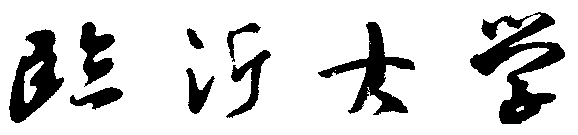
分 类 号：TP311

2023届

单位代码：10452



毕业论文（设计）

基于SpringBoot和Vue的流浪动物救助系统

姓 名 张天赐

201909000242

学 号

2019级

年 级

软件工程

专 业

信息科学与工程学院

学 院

韩丽花

指导教师

2023年4月10日

基于SpringBoot和Vue的流浪动物救助系统

摘 要

现代社会的发展迅速，社会生产力和效率不断提高，人们对于信息科学化管理的认知早已提升到了很高的层次，信息管理的重要性已经深入人心，科学化信息管理能够使信息的存取更加快捷、完善，并且极大的提高工作效率。

论文主要是对流浪动物救助系统进行介绍，内容阐述了研究背景及意义、研究现状和系统设计，对系统的使用前景和市场价值有比较清晰的解释，最后还包括了对流浪动物救助系统的一些测试。

本系统是以Spring Boot和Vue为开发技术。系统的主要使用者分别为救助站工作人员、管理员、用户和成为志愿者的用户，针对使用者的不同细分为管理员端和用户端。管理员端功能包括：动物管理、用户管理、志愿者管理、救助站管理、领养申请处理、论坛管理、紧急救助等功能；用户端功能包括查看社区、查看小动物们、在社区发帖评论、发布紧急救助信息、修改用户个人资料、领养小动物、申请成为志愿者、查看消息等。通过这些功能模块设计，基本实现了系统功能。

具体在系统设计上，后端采用了Spring Boot框架；数据库使用MySQL；前端使用Vue框架。是一个完善且优秀的流浪动物救助系统。

关键词：Spring Boot；MySQL；Vue；流浪动物救助

**STRAY ANIMAL RESCUE SYSTEM BASED ON SPRINGBOOT AND VUE**

**ABSTRACT**

With the rapid development of modern society and the continuous improvement of social productivity and efficiency, people's cognition of scientific management of information has been promoted to a very high level, and the importance of information management has been deeply rooted in the hearts of the people. Scientific information management can make information access faster and more perfect, and greatly improve work efficiency.

This paper mainly introduces the stray animal rescue system, expounds the research background and significance, research status and system design, and has a clear explanation of the use prospect and market value of the system. Finally, it also includes some tests of the stray animal rescue system.

This system is developed with Spring Boot and Vue. The main users of the system are rescue station staff, administrators, users and users who become volunteers, and the different subdivisions of users are administrators and users. The functions of the administrator include: animal management, user management, volunteer management, rescue station management, adoption application processing, forum management, emergency rescue and other functions. User functions include viewing the community, viewing animals, Posting comments in the community, releasing emergency rescue information, modifying user profile, adopting small animals, applying to be a volunteer, viewing messages, and so on. Through the design of these functional modules, the system functions are basically realized.

Specifically, in the system design, the back-end uses the Spring Boot framework. The database is MySQL. The frontend uses Vue framework. It is a perfect and excellent rescue system for stray animals.

**KEY WORDS:** spring boot,mysql,vue,stray animal rescue

页脚居中插入页码，页码为罗马数字编码，Times New Roman小五号字，居中

目 录

[前言 1](#_Toc8956)

[1绪论 2](#_Toc20112)

[1.1研究背景及意义 2](#_Toc7948)

[1.2国内外研究现状 2](#_Toc21795)

[1.3论文结构 2](#_Toc6312)

[2系统分析 4](#_Toc18280)

[2.1可行性分析 4](#_Toc3883)

[2.1.1经济可行性 4](#_Toc15370)

[2.1.2操作可行性 4](#_Toc26722)

[2.1.3技术可行性 4](#_Toc19501)

[2.1.4法律可行性 4](#_Toc8518)

[2.2数据流程分析 5](#_Toc12220)

[2.2.1数据增加流程 5](#_Toc28038)

[2.2.2数据删除流程 5](#_Toc345)

[2.2.3数据修改流程 6](#_Toc1084)

[2.3系统功能分析 7](#_Toc3753)

[2.3.1功能性分析 7](#_Toc2959)

[2.3.2非功能性分析 8](#_Toc10)

[2.4系统用例分析 8](#_Toc25094)

[2.5本章小结 10](#_Toc15096)

[3系统设计 11](#_Toc15324)

[3.1系统架构设计 11](#_Toc15538)

[3.2系统功能模块设计 11](#_Toc31642)

[3.2.1整体功能模块设计 12](#_Toc22231)

[3.2.2小动物模块设计 12](#_Toc10421)

[3.2.3志愿者管理模块设计 13](#_Toc17936)

[3.2.4论坛模块设计 15](#_Toc15375)

[3.2.5领养模块设计 16](#_Toc6927)

[3.2.6紧急救助模块设计 17](#_Toc31113)

[3.2.7用户信息管理模块设计 19](#_Toc28767)

[3.2.8救助站管理模块设计 20](#_Toc1793)

[3.2.9图片上传模块设计 21](#_Toc31707)

[3.3数据库设计 22](#_Toc13433)

[3.3.1数据库概念结构设计 22](#_Toc16589)

[3.3.2数据库逻辑结构设计 24](#_Toc17975)

[3.4本章总结 28](#_Toc4090)

[4系统实现 29](#_Toc22811)

[4.1用户模块实现 29](#_Toc11566)

[4.1.1登录注册模块实现 29](#_Toc31363)

[4.1.2小动物模块实现 31](#_Toc28163)

[4.1.3志愿者模块实现 32](#_Toc6263)

[4.1.4论坛模块实现 33](#_Toc1010)

[4.1.7用户信息模块实现 35](#_Toc21482)

[4.2管理员模块实现 36](#_Toc29689)

[4.2.1领养信息管理实现 36](#_Toc20054)

[4.2.2志愿者模块实现 37](#_Toc4180)

[4.2.3小动物模块实现 41](#_Toc5381)

[4.2.4紧急救助模块实现 44](#_Toc11551)

[4.3图片上传模块实现 44](#_Toc8125)

[5系统测试 46](#_Toc20181)

[5.1登录注册模块功能测试 46](#_Toc17831)

[5.2小动物模块功能测试 47](#_Toc18957)

[5.3志愿者模块功能测试 48](#_Toc25268)

[5.4论坛模块功能测试 50](#_Toc17897)

[5.5用户信息管理模块功能测试 51](#_Toc30913)

[5.6领养信息管理模块功能测试 51](#_Toc24324)

[5.7紧急救助模块功能测试 52](#_Toc13131)

[6总结 53](#_Toc22256)

[参考文献 54](#_Toc10626)

[致谢 55](#_Toc23679)

# 前言

本论文旨在对系统进行全面的分析、设计和实现，并通过部分测试结果来验证系统功能的有效性。具体结构如下：

第一章：绪论。本章节将介绍研究背景和选题的意义，同时探讨当前相关领域的研究现状，以及本研究的目标和研究方法。

第二章：需求分析与功能设计。本章将从用户和开发者的角度出发，对系统的需求进行分析，并通过功能等方面进行设计。此外，经济方面的分析也将被纳入考虑范围。

第三章：系统架构设计。本章将详细阐述系统的架构设计，包括系统框架结构、功能模块、数据库设计等。

第四章：系统实现与优化。本章将介绍系统的实现过程，包括框架搭建、界面设计等。同时，对系统进行优化，以提高系统的性能和用户体验。

第五章：系统测试与结果分析。本章将介绍系统的测试方法和测试结果，分析测试数据并得出相应结论。

第六章：总结与展望。本章将对本次毕业设计进行总结，并展望未来可能的研究方向和改进方案。

总之，本论文将全面、系统地介绍本研究的过程和成果，为读者提供有关系统设计与实现的详尽信息。

# 1绪论

## 1.1研究背景及意义

随着这几年疫情带来的冲击，很多流浪动物都失去了往日的食物来源。这对这些动物来说无疑是一场灾难。动物救助系统就可以起到极大的促进作用，既可以帮助救助站的工作人员更加轻松的管理，又可以号召社区爱心人士共同帮助小动物。救助站工作人员可以登陆该系统管理了解当前社区流浪动物分布区域及救助情况，通过该系统规划救助工作。社区爱心人士可以通过该系统了解工作人员的工作详情，同时可以领养动物或利用商城模块为小动物们购买必需品。流浪的小动物也是城市的一员。帮助和救助处于困难的小动物们也是我们的责任。希望可以通过该系统能够使得更多的小动物得以生存。

## 1.2国内外研究现状

随着城市化的发展，流浪动物已经成为城市里的一个庞大的群体，对救助流浪动物的管理也应该更加规范认真,然而传统的管理主要靠人工进行，每天可能都会有大量的数据需要进行处理，从而增加了管理人员消耗的精力和成本，很大程度影响了管理信息的有效性，因此对于城市流浪动物信息化管理，已经成为了必然不可忽视问题。

伴随着“互联网+”的概念的提出，计算机在各大领域都得到了更加广泛的应用，它已经成为了人类脑力劳动者不可缺少的工具，使用这些信息化管理软件，不仅可以给使我们得工作更加高效，而且给我们提供了极大的方便。同时计算机操作相比手工管理有着无法比拟的优点，比如：查询检索迅速、方便，可靠性高，存储量大、保密性高、使用时间长、成本较低等。

综上所述，开发一个适合管理流浪动物的系统，来解决救助流浪动物中遇到的问题，具有相当重要的现实意义。

1.3论文结构

本篇论文的目的是全面地分析、设计和实现一个系统，并且通过部分测试结果来证明该系统的有效性。论文将提供有关系统设计和实现的详尽信息

首先，论文会介绍研究的背景和选题的意义，以及探讨当前相关领域的研究现状。其次，论文将从用户和开发者的角度出发，对系统的需求进行分析，并进行功能等方面的设计。此外，还将包括经济方面的分析。

然后，论文将详细阐述系统的架构设计，包括系统框架结构、功能模块和数据库设计等。接下来，将介绍系统的实现过程，包括框架搭建、界面设计等。同时，将对系统进行优化，以提高系统的性能和用户体验。

最后，论文将介绍系统的测试方法和测试结果，并对测试数据进行分析，得出相应的结论。最后一章将对本次毕业设计进行总结，并展望未来可能的研究方向和改进方案。

总之，本论文将全面、系统地介绍本研究的过程和成果，为读者提供有关系统设计和实现的详尽信息。

# 2系统分析

系统分析是开发系统前必须的重要过程，通过分析能够更进一步了解用户的真实需求和开发重难点，完善明了的分析结果能够极大地提高开发的效率。分析的一般内容为可行性分析、需求分析等，可行性分析通常包括经济可行性、操作可行性等；随后就是对项目的具体需求进行分析，分析时一般会借助图例的方式。具体内容请看下文。

## 2.1可行性分析

### 2.1.1经济可行性

经济可行性分析的问题为开发成本和运营成本。该系统采用了论坛的模式，在该模式运营下容易积攒大量的粘性用户，当用户数量持续上升时，亦可通过广告等方式盈利。此时运营成本便被填补，开发成本亦可逐渐收回。

因此经济可行性没有问题。

### 2.1.2操作可行性

该系统的操作界面参考了数个在该模式下开发的案例，汲取众家之所长，凸显简介明了的操作方式，只要是具备基本计算机操作的用户都能快速上手。

因此操作可行性没有问题。

### 2.1.3技术可行性

技术可行性的问题主要是所采用框架的可用性，当有更加优秀的技术出现时系统的更新性，以及开发系统的进度推进效率。

针对这些问题，主要的解释如下：

（1）现有的Spring boot和Vue都是公认的主流框架，市面上的大小公司都有使用。

（2）Spring Boot框架是用于简化Spring应用的产品，Spring又是为了简化开发的复杂性而诞生的，新出现的技术都是为了更加简化开发过程而诞生的，底层的结构是不会受到影响的。

（3）采用框架开发的意义就在于简化开发过程，效率只会提高。

综上所述，技术可行性也没有问题。

### 2.1.4法律可行性

从开发者的角度考虑，该系统使用的框架均为网上开源免费的，素材均为作者个人制作，知识产权方面不会造成侵权。

从用户的角度考虑，只要同意系统提供的使用协议，不从事传播违禁信息等违法活动就能避免。

因此法律可行性没有问题。

## 2.2数据流程分析

### 2.2.1数据增加流程

用户在使用注册账号和发帖等功能时会执行数据添加的操作，添加数据时系统会生成一个唯一的ID，用户无法决定该ID的内容，其余可由用户决定的信息交由用户自行填写。数据填写完毕后会先经过系统的校验，验证合法时才能够成功添加，否则将取消执行添加操作，驳回用户修改后在执行以上操作，直到添加成功或操作取消。具体流程如图2-1显示。

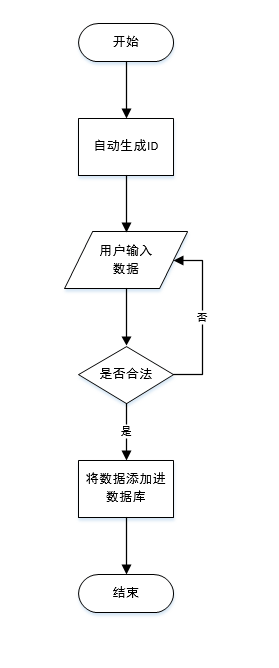


图2-1 数据添加流程图

Fig. 2-1 Data addition flow chart

### 2.2.2数据删除流程

用户在删除自己发过的帖子或系统存在一些没有必要的数据时，可以执行删除的相关操作。在选择了选择了想要删除的数据时，需要先通过数据ID到数据库中查找相应记录，当数据存在时方可删除，否侧删除失败。图2-2就是数据删除时的流程图。

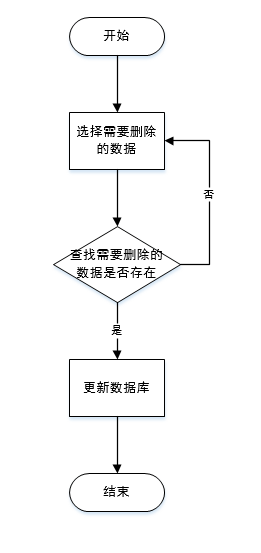


图2-2 数据删除流程图

Fig. 2-2 Data deletion flow chart

### 2.2.3数据修改流程

在执行数据修改的操作时流程和删除时相似，如图2-3所示。

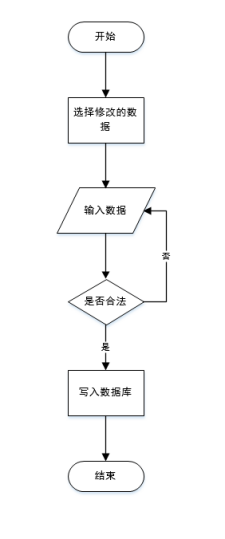


图2-3 数据修改流程图

Fig. 2-3 Data modification flow chart

## 2.3系统功能分析

### 2.3.1功能性分析

根据本系统的角色划分，可以分为管理员模块和用户模块。具体功能如下所示。

用户模块：

（1）登录、注册：用户使用数据库中有记录的账号密码登录方可进入个人空间，若是没有账号也可进行注册来为自己创建一个账号。

（2）主页展示：展示封面和一些公告。

（3）小动物们：用户进行小动物信息阅览。发现喜欢的小动物后可以加入我喜欢和评论。

（4）成为志愿者：用户可以在该模块填写信息成为志愿者。成为志愿者后可以在该模块报名参与到日常的救助活动中，并且获得志愿者称号，该称号可以随着志愿活动的参加次数增加而升级。在论坛将会显示该称号。

（5）领养我：用户可以在该模块领养小动物。用户选择想领养的小动物后填写相关的信息可以申请领养。可以在该模块查看领养进度和结果。成功领养小动物将会获得对应称号。在论坛将会显示该称号。

（6）论坛：用户可以在该模块发帖讨论，探讨养宠物的经验。题主可以进行置顶评论的操作。用户可以点击其他用户头像查看该用户的个人空间的救助统计。用户可以在搜索框中模糊查询想要寻找的主题帖。

（7）联系我们：用户可以在该模块联系救助站，发布发现的流浪动物线索。成功救助后可以获得救助者称号。在论坛将会显示该称号。

（8）个人空间：用户可以在该模块填写和修改自己的个人信息，如名称、头像、密码等。可以在该模块的我喜欢中查看自己加入喜欢的小动物。可以在该模块的“我的救助统计”查看自己的志愿活动、救助活动记录、领养记录以及各种称号。

管理员模块：

（1）公告管理：对用户端展示的主页进行增删改查。

（2）用户管理：既对救助站的员工和前端注册的用户进行管理。员工一般分为系统超级管理员，管理员，普通员工，主要用于对员工的增删改查。对于前端注册的用户的管理，主要用于用户的权限管理，如禁止发帖。

（3）动物信息管理：管理小动物的基本信息。记录小动物的救助日期，种类，体重，疫苗接种情况，年龄等基本信息。对这些信息实现增删改查以及分页。

（4）志愿者管理：主要对志愿者进行管理。处理前端用户发布的申请信息。对已有的志愿者实现删改查。发布救助活动来召集志愿者参加。完成活动后为志愿者的志愿活动记录自动添加本次记录。

（5）领养管理：用于管理可领养动物和处理用户的领养请求。对可领养动物进行增删改查。对用户的领养请求进行回复。

（6）救助站管理：对救助站信息进行增删改查。

（7）论坛管理：对论坛的帖子可以进行删除、置顶等操作，对发帖的用户也可以进行禁言等操作。

（8）紧急救助：处理前端用户所发布的流浪动物救助信息。点击收到既开始工作，完成工作后选择已完成反馈给用户。

### 2.3.2非功能性分析

本系统的非功能性需求具体如下：

**表2-1 流浪动物救助系统非功能需求表**

**Tab. 2-1 Table of non-functional requirements for stray animal rescue systems**

|  |  |
| --- | --- |
| 安全性 | 对于数据库的安装和使用应当注意，数据库密码使用应当合乎规范。 |
| 可靠性 | 对于用户的操作理应能够成功执行，成功率不应小于90%。 |
| 性能 | 性能好就有潜力扩大市场份额，应当使任意一部主流设备都能流畅运行。 |
| 可扩展性 | 多预留几个接口就能提高可扩展性。 |
| 易用性 | 应当使用户看到界面就能快速上手使用。 |
| 可维护性 | 使用高级语言开发，工具使用IDEA等现金开发工具，有明确的开发目标，做到以上几点就能大大提高可维护性。 |

## 2.4系统用例分析

通过2.3的功能分析，得到了本系统的用例图：

用户角色的用例图如2-4所示。

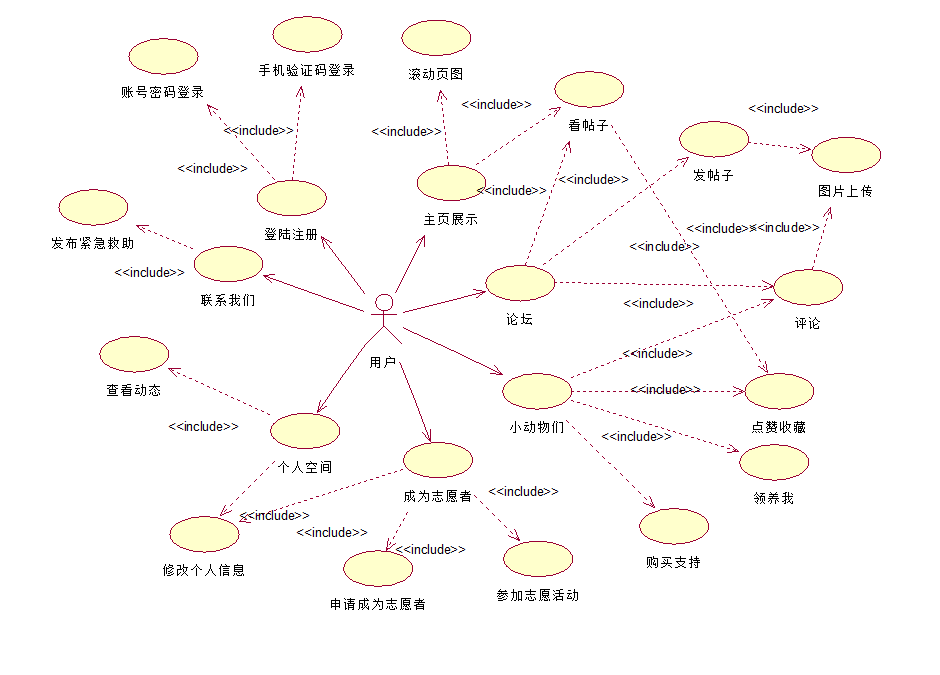


图2-4 流浪动物救助系统用户角色用例图

Fig. 2-4 User role use case diagram of the stray animal rescue system

管理员角色的用例图如2-5所示。

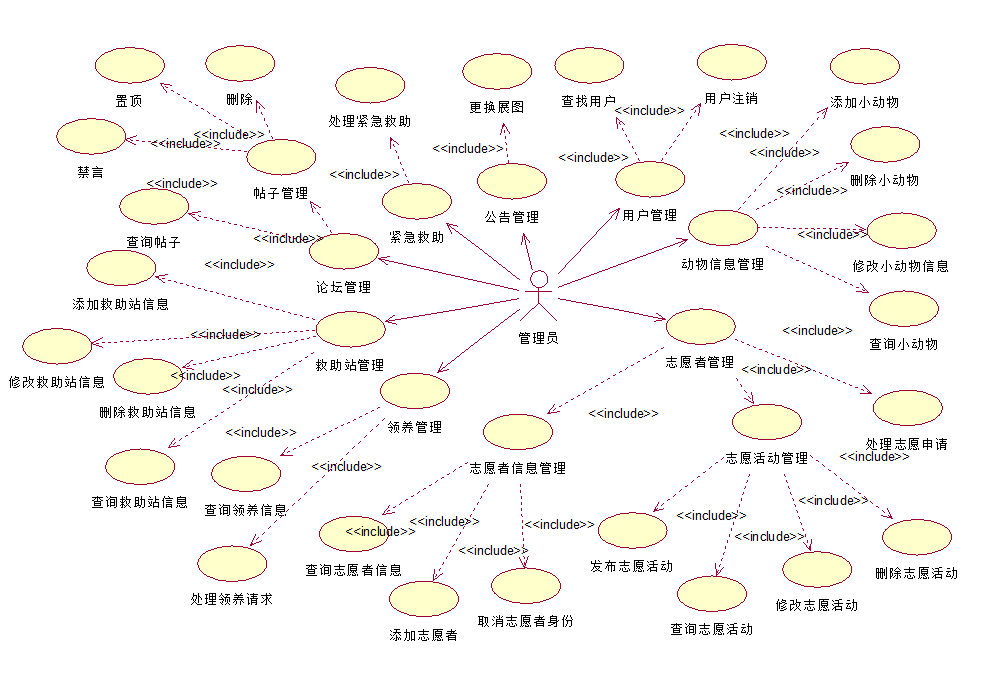


图2-5 流浪动物救助系统管理员角色用例图

Fig. 2-5 Stray animal rescue system administrator role use case diagram

## 2.5本章小结

本章主要通过对流浪动物救助系统的可行性分析、流程分析、功能需求分析、系统用例分析，确定整个流浪动物救助系统要实现的功能。同时也为流浪动物救助系统的代码实现和测试提供了标准。

3系统设计

本章主要介绍的是流浪动物救助系统的系统架构设计、系统功能模块、数据库设计。

3.1系统架构设计

本系统从架构上分为三部分：表现层（UI）、业务逻辑层（BLL）、数据层（DL），如图3-1所示。

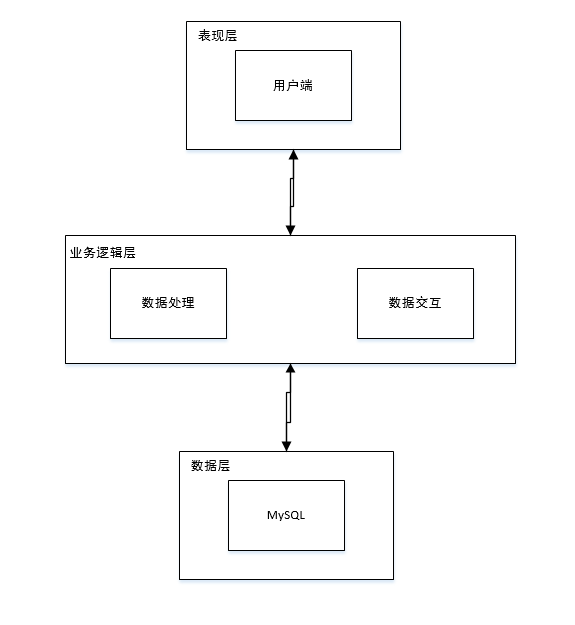


图3-1 流浪动物救助系统系统架构图

Fig. 3-1 Stray animal rescue system system architecture diagram

表现层（UI）：该层主要完成与用户的交互功能。优美简洁的UI设计能够使人眼前一亮的同时还能增加用户好感，起到提升使用体验的效果；其次，对于电脑端与手机端的不同尺寸影响，应当做到相互兼容，不会出现界面错乱，功能丢失等错误；最后，UI界面的设计应当合理合情，使用户在使用过程中能够轻松找到并理解相应模块对应的功能。

业务逻辑层（BLL）：该层主要实现对表现层传来的请求进行处理的功能。用户通过表现层交互进行了一些操作，传输了一些数据和请求，这些数据和请求经业务逻辑层处理后交付数据层；系统从数据层读取的数据也经业务逻辑层进行处理交付于表现层。

数据层（DL）：该层主要完成对数据的存储和管理功能。对于数据的交互功能整合进了业务逻辑层，因此该层只有数据库。

3.2系统功能模块设计

### 3.2.1整体功能模块设计

根据第二章的需求分析可以得到整体功能模块设计的图如3-2所示。

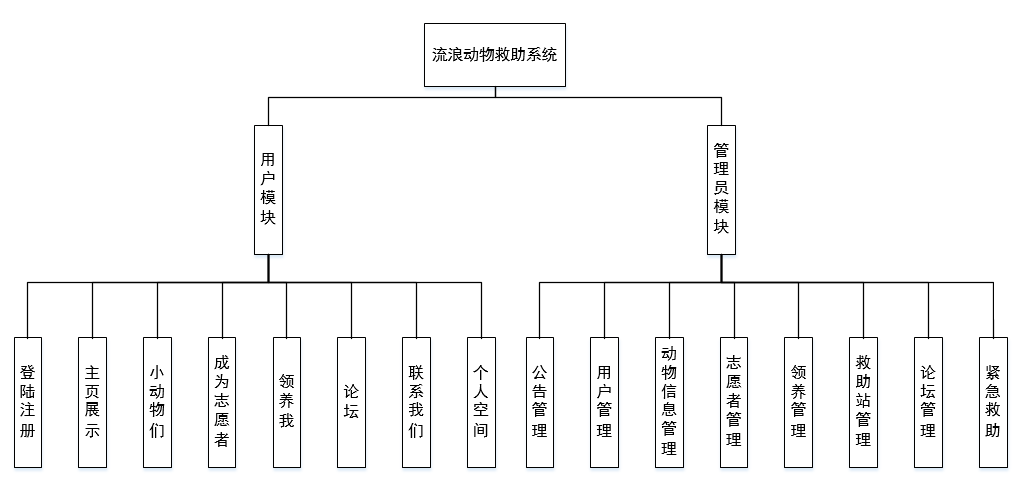


图3-2 流浪动物救助系统整体功能模块设计

Fig. 3-2 The overall functional module design of stray animal rescue system

3.2.2小动物模块设计

小动物模块在该流浪动物救助系统中是一个重要的模块，它的作用是为用户提供了浏览和了解流浪动物信息的渠道。在后端实现上，该模块使用了同一个后台来进行增删改查等操作，但在前端展示上，分为了管理员模块的动物信息管理和用户模块的小动物们，其详细功能介绍如下。

管理员模块是针对小动物的增删改查等操作而存在的一个模块。在该模块中，管理员可以通过简单的操作实现对小动物的管理和维护。

首先，管理员可以在该模块中进行小动物的添加操作。添加小动物时，管理员需要填写相应的信息，例如小动物的种类、性别、年龄、照片等等。在添加完成后，小动物的信息会被存储在系统中，并可以在其他模块中进行查看和管理。

其次，管理员可以在该模块中进行小动物的删除操作。删除小动物时，管理员需要选择要删除的小动物，确认操作后系统会将该小动物的信息从系统中永久性地删除。在删除操作之前，管理员需要仔细考虑删除的后果，以避免误删或不必要的损失。

除了添加和删除操作之外，管理员还可以在该模块中进行小动物信息的编辑和查看操作。编辑操作可以修改小动物的基本信息和照片等内容，而查看操作可以查看小动物的详细信息和当前状态等信息。

总之，动物管理模块简单实现了对小动物的增删改查等基本操作。通过该模块，管理员可以快速地进行小动物管理和维护，以保障小动物的健康和福利。

用户模块提供了一个平台，让用户可以了解到流浪动物的信息。在流浪动物信息页面，用户可以了解到有关流浪动物的各种信息，例如它们的品种、年龄、性别、健康状况等等。此外，该模块还允许用户在流浪动物信息下方留言评论，以分享自己对流浪动物的观察和体验，并与其他用户进行交流和讨论。

如果用户在浏览流浪动物信息时遇到喜欢的动物，他们可以将其添加到自己的收藏夹中，以便下次快速浏览和查看。这样，用户就可以方便地保存自己感兴趣的流浪动物信息，随时回顾和分享给其他人。总之，该模块为用户提供了一个全面、便捷的流浪动物信息平台，让他们可以更好地了解和关心流浪动物，为它们提供帮助。小动物模块的时序图见图3-3。

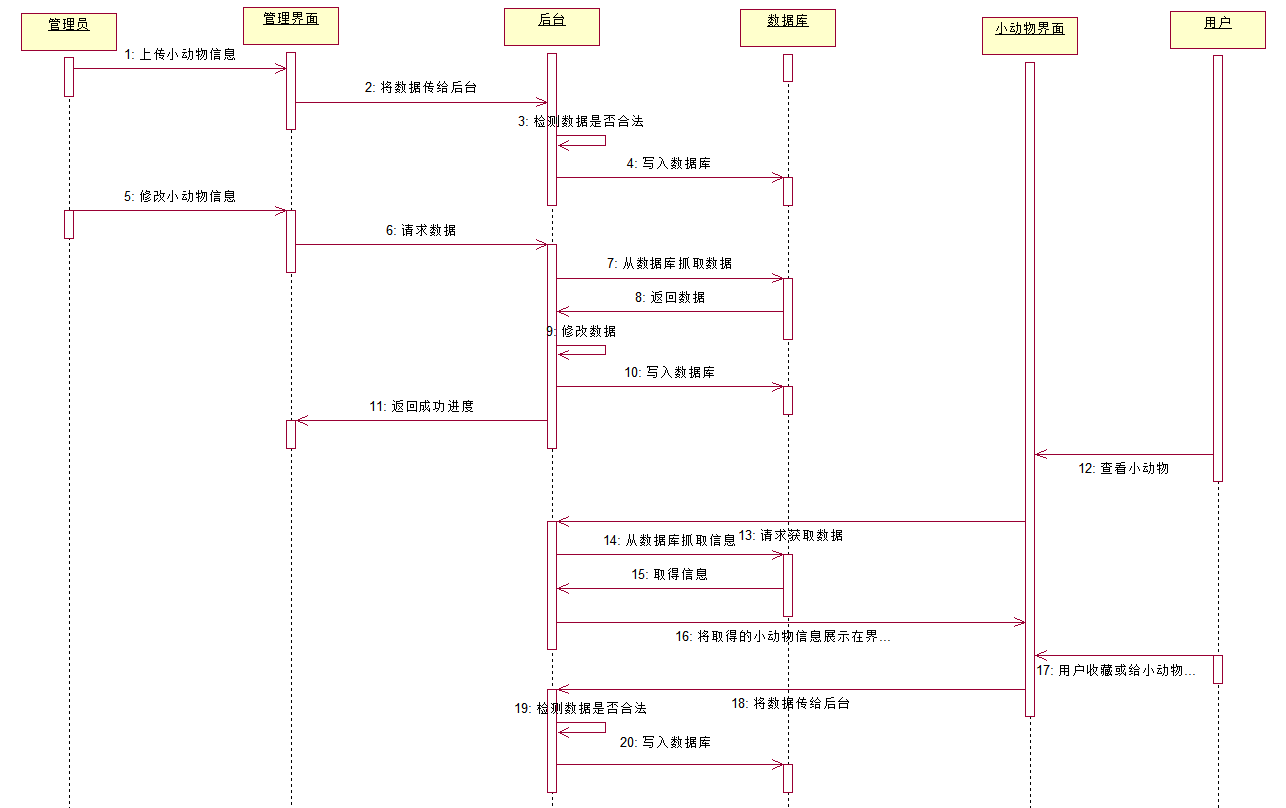


图3-3 小动物们模块时序图

Fig. 3-3 The critters module sequence diagram

3.2.3志愿者管理模块设计

志愿者管理模块的作用是为用户提供了一个浏览和了解流浪动物信息的途径。在后台方面，该模块使用同一个平台来执行添加、删除和修改数据等操作。但在前端方面，它被分成管理员模块的志愿者管理和用户模块的成为志愿者两个部分，分别提供不同的功能。

管理员模块包含三个小模块。首先是处理志愿者申请，用户可以在用户模块申请成为志愿者，而管理员需要通过该模块进行处理，审核申请并决定是否批准。对于审核通过的志愿者，管理员可以将其添加至志愿者名单中。

其次是志愿活动管理，管理员可以在该模块中进行志愿活动的增删改查等操作，包括发布新的志愿活动，修改已有的志愿活动信息，删除不需要的活动等。同时，管理员可以查看所有活动的详细信息，如参加人数、活动时间、地点等，并对已有活动进行管理。

最后是志愿者管理，管理员可以在该模块中进行志愿者的增删查操作。管理员可以浏览所有的志愿者名单，并可以将其进行分类管理，如按照活跃程度、地理位置等因素进行筛选和排序。管理员还可以对志愿者进行评价和奖励，以便更好地激励志愿者的积极性和参与度。

综上所述，该模块为流浪动物救助系统提供了重要的管理工具，可以帮助管理员更好地管理志愿者和志愿活动，并为用户提供更好的服务。

用户模块提供了一个平台，让用户可以申请成为志愿者。用户可以通过填写个人信息并提交申请来表达自己想要为社区或者环境做出贡献的愿望。经过管理员的审核后，用户可以成为正式的志愿者，参加管理员发布的志愿活动，为社区或环境做出积极的贡献。

志愿者可以在该模块中查看管理员发布的志愿活动，并选择自己感兴趣的活动进行报名参加。同时，志愿者还可以通过参加志愿活动来提升自己的志愿者称号，这将是一项令人骄傲的成就。为了更好地管理志愿者，管理员可以在该模块中查看每个志愿者的参加记录和积分等信息。

此外，该模块还提供了基础功能，包括修改个人信息、查看参加记录等功能。志愿者可以根据自己的实际情况随时修改个人信息，以确保志愿者信息的准确性和完整性。同时，查看参加记录可以让志愿者了解自己的志愿活动历史，并对未来的志愿活动做出更好的安排。

总之，该模块为志愿者提供了一个全面、便捷的平台，让他们可以积极参与志愿活动，提升自己的社会责任感和社区意识，为社区和环境做出贡献。同时，管理员也可以通过该模块对志愿者进行管理和监督，确保志愿者活动的质量和效果。时序图具体参照图3-4。

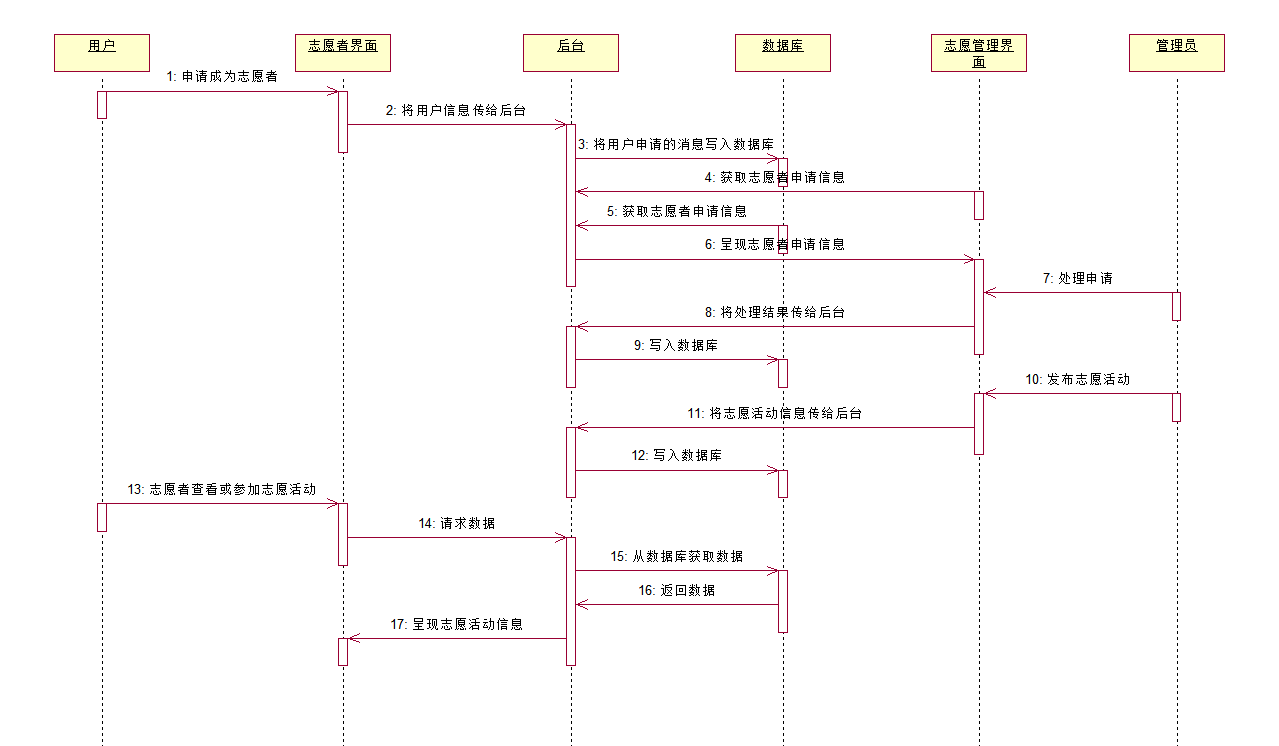


图3-4 志愿者管理模块时序图

Fig. 3-4 Time sequence diagram of volunteer management module

3.2.4论坛模块设计

论坛模块的作用是为用户提供了一个交流和分享信息的途径。在后台方面，该模块使用同一个平台来执行添加、删除和修改数据等操作。但在前端方面，它被分成管理员模块的论坛管理和用户模块的论坛两个部分，分别提供不同的功能。

管理员论坛管理模块主要提供了管理员对论坛的管理和监督功能。管理员可以通过该模块实现对用户发帖内容的审核、删除、编辑、禁言、封号等管理操作，以确保论坛内的言论健康、积极向上，维护平台形象和用户体验。

在该模块中，管理员可以浏览用户发表的所有帖子，对帖子进行分类，如：热帖、置顶、推荐等，使优质帖子得到更多关注。管理员还可以针对不当言论或违规内容进行删除或编辑，确保论坛内容的合法性和健康性。此外，管理员还可以对用户进行禁言、封号等操作，以处置那些违反平台规定、扰乱论坛秩序的用户。

通过对该模块的管理和监督，管理员可以对论坛的运营情况进行全面的掌控和管理，使论坛始终保持良好的发展和运营状态，为用户提供一个积极向上、健康活泼的社区平台。

用户论坛模块提供了一个讨论和交流的平台，让用户可以分享和交流自己的经验、观点和问题，并从其他用户的回复和评论中获得帮助和启发。

为了提高论坛用户之间的互动性，该模块为帖子添加了点赞、评论、转发等功能，让用户可以更方便地表达自己的看法和情感，同时也可以更方便地了解其他用户的意见和建议。这些功能不仅可以增加用户之间的互动性和联系，还可以提高用户的参与度和满意度，让他们更愿意在论坛中分享和交流自己的想法和体验。

在论坛模块中，用户可以浏览和搜索已有的帖子，并对感兴趣的帖子进行点赞、评论、转发等操作。点赞功能可以让用户对帖子的内容表示认可和赞同，同时也可以让其他用户更容易地发现和关注高质量的帖子；评论功能可以让用户对帖子的内容进行详细的解释、补充和讨论，同时也可以让其他用户更全面地了解帖子的内容和观点；转发功能可以让用户将感兴趣的帖子分享给自己的好友和关注者，扩大帖子的影响和传播范围。

总之，论坛模块为用户提供了一个丰富、互动的交流平台，让他们可以更方便地分享和交流自己的经验、观点和问题，并从其他用户的回复和评论中获得帮助和启发。同时，点赞、评论、转发等功能也提高了用户的参与度和满意度，增强了用户对论坛的依赖和使用感官。论坛模块的时序图见图3-5。

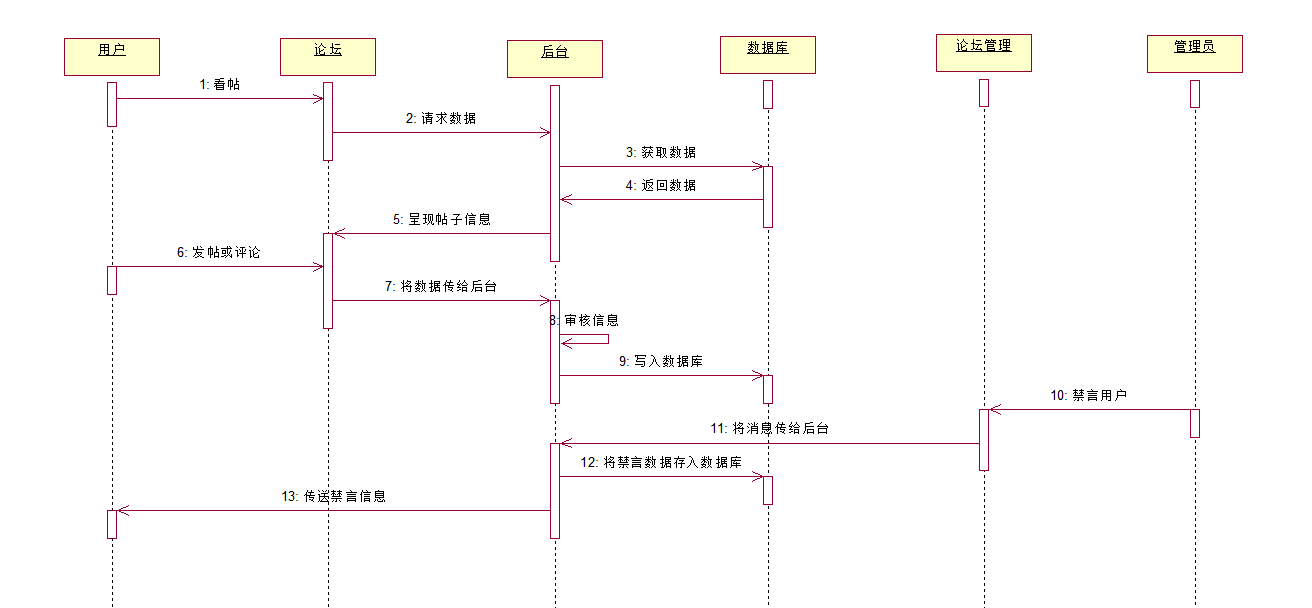


图3-5 论坛模块时序图

Fig. 3-5 Forum module sequence diagram

3.2.5领养模块设计

领养模块的作用是为用户提供了一个领养和了解流浪动物领养信息的途径。在后台方面，该模块使用同一个平台来执行添加、删除和修改数据等操作。但在前端方面，它被分成管理员模块的领养管理和用户模块的领养我两个部分，分别提供不同的功能。

管理员模块是针对用户在流浪动物救助系统中提出的领养请求进行处理的模块。当用户在动物信息模块中发现一只想要领养的流浪动物时，可以在该模块中提交领养申请。申请将进入待审核状态，管理员会对申请进行审核并决定是否同意领养申请。一旦审核通过，用户便可以正式领养该流浪动物。

在领养请求处理过程中，管理员可以对领养申请进行查看、审核、批准或拒绝等操作。管理员可以查看用户的申请信息，包括领养的流浪动物的信息、申请人的个人信息等，以便对申请进行更准确的审核。管理员还可以根据实际情况进行批准或拒绝申请，并向用户发送相关通知信息。

总之，该模块是为了确保流浪动物能够被领养到负责任的家庭而存在的，通过流程化的领养申请和审核流程，保障流浪动物的福利和用户的权益。

用户模块提供了一个平台，让用户可以申请领养小动物。用户可以通过填写领养申请并提交来表达自己想要领养小动物的意愿。管理员将对用户的申请进行审核，确保领养的小动物可以得到合适的照顾和生活环境。

在审核过程中，用户可以随时查看审核状态，以了解审核进度和结果。如果用户的领养申请被批准，他们可以成功领养小动物并开始为它们提供照顾和关爱。同时，如果用户的申请被拒绝，他们可以根据管理员提供的理由进行调整和改进，并重新提交申请。

总之，该模块为用户提供了一个全面、便捷的领养小动物平台，让他们可以为流浪小动物提供一个温暖、安全的家。同时，管理员也可以通过该模块对领养申请进行管理和监督，确保领养小动物的合法性和合适性。领养模块的时序图见图3-6。

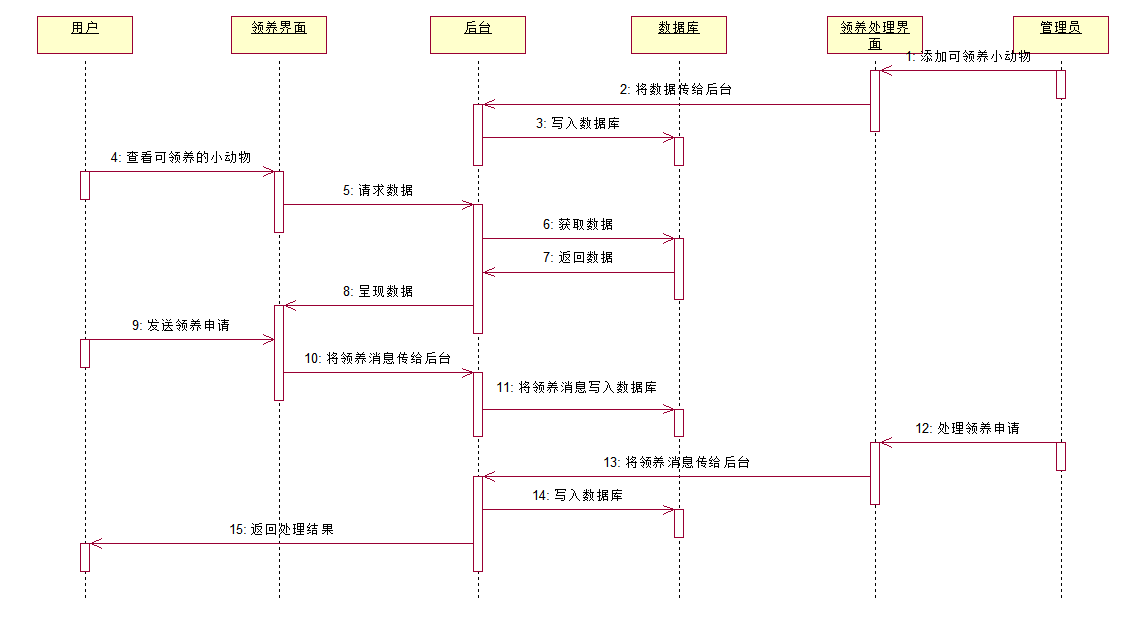


图3-6 领养模块时序图

Fig. 3-6 Adoption module timing diagram

3.2.6紧急救助模块设计

为了提升作为一个救助流浪动物的平台的专业性，让用户对平台更加信任并切身参与到救助流浪动物的行动中，本平台特别设计了紧急救助模块。其作用是为用户提供了一个快速寻求帮助和报告紧急情况的途径。在后台方面，该模块使用同一个平台来执行添加、删除和修改数据等操作。但在前端方面，它被分成管理员模块的紧急救助管理和用户模块的联系我们两个部分，分别提供不同的功能。

管理员模块主要功能是处理用户在用户模块中发布的紧急救助请求。在该模块中，管理员可以浏览所有的紧急救助请求，并根据情况对其进行处理，比如直接提供救助、联系其他志愿者前往提供救助等。同时，在该模块中管理员也可以将紧急救助作为志愿活动发布，吸引更多志愿者参与到紧急救助行动中来，增强救助流浪动物的行动力。

在管理员模块中，管理员需要对所有的紧急救助请求进行详细的审核，并根据实际情况进行处理。紧急救助模块的存在，可以帮助用户在紧急情况下快速寻求帮助，并能够及时得到管理员和其他志愿者的支持和协助，为流浪动物提供更多的救助和保护。

用户模块可以让用户在紧急情况下快速发布救助请求，同时也可以查看自己的救助请求的处理进度。

紧急救助模块为用户提供了一个快速、专业、高效的救助平台，让用户可以在紧急情况下第一时间获得救助和支持。在该模块中，用户可以详细描述流浪动物的情况和需要救助的内容，同时也可以上传照片和视频等证据资料，以便管理员和其他志愿者更好地了解情况和处理请求。一旦救助请求发布成功，管理员和其他志愿者就可以在第一时间查看和处理请求，并尽快提供救助和支持。

通过紧急救助模块，用户可以直接参与到救助流浪动物的行动中，提升自己的社会责任感和参与度。同时，该模块也可以让用户更加信任平台的专业性和可靠性，增强用户对平台的信任和依赖。因此，紧急救助模块不仅是本流浪动物救助系统的重要组成部分，也是提升平台专业性和用户信任的关键环节。紧急救助模块的时序图见图3-7。

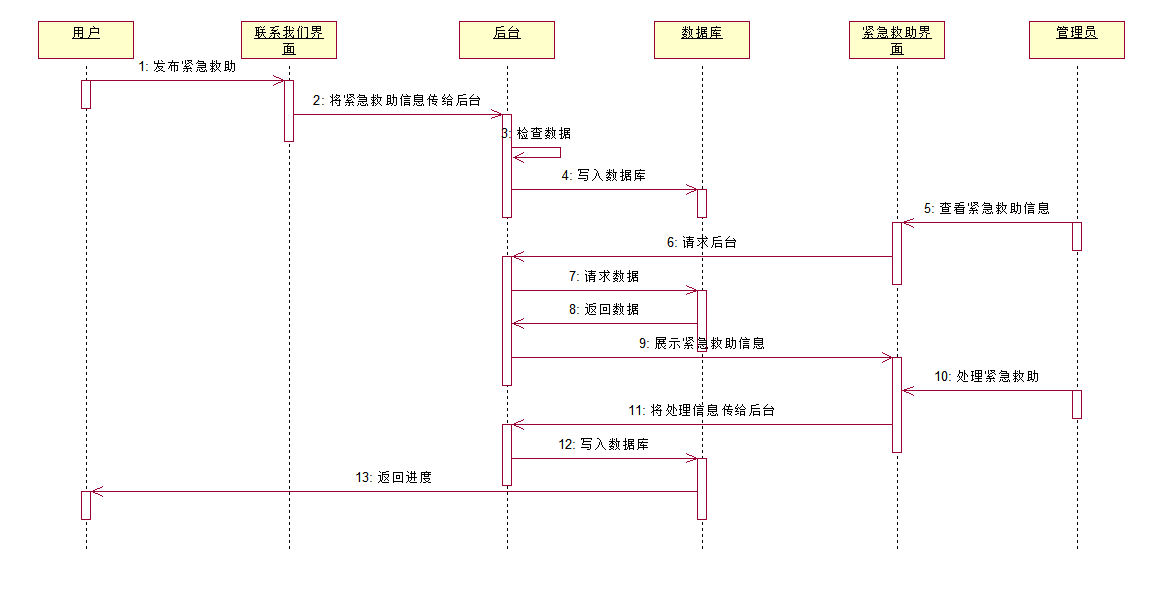


图3-7 紧急救助模块时序图

Fig. 3-7 Emergency rescue module sequence diagram

3.2.7用户信息管理模块设计

用户信息管理模块的作用是为用户提供了一个浏览和管理个人信息的途径。在后台方面，该模块使用同一个平台来执行添加、删除和修改数据等操作。但在前端方面，它被分成管理员模块的用户管理和用户模块的我的空间两个部分，分别提供不同的功能。

管理员模块是为实现管理员管理用户和超级管理员管理管理员的功能而存在的一个模块。在该模块中，超级管理员可以添加、编辑、删除普通管理员账号，并授予或收回其相关的管理权限。同时，普通管理员可以通过该模块对普通用户的信息进行管理和操作。

需要注意的是，管理员账号是无法自行注册的，只能由超级管理员通过该模块进行添加。在添加管理员时，超级管理员需要填写相应的账号信息，并设定其管理权限。管理员账号的权限一般包括用户管理、动物信息管理、志愿者管理、活动管理、救助管理等。不同的管理员账号可以拥有不同的管理权限，以便实现更精细化的管理和操作。

在管理员模块中，超级管理员还可以对管理员账号进行编辑和删除操作。编辑操作可以修改管理员账号的基本信息和管理权限，而删除操作可以永久性地删除该管理员账号，同时也会撤销其所有的管理权限。此外，超级管理员还可以查看所有管理员账号的列表和相关的操作日志，以便及时了解管理员账号的使用情况和操作历史。

综上所述，管理员管理模块是为实现管理员管理用户和超级管理员管理管理员的功能而存在的一个模块。通过该模块，超级管理员可以添加、编辑、删除普通管理员账号，并授予或收回其相关的管理权限，以实现更加精细化的管理和操作。

用户模块是为了方便用户对自己的个人资料进行修改等操作而存在的一个模块。该模块提供了基础的个人资料修改功能，包括用户基本信息、联系方式、账号安全等方面的修改。用户可以在该模块中自由编辑和更改自己的个人资料，以便更好地管理自己的账号和与他人互动。

此外，用户资料模块还丰富了功能的多样性，增加了许多与其他模块交互的功能。比如，在论坛模块中，用户可以通过个人资料模块来编辑自己的论坛个性签名、头像和简介等信息，以便更好地展示自己的个性和特点。

综上所述，用户资料模块不仅提供了基础的个人资料修改功能，还丰富了多样化的功能，提高了用户对平台的使用体验和满意度。同时，该模块与其他模块的交互，进一步增强了平台的整体性和互动性，提升了用户对平台的认可和信任。用户管理模块的时序图见图3-8。

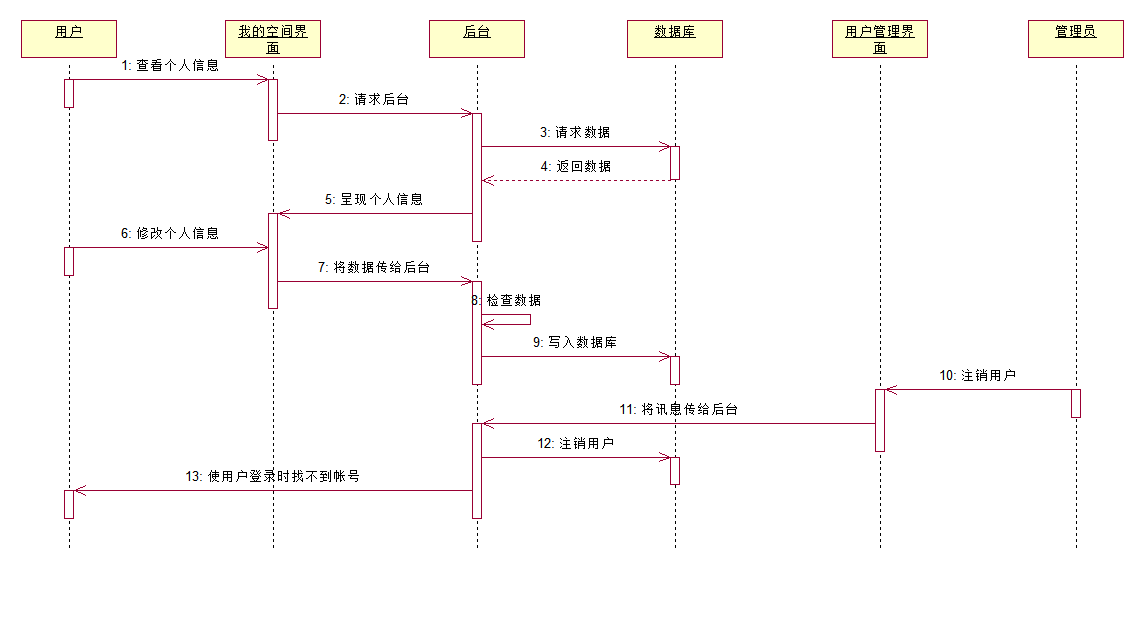


图3-8 用户信息管理模块时序图

Fig. 3-8 User information management module sequence diagram

3.2.8救助站管理模块设计

该模块是用来管理救助站信息的。救助站是指提供流浪动物救助服务的机构或组织，如流浪动物收容所、动物保护组织等。在这个模块中，管理员可以对救助站的信息进行增加、删除、修改、查询等操作。同时，为了提高用户对救助站的信任度，管理员可以对救助站进行审核，以确保其资质和服务质量。

在增加救助站信息时，管理员需要填写救助站的基本信息，包括名称、地址、联系方式、服务范围等。这些信息可以帮助用户更好地了解救助站的情况和服务内容。同时，管理员还可以上传救助站的照片和介绍，增加信息的可信度和吸引力。

在删除救助站信息时，管理员需要谨慎操作，避免误删或删错。一般情况下，管理员应该在删除前进行确认操作，以确保操作的正确性和安全性。

在修改救助站信息时，管理员可以根据实际情况进行修改，如修改地址、联系方式等信息。同时，管理员还可以对救助站的介绍和照片进行修改，以保证信息的准确性和完整性。

在查询救助站信息时，管理员可以根据关键词进行搜索，如救助站的名称、地址等。同时，管理员还可以对查询结果进行筛选和排序，以快速找到符合条件的救助站信息。

总之，救助站管理模块是一个非常重要的模块，它可以帮助管理员更好地管理救助站信息，提高用户对救助站的信任度和满意度。救助站管理模块的时序图见图3-9。

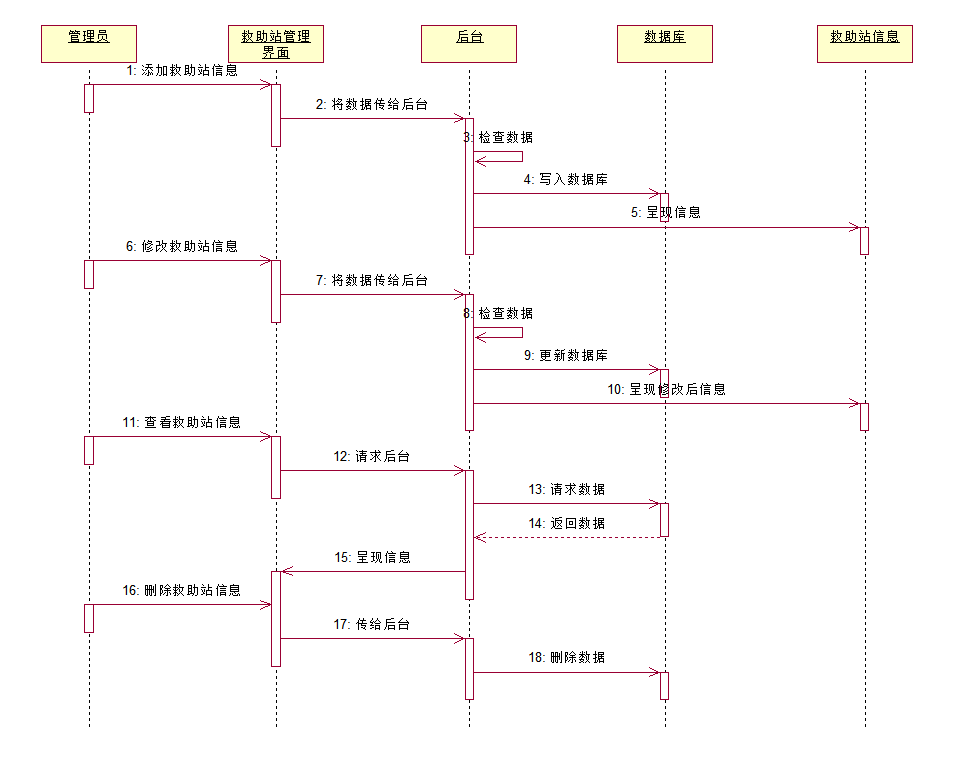


图3-9 救助站管理模块时序图

Fig. 3-9 Time sequence diagram of rescue station management module

3.2.9图片上传模块设计

图片上传模块是服务于其他需要实现上传图片功能的模块，依托七牛云的图片上传服务，将图片上传至七牛云后的地址存入数据库在需要使用上传图片功能的模块中，引入该模块提供的API接口和相关依赖库；通过API接口调用该模块的上传图片功能，将需要上传的图片文件发送到该模块；该模块会将图片上传至预先配置好的云存储服务提供商（如七牛云），并将图片上传后的URL地址存储至数据库中；其他模块可以通过数据库查询或者API接口调用，获取已经上传的图片URL地址，并在页面或者其他应用场景中展示出来。图片上传模块的时序图见图3-10。

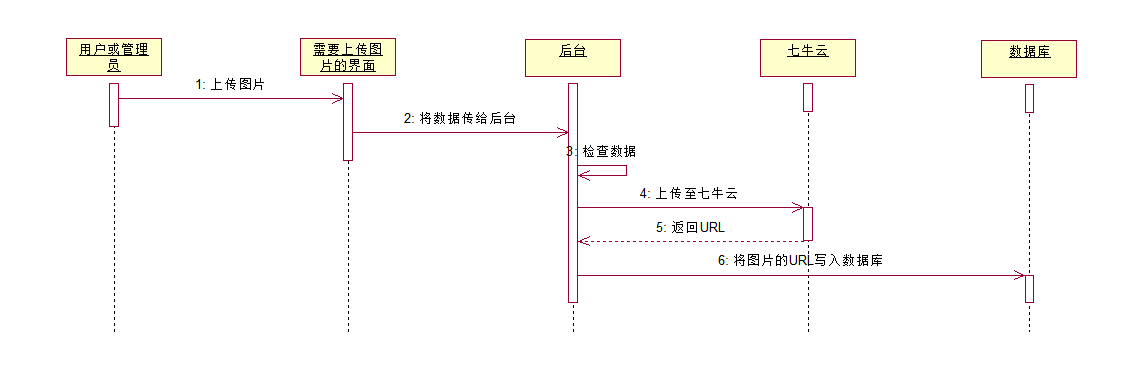


图3-10 图片上传模块时序图

Fig. 3-10 Picture upload module timing diagram

3.3数据库设计

数据库设计一般是指需求分析、概念结构设计、数据库表建立三大部分。

3.3.1数据库概念结构设计

下图3-11为整个流浪动物救助系统的总E-R实体关系图。

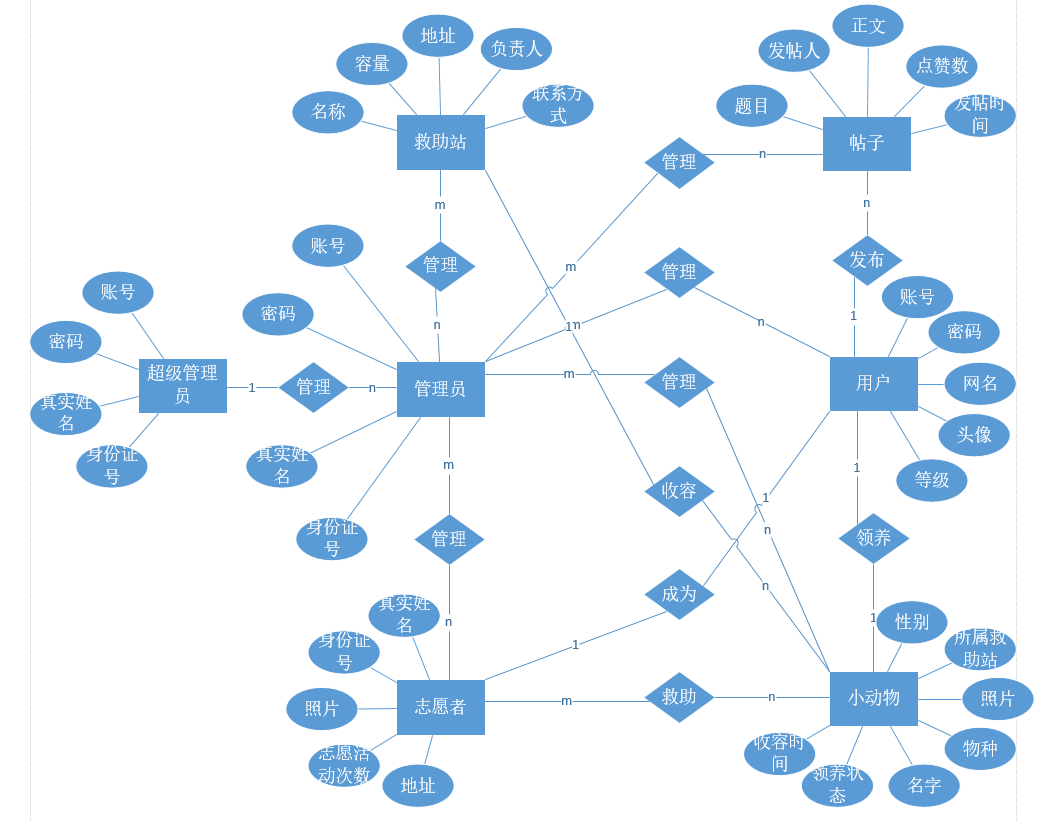


图3-11 流浪动物救助系统总E-R图

Fig. 3-11 Total E-R diagram of the stray animal rescue system

根据上图的结构可以清晰的了解本系统实体之间的联系方式，下面再罗列一些主要数据库的E-R模型图。

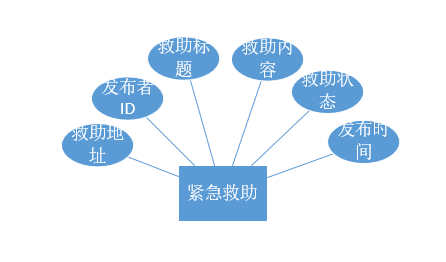


图3-12 紧急救助E-R图

Fig. 3-12 Emergency rescue E-R diagram

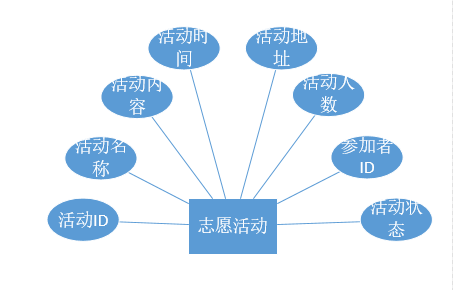


图3-13 志愿活动E-R图

Fig. 3-13 E-R diagram of volunteering activities

3.3.2数据库逻辑结构设计

根据3.3.1的数据库概念结构设计可以着手创建数据表了，下面我将罗列一些重要的数据表的设计。

**表3-1 管理员信息表**

**Tab. 3-1 admin\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| admin\_id | bigint | 0 | √ | √ | 管理员id(自动递增) |
| admin\_account | varchar | 255 |  |  | 管理员账号 |
| admin\_password | varchar | 255 |  |  | 管理员密码 |
| admin\_realname | varchar | 255 |  |  | 管理员真实姓名 |
| admin\_ID\_number | varchar | 255 |  |  | 管理员身份证号 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除(默认0) |

**表3-2 动物信息表**

**Tab. 3-2 animal\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| animal\_id | bigint | 0 | √ | √ | 流浪动物ID(自动递增） |
| animal\_name | varchar | 255 |  |  | 动物名字 |
| animal\_pictures | varchar | 255 |  |  | 动物照片 |
| animal\_sex | varchar | 255 |  |  | 动物性别 |
| animal\_age | varchar | 255 |  |  | 动物年龄 |
| animal\_rescue\_station\_id | bigint | 0 | √ |  | 所属救助站 |
| species | varchar | 255 |  |  | 物种 |
| varieties | varchar | 255 |  |  | 品种 |
| vaccination | varchar | 255 |  |  | 是否接种 |
| vaccination\_time | datetime | 0 |  |  | 接种时间 |
| sterilization | varchar | 255 |  |  | 是否绝育 |
| sterilization\_time | datetime | 0 |  |  | 绝育时间 |
| repellent | varchar | 255 |  |  | 是否驱虫 |
| repellent\_time | datetime | 0 |  |  | 驱虫时间 |
| animal\_adoption | varchar | 255 |  |  | 领养状态 |
| user\_id | bigint | 0 |  |  | 领养人ID |
| animal\_adoption\_time | datetime | 0 |  |  | 领养时间 |
| adoption\_notes | text | 0 |  |  | 领养备注 |
| detailed\_information | text | 0 |  |  | 详细信息 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 更新时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除(默认0) |

**表3-3 救助站信息表**

**Tab. 3-3 rescue\_station\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| rescue\_station\_id | bigint | 0 | √ | √ | 救助站ID(自动递增) |
| rescue\_station\_name | varchar | 255 |  |  | 救助站名称 |
| rescue\_station\_capacity | int | 0 |  |  | 救助站容量 |
| rescue\_current\_capacity | int | 0 |  |  | 目前容纳数量 |
| rescue\_station\_address | varchar | 255 |  |  | 救助站地址 |
| rescue\_station\_principal | varchar | 255 |  |  | 救助站负责人 |
| rescue\_station\_phone | varchar | 255 |  |  | 救助站联系方式 |
| rescue\_station\_picture | varchar | 255 |  |  | 救助站照片 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除(默认0) |

**表3-4 紧急救助信息表**

**Tab. 3-4 sos\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| sos\_id | bigint | 0 | √ | √ | 紧急救助ID（自动递增） |
| user\_id | bigint | 0 |  |  | 发布救助用户ID |
| sos\_name | varchar | 255 |  |  | 救助名称 |
| sos\_address | varchar | 255 |  |  | 救助地址 |
| sos\_status | varchar | 255 |  |  | 处理状态(未处理 处理中 已完成) |
| sos\_detailed\_information | text | 0 |  |  | 救助详细信息 |
| sos\_reply | varchar | 255 |  |  | 暂返信息 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除(默认0) |

**表3-5 用户信息表**

**Tab. 3-5 user\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| user\_id | bigint | 0 | √ | √ | 用户ID（自动递增） |
| user\_net\_work\_name | varchar | 255 |  |  | 用户网名 |
| user\_account | varchar | 255 |  |  | 用户账号 |
| user\_password | varchar | 255 |  |  | 用户密码 |
| user\_realname | varchar | 255 |  |  | 用户真实姓名 |
| user\_ID\_number | varchar | 255 |  |  | 用户身份证号 |
| user\_avatar | varchar | 255 |  |  | 用户头像 |
| user\_email | varchar | 255 |  |  | 用户邮箱 |
| user\_phone | varchar | 255 |  |  | 用户手机 |
| user\_sex | varchar | 255 |  |  | 用户性别 |
| user\_nation | varchar | 255 |  |  | 用户民族 |
| user\_birthday | datetime | 0 |  |  | 用户出生日期 |
| user\_age | varchar | 255 |  |  | 用户年龄 |
| user\_address | varchar | 255 |  |  | 用户地址 |
| user\_reply | varchar | 255 |  |  | 领养暂返信息 |
| user\_level | varchar | 255 | √ |  | 用户等级（初始为0） |
| user\_concern | int | 0 | √ |  | 用户的关注数(初始为0) |
| user\_fans | int | 0 | √ |  | 用户的粉丝数(初始为0) |
| user\_like | bigint | 0 | √ |  | 用户的获赞数(初始为0) |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除（默认0） |

**表3-6 志愿活动信息表**

**Tab. 3-6 voluntary\_activity\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| voluntary\_activity\_id | bigint | 0 | √ |  | 志愿活动ID |
| order\_number | bigint | 0 | √ | √ | 序号(自动递增) |
| volunteer\_id | bigint | 0 |  |  | 志愿者ID |
| voluntary\_activity\_name | varchar | 255 |  |  | 活动名称 |
| voluntary\_activity\_address | varchar | 255 |  |  | 活动地址 |
| voluntary\_activity\_time | datetime | 0 |  |  | 活动时间 |
| voluntary\_activity\_people\_numble | int | 0 |  |  | 活动人数 |
| status | varchar | 255 |  |  | 活动状态 |
| voluntary\_activity\_detailed\_information | text | 0 |  |  | 活动详细信息 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除(默认0) |

**表3-7 志愿者信息表**

**Tab. 3-7 volunteer\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| volunteer\_id | bigint | 0 | √ | √ | 志愿者ID(自动递增) |
| volunteer\_user\_id | bigint | 0 | √ |  | 申请成为志愿者的用户ID |
| volunteer\_examine\_state | int | 0 | √ |  | 审核状态(默认0) |
| volunteer\_old\_activities | text | 0 |  |  | 过去参加志愿活动情况 |
| volunteer\_activities\_number | int | 0 | √ |  | 志愿活动次数(默认0) |
| volunteer\_title | varchar | 255 |  |  | 志愿者称号 |
| adoption\_notes | varchar | 255 |  |  | 申请备注 |
| volunteer\_reply | varchar | 255 |  |  | 暂返信息 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除(默认0) |

**表3-8回复信息表**

**Tab. 3-8 answer\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| answer\_id | bigint | 0 | √ | √ | 回复ID |
| comment\_id | bigint | 0 | √ |  | 回复的评论ID |
| answer\_user\_id | bigint | 0 | √ |  | 回复者用户ID |
| to\_be\_answer\_user\_id | bigint | 0 | √ |  | 被回复用户ID |
| answer\_text | text | 0 |  |  | 内容 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除（默认0） |

**表3-9 评论信息表**

**Tab. 3-9 comment\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| comment\_id | bigint | 20 | √ | √ | 评论ID |
| post\_id | bigint | 0 | √ |  | 帖子ID |
| user\_id | bigint | 0 | √ |  | 发帖用戶ID |
| comment\_text | text | 0 |  |  | 正文 |
| comment\_like | int | 0 |  |  | 评论点赞数 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除（默认0） |

**表3-10 图片信息表**

**Tab. 3-10 picture\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| picture\_id | bigint | 0 | √ | √ | 图片ID |
| post\_id | bigint | 0 |  |  | 帖子ID |
| comment\_id | bigint | 0 |  |  | 评论ID |
| animal\_id | bigint | 0 |  |  | 小动物ID |
| picture\_url | varchar | 255 |  |  | 图片地址 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除（默认0） |

**表3-11 帖子信息表**

**Tab. 3-11 post\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **不是null** | **主键** | **注释** |
| post\_id | bigint | 0 | √ | √ | 帖子ID |
| user\_id | bigint | 0 | √ |  | 发帖人ID |
| post\_headline | varchar | 255 |  |  | 帖子标题 |
| post\_text | text | 0 |  |  | 正文 |
| post\_like | int | 0 |  |  | 点赞数 |
| post\_comment | int | 0 |  |  | 评论数 |
| create\_time | datetime | 0 | √ |  | 创建时间 |
| update\_time | datetime | 0 | √ |  | 修改时间 |
| is\_delete | int | 0 | √ |  | 逻辑删除（默认0） |

3.4本章总结

经过了模块设计、数据库设计后，已经非常清晰明了的获知系统的结构和功能。

4系统实现

本章就开始对流浪动物救助系统进行详细设计和实现，将根据前两章的需求分析和功能模块设计来开展工作。下面将着手从系统页面实现与业务逻辑实现两方面开始介绍。

4.1用户模块实现

4.1.1登录注册模块实现

用户在访问本系统时，再进行一些只有登录后才能进行的操作时，系统会提示用户登录，实现代码如下：

this.isFocus = true;

                this.handleInput();

                if(!this.$cookie.get('userId')){

                    e.srcElement.blur();

                    this.$confirm('当前还未登录哦，是否前往登录?', '提示', {

                    confirmButtonText: '确定',

                    cancelButtonText: '取消',

                    type: 'warning'

                    }).then(() => {

                        this.$router.push({name:'login'})

                    }).catch(() => {

                        this.$message({

                        type: 'info',

                        message: '已取消'

                    });

                    });

                }

界面如下图4-1所示。



图4-1 提示用户登录界面

Fig. 4-1 E-R The login page is displayed

此时点击确认就会跳转到登录界面，如果用户有账号就可输入手机号和密码登录，账号密码错误就会提示错误原因，成功登录后会返回刚才浏览的界面，可以继续因提醒登录而被中断的操作，登陆界面如下图4-2所示。

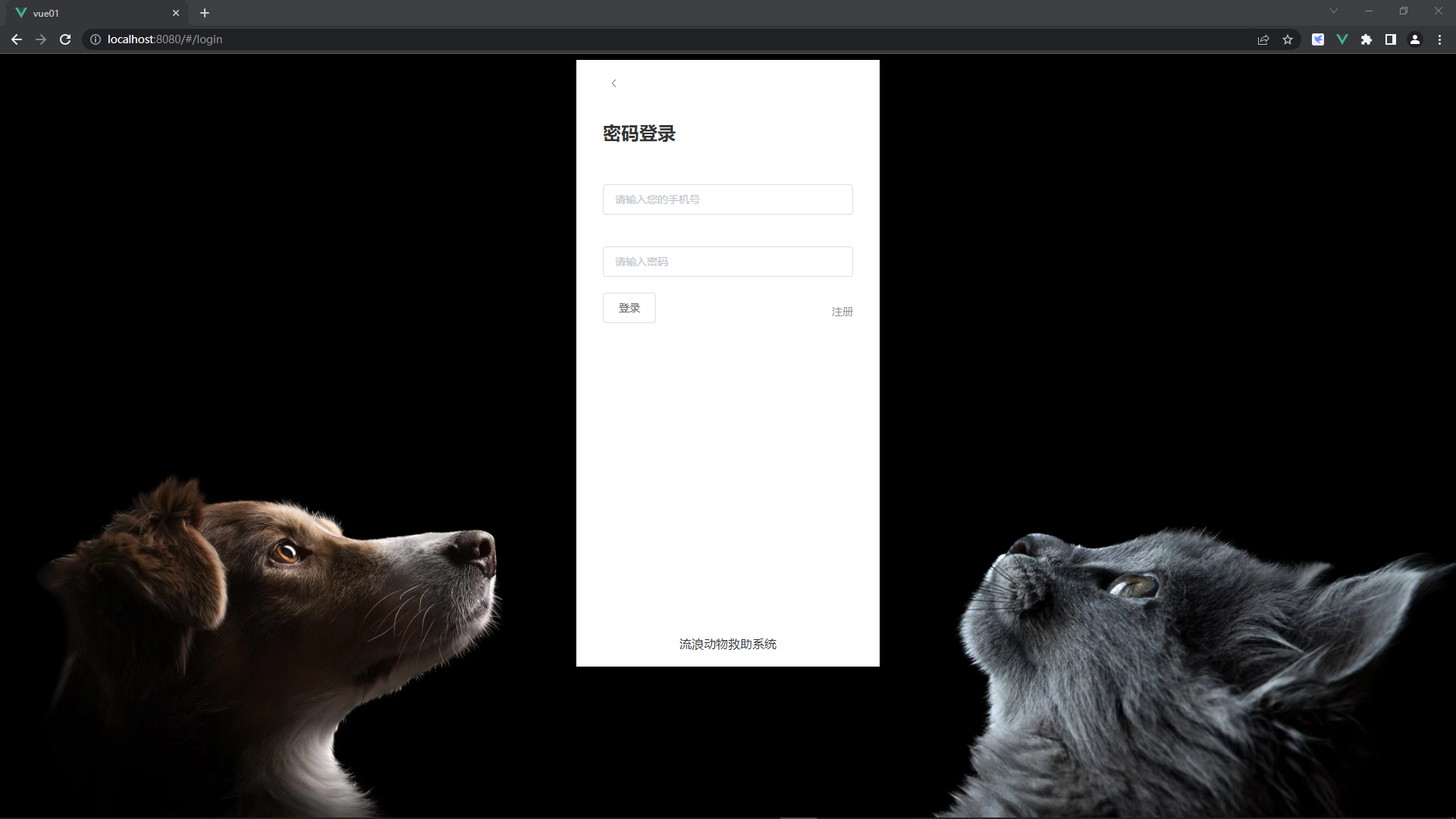


图4-2 用户登录界面

Fig. 4-2 User login interface

如果用户没有账号，可以点击注册后进行极简注册，只需手机号和自己设定的密码即可快速注册，注册成功后会直接跳转到登录界面，并且输入框会保留在注册界面所输入的信息，方便用户直接登录，注册界面如下图4-3所示。

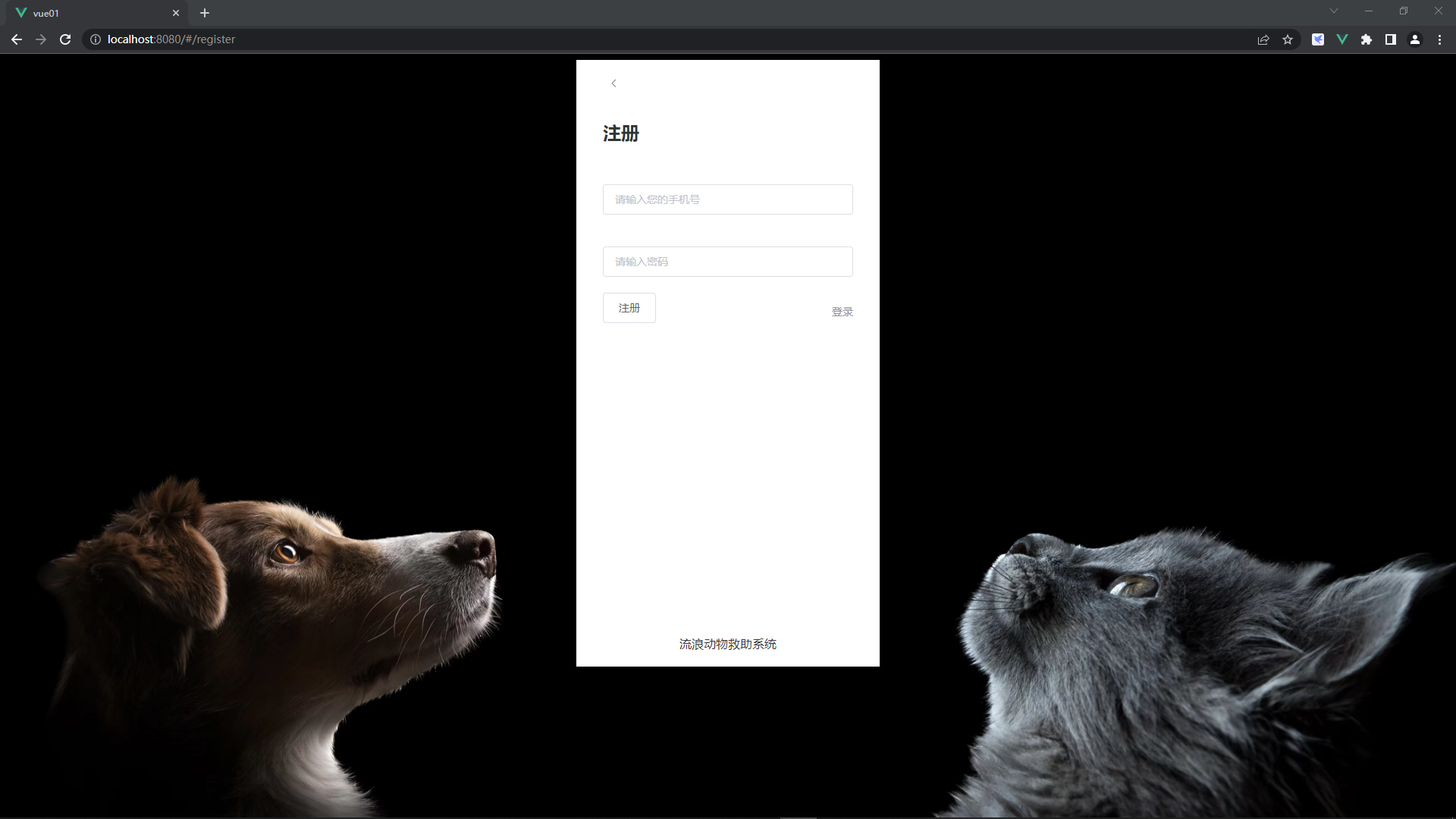


图4-3用户注册界面

Fig. 4-3User registration interface

4.1.2小动物模块实现

进入小动物模块后可以看到我们展示小动物们，点击感兴趣的即可查看详情。界面如图4-4和4-5所示。

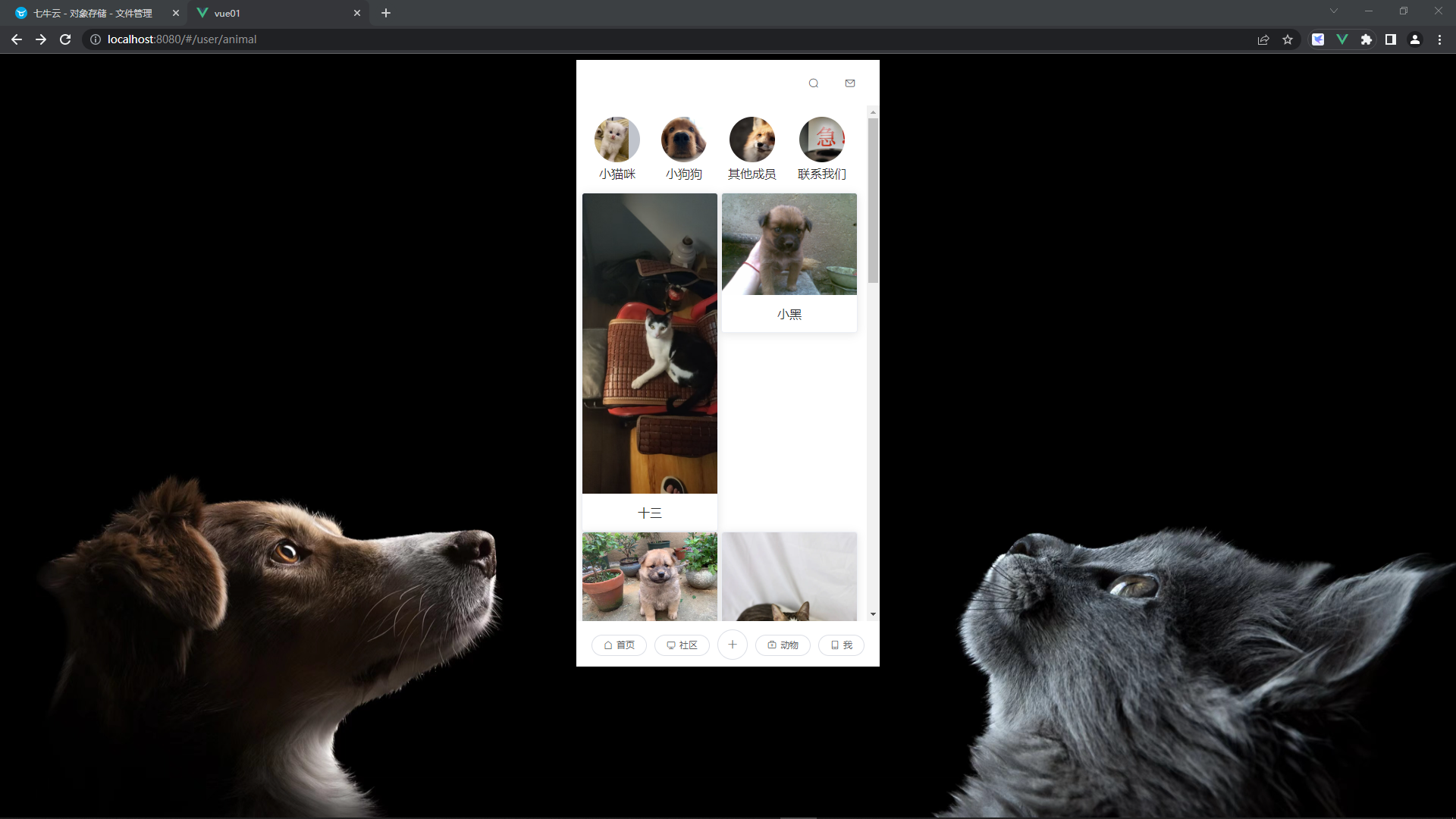


图4-4小动物界面

Fig. 4-4 Small animal interface

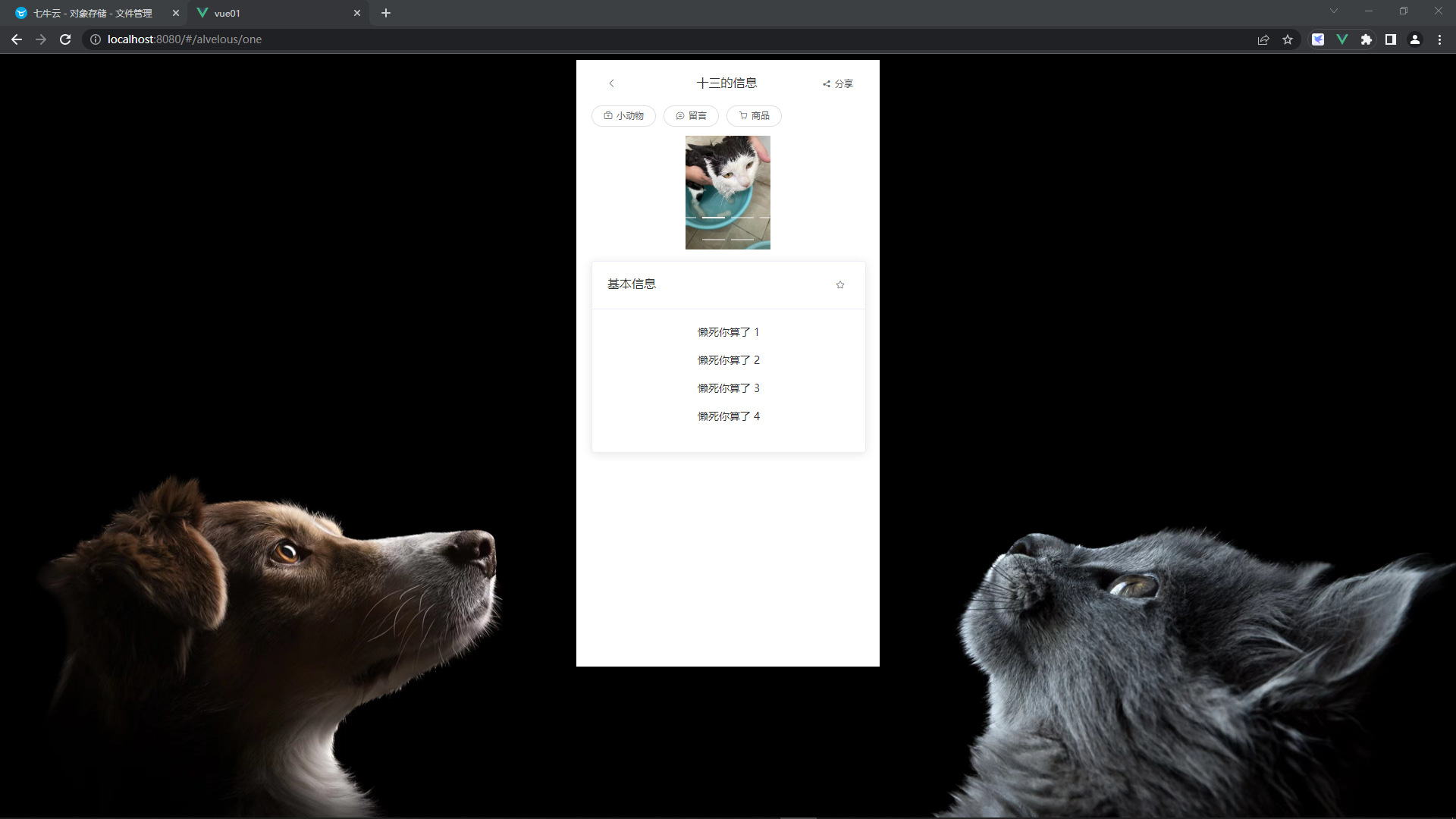


图4-5小动物详情界面

Fig. 4-5 Small animal details interface

4.1.3志愿者模块实现

用户在进入我的空间界面时，可以看到界面下方的志愿者模块，在该模块可以清晰地看到成为志愿者后的服务时长、服务次数、参加的志愿活动等信息。在该模块用户还可以看到正在招募志愿者的志愿活动信息并参加该活动，实现代码如下所示：

<div class="avatar">

          <img :src="volunteer.avatar" alt="volunteer avatar">

        </div>

        <div class="info">

          <div class="name">{{ volunteer.name }}</div>

          <div class="id">{{ 'ID:' + volunteer.id }}</div>

          <div class="title">{{ volunteer.title }}</div>

          <div class="service">

            <span>服务时长：</span>

            <span>{{ volunteer.serviceTime }}</span>

          </div>

          <div class="service">

            <span>服务次数：</span>

            <span>{{ volunteer.serviceCount }}</span>

          </div>

        </div>

      </div>

      <div class="bottom">

        <div class="title">参加过的志愿活动</div>

        <el-table :data="volunteer.activities" style="width: 100%">

          <el-table-column prop="name" label="活动名称"></el-table-column>

          <el-table-column prop="time" label="参加时间"></el-table-column>

        </el-table>

      </div>

界面如下图4-6所示。

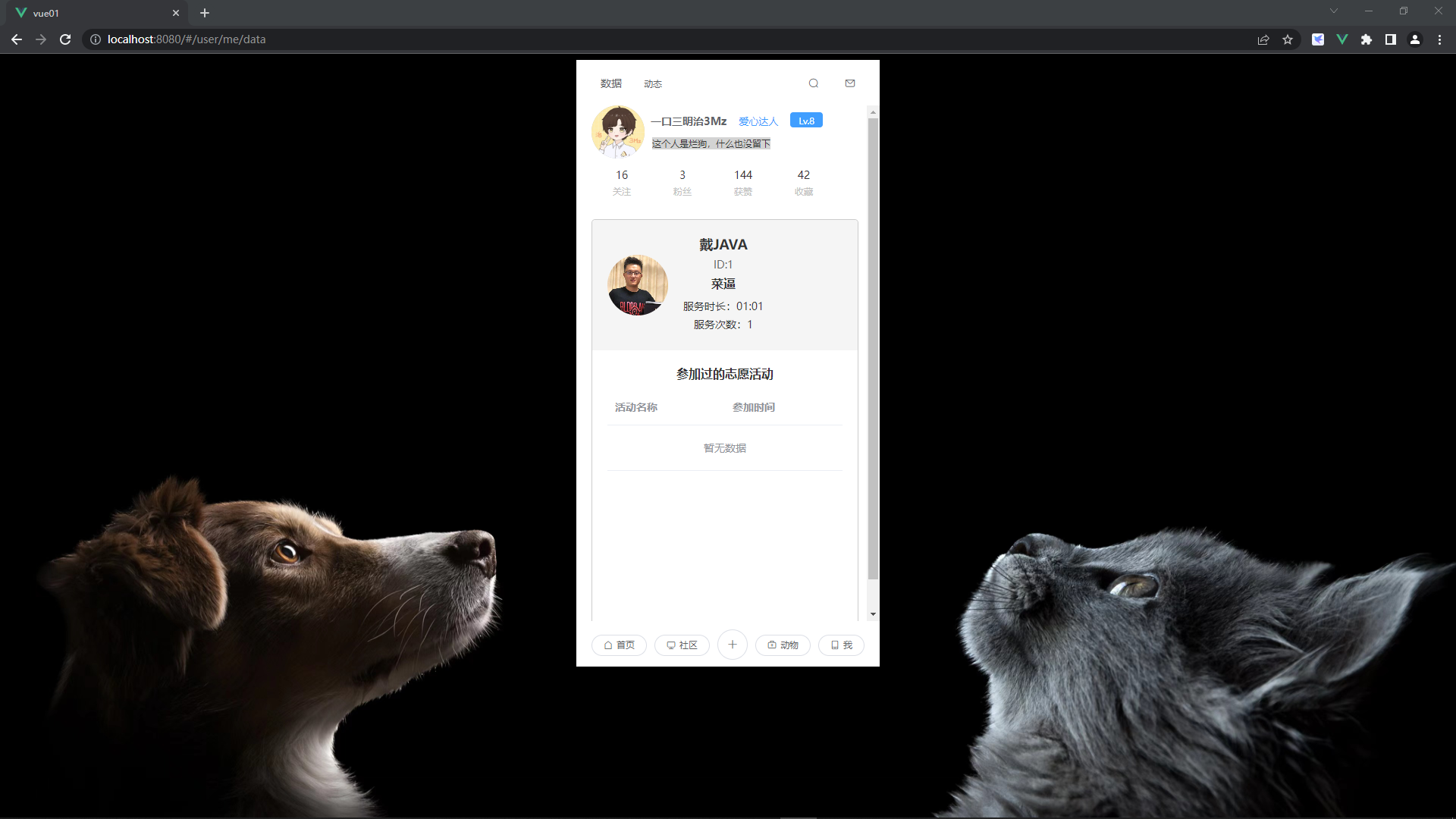


图4-6志愿者界面

Fig. 4-6 Volunteer interface

4.1.4论坛模块实现

在论坛模块，用户可以随意浏览其他用户发送的帖子，帖子的呈现形式是无限滚动条形式，先只显示一部分帖子，当滚动条拉到底部时会刷新出一些新得帖子。具体代码为： <ul

      class="list"

      v-infinite-scroll="load"

      infinite-scroll-disabled="disabled"

      infinite-scroll-distance="10px">

      <li v-for="i in count" :key="i" class="list-item">

        <PostComment :Obj="obj1[i-1]" v-if="obj1[i-1]" />

        <el-divider v-if="obj1[i-1]"></el-divider>

      </li>

    </ul>

    <p v-if="loading">加载中...</p>

    <p v-if="noMore">没有更多了</p>

当用户刷自己感兴趣的帖子时，可以通过该模块的点赞、评论等功能实现互动。发表评论可以发送图片、emoji等元素，增添了社区的欢快气氛。当用户点击发表评论的输入框前会显示点赞和收藏的按钮，点击后会显示emoji和添加图片的按钮。只有输入框有内容时才会出现发布按钮，监听焦点获取状态代码如下：

         mounted() {

            // 监听文本框的输入状态

            this.$refs.wenben.$el.addEventListener('input', this.handleInput);

        },

添加的图片会排布在输入框上，每个照片带有一个删除按钮方便于用户管理，显示照片实现代码如下：

<div v-if="imageList.length>0" class="image-list-container">

            <div class="image-list" v-for="(image, index) in imageList" :key="index">

                <el-image  :src="image" style=" height: 50px;" ></el-image>

                <button class="delete-btn" @click="deleteImage(index)">×</button>

            </div>

        </div>

论坛刷帖界面如下图4-7所示。

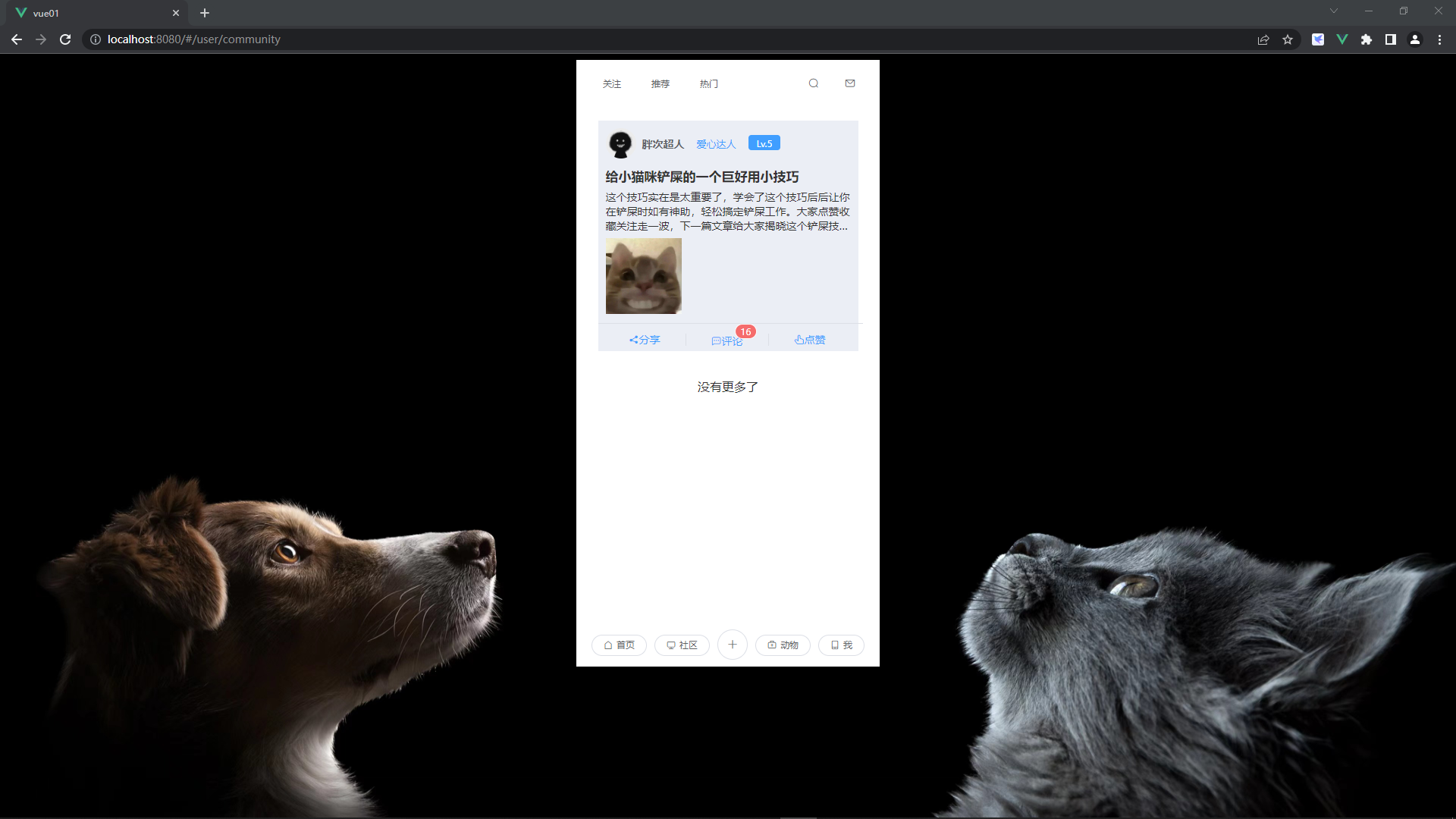


图4-7论坛界面

Fig. 4-7 Forum interface

帖子详情界面如下图4-8所示。

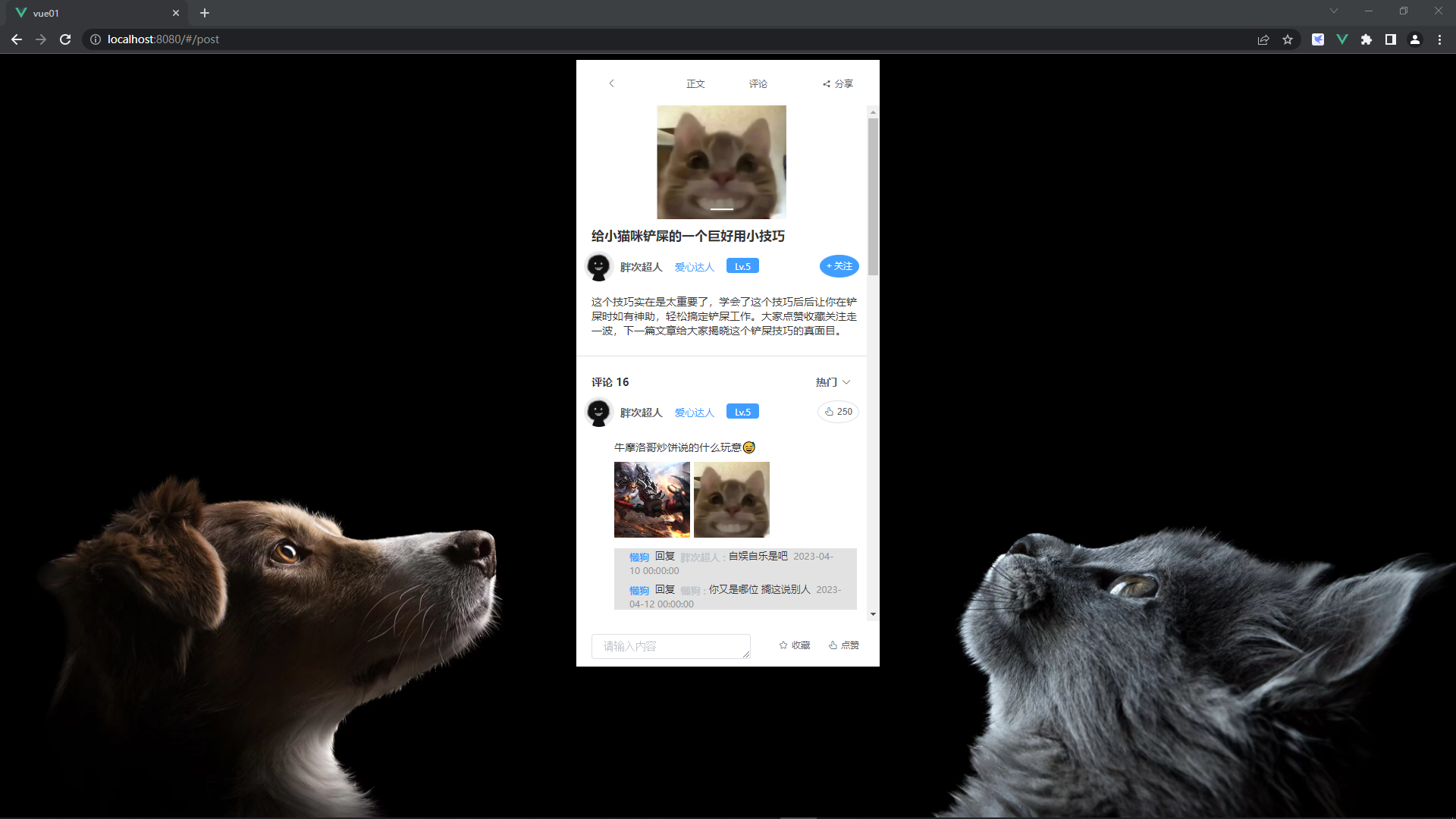


图4-8帖子详情界面

Fig. 4-8 Post details interface

4.1.7用户信息模块实现

在用户登录后，进入“我的空间”界面，用户可以对个人信息进行管理。这个功能包括修改头像、网名、个性签名等内容，帮助用户更好地展现自己的个性和特点。

此外，该模块还与论坛模块关联，用户可以通过在论坛点击其他用户的网名，跳转到该用户的个人空间来查看该用户的详细信息，如个人资料、发表的文章、评论记录等等。这样不仅可以提高用户之间的互动和交流，还可以让用户更加了解其他用户的兴趣和经历，有助于用户之间的互相学习和交流。同时，也方便了用户管理自己的信息和资源，提高了用户的使用体验和满意度。实现信息展示代码如下：

<div style="width: 350px;float:left;height: 70px;margin-left: 10px;margin-top: 10px;margin:-20px 0">

        <!--头像-->

        <div style="float:left">

            <el-avatar :size="70" :src="Obj.url"></el-avatar>

        </div>

        <!--姓名-->

        <div style="float:left;padding: 8px;">

  <el-link href="#" target="\_blank" style="font-weight:bold;" :underline="false">{{ Obj.name }}</el-link>

        </div>

        <!--称号-->

        <div style="float:left;padding: 8px;">

<el-link href="#" target="\_blank" type="primary" style="font-size:13px" :underline="false">{{ Obj.chenghao }}</el-link>

        </div>

        <!--等级-->

        <div style="float:left;padding: 8px;">

            <el-tag effect="dark">Lv.{{ Obj.levie }}</el-tag>

        </div>

        <div style="margin:0 10px ;float:left">

<span style="font-size: 1px;background: #d3d3d3;">{{ Obj.qianming }}</span>

        </div>

    </div>

界面如下图4-9所示。

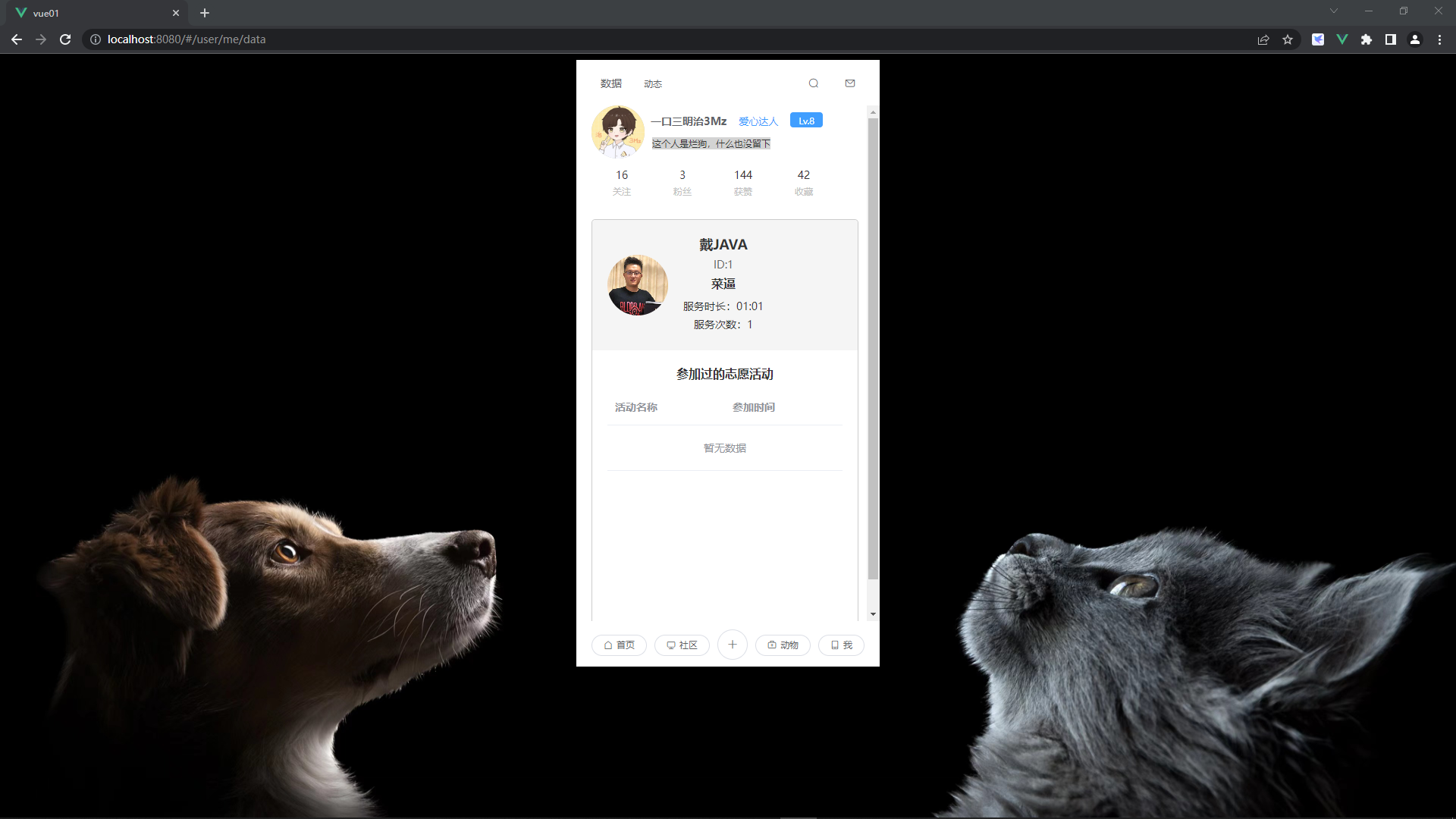


图4-9我的空间界面

Fig. 4-9 My spatial interface

4.2管理员模块实现

4.2.1领养信息管理实现

领养信息界面用于处理用户申请领养请求，点击查看详情后便可对请求进行处理，可以选择处理状态来查看不同状态的请求，可以根据需求将请求页以表格的形式导出，具体代码如下所示：

@GetMapping("/exportAnimalInfo")  
 public void exportAnimalInfo(@ModelAttribute Vague vague, @RequestParam(required = false, defaultValue = "99999") Integer pageSize,  
 @RequestParam(required = false, defaultValue = "1") Integer current, HttpServletResponse response) {  
 List<ExportAnimal> animalInfos = animalInfoService.exportAnimalData(vague, pageSize, current);  
 ExcelUtils.export(response,"动物领养信息",animalInfos,ExportAnimal.class);  
  
}

详细界面见下图4-10。

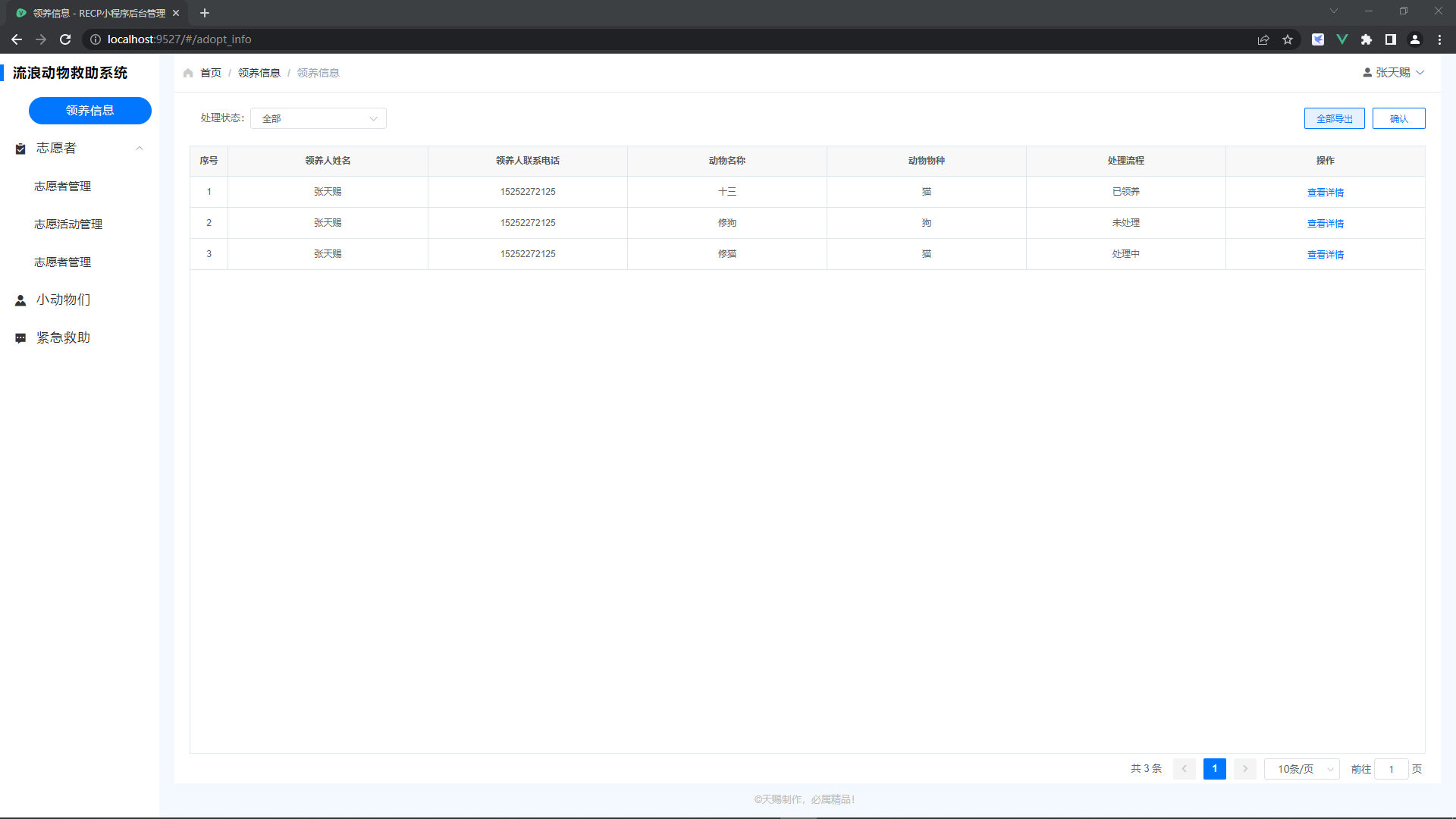


图4-10 领养信息界面

Fig. 4-10 Adoption Information Interface

4.2.2志愿者模块实现

志愿者界面下分三个模块：志愿者管理、志愿活动管理、志愿申请处理，每个分模块负责各不同的功能，下面为三个模块的详细介绍。

4.2.2.1志愿申请界面

志愿申请界面是用于处理用户模块的志愿者申请操作的界面。点击查看详情即可对申请进行处理。当用户的申请信息符合要求时，点击同意即可。当条件不符合时，可以给申请人返回一条拒绝理由。同意和拒绝信息都将保存在数据库中，具体sql语句如下所示：

<!--保存暂返信息-->  
 <update id="updateVolunteerReply">  
 update volunteer\_info set volunteer\_reply = #{volunteerReply} where volunteer\_id = #{volunteerId};  
 </update>  
 <!--同意成为志愿者-->  
 <update id="updateVolunteerExamineState">  
 update volunteer\_info set volunteer\_examine\_state = 1 where volunteer\_id = #{volunteerId};  
 </update>

界面如下图4-11所示。

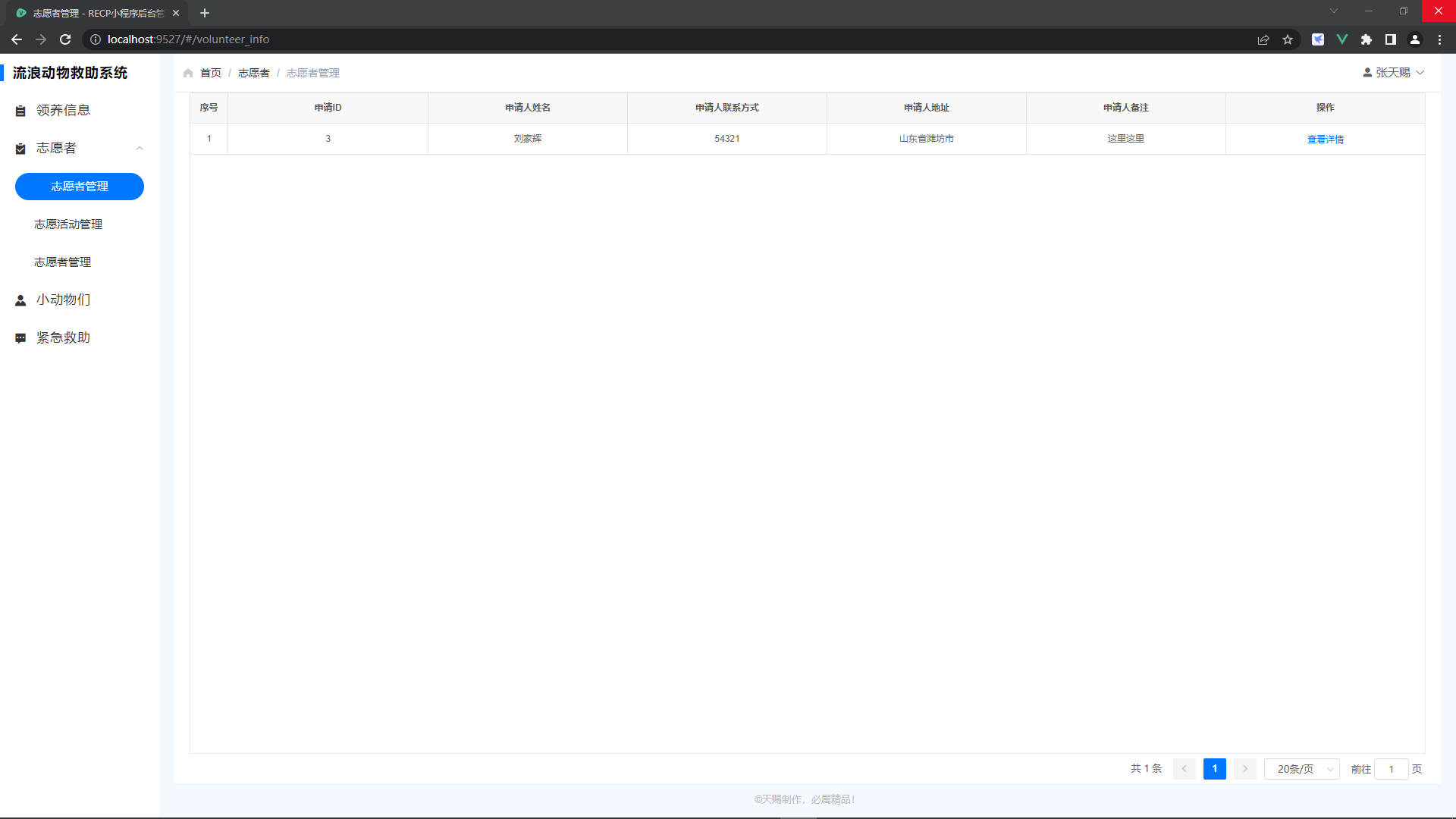


图4-11 志愿申请界面

Fig. 4-11 Volunteer Application Screen

4.2.2.2志愿活动管理界面

志愿活动管理界面用于对志愿活动进行增删改查，点击添加按钮编辑志愿活动信息，编辑完成信息后即可选择发布，志愿活动有三种状态，未发布、进行中、已完成。可以从菜单栏选择查看三种状态中的一种来查看。

支持根据志愿活动名称和志愿者名称模糊查询，相关的sql语句如下所示：

<!--模糊查询志愿活动-->  
<select id="selectVoluntaryActivityInfo" resultType="com.ztc.strayanimal.entiy.VoluntaryActivityInfo">  
 SELECT DISTINCT v1.voluntary\_activity\_id,v1.voluntary\_activity\_name,v1.voluntary\_activity\_time,  
 v1.voluntary\_activity\_address,v1.voluntary\_activity\_people\_numble,v1.status,  
 v1.voluntary\_activity\_detailed\_information  
 FROM voluntary\_activity\_info v1, volunteer\_info v2 , user\_info u  
 where  
 v1.is\_delete = 0  
 and v2.is\_delete = 0  
 and u.is\_delete = 0  
 <if test="vag.userRealname !=null and vag.userRealname !=''">  
 and v1.volunteer\_id = v2.volunteer\_id and v2.volunteer\_user\_id = u.user\_id  
 </if>  
 <if test="vag.volunteerId !=null and vag.volunteerId !=''">  
 and v1.volunteer\_id = #{vag.volunteerId}  
 </if>  
 <if test="vag.voluntaryActivityName !=null and vag.voluntaryActivityName !=''">  
 and v1.voluntary\_activity\_name like CONCAT('%',#{vag.voluntaryActivityName},'%')  
 </if>  
 <if test="vag.userRealname !=null and vag.userRealname !=''">  
 and u.user\_realname like CONCAT('%',#{vag.userRealname},'%')  
 </if>  
 <if test="vag.status !=null and vag.status !=''">  
 and v1.status = #{vag.status}  
 </if>  
</select>

并且引入了导出功能。具体界面见下图4-12。

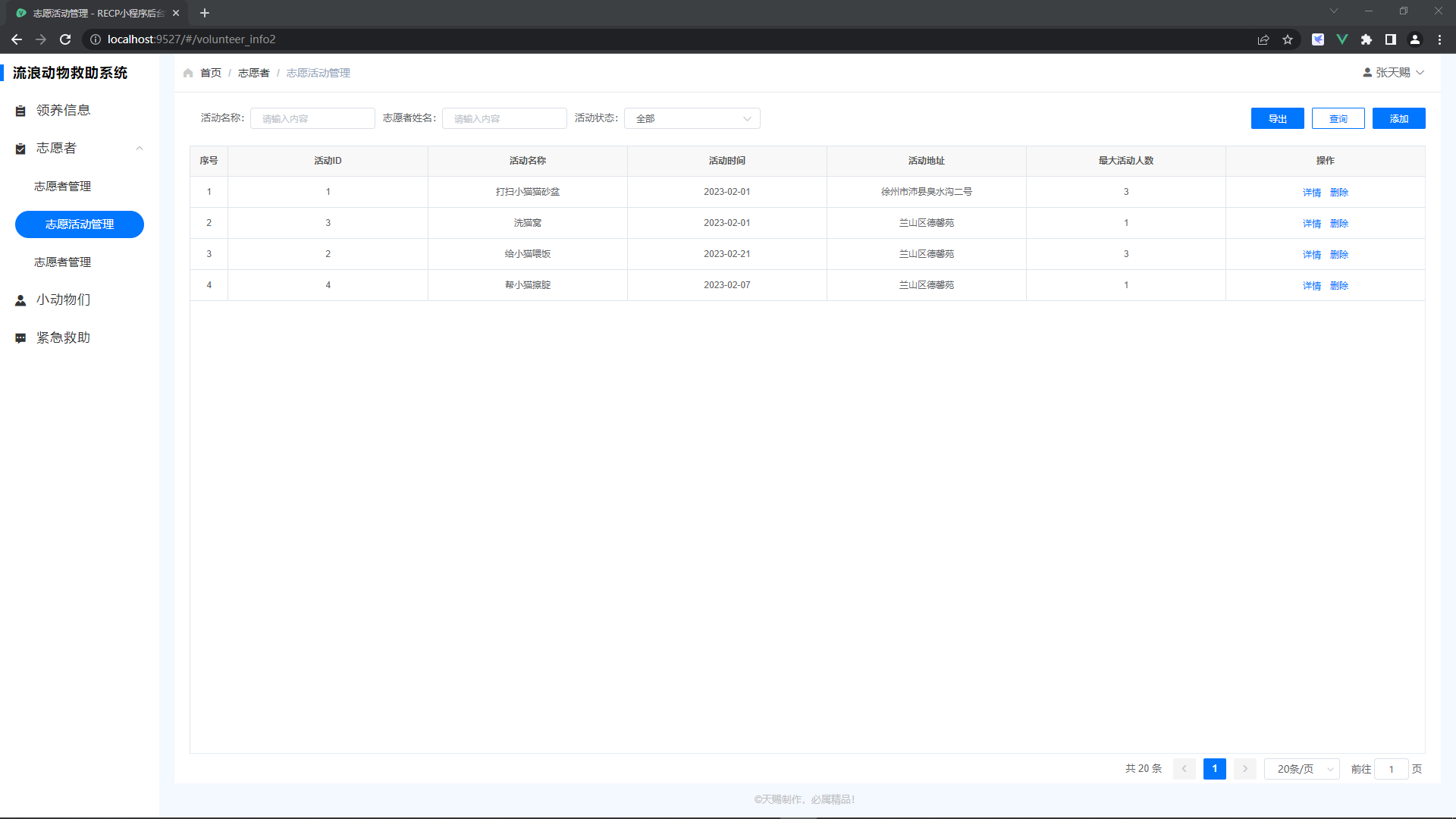


图4-12 志愿活动管理界面

Fig. 4-12 Volunteer management interface

4.2.2.3志愿者管理界面

志愿者管理界面是一个用于对志愿者进行基本的增删改查管理的工具。在该界面中，管理员可以方便地查看所有志愿者的基本信息，并对其进行增加、删除、修改等操作。

管理员可以通过在搜索框中输入志愿者的姓名或联系方式来进行模糊查询，以方便快速地找到目标志愿者的信息。这个功能非常有用，特别是在志愿者数量较多的情况下，可以减轻管理员的管理负担，提高管理效率。

此外，志愿者管理界面还提供了详细的志愿者信息展示，包括志愿者的姓名、性别、年龄、联系方式、加入时间等等。展示时间信息需要将其转变为年-月-日 时-分-秒，具体代码如下：

if(volunteerInfo.getUserBirthday() != null){  
 String strUserBirthday = new SimpleDateFormat("yyyy-MM-dd").format(volunteerInfo.getUserBirthday());  
 volunteerInfo.setStrUserBirthday(strUserBirthday);  
}

管理员可以随时查看志愿者的服务记录和参加过的志愿活动等信息，帮助管理员更好地了解志愿者的情况，提高管理质量和效率。

界面如下4-13所示。

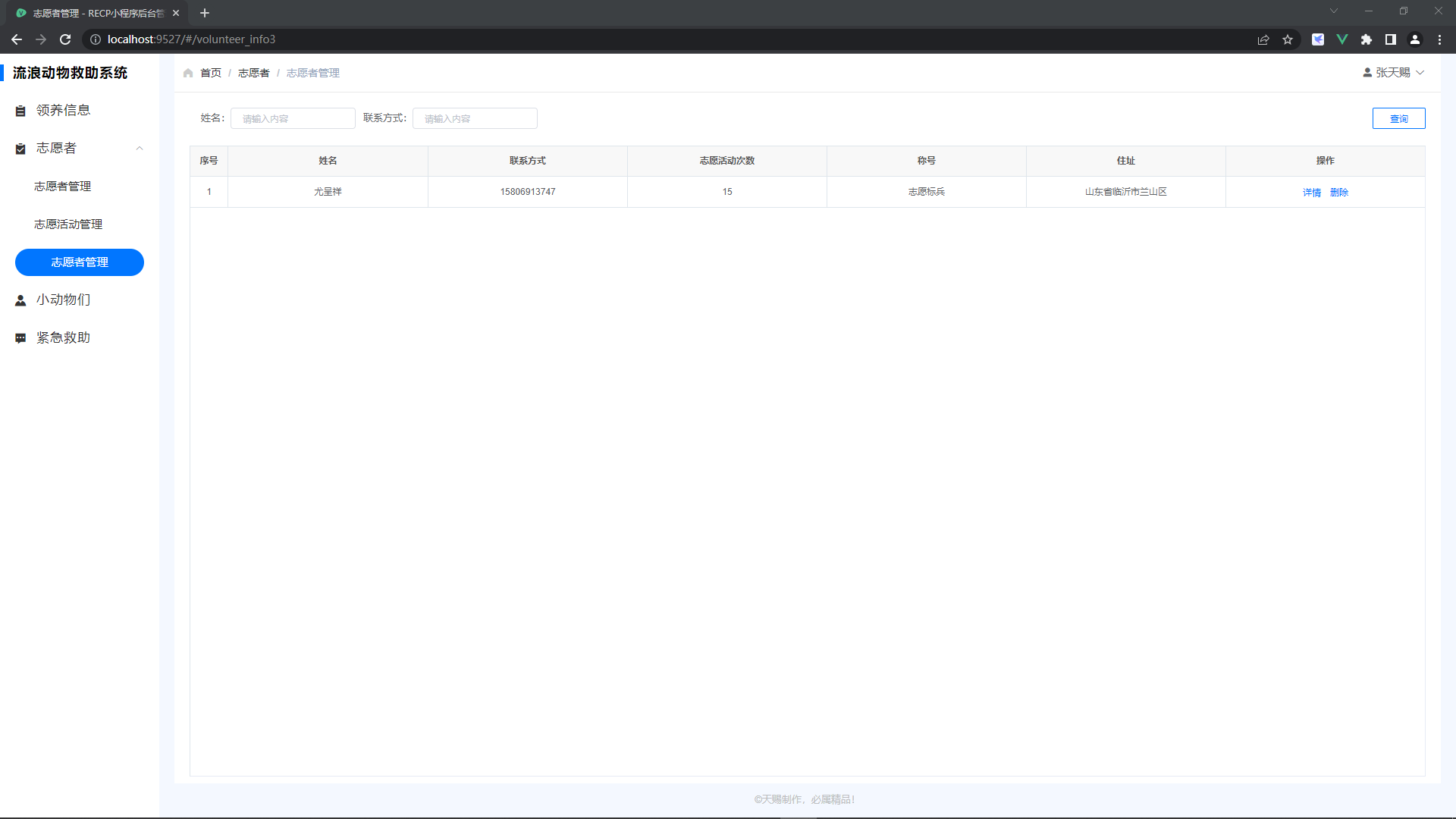


图4-13 志愿者管理界面

Fig. 4-13 Volunteer Management interface

4.2.3小动物模块实现

小动物们界面是一个重要的功能模块，旨在提供对小动物基本操作的管理。在该界面，用户可以进行小动物的增加、删除、修改和查询操作。展示这些信息使用了分页，相关sql语句如下所示：

<select id="getThisPageAllAnilmalsInfo" resultType="com.ztc.strayanimal.entiy.ExportAnimal" >  
 SELECT  
 animal\_info.user\_id,  
 animal\_info.animal\_id,  
 animal\_info.animal\_name,  
 animal\_info.animal\_sex,  
 animal\_info.animal\_age,  
 animal\_info.species,  
 animal\_info.varieties,  
 animal\_info.vaccination,  
 animal\_info.animal\_adoption,  
 animal\_info.vaccination\_time,  
 animal\_info.sterilization,  
 animal\_info.sterilization\_time,  
 animal\_info.repellent,  
 animal\_info.repellent\_time,  
 user\_info.user\_realname,  
 user\_info.user\_phone,  
 user\_info.user\_email,  
 rescue\_station\_info.rescue\_station\_name,  
 user\_info.user\_address  
 FROM  
 animal\_info  
 INNER JOIN user\_info ON animal\_info.user\_id = user\_info.user\_id  
 INNER JOIN rescue\_station\_info ON animal\_info.animal\_rescue\_station\_id = rescue\_station\_info.rescue\_station\_id  
 WHERE  
 animal\_info.animal\_adoption != '未领养'  
  
</select>

对于需要查询小动物信息的管理员，可以使用模糊查询的方式通过小动物的名称或者物种来进行查询。此外，用户还可以点击小动物的详情页面，查看小动物的详细信息，包括小动物的名称、年龄、性别、品种、体重、体长、饲养方法和注意事项等信息。在小动物们界面中，管理员可以通过对小动物进行管理，提供更好的小动物服务和保障。

具体界面如下图4-14所示。

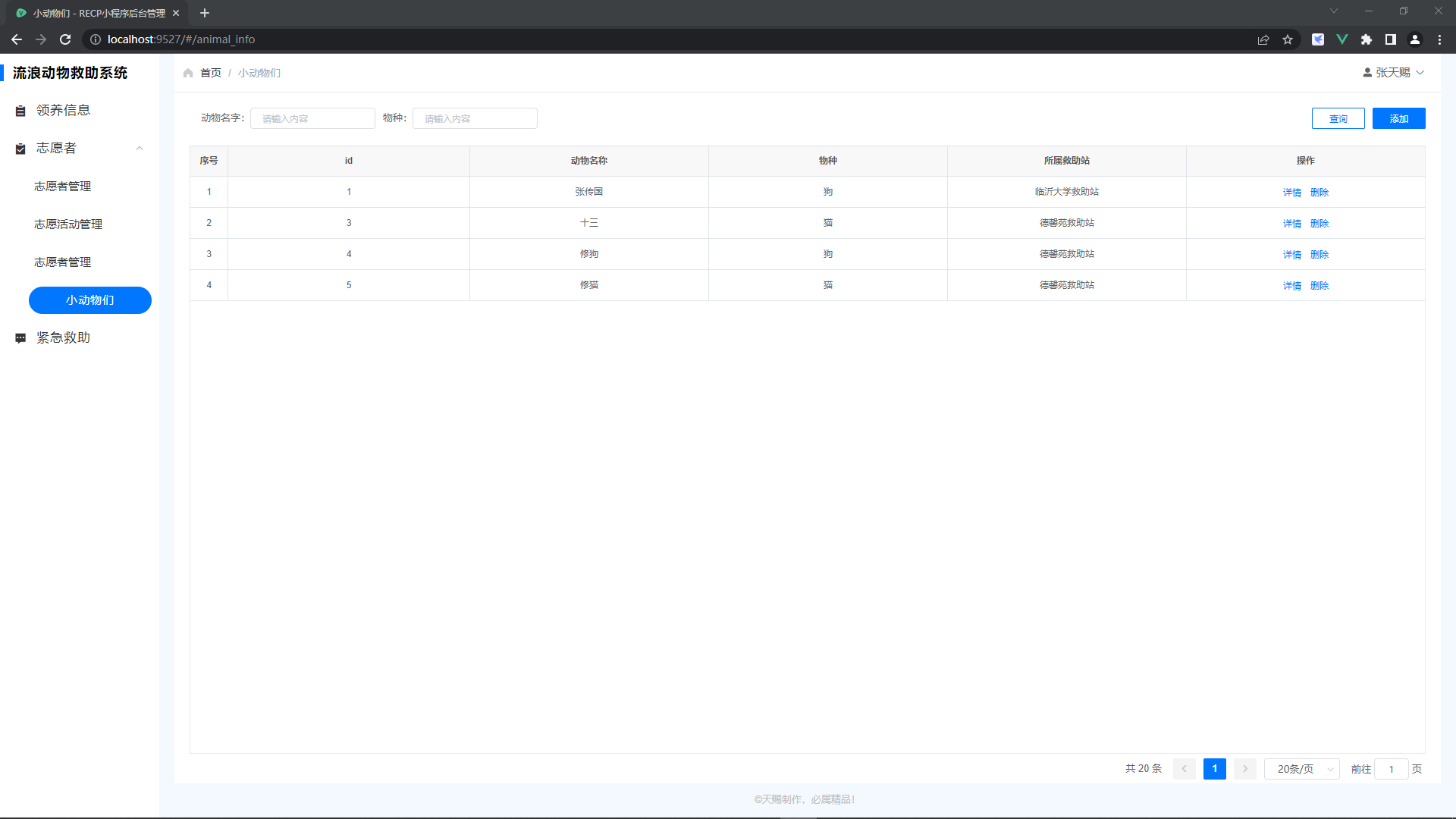


图4-14 小动物们界面

Fig. 4-14 Little animals interface



图4-15 小动物们界面查看详情

Fig. 4-15 Small animals interface for details

4.2.4紧急救助模块实现

紧急救助界面用于实现对紧急救助处理的操作，点击处理会显示用户发布的救助信息，包括时间、地点、现场图片等信息。点击处理后会将处理数据存入数据库中，并将处理结果返回给用户。

界面如下图4-16所示。

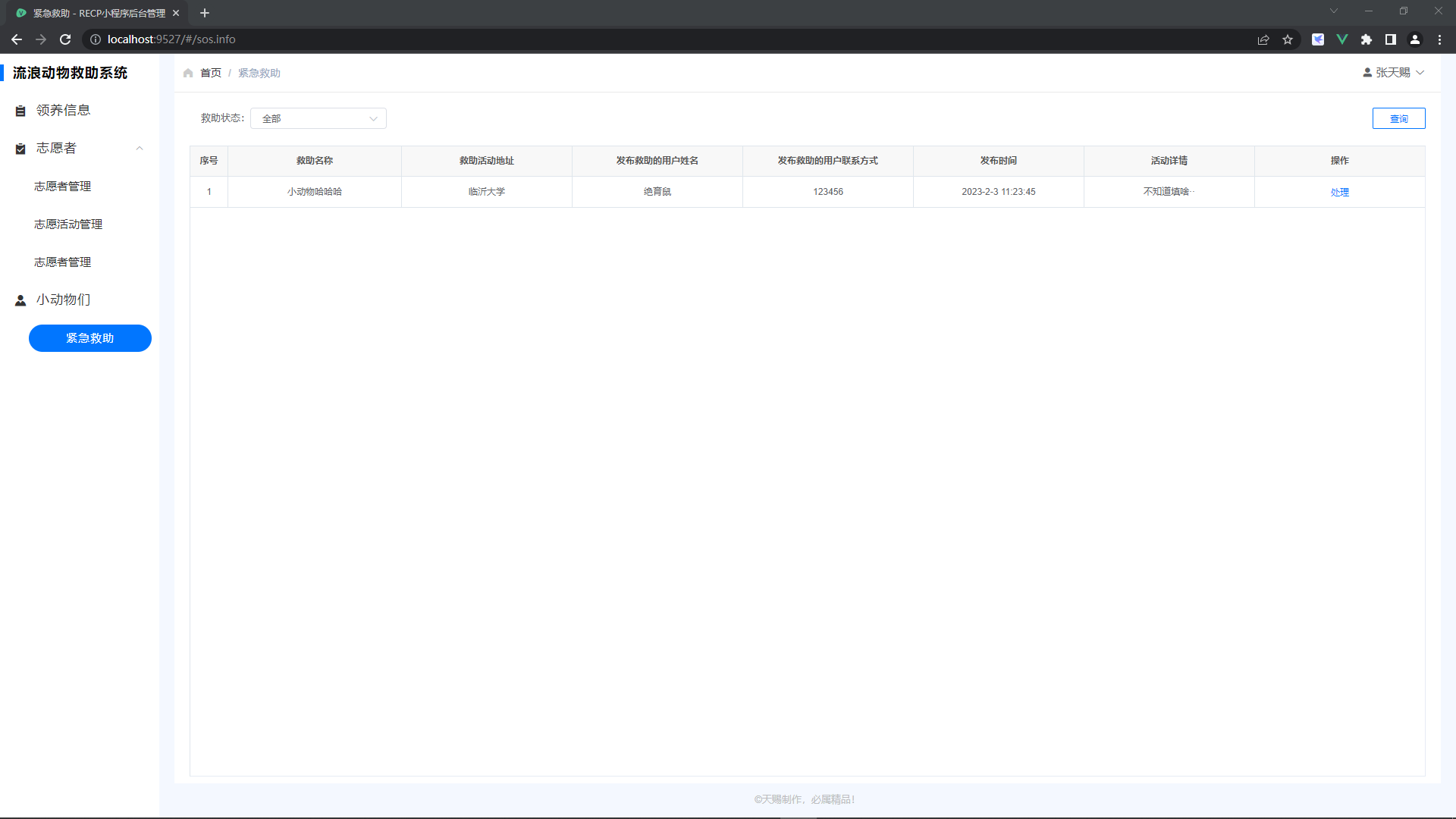


图4-16 紧急救助界面

Fig. 4-16 Emergency rescue interface

4.3图片上传模块实现

当用户或者管理员进行某些需要上传图片的操作时，就要使用本模块来牵线搭桥。系统会将用户或管理员上传的图片发送至系统后台，后台在校验图片后商城遗传唯一的uuid并将图片上传至七牛云照片储存平台，代码实现如下：

JsonResult jsonResult = new JsonResult();  
 String originalFilename = file.getOriginalFilename();  
 //唯一的文件名称 使用UUID  
 String fileName = UUID.randomUUID() +"." + StringUtils.substringAfterLast(originalFilename,".");  
  
 //上传文件 长传到哪呢？ 七牛云  
 boolean upload = qiniuUtils.upload(file, fileName);  
 if(upload){  
 jsonResult.requestSuccess(QiniuUtils.url + fileName);  
 }else {  
 jsonResult.resultFail("上传失败");  
 }  
 return jsonResult;  
 }

上传至七牛云的界面如下图4-17所示。

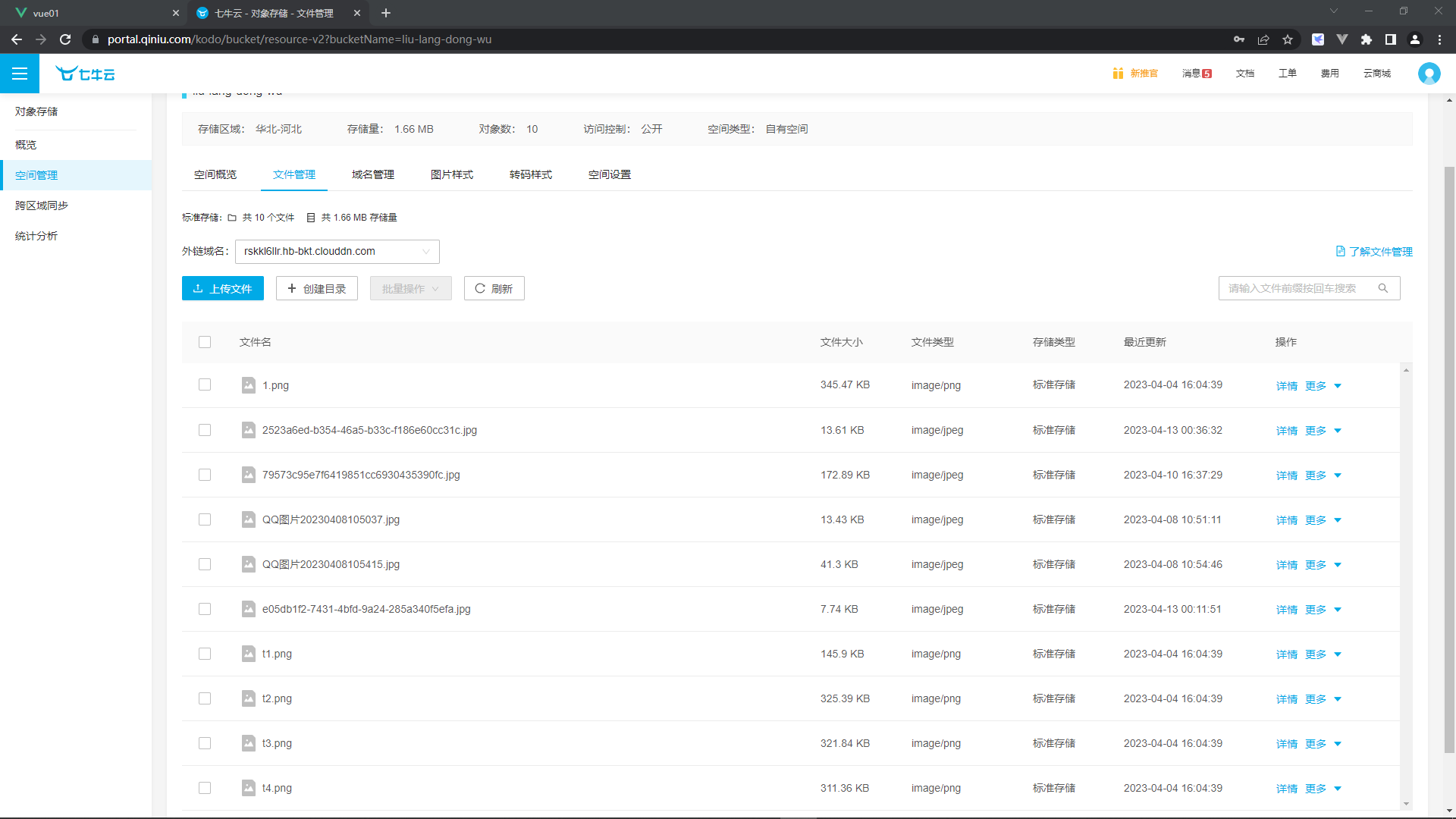


图4-17 七牛云界面

Fig. 4-17 Seven Niuyun interface

上传至七牛云后，后台会返回图片的url给前端，前端再调用相关方法将图片的url存储至数据库中。

5系统测试

本章主要对登陆注册模块、小动物模块、志愿者模块、论坛模块、用户信息管理模块、领养信息管理模块、紧急救助模块进行测试，对于图片上传模块的测试在测试其他模块的过程中就可以测试。

5.1登录注册模块功能测试

登录注册模块主要测试登录和注册时对输入不同信息的应对能力，登录功能的测试用例表如表5-1所示，注册功能的测试用例表如表5-2所示。

**表5-1登录功能测试用例表**

**Tab. 5-1 Log in to the functional test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC001 | **用例名称** | 对系统登录功能测试 | |
| **模块名称** | 登录注册模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对系统登录功能测试，是否可以正常登录系统。 | | | |
| **测试方法** | 错误推测法，等价类 | | | |
| **操作描述** | 1、输入正确账号以及对应正确密码； 2、输入正确账号和错误密码； 3、输入错误账号和正确密码。 | | | |
| **期望结果** | 1、正常登录到系统中，返回跳转页面； 2、提示账号、密码错误； 3、提示账号、密码错误。 | | | |
| **实际结果** | 1、正常登录到系统中，返回跳转页面； 2、提示账号、密码错误； 4、提示账号、密码错误。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

**表5-2注册功能测试用例表**

**Tab. 5-2 Register the functional test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC002 | **用例名称** | 对系统注册功能测试 | |
| **模块名称** | 登录注册模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对系统注册功能测试，是否可以正常账号注册。 | | | |
| **测试方法** | 错误推测法，等价类 | | | |
| **操作描述** | 1. 输入正确信息，点击注册； 2、不输入信息，点击注册；   3、输入已注册过的账号，点击注册。 | | | |
| **期望结果** | 1. 正常注册用户，跳转到登录页面； 2、提示请输入信息；   3、提示该账号已经注册过了。 | | | |
| **实际结果** | 1. 正常注册用户，跳转到登录页面； 2、提示请输入信息；   3、提示该账号已经注册过了。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

5.2小动物模块功能测试

小动物模块测试主要是对用户端的小动物查看、收藏、留言功能；管理员端的增删改查功能进行测试，用户端的测试用例表如表5-3所示，管理员端的测试用例表如表5-4所示。

**表5-3小动物用户端功能测试用例表**

**Tab. 5-3 Small animal client function test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC003 | **用例名称** | 对小动物查看、收藏、留言功能测试 | |
| **模块名称** | 小动物模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对小动物查看、收藏、留言功能测试，是否可以正常对小动物查看、收藏、留言。 | | | |
| **操作描述** | 1. 点开小动物模块，查看小动物； 2、点击喜欢的小动物，查看小动物信息； 3、点击收藏小动物；   4、发布给小动物的留言。 | | | |
| **期望结果** | 1. 成功显示很多个小动物信息； 2、显示点击的小动物信息； 3、成功收藏小动物；   4、成功发布给小动物的留言。 | | | |
| **实际结果** | 1、成功显示很多个小动物信息； 2、显示点击的小动物信息； 3、成功收藏小动物；  4、成功发布给小动物的留言。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

**表5-4小动物管理员端功能测试用例表**

**Tab. 5-4 Small animal administrator side function test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC004 | **用例名称** | 对小动物信息增删改查功能测试 | |
| **模块名称** | 小动物模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对小动物信息增删改查功能测试，是否可以正常对小动物信息增删改查 | | | |
| **操作描述** | 1、输入小动物信息，点击添加； 2、选择小动物信息，点击删除； 3、输入准备修改的小动物信息，点击修改；  4、输入想要查询的小动物信息，点击模糊查询。 | | | |
| **期望结果** | 1、成功添加小动物信息； 2、成功删除小动物信息； 3、成功修改小动物信息；  4、成功模糊查找的想要的小动物信息。 | | | |
| **实际结果** | 1、成功添加小动物信息； 2、成功删除小动物信息； 3、成功修改小动物信息；  4、成功模糊查找的想要的小动物信息。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

5.3志愿者模块功能测试

志愿者模块测试主要是对用户端的志愿者申请、查看个人信息、查看志愿活动、参加志愿活动功能；管理员端的对志愿活动增删改查、处理志愿申请、对志愿者信息增删改查一系列功能进行测试，用户端的测试用例表如表5-5所示，管理员端的测试用例表如表5-6所示。

**表5-5志愿者用户端功能测试用例表**

**Tab. 5-5 Volunteer client functional test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC005 | **用例名称** | 对用户志愿者申请、查看个人信息、查看志愿活动、参加志愿活动功能测试 | |
| **模块名称** | 志愿者模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对用户志愿者申请、查看个人信息、查看志愿活动、参加志愿活动功能测试，是否可以正常对用户志愿者申请、查看个人信息、查看志愿活动、参加志愿活动。 | | | |
| **操作描述** | 1、输入个人信息，点击申请成为志愿者； 2、管理员同意成为志愿者后，查看志愿活动信息； 3、选择想要参加的志愿活动，点击参加；  4、参加若干活动后，查看个人信息。 | | | |
| **期望结果** | 1、申请成功发送； 2、收到申请审批通过消息提示，成功查看到志愿活动信息； 3、成功参加志愿活动，收到参加活动的详细信息消息；  4、成功查看到参加过的志愿活动信息和志愿者信息。 | | | |
| **实际结果** | 1、申请成功发送； 2、收到申请审批通过消息提示，成功查看到志愿活动信息； 3、成功参加志愿活动，收到参加活动的详细信息消息；  4、成功查看到参加过的志愿活动信息和志愿者信息。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

**表5-6志愿者管理员端功能测试用例表**

**Tab. 5-6 Volunteer administrator side functional test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC006 | **用例名称** | 对志愿活动增删改查、处理志愿申请、对志愿者信息增删改查功能测试 | |
| **模块名称** | 志愿者模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对志愿活动增删改查、处理志愿申请、对志愿者信息增删改查功能测试，是否可以正常对志愿活动增删改查、处理志愿申请、对志愿者信息增删改查。 | | | |
| **操作描述** | 1、输入志愿活动信息，点击添加； 2、选择志愿活动信息，点击删除； 3、输入准备修改的志愿活动信息，点击修改；  4、输入想要查询的志愿活动信息，点击模糊查询；  5、输入志愿者信息，点击添加；  6、选择志愿者信息，点击删除；  7、输入准备修改的志愿者信息，点击修改；  8、输入想要查询的志愿者信息，点击模糊查询；  9、查看志愿申请；  10、拒绝志愿申请，并输入拒绝理由；  11、同意成为志愿者。 | | | |
| **期望结果** | 1、成功添加志愿活动信息； 2、成功删除志愿活动信息； 3、成功修改志愿活动信息；  4、成功模糊查询到志愿活动信息；  5、成功添加者活动信息；  6、成功删除志愿者信息；  7、成功修改志愿者信息；  8、成功模糊查询到志愿者信息；  9、成功查看志愿申请；  10、成功拒绝志愿申请，并发送拒绝理由；  11、成功同意成为志愿者，申请用户成为了志愿者。 | | | |
| **实际结果** | 1、成功添加志愿活动信息； 2、成功删除志愿活动信息； 3、成功修改志愿活动信息；  4、成功模糊查询到志愿活动信息；  5、成功添加者活动信息；  6、成功删除志愿者信息；  7、成功修改志愿者信息；  8、成功模糊查询到志愿者信息；  9、成功查看志愿申请；  10、成功拒绝志愿申请，并发送拒绝理由；  11、成功同意成为志愿者，申请用户成为了志愿者。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

5.4论坛模块功能测试

论坛模块测试主要是对浏览帖子、查看帖子详情、点赞收藏帖子、发布评论、发布评论的回复功能进行测试，测试用例表如表5-7所示。

**表5-7论坛模块功能测试用例表**

**Tab. 5-7 Forum module function test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC007 | **用例名称** | 对浏览帖子、查看帖子详情、点赞收藏帖子、发布评论、发布评论的回复功能进行测试 | |
| **模块名称** | 论坛模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对浏览帖子、查看帖子详情、点赞收藏帖子、发布评论、发布评论的回复功能进行测试，是否可以正常对浏览帖子、查看帖子详情、点赞收藏帖子、发布评论、发布评论的回复。 | | | |
| **操作描述** | 1、点击论坛模块，浏览帖子； 2、选择感兴趣的帖子，点击查看帖子详情； 3、给帖子点赞+收藏；  4、发布帖子评论；  5、给发布评论的人回复。 | | | |
| **期望结果** | 1、成功查看到帖子； 2、成功查看到帖子详情； 3、成功给帖子点赞+收藏；  4、成功发布帖子评论；  5、成功给发布评论的人回复。 | | | |
| **实际结果** | 1、成功查看到帖子； 2、成功查看到帖子详情； 3、成功给帖子点赞+收藏；  4、成功发布帖子评论；  5、成功给发布评论的人回复。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

5.5用户信息管理模块功能测试

用户信息管理模块测试主要是对查看个人信息、点击查看他人的个人信息、修改个人信息功能进行测试，测试用例表如表5-8所示。

**表5-8用户信息管理模块功能测试用例表**

**Tab. 5-8 User information management module function test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC008 | **用例名称** | 对查看个人信息、点击查看他人的个人信息、修改个人信息功能进行测试 | |
| **模块名称** | 用户信息管理模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对查看个人信息、点击查看他人的个人信息、修改个人信息功能进行测试，是否可以正常对查看个人信息、点击查看他人的个人信息、修改个人信息处理。 | | | |
| **操作描述** | 1、点击用户信息管理模块，查看个人信息； 2、上传修改后的个人信息，点击修改； 3、在浏览论坛时，点击他人的头像或网名。 | | | |
| **期望结果** | 1、成功查看个人信息； 2、成功修改后的个人信息； 3、成功查看他人的个人信息。 | | | |
| **实际结果** | 1、成功查看个人信息； 2、成功修改后的个人信息； 3、成功查看他人的个人信息。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

5.6领养信息管理模块功能测试

领养信息管理模块测试主要是对处理领养申请功能进行测试，测试用例表如表5-9所示。

**表5-9领养信息管理模块功能测试用例表**

**Tab. 5-9 Adoption information management module function test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC009 | **用例名称** | 对处理领养申请功能进行测试 | |
| **模块名称** | 领养信息管理模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对对处理领养申请功能进行测试，是否可以正常处理领养申请。 | | | |
| **操作描述** | 1、点击领养信息管理模块，查看领养申请信息； 2、选择处理状态，查看未处理申请； 3、点击同意领养申请。 | | | |
| **期望结果** | 1、成功查看领养申请信息； 2、成功未处理领养申请信息； 3、成功审批领养申请、发送通知给申请人。 | | | |
| **实际结果** | 1、成功查看领养申请信息； 2、成功未处理领养申请信息； 3、成功审批领养申请、发送通知给申请人。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

5.7紧急救助模块功能测试

紧急救助模块测试主要是对处理紧急救助信息功能进行测试，测试用例表如表5-10所示。

**表5-10紧急救助模块功能测试用例表**

**Tab. 5-10 Emergency rescue module functional test case table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | TC009 | **用例名称** | 对处理紧急救助信息功能进行测试 | |
| **模块名称** | 紧急救助模块 | **编制人** | 张天赐 | |
| **编制时间** | 2022.04.02 | **修改历史** |  | |
| **测试目的** | 对处理紧急救助信息功能进行测试，是否可以正常处理紧急救助。 | | | |
| **操作描述** | 1. 点击紧急救助模块，查看紧急救助信息； 2、选择未处理紧急救助，查看详细信息； 3、点击处理紧急救助；   4、将紧急救助作为志愿活动发布。 | | | |
| **期望结果** | 1. 成功查看紧急救助信息； 2、成功查看未处理紧急救助详细信息； 3、成功处理紧急救助、并发送通知给发布人；   4、成功将紧急救助作为志愿活动发布。 | | | |
| **实际结果** | 1. 成功查看紧急救助信息； 2、成功查看未处理紧急救助详细信息； 3、成功处理紧急救助、并发送通知给发布人；   4、成功将紧急救助作为志愿活动发布。 | | | |
| **测试人员** | 张天赐 | **开发人员** | **测试日期** | 2022.04.02 |

6总结

在开发流浪动物救助系统之前，我们花费了大量时间进行了充分的准备。我们阅读了大量的文献资料，深入了解了该领域的最新技术和最佳实践，这使我们在设计和开发过程中有了更好的方向和思路。通过这些准备工作，我们积累了宝贵的经验和技能，使得在实际的项目开发中能够更加自信和高效地应对挑战。

在技术选型方面，我们选择了一些自己比较熟悉的技术，如Vue、SpringBoot和MySQL等。这些技术是我们在以前的学习和工作中广泛掌握的，并在该项目中得到了进一步的应用和实践。当然，在开发过程中，我们也学到了许多新的技术和工具，不断地扩展和更新我们的技能树，以便更好地应对未来的挑战。

在这个项目中，我们也受益于老师和同学们的指导和支持。老师们提供了有价值的建议和反馈，帮助我们解决了许多问题。同学们之间也相互协作和交流，分享经验和知识，共同推进项目的进展。在这个过程中，我们学到了很多东西，不仅是关于技术方面的，还包括团队合作和项目管理等方面的经验。

当然，该系统还有许多可以改进和完善的地方。我们会继续关注用户的反馈和需求，不断优化和升级系统的功能和性能。我们相信，随着我们的经验和技能的不断提高，这个系统将变得更加完美和可靠，为流浪动物救助事业做出更大的贡献。

# 参考文献

[1]刘在英,杨磊. 一种基于SpringBoot框架的企业进存销系统的设计方法[P]. 山东省：CN114003204A,2022-02-01.

[2]熊柏祥.基于Springboot和Vue框架的考试资源服务平台的设计与实现[J].信息与电脑(理论版),2022,34(01):97-99+103.

[3]霍福华,韩慧.基于SpringBoot微服务架构下前后端分离的MVVM模型[J].电子技术与软件工程,2022(01):73-76.

[4]张思璇,那艳平,孙丽娆, 3 环境艺术篇　Rubiks——流浪动物救助站设计. 李杰 主编,中国高等院校设计作品精选年鉴,中国商业出版社,2021,374,年鉴.

[5]曹明昊. 基于SpringBoot和Vue框架的邯郸市现代农业园区信息管理系统的研发[D].河北工程大学,2021.

[6]单树倩,任佳勋.基于SpringBoot和Vue框架的数据库原理网站设计与实现[J].电脑知识与技术,2021,17(30):40-41+50.

[7]苏文强,刘紫阳,苗玉恒.基于SpringBoot的缺陷管理系统设计与实现[J].大众科技,2021,23(10):4-6+10.

[8]张画仪,王楚桐,郑鑫楠,袁浩育,谢璠.法律体制下高校流浪动物救助工作的现状及管理措施分析[J].法制博览,2021(23):24-25.

[9]Fuyuan Cheng. Talent Recruitment Management System for Small and Micro Enterprises Based on Springboot Framework[J]. Advances in Educational Technology and Psychology,2021,5(2).

[10]刘欣,李亮亮,牛聪. 基于Vue和SpringBoot框架的流域监管平台的研究和应用[C]//.第十一届防汛抗旱信息化论坛论文集.,2021:118-122.

[11]危炜. 基于SpringBoot和Vue框架的刹车片工厂流程再造系统的设计与实现[D].浙江理工大学,2021.

[12]林翔. 基于SpringBoot技术的M公司厨电产品质量管理系统的设计与实现[D].电子科技大学,2021.

[13]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.

[14]季琛其. 一种动物小型庇护救助站[P]. 浙江省：CN212212264U,2020-12-25.

[15]Guanhong Chen,Jiangming Xu. Design and implementation of efficient Learning platform based on SpringBoot Framework[J]. Journal of Electronics and Information Science,2020,6(1).

[16]Jian Chen,Chen Jian,Pan Hailan. Design of Man Hour Management Information System on SpringBoot Framework[J]. Journal of Physics: Conference Series,2020,1646(1).

[17]陈薇,刘梓媛,尹婷.新媒体背景下流浪动物救助微信公众号运营研究[J].现代营销(下旬刊),2020(05):85-86.

[18]纪好. 流浪动物救助公益服务设计研究[D].武汉理工大学,2018.

[19]王波,同雯,王梓昕,李子杰.基于APPCAN的流浪动物救助APP系统设计[J].电脑知识与技术,2017,13(36):76-77+79.

# 致谢

随着时间的流逝，大学生活即将结束。回首往事，面对离别和结局，我不禁伤感和彷徨。我还记得一个秋日，我怀着热切的期待来到大学。站在开放的校门前，我在想这是不是我新生活的起点。欢迎仪式很热烈，那天阳光明媚。当我面对着微笑的同学们时，感觉就像一缕阳光照进了我的心里。那天我还遇到了我可爱的室友，我们一起度过了难忘的两年。

但现在，当我看到这篇论文的致谢时，我不禁问自己:是时候说再见了吗?

大学生活带给我的记忆永远不会离开我。时间过得太快了，我觉得我还没有完全欣赏到这个校园的美丽。她给我的回忆将永远伴随着我。也许在我真正离开的那一天，我的眼睛会充满泪水。不是因为我难过，而是因为我想把她的形象印在眼泪里，刻在心里。最后，我要感谢我的老师教会我做人的道理和知识。我要感谢我的室友陪了我这么久。我还要感谢所有关心和支持我的人。

感谢母校临沂大学为我提供了一个探索知识的平台，让我从一个懵懂无知的少年成长为一个有专业知识和社会经验的人。沂蒙地区的人民淳朴、善良、无私。每时每刻，我都更爱这片土地。

首先，我要感谢我的导师韩丽花老师。韩老师学识渊博，一丝不苟。作为我的学术导师，她对提供我无微不至的关怀和帮助，并提醒我大学生活是宝贵的，帮助我规划未来。在我完成毕业论文的过程中，韩老师给予了我细心的指导和帮助。虽然老师课务繁重，但还是抽出时间来审阅我的论文，并提出修改意见，让我顺利完成了论文。韩老师积极的生活态度和细致的工作精神，将是我一生学习和实践的榜样。

其次，我要感谢我的辅导员唐霞老师。大学生活的圆满结束，离不开唐老师无微不至的关怀和关注。唐老师对我的生活产生了重大影响，她经常和我们交谈，耐心地帮助我解开困惑，克服挫折。唐老师不仅是我的老师，也是我的朋友，从她身上我学到了很多，尤其是面对逆境时的毅力和乐观精神。

当我告别我的大学生活，我意识到在这个世界上没有永远的朋友或敌人，只有永远的回忆。我会珍惜大学生活带给我的回忆和经历，并将它们带在我的人生道路上。前面的道路也许漫长而曲折，但我已经准备好用我在临沂大学获得的知识和经验去面对它。

张天赐

2023年4月