



成绩	
----	--

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： 年 月 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
ABSTRACT	II
1 绪论	1
1.1 应用背景及意义	1
1.2 国内外应用现状	1
1.2.1 国外应用现状	1
1.2.2 国内应用现状	1
1.3 论文结构安排	2
2 流浪动物救助平台需求分析	3
2.1 流浪动物救助平台功能性需求	3
2.1.1 普通用户功能	4
2.1.2 志愿者功能	4
2.1.3 管理员功能	4
2.2 流浪动物救助平台非功能性需求	4
2.2.1 性能需求	4
2.2.2 安全性需求	5
2.2.3 易用性需求	5
2.2.4 可靠性需求	5
2.3 流浪动物救助平台的可行性	5
2.4 小结	5
3 流浪动物救助平台的设计	6
3.1 流浪动物救助平台功能设计	6
3.1.1 社区模块	6
3.1.2 用户模块	6
3.1.3 管理模块	6
3.1.4 安全模块	7
3.2 流浪动物救助平台架构设计	7
3.2.1 用户层	7
3.2.2 网关层	8
3.2.3 应用层	8

3.2.4	领域层	8
3.2.5	基础设施层	8
3.3	流浪动物救助平台数据库设计	8
3.3.1	数据库概念结构设计	8
3.3.2	数据库逻辑结构设计	9
3.3.3	数据库物理结构设计	10
3.4	小结	15
4	流浪动物救助平台的实现	16
4.1	安全模块	16
4.1.1	登录注册	16
4.1.2	身份鉴权	19
4.1.3	数据加密	20
4.2	用户模块	21
4.2.1	个人主页	21
4.2.2	编辑资料	22
4.2.3	关注	22
4.2.4	投喂	23
4.2.5	领养申请	24
4.3	社区模块	25
4.3.1	首页预览	25
4.3.2	帖子详情	26
4.3.3	发帖评论	26
4.3.4	消息私信	27
4.4	管理模块	29
4.4.1	用户管理	29
4.4.2	动物管理	30
4.4.3	内容管理	31
4.4.4	申请审核	31
4.5	小结	32
5	流浪动物救助平台的测试	33
5.1	功能测试	33
5.1.1	用户登录注册测试	33

- 5.1.2 宠物管理与投喂功能测试 33
 - 5.1.3 社区互动与论坛交流功能测试 33
 - 5.1.4 领养申请与审核流程测试 34
 - 5.1.5 后台管理功能测试 34
- 5.2 系统测试 34
 - 5.2.1 响应时间测试 34
 - 5.2.2 吞吐量测试 35
 - 5.2.3 易用性测试 35
 - 5.2.4 安全性测试 35
- 5.3 小结 36
- 6 总结与期望 37
 - 6.1 项目总结 37
 - 6.2 未来展望 37
- 参考文献 38

1 绪论

1.1 应用背景及意义

当前我国流浪动物传统救助模式主要借助线下登记、人工管理以及分散的志愿者活动开展，然而存在信息不对称、管理效率低下、领养流程不透明等状况，鉴于此作者设计并达成了一套流浪动物救助平台，该平台运用层次架构设计^[1]，前端依据 Vue.js 和 Uni-App 进行开发，达成跨终端适配，提高用户操作体验。后端采用 Spring Boot^{[2], [3]} 框架，承担业务逻辑处理，保证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，保证数据安全和一致性，

系统架构依照模块化设计原则，各功能模块相互解耦，支持独立部署且灵活扩展。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，平台借助缓存优化与消息队列技术提升响应速度，保障数据一致性，在安全层面，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

1.2.2 国内应用现状

国内对于流浪动物问题的研究一直处于持续发展的进程中，在流浪动物救助站的研究范畴，雷思雨等人^[10]清晰说明，流浪动物数量呈急剧上升趋势，公益性救助站面临诸多棘手难题，涉及资金与物资严重短缺，以及运营管理遭遇诸多险阻，以兰州流浪动物救助站为例，其在运营时承受着巨大压力。

在互联网技术应用方面，姜苏^[11]认为搭建线上线下融合的救助系统很有意义，该系统能发挥互联网公益独特优势，调动广大公众积极参与流浪动物救助行动，然而像智能流浪猫屋“云投喂”这种模式，存在公益属性与商业属性边界模糊等问题，

在救助实践优化上，众多研究者提出一系列有价值策略。邢月等人^[12]提议依靠鼓励大学生群体参与救助活动，引入第三方机构协调等，提升救助工作成效，张秋雨^[13]以宜宾地区实践为例，强调实施针对性帮扶措施及开展 TNR 项目的关键性，纪好^[14]从服务设计角度构建了一套完整的流浪动物救助服务系统蓝图，

虽然国内在流浪动物救助研究领域取得一定成果，但在救助站高效运营管理、互联网救助模式规范化等关键环节，仍有许多未知领域需探索完善。。

1.3 论文结构安排

本次研究严格遵循软件工程的开发流程，全面且细致地阐述“流浪动物救助平台”从设计到开发的整个过程，最初采用问卷调查、用户访谈以及实地考察等多种方式，广泛收集用户需求并加以深入分析，接着依据需求分析得出的结论，科学规划平台的系统架构以及各个功能模块，成功完成系统所有功能的开发与实现。论文将依据以下具体框架展开论述：

第一章是绪论，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情况，明确论文的研究方向。

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

第三章是系统设计，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，像安全模块、用户模块、社区模块、管理模块的具体实现方式。

第五章是系统测试，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、安全性以及易用性。

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

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。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。

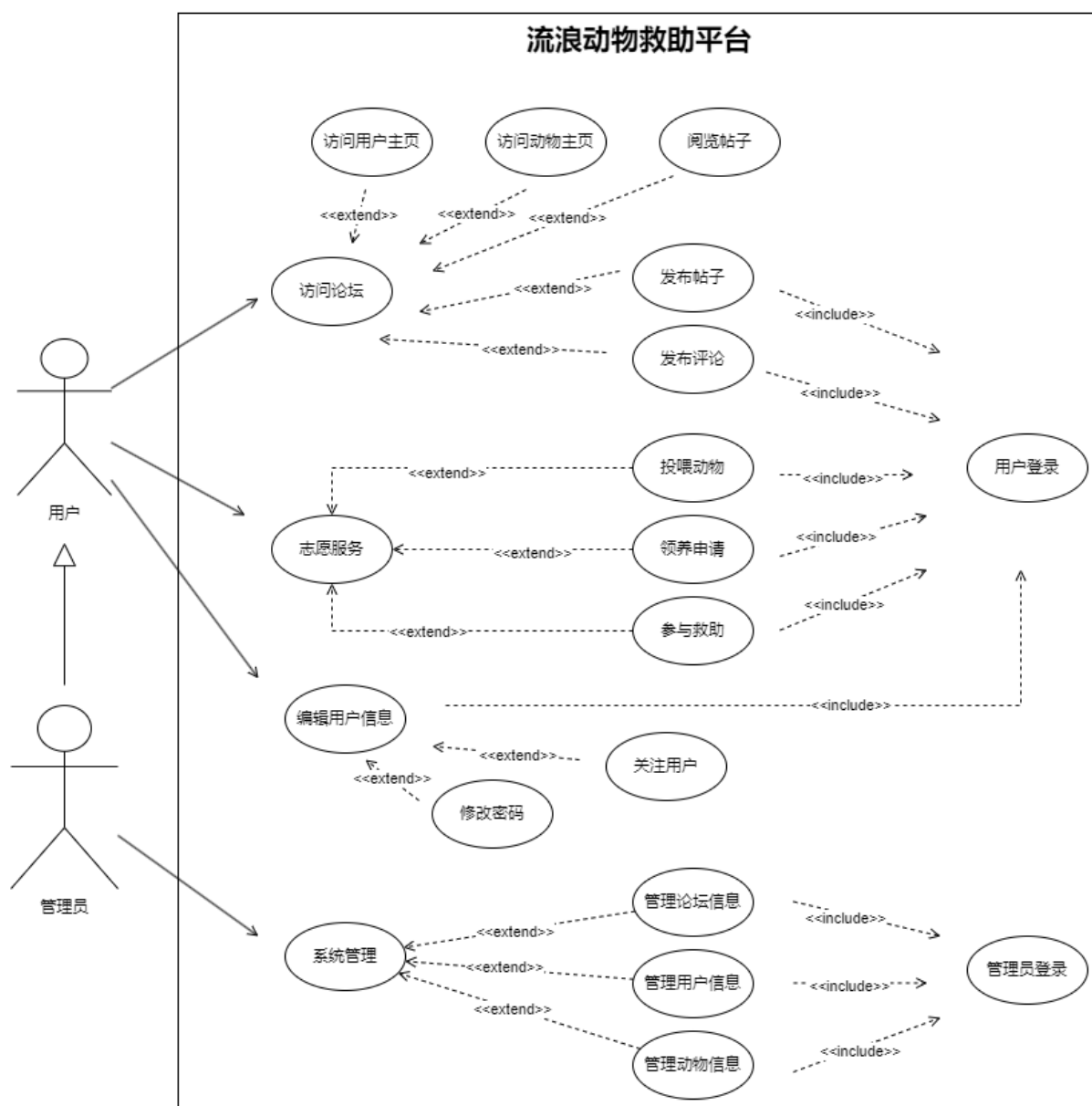


图 2-1 用例图

2.1.1 普通用户功能

浏览论坛：用户有查看流浪动物基础信息、论坛帖子详细情况以及用户资料等各类内容的能力，

内容互动：当用户完成登录操作后，便可在平台开展创作与交流活动，比如发布帖子来分享自身想法、评论其他用户所发布的内容，参与各种话题的讨论，

投喂记录：登录状态的用户可上传投喂动物的记录，并且可以查看当天动物的投喂具体情形。

领养流程：在用户完成实名认证之后，可提交相关证明材料用以申请领养动物，还可随时跟踪审核的进展状况，

资料管理：登录用户可针对个人资料、账号绑定等信息实施修改以及管理操作。

2.1.2 志愿者功能

志愿者登录平台之后，可查看系统所发布的各类救助活动信息并在线进行报名，同时还可对已经参与的活动记录展开查询，以便于了解志愿服务的实际情况。

沟通反馈：借助平台与其他志愿者、管理员实现实时沟通，反馈救助过程中出现的情况以及遇到的问题。

2.1.3 管理员功能

数据管理：管理员拥有对动物档案、投喂记录以及用户基础信息进行全面管理的权力，可执行添加、删除、修改以及查询等操作，以此保证平台数据的准确并及时更新。

领养审批：管理员承担着对平台收到的领养申请进行严格审核的职责，依据相关规定和标准对申请内容进行评估，决定是否批准领养，保证流浪动物可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，可以进行审核、屏蔽或者删除处理，营造健康良好的社区交流氛围。

权限管控：管理员借助分级管理的方式，为不同用户精确设置权限，以适应平台多样化的运营需求，另外为保障平台的安全和秩序，管理员有权冻结存在违规行为或者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

对普通用户、志愿者以及管理员的多样化需求展开调研，经过仔细分析后明确，功能需求囊括信息浏览、发布以及管理等多个维度，同时还提出了性能、安全、易用性与可靠性等非功能性需求，以此全方位保障平台质量，从经济、技术、社会和操作四个层面展开论证，证实平台建设有可行性，为后续设计与实现工作奠定基础。

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。

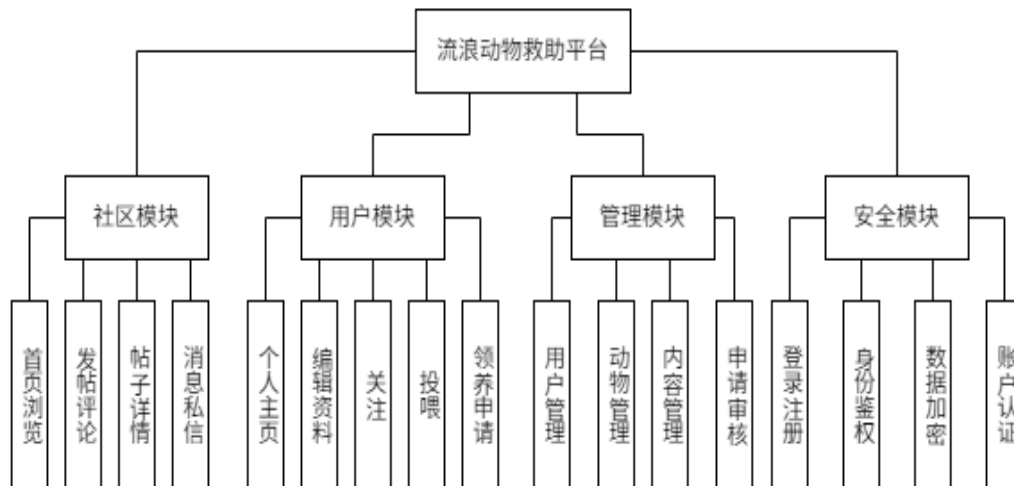


图 3-1 功能图

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。

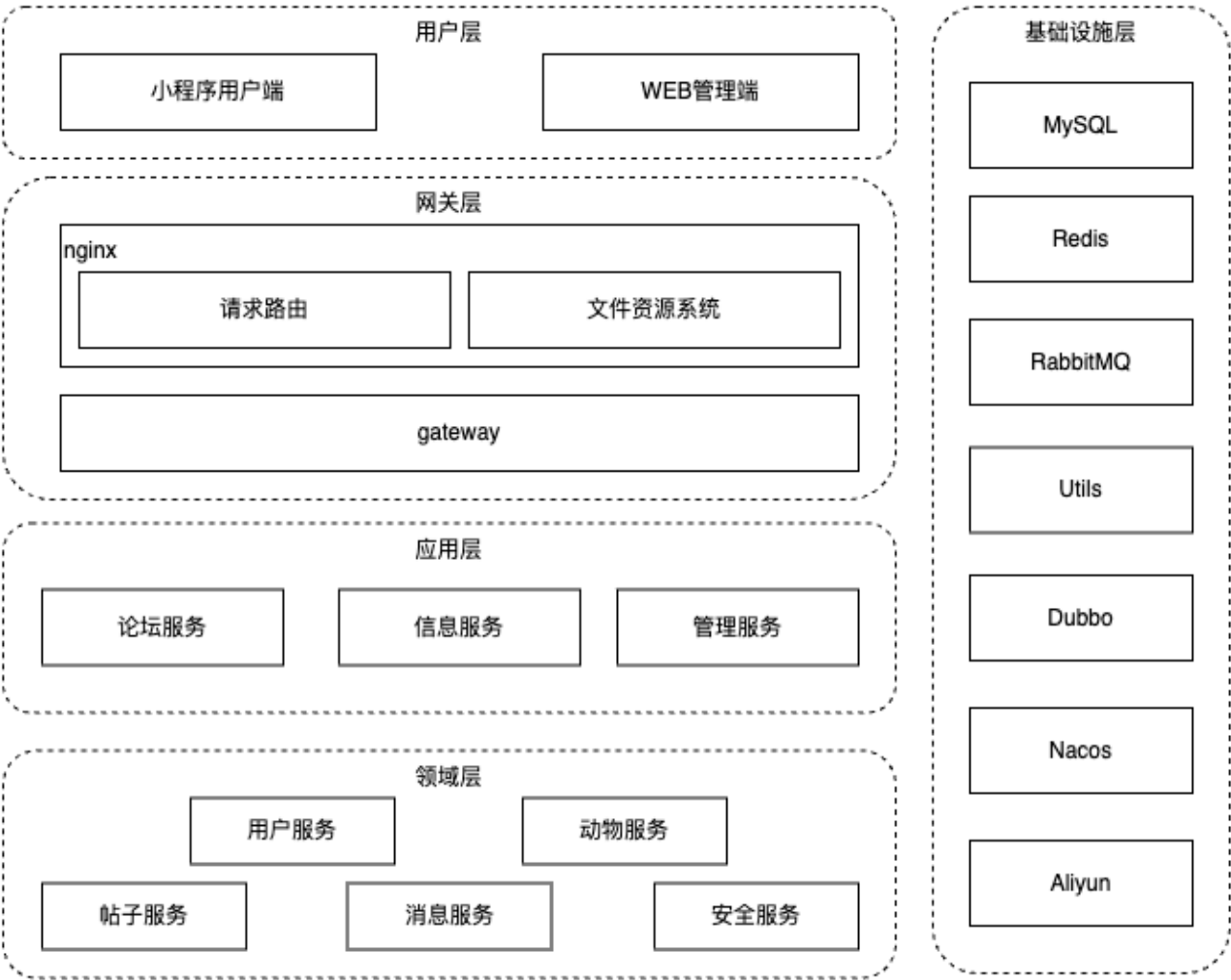


图 3-2 系统架构图

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。

3.2.2 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。

3.2.3 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。

3.2.4 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。

3.2.5 基础设施层

基础设施层把 Spring Boot^[20] 框架当作核心，将数据库、缓存、消息队列等中间件进行集成，选择 MySQL 来存储业务数据，运用 Redis^[21] 缓存高频数据以及管理用户在线状态与消息，借助 RabbitMQ 达成异步消息处理，另外依靠 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。

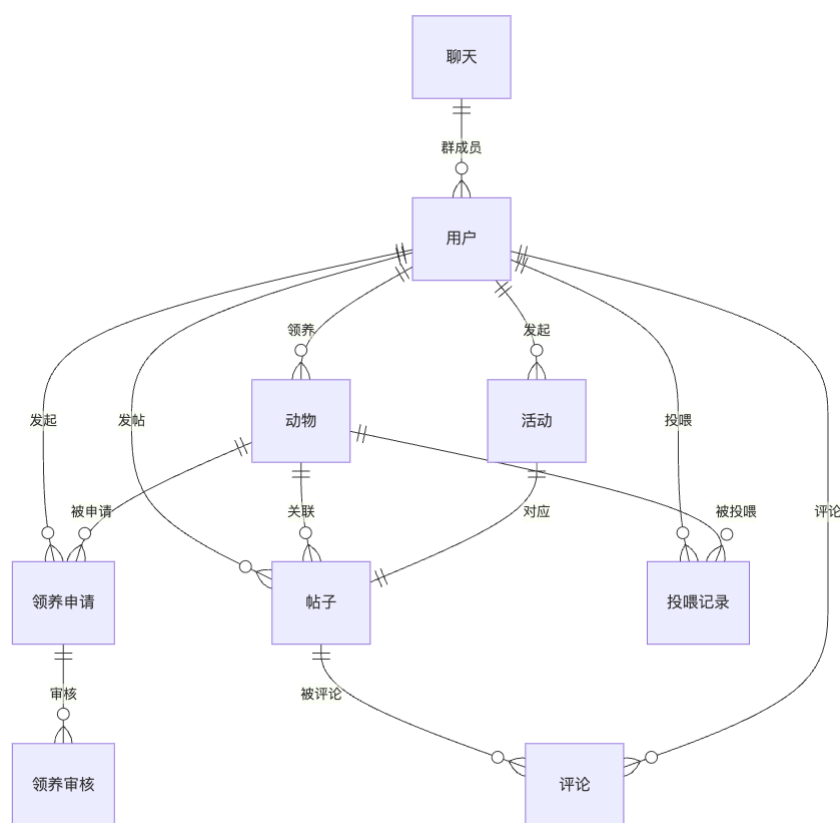


图 3-3 数据库 ER 图

3.3.2 数据库逻辑结构设计

在概念模型向逻辑模型转化过程中，逻辑结构设计关键点如下：

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

依据 E-R 图以及逻辑结构规划，结合 MySQL 数据库 InnoDB 引擎的特性，构建平台的数据库表结构，表结构里面包含实体的各项属性，还另外增添 ctime 和 utime 属性，用来记录数据的创建时间与更新时间，借助设置 version 属性达成乐观锁机制，能避免并发操作时出现的冲突。按照不同功能以及业务场景的要求，给各个数据表合理设置索引，提升数据查询效率，优化系统整体性能。

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

图 3-4 用户关注表数据定义

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

图 3-5 群组表数据定义

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

图 3-6 投喂记录表数据定义

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

图 3-7 活动信息表数据定义

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

图 3-8 用户活动参与表数据定义

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

图 3-9 用户账户表数据定义

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

图 3-10 聊天记录表数据定义

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）

图 3-11 动物账户表数据定义

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

图 3-12 帖子表数据定义

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

图 3-13 申请审核记录表数据定义

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

图 3-14 评论表数据定义

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

图 3-15 群组成员表数据定义

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

图 3-16 领养申请表数据定义

3.4 小结

在这一章节当中，针对流浪动物救助平台展开了全方位、较为细致的设计工作，从功能设计方面来看，会把平台划分成为社区、用户、管理以及安全这四个主要模块，并且明确每一个模块的功能范畴：社区模块重点关注用户之间的交流互动情况，管理模块可帮助管理员实现高效管理，在架构设计上，运用微服务以及分层架构的方式，搭建起包含用户层、网关层等在内的五层体系结构，各层之间相互配合共同运作，以此来保障平台的性能以及安全。而数据库设计则贯穿于概念、逻辑以及物理结构等方面，凭借实体关系建模、数据模型转换等一系列操作，精心设计出表结构与索引，为平台的稳定运行提供坚实的数据支持。

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(a) 流浪动物救助登录注册效果图

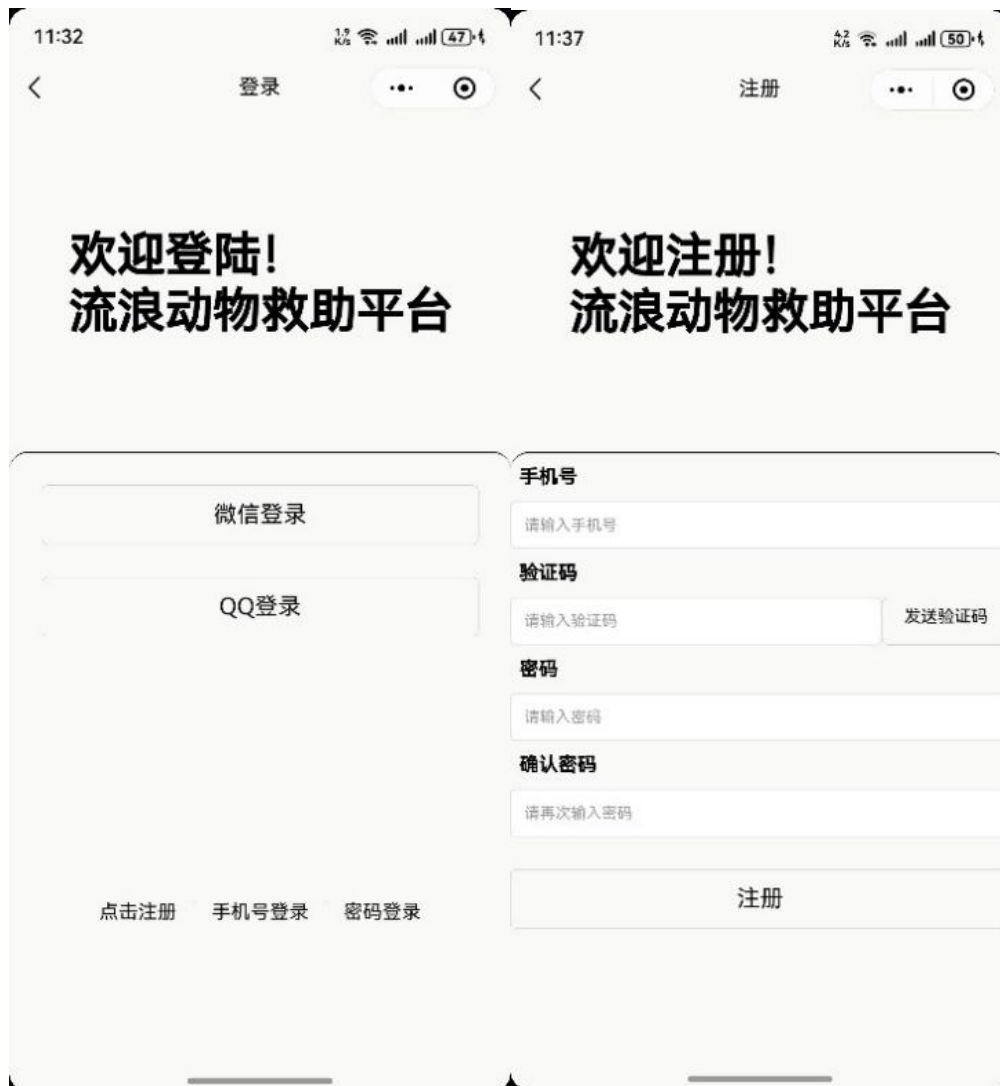


图 4-1(b) 流浪动物救助登录注册效果图

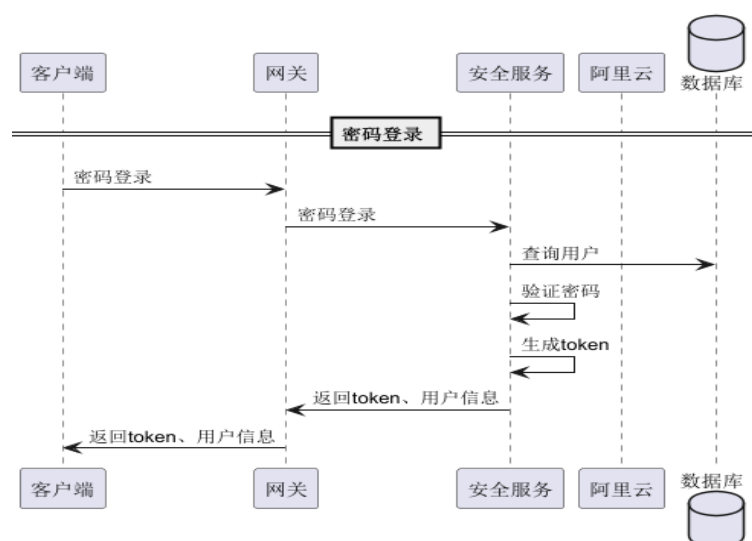


图 4-2(a) 传统密码登录交互流程

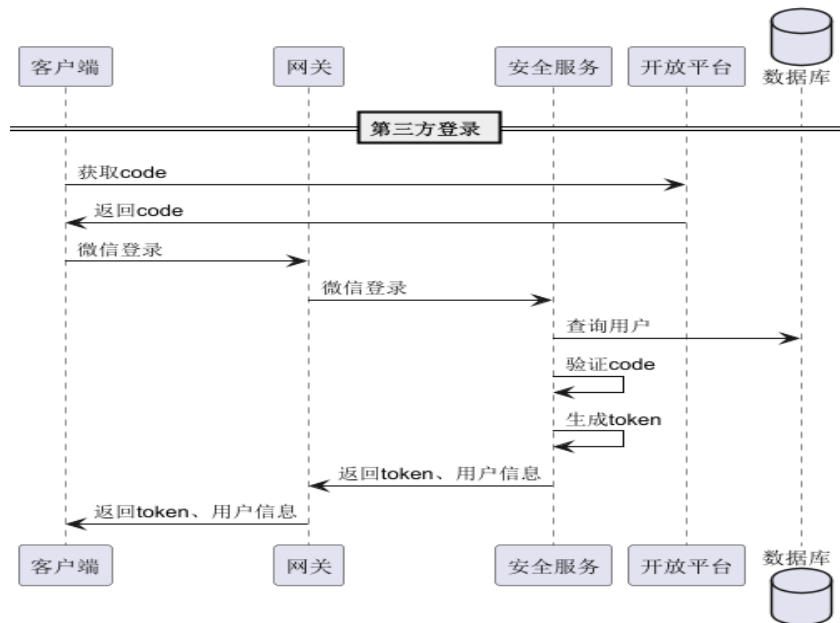


图 4-2(b) 第三方 OAuth 登录交互流程

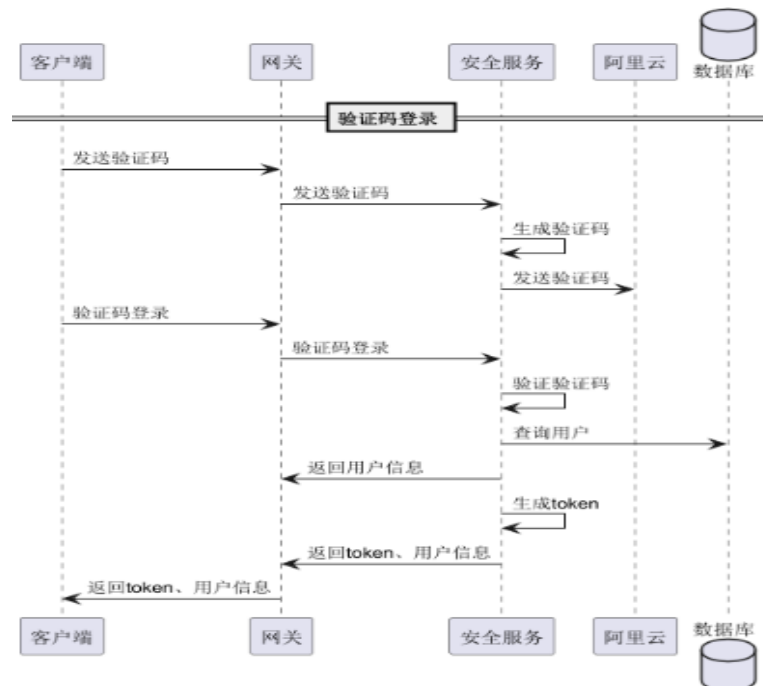


图 4-2(c) 短信动态验证码登录交互流程

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，其核心运算逻辑如公式 (4-1) 所示：

其中 K_{seed} 为服务器预置密钥， T_{epoch} 为 unix 时间戳， 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)。

$$CODE = (Hash(K_{seed} \oplus (T_{epoch} \gg 30) \oplus N_{nonce})) \bmod 10^6 \quad (4-1)$$

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。映射关系如（4-2）所示：

$$ID_{union} = SHA256(OpenID || PlatformCode) \quad (4-2)$$

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。

4.1.2 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。

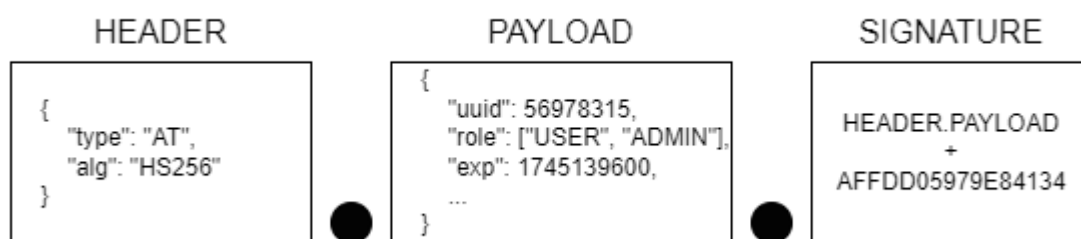


图 4-3 JWT 组成示意图

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，其令牌如（公式 4-3）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（如公式 4-4）实现。

$$token = Base64(HEADER).Base64(PAYLOAD).Base64(SIGNATURE) \quad (4-3)$$

$$SIGNATURE = SHA256(HEADER.PAYLOAD + SECRET) \quad (4-4)$$

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 4-4）保障 OAuth 流程安全性。

Access Token(AT)短期有效，用于业务请求。

Refresh Token(RT)长期有效，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。

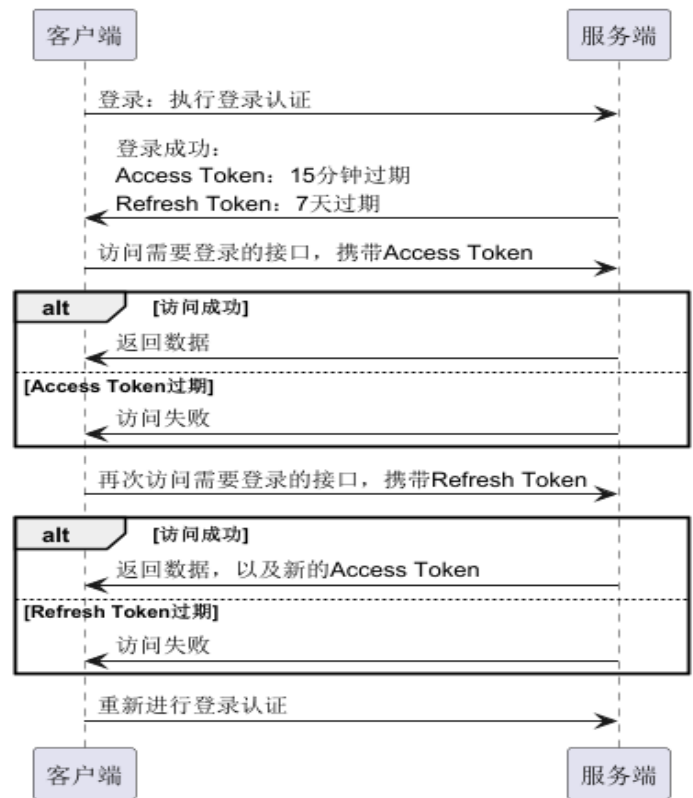


图 4-4 AT-RT 双 token 流程图

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（如图 4-5），涵盖数据对外展示与持久化存储两个环节。

	行 #1
id	1
version	0
user_id	10,013,679
phone	6QjMU\$YQBzWCVjIhE9Pi8Q==
password	8C58952D9F6A2601E319EFF405DFEA2DCF6352D7AB668F61187828213C52D313
account_available_status	1
avatar	/system/user/default_avatar.png
username	小鸡炖蘑菇
sex	1
birth	946,684,800,000
location	100,110
slogan	Hello, World!

图 4-5 数据加密效果图

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，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，对数据进行加密处理，前端接收后再进行解密。

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 4-6 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 4-6 个人主页实现效果图

4.2.2 编辑资料

编辑资料页面所呈现出的实现效果可见于图 4-7，它可支持用户针对基础信息以及账户信息展开编辑修改操作，为了保障数据的安全并且防止出现误操作的情况，该功能当中集成了数据预修改快照机制，当用户对基础信息进行修改时，数据会暂时存储在本地，一直到用户点击“提交”按钮之后，才会发起请求去进行更新。而当用户修改账户信息的时候，就需要依靠手机验证码校验流程，以此来严格验证操作主体的身份，依靠这样的双重机制，可保证数据修改有准确性以及安全性。



图 4-7 编辑资料实现效果图

4.2.3 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 4-8 关注列表效果图

4.2.4 投喂

投喂功能的实现效果如图 4-9 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 4-9 动物投喂实现效果图

1. 数据采集约束

通过微信小程序`uni.getLocation()`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：

$$Havrsine(\theta) = \sin \frac{\theta}{2} \tag{4-5}$$

$$Havrsine(\Delta\phi) + \cos \phi_1 \cdot \cos \phi_2 \cdot Havrsine(\lambda_2 - \lambda_1) \leq \sin^2 \frac{0.0015\pi}{2} \tag{4-6}$$

其中 ϕ_1, ϕ_2 为两点纬度， λ_1, λ_2 为两点经度， 0.0015π 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 `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。

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) 如公式（4-7）。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。

$$\delta = 2R \cdot \arcsin \left(\sqrt{Haversine(\theta) + \frac{(h_1 - h_2)^2}{4R^2}} \right) \quad (4-7)$$

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)。

4.2.5 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 4-10 所示。

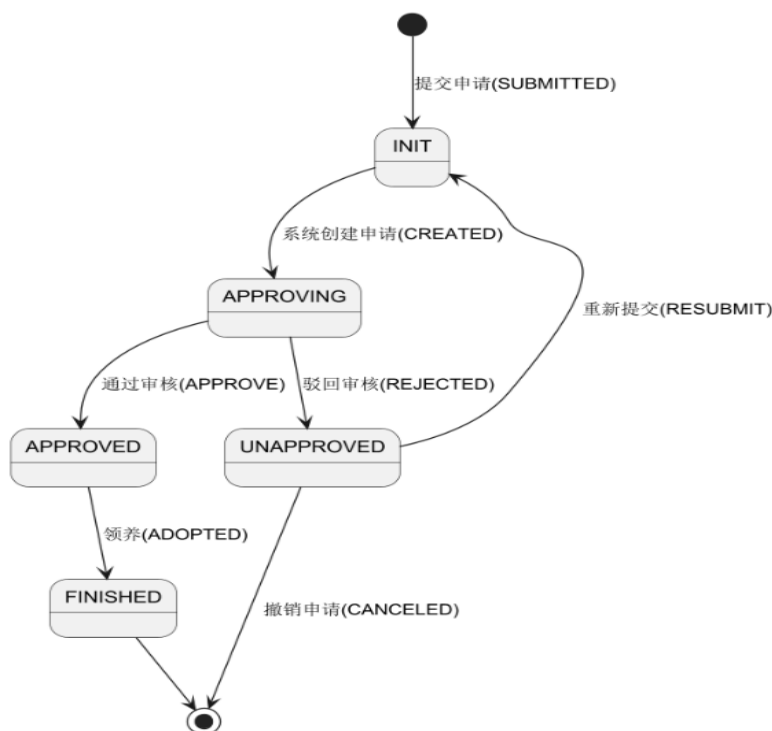


图 4-10 领养申请状态图

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 4-11 领养申请效果图



图 4-12 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 4-13。



图 4-13 首页实现效果图

图 4-14 帖子详情效果图

4.3.2 帖子详情

帖子详情界面运用分区块的布局形式来展示内容，可让用户查看主帖的相关信息以及与之关联的互动数据，此页面设有点赞、收藏以及分享等交互功能按钮，并且还准许用户依靠点击发帖人的头像，迅速访问到其个人主页。

4.3.3 发帖评论

用户可在指定页面（如图 4-15）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 4-16），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 4-15 帖子发布效果图

图 4-16 帖子详情实现效果图

4.3.4 消息私信

消息中心页面效果如图 4-17 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 U_{on} ，则有公式如（4-8）所示：

$$U_{on} = \{u_i | WebSocket_i = connected, u_i \in U\} \quad (4-8)$$

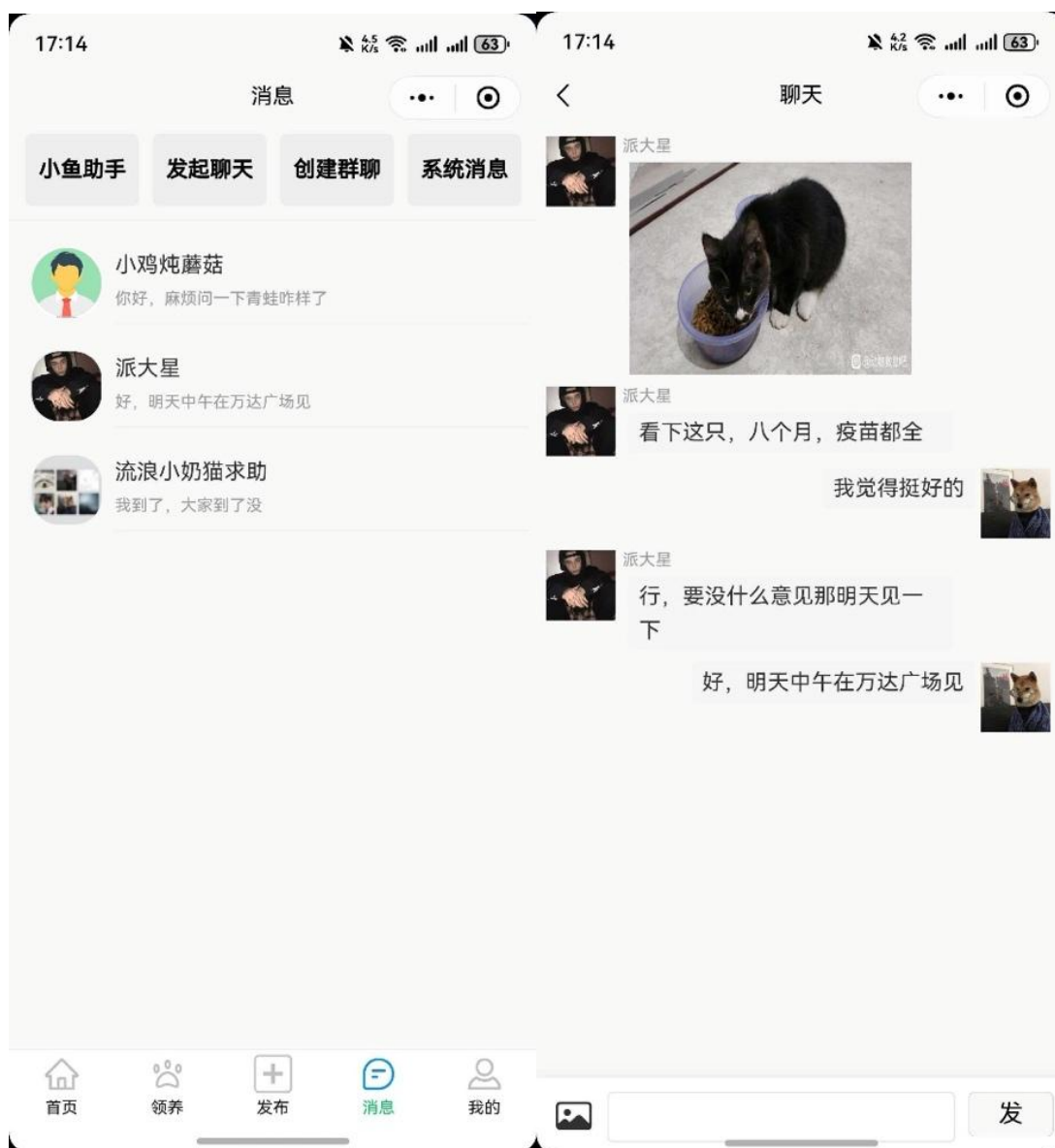


图 4-17 消息私信效果图

系统借助客户端定时发送心跳信号的方式来维持连接活性，发送间隔设定为 30 秒，当服务端接收到 PING 指令后，会即刻对用户在线状态标识进行刷新，该状态管理机制是基于 Redis 集群得以实现的，其状态更新公式如（4-9）所示：

$$S(u_i, t) = \begin{cases} 1, & \text{若在 } [t - \Delta t, t] \text{ 区间内收到心跳} \\ 0, & \text{否则} \end{cases} \quad (4-9)$$

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 4-18、4-19 所示。

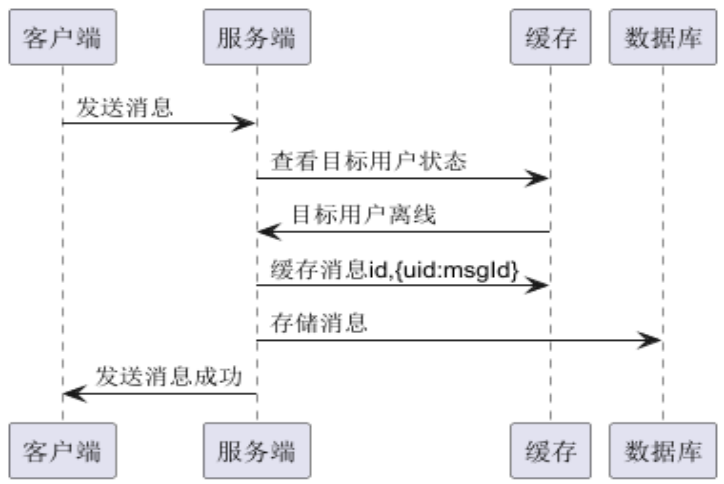


图 4-18 离线消息发送交互图

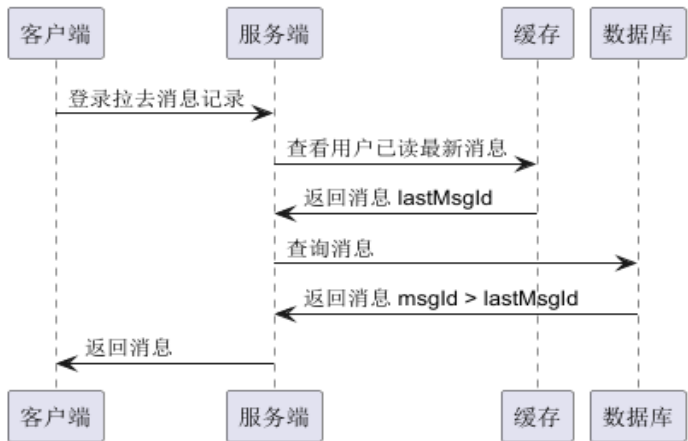


图 4-19 离线消息更新交互图

4.4 管理模块

4.4.1 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 4-20 用户列表效果图

4.4.2 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 4-21 动物列表效果图

4.4.3 内容管理

内容管理功能囊括了帖子内容管理以及评论内容管理两个方面，针对重复发布垃圾信息以及恶意评论这类行为，平台推行梯度处罚机制：首次给予警告，第二次封禁 7 天，第三次实施永久冻结，相关操作会同步触发站内信通知。

后台管理系统-评论列表							
<div>全部评论 正常评论 异常评论</div>							
评论ID	评论用户	评论内容	评论时间	状态	异常原因	屏蔽	操作
13	小鸡炖蘑菇	要是有人敢惹我，我直接杀了他	2025-04-23	待处理	杀	<input type="checkbox"/>	删除
12	小鸡炖蘑菇	建议大家出门记得带伞，天气预报说今天有雨	2025-04-18	正常	正常	<input type="checkbox"/>	删除
11	小鸡炖蘑菇	重生归来，父亲联合养子排挤我，我断绝所有关系后，父亲养子却后悔疯了。	2025-03-22	正常	正常	<input type="checkbox"/>	删除
10	小鸡炖蘑菇	哈士基：宝宝别怕，爸爸赶走坏狗狗。	2025-03-22	正常	正常	<input type="checkbox"/>	删除
9	小鸡炖蘑菇	“喝忘崽牛奶了。”	2025-03-22	正常	正常	<input type="checkbox"/>	删除
8	小鸡炖蘑菇	小狗：上一世我爹为了三个假少爷把我活活饿死。	2025-03-22	正常	正常	<input type="checkbox"/>	删除
7	小鸡炖蘑菇	免费领取名牌包包，加QQ号XXXX获取详情！	2025-03-15	已处理	免费领取QQ号	<input checked="" type="checkbox"/>	删除
6	小鸡炖蘑菇	儿：你真彪个不轻还色盲。	2025-03-22	正常	正常	<input type="checkbox"/>	删除
5	小鸡炖蘑菇	小狗：爸你看我不眼熟吗？	2025-03-22	正常	正常	<input type="checkbox"/>	删除

图 4-22 评论列表效果图

4.4.4 申请审核

审核功能的实现效果如图 4-23 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。

公子世无双对金渐层妹妹的领养申请

领养动物信息



金渐层妹妹
猫
0岁

申请人信息



公子世无双
123****01

申请状态

审核中
申请时间: 2025-04-03

佐证材料





图 4-23 (a) 申请详情实现效果图

申请信息	
居住类型	是否有院子
公寓	否
家庭成员	现有宠物
3人	1只猫
工作状态	年收入
全职	10万元
养宠经验	
养过三年猫	
饲养计划	
计划每天陪伴	
联系电话	
15112312121	

审核操作
审核意见
<div></div>
<div>拒绝 通过</div>

图 4-23 (b) 申请详情实现效果图

4.5 小结

在这一章节当中主要针对流浪动物救助平台核心功能模块的实现进行深入细致的剖析，安全模块构建起认证、鉴权以及加密这三重防护体系以此来稳固系统的安全防线，用户模块达成个人主页展示、资料编辑等相关功能，融入安全校验机制用以保障操作的安全性，社区模块支持发帖、评论以及私信等多种不同形式的互动，可高效地处理用户所发送的消息，管理模块为管理员提供用户、动物以及内容全流程管理以及申请审核等功能。各个模块得以成功实现，为平台后续的测试优化提供了可靠的保障，也为平台可稳定运行、服务于流浪动物救助奠定了坚实的基础。

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。

测试项	测试场景	测试步骤	预期结果
正常注册	手机号+验证码注册	输入有效信息注册	注册成功，跳转登录页
异常手机号校验	输入格式错误手机号	提交错误手机号获取验证码	格式错误，不发验证码
微信登录	微信授权登录	点击微信登录并授权	登录成功，关联账号
正常注册	手机号+验证码注册	输入有效信息注册	注册成功，跳转登录页

图 5-1 申请详情实现效果图

5.1.2 宠物管理与投喂功能测试

在动物管理测试里，包含了数据增加、删除、修改以及查询的整个流程操作，以此检验管理员和志愿者共同维护动物档案的效率与准确性，对于投喂功能而言，着重评估了依据地理位置的围栏限制、时间间隔控制、实时拍摄验证以及数据校验等安全机制的实际成效，借助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以及信息一致性表现。

测试项	测试场景	测试步骤	预期结果
动物档案管理	增删改查操作	后台录入/修改/删除宠物信息	数据操作成功，查询显示正确信息
围栏限制投喂	非允许区域投喂	在围栏外点击投喂按钮	不在区域内，禁止操作
高并发投喂	多用户同时提交投喂记录	模拟 100 用户并发操作	系统处理流畅，数据正常

图 5-2 申请详情实现效果图

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。

测试项	测试场景	测试步骤	预期结果
发帖权限校验	普通访客发帖	未登录尝试发帖	跳转登录页，提示需登录
敏感词过滤	发布含敏感词内容	提交含敏感词帖子	拦截内容，提示修改
离线消息同步	退出后接收私信	退出 App 后接收新消息	重新登录接收所有未读消息

图 5-3 申请详情实现效果图

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。

测试项	测试场景	测试步骤	预期结果
申请材料校验	漏传身份证照片提交申请	提交缺失材料的领养申请	提示补传材料，禁止提交
审核权限分配	普通用户审核申请	普通用户进入审核模块	提示无权限，隐藏审核按钮
状态变更通知	审核通过后通知申请人	管理员通过申请	申请人收到审核结果通知

图 5-4 申请详情实现效果图

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。

测试项	测试场景	测试步骤	预期结果
敏感操作确认	管理员删除用户账号	执行删除操作并确认	确认弹窗，记录操作日志
违规内容处置	上传违规图片至论坛	提交违规图片	自动标记，通知管理员复核
数据统计实时性	查看当日统计数据	筛选今日数据查看	统计结果延迟≤5 分钟

图 5-5 申请详情实现效果图

5.2 系统测试

5.2.1 响应时间测试

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。

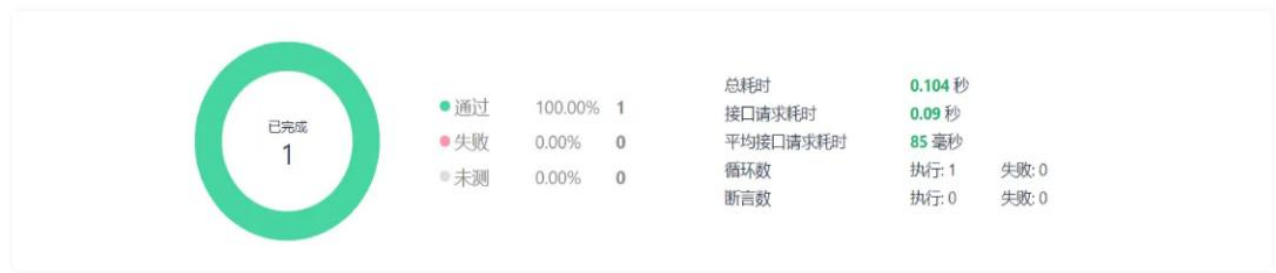


图 5-6 响应测试结果

5.2.2 吞吐量测试

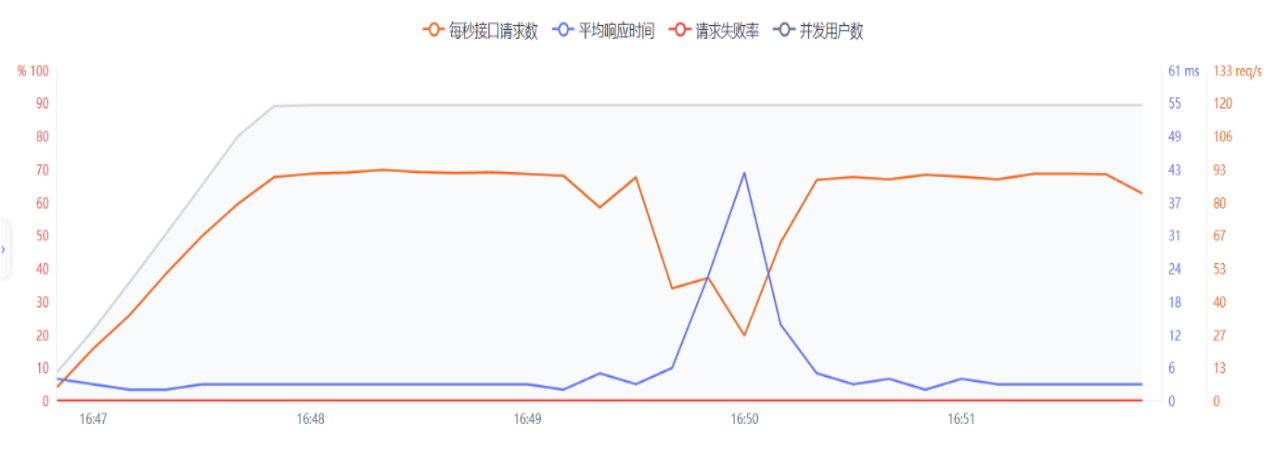


图 5-7 吞吐量测试结果

在压力测试这个环节当中，特意设定了 100 个并发用户来模拟进行浏览页面以及提交表单等一些典型的操作，测试最终呈现的结果显示，系统每分钟大概可处理 4000 次左右的有效请求，这也就说明当前系统所有的处理能力完全可以契合预先设定的业务需求，并且还拥有着不错的扩展潜力。

5.2.3 易用性测试

对于系统易用性的评估采用的是将问卷调查和用户访谈这两种方式结合起来的方法，从中挑选了不同年龄、技术水平以及需求的用户群体当作样本，调查得出的结果显示：系统界面的设计比较合理，导航也很清晰，操作逻辑十分直观，新用户在经过简单的引导之后就可完成浏览帖子以及查询动物信息等基础操作，这充分证明系统是契合易用性要求的。

5.2.4 安全性测试

在安全测试这个环节里，系统借助 SQLMap 工具注入恶意 SQL 语句来验证数据库的防护机制，同时在用户输入框植入恶意脚本来检测内容过滤与转义功能，测试的结果说明系统并没有出现安全漏洞。

5.3 小结

平台的各项功能实现都和设计目标相契合。用户注册、登录、动物信息管理、社区互动、领养申请等这些模块的功能都比较完善，操作流程也很顺畅，完全可契合流浪动物救助的实际业务需求，系统运用了多级审核、权限分级、内容过滤等一系列机制，保障了数据的准确性以及平台运行的规范性，

平台在高并发测试的情况下系统响应速度很快，有良好的扩展性与稳定性。用户在多端适配功能之下的操作体验比较友好，安全测试没有发现严重的漏洞，数据加密、权限控制等措施切实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

未来平台的优化工作会着重关注技术迭代以及功能扩展，在技术迭代方面，会引入计算机视觉技术，以此达成动物影像的智能识别，并且运用自然语言处理技术来优化内容审核的效率，提升平台的自动化处理能力，在功能扩展方面，会推动与政府监管系统以及宠物医疗平台的数据接口对接，构建起“救助-医疗-领养 - 监管”的全链路生态闭环。另外还会借助区块链技术让救助记录可上链存证，利用 VR 技术打造出沉浸式的领养体验场景，提高平台的技术赋能价值，让其成为流浪动物救助领域中高效且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 to running a successful animal 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.