****

|  |  |
| --- | --- |
| **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工作主要由个人志愿者、民间组织以及收容所负责，借助线下登记、公告发布等方式来开展相关工作，然而在传统模式下，信息登记、领养申请等一系列流程都依靠人工完成，并且传播渠道较为有限，这使得救助效率以及领养透明度都有所欠缺，

鉴于此，本文着手设计并实现了流浪动物救助平台。该平台以SpringBoot与UniApp为基础搭建起社区论坛，为用户提供发帖、评论等操作支持，拓宽了信息传播的途径，运用WebSocket达成实时聊天以及消息推送的功能，采用MySql进行数据的持久化存储，构建基于事件驱动的有限自动机模型，以此简化用户提交领养材料以及管理员后台审核的流程，保证领养信息的透明度，借助RabbitMQ实现区域内志愿活动的广播，可迅速召集志愿者，利用Dubbo实现服务间的解耦以及高效通信。

**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 is mainly carried out by individual volunteers, civil organizations, and shelters, using offline registration, announcement release, and other methods to carry out related work. However, in the traditional mode, a series of processes such as information registration and adoption application rely on manual completion, and the dissemination channels are relatively limited, which leads to a lack of efficiency in rescue and transparency in adoption,

In view of this, this article focuses on designing and implementing a platform for rescuing stray animals. The platform is based on SpringBoot and UniApp to build a community forum, providing users with support for posting, commenting, and other operations, expanding the ways of information dissemination. WebSocket is used to achieve real-time chat and message push functions, MySql is used for persistent data storage, and an event driven finite automaton model is constructed to simplify the process of users submitting adoption materials and administrators reviewing them in the background, ensuring the transparency of adoption information. RabbitMQ is used to broadcast volunteer activities in the region, and volunteers can be quickly summoned. Dubbo is used to achieve decoupling and efficient communication between services.

**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

目前在我国，关于流浪动物的传统救助模式，主要依赖于线下登记、人工进行管理以及分散式的志愿者活动。然而，这种传统模式存在着诸如信息不对称、管理效率不高、领养流程不够透明等一系列问题。鉴于此，作者做出决定，要设计并实现一个流浪动物救助平台。

该平台运用层次架构进行设计[1]。其前端是基于Uni-App来进行开发的，能够达成跨终端适配的效果。后端采用的是Spring Boot[2,3]框架，此框架主要负责处理业务逻辑，进而保障系统具备良好的稳定性与高效性。在数据层方面，选用了MySQL[4]数据库，它承担着动物、用户、领养、社区等核心数据的存储与管理工作，以此来确保数据的安全性和一致性。

系统架构严格遵循模块化的设计原则，各个功能模块之间相互解耦，不仅支持独立部署，还具备灵活扩展的特性。这个平台成功实现了多项核心业务功能，比如动物信息登记、健康档案的维护、投喂记录的上传、领养申请以及审核、社区互动、志愿活动管理、后台数据统计等。

针对可能出现的高并发访问场景，平台借助缓存优化与消息队列技术，提升响应速度，同时保障数据的一致性。在安全方面，平台集成了用户身份认证、权限分级、数据加密等多种机制，能够有效防止未授权的访问以及数据的泄露情况，从而提升了平台整体的安全性。

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这一领域的研究开展得相对比较早。像是Tina Rich等人[5]所撰写的书籍，对动物福利方面的知识进行了整合，这对于相关的教育工作以及跨领域的研究起到了有力的助力作用。Shang Xianru等人[6]把服务设计理论运用到了校园流浪动物的管理工作当中，构建起了一个混合系统，从而改善了校园的生态环境，同时也提升了动物福利水平。Workman Miranda K等人[7]对Petfinder网站在猫咪领养方面所产生的影响进行了评估，结果发现，网站上猫咪信息的点击量、猫咪的年龄、毛色等因素都与领养情况存在关联。Alexandra Protopopova等人[8]则对收容所里领养者和狗之间的互动情况展开了研究，分析了那些会对领养产生影响的行为和因素。还有Sternberg等人[9]所著的书籍，对收容所管理的相关因素与关键指标之间的关系进行了评估，为提高收容所的管理效果提供了数据支撑，并且给出了相应的改进建议。上述这些研究从多个不同的角度和层面，有力地推动了动物福利和管理工作的发展。

1.2.2 国内应用现状

国内针对流浪动物问题的研究始终处于不断深入推进的状态。在流浪动物救助站的研究领域，雷思雨等人[10]明确指出，由于流浪动物数量呈现急剧增长的态势，公益性救助站正面临着诸多棘手问题，其中包括资金与物资的严重短缺，以及运营管理过程中遭遇的重重困难。以兰州流浪动物救助站为例，其在运营过程中承受着巨大压力。

在互联网技术的应用层面，姜苏[11]强调搭建线上线下相融合的救助系统具有极为重要的意义。该系统能够充分发挥互联网公益的独特优势，有效调动广大公众积极参与到流浪动物救助行动中来。不过，目前诸如智能流浪猫屋 “云投喂”这样的模式，还存在着公益属性与商业属性边界模糊等问题。

在救助实践的优化方面，众多研究者提出了一系列富有价值的策略。邢月等人[12]主张通过鼓励大学生群体参与救助活动，并引入第三方机构进行协调等方式，提升救助工作的成效；张秋雨[13]以宜宾地区的实践为例，着重强调了实施针对性帮扶措施以及开展TNR（诱捕-绝育-放归）项目的重要作用；纪好[14]从服务设计的视角出发，构建出了一套完整的流浪动物救助服务系统蓝图。

尽管国内在流浪动物救助研究领域已取得了一定成果，但在救助站的高效运营管理、互联网救助模式的规范化等关键方面，仍然有许多未知领域等待进一步探索与完善。

1.3 论文结构安排

本次研究严格依照软件工程的开发流程，详细阐述“流浪动物救助平台”的设计与开发全过程。最开始，通过实施问卷调查、开展用户访谈以及进行实地考察等多样化方式，广泛收集用户需求，并深入细致地进行分析。随后，基于需求分析得出的结论，科学规划平台的系统架构与各个功能模块。最后，顺利完成系统全部功能的开发与实现工作。论文将按照以下具体框架展开论述：​

第一章为绪论，阐述流浪动物救助平台的应用场景、研究价值以及国内外同类应用的发展状况，确定论文的研究方向。​

第二章为需求分析，借助问卷调查、用户访谈等途径获取用户需求，同时评估系统的可行性。​

第三章为系统设计，基于需求分析的结果，提出平台的整体架构，详细设计功能模块，构建数据库结构，并说明模块间的交互逻辑。​

第四章为系统实现，介绍平台核心模块，如安全模块、用户模块、社区模块、管理模块的具体实现方式。​

第五章为系统测试，运用功能测试和系统测试方法，检验平台的功能完整性、性能表现、安全性及易用性。​

第六章为总结与展望，归纳平台建设所取得的成果，提出后续改进和完善的思路。

2 流浪动物救助平台需求分析

为推动流浪动物救助工作朝着规范化、信息化方向发展，平台精准锚定普通用户、志愿者和管理员三类核心角色，精心设计并成功实现了集动物信息管理、投喂记录留存、领养申请处理、社区互动交流以及后台综合管理等功能于一体的流浪动物救助平台。接下来，我将紧密结合实地调研情况，深入剖析不同用户群体的功能诉求，同时对系统的非功能性需求展开探讨，并全面论证该平台建设在实际应用中的可行性。

2.1 流浪动物救助平台功能性需求

通过问卷调查、深度访谈及实地勘察等方式，对流浪动物救助领域相关利益群体开展调研，明确“流浪动物救助平台”主要用户为普通用户、志愿者和管理员。不同角色被赋予差异化权限，对应不同功能模块，基于调研数据深入分析各角色功能需求，具体用例图见图2.1。

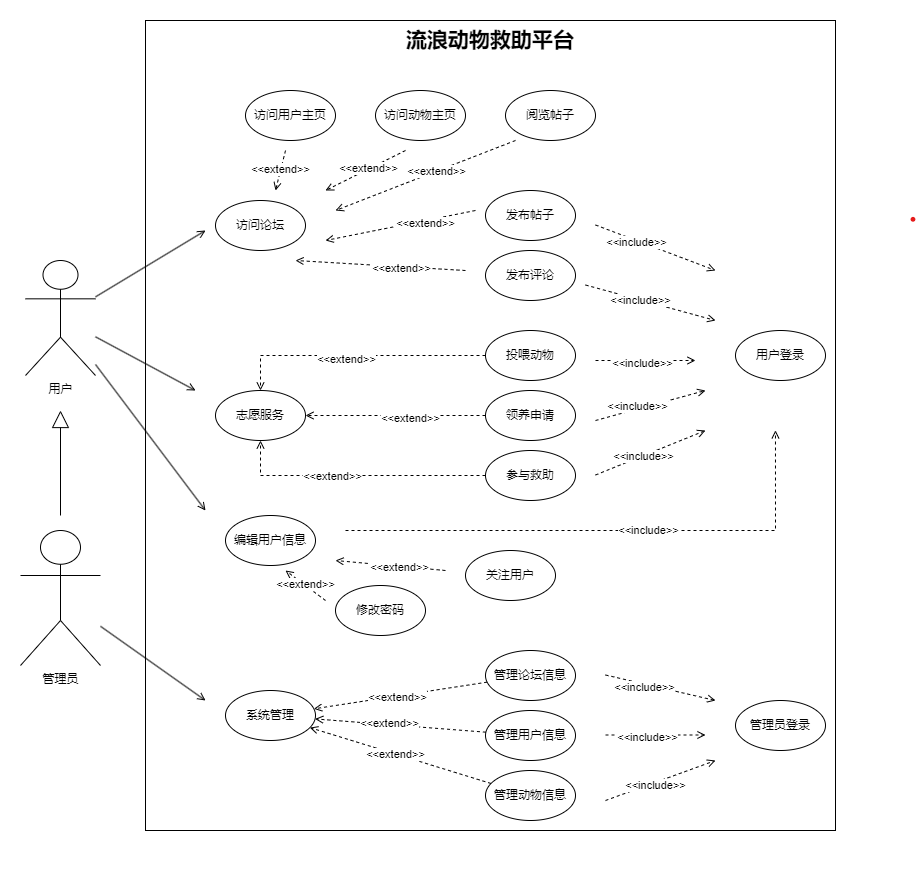


图2.1 用例图

2.1.1 普通用户功能

浏览论坛：用户可以查看流浪动物的基础信息、论坛帖子详情以及用户资料等内容。

内容互动：用户完成登录后，可在平台进行创作与交流，包括发布帖子分享想法、评论其他用户内容，参与各种话题讨论。

投喂记录：登录用户能够上传投喂动物的记录，并查看当天动物的投喂情况。

领养流程：用户实名认证后，可提交相关证明材料申请领养动物，并随时跟踪审核进度。

资料管理：登录用户可对个人资料、账号绑定等信息进行修改和管理。

2.1.2 志愿者功能

志愿者登录平台后，可查看系统发布的各类救助活动信息并在线报名，同时还能对已参与的活动记录进行查询、整理和管理，方便了解志愿服务情况。

档案维护：协助录入并更新救助动物的健康信息和救助进展。

沟通反馈：通过平台与其他志愿者、管理员实时沟通，反馈救助过程中的情况和遇到的问题。

2.1.3 管理员功能

数据管理：管理员有权对动物档案、投喂记录以及用户基础信息进行全方位管理，可执行添加、删除、修改和查询操作，保证平台数据的准确和及时更新。

领养审批：管理员负责对平台收到的领养申请进行严格审核，依照相关规定和标准评估申请内容，决定是否批准领养，确保流浪动物能进入合适的领养家庭。

内容监管：管理员对社区发布的所有内容进行监督管理，对于不符合平台规范的内容，可进行审核、屏蔽或删除处理，打造健康良好的社区交流氛围。

权限管控：管理员通过分级管理方式，为不同用户精准设置权限，以适应平台的多样化运营需求。此外，为保障平台安全和秩序，管理员有权冻结存在违规行为或安全隐患的用户账户。

2.2 流浪动物救助平台非功能性需求

2.2.1 性能需求

平台具备支持高并发访问的能力，即便处于高并发用户访问的情形下，也能确保响应时间控制在 2 秒以内。

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

当系统的部分组件出现故障状况时，该系统具备自动实施故障转移的能力，并能迅速恢复系统的正常运行状态。

2.3 流浪动物救助平台的可行性

经济可行性：系统开发大量运用开源技术，大幅削减了开发成本。从长期视角来看，后续系统维护成本也在可控范围内，不会给项目带来沉重的经济压力，在经济方面具有较高的可行性。

技术可行性：项目选用的技术体系成熟稳定，在功能实现与性能表现上，都能充分契合本系统的开发要求。而且，项目团队成员拥有相关技术专长与丰富实践经验，在技术落地实施上具备可靠能力，能够保障系统开发工作顺利开展。

社会可行性：近年来，公众对动物福利的关注度持续上升，流浪动物保护与救助成为社会各界热议的重要议题。在此大环境下，本系统的开发上线顺应了社会发展潮流与实际需求，极有可能获得社会各界的广泛支持与认可，具备良好的社会可行性。

操作可行性：系统在设计时将用户体验放在首位，界面设计直观简洁、易于理解，操作流程清晰简便、方便快捷。无论是专业救助人员，还是普通使用者，都能迅速熟悉并熟练操作，具有很强的实用性和可操作性。

2.4 小结

通过深入调研普通用户、志愿者及管理员的多样化需求，并进行细致分析，明确功能需求涉及信息浏览、发布与管理等多个维度。同时，提出性能、安全、易用性和可靠性等非功能性需求，全方位保障平台质量。此外，从经济、技术、社会和操作四个层面展开充分论证，证实平台建设切实可行，为后续的设计与实现工作筑牢根基。

3 流浪动物救助平台的设计

基于前期需求分析的成果，对流浪动物救助平台开展细致入微的设计工作，涵盖功能模块规划、系统架构搭建以及数据库构建等多个关键层面。

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。

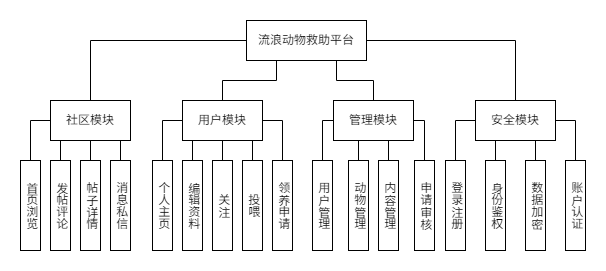


图3.1功能图

3.1.1 社区模块

社区模块作为用户互动交流的关键场所，集成了发帖、评论、点赞和私信等多元功能。在权限设置上，未注册用户仅可查看公开内容；注册用户不仅能浏览信息，还能参与发帖、评论和点赞等互动操作；已完成实名认证的用户，则可发布救援求助、经验分享等专业内容，进一步充实社区资源。

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

管理模块为管理员提供一站式管理功能，涵盖用户、动物、内容管理及申请审核。可增删用户、调整权限；编辑、增减动物信息；审查帖子评论，屏蔽违规内容；严格审核领养、认证等申请，保障平台规范运行。

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]。系统整体架构详见图 3.2，主要由用户层、网关层、应用层、领域层及基础设施层构成。

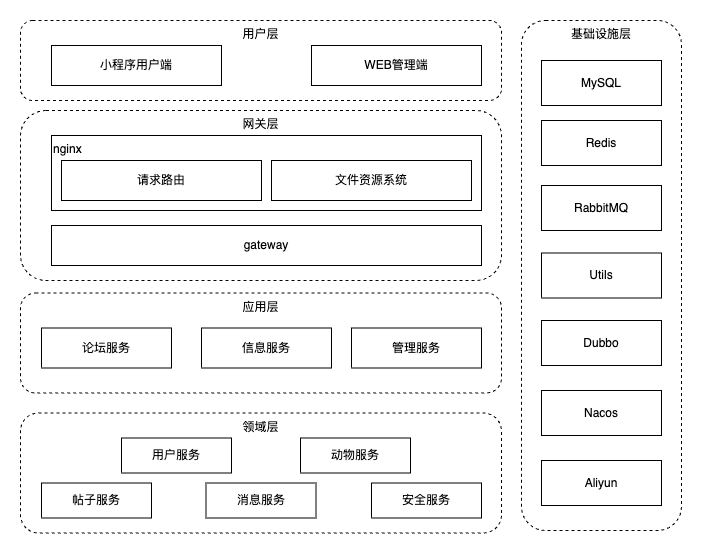


图3.2 系统架构图

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

网关层采用Nginx实现负载均衡，同时，通过搭建文件服务器并配置CDN加速。此外，gateway服务作为后端服务的统一接入点，负责处理与外部服务的交互工作。

3.2.2 应用层

应用层统筹调度各业务模块逻辑，实现系统业务操作。各服务模块独立部署，基于Dubbo[19]实现远程通信。主要包含：

论坛服务：处理社区帖子发布、评论、私信等请求；

信息服务：负责用户与动物数据的查询、更新及同步；

管理服务：执行用户、动物、内容等后台管理操作。

3.2.3 领域层

领域层专注实现并抽象平台核心业务逻辑，将业务规则封装为领域服务，核心服务如下：

用户服务：涵盖注册、登录、实名认证与权限分配；

动物服务：处理流浪动物信息登记、健康档案维护及救助记录更新；

领养服务：管理领养申请提交、审核、状态跟踪与通知；

社区服务：支持发帖、评论、活动组织及消息推送；

安全服务：负责身份校验、数据加密与权限验证。

3.2.4 基础设施层

基础设施层以Spring Boot[20]框架为核心，集成数据库、缓存、消息队列等中间件。选用MySQL进行业务数据存储；利用Redis[21]缓存高频数据、管理用户在线状态与消息；通过RabbitMQ实现异步消息处理。此外，借助Docker和Nginx完成平台容器化部署。

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。

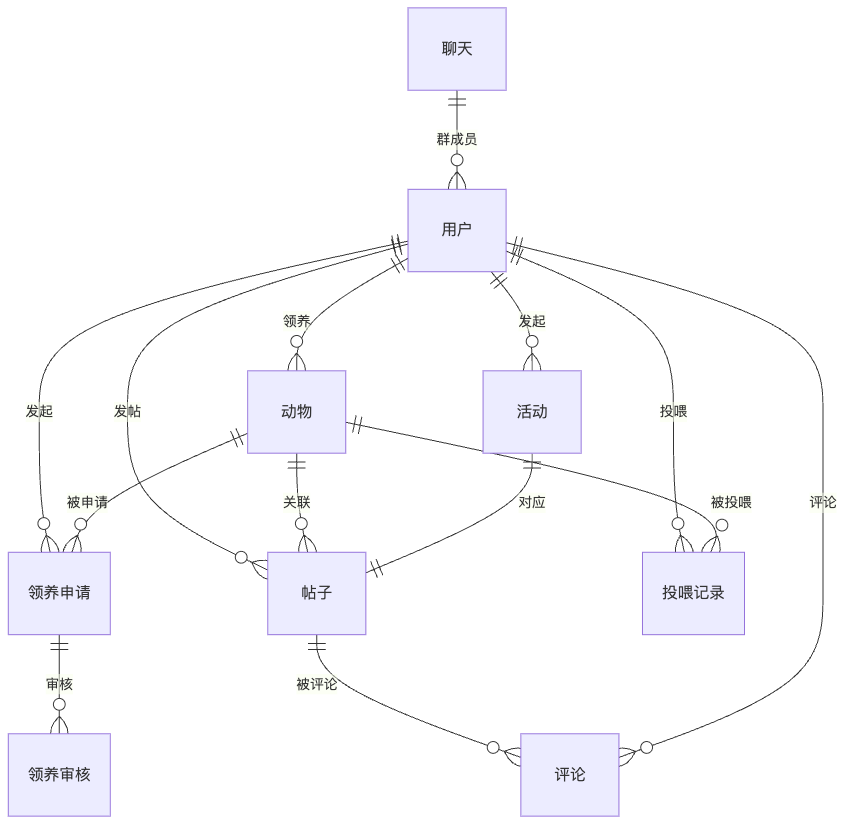


图3.3 数据库ER图

3.3.2 数据库逻辑结构设计

在概念模型向逻辑模型转化过程中，逻辑结构设计关键要点如下：

1. 数据模型映射：将概念设计中的实体对象对应至 MySQL 数据库表，如设用户账户表、动物账户表。

2. 完整性约束设置：为各表设主键 id 确保实体完整；建外键关联，如评论表 post\_id 关联帖子表 id 维护参照完整；用唯一索引、非空约束实现业务规则自定义完整。

3. 关系模型构建：一对一、一对多关系用外键引用处理，如动物表 host\_id 字段；多对多关系通过关联表，如用户关注表实现。

4. 范式化处理：消除数据冗余与传递依赖，使数据库结构达第三范式标准。

5. 查询性能提升：根据业务场景为表建合适索引，引入 Redis 缓存，减轻数据库负载，提高系统可用性。

3.3.3 数据库物理结构设计

根据E-R图和逻辑结构规划，结合MySQL数据库InnoDb引擎的特点，搭建平台的数据库表结构。表结构中不仅涵盖实体的各项属性，还额外增加ctime和utime属性，用于记录数据的创建与更新时间。通过设置version属性来实现乐观锁机制，有效避免并发操作时产生的冲突。此外，按照不同功能和业务场景的需求，为各个数据表合理设置索引，以此提高数据查询效率，优化系统整体性能。。

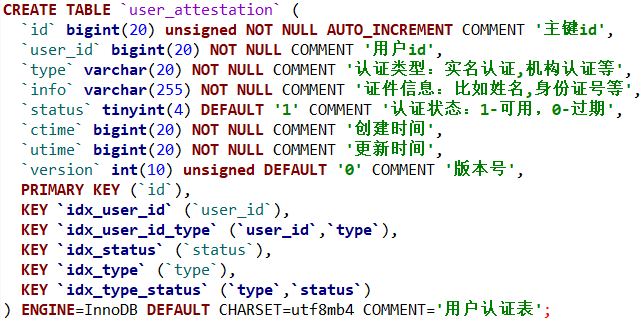


图3.4 用户认证表数据定义

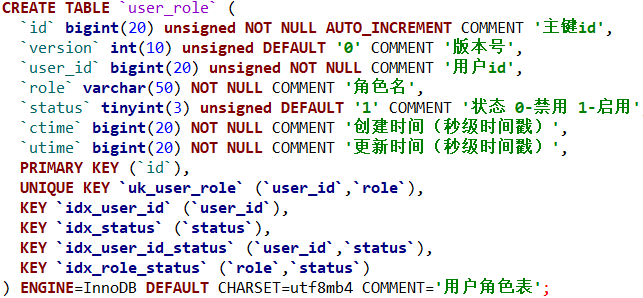


图3.5 用户角色表数据定义



图3.6 用户账户表数据定义

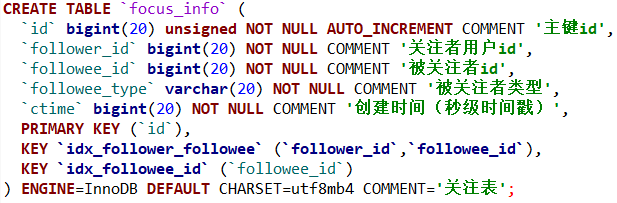


图3.7 用户关注表数据定义



图3.8 动物账户表数据定义



图3.9 领养申请表数据定义

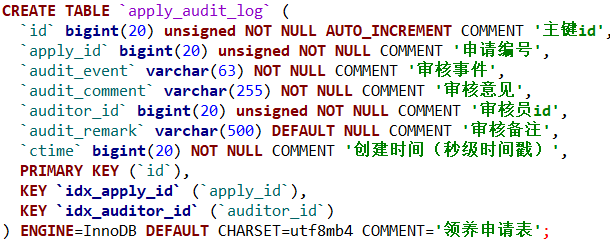


图3.10 申请审核记录表数据定义

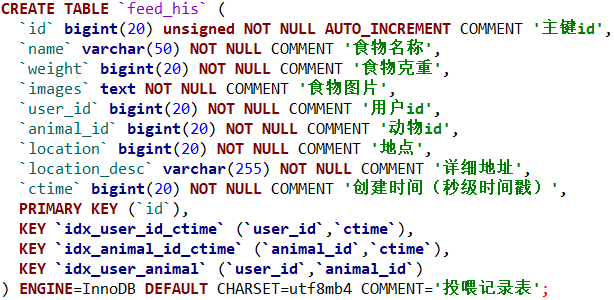


图3.11 投喂记录表数据定义

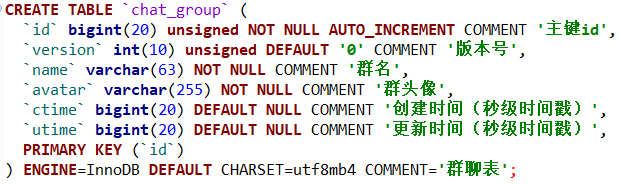


图3.12 群组表数据定义



图3.13 帖子表数据定义

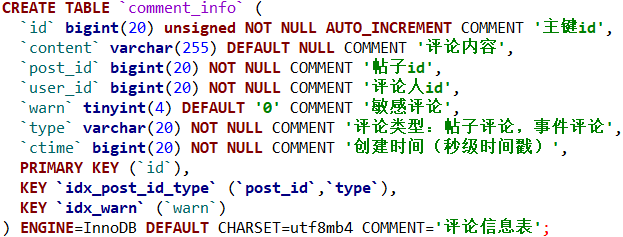


图3.14 评论表数据定义

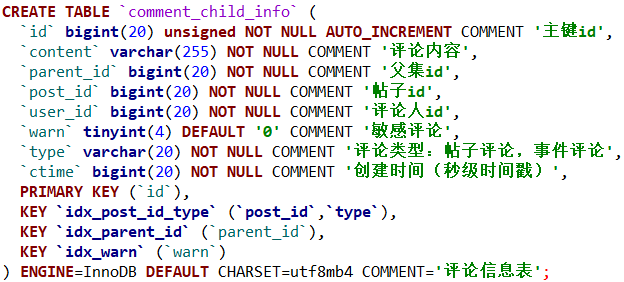


图3.15 二级评论表数据定义

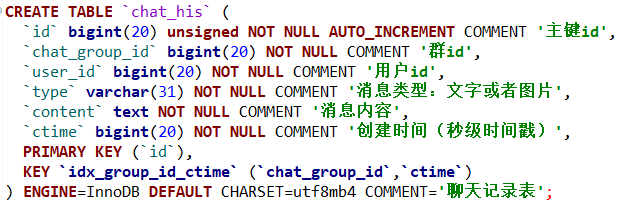


图3.16 聊天记录表数据定义

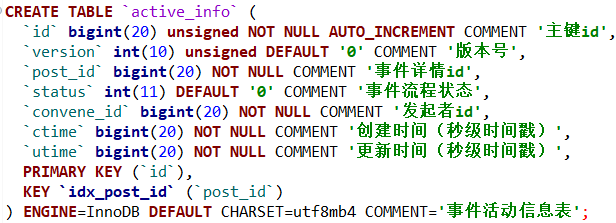


图3.17 活动信息表数据定义

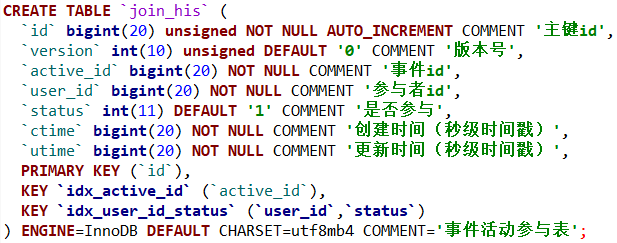


图3.18 用户活动参与表数据定义

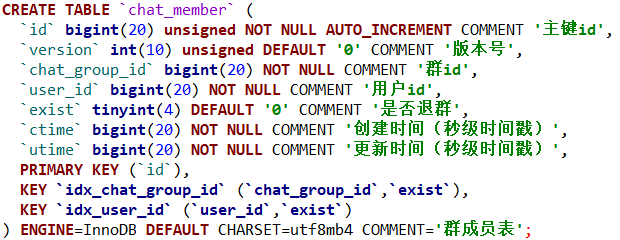


图3.19 群组成员表数据定义

3.4 小结

本章围绕流浪动物救助平台展开全面且细致的设计。功能设计上，将平台拆解为社区、用户、管理、安全四大模块，明确各模块功能范围：社区模块聚焦用户交流互动，管理模块助力管理员高效管理。架构设计采用微服务与分层架构，构建包含用户层、网关层等在内的五层体系，各层协同运作，保障平台性能与安全。数据库设计贯穿概念、逻辑、物理结构，经实体关系建模、数据模型转换，精心设计表结构与索引，为平台稳定运行提供坚实的数据支撑。

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

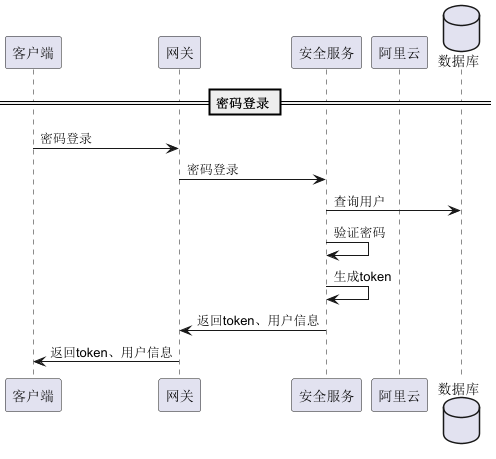


图4.2(a) 传统密码登录交互流程

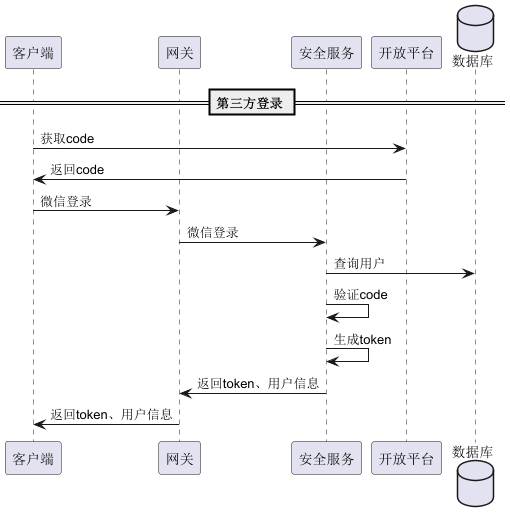


图4.2(b) 第三方Oauth登录交互流程

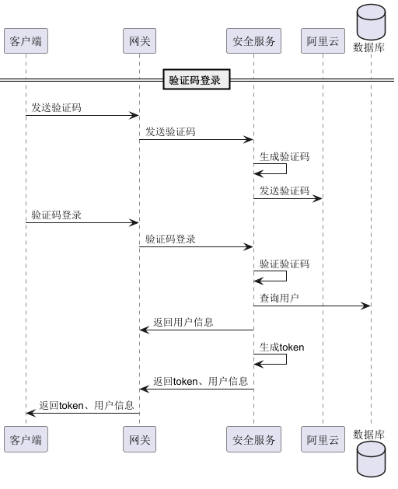


图4.2(c)短信动态验证码登录交互流程

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。

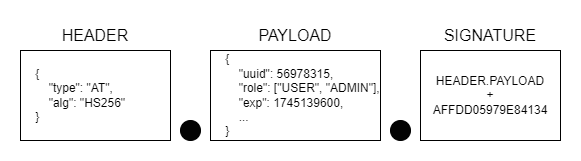


图4.3 JWT组成示意图

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。

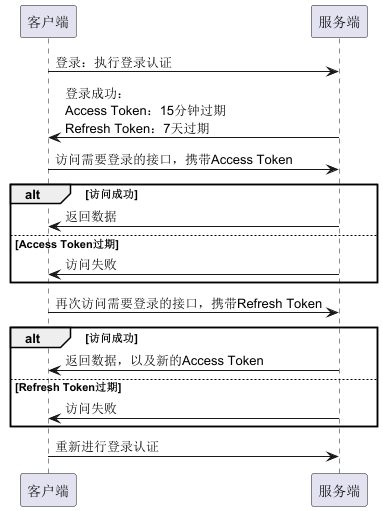


图4.4 AT-RT双token流程图

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。

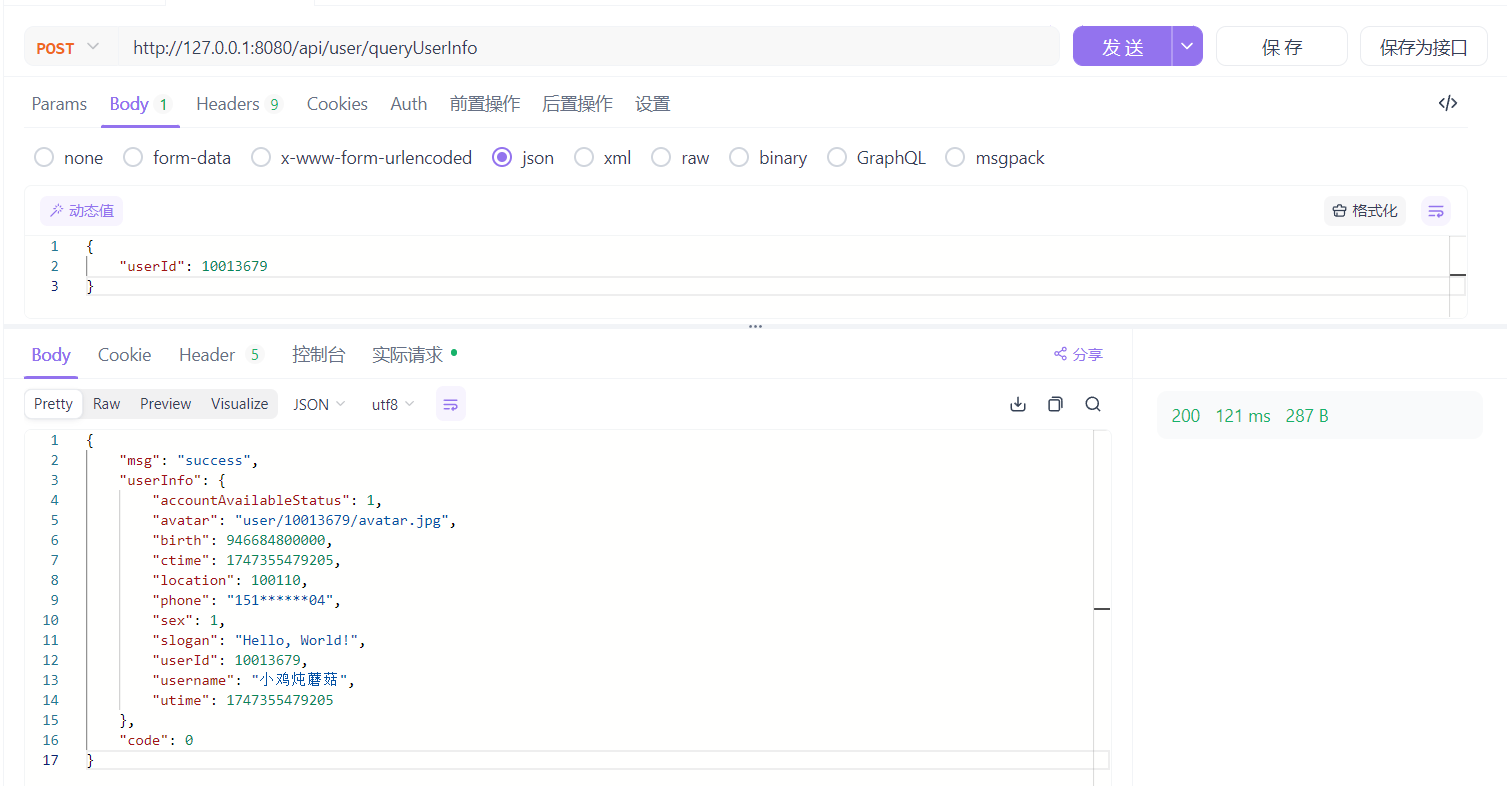
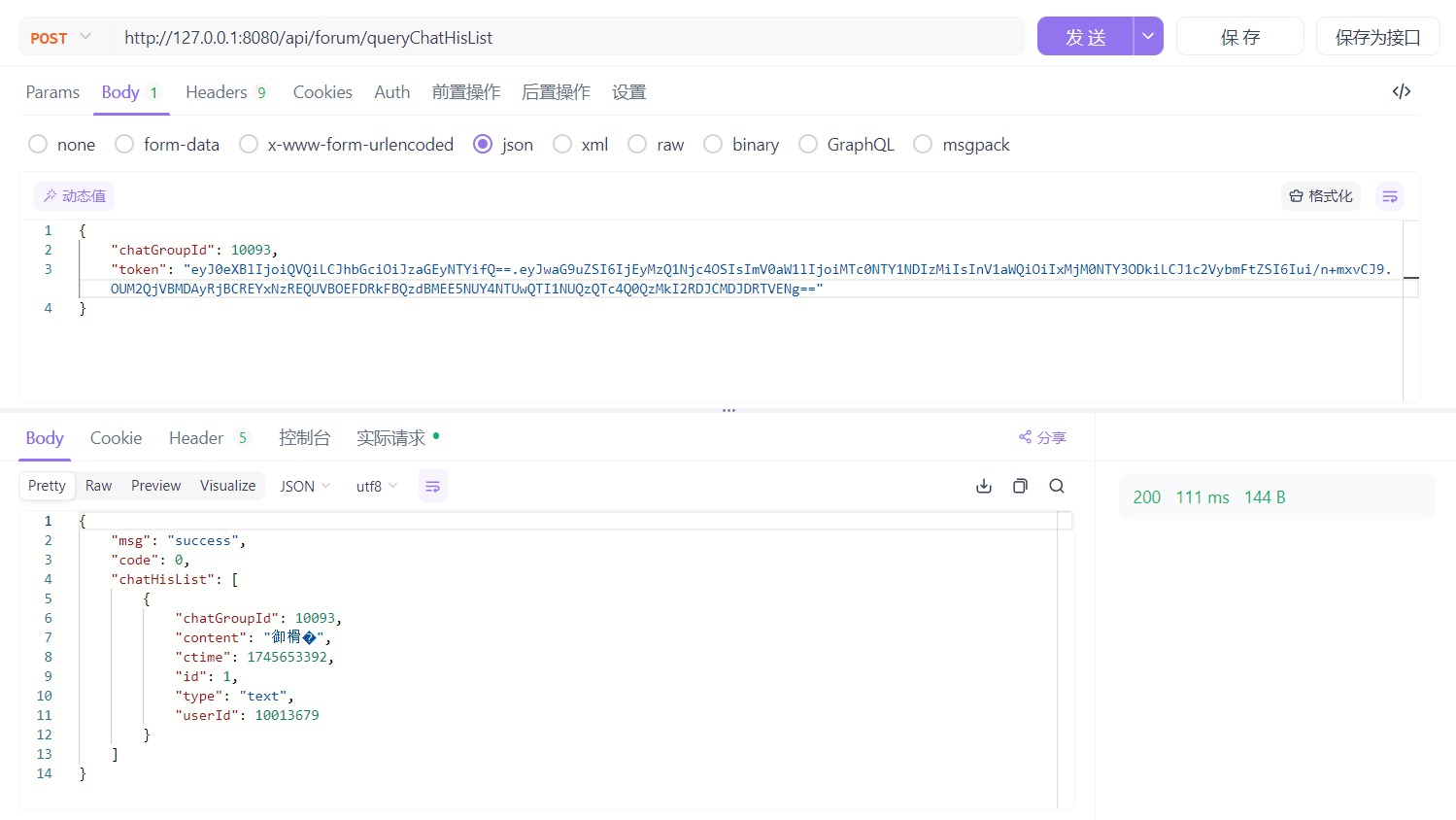
 

图4.5 数据脱敏效果图 图4.6 文本加密效果图

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。

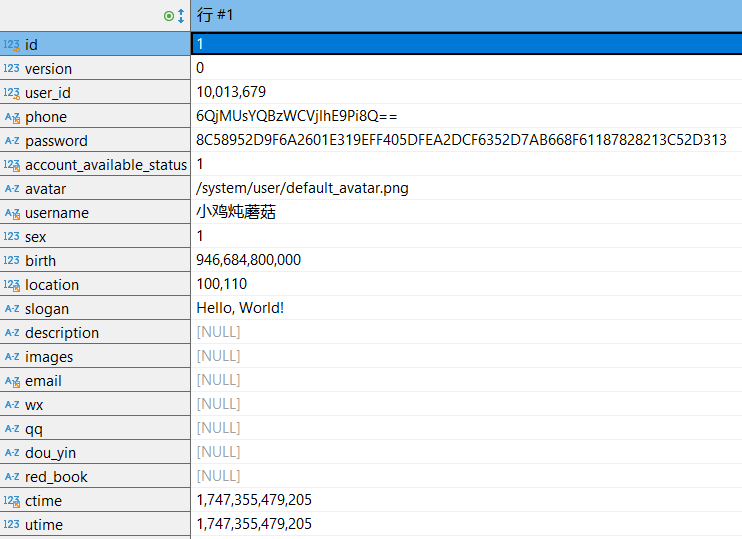


图4.7 持久化数据加密效果图

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

账户认证存在官方认证和实名认证这两种途径。进行官方认证时，用户得提交相应信息以及能够作为证明的材料，而后由平台的管理员人工核查这些材料是否真实、完整。而实名认证功能则是借助阿里云的身份二要素核验来达成的。

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，支持用户对基础信息与账户信息进行编辑修改。为保障数据安全、防止误操作，该功能集成数据预修改快照机制：用户修改基础信息时，数据暂存于本地，待点击“提交”后再发起请求更新；而修改账户信息时，则需通过手机验证码校验流程，严格验证操作主体身份，以此双重机制确保数据修改的准确性与安全性。



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。

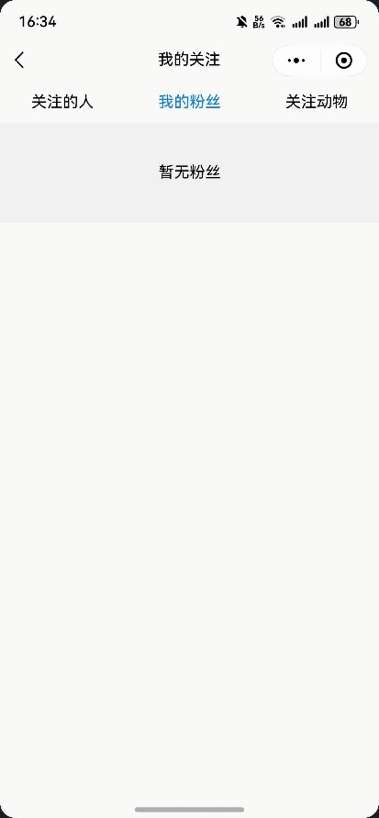


图4.10 关注列表效果图

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

依据事件驱动的原理，构建出了领养流程的有限状态自动机模型，该模型如图4.12所示。

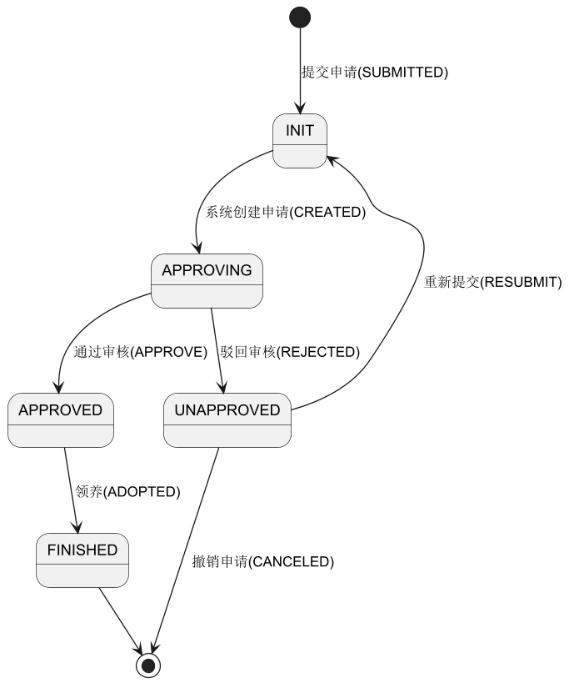


图4.12 领养申请状态图

用户提交领养申请后，领养流程状态从初始化经“提交申请”事件转为“审核中”。若审核通过，状态更新为“审核通过”；若被驳回，则变为“审核未通过”，用户可重新提交申请再次审核。在流程结束前，用户随时可通过撤销申请使状态进入终态；当审核通过且完成领养后，状态最终变更为“领养完成”，整个流程结束。

图4.13 领养申请效果图 图4.14 申请详情效果图

4.3 社区模块

4.3.1 首页预览

首页具备关键词模糊检索的功能，用户输入部分关键词，就能查询到与之相关的内容。同时，系统会轮播展示热点帖子，还能依据关注、推荐、同城、活动等不同的维度，对帖子进行聚合展示，具体的实现效果可参考图4.15。

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。

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。

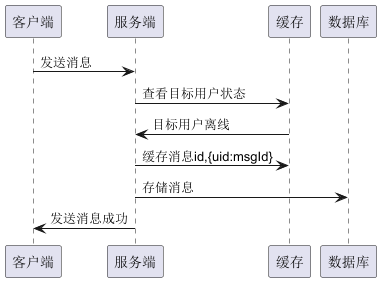


图4.20 离线消息发送交互图

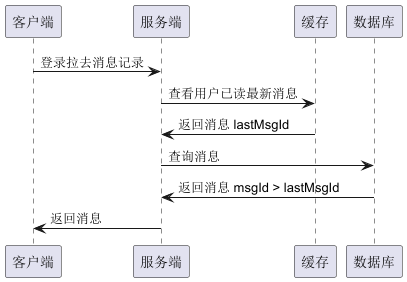


图4.21 离线消息更新交互图

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

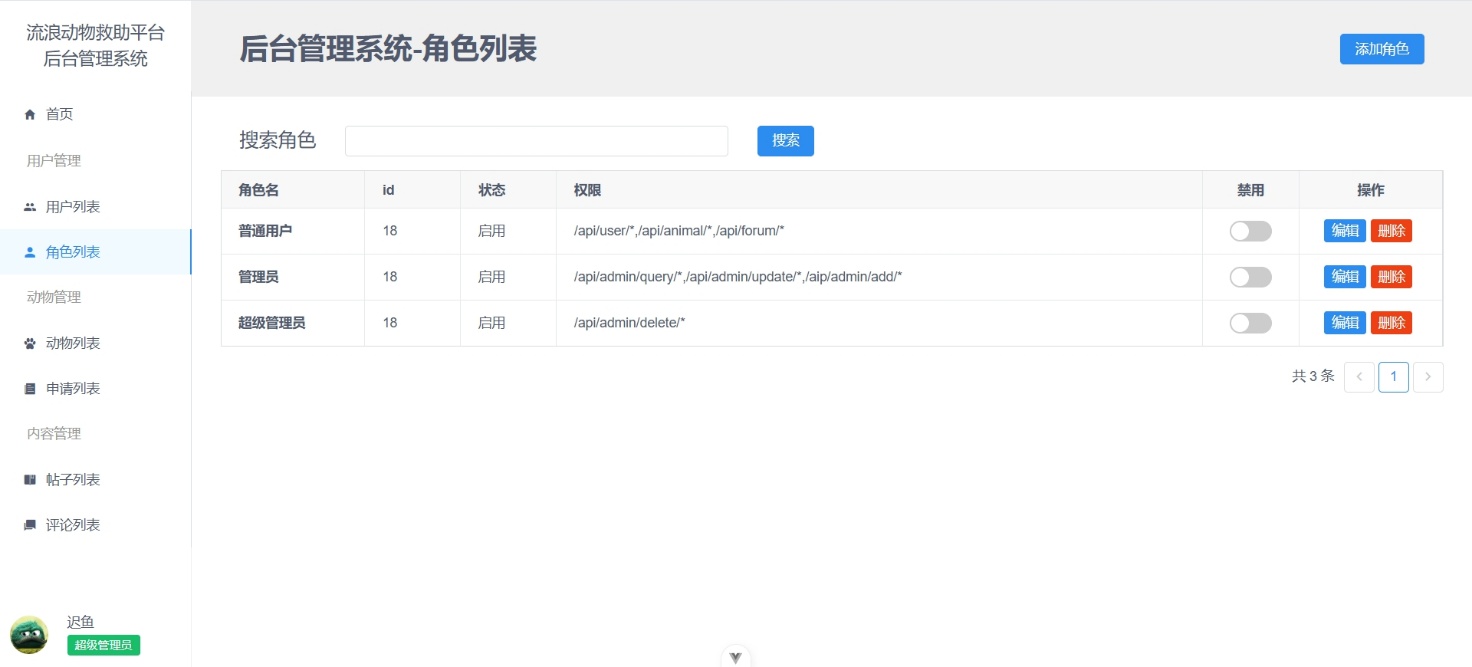


图4.23 角色列表效果图

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

审核功能的实现效果如图4.27所示。管理员可在管理端查看用户申请的详细信息，对申请内容及佐证材料进行细致校验，之后填写审核意见并执行相应操作。



图4.26 申请列表效果图



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

本章围绕流浪动物救助平台核心功能模块的实现展开深度解析。安全模块通过认证、鉴权与加密三重防护体系，筑牢系统安全防线；用户模块实现个人主页展示、资料编辑等功能，同时融入安全校验机制保障操作安全；社区模块支持发帖、评论、私信等多样化互动，高效处理用户消息；管理模块为管理员提供用户、动物、内容全流程管理及申请审核功能。各模块的成功实现，不仅为平台后续测试优化提供了可靠保障，更为平台稳定运行、服务流浪动物救助奠定了坚实基础。

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

对管理员端各项功能开展全方位测试，着重检验用户管理、动物信息管理及内容审核模块的敏感操作双重确认机制、违规内容自动化处置与人工复核流程，同时核查数据统计结果的精确性与实时性状况。

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

文章针对流浪动物救助场景的数字化发展诉求，构思并完成了流浪动物救助平台系统的构建工作。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.