



成绩

(采用四级记分制)

西北大学

本科毕业论文（设计）

题目：流浪动物救助平台的设计与实现

学生姓名 俞家宝

学 号 2021117338

指导教师 刘晓霞

院 系 信息科学与技术学院（软件学院）

专 业 软件工程

年 级 2021 级

教务处制

二〇二五年五月

诚信声明

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

特此声明。

论文作者签名：

日 期： 2025 年 4 月 6 日

摘 要

我国宠物行业快速发展，但流浪动物数量也在增加，传统救助模式存在不少问题，比如缺乏科学管理、领养渠道不通畅、公众认知不够等，这些问题导致“救助-弃养”的恶性循环。为了解决这个情况，本论文设计并实现了一个流浪动物救助平台，平台采用微服务架构和领域驱动设计（DDD），使用 Uni-App、Spring Boot、Dubbo 等技术，具备宠物领养、社区互动、活动管理以及后台管理的核心功能。

系统功能设计包括：

1. 宠物领养模块：可以展示动物信息，追踪投喂记录，还能处理领养申请的审核流程。
2. 社区互动功能：提供图文/视频发帖、评论、关注、群聊等社交功能；
3. 活动管理模块：支持活动发布与在线报名；
4. 后台管理功能：实现用户、动物、帖子等数据的集中审核与统计。

系统采用模块化设计和微服务架构，提升了扩展性和维护性，解决了传统救助模式的难题，为流浪动物救助带来了高效便捷的方法，实际应用显示，这个平台提高了领养匹配的速度，减少了救助开支，还方便了公众参与救助工作，展现了明显的实用价值。

关键词：流浪动物救助；微服务架构；领域驱动设计；Uni-App；Spring Boot

Abstract

In recent years, China's pet industry has grown fast. But at the same time, the number of stray animals has shot up. The old way of rescuing these animals often leads to a bad cycle. This is because there is no good system to manage things, adoption options are limited, and not many people understand the issue well. To solve this problem, this paper designs and implements a stray animal rescue platform. The system uses micro service architecture and Domain Driven Design (DDD). It also brings in Uni-App, spring boot, Dubbo and other technologies. With these, the system achieves key functions like pet adoption, community interaction, activity management and background management.

The system function design includes:

1. Pet adoption module: support animal information display, feeding record tracking and adoption application review process;
2. Community interaction function: provide social functions such as text/video posting, comments, attention, group chat, etc;
3. Activity management module: support activity publishing and online registration;
4. Background management function: realize centralized review and statistics of user, animal, post and other data.

The system uses a modular design and microservice architecture. It achieves high scalability and maintainability. This setup helps solve the big problems of traditional rescue methods. It offers an easy and quick way to help stray animals. Practice shows the platform greatly boosts adoption matching efficiency and cuts rescue costs. It also gives the public an easy way to join in rescue efforts, which makes it highly valuable.

Keywords: Rescue of stray animals; Micro service architecture; Domain Driven Design; 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 国内外研究现状	1
1.2.1 国外研究现状	1
1.2.2 国内研究现状	2
1.3 论文结构安排	2
2 流浪动物救助平台需求分析	4
2.1 流浪动物救助平台功能性需求	4
2.1.1 用户功能	4
2.1.2 管理员功能	5
2.2 流浪动物救助平台非功能性需求	5
2.2.1 性能需求	5
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	6

3	流浪动物救助平台的设计	7
3.1	流浪动物救助平台架构设计	7
3.1.1	用户接口层	7
3.1.2	应用层	8
3.1.3	领域层	8
3.1.4	基础设施层	8
3.2	流浪动物救助平台数据库设计	9
3.2.1	数据库 ER 图	10
3.2.2	数据库表结构	10
3.2.3	数据库范式	10
3.3	流浪动物救助平台功能设计	10
3.4	流浪动物救助平台原型设计	11
4	流浪动物救助平台的实现	15
4.1	开发环境搭建	15
4.2	核心模块实现	15
4.2.1	宠物领养模块	15
4.2.2	社区互动模块	16
4.2.3	活动管理模块	17
4.2.4	后台管理模块	17

4.3	系统部署方案	18
5	流浪动物救助平台的测试	19
5.1	功能测试	19
5.1.1	社区论坛功能	19
5.1.2	用户信息功能	19
5.1.3	动物领养申请功能	19
5.1.4	聊天功能	20
5.1.5	后台管理功能	20
5.2	系统测试	20
5.2.1	响应时间测试	20
5.2.2	吞吐量测试	21
5.2.3	易用性测试	21
5.2.4	安全性测试	21
5.3	测试结论	22
5.3.1	测试结果	22
5.3.2	性能	22
5.3.3	易用性	22
5.3.4	安全性	22
5.3.5	可用性	22

6 总结与期望	24
6.1 项目总结	24
6.2 测试结论	24
参考文献	25

1 绪论

1.1 应用背景及意义

城市化进程加快，流浪动物问题引起社会广泛关注，为解决这一难题，人们开始搭建流浪动物救助平台，借助科技力量来应对挑战，这个平台将软件工程和领域驱动设计^[1]结合在一起，打造出一个全新的解决方案[1]等方法，采用 Spring Boot^{[2], [3]}和 Uni-App 框架实现，旨在打造一个全面、高效的救助体系

流浪动物救助平台把数据收集、分析和资源调配等功能整合起来，让救助流程更顺畅，效率更高。平台还鼓励大家参与进来，提升保护动物的意识，让人和动物相处得更加和谐，这展现了信息化时代下动物福利与城市治理的新方法。

实现平台功能需求是关键，前端用 Vue.js，保证用户体验，后端靠 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 国内外研究现状

1.2.1 国外研究现状

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 国内研究现状

张晓梅（2020）^[9]在《图书馆微信小程序应用研究》中提到，微信小程序依托微信运行，给用户带来多种服务，文章分析了公共图书馆使用微信小程序的重要意义和当前情况，根据用户的个性化需求改进了图书馆微信小程序的运营方式。

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

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

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

1.3 论文结构安排

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

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

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

第三章讲的是流浪动物救助平台的设计，先说明系统的设计目标，再根据需求分析结果确定整体架构、模块划分以及它们之间的交互关系。接着具体设计每个模块的功能、界面布

局和数据库表结构，重点放在确保设计的合理性和未来扩展性上。

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

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

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

2 流浪动物救助平台需求分析

软件开发前期，要先做项目需求分析，第一步是搞清楚业务内容和功能目标，接着，在功能模块里详细划分每个模块的使用场景，这样做能避免因架构设计问题，反复修改底层数据结构，另外，还要考虑一些非功能性因素，比如数据库性能、安全性以及网络响应速度，这些既是需求分析时的限制条件，也是项目完成后运行阶段需要关注的重点。

2.1 流浪动物救助平台功能性需求

流浪动物救助平台上有两种角色，一种是用户，另一种是管理员，这两种角色在平台上的操作权限不一样，具体的功能分配看图 2.1 的用例图。

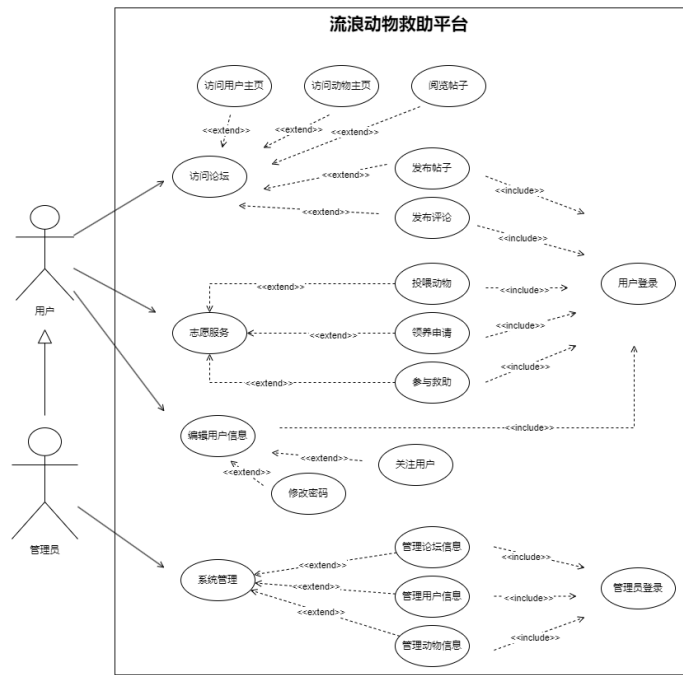


图 2.1 用例图

2.1.1 用户功能

论坛有这些功能，用户可以进入论坛，还能顺便看看其他用户的主页和动物主页，顺便浏览帖子，在论坛里，用户能发帖子，发帖子的时候还可以带上评论，这样大家就能分享信息，互动交流了。

动物救助功能让用户能投喂动物，这 and 用户登录有联系，用户可以参与志愿服务，像领养申请、参与救助这些子用例都包含在内，这能看出用户在流浪动物救助时的实际行动。领养申请、参与救助这些操作也要以用户登录为前提，这样才能保证操作可追溯，用户身份真实。

个人信息管理功能让用户能修改自己的资料，像更改密码、关注别人等，这些都能完善

用户在平台上的设置和社交互动，要编辑信息，用户得先登录。

2.1.2 管理员功能

管理员登录平台后，开始进行系统管理，他要管理论坛信息，保证内容合规有序，还要管理用户信息，控制账号状态和权限，同时，管理动物信息，及时更新资料，确保信息准确，这些操作都基于管理员登录，目的是保障系统管理的安全和规范。

2.2 流浪动物救助平台非功能性需求

2.2.1 性能需求

系统必须有很快的响应速度，这样才能让用户操作时得到及时反馈，平常使用时，像查看动物信息、浏览帖子这些常见动作，系统平均响应时间要控制在 0.5 秒之内，最长不能超过 1 秒，这样用户操作时几乎感觉不到延迟，用起来很流畅。遇到很多用户同时操作的情况，系统也要保持稳定，在预计的高峰时段，最少要能承受 100 个用户同时进行领养申请、发布评论等操作，这时系统的平均响应时间也要在 2 秒之内，保证即使有很多人同时使用，系统也不会卡顿或者崩溃，平台能够正常运转。此外，系统运行时对服务器 CPU、内存等资源的占用率要合理，不能让资源被过度消耗而影响系统性能，还要确保系统有好的扩展性，随着用户和业务量增加也能稳定运行。

2.2.2 安全性需求

平台要特别关注数据安全，严格保护用户的个人信息，像姓名、联系方式、地址这类信息，还有动物相关数据也得看好。数据传输时要用先进的加密技术，比如 HTTPS 协议，避免数据被窃取或者篡改，存储数据的时候，对用户密码等敏感信息进行加盐哈希加密，这样就算数据泄露了，别人也很难破解用户信息。用户认证和授权方面，得有严格的身份验证机制，用户注册和登录时，可以结合多种验证方式，像密码、短信验证码之类的，让账号更安全，还要根据用户角色，像普通用户、志愿者、管理员等，仔细划分权限，不同角色只能操作和访问被允许的功能与数据，防止越权访问，保证平台数据的安全和完整。

2.2.3 易用性需求

要吸引更多用户参与流浪动物救助，平台用起来方便很重要，界面设计要简单直观，布局合理，颜色搭配舒服，别让信息太复杂，用户才能很快找到功能入口。操作流程也要尽量简单，比如领养申请的步骤要清楚明白，每一步都要有提示和引导，让用户知道要填什么信息、怎么操作，少出错。

2.2.4 可靠性需求

流浪动物救助平台必须非常可靠，服务要一直保持稳定，硬件上，服务器得有备用设备，比如备用电源和多余的网络设备，这样遇到硬件出问题，系统不会因为一个地方出错就瘫痪。软件上，要用经过大量实践检验过的、成熟的技术框架和中间件，这些技术很稳当，还要建立完整的容错机制，系统部分组件出故障时，能自动转移故障并恢复，核心功能不会受影响，平台还能正常运行。数据备份和恢复机制也很重要，要定期对平台数据进行全部和部分备份，备份数据存到别的地方，防止数据丢失。要是发生数据灾难，能很快恢复数据，保证数据完整和可用，用户使用平台时也就更安心。

2.3 流浪动物救助平台的可行性

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

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

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

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

3 流浪动物救助平台的设计

3.1 流浪动物救助平台架构设计

这个系统用了微服务架构和领域驱动设计（DDD）^{[16],[17]}的理念，还结合了前后端分离的技术，目的是应对高并发、高扩展性以及方便维护的需求，系统的结构像图 3.1 展示的那样，分成用户接口层、应用层、领域层和基础层，下面讲一下具体的设计：

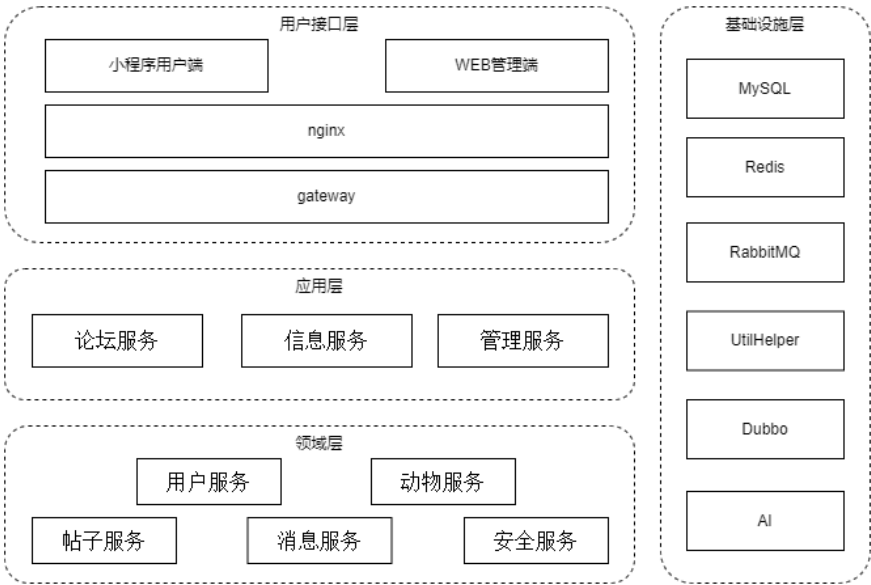


图 3.1 系统架构图

3.1.1 用户接口层

向用户展示信息，解释用户命令，再请求应用层执行用户的请求。

小程序用户端用 Uni - App 框架开发，这个框架的好处是支持一次开发，多端运行，能兼容小程序、H5 等不少平台，它借助 axios 和后端 API 交互，完成数据的传输与接收，这样一来，用户在不同终端上使用平台功能时就很流畅，像浏览动物信息、参与社区互动等功能都能正常使用。

后台管理用的是 Vue.js 框架，搭配 Element UI 等 UI 库来搭建，Vue.js 的组件化开发让后台管理的搭建更高效，Element UI 则提供了很多组件，界面设计起来很方便。管理员通过 axios 调用后端接口，能对用户、动物、帖子这些数据进行管理与审核。

前端服务器用 Nginx 来托管小程序和 Web 端的静态文件，像页面样式、脚本、图片以及系统用户资源这些都包含在内。Nginx 还通过反向代理把用户请求转到对应后端服务，这样能提升系统安全与性能，网关服务统一收 HTTP 请求，既做权限鉴权，确保只有合法用户能访问资源，又负责后端微服务路由，把请求合理分给对应微服务，保证系统稳定运行。

3.1.2 应用层

它在领域层上面，主要负责协调领域层的对象去完成任务，里面没有业务规则或者知识，不过明确了软件的应用逻辑。

论坛服务负责社区功能的运作，包括帖子的发布、查看、评论，还有活动的管理，它把领域层里的帖子服务和消息服务结合起来，给用户提供一个活跃的交流地方，推动用户之间的信息分享和互动。

信息服务负责管理用户、动物和关注等核心数据，它与用户服务、动物服务紧密配合，完成用户信息的存储、查询和更新，同时处理动物信息管理以及领养流程的跟踪工作。

管理服务主要负责 Web 管理端的功能，管理员能通过调用领域层的相关服务，对用户提交的内容进行审核，管理用户数据，这样可以全面了解平台的运行状态，快速处理违规信息，保障平台稳定运营。

3.1.3 领域层

核心业务逻辑，比如用户认证、领养流程和消息处理，都封装在一起，服务边界通过接口来定义。

用户服务主要处理用户权限、认证和角色管理，用户注册或登录时，会进行身份验证并分配相应权限，保证用户只能使用被授权的功能和查看对应的数据。比如，普通用户、志愿者和管理员的权限各不相同，用户服务可以准确区分并管理这些差异。

动物服务主要负责管理动物信息、投喂记录和领养申请，它详细记录流浪动物的品种、健康状况等内容，跟踪志愿者和用户的投喂情况，还审核领养申请，确保动物福利和领养流程的规范性。

帖子服务负责处理和存储用户在社区发布的帖子、评论，还有活动报名信息，它提供帖子发布、评论和活动报名管理的功能，保障社区互动顺利进行。

消息服务负责处理聊天记录和通知推送，用户之间的私信、系统发来的领养审核通知、活动提醒等，都能通过消息服务实现快速传递和精准送达。

安全服务负责用户登录验证和敏感词过滤，用户登录时，系统会检查身份是否合法，用户发布内容时，系统会过滤掉敏感词，保持平台环境的健康。

3.1.4 基础设施层

提供技术支持，涵盖为其他层提供持久化机制、消息传递和应用框架等内容。

多个独立服务用 Spring Boot^[18]搭建，Nacos 用来做服务注册和发现，各个微服务的注册

信息能被 Nacos 自动管理，要是服务出故障或者新加了服务，其他服务就能马上得到最新消息。Dubbo^[19]负责服务间的 RPC 通信，这种方式又快又稳，还让服务间调用没那么复杂，系统性能也提升了。

系统用 MySQL 来存结构化数据，像用户信息、动物档案、帖子内容这些，MySQL 的稳定和数据处理能力能满足系统存数据的要求，Redis 用来缓存高频访问的数据^[20]，好比热门帖子、用户会话之类，这能加快系统响应速度，减轻数据库压力。而且 Redis 可以实现分布式锁，在多线程或者多服务情况下保证数据一致。

消息传递组件用了 RabbitMQ 消息队列中间件来处理异步任务，像评论通知、活动报名确认邮件这些，用了消息队列，就能把异步任务排队处理，这样就不会因为任务处理不及时让系统性能变差

系统架构把服务模块按业务领域分开，像论坛、信息、管理这些方面，这样做的好处是让各个部分独立性更强，可以自己部署和扩展，还减少了相互之间的牵扯，Dubbo 让服务之间通信时联系不那么紧密，就算一个地方出问题也不会影响整体，系统可靠性更高了。Nginx 负责负载均衡，再搭配上 Docker 容器来部署，就能按照业务需求随时增加服务节点，让系统能同时处理更多任务。Redis 把热点数据，例如用户资料、帖子列表等存起来，系统反应速度就快了很多，领域模型明确了用户、动物、帖子等业务范围，复杂业务逻辑维护起来也方便，另外，文件上传这些通用功能放在基础服务层，不用每次都重新做，开发效率也就提高了。

3.2 流浪动物救助平台数据库设计

这个系统用了 MySQL 5.7 关系型数据库，存储引擎是 InnoDB，字符集选了 utf8mb4，数据库的设计有 16 个核心表，分成了四个大模块。

1. 用户体系模块：实现用户账户管理、角色权限控制和实名认证功能
2. 动物信息模块：记录动物档案、投喂记录等基础信息
3. 领养流程模块：管理领养申请、审核流程及操作日志
4. 社区互动模块：支持帖子发布、评论交流、群聊和活动管理

数据库设计按照三范式原则进行，表结构和索引经过合理优化，保证数据完整和查询速度快。表与表之间的主要关系用外键连接，并用乐观锁解决并发问题，设计不仅满足现在的业务需求，还为以后的扩展留出了空间。

3.2.1 数据库 ER 图

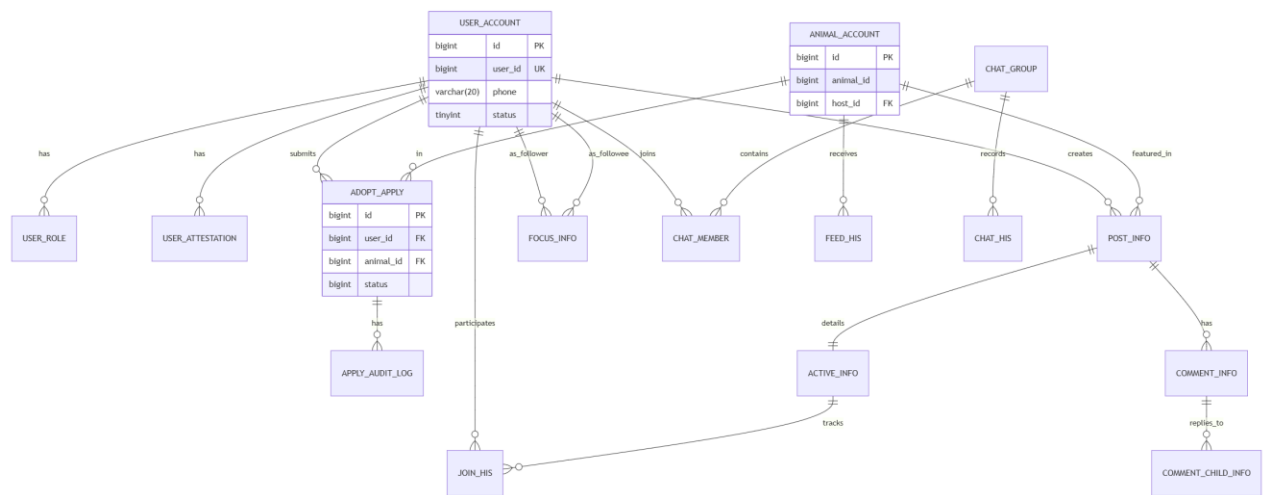


图 3.2 数据库实体关系图

3.2.2 数据库表结构

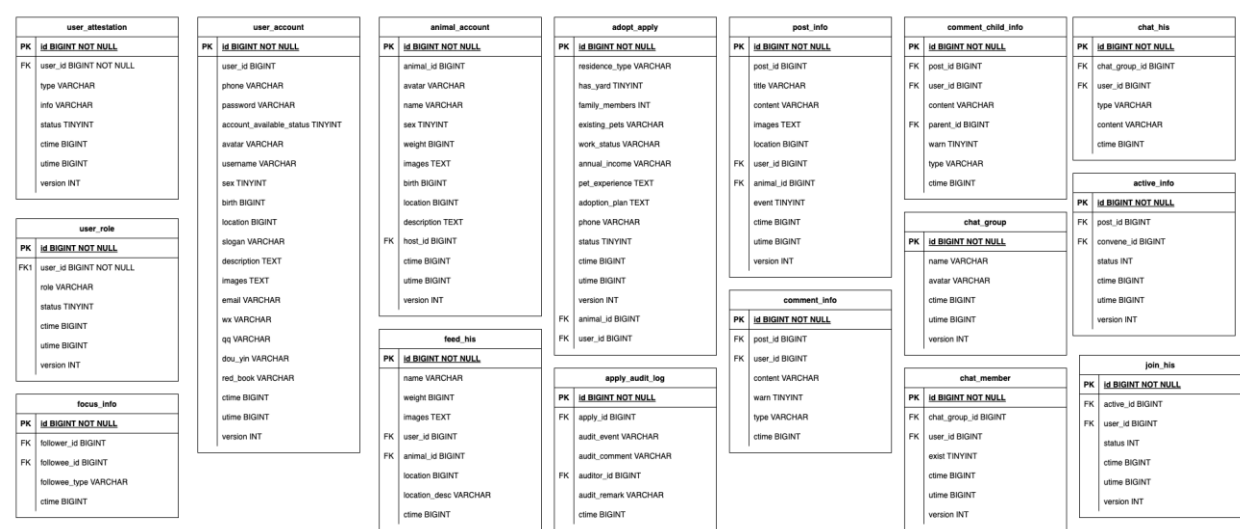


图 3.3 数据库表结构图

3.2.3 数据库范式

本设计达到第三范式（3NF）要求：

1. 每个表都有主键且无重复组
2. 所有非主属性完全依赖于主键
3. 消除传递依赖（如用户信息与认证信息分离）

3.3 流浪动物救助平台功能设计

流浪动物救助平台主要设计了四大核心模块，包括宠物领养、社区互动、志愿活动和后台管理，这样做是为了提升救助流浪动物的效率，让用户更愿意参与进来，同时方便志愿者之间更好地交流。

社区互动模块设置了多层交流系统，给不同用户提供了不一样的功能，普通访客能看公

开的求助帖和领养信息，注册用户可发基础内容并且参与评论，实名认证用户能创建救援求助、经验分享等专业帖子，系统还有实时通讯功能，用户之间能私信聊天和群组讨论。个人中心把用户资料、发帖历史、志愿服务记录等信息整合起来，组成完整的用户画像。

宠物领养模块搭建了一套完整的数字化管理系统，包含动物信息登记、健康档案更新、日常喂养记录以及领养审核流程。志愿者和管理员能录入流浪动物的基本情况、健康状态和行为特点，认证用户可以查看动物的具体信息并记录喂养情况。领养申请设置了多级审核，申请人要提交相关证明材料并通过背景核查，最后由管理员完成终审，匹配领养对象，保障动物福利和领养责任的到位。

志愿活动模块负责管理从发布到参与的整个流程，认证机构或个人能发起带有具体要求的志愿活动，系统会自动生成活动页面并开放报名入口。志愿者可以浏览活动信息、报名参加并了解实时进展，组织者还能通过专属群组安排任务，平台会保存每位志愿者的服务时长和参与记录，为评价和奖励提供依据。

后台管理模块用了权限分级机制，能全面监控平台的各个环节，管理员可以管理用户账号、审核内容、审批活动这些核心功能，系统会自动记录操作日志，保证可追溯。数据分析面板把领养成功率、活动参与度等关键指标汇总起来，帮助管理决策，敏感操作要二次验证，重要流程的审批记录会保留，这样能保障平台规范安全地运行。

3.4 流浪动物救助平台原型设计

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

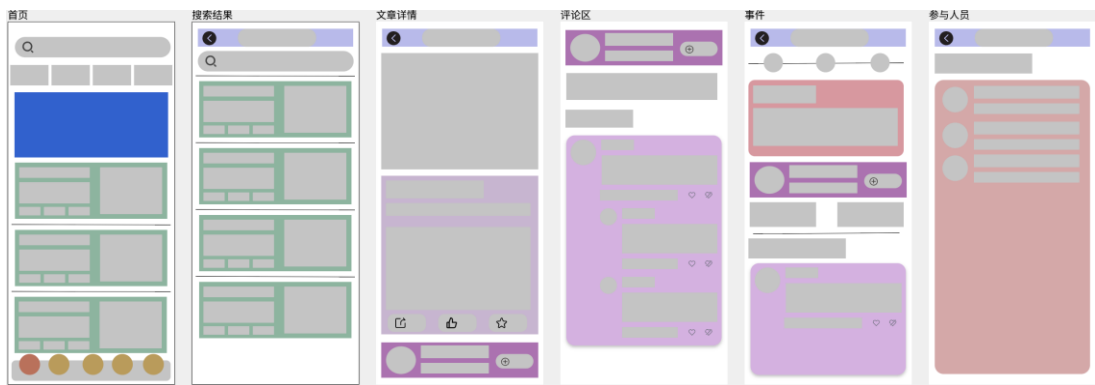


图 3.4 社区论坛低保真原型图

社区论坛的页面如图 3.4，用列表展示各种帖子，标题很明显，还有发帖人信息、发帖时间和简短的内容摘要。普通访客能看公开的帖子，注册用户可以点进帖子详情页评论和点

赞，页面底部有发帖入口，只有注册用户能看到，完成实名认证的用户能发布更多类型的帖子，像救援求助、经验分享之类的，页面上还有个搜索框，用户可以用它快速找到想看的帖子。



图 3.5 领养中心低保真原型图

领养中心的页面如图 3.5，主要显示等待领养的动物信息，每只动物都用卡片展示，卡片上有照片、基本信息和领养条件，用户点开卡片后，可以看到详细资料和喂养记录，认证用户还能在详情页申请领养，页面上方有筛选功能，能按推荐或地域等条件过滤，让用户更快找到想要领养的动物。

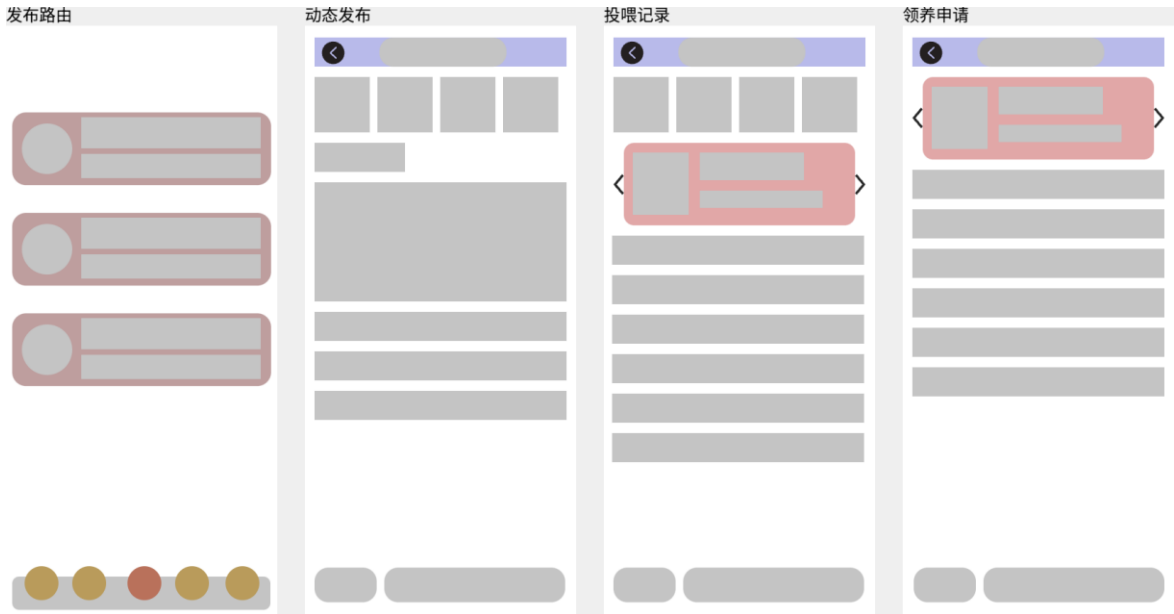


图 3.6 发布中心低保真原型图

发布中心的界面如图 3.6，只有认证过的机构和个人能看到，用来发布志愿活动和救援

求助信息，发布页面上有好几个输入框，分别用来填活动或求助的标题、具体内容、时间要求、地点、参与条件等信息。发布的人还能上传图片或者文件来帮忙说明情况，填完之后，点一下发布按钮就能提交信息，提交完要等管理员审核。

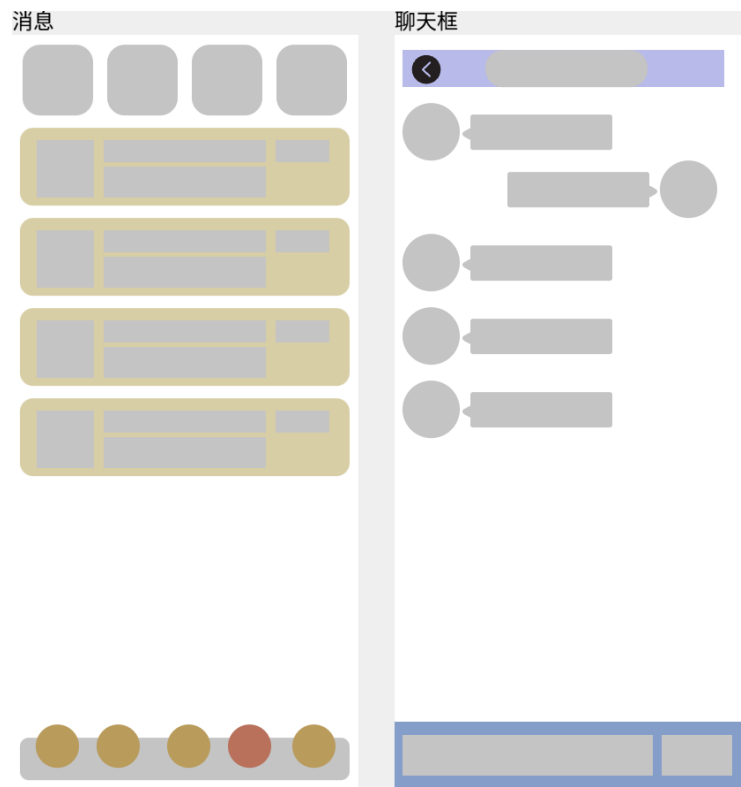


图 3.7 消息中心低保真原型图

消息中心界面如图 3.7，展示系统通知、私信和评论回复等内容，消息用列表形式排列，每条消息标明发送者、内容摘要以及接收时间，用户点开消息后可以查看具体信息，如果收到私信，还能在详情页直接回复，方便用户快速处理各类消息。



图 3.8 个人中心低保真原型图

个人中心页面如图 3.8，汇总了用户资料、发帖记录、志愿活动和服务、领养申请等内容，在用户资料区域，显示头像、昵称和认证状态等核心信息，并提供编辑选项，便于修改资料。发帖记录用列表方式列出帖子，点进去能查看详细内容并进行管理，比如删帖操作，

志愿活动部分呈现用户参与的活动名称、服务时长以及评价反馈，领养申请区域则跟踪用户的申请状态，方便了解处理进度。

4 流浪动物救助平台的实现

这一章仔细讲了系统开发时的环境、核心功能模块的开发详情，还有系统部署的办法。

4.1 开发环境搭建

流浪动物救助平台开发时，先要搭建好开发环境，这是项目顺利进行的关键，前端开发用 Node.js 当运行环境，通过 npm 管理项目依赖包，这样能快速引入开发要用到的各种工具和库。WebStorm 是个集成开发环境(IDE)，它为用 Uni-App 框架开发小程序用户端、用 Vue.js 框架开发后台管理 web 端提供代码编辑、智能提示和调试功能，这些功能让开发效率提高不少。

后端开发用的是 JDK 8，项目依赖关系通过 Maven 构建工具管理，这样能保证 Spring Boot 项目所需的技术框架和工具包准确引入并整合，数据库选了 MySQL 5.7，配置时采用 InnoDB 存储引擎和 utf8mb4 字符集，确保数据存储和处理性能，同时，Redis 服务被用来缓存高频访问的数据，提升系统响应速度，RabbitMQ 消息队列也配置好了，专门处理异步任务，让系统在应对复杂业务逻辑时更稳定高效。

4.2 核心模块实现

4.2.1 宠物领养模块

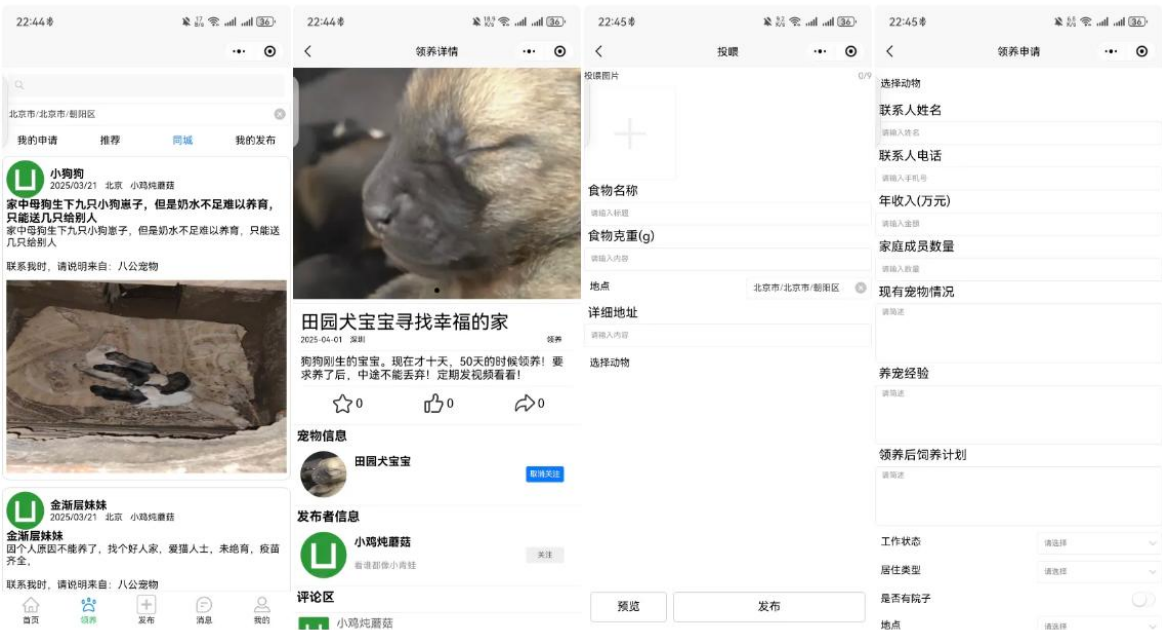


图 4.1 宠物领养用户界面

实现宠物领养模块时，动物信息展示功能负责从数据库提取详细资料，像品种、年龄、性别和健康状况这些数据都存放在对应的数据表里。前端与后端交互获取信息后，把动物资

料直观地展现出来，比如，在领养中心界面用卡片形式展示，让用户能轻松浏览和筛选。

投喂记录功能靠前后端配合完成，用户投喂时，前端先收集信息，再传给后端，后端接到信息后，把记录存入数据库，方便后面跟踪和管理动物的喂养情况。

宠物领养模块中，审核流程是关键，用户提交申请后，后端按照规则审查资质，这个环节需要核对证明材料，评估养宠能力和条件，审核完成后，结果会尽快反馈给用户，保证领养公平、公正和规范。

4.2.2 社区互动模块

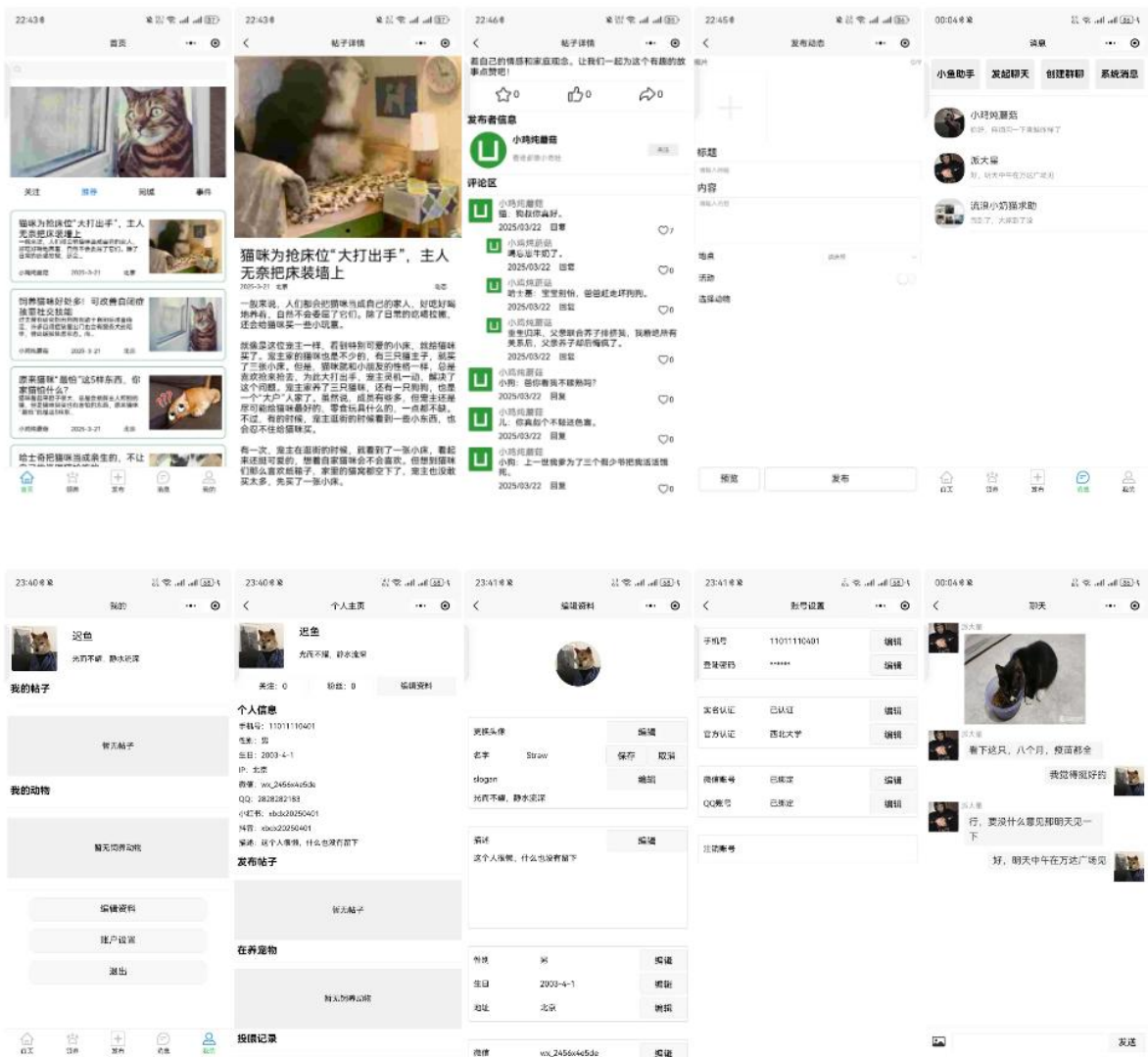


图 4.2 社区互动用户界面

社区互动模块给用户带来多种交流功能，用户能通过图文或视频发帖功能，在社区论坛分享流浪动物的故事、经验以及救助信息等内容。用户编辑完帖子，前端就把内容传给后端，后端对帖子进行格式转换和存储优化这些必要操作，保证帖子能在平台正常显示。

用户可以对感兴趣的帖子发表看法，评论功能让交流和互动更加方便，用户提交评论后，前端把内容传给后端，后端负责将评论和对应的帖子、用户关联起来，再存入数据库。

用户能轻松建立联系，关注功能起了大作用，用户可以盯住自己感兴趣的其他用户，这样就能马上看到他们发的内容，后端管着用户之间的关注关系数据，让关注功能可以正常用。

群聊功能用了 WebSocket 技术，让用户能实时聊天，前端连接 WebSocket，后端处理连接请求和消息收发，保证群聊消息及时准确传给每个成员，给用户方便交流的环境。

4.2.3 活动管理模块



图 4.3 活动管理用户界面

认证机构和个人能通过活动发布功能找到一个平台，用来发布流浪动物救助活动，发布者要在前端填上活动的详细情况，像主题、内容、时间、地点和参与条件之类的，还能上传图片或者文件来帮忙说明活动。前端把这些信息传给后端，后端负责验证活动信息并存储起来，保证信息完整又准确。

志愿者参与活动变得更简单了，因为有了在线报名功能，他们在前端查看活动详情后，点击报名按钮提交申请，接着，前端把报名信息传给后端，后端处理并保存这些信息，再更新活动的报名状态，这样一来，发布者和志愿者都能马上知道活动的报名情况。

4.2.4 后台管理模块



图 4.4 后台管理界面

管理员通过后台管理模块可以全面管理平台，用户管理功能让管理员查看所有用户的信息，像基本资料、账号状态和权限等内容，管理员还能冻结、解冻用户账号，调整用户权限，以此来维护平台的用户秩序。

管理员能用动物管理功能处理流浪动物的信息，像添加、修改和删除一样，保证信息准确又及时。比如，新救助的流浪动物出现时，管理员能在后台加上详细信息，动物的健康状况或者其他信息变了，管理员马上就能修改。

管理员能通过帖子管理功能来处理社区论坛的帖子，可以看所有帖子的列表，删掉或者屏蔽违规的帖子，这么做是为了让社区内容合法又规范，给大家一个健康积极的交流地方。

4.3 系统部署方案

系统部署时，先挑性能稳、能按需调整配置的云服务器，按照平台预计的用户同时访问量和数据存储需求，选 4 核 16G 的服务器，保证系统运行时能平稳高效处理各类请求。

使用容器化技术，把每个微服务用 Docker 打包成独立的镜像，镜像里包含所有需要的依赖和配置。接着把镜像放到 Kubernetes 集群中运行，Kubernetes 负责调度容器、分配资源以及发现服务，这样一来，系统实现了自动部署和管理，扩展性和容错能力也得到了提升。

Nginx 是前端服务器，负责托管静态资源和转发请求，配置 Nginx 反向代理规则后，用户请求能被精准转发到对应的微服务，同时，使用 Nginx 的负载均衡功能分配请求流量，提高系统并发处理能力，保证平台稳定服务大量用户。

5 流浪动物救助平台的测试

流浪动物救助平台开发结束后，要保证系统稳定高效运行，满足用户需求，就必须进行全面测试，测试包括功能测试、系统测试等多个部分，目的是找出并解决系统中的问题，提高质量和可靠性。

5.1 功能测试

功能测试重点放在系统的各个功能模块上，保证每个功能正常运行，符合预期，满足用户实际需求。

5.1.1 社区论坛功能

社区论坛是用户交流互动的重要地方，功能的稳定和可用性非常关键，测试时，模拟了多种用户场景，比如普通访客、注册用户、实名认证用户的不同操作，发帖功能方面，测试了图文、视频等不同格式帖子的发布，查看帖子内容能否正确保存和展示，图片与视频是否能正常加载。评论功能则检查了评论的提交、显示以及和帖子的关联是否准确，点赞功能也进行了测试，确保点赞数量的统计和显示无误，多次点赞或取消点赞不会出现数据问题。这些测试保证了各功能模块间数据交互的准确性与响应速度，让社区论坛功能可以正常使用。

5.1.2 用户信息功能

用户信息管理功能包含修改个人信息和设置账户安全等关键操作，测试时，主要查看数据加密传输和隐私保护是否到位，用抓包工具分析传输过程，看敏感信息像密码、身份证号之类，在网络传输时有没有用加密算法，比如 HTTPS 协议来保护，避免被窃取。在账户安全设置方面，测试了修改密码、找回密码这些功能，看密码强度校验、验证码的发送与验证是否正常。另外，也检查了隐私设置功能，保证用户能按自己需求调整个人信息的可见范围，保护隐私安全。

5.1.3 动物领养申请功能

流浪动物的领养效率和用户体验，直接受到动物领养申请流程的影响，测试这一环节时，仔细查看了从提交应用到审核完成的每个步骤，模拟用户提交申请时，填写了各种真实数据和边界值，像是不同的养宠经验、家庭环境等，看系统能不能正确验证和处理这些信息。还跟踪了审核流程，看管理员能不能及时收到申请并审核，审核结果是不是能准确、快速通知到申请人。另外，也测试了申请时的异常情况，像网络断了、重复提交申请之类的，保证系统在各种状况下都能让领养申请流程顺利进行。

5.1.4 聊天功能

聊天功能可以增强用户粘性，它的易用性和可靠性直接关系到用户的交流体验，测试时模拟了多种聊天场景，像一对一聊天、群聊这些都包括在内，消息发送和接收的实时性方面，在 4G、WiFi 等不同网络环境下发送消息，看消息送达时间是否符合用户要求。测试群聊功能的时候，查看了群成员加入、退出，群消息广播以及群公告显示等功能是否齐全。还检查了聊天记录的保存状况，保证聊天记录能完整准确地存储，用户用不同设备登录也能正常查看历史聊天记录。

5.1.5 后台管理功能

后台管理系统集中审核和统计用户、动物、帖子等核心数据，测试时，重点关注管理任务的操作是否方便，数据查询是否精准，以及权限控制是否有效，在用户管理上，管理员冻结、解冻账号或调整权限时，操作要简单快速，结果还需立刻同步到系统里。动物管理方面，管理员添加、修改或删除动物信息时，必须保证准确无误，更新的内容也要马上显示在前端页面。帖子管理中，管理员识别、删除或屏蔽违规帖子时，操作流程需要顺畅，同时帖子的数量、浏览量等统计数据也得确保正确。另外，权限控制机制经过严格测试，确保各级管理员只能执行被允许的任务，避免越权访问引发的安全隐患。

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

系统性能好不好，响应时间很关键，它直接决定了用户用起来顺不顺手，我们做了个测试，看看系统在普通使用和高负载时的表现，用 JMeter 这样的专业工具，模拟了从几个用户

慢慢加到 200 个用户同时在线的情况，这些用户会做各种操作，比如查看动物资料、发帖子或者提交领养申请。结果发现，平时用的时候，系统反应很快，平均只要 0.2 秒，几乎感觉不到卡顿，就算用户多到 200 个同时在线，响应时间也没超过 1 秒，完全符合性能要求，说明系统不管人多人少都能迅速处理请求。

5.2.2 吞吐量测试



图 5.2 吞吐量测试结果

系统在单位时间内能处理的最大请求数量就是吞吐量，这展现了系统的处理能力与扩展空间。实际应用场景中预计并发量较低，所以测试时把最大并发用户数定为 100 人，利用性能测试工具模拟 100 个并发用户不断向系统发出各类请求，像浏览页面、提交表单之类的，再统计系统每分钟处理的有效请求数量。测试结果显示，在这种情况下，系统每分钟可以处理大概 4000 次有效请求，这表明系统处理能力不错，能满足预期的业务需求，以后业务量增加的话，系统也有一定的扩展空间。

5.2.3 易用性测试

系统是否好用，易用性是个重要标准，这会直接影响用户接不接受这个系统，为了评估易用性，我们找来了不同类型的用户，像年龄不一样、技术水平有差别、使用目的也不同的用户，收集了他们的反馈。我们用了问卷调查和用户访谈这些办法，去搞清楚用户在用系统时的感受，结果发现，大家普遍觉得系统界面设计得很合理，导航也很清晰，操作逻辑容易懂，哪怕是没有用过的用户，简单了解一下功能后，也能很快上手做一些基本操作，比如浏览帖子或者查找动物信息之类的。这就说明系统在易用性上做得不错，能满足各类用户的需要。

5.2.4 安全性测试

保障用户数据安全和系统稳定运行，关键在于系统的安全性，为此，进行了一系列的安全测试，包括模拟 SQL 注入攻击、测试 XSS 跨站脚本攻击等，在 SQL 注入攻击模拟中，用专门的工具向系统输入可能引发攻击的恶意语句，查看系统能否识别并防范这些攻击，确保数据库安全。在 XSS 跨站脚本攻击测试里，在用户输入框（例如发帖或评论内容）中输入恶意脚本代码，检查系统是否能过滤和转义输入内容，防止恶意脚本在用户浏览器中运行，保护用户隐私和系统安全。经过全面测试，没有发现明显的安全漏洞，这说明系统有较强的安全防护能力。

5.3 测试结论

5.3.1 测试结果

软件系统经过多轮严格的功能和系统测试，能够稳定运行，测试结果的各项指标大多满足需求规格说明书的要求，社区论坛、用户信息管理、动物领养申请、聊天功能以及后台管理等功能模块都能正常使用，没有发现严重缺陷或异常问题。

5.3.2 性能

软件系统性能很强，反应速度快，平时用的时候，平均反应时间是 0.2 秒，就算负载高了，也不会超过 1 秒，在很多任务一起进行时，系统承受能力依旧不错，能达到性能上的严格标准，用户使用系统时，不管看信息还是做复杂操作，都不会被系统慢吞吞的反应影响，体验很流畅。

5.3.3 易用性

用户认可了软件系统的易用性，界面好看，操作逻辑也简单明了，不管是哪个知识层次或技术水平的用户，只要熟悉了基本操作，都能顺利使用这个软件，这样可以吸引更多人加入流浪动物救助平台，让平台使用率和用户粘性都得到提升。

5.3.4 安全性

模拟攻击测试里，系统表现出了强大的安全防护能力，它成功挡住了 SQL 注入攻击和 XSS 跨站脚本攻击这些常见的安全威胁，没发现重大安全漏洞，用户的数据安全和系统的稳定运行有了可靠保障，用户可以安心在平台上操作。

5.3.5 可用性

软件系统可用性很高，每次运行基本都能得到预期结果，崩溃率也很低，测试时，系统一直很稳定，没有频繁崩溃或者异常退出的问题，软件运行时不会影响本地其他数据，安全

性与可靠性都有保障，给用户提供了稳定可靠的使用环境。

流浪动物救助平台已经过全面测试，可以正式上线运行，使用过程中要持续关注系统运行状况，及时解决可能出现的问题，根据用户反馈和业务需求，不断优化和升级系统。

6 总结与期望

6.1 项目总结

这个项目研究了流浪动物救助领域的难题^{[21], [22]}，利用现代信息技术，打造了一个综合救助平台，平台主要解决信息不对称和救助流程复杂的问题，提升救助效率，推动公益事业发展，我们用领域驱动设计（DDD）方法构建核心业务逻辑，保证软件开发与业务一致，同时，采用微服务架构和消息队列技术，确保系统稳定且易于扩展，为了保护用户数据安全，引入了 JWT 令牌机制，平台功能丰富，包括信息发布、互动交流、动物管理等，覆盖从发现流浪动物到完成领养的全过程。另外，为志愿者和救助组织提供了明确的参与方式，吸引更多人加入这项有意义的工作，平台既是一次技术创新，也是探索服务社会公共利益的新方法。

6.2 测试结论

未来，我们希望这个流浪动物救助平台能成为爱心人士和需要帮助的动物之间的纽带，不只是现在的功能和服务范围，技术不断进步，社会需求也在变化，我们会持续升级改进，先从用户体验说起，界面设计会不断优化，变得更友好、更好用，还会增加多语言支持，让全球用户都能使用。再看救助效率，打算引入人工智能技术，比如用图像识别分析上传的照片，快速判断动物种类和健康状况，还用自然语言处理技术自动审核帖子内容，减少人工操作，加快信息传播。另外，我们想跟更多非政府组织、慈善机构甚至政府部门合作，一起推动相关政策的制定和完善，给流浪动物营造更友善的生活环境。

参考文献

- [1] 宁小庚, 黄晓芳. 一种基于领域驱动设计划分微服务的方法[J]. 西南科技大学学报, 2019, 34(01):80-85.
- [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, 顾峻琯, 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] 王忠, 程磊. 基于领域驱动设计的软件开发[J]. 软件导刊, 2008, (02):37-39.
- [17] 李引, 袁峰. 基于领域驱动设计的应用系统模型[J]. 计算机工程与应用, 2013, 49(16):1-8.
- [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.