****

|  |  |
| --- | --- |
| **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**

目前我国流浪动物救助主要依靠个人志愿者、民间救助组织和动物收容所，通过线下登记、公告发布、志愿者活动等形式开展救助和领养工作。传统模式下，动物信息登记、领养申请、志愿活动等流程多依赖人工操作，信息传播渠道有限，导致救助效率和领养透明度存在提升空间。为此，本文设计并实现了一套流浪动物救助平台。

平台采用Spring Boot+UniApp实现社区论坛功能，支持用户发帖、评论、点赞和私信，增加信息传播渠道和效率；采用WebSocket+STOMP实现互动交流功能，支持用户实时聊天和消息推送；采用MySql+Redis双数据库模式实现数据的存储，将动物的管理信息化；构建基于事件驱动的有限自动机模型，用户在线提交领养材料，管理员在后台审核，简化领养流程，提高领养效率并保证领养信息的透明度；采用RabbitMQ实现广播机制，对于志愿救助活动在区域内广播召集志愿者，提高流浪动物的救助效率；采用Dubbo实现服务解耦与高效通信，前端采用Uni-App实现多端适配，保证平台的可用性、扩展性和易用性。

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

**ABSTRACT**

At present, the rescue of stray animals in China mainly relies on individual volunteers, non-governmental rescue organizations, and animal shelters, and carries out rescue and adoption work through offline registration, announcement release, volunteer activities, and other forms. Under the traditional mode, animal information registration, adoption application, volunteer activities and other processes rely heavily on manual operations, with limited channels for information dissemination, resulting in room for improvement in rescue efficiency and adoption transparency. Therefore, this article designs and implements a stray animal rescue platform.

The platform adopts Spring Boot+UniApp to implement community forum functions, supporting users to post, comment, like, and send private messages, increasing information dissemination channels and efficiency; Using WebSocket+STOMP to implement interactive communication function, supporting real-time chat and message push for users; Using MySql+Redis dual database mode to store data and digitize animal management; Build an event driven finite automaton model, where users submit adoption materials online and administrators review them in the background, simplifying the adoption process, improving adoption efficiency, and ensuring transparency of adoption information; Using RabbitMQ to implement a broadcasting mechanism, volunteer rescue activities are broadcasted and gathered within the region to improve the efficiency of rescuing stray animals; Dubbo is used to achieve service decoupling and efficient communication, and Uni App is used in the front-end to achieve multi terminal adaptation, ensuring the platform's availability, scalability, and ease of use.

**Keywords:** Rescue of stray animals; Information technology platform; Microservice architecture; Hierarchical structure style; Uni-App; Spring Boot

**目 录**

[**摘 要** I](#_Toc197004577)

[**ABSTRACT** II](#_Toc197004578)

[1 绪论 1](#_Toc197004579)

[1.1 应用背景及意义 1](#_Toc197004580)

[1.2 国内外应用现状 1](#_Toc197004581)

[1.2.1 国外应用现状 1](#_Toc197004582)

[1.2.2 国内应用现状 1](#_Toc197004583)

[1.3 论文结构安排 2](#_Toc197004584)

[2 流浪动物救助平台需求分析 3](#_Toc197004585)

[2.1 流浪动物救助平台功能性需求 3](#_Toc197004586)

[2.1.1 普通用户功能 4](#_Toc197004587)

[2.1.2 志愿者功能 4](#_Toc197004588)

[2.1.3 管理员功能 4](#_Toc197004589)

[2.2 流浪动物救助平台非功能性需求 4](#_Toc197004590)

[2.2.1 性能需求 4](#_Toc197004591)

[2.2.2 安全性需求 4](#_Toc197004592)

[2.2.3 易用性需求 4](#_Toc197004593)

[2.2.4 可靠性需求 4](#_Toc197004594)

[2.3 流浪动物救助平台的可行性 4](#_Toc197004595)

[2.4 小结 5](#_Toc197004596)

[3 流浪动物救助平台的设计 6](#_Toc197004597)

[3.1 流浪动物救助平台功能设计 6](#_Toc197004598)

[3.1.1 社区模块 6](#_Toc197004599)

[3.1.2 用户模块 6](#_Toc197004600)

[3.1.3 管理模块 6](#_Toc197004601)

[3.1.4 安全模块 6](#_Toc197004602)

[3.2 流浪动物救助平台架构设计 7](#_Toc197004603)

[3.2.1 用户层 7](#_Toc197004604)

[3.2.1 网关层 7](#_Toc197004605)

[3.2.2 应用层 7](#_Toc197004606)

[3.2.3 领域层 8](#_Toc197004607)

[3.2.4 基础设施层 8](#_Toc197004608)

[3.3 流浪动物救助平台数据库设计 8](#_Toc197004609)

[3.3.1 数据库概念结构设计 8](#_Toc197004610)

[3.3.2 数据库逻辑结构设计 9](#_Toc197004611)

[3.3.3 数据库物理结构设计 10](#_Toc197004612)

[3.4 小结 16](#_Toc197004613)

[4 流浪动物救助平台的实现 17](#_Toc197004614)

[4.1 安全模块 17](#_Toc197004615)

[4.1.1 登录注册 17](#_Toc197004616)

[4.1.2 身份鉴权 19](#_Toc197004617)

[4.1.3 数据加密 21](#_Toc197004618)

[4.2 用户模块 22](#_Toc197004619)

[4.2.1 个人主页 22](#_Toc197004620)

[4.2.1 编辑资料 23](#_Toc197004621)

[4.2.1 关注 23](#_Toc197004622)

[4.2.1 投喂 24](#_Toc197004623)

[4.3.1 领养申请 25](#_Toc197004624)

[4.3 社区模块 26](#_Toc197004625)

[4.3.1 首页预览 26](#_Toc197004626)

[4.3.1 帖子详情 27](#_Toc197004627)

[4.3.1 发帖评论 27](#_Toc197004628)

[4.3.1 消息私信 28](#_Toc197004629)

[4.5 管理模块 30](#_Toc197004630)

[4.5.1 用户管理 30](#_Toc197004631)

[4.5.2 动物管理 31](#_Toc197004632)

[4.5.3 内容管理 32](#_Toc197004633)

[4.5.4 申请审核 33](#_Toc197004634)

[4.6 小结 34](#_Toc197004635)

[5 流浪动物救助平台的测试 35](#_Toc197004636)

[5.1 功能测试 35](#_Toc197004637)

[5.1.1 用户登录注册测试 35](#_Toc197004638)

[5.1.2 宠物管理与投喂功能测试 35](#_Toc197004639)

[5.1.3 社区互动与论坛交流功能测试 35](#_Toc197004640)

[5.1.4 领养申请与审核流程测试 35](#_Toc197004641)

[5.1.5 后台管理功能测试 35](#_Toc197004642)

[5.2 系统测试 36](#_Toc197004643)

[5.2.1 响应时间测试 36](#_Toc197004644)

[5.2.2 吞吐量测试 36](#_Toc197004645)

[5.2.3 易用性测试 37](#_Toc197004646)

[5.2.4 安全性测试 37](#_Toc197004647)

[5.3 小结 37](#_Toc197004648)

[6 总结与期望 38](#_Toc197004649)

[6.1 项目总结 38](#_Toc197004650)

[6.2 未来展望 38](#_Toc197004651)

[参考文献 39](#_Toc197004652)

1 绪论

1.1 应用背景及意义

目前我国流浪动物传统的救助模式主要依靠线下登记、人工管理和分散的志愿者活动，存在信息不对称、管理效率低、领养流程不透明等问题，对此作者设计并实现了一套流浪动物救助平台。平台采用层次架构设计[1]，前端基于Vue.js和Uni-App开发，实现跨终端适配，提升用户操作体验。后端采用Spring Boot[2,3]框架，负责业务逻辑处理，保障系统稳定性与高效性。数据层选用MySQL[4]数据库，负责动物、用户、领养、社区等核心数据的存储与管理，确保数据安全和一致性。

系统架构遵循模块化设计原则，各功能模块相互解耦，支持独立部署和灵活扩展。平台实现了动物信息登记、健康档案维护、投喂记录上传、领养申请与审核、社区互动、志愿活动管理、后台数据统计等核心业务功能。针对高并发访问场景，平台通过缓存优化与消息队列技术提升响应速度，保障数据一致性。安全层面，平台集成用户身份认证、权限分级、数据加密等机制，防止未授权访问和数据泄露，提升整体安全性。

1.2 国内外应用现状

1.2.1 国外应用现状

国外在流浪动物救助及相关领域应用研究成果相对先进。Tina Rich等[5]所著书籍整合动物福利知识，助力教育及跨领域研究。Shang Xianru等人[6]将服务设计理论用于校园流浪动物管理，构建混合系统改善校园生态与动物福利。Workman Miranda K等[7]评估Petfinder网站对猫咪领养的影响，发现点击量、年龄、毛色等与领养相关。Alexandra Protopopova等[8]研究收容所领养者与狗的互动，分析影响领养的行为和因素。Sternberg等[9]所著书籍评估收容所管理因素与关键指标的关系，为提升管理效果提供数据支持和改进建议。这些研究从多方面推动了动物福利和管理的发展。

1.2.2 国内应用现状

国内对流浪动物问题的研究也在持续深入。在救助站研究上，雷思雨等[10]指出，公益性救助站因流浪动物数量激增，面临资金、物资短缺与运营管理困难等问题，如兰州流浪动物救助站运营压力巨大。互联网技术应用方面，姜苏[11]提出搭建线上线下结合的救助系统意义重大，其利用互联网公益优势，能调动公众参与。但当前智能流浪猫屋 “云投喂” 等模式存在公益与商业边界不清等问题。救助实践优化上，邢月等[12]提出鼓励大学生参与、引入第三方协调等策略；张秋雨[13]以宜宾为例，强调针对性帮扶与 TNR 项目的作用；纪好[14]从服务设计出发，构建了救助服务系统蓝图。总体而言，国内虽有成果，但在救助站运营、互联网模式规范等方面仍需探索。

1.3 论文结构安排

本文按照软件工程开发流程，介绍“流浪动物救助平台”的设计与实现。首先，通过问卷调查、用户访谈和实地调研收集用户需求，并进行分析。随后基于分析结果，设计平台的系统架构和功能模块。最后，开发并实现各项系统功能。论文结构如下：

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，明确论文研究目标。

第二章为需求分析，通过问卷调查、用户访谈等方式获取用户需求，并分析系统可行性。

第三章为系统设计，基于需求分析，提出平台整体架构，细化功能模块，设计数据库结构，说明模块间交互。

第四章为系统实现，描述平台核心模块的实现方法，包括安全模块、用户模块、社区模块、管理模块。

第五章为系统测试，采用功能测试和系统测试，验证平台功能、性能、安全性和易用性。

第六章为总结与展望，总结平台建设成果，提出后续优化方向。

2 流浪动物救助平台需求分析

为提升救助工作的规范化和信息化水平，平台面向普通用户、志愿者和管理员三类角色，设计实现了涵盖动物信息管理、投喂记录、领养申请、社区互动和后台管理等功能的流浪动物救助平台。以下将结合实际调研，分析各类用户的功能需求和系统的非功能性需求，并论证平台建设的可行性。

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。

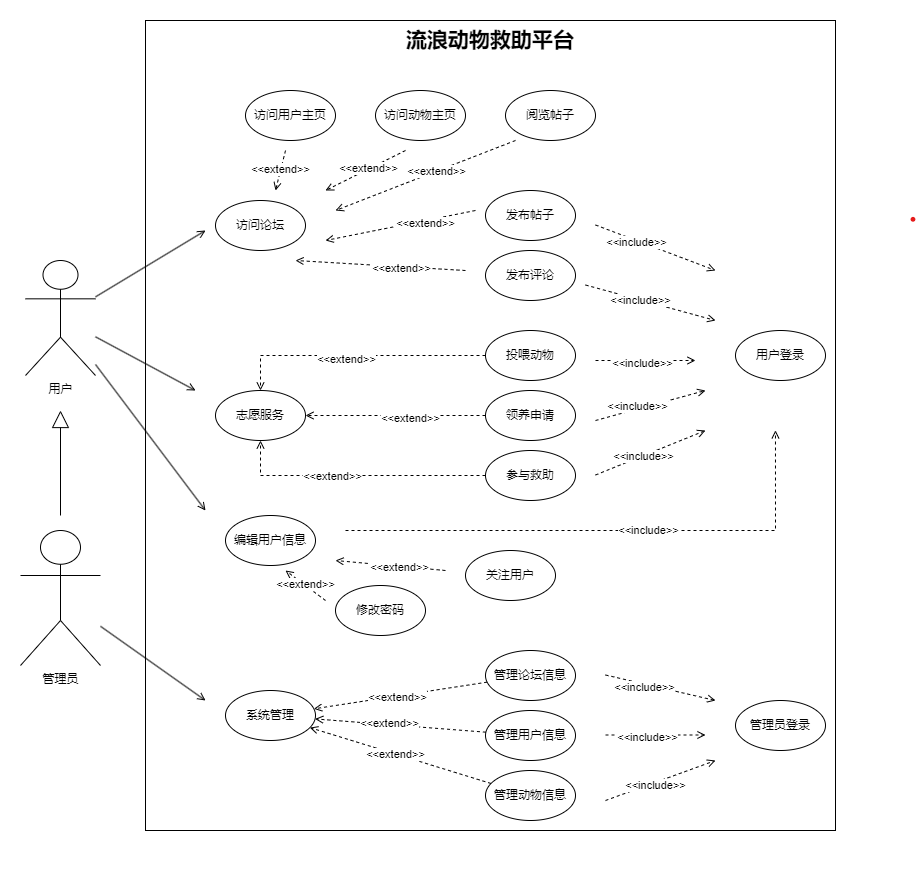


图2.1 用例图

2.1.1 普通用户功能

访问论坛：用户或游客可以浏览流浪动物基本信息、论坛帖子信息、用户信息等内容。

内容发布：用户登录之后可以发布帖子、评论和参与话题讨论。

投喂上传：用户登录之后可上传对动物投喂记录，同时查看动物当天的投喂记录。

领养申请，用户账号实名认证后可以上传证明材料申请领养动物，实时查看审核进度。

信息管理：用户登录之后可以对个人资料、账号绑定等信息进行管理。

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

信息管理：管理员可以对动物档案、投喂记录、用户信息进行增删改查操作。

领养申请审核：管理员可以对领养申请进行审核。

社区内容审核：对社区发布的内容进行审核、屏蔽或删除。

权限与安全管理：分级管理用户权限，有权冻结用户账户。

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。

2.4 小结

通过调研获取普通用户、志愿者和管理员的需求并进行分析，功能需求涵盖信息浏览、发布、管理等。同时提出性能、安全、易用性和可靠性等非功能性需求，并从经济、技术、社会和操作方面论证平台建设可行，为后续设计与实现奠定基础。

3 流浪动物救助平台的设计

根据需求分析结果对流浪动物救助平台进行详细设计，包含功能设计、系统架构设计以及数据库设计。

3.1 流浪动物救助平台功能设计

结合实际业务需求，对流浪动物救助平台功能模块进行如图3.1的划分。

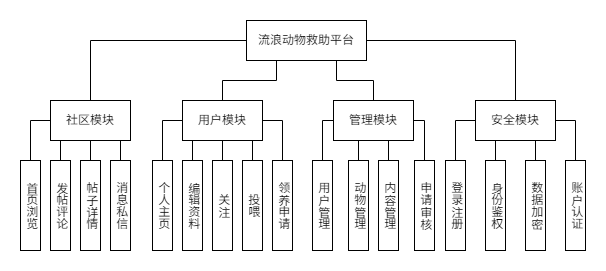


图3.1 流浪动物救助平台功能图

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

为具备良好的性能、安全性、可用性以及易用性，平台采用微服务架构与层次结构风格设计[15,16,17]。整体架构如图3.2所示，分为用户层、网关层、应用层、领域层和基础设施层。

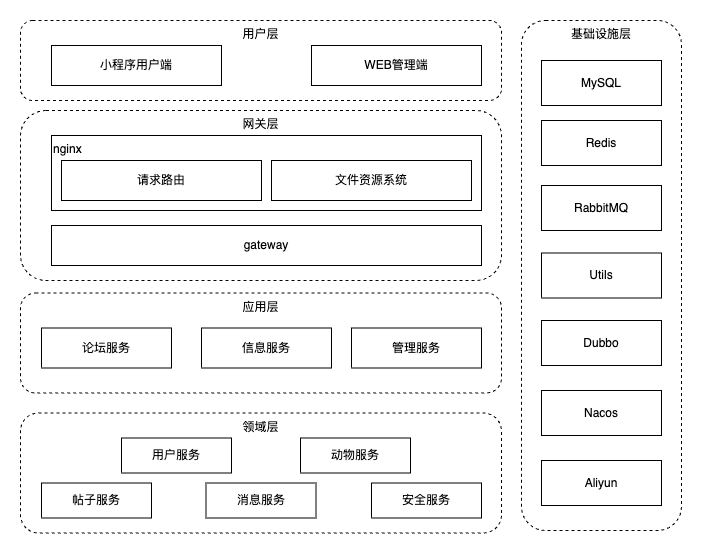


图3.2 系统架构图

3.2.1 用户层

用户层[18]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端和移动端的多终端适配。

3.2.1 网关层

网关层通过Nginx实现负载均衡，对用户请求进行分发和路由。搭建文件服务器并配置CDN以提升响应速度。gateway服务作为后端服务的唯一入口，提供和外部服务的交互。

3.2.2 应用层

应用层负责将各业务模块的逻辑流程进行组织和调度，具体实现系统的业务操作。各服务模块采用独立部署方式，通过Dubbo[19]实现服务间的远程通信。应用层主要包括以下服务：

论坛服务：处理社区板块相关的帖子发布、评论、私信等操作请求。

信息服务：负责用户信息、动物信息等数据的查询、更新、同步等业务流程。

管理服务：实现对用户、动物、内容等数据的后台管理操作。

3.2.3 领域层

领域层主要承担平台核心业务逻辑的实现与抽象，将具体业务规则封装为领域服务。领域层包含以下核心服务：

用户服务：实现用户的注册、登录、实名认证和权限分配等操作。

动物服务：完成流浪动物信息的登记、健康档案维护、救助记录更新等业务处理。

领养服务：处理领养申请的提交、审核、状态跟踪及通知推送等流程。

社区服务：支持用户发帖、评论、活动组织和消息推送等社区相关功能。

安全服务：负责用户身份校验、数据加密、权限校验等安全相关业务。

3.2.4 基础设施层

基础设施层基于Spring Boot[20]框架整合中间件，包括数据库、缓存、消息队列等。平台采用MySQL实现业务数据的持久化管理；Redis[21]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。

3.3 流浪动物救助平台数据库设计

3.3.1 数据库概念结构设计

根据功能模块及架构设计，对数据实体及其相关属性进行了如下设计。实体关系图如图3.3所示。

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。

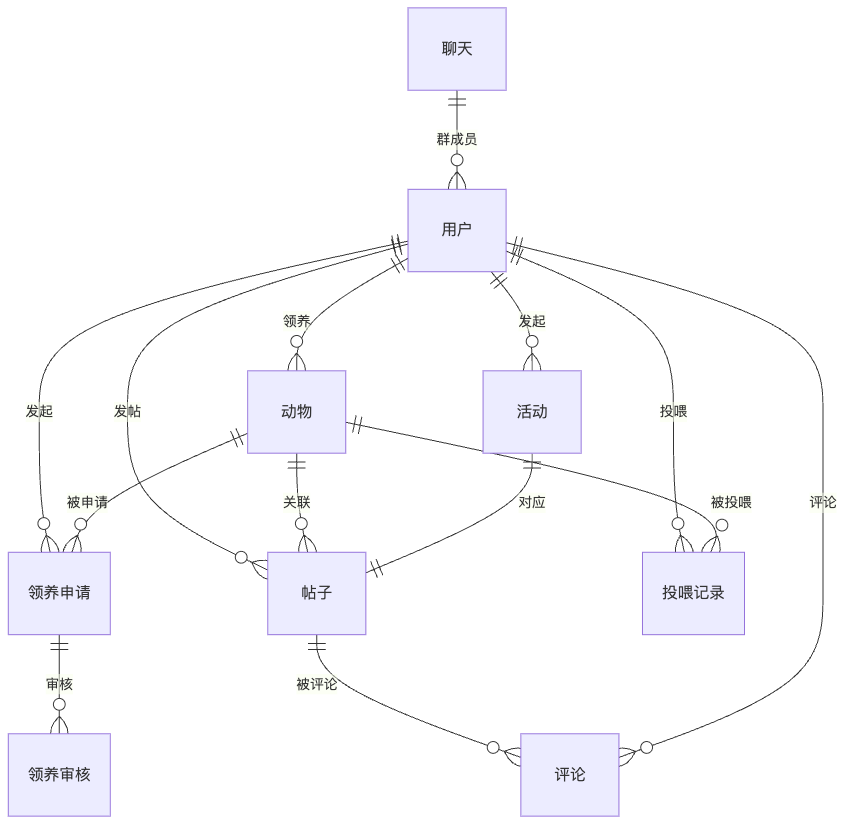


图3.3 数据库ER图

3.3.2 数据库逻辑结构设计

将概念模型转换为逻辑模型过程中，逻辑结构设计主要包含以下核心要素：

1. 数据模型转换：将概念设计阶段的实体对象映射为MySQL数据库表结构，例如用户账户表与动物账户表等具体实现

2. 完整性约束定义：为每张数据表设置主键id确保实体完整性；建立外键关联（如评论表post\_id关联帖子表id）保障参照完整性；通过唯一索引、非空约束等实现业务规则的自定义完整性

3. 关系模型构建：处理一对一和一对多关系时采用外键引用方式（如动物表host\_id字段）；多对多关系则通过关联表（如用户关注表）实现

4. 范式化处理：通过消除数据冗余和传递依赖，使数据库结构满足第三范式要求

5. 查询性能优化：基于业务场景为数据表创建合适索引，并引入Redis缓存机制降低数据库负载，提升系统整体可用性

3.3.3 数据库物理结构设计

依据E-R图和逻辑结构设计，结合 MySQL 数据库 InnoDb 引擎特性，构建平台数据库表结构。表结构除包含实体属性，增设 ctime、utime 属性记录数据创建与更新时间，用 version 属性实现乐观锁，防止并发冲突。同时，依据功能及业务场景为数据表设置索引，提升查询效率，优化系统性能。

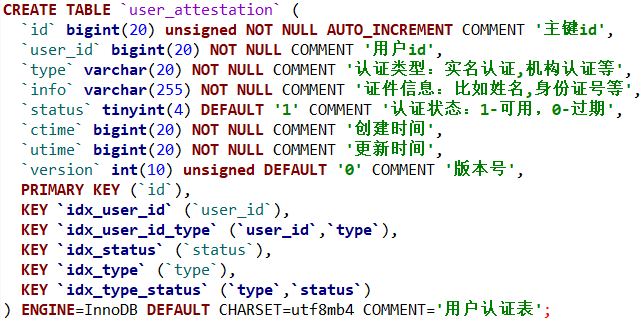


图3.4 用户认证表数据定义

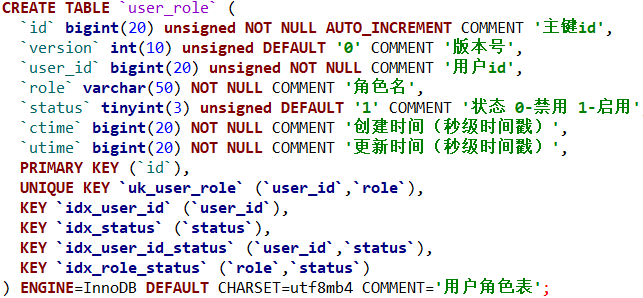


图3.5 用户角色表数据定义



图3.6 用户账户表数据定义

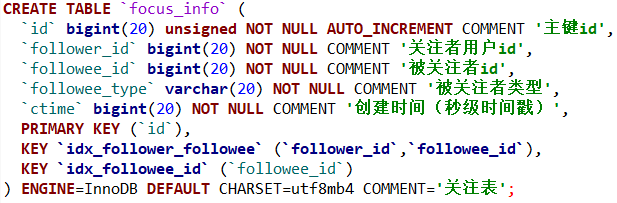


图3.7 用户关注表数据定义



图3.8 动物账户表数据定义



图3.9 领养申请表数据定义

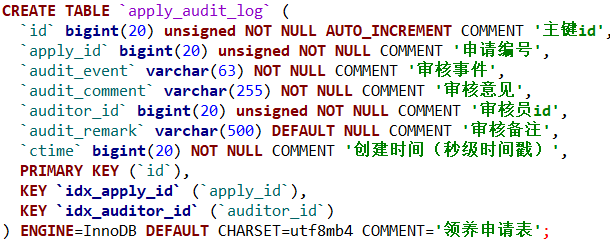


图3.10 申请审核记录表数据定义

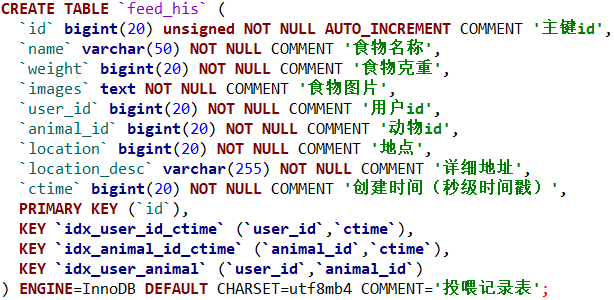


图3.11 投喂记录表数据定义

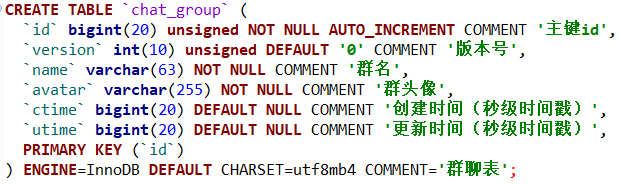


图3.12 群聊群组表数据定义



图3.13 帖子详情表数据定义

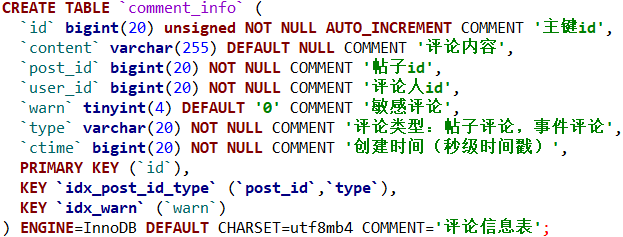


图3.14 评论表数据定义

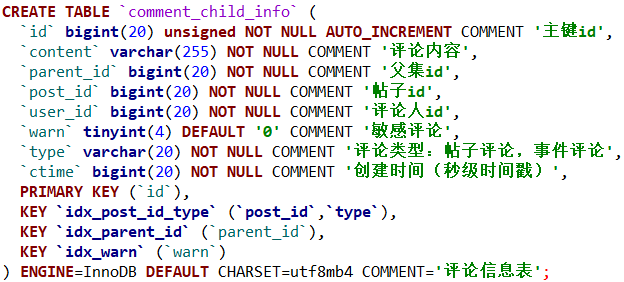


图3.15 二级评论表数据定义

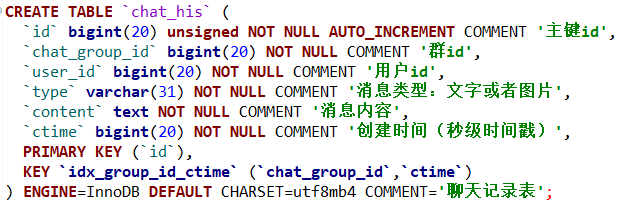


图3.16 聊天记录表数据定义

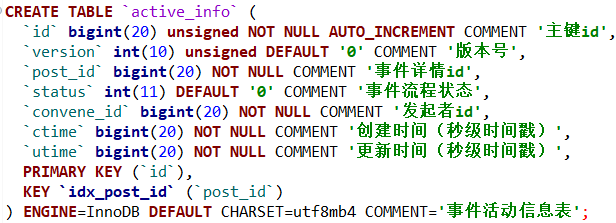


图3.17 活动信息表数据定义

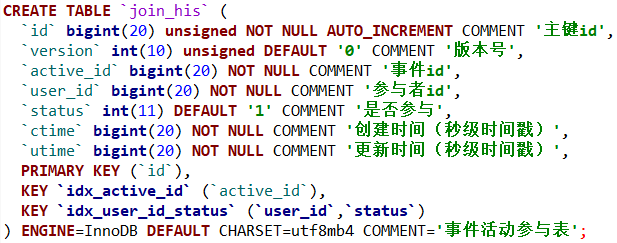


图3.18 用户活动参与表数据定义

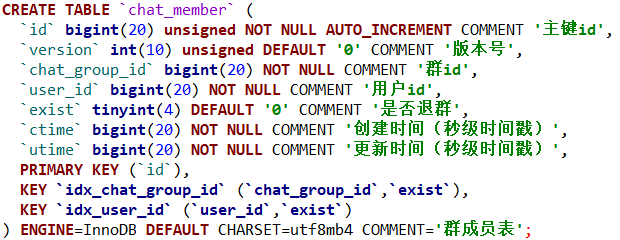


图3.19 群聊成员表数据定义

3.4 小结

第三章围绕流浪动物救助平台展开详细设计。功能设计上，划分了社区、用户、管理、安全模块，明确各模块具体功能，如社区模块支持用户互动，管理模块服务管理员工作。架构设计采用微服务架构与层次结构风格，涵盖用户层、网关层等五层，各层分工协作，保障平台性能、安全等特性。数据库设计包括概念、逻辑和物理结构设计，构建数据实体关系，完成模型转换，设置数据库表结构及索引，确保数据存储与管理的高效性和稳定性，为平台后续实现与运行奠定基础。

4 流浪动物救助平台的实现

根据平台设计思路，将围绕平台的核心功能详细介绍其实现思路、关键技术及实现过程，为后续的系统测试和优化提供基础。

4.1 安全模块

4.1.1 登录注册

登陆注册功能的实现效果和交互流程如图4.1和图4.2所示，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

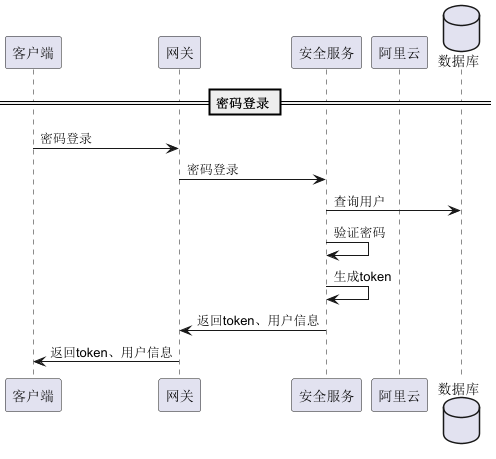


图4.2(a) 传统密码登录交互流程

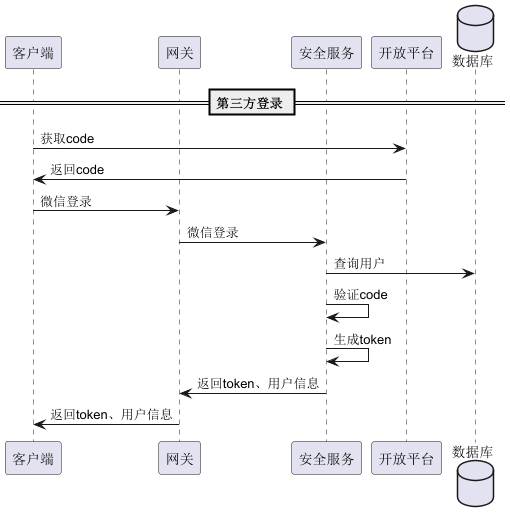


图4.2(b) 第三方Oauth登录交互流程

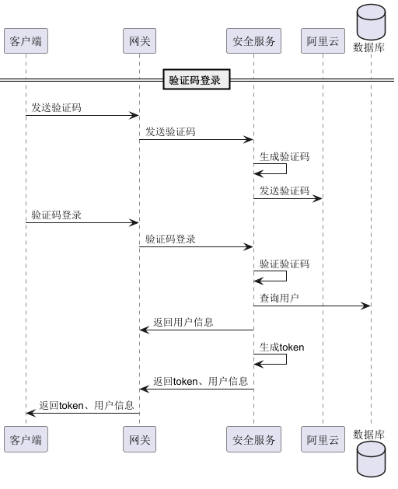


图4.2(c)短信动态验证码登录交互流程

1. 手机验证码生成

为确保验证码的不可预测性和时效性，系统运用基于时间的一次性密码算法(TOTP)动态生成6位数字验证码，其核心运算逻辑可表示为：

(4.1)

其中为服务器预置密钥，为unix时间戳，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。

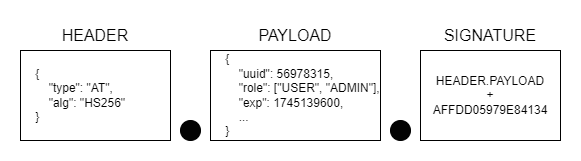


图4.3 JWT组成示意图

JWT由三个部分组成(如图4.3)：头部(HEADER)，载荷(PAYLOAD)，签证(SIGNATURE)。系统采用JWT规范实现身份认证机制，其令牌结构包含三个核心部分：头部定义令牌类型和采用的加密方式；载荷记录用户ID、权限角色及有效期等关键信息；签名则通过头部、载荷与密钥按算法实现。

(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。

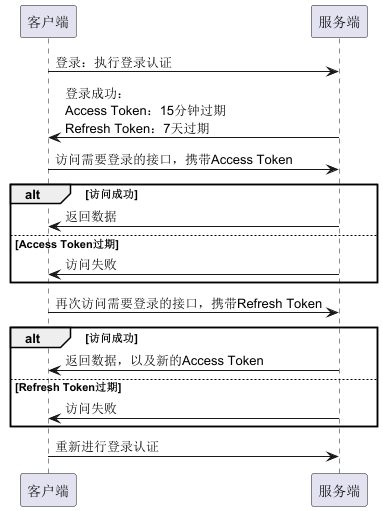


图4.4 AT-RT双token流程图

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。

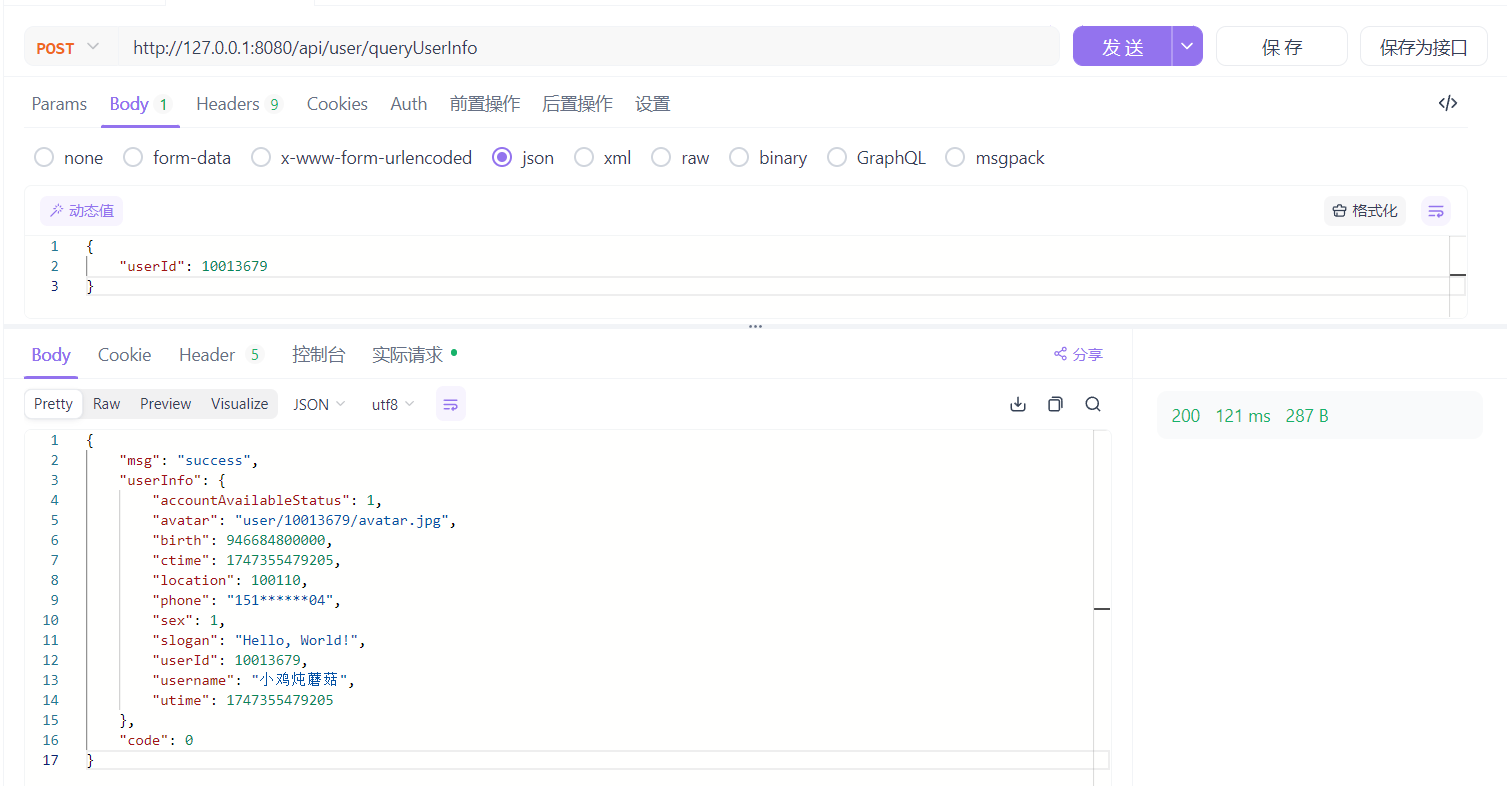
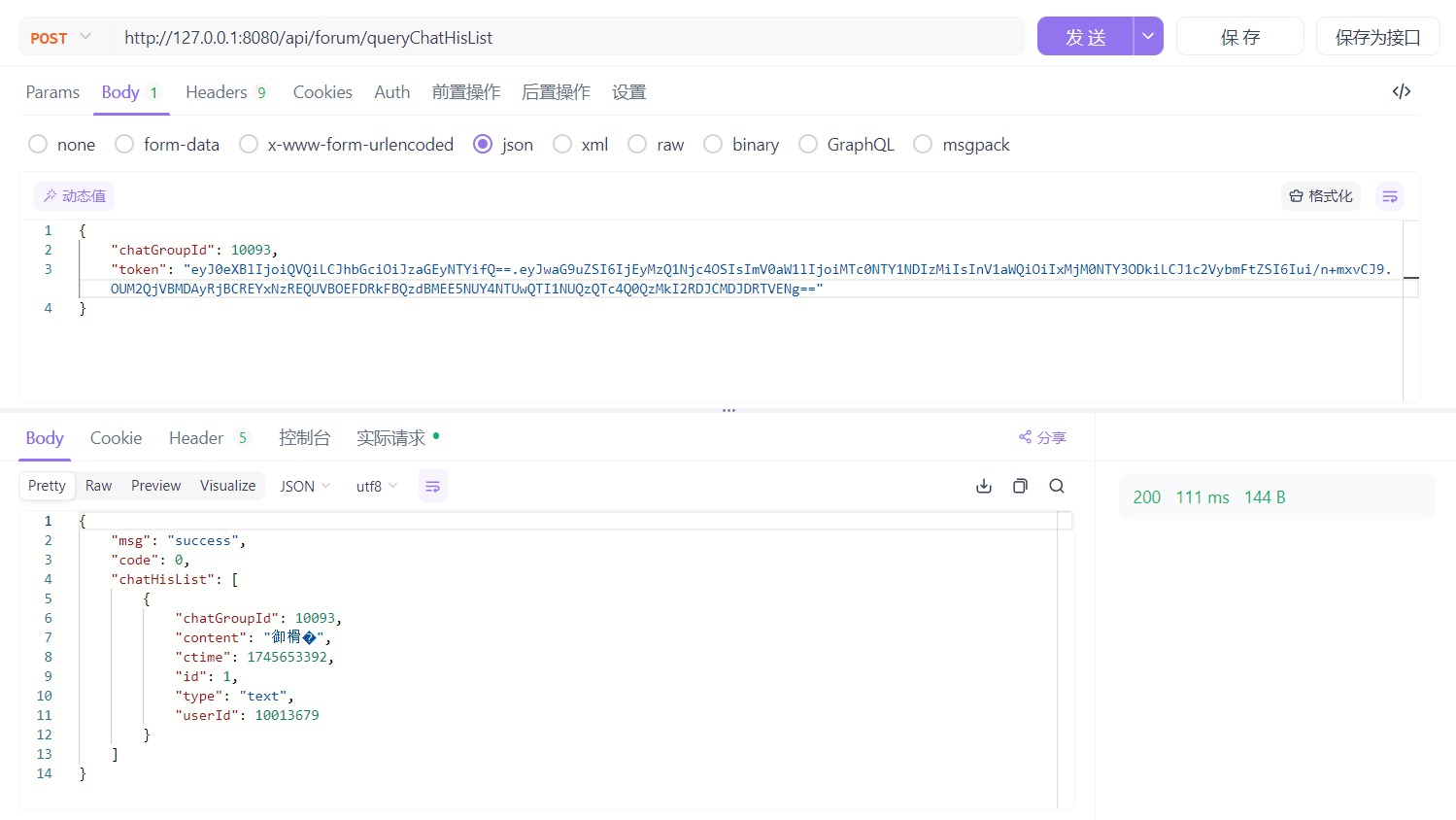
 

图4.5 数据脱敏效果图 图4.6 敏感文本加密效果图

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。

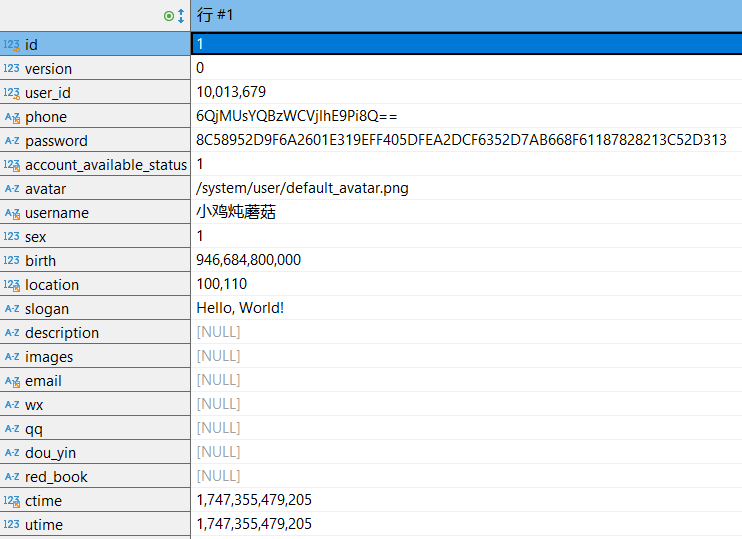


图4.7 持久化数据加密效果图

在持久化存储方面，对密码等关键信息使用SHA256哈希算法进行单向加密处理，确保无法通过逆向工程还原原始数据；而对于需要频繁调用的手机号等信息，则运用AES对称加密技术实现可逆加密存储。

4.1.4 账户认证

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。通过阿里云的身份二要素核验实现实名认证功能。

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

个人主页实现效果如图4.8所示。系统根据访问权限动态展示功能入口：当前用户访问自身主页时可查看关注列表和编辑资料入口；访问他人主页时则隐藏关注信息和资料编辑选项。当用户建立关注关系后，界面交互元素会实时更新，关注按钮切换为取消关注状态，并开放私信和电话联系功能入口。



图4.8 个人主页实现效果图

4.2.1 编辑资料

编辑资料页面实现效果如图4.9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，功能实现了对数据进行预修改的快照机制。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4.9 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。

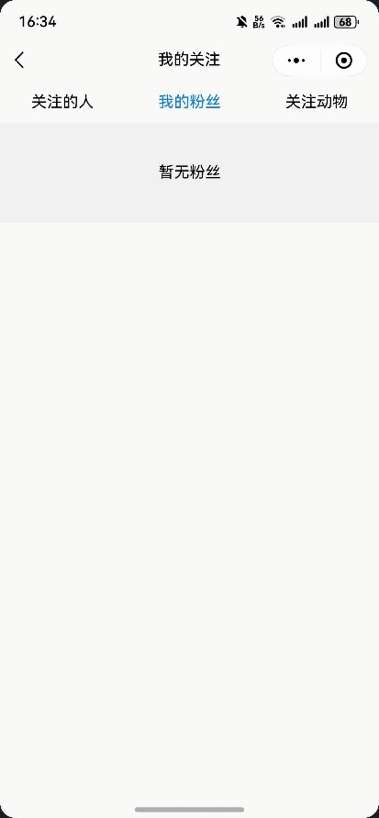


图4.10 关注列表实现效果图

4.2.1 投喂

投喂功能的实现效果如图4.11所示，通过数据采集约束、服务端验证，确保投喂信息的真实性。



图4.11 动物投喂实现效果图

1. 数据采集约束

通过微信小程序接口获取坐标系定位，应用地理围栏技术限制操作范围，定义有效区域以宠物坐标为中心、半径100米的球冠区域，表达式如4.5、4.6所示：

(4.5)

(4.6)

其中为两点纬度，为两点经度，为100米弧长。

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

服务端对于投喂记录的请求进行二次验证。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()。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。

(4.7)

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

4.3.1 领养申请

基于事件驱动构建领养流程的有限状态自动机模型，如图4.12所示。

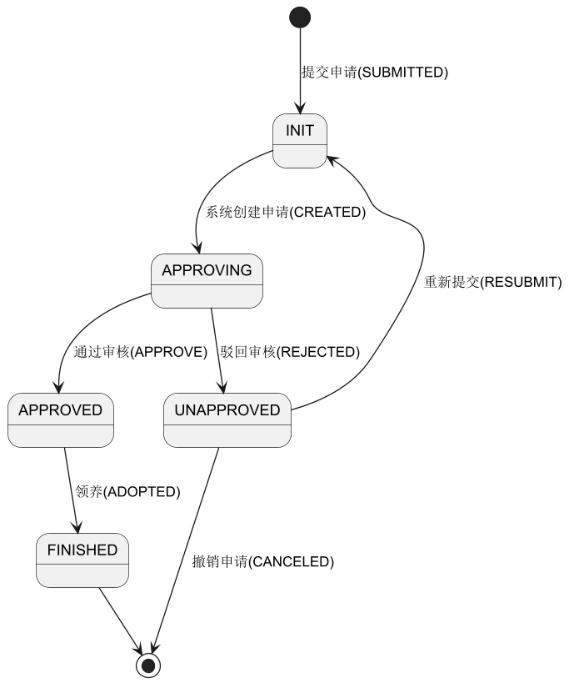


图4.12 领养申请状态图

用户发起领养申请后，状态由申请初始化经由提交申请事件转为审核中；若通过审核，状态转为审核通过，若申请被驳回，则转为审核未通过；审核被驳回后，用户重新提交进行再次审核；在申请到达终态之前，用户都可以撤销申请终止流程使其到达终态。审核通过后，用完成领养，状态转为领养完成到达终态。

图4.13 领养申请实现效果图 图4.14 申请详情实现效果图

4.3 社区模块

4.3.1 首页预览

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4.15所示。

图4.15 首页实现效果图 图4.16 帖子详情效果图

4.3.1 帖子详情

帖子详情界面采用分区块布局方式呈现内容，支持用户查看主帖信息及关联互动数据。该页面提供点赞、收藏和分享等交互功能按钮，同时允许用户通过点击发帖人头像快速访问其个人主页。评论区域采用瀑布流形式展示用户讨论内容。

4.3.1 发帖评论

用户可在指定页面（如图4.17）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4.18），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4.17 帖子发布效果图 图4.18 帖子详情实现效果图

4.3.1 消息私信

消息中心页面效果如图4.19所示，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

当用户处于在线状态时，系统基于WebSocket协议建立持久化连接通道，实现消息的即时推送功能，并对与在线用户通过集合列表进行维护。设用户集合为U，当前在线用户为，则有：

(4.8)



图4.19 消息私信效果图

系统通过客户端定时发送心跳信号（间隔30秒）来维持连接活性，服务端接收到PING指令后立即刷新用户在线状态标识。该状态管理机制基于Redis集群实现，其状态更新公式可表示为（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4.20、4.21所示。

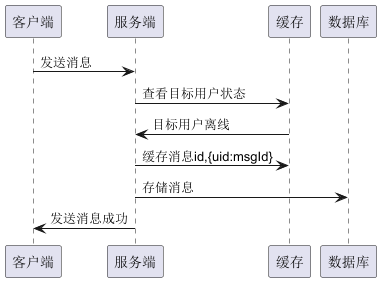


图4.20 离线消息发送交互图

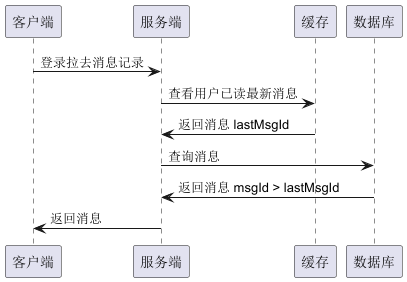


图4.21 离线消息更新交互图

4.5 管理模块

4.5.1 用户管理

用户管理模块基于角色的权限控制模型，设有游客、用户、管理员、超管四级角色。管理员可审核用户信息、管理动物档案、管控论坛内容。超级管理员可配置角色权限策略，并进行接口级权限分配。



图4.22 用户列表效果图

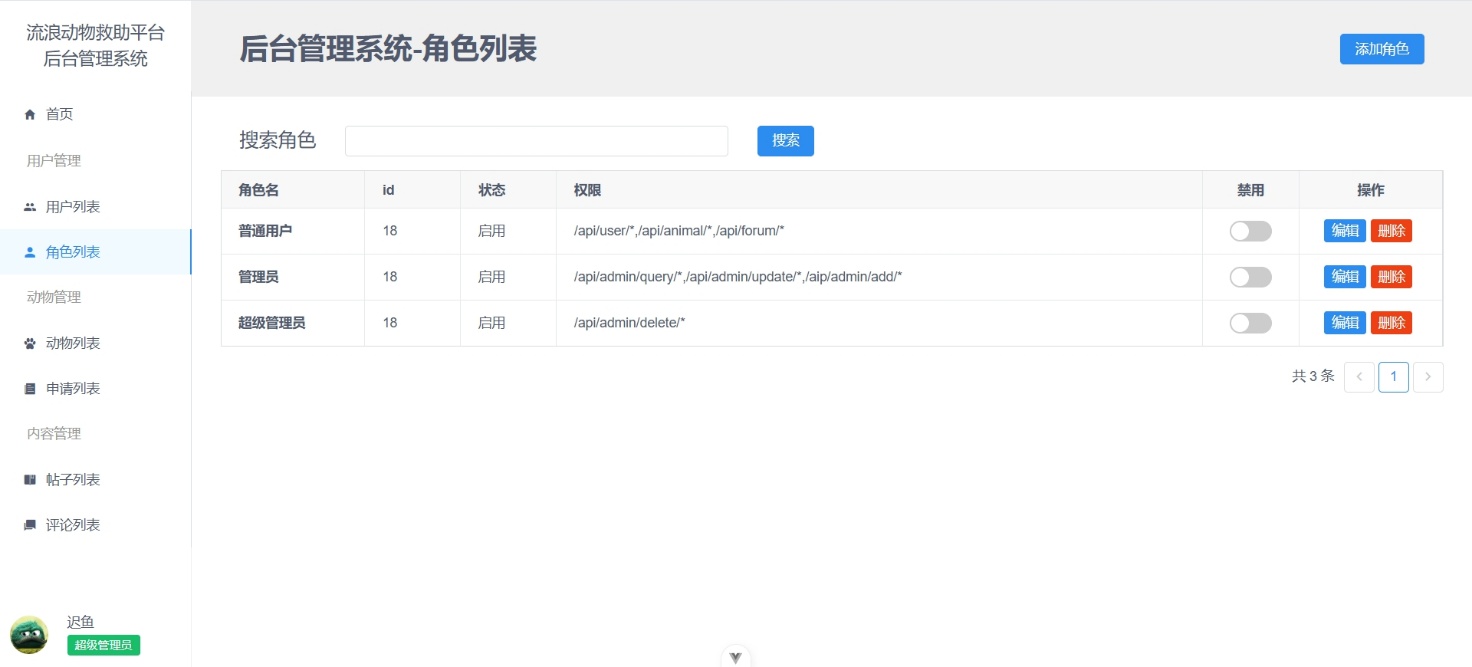


图4.23 角色列表效果图

4.5.2 动物管理

系统实现对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4.24 动物列表效果图

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，对包含敏感关键词的内容进行标识与展示。管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4.25 评论列表效果图

4.5.4 申请审核

审核功能实现效果如图4.27所示。管理员通过管理端查看用户申请详细信息，对申请信息和佐证材料进行详实校验后，填写审核意见并执行相关操作。



图4.26 申请列表实现效果图



图4.27(a) 申请详情实现效果图



图4.27(b) 申请详情实现效果图

4.6 小结

第四章详细介绍了流浪动物救助平台核心功能模块的实现。安全模块通过多种认证、鉴权和加密机制保障系统安全；用户模块实现个人主页展示、资料编辑等功能，并确保操作安全；社区模块支持多种互动操作和消息处理；管理模块助力管理员进行用户、动物和内容管理及申请审核。这些模块的实现为平台后续测试和优化奠定了坚实基础，支撑平台各项功能的正常运行。

5 流浪动物救助平台的测试

系统实现后，为确保功能正常，不包含潜在缺陷，将从功能测试和系统测试两方面对系统进行测试。

5.1 功能测试

5.1.1 用户登录注册测试

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 宠物管理与投喂功能测试

动物管理测试中，涵盖数据增删改查全流程操作，检验管理员与志愿者协同维护动物档案的效率和准确性。对于投喂功能，重点评估了基于地理位置的围栏限制、时间间隔控制、实时拍摄验证及数据校验等安全机制的实际效果，通过模拟高并发投喂场景，验证系统在多用户同时操作时的数据处理能力和信息一致性表现。

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

模拟不同类型用户的操作，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。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

对管理员端功能进行全面测试，重点验证用户管理、动物信息管理和内容审核模块的敏感操作二次确认机制、违规内容自动处理与人工复核流程，以及数据统计的准确性和实时性表现。

5.2 系统测试

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

在压力测试环节，设定100并发用户进行模拟浏览页面、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，证明系统满足易用性要求。

5.2.4 安全性测试

在安全测试环节，系统通过SQLMap工具注入恶意SQL语句验证数据库防护机制，并在用户输入框植入恶意脚本检测内容过滤与转义功能，测试结果表明系统未出现显著安全漏洞。

5.3 小结

平台的功能实现均符合设计目标。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。

平台在高并发测试下系统响应迅速，具备良好的扩展性和稳定性。用户在多端适配的功能下操作体验友好。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。

6 总结与期望

6.1 项目总结

本文围绕流浪动物救助场景的数字化需求，设计并实现了流浪动物救助平台系统。平台采用调用/返回软件架构风格设计，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. Tina Rich,Emily Patterson Kane. Animal Welfare Science:An Interdisciplinary Guide[M].CRC Press:2025-02-06.
6. Shang Xianru,Ma Yanfang,Chen Siyan,Li Le,Lu Mingchong. Research and Application of Service Design-Based Models for Stray Animal Management in Universities[J].Scientific and Social Research,2025,6(12).
7. 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).
8. Alexandra Protopopova,Clive David Lawrence Wynne. Adopter-dog interactions at the shelter: Behavioral and contextual predictors of adoption[J].Applied Animal Behaviour Science,2014,157.
9. Sternberg, N., channell, R., & Muller, M. (2014).Animal shelter management: A Guide torunning a successfulanimal shelter. Humane Society Press.
10. 雷思雨,武佳雪,胡月馨.流浪动物救助站的现状及对策研究[C]//成都信息工程大学管理学院.“劳动保障研究”2024研讨会论文集（上册）.成都信息工程大学;,2024:93-96.DOI:10.26914/c.cnkihy.2024.013876.
11. 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]. 导师：张桂国. 山东农业大学, 2020.
12. 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.
13. 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.DOI:10.19387/j.cnki.1009-0592.2017.05.371.
14. 纪好.流浪动物救助公益服务设计研究[D].武汉理工大学,2018.
15. 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.
16. PetRescue UK and Ireland aiming to protect and rehome UK 'Easter Bunnies' with launch of new adoption platform[J].M2 Presswire,2021,
17. 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.
18. 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.
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. 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.
21. 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.