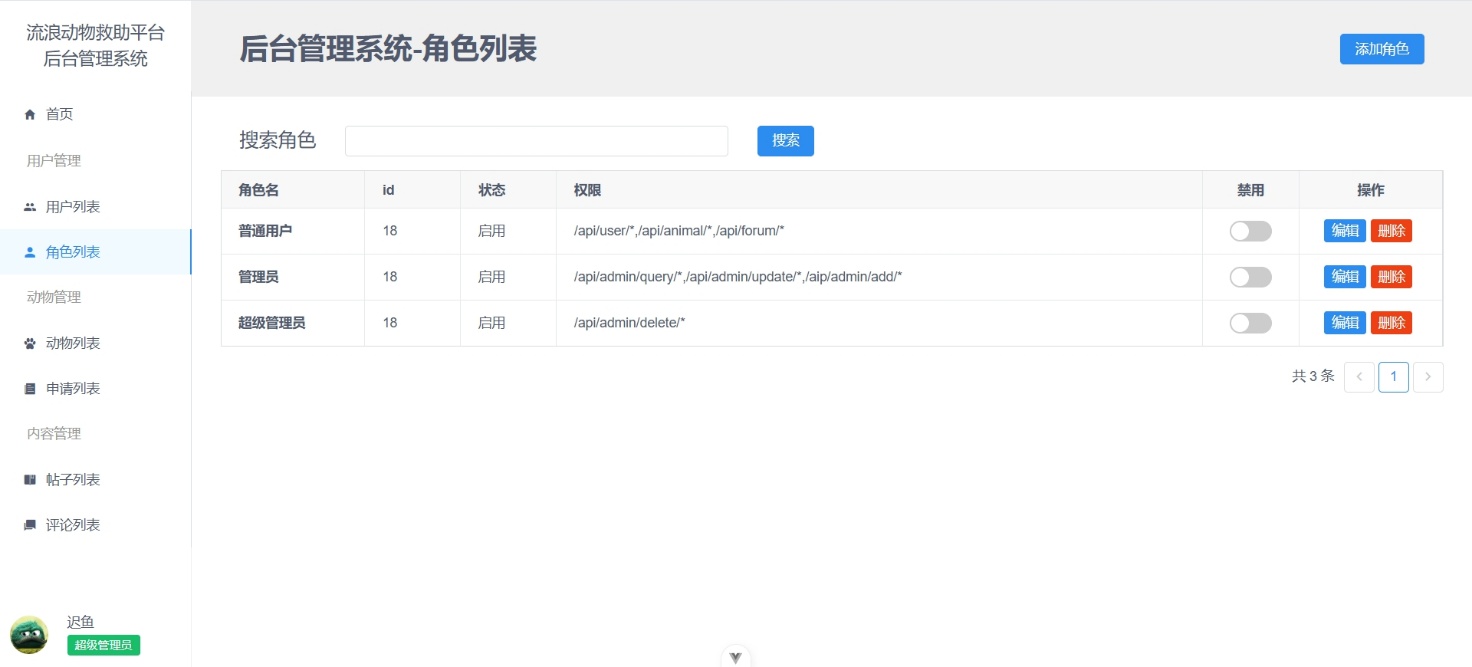


图4.20 角色列表实现效果图

4.5.2 动物管理

动物管理功能包括对系统内动物信息的新增与编辑，以及动物投喂记录的查询。同时，支持对存在异常的投喂记录进行屏蔽处理，使其对普通用户不可见。



图4.21 动物列表实现效果图

4.5.3 内容管理

内容管理功能包括帖子内容管理与评论内容管理，可对包含敏感关键词的内容进行标识与展示。管理员可对标记内容进行审核。并针对重复发布垃圾信息、恶意评论等行为，平台支持梯度处罚机制：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。



图4.22 评论列表实现效果图

4.5.4 申请审核

管理端实现了管理员对用户发起的领养申请审核的功能，实现效果图4.24如所示。



图4.23 申请列表实现效果图



图4.24(a) 申请详情实现效果图



图4.24(b) 申请详情实现效果图

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.