|  |  |
| --- | --- |
| **成绩** | （采用四级记分制） |

****

**本科毕业论文（设计）**

**题目：**

**学生姓名**

**学 号**

**指导教师**

**院 系**

**专 业**

**年 级**

**教务处制**

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： 2008年6月7日

摘要

随着城市化进程的加快，流浪动物问题日益凸显，持续影响着城市的公共卫生和安全。流浪动物往往面临食物短缺、疾病缠身等问题，同时它们的存在也可能给人类生活带来不便甚至安全隐患。因此，构建一个有效的流浪动物救助平台对于改善这一状况具有重要意义。该平台可以有效地促进信息共享，帮助更多有爱心的人士参与到救助活动中来，为想要领养宠物的家庭提供了一个可靠的渠道。此外，该平台还能协助相关机构更高效地管理流浪动物资源，提高救助效率，最终达到减少流浪动物数量的目的。本项目旨在利用现代信息技术手段，解决流浪动物救助过程中存在的诸多难题，如信息不对称、救助流程繁琐等，以期能够更好地服务于社会公益事业。

关键字：流浪动物救助；软件工程；领域驱动设计（DDD）；微服务；消息队列。

**ABSTRACT**

With the acceleration of urbanization, the problem of stray animals is becoming increasingly prominent, continuously affecting the public health and safety of cities. Stray animals often face problems such as food shortages and illnesses, and their presence may also bring inconvenience and even safety hazards to human life. Therefore, building an effective platform for rescuing stray animals is of great significance in improving this situation. This platform can effectively promote information sharing, help more caring people participate in rescue activities, and provide a reliable channel for families who want to adopt pets. In addition, the platform can assist relevant institutions in managing stray animal resources more efficiently, improving rescue efficiency, and ultimately achieving the goal of reducing the number of stray animals. This project aims to use modern information technology to solve many problems in the process of rescuing stray animals, such as information asymmetry and cumbersome rescue processes, in order to better serve social welfare undertakings.

**keyword:** Software engineering; Domain Driven Design (DDD); Microservice; Message queue.

1 绪论

1.1 研究背景及意义

随着城市化进程的加速，流浪动物问题日益凸显，不仅影响城市的公共卫生与安全，还给城市环境造成了负担。流浪动物面临着食物短缺、疾病困扰等问题，同时它们的存在也可能给人类生活带来不便甚至安全隐患。因此，构建一个有效的流浪动物救助平台，对于改善这一状况具有重要意义。此平台不仅能促进信息共享，还能帮助更多有爱心的人士参与到救助活动中来，为想要领养宠物的家庭提供一个可靠的渠道。此外，平台还能协助相关机构更高效地管理流浪动物资源，提高救助效率，最终达到减少流浪动物数量的目的。

1.2 国内外研究现状

在国内，张秋雨在其研究中指出，流浪动物救助工作依赖于非政府的民间组织，而这些组织在物资、人力、条件等方面存在诸多问题。他提出通过鼓励大学生志愿者参与、加强与地方媒体合作、推行政府购买服务等方式来优化救助路径[1]。同样，雷思雨等人也指出了流浪动物救助站所面临的困境，如缺乏科学管理体系、救助资金及物资匮乏、领养渠道不畅等问题，并建议通过优化管理体系、拓宽救助资金渠道和打通领养渠道来解决这些问题[2]。

国外的情况也不尽相同，虽然有些国家通过立法来规范宠物领养和管理，如美国和英国等地通过严格的宠物登记制度和动物福利法律来减少流浪动物数量，但是这些国家同样面临着管理和执行上的挑战。一些非营利组织利用互联网平台进行流浪动物的救助，如美国的Petfinder和英国的Dogs Trust等，它们通过网站提供领养信息，帮助流浪动物找到新家。然而，这些平台的效果仍有局限，尤其是在信息覆盖范围、用户参与度以及救助流程透明度等方面存在改进空间。

综上所述，无论是国内还是国际上，都存在着缺乏一个综合性强、功能完善的流浪动物救助平台的问题。现有的一些尝试虽然在一定程度上缓解了流浪动物问题，但未能形成系统性的解决方案。因此，开发一个集信息共享、用户互动、救助流程透明化于一体的平台是必要的，这样的平台不仅能提高救助效率，还能促进更多人的参与，进而为解决流浪动物问题提供有力支持。本项目正是在此背景下展开，旨在通过现代信息技术手段，构建一个高效、透明的流浪动物救助平台，以期更好地服务于社会公益事业。

1.3 论文研究内容

2 相关技术与工具

2.1 相关知识

2.1.1 软件工程

软件工程涵盖了从软件需求分析到设计、编码、测试、维护等多个阶段的过程管理，其目的在于提高软件产品的质量和开发效率。随着技术的发展，软件工程不断引入新的理论、方法、语言和框架，如从传统的瀑布模型到敏捷开发模型的转变，以及从面向过程编程到面向对象编程的演进等[3]。

2.1.2 领域驱动设计（**DDD**）

领域驱动设计（Domain-Driven Design，简称DDD）是一种以业务为核心，强调软件系统与业务模型紧密结合的开发方法[4]。DDD的核心是建立一个既符合软件所在领域本身，又适合软件分析开发需求的领域模型，以此来提炼业务，表达业务以及实现业务。这种方法强调了领域模型的重要性，将其作为领域专家、分析人员和开发人员之间交流沟通的核心，确保所有参与者都能够共享相同的理解和语言，从而提高建模能力和软件的正确性[5]。DDD的核心思想包括：

统一语言（Ubiquitous Language）：确保所有参与者共享相同的专业术语和概念，以便于有效的沟通。

领域模型（Domain Model）：通过实体、值对象、聚合、服务、仓储等模型表示业务逻辑。

限界上下文（Bounded Context）：定义了领域模型的边界，以便于理解和管理复杂性。

聚合（Aggregates）：一组不可分割的对象，确保了对象内部一致性。

DDD鼓励开发者深入理解业务领域，并通过持续的模型重构来保持模型与领域发展的同步，从而提高软件的灵活性和可维护性[4]。

2.1.3 微服务

微服务架构是一种将单个应用程序拆分为一组小型、独立服务的架构方式。这些服务围绕特定的业务功能构建，并且可以独立地部署和扩展[6]。相较于传统的单体架构，微服务架构具有更高的灵活性和扩展性，但也带来了服务间的通信、数据一致性等挑战[7]。微服务通常通过轻量级的通信机制（如HTTP RESTful API，gRPC）进行交互，并且每个服务都可以选择最适合自身的技术栈[8]。

2.1.4 **RPC**通信

远程过程调用（Remote Procedure Call，简称RPC）是一种让程序调用位于另一台机器上的子程序的方式，而无需了解底层网络细节。在微服务架构中，RPC通常作为一种服务间通信的有效方式，尤其是在需要高效、低延迟通信的情况下。RPC框架如Dubbo，提供了高性能、透明化的远程服务调用能力，它支持传统的RPC远程服务调用模式，同时也兼容RESTful风格的远程调用，使得服务提供者和服务消费者之间的交互更加便捷和灵活[7]。

RPC通信的关键特性包括：

透明性：RPC调用看起来就像本地函数调用一样，开发人员不需要关心网络细节。

序列化机制：数据在发送之前会被序列化，在接收端会被反序列化，常用的序列化技术有JSON、XML等。

安全性：RPC框架支持认证和授权机制，保障了服务之间的安全通信。

容错性：提供了诸如超时重试、断路器等机制来增强服务的鲁棒性。

RPC作为一种服务间的通信方式，对于实现服务解耦、提升系统性能等方面有着重要作用，尤其是在微服务架构中，RPC成为了不可或缺的一部分[7]。

2.2 相关技术与工具

2.2.1 **MySQL**关系型数据库

MySQL是一种开放源代码的关系型数据库管理系统，因其功能强大、使用简便、管理方便、运行速度快以及安全性高等特点，被广泛应用在多种系统中。在进行数据库设计时，需要进行规范化处理来消除数据冗余，但过度规范化可能会导致性能下降，因为在完成某些检索时可能需要进行复杂的联结操作。因此，需要在规范化和性能之间找到平衡点，采取诸如建立内存表、增加冗余列、表分割等措施来提高性能[9]。

2.2.2 **Redis**键值对缓存

Redis是一个开源的、高性能的键值对存储系统，除了提供简单的键值存储外，还支持字符串、列表、集合、有序集合等多种数据类型。Redis具有内存存储特性，可以快速进行数据的读写操作，并且支持持久化机制，以防止数据丢失。Redis可以用于缓存数据库中的数据、Web页面，以及存储复杂计算的结果等，以减轻数据库的负载，提高系统响应速度[10]。

2.2.3 **RabbitMQ**消息队列

RabbitMQ是一种基于AMQP协议的开源消息中间件，它支持多种工作模式，包括但不限于工作队列模式、发布订阅模式、路由模式、通配符模式等。RabbitMQ允许应用程序解耦，实现异步处理，流量削峰等功能，提高了应用程序的响应速度和处理效率。通过RabbitMQ，消息发布者可以将消息发送到消息队列中，消息消费者可以从队列中获取消息进行处理[11]。

2.2.4 JWT令牌

JSON Web Tokens（JWT）是一种开放标准（RFC 7519），用于安全地在各方之间传输信息。JWT令牌允许信息以一种紧凑、URL安全的形式在网络中传输，并且可以被任何一方验证和解析。这种令牌机制非常适合于跨站点请求伪造保护和单点登录等场景[12]。

2.3 开发运行环境

操作系统：Linux/Windows

开发语言：Java、Vue

开发工具：JetBrains IntelliJ IDEA、Visual Studio Code

运行环境：JDK8，Node.js 20，Erlang 26，GCC

部署环境：Docker云服务器（CentOS、Alibaba Cloud）

3 系统设计

3.1 需求分析

本项目致力于打造一个专为流浪动物救助与管理设计的综合性信息共享平台。通过深入的问卷调查与用户访谈，我们收集并细致分析了广泛的需求，从中提炼出了几项关键的核心需求。首先，平台需满足用户的信息共享需求，这意味着用户应能便捷地获取流浪动物的基础资料，如种类、年龄、性别、健康状态及图片等；同时，用户还期望能够追踪每一只动物的最新动态，比如是否已被成功救援、当前的身体状况以及是否有意领养的家庭正在寻找它。此外，为了更好地关注动物的健康，用户也表达了想要了解每只动物具体的喂食记录，包括喂食的具体时间、执行喂食的人以及喂食的数量等信息的愿望。

在救助需求方面，平台应提供一个功能强大的空间，使用户能够发布关于发现流浪动物的求助信息，内容涵盖发现的时间、地点及其当时的状况等细节。同时，平台还需具备高效的求助信息管理能力，确保所有求助帖子都能经过适当的审核、分类与存档。用户还希望能清楚地看到每条求助信息的处理进度，例如已有人响应、正在进行救援或是已经完成等，以帮助他们及时掌握事态发展。此外，平台应鼓励和支持救助组织的加入，允许其注册成为正式成员，并赋予它们接收和处理用户求助请求的能力，从而更有效地参与到救援行动中来。

对于有意向领养流浪动物的用户，平台则需提供一个安全可靠的领养流程。这包括但不限于允许用户提交领养申请，填写必要的个人信息，如联系方式、居住地址及领养动机等。同时，平台上的救助机构应当有能力对潜在领养者进行资格审查，涉及家访检查、背景调查等一系列步骤，确保每一只动物都能找到真正适合它的新家。一旦领养请求获得批准，平台将负责记录下所有的领养详情，包括领养日期、领养人的基本信息，并及时更新该动物的状态至“已领养”，以此完成整个领养过程的闭环管理。

用户还提到了一些其他需求，比如平台必须保证高度的安全性和隐私保护，确保所有用户数据不被非法访问或泄露。此外，系统的可用性也是至关重要的，平台需要保持高稳定性和响应速度，即使在高并发访问的情况下也能保证流畅的用户体验。为了适应不同用户群体的需求，界面设计应简洁直观，易于操作，同时提供多语言版本，覆盖更广泛的用户群体。最后，考虑到技术的持续进步，平台还应具有良好的可扩展性和灵活性，便于未来根据实际使用情况和技术发展趋势进行升级和优化。

3.2 系统架构设计

[image]系统架构图

3.1.1 前端架构设计

本系统的前端部分设计为适应多终端访问需求，支持通过H5网页、微信小程序及原生APP客户端等多种方式访问。前端开发采用了Vue.js框架配合Element UI组件库来构建响应式Web界面，并针对不同平台进行了适配优化以确保用户体验的一致性。同时，通过Nginx作为反向代理服务器来处理HTTP请求，实现了前后端之间的路由映射与分离。

3.1.2 后端架构设计

（1）网关层：为了简化外部调用者与内部服务交互的复杂度，我们引入了API Gateway模式。Gateway服务作为系统的统一入口点，接收来自前端的所有请求，并根据请求头中的元数据（如token、请求类型等）进行鉴权和路由到相应的微服务。此外，Gateway还负责负载均衡、限流、熔断等功能，以增强系统的稳定性和安全性。

（2）应用层：该层主要包含论坛服务、核心服务、管理服务等面向业务的服务。这些服务直接与业务逻辑紧密相关，负责处理具体的业务请求，并调用领域层提供的功能完成相应的业务操作。

（3）领域层：这一层次按照领域驱动设计（DDD）原则划分为不同的子域，包括用户、动物、消息、文章、机构以及验证码等领域服务。每个领域服务都专注于某一特定的功能集合，它们之间通过定义良好的接口相互协作，共同支撑起整个系统的业务逻辑。

（4）基础设施层：为了保障系统的高效运行，我们在这一层集成了多种技术组件和服务。其中包括MySQL数据库用于存储持久化数据；Redis作为缓存提高读写性能；RabbitMQ作为消息队列处理异步任务；Elasticsearch提供全文检索能力；Nacos则用于服务发现与配置管理。此外，还开发了一个通用工具包，封装了一系列常用的工具方法，如加密解密、日期处理、字符串操作等，以减少重复代码，提升开发效率。

3.3 系统功能设计

[image]系统功能树

根据3.1节的需求分析，本项目设计了如上图的功能模块，，以满足不同用户群体的需求。

3.3.1 用户登录及信息管理

3.3.2 动物信息管理

3.3.3 用户捐赠捐款

3.3.4 寻求救助功能

3.3.5 领养申请及审核

3.3.6 论坛发帖功能

3.3.7 发帖评论功能

3.3.8 投喂动物功能

3.4 数据库设计

[image]数据库主从架构图

3.4.1 用户数据库设计

[image]用户ER

3.4.2 动物数据库设计

[image]动物ER

3.4.3 帖子数据库设计

[image]帖子ER

3.4.4 机构数据库设计

[image]机构ER

3.4.5 消息数据库设计

[image]消息ER

3.4.6 验证码数据库设计

[image]验证码ER

3.5 界面原型设计

4 系统实现

4.1 用户管理

4.2 动物管理

4.3 机构管理

4.4 论坛管理

4.5 安全管理

5 系统测试

参考文献

[1]张秋雨.流浪动物救助实践困境与路径优化——基于四川省宜宾市的实证分析[J].法制与社会,2017,(15):188-190.DOI:10.19387/j.cnki.1009-0592.2017.05.371.

[2]雷思雨,武佳雪,胡月馨.流浪动物救助站的现状及对策研究[C]//成都信息工程大学管理学院.“劳动保障研究”2024研讨会论文集（上册）.成都信息工程大学;,2024:4.DOI:10.26914/c.cnkihy.2024.013876.

[3]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.

[4]严欣喆.领域驱动设计方法的研究及其应用[D].南昌大学,2010.

[5]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.

[6]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.DOI:10.16667/j.issn.2095-1302.2024.05.017.

[7]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.

[8]李阳春.面向服务的架构（SOA）开发设计方法综述[J].汽车制造业,2022,(04):22-24+29.

[9]兰旭辉,熊家军,邓刚.基于MySQL的应用程序设计[J].计算机工程与设计,2004,(03):442-443+468.DOI:10.16208/j.issn1000-7024.2004.03.037.

[10]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.DOI:10.19358/j.issn.1674-7720.2013.12.004.

[11]唐权,周蓉,张勇.RabbitMQ消息中间件在Spring Boot教学中的应用[J].现代信息科技,2020,4(18):125-127.DOI:10.19850/j.cnki.2096-4706.2020.18.035.

[12]范展源,罗福强.JWT认证技术及其在WEB中的应用[J].数字技术与应用,2016,(02):114.DOI:10.19695/j.cnki.cn12-1369.2016.02.087.