

摘要

随着现代社会中流浪动物问题的愈演愈烈与广泛性动物福利意识的觉醒，一些热心人士组建了各种大大小小的动物救助机构，用自己微薄的力量贡献出一份爱心。然而，因为受物质和人力的限制，动物救助机构的能力有限。需要切实可行的方法提高动物的收养率，加快工作效率。流浪动物救助中心的需求较为确定，流程较为明了，适合于开发对应的信息系统，以加快领养流程，缓解相关人员的工作压力。

流浪动物救助管理系统是基于 web 开发的领养管理系统。相关的动物救助组织可以借助该系统，管理审核注册用户、动物、用户提交的领养申请、义工申请以及维护领养关系，普通用户可以查看待领养的动物的相关资料，提交领养申请、义工申请同时查看救助组织提供的公示信息。

关键词：信息系统，流浪动物，领养管理，spring boot

目 录

一、绪论.....	1
(一) 研究背景及意义.....	1
(二) 研究内容和组织结构.....	1
1. 论文研究内容.....	1
2. 论文组织结构.....	2
二、系统需求分析.....	2
(一) 系统目标.....	2
(二) 功能性需求分析.....	3
(三) 非功能性需求分析.....	4
(四) 需求建模.....	5
三、系统设计.....	5
(一) 架构设计.....	5
(二) 功能模块设计.....	6
1. 领养申请管理模块.....	6
1.1 动物管理子模块.....	6
1.2 领养申请子模块.....	6
2. 领养关系维护模块.....	8
2.1. 凭证管理子模块.....	8
2.2 回访管理子模块.....	8
3. 系统管理模块.....	9
3.1 用户管理子模块.....	9
3.2 角色权限管理.....	9
4. 义工申请模块.....	9
5. 信息公示管理模块.....	10
(三) 数据库设计.....	10
四、系统实现.....	14
(一) 系统架构.....	15
(二) 领养管理模块.....	16
2.1 动物管理子模块.....	16
3.2 领养申请管理子模块.....	18
(三) 领养关系维护模块.....	19

3.1 凭证上传子模块	19
3.2 回访管理子模块	20
(四) 系统管理模块	21
(五) 信息公示模块	22
5.1 公告管理子模块	错误!未定义书签。
5.1 资金公示管理子模块	错误!未定义书签。
(六) 义工申请模块	22
五、结论	24
(一) 技术总结	24
(二) 系统展望	24
参考文献	26

一、绪论

本章首先介绍设计与实现基于 web 的流浪动物救助领养管理系统的背景，以及本系统的使用在流浪动物救助领养相关活动中的意义，同时简明扼要地介绍本论文的主要内容以及内在结构。本章为本系统的设计与开发提供了理论依据与现状支持。

(一) 研究背景及意义

随着城市化的进程不断加快，一系列社会问题出现在人们的视野中，其中就包括流浪动物问题。所谓流浪动物，主要是指分布在城镇、乡村等人类聚居地的无主动物，主要由猫和狗组成。流浪动物不仅自身的安全与健康方面得不到保证，同时也给城市的公共卫生、交通、治安、生态等领域带来一定程度的隐患。如何在保证动物福利不受或者尽可能少受侵害的前提下减少流浪动物数量，一直是人类社会的重要难题。解决该类问题的思路一般分为两种：第一、从源头减少流浪动物的出现可能，通过加强相关的社会宣传教育、制定对应的规范与法律等方式，防范宠物转化为流浪动物现象的出现。但国内在相关问题上，尚缺乏系统有效的解决方案。第二、对于现有的流浪动物，应使之逐步可控化。所对应的救助措施包括且不限于：推动流浪动物绝育、进行集中收容、提高动物领养率等。目前，城市中有组织性的流浪动物救助行为主要由民间流浪动物救助集合完成。然而由于救助组织自身的一些内在特点，救助行为的效果和可持续性都有待改善。

城市中的流浪动物救助组织主要有以下特征：一，人员构成简单，且流动性较大，以志愿者为主。二，组织为纯公益性质，不涉及商业行为。三，参与救助的全流程，从救助源头的教育计划到救助尾端的领养。四，组织无稳定资金来源，主要依靠社会募捐等爱心行为。^[1]流浪动物救助组织在献出爱心的同时，也面临着一系列的生存挑战：资金匮乏，难以持久承担动物相关的费用，包括疫苗、绝育、生活、动物健康检查等；义工大多身份为非全职的爱心人士，能够稳定参与救助的人手不足；在流浪动物数量增加趋势不断显著的同时，救助机构受限于容量和能力有限，无法回应所有的救助请求，面临着一定的社会舆论压力。^[2]

为了减轻压力和持续生存，必须提高流浪动物救助组织的领养率。而在目前常见的领养流程中，存在以下问题：领养者之前缺乏事先对动物的了解，在挑选心仪的动物时消耗较多的时间与精力；^[3]领养流程较为繁琐复杂，消耗双方大量时间与精力；组织与领养人之间缺乏沟通，造成信任不足，影响领养后期的回访以及身份信息交换等重要工作。本文希望以设计和开发相关系统的方式，达到减少相关人员的工作量，加快领养流程的目的，因此将流浪动物救助领养系统作为本文的研究课题。

本文在结合流浪动物救助领养现状和现有领养流程的基础上，为流浪动物救助组织设计相应的救助领养管理系统并优化相应流程，以加快流浪动物救助领养工作的效率。本系统适用于相关的救助组织以及普通大众用户。

本系统基于 B/S 架构，以 spring boot+springMVC+mybatis plus 作为底层框架，相关数据存储在 MySQL 数据库中。本系统不仅支持相关动物救助组织将其作为基本的管理系统，也可以供有意领养和参与救助的爱心人士使用，注册登录后可以完成领养申请、义工申请等多种功能。

(二) 研究内容和组织结构

1. 论文研究内容

流浪动物救助领养管理系统专门为流浪动物救助组织服务，通过对流浪动物救助组织的生存现状和领养流程进行分析和研究，整理总结了相关需求。本系统集成领养申请管理、领养关系维护、义工申请管理、信息公示等多个功能模块，使之成为功能完整、流程清晰，能够满足日常管理领养

事务需求的领养管理平台。该流浪动物救助管理系统的使用者包括流浪动物救助组织的相关志愿者、有意领养和加入救助行动的相关爱心人士以及系统的管理员等。

本流浪动物救助管理系统后端采用基于 springboot+springMVC+mybatis plus 的框架结构。Springboot 是由 Pivotal 团队贡献的基于 Java 语言的服务端开发框架,在 spring 框架的基础上通过简化配置的方式,比之进一步降低应用的搭建和开发工作量。本系统使用 SpringMVC 框架作为控制层,前端选择使用 Vue.js 框架,界面中的组件选择使用 elemnt UI,减少前端开发调试工作量。数据库方面则选用了 MySQL。经过分析与总结,本文最终设计并实现了一个基于 web 的流浪动物救助领养管理系统。

2. 论文组织结构

第一章为绪论部分,提出了研究本课题的契机以及相关背景。同时,对国内相关的动物救助组织面临的困境以及解决方案进行了探索与分析,借此提出了设计与开发本流浪动物救助领养管理系统的主要目的

第二章为需求分析部分。对相关流浪动物救助组织现有的领养管理流程进行分析,拆分相关需求,列举功能模块。

第三章为系统设计部分。对第二章中提出的功能模块进行进一步的细化,对子模块进行详尽设计。同时,绘制系统 ER 图,并且将实体转化为对应的数据库表。

第四章为系统实现部分。分析了系统实现选用的架构并且解释选用相关技术的理由。其次,展示功能模块的具体实现样式。

第五章为结论部分。对现在已经实现的工程进行总结,列举关键技术点。对当前项目中存在的瑕疵进行分析,对后续可能的改良进行了展望。

二、系统需求分析

(一) 系统目标

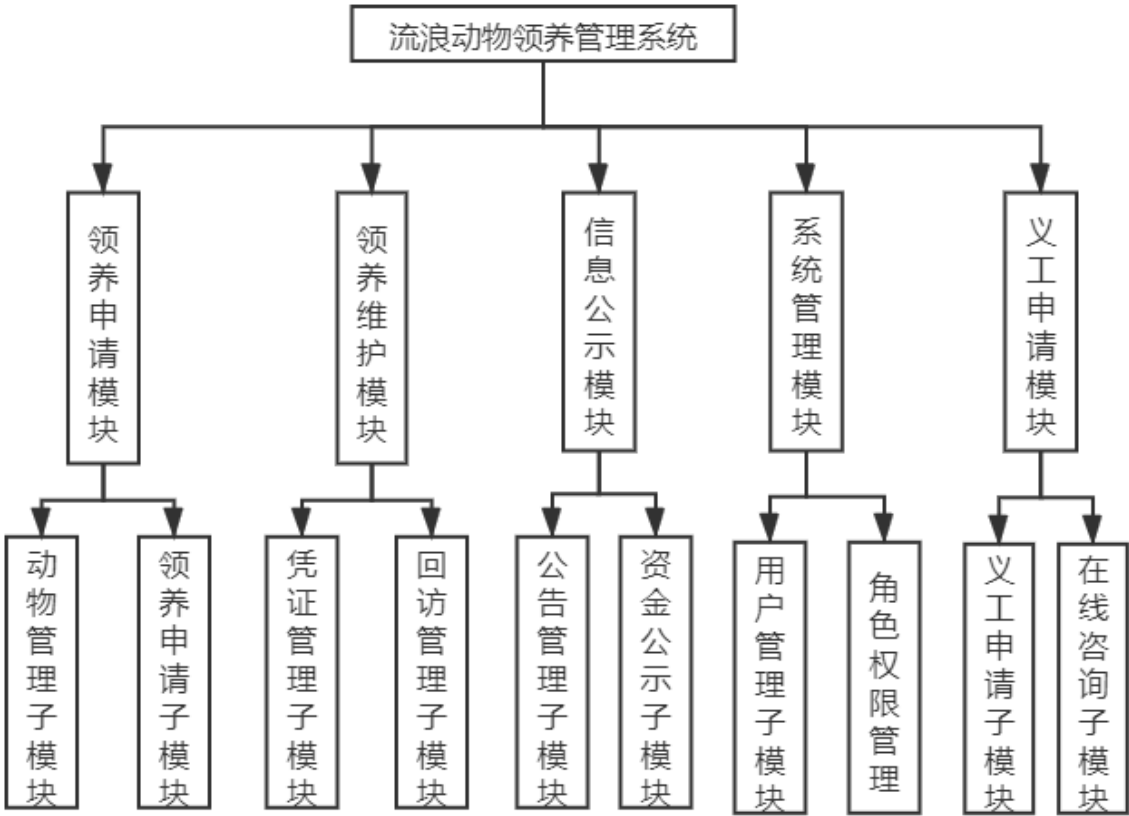
随着社会文明程度的增加,解决流浪动物问题的紧迫性逐渐增加。国内外现存的解决思路主要分为两种:从源头上发力,减少流浪动物的出现,抑制流浪动物的数量增长;对于已经出现的流浪动物,加强集中管制收容,同时推动领养。对于后者来说,现有的执行主要依赖于动物救助机构。但是目前国内的相关组织机构多由民间自发组织成立,受资金以及人力限制,能力有限。因此,比起广泛救助,控制救助组织内现有动物数量,推动动物领养以及关注被领养动物的后续生活有着更重要的意义。

随着信息时代的发展,一些相关机构开始借助互联网的方式来推动领养工作,如成都爱之家领养救助中心、南京平安阿福流浪动物救助中心等。其现有的在线领养模式中,存在以下问题:其现有的领养信息主要依赖于微博、微信公众号、贴吧等主社交平台发布,信息缺乏整合,不方便领养者进行查看;领养的对接与后续关系维护依赖于志愿者以及社群进行,效率较低,且不利于相关记录的整理与维护;救助组织人员流动性较强,在需要进行交接时信息杂乱繁多,影响相关工作效率。本系统的目标是设计一个适用于中小型流浪动物救助组织以及有意领养和参与救助活动的普通爱心人士使用的领养管理系统,且具有以下特征:

1. 实现了动物领养、领养关系维护、义工申请、信息公示等核心功能,功能齐全。能够有效整合领养过程中产生的各种信息,便于管理和维护,减轻相关志愿者的工作压力。
2. 合理的权限和用户管理机制。本系统角色控制较为灵活,方便调节。根据需要可以新增角色和为角色分配相应的权限。
3. 交互界面简洁明了,便于操作,在保证功能性的同时兼顾美观,体现对于使用者的关怀。

(二) 功能性需求分析

首先，梳理功能性需求，并且按照需求划分模块。本流浪动物救助领养管理系统主要由领养申请模块、领养维护模块、信息公示模块、义工申请模块、系统管理模块。依据每个模块的具体要求和特点进行细分设计。领养申请模块内含动物管理子模块和领养申请子模块两个子模块，领养维护模块下包括凭证管理子模块和回访管理子模块两个子模块，其余模块描述从略。对应的功能结构图如下图所示。



流浪动物领养管理系统

流浪动物救助领养管理系统主要面向三类用户：普通用户、为机构服务的志愿者、超级管理员。所有的用户必须登录到本系统才可以使用。普通用户指的是非救助中心内部人员的一般用户，可以使用在线咨询、发布领养申请、义工申请以及上传领养后相关凭证（狗证、绝育证明、疫苗接种证明等）等功能。志愿者指的是使用系统管理日常事务的志愿者，可以以志愿者身份登录系统，并且对动物信息、领养申请、回访信息、通知信息、义工申请，公示信息等信息进行管理、登记、审核等操作以及查询统计。超级管理员指的是主要是系统的管理员以及维护人员，日常需要对系统用户进行管理，以及对其角色和权限进行分配。

下面按模块对功能一一介绍。

1. 领养申请模块

领养申请模块包括动物信息管理和领养申请管理两个子模块。用户可以查看动物的资料卡，对待领养的动物进行了解。并且选择自己心仪的动物，向流浪动物救助中心提交领养申请。展示的动物信息都是由志愿者统一发布。志愿者可以根据用户提交的领养申请中填写的信息，对领养人进行

初步了解，方便和领养人进行进一步的联络与沟通。在确定最终领养人后，审核通过领养申请，将领养申请转化为领养关系。

2. 领养维护模块

领养维护模块包括凭证管理和回访管理两个子模块。在领养申请通过审核后，领养关系即成立。为了保障动物的生命健康安全，领养人需要向流浪动物救助中心提交身份证件以及后续动物的一些相关凭证，包括狗证、绝育证明、疫苗接种证明等。同时，相关的工作人员也会对动物去向家庭进行回访，查看动物生活状态，录入回访记录。凭证管理和回访管理即对以上信息进行增删改查等相关操作。

3. 系统管理模块

本系统的系统管理需求主要分为用户管理和角色分配两部分，对应划分为两个子模块。用户管理子模块主要包括用户注册、用户登录、用户对自己个人信息的修改、管理员对用户信息的增删改查等功能。角色权限管理子模块主要管理系统中存在的主要角色，并且为角色分配对应的权限与功能，控制角色的视图。

4. 义工申请模块

义工申请模块包括义工申请子模块以及在线咨询模块。想要参与动物救助的用户可以向流浪动物救助组织提交义工申请。志愿者可以根据义工申请中的信息对义工申请进行审核。用户可以通过在线咨询模块，与志愿者进行实时聊天，咨询了解关于领养、救助、义工申请等相关事宜。

5. 信息公示模块

信息公示模块包括通知信息管理和资金公示管理两个子模块。为了向用户普及领养救助相关知识，志愿者可以通过公告信息管理模块上传公示信息。为了增加组织的透明度和公信力，首页设置了资金公示图表。志愿者可以通过资金公示模块管理用于展示的相关数据。

(三) 非功能性需求分析

在流浪动物救助领养管理系统的相关需求进行分析时，为了能够较长时间对组织进行服务，还要对系统的非功能需求进行分析。

1. 系统安全性需求

因为本系统涉及到领养申请与义工申请等功能，涉及到一定的个人隐私，因此对于系统的安全提出了要求。本系统采用合理的权限管理机制，保证用户只能执行符合自己拥有权限的操作，防止出现越权行为。同时，采取在登陆界面放置验证码，使用 `restful` 风格的 `api` 设计，并且配置相关的表单过滤器等措施，来有效避免 `xss` 攻击，一定程度上维护系统安全。

2. 系统健壮性需求

本系统内含有大量表单，更应该关注系统的健壮性。对于可能的非法请求、非法数据输入以及系统异常，应该提供良好的应对措施。对相关参数设置合理的默认值，设置多重前端表单验证等方式可以在保证系统健壮性的同时，减轻服务端压力。

3. 系统可扩展性需求

在开发和设计系统的时候，不可以忽略的是系统的可扩展性。应保证在后期需求增加需要添加功能模块时，不过多影响已编写的程序。因此，应统一系统采用的各类接口，维护数据的一致性与完整性，减少开发成本。

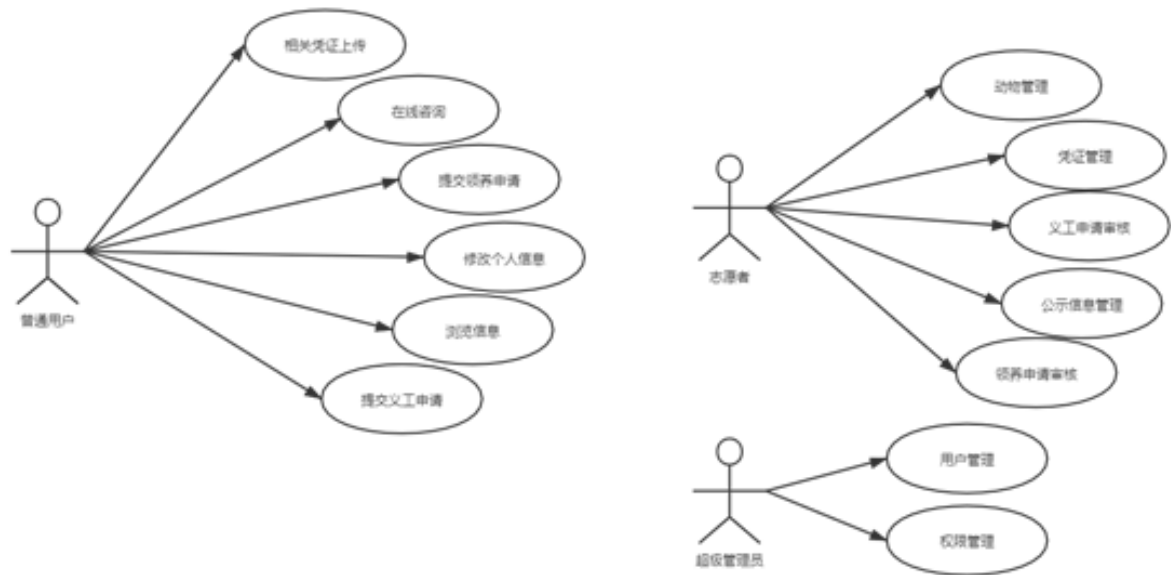
4. 用户界面需求

在设计界面时，应注重对于用户的关怀。界面应简洁规范，排版简单明了，便于用户操作。功能模块排列有序，一目了然。界面应风格统一，同一模块的图片、文字风格应尽量相近，同时突出重

点。界面设计应人性化，重要图标应配有文字，且图标明显易懂；文字、图标、图片应采用适合客户端查看的规格。

(四) 需求建模

在流浪动物救助管理系统中存在三类用户：普通用户、执行日常事务的志愿者以及超级管理员。以下是根据角色划分和用户的不同需求绘制的系统用例图。



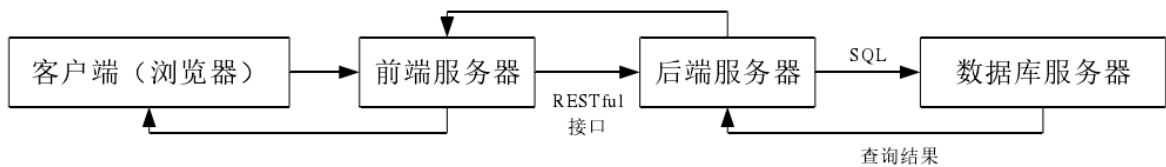
系统用例图

三、系统设计

(一) 架构设计

本系统采用前后端分离的模式进行开发，使用 restful 风格的 api 进行前端和服务端的通信和数据传输。前端通过 ajax 调用后端提供的 api，以 json 的方式请求相关数据，这样后端开发人员不需要关注前端，只需要提供符合规范的接口，前端相关开发也不需要依赖后端进行，将前后端进行了一定程度的解耦。^[10]

系统数据库选用的是 MYSQL。在当前业务活动产生的数据量较小的条件下，可以选择和前后端部署在同一服务器上。



3-1 系统层次架构图

(二) 功能模块设计

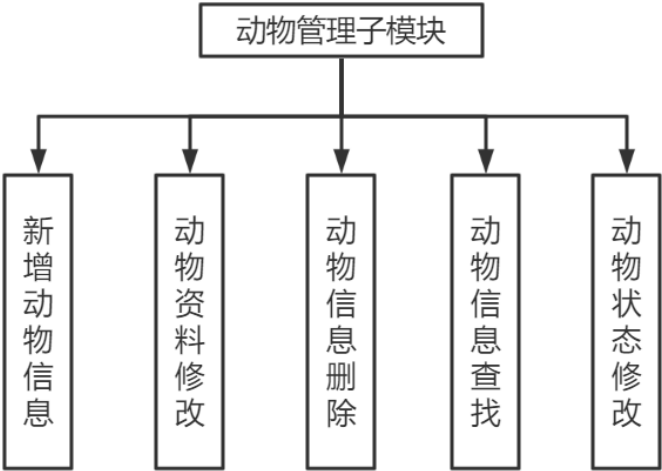
根据在 2.2 节中的功能性需求分析中进行的模块划分，系统被划分为领养申请、领养维护、信息公示、义工申请、系统管理等五个部分。以下对其中的重要组成部分进行详细介绍。

1. 领养申请管理模块

根据前文 2.2 节中的模块划分，领养申请模块又可以细分为动物管理子模块以及领养申请子模块。

1.1 动物管理子模块

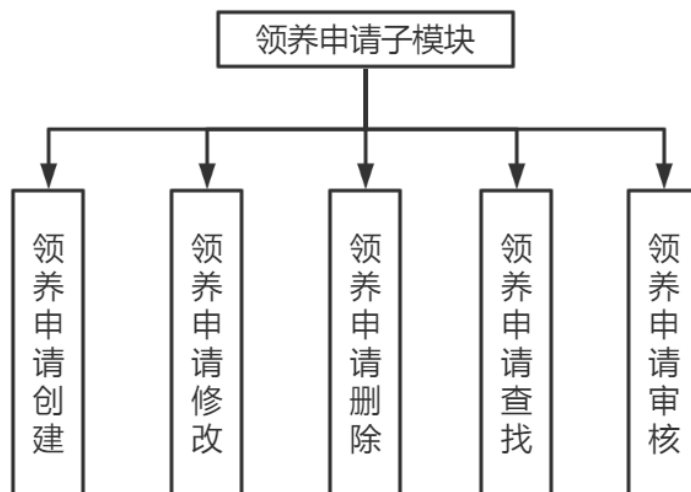
动物管理子模块用于管理系统中可供领养的动物信息。志愿者以列表的形式管理在动物资料卡上显示的内容，包括领养者较为感兴趣的动物的年龄、性别、品种、年龄以及一些详细介绍。动物管理子模块包括对于动物资料信息的增删改查等基础功能。动物状态由志愿者在系统中进行调整，分为可以领养、领养申请已通过尚在基地以及动物已经抵达新家庭需要回访等三种状态。动物状态修改关系到动物是否在系统中展示，以及是否需要进行后续的回访。动物管理子模块结构图如 3-2 所示。



3-2 动物管理子模块结构图

1.2 领养申请子模块

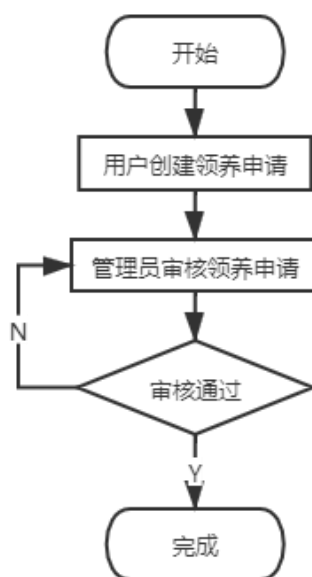
领养管理子模块是本系统内最重要最关键的模块之一，关系到领养关系的建立以及后续领养关系的维护。领养申请子模块包括领养申请创建、领养申请修改、领养申请删除、领养申请查找、领养申请审核等功能。领养审核通过后，相关记录将转化为领养关系。相关结构图如 3-3 所示。



3-3 领养管理子模块结构图

领养申请管理模块是由普通用户和相关志愿者共同使用的功能模块。其中，领养申请的审核是志愿者独有的功能。一旦志愿者审核通过了领养申请，用户将无法修改、删除该条申请。用户可以在领养申请表单中填写所需的个人信息，并且提交给组织进行审核。同一个用户应只能对一只动物提出唯一一次申请。这样确保在转化为领养关系时不会发生错乱。在申请并未通过时，用户拥有编辑申请以及删除申请的权利。一旦申请通过审核，领养申请转化为领养关系，用户此时不能再变更领养申请。

领养申请管理子模块的基本流程设计如下：



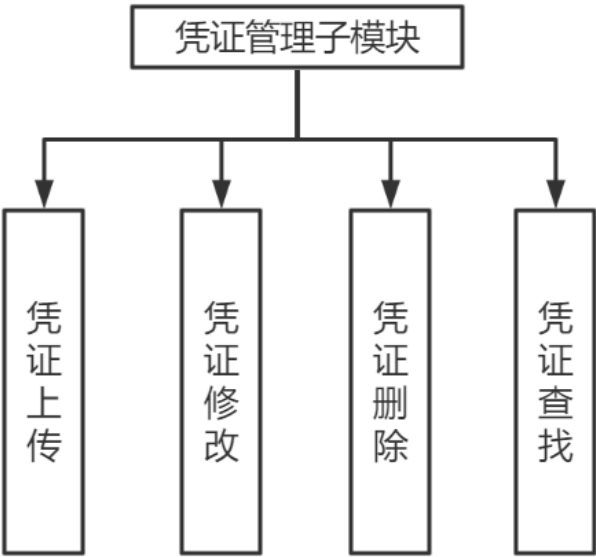
3-4 领养申请管理子模块流程设计

2.领养关系维护模块

根据前文 2.2 节中的模块划分，领养关系维护模块又可以分为凭证管理子模块以及回访管理子模块。领养关系维护子模块对于确认已领养动物的生活、健康状态十分重要，关系到对动物的后续保障。

2.1. 凭证管理子模块

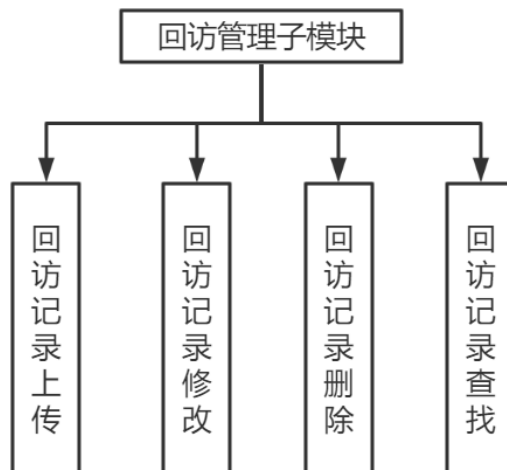
凭证管理子模块用于管理动物被领养后饲主应该向组织提供的相应凭证，包括身份信息、绝育证明等。只有已经通过审核的领养申请，即已经成立的领养关系，可以上传并且关联凭证。凭证管理子模块包括凭证上传、凭证修改、凭证删除、凭证修改等功能。相关结构图如下：



3-5 凭证管理子模块结构图

2.2 回访管理子模块

回访管理子模块用于管理动物被领养后志愿者进行回访行动的记录，包括动物健康信息、状态评估等，方便相关的工作人员判断动物是否得到良好的照顾。回访管理的使用者是志愿者。回访管理子模块包括凭证上传、凭证修改、凭证删除、凭证修改等功能。相关结构图如下



3-7 回访管理子模块结构图

3.系统管理模块

根据前文 2.2 节中的模块划分,系统管理模块又可以分为用户管理子模块以及角色权限子模块。

3.1 用户管理子模块

用户管理子模块实现了对本系统的所有用户的管理。

超级管理员以列表的形式管理系统中所有的用户,向其提供了用户数据的增删改查和用户数据的 excel 导出等功能。

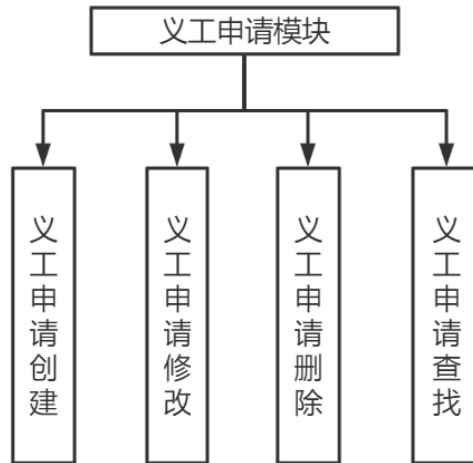
用户可以自行注册账号登录进入系统,初始默认为普通用户。同时,用户应能够更改自己的重要个人信息。

3.2 角色权限管理

系统中内初始设定应有普通用户、志愿者、超级管理员三类角色。系统管理员可以设定角色并且为每个角色配置相应权限,系统中每一个用户必须在拥有其角色的前提下才可以使用系统的相关权限,即在添加用户时必须要为其指定角色。通过注册进入系统的角色默认角色设置为普通用户。角色权限管理子模块主要包括对角色信息的增删改查和对角色进行授权等功能。

4. 义工申请模块

义工申请模块用于管理用户提交的义工申请。用户需在填写义工申请所需的个人信息并且提交。志愿者以列表的形式管理义工申请信息,并且根据实际情况进行审核。义工申请子模块包括义工申请创建、义工申请修改、义工申请删除、义工申请查找、义工申请审核等功能。相关结构图如 3-13 所示。



3-11 义工申请模块

义工申请模块是关联用户和志愿者的功能模块。用户只能进行创建义工申请的用例。志愿者可以对申请进行修改、删除、查看、审核。相关的流程设计与领养申请基本一致，在此略去。

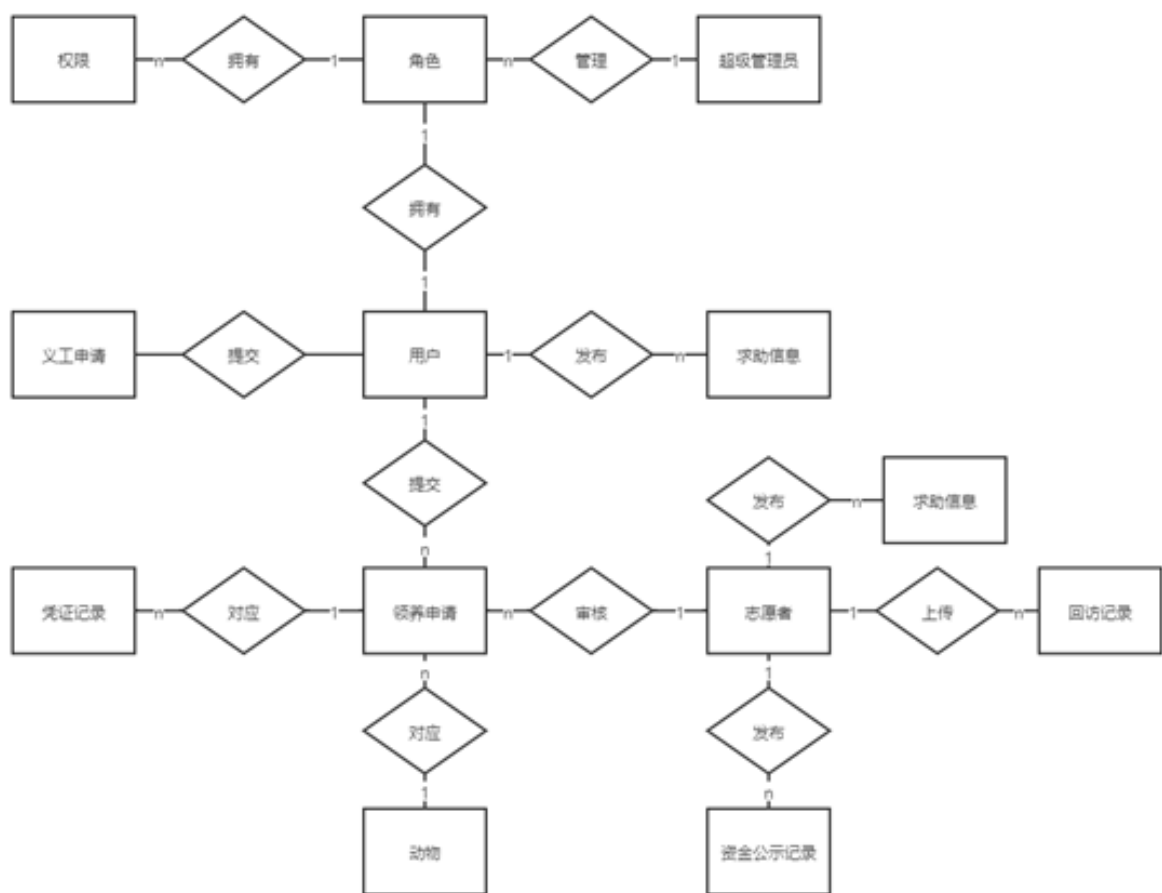
5. 信息公示管理模块

信息公示管理用于志愿者管理在首页进行展示的通知信息以及资金公示相关图表对应数据，主要对应于相关数据的增删改查操作，较为简单，在此不做赘述。

(三) 数据库设计

本系统开发数据库选择：MySQL。相较于主流的关系型数据库 Oracle、SQL Server 等，MySQL 更适合中小型网站开发。

根据第二章进行的需求分析，已经创建的主要实体有：用户、角色、领养申请、动物、通知、凭证记录、回访记录、义工申请、资金公示信息，如 ER 图所示。



系统 ER 图

根据关系模型，将实体转化为实际的数据库中的表。义工申请表和领养申请表在实际使用的表单基础上设计完成，考虑到审核需求添加了审核状态字段。对表结构进行一定规范化处理后，以下展示一些重要表的具体结构。

3-1 用户信息表

字段名称	字段描述	数据类型	长度	备注
id	用户 ID	bigint	20	主键
username	用户名	varchar	32	
password	密码	varchar	32	
email	邮箱	varchar	20	
avatar	头像	varchar	255	
role	角色	varchar	255	

3-2 角色信息表

字段名称	字段描述	数据类型	长度	备注
------	------	------	----	----

id	角色 ID	bigint	20	主键
name	角色名称	varchar	32	
description	角色描述	varchar	255	
role	角色	varchar	255	

3-3 权限信息表

字段名称	字段描述	数据类型	长度	备注
id	权限 ID	bigint	20	主键
name	权限名称	varchar	32	
description	权限描述	varchar	255	
path	权限路径	varchar	255	
flag	权限标识	varchar	32	

3-4 动物信息表

字段名称	字段描述	数据类型	长度	备注
id	动物 ID	bigint	20	主键
name	名称	varchar	32	
type	品种	varchar	20	
gender	性别	varchar	20	
birthday	生日	date		
pic	照片	varchar	100	
state	状态	int	12	
describe	描述	varchar	100	

3-5 领养申请信息表

字段名称	字段描述	数据类型	长度	备注
animal_id	动物 ID	bigint	20	主键、外键
user_id	用户 ID	bigint	20	主键、外键
age	年龄	int	12	
gender	性别	varchar	20	
maritalstatus	婚姻状况	int	12	
occupation	职业	varchar	20	
tel	联系方式	bigint	20	

location	住址	varchar	100	
fix_resident	是否有固定住所	int	12	
income	收入	int	12	
experience	有无养宠经验	int	12	
pet_num	现有宠物数量	int	12	
family_agree	是否家人同意	int	12	
wechat	微信号	varchar	20	
state	审核状态	int	12	

3-6 义工申请信息表

字段名称	字段描述	数据类型	长度	备注
id	义工申请 ID	bigint	20	主键
name	真实姓名	varchar	20	
age	年龄	int	12	
tel	电话	varchar	20	
email	电子邮件	varchar	20	
wechat	微信号	varchar	20	
location	现住址	varchar	100	
company	单位	varchar	100	
is_visit	是否到过救助中心	int	12	
ability	能力自述	varchar	255	
sparetime	每周空闲时间	int	12	
state	审核状态	int	12	

3-7 凭证记录信息表

字段名称	字段描述	数据类型	长度	备注
id	凭证 ID	bigint	20	主键
animal_id	动物 ID	bigint	20	外键

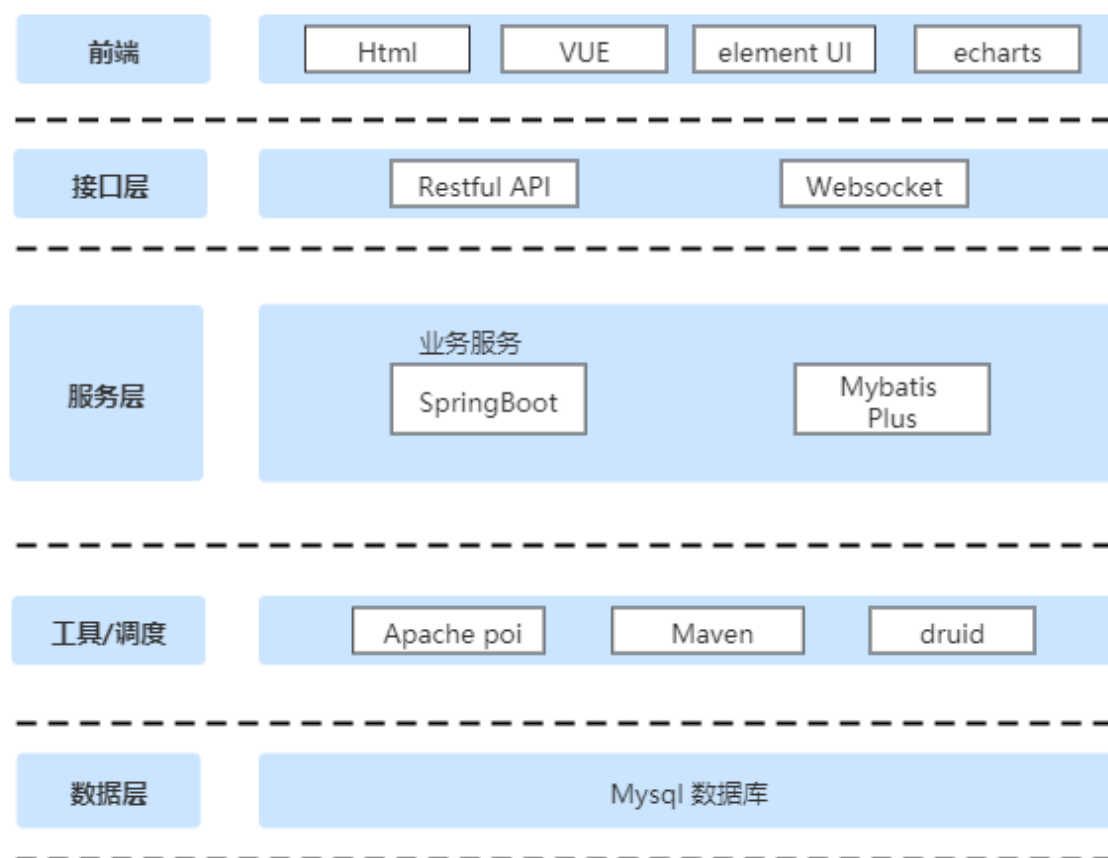
user_id	用户 ID	bigint	20	外键
title	凭证标题	varchar	100	
pic	凭证照片	varchar	255	

3-8 回访记录信息表

字段名称	字段描述	数据类型	长度	备注
id	回访记录 ID	bigint	20	主键
animal_id	动物 ID	bigint	20	外键
user_id	用户 ID	bigint	20	外键
visit_time	回访时间	date		
state	动物健康状态	int	12	
remark	回访评价	varchar	255	
visit_name	回访人姓名	varchar	20	

四、系统关键实现

(一) 系统架构



4-1 系统架构图

客户端使用的关键技术或框架为：Vue.js、ajax、jQuery、Element UI。

服务端使用的关键技术或框架为：Spring Boot、MyBatis Plus。

项目运行环境：

Java 开发环境：JDK 1.8

依赖管理工具：Maven 3.6.1

数据库环境：MySQL 5.6.

代码开发工具：Idea 2019.2.3

数据库管理工具：Navicat for MySQL

本系统前端部分使用了 vue 和 element UI。Element UI 是一套基于 Vue 2.0 的组件库，提供了配套设计资源，能够有效提高开发效率。

服务端基于 MyBatis Plus 和 Spring Boot 构建。目前业界普遍使用的还是全称 Spring+SpringMVC+MyBatis 的 SSM 框架。Spring 通过注入依赖等方式，为企业级 Java 开发即 J2EE 提供了一种相对简单的方法。但是同时，它也存有配置繁琐、对 xml 文件依赖严重等问题，一定程度上影响开发和部署工程的效率。和 Spring 相比，Spring Boot 可以显著减少开发中配置工作量，对 xml 配置文件要求较低。MyBatis 是专门用于和数据库进行交互的持久层框架。MyBatis Plus 是一个封装了 MyBatis 的增强工具包，可以起到简化开发、提高效率的作用。

本系统选择 springmvc 作为控制层。MVC 模式将应用分为三层：模型层、视图层、控制层。经典的 J2EE 开发常常使用 JSP+Servlet+JavaBean 的开发模式。当采用 MVC 模式开发时，有较高的耦合度，造成维护困难，且项目中需要书写较多的重复代码。为了解决这些问题，SpringMVC 给出了新的方案：分离了控制器、模型对象、过滤器以及处置程序对象的角色，用控制器取代之前 Servlet 处理请求做响应，传统的模型层被拆分为了业务层和数据访问层。在经典的 SSM 的框架中，Spring MVC 取代 servelt 充当以往控制层的角色，在 Service 下可以通过 Spring 的声明式事务操作数据访问层，Mybatis 则作为数据访问层的解决方案，而 Spring 是横跨三层的一种容器框架。SpringMVC 的特点有结构松散，几乎可以使用各类视图，不会强制使用 jsp 技术；耦合度低，便于实现各模块分离；与 Spring 无缝集成，增加开发效率等。

前端构建页面和设置样式主要使用了 HTML+CSS+javas 的组合，同时选择了 ajax 来访问后端提供的 API 接口。在与后台交互方面选择了使用 vue。获得由调用 API 接口提供的数据后，将数据保存在对应的 Vue 实例对象的 data 仓库中，用插值表达式将数据标记在需要的地方。数据即可在视图中被展示。相比于以往 DOM 操作，数据的处理更加直接便捷。

服务端的代码在实现时进行了较为严格的分层，其中，各个 package 的内容如下：

1. common: 保存的是一些配置类，包括处理跨域请求的 CORS 配置类、拦截器、mybatis-plus 分页插件、开启 websocket 支持的配置类等。

2. core: 实现实时数据传输的 WebSocketServer 类

3. entity: 与数据表严格映射的实体类。

4. controller: 保存的是所有的 API 请求类。

5. dto: 向页面返回数据的文件 filevo 类。

6. mapper: 集成了 MyBatis Plus 的通用 Mapper 接口。

7. service: 具体的业务实现类。

8. exception: 异常处理类

使用的 API 风格为 restful 风格的 api。在 REST 中，整个 Web 被看作一组资源的集合，资源是基于 URI 的实体，包括实体对象和业务逻辑，对资源进行的操作是由客户端指定的 URI 和 HTTP 请求的组合来实施的。^[6]

(二) 领养管理模块

2.1 动物管理子模块

动物管理实现了志愿者对于动物信息的增加、删除、编辑、条件查找，以及动物状态的修改。志愿者以列表的形式对动物信息进行管理。图 4-4 展示了动物管理相关界面。点击新增或者编辑，会出现如图 4-5 的新增/编辑动物信息相关界面。图 4-5 展示了新增或者编辑动物信息所需要填写的对应表单，志愿者需要填入动物名称、品种、性别图片等关键信息。

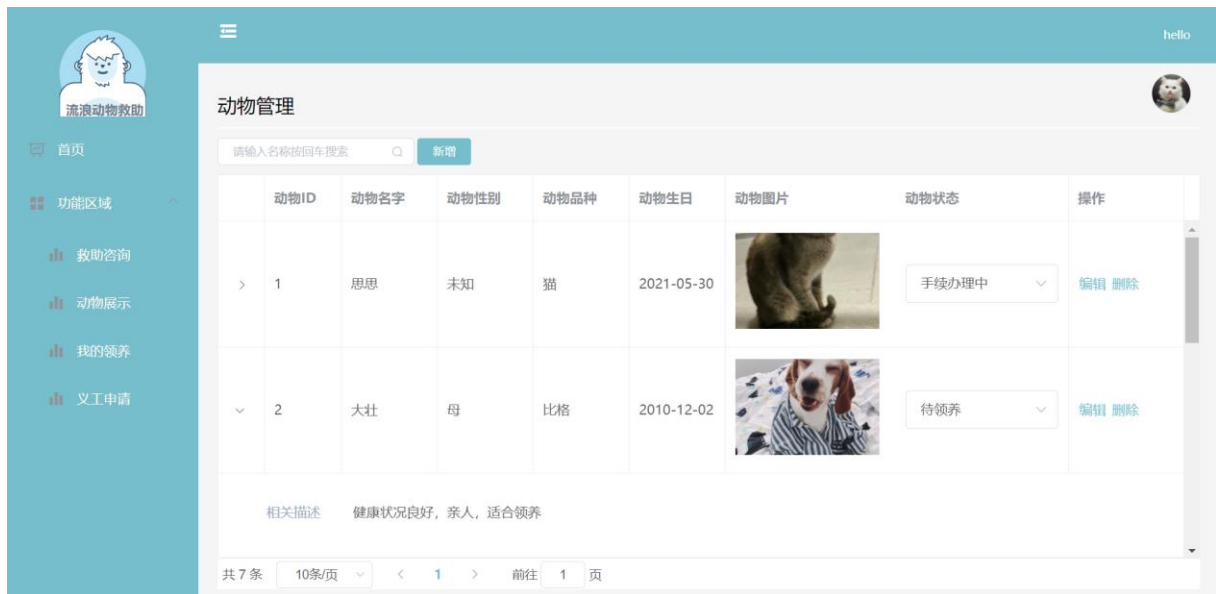


图 4-4 动物管理界面

动物信息管理

×

动物名字

动物品种

动物描述

动物性别

下拉选择

▼

动物生日

📅

选择日期

动物图片

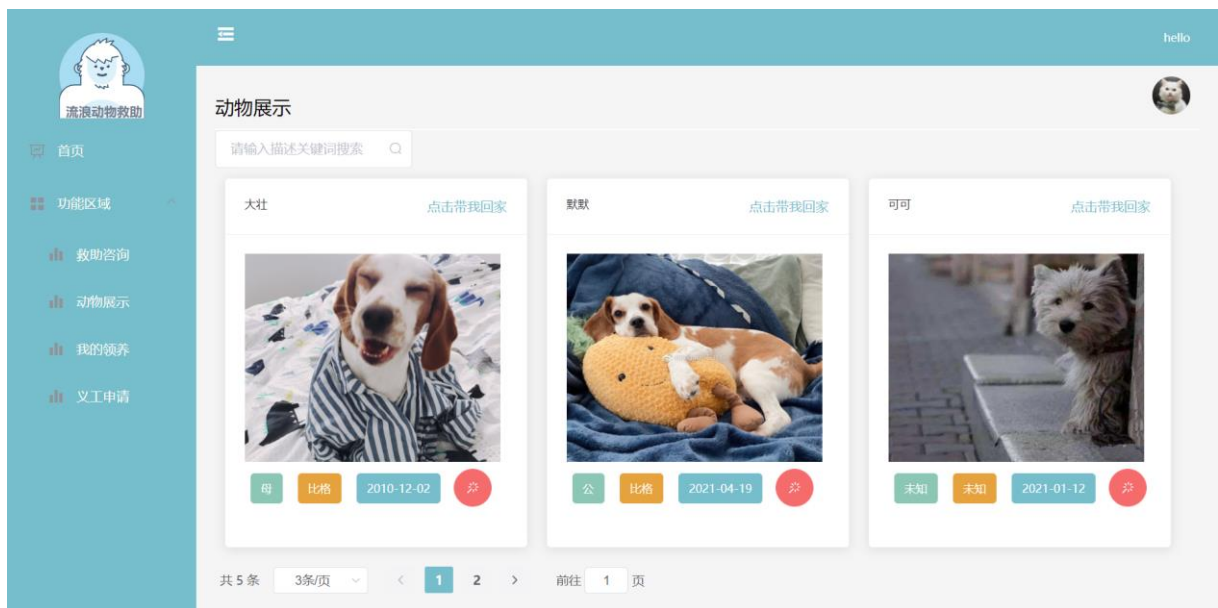
点击上传

取消

确定

图 4-5 新增/编辑动物信息相关表单

图 4-6 展示了动物信息发布后系统中显示的界面。普通用户登录系统后，点击左侧动物展示页面即可进入查看。未领养动物以卡片式列表的方式展现，特征以标签的形式显示在卡片上。部分较长内容放在弹出框中，点击红色按钮即可弹出。当内容较多时可以自动分页。用户可以根据自己感兴趣的点对隐藏的描述信息（即我的故事）进行模糊搜索。



4-6 动物信息展示界面

2.2. 领养申请管理子模块

领养申请管理子模块实现了志愿者对用户提交的领养申请信息的修改、删除以及审核。图 4-7 展示了领养申请管理界面。在一些影响审核判断的关键字段（固定住所、养宠经验、家人同意）配置了筛选器，同时设置了现有宠物数量、申请人年龄字段可排序，便于用户选择合适的视图方式。点击导出可以将领养申请导出到 excel。考虑到表单长度以及用户隐私，在表格上添加了展开行，部分信息用户必须打开才可以看到。志愿者可以对领养申请的状态进行审核，点击按钮切换状态。



4-7 领养申请管理页面

图 4-8 展示了用户端提交领养申请的相关界面。用户在动物展示界面选择对应的动物即可进入此页面。因为领养表的特殊性（双主键），单个用户只能对单个动物提交一次申请。若尝试多次提交

会提示用户前往我的申请页面对已存在的记录进行修改。同时，对部分字段名设置了数值、格式上的验证。图 4-9 展示了用户登录系统后查看自己提交过的领养申请的界面。未通过审核的领养申请可以被修改、删除。点击编辑按钮，会出现如 4-11 的编辑领养申请界面。

4-8 用户端领养申请提交页面

4-9 我的领养界面

(三) 领养关系维护模块

3.1 凭证上传子模块

凭证管理子模块实现了志愿者对于用户在领养关系成立（领养申请通过审核后）上传的相关凭证信息的编辑、删除功能。志愿者以列表的形式对凭证进行管理。在搜索框中可以对凭证名称进行搜索。图 4-10 展示了志愿者端的凭证管理页面。



4-10 凭证管理界面


图 4-11 展示了用户端上传相关凭证的界面。凭证上传后，即显示在下方和志愿者端对应列表中。



4-11 我的凭证上传界面

3.2 回访管理子模块




本模块实现了志愿者对于对已经被领养的动物的回访记录的新增、编辑、删除等功能。志愿者以列表形式管理所有记录。图 4-12 中展示了回访记录管理页面。在领养申请管理页面点击回访录入。动物状态使用了 element 的组件打分器进行标识和显示，较为直观。



[首页](#)
[功能区域](#)

tom

回访管理

动物ID	回访时间	动物状态	回访图片	回访评价	回访人姓名	操作
1	2021-04-06	★★★★☆ 一		健康状况良好，即将绝育	张文玥	编辑 删除
1	2021-04-05	★★★★★ 满		泪痕较重，需要主人进一步的呵护。精神状态良好	叶勇	编辑 删除
1	2021-05-02	★★★★★ 惊		狗狗状态较为良好，持续关注	张文玥	编辑 删除

共 3 条
10条/页
1
前往 1 页

4-12 回访记录管理界面

(四) 系统管理模块

在浏览器中输入网址 <http://localhost:9999/page/end> 后，即自动重定向到登录页面。登录页面如下图所示 4-13 所示。如果用户没有注册，点击“去注册”跳转到如 4-14 所示的注册界面。用户输入账号、密码、邮箱、电话等信息即可注册成功。注册用户默认为普通用户。





登录

[还没账号? 去注册](#)

4-13 登录界面

The registration interface features a light blue background with a central white rounded rectangle. At the top of the rectangle is a cartoon character icon. Below the icon are four input fields: the first contains the text 'hello', the second contains '*****', the third is labeled '请输入邮箱' (Please enter email), and the fourth is labeled '请输入电话' (Please enter phone). A blue '注册' (Register) button is positioned below the input fields. At the bottom right of the white rectangle, there is a link that says '已有账号, 去登录' (Already have an account, go to login).

4-14 注册界面

(五) 信息公示模块

图 4-15 为系统首页，主要用于展示公告以及关键图表。为了保证界面的美观，通知公告设置了手风琴效果，单次只能展开一条记录。



4-15 系统首页

(六) 义工申请模块

义工申请管理子模块实现了志愿者对用户义工申请管理子模块中所提交的义工申请信息的修改、删除以及审核。图 4-16 展示了义工申请管理界面。考虑到表单长度以及用户隐私，在表格上添加了展开行，用户必须打开才可以看到。志愿者可以对义工申请的状态进行审核，点击按钮切换

状态。图 4-17 展示了用户端提交义工申请对应的相关界面。登录后在菜单栏选择义工申请即可进入。相关记录提交后即出现在义工申请管理相应列表中。图 4-18 展示了系统中配置的一个基于 websocket 的在线聊天室，用户可以在线向在线的志愿者进行咨询。



4-20 义工申请管理界面



4-21 用户端义工申请提交界面



4-22 在线咨询界面

五、结论

(一) 技术总结

在实现该救助领养管理系统的过程中,遇到了一些技术问题,以下依次列举:

1. 跨域问题的处理

因为前端和后端可能需要运行在不同的端口下,很难避免产生跨域问题。本项目中采取了 CORS 的方案进行解决,配置使用了 Spring MVC 提供的 corsFilter 过滤器。

2. 记录分页问题的处理

关于分页的实现,前端使用的是 element UI 的 el-pagination 组件,由两个钩子函数 handleSizeChange 以及 handleCurrentChange 来处理页面大小的变化。服务端使用了 mybatis plus 的分页插件。经过测试,如果不在系统中配置分页插件,直接使用 mybatis plus 的 selectPage 等分页方法,在控制台输出的 sql 语句显示并没有物理分页,即事实上会先把记录先全部从数据库中取出,根据偏离值和限制值截断记录返回。在数据量较大的情况下会造成内存溢出,将对系统安全造成影响,故而不可取。

3. 实时聊天功能的实现

本处的实时聊天功能实现后端需要定义 websocket 服务端,书写 @OnMessage 等调用方法,使用 hutool 工具类对 json 进行解析。前端需要新建一个 WebSocket 对象来和服务器端进行交互,来发送信息以及将收到的消息动态加载在聊天框中。

(二) 系统展望

在线咨询聊天室目前还只支持文字聊天,如果需要发送更多信息还需要进一步集成富文本框等插件。同时,因为需要从 session 中获取用户数据,需要聊天双方都保持在在线咨询页面,离开页面会关闭连接移除 session。后续需要实现离线功能需要进一步设计对应的数据库。

从页面上来说,大量使用 element UI 的相关组件配置,总体较为整洁。但是页面上多出使用差值表达式,有时候先显示差值表达式后显示值,产生“差值表达式闪烁现象”。后续有待解决。

系统内存储了较多的个人信息，安全性有待进一步增加。后续可以考虑添加图片水印功能，增加使用者对于系统的信任感。

以上为本系统现阶段较为突出的问题，希望通过进一步的学习能够给予优化。

参考文献

- [1] 李玲. 城市流浪猫助养公益组织服务创新设计研究[D].南京艺术学院,2019.
- [2] 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.
- [3] 唐玄輝,陳沁蔚. 動物之家領養流程之服務設計[A]. UXPA 中国.UXPA 中国 2016 行业文集 [C].UXPA 中国:上海优帕会展有限公司,2016:10.
- [4] 徐雯,高建华.基于 Spring MVC 及 MyBatis 的 Web 应用框架研究[J].微型电脑应用,2012,28(07):1-4+10.
- [5] 李洋.SSM 框架在 Web 应用开发中的设计与实现[J].计算机技术与发展,2016,26(12):190-194.
- [6] 潘冰.面向资源的 RESTful Web 应用研究[J].微计算机应用,2010,31(07):38-43.
- [7] 王小铭. 管理信息系统学习与开发实践指南[M]. 华南理工大学出版社, 1999.
- [8] 杨华. 基于 SSM 的宠物店信息系统的设计与实现[D].吉林大学,2019.
- [9] 陈夫真. 基于 SSM 的某高校教室管理信息系统的设计与实现[D].苏州大学,2012.
- [10] 李宇,刘彬.前后端分离框架在软件设计中的应用[J].无线互联科技,2018,15(17):41-42.
- [11] 陈夫真. 基于 SSM 的某高校教室管理信息系统的设计与实现[D].苏州大学,2012.

附录

以下附录部分重要代码

1. pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>animal-home</artifactId>
    <version>1.0-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.1.0.RELEASE</version>
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
        <mybatis-plus.version>3.2.0</mybatis-plus.version>
    </properties>

    <dependencies>
        <!-- 添加 Springmvc 依赖 -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <!-- springBoot JPA 的起步依赖 -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>

        <!-- Mybatis -->
        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
```

```

        <artifactId>mybatis-spring-boot-starter</artifactId>
        <version>2.1.4</version>
    </dependency>

    <!-- mybatis-plus -->
    <dependency>
        <groupId>com.baomidou</groupId>
        <artifactId>mybatis-plus-boot-starter</artifactId>
        <version>${mybatis-plus.version}</version>
    </dependency>
    <dependency>
        <groupId>com.baomidou</groupId>
        <artifactId>mybatis-plus-generator</artifactId>
        <version>${mybatis-plus.version}</version>
    </dependency>
    <dependency>
        <groupId>org.apache.velocity</groupId>
        <artifactId>velocity-engine-core</artifactId>
        <version>2.1</version>
    </dependency>

    <dependency>
        <groupId>com.github.jeffreyning</groupId>
        <artifactId>mybatisplus-plus</artifactId>
        <version>1.2.0-RELEASE</version>
    </dependency>

    <!-- websocket -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-websocket</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
    </dependency>

    <!-- 数据连接池 druid-->
    <dependency>
        <groupId>com.alibaba</groupId>
        <artifactId>druid</artifactId>
        <version>1.2.2</version>

```



```

</dependency>
<!--mysql-->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
</dependency>
<dependency>
    <groupId>cn.hutool</groupId>
    <artifactId>hutool-all</artifactId>
    <version>5.5.4</version>
</dependency>
<!-- excel 导出 -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>4.1.2</version>
</dependency>
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
</dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
                <fork>true</fork>
            </configuration>
        </plugin>
    </plugins>
</build>
<repositories>
    <repository>
        <id>public</id>
        <name>aliyun nexus</name>
        <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
        <releases>
            <enabled>true</enabled>
        </releases>
    </repository>
</repositories>

<pluginRepositories>

```

```

<pluginRepository>
  <id>public</id>
  <name>aliyun nexus</name>
  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</pluginRepository>
</pluginRepositories>
</project>

```

2. 跨域部分处理

```

@Configuration
public class CorsConfig {

    // 当前跨域请求最大有效时长。这里默认 1 天
    private static final long MAX_AGE = 24 * 60 * 60;

    private CorsConfiguration buildConfig() {
        CorsConfiguration corsConfiguration = new CorsConfiguration();
        corsConfiguration.addAllowedOrigin("*"); // 1 设置访问源地址
        corsConfiguration.addAllowedHeader("*"); // 2 设置访问源请求头
        corsConfiguration.addAllowedMethod("*"); // 3 设置访问源请求方法
        corsConfiguration.setMaxAge(MAX_AGE);
        return corsConfiguration;
    }

    @Bean
    public CorsFilter corsFilter() {
        UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurati
onSource();
        source.registerCorsConfiguration("/**", buildConfig()); // 4 对接口配
置跨域设置
        return new CorsFilter(source);
    }
}

```

致谢

时光飞逝，光阴荏苒，转眼已经走到大学生活的尾声。在这半年挣扎毕业设计和考研之间，我真真切切地学到了很多，心态也坚忍了许多。在此，想对所有帮助过我的人说一声感谢。

首先，我要感谢我的导师唐晓新老师。在进行毕业设计的过程中，老师一直耐心指导我，给我提出了很多宝贵的修改意见，在学习、工作、生活上都给了我莫大的帮助和鼓励。在此，向老师表达深深的谢意。

其次，我要感谢四年来所有在本科阶段教授过我课程的老师对我的辛勤培养和悉心教育。是你们的悉心教导，让我在学习知识的同时，培养了独立处理问题的能力。

最后要感谢我的亲人和朋友们，感谢他们永远无条件的关爱与支持，让我有勇气和信心在求知这条道路上坚定地走下去。