



廣東科技學院
GUANGDONG UNIVERSITY OF SCIENCE & TECHNOLOGY

葉選平

毕业设计

题目：宠物收养系统的设计与实现

| | |
|-------|-------------|
| 姓 名： | 许家俊 |
| 学 号： | 2019134243 |
| 层 次： | 本科 |
| 专 业： | 软件工程 |
| 班 级： | 19 软件本科 9 班 |
| 指导教师： | 孟雨 |

2023 年 5 月

广东科技学院教务处 制

毕业论文（设计）独创性声明

本人郑重承诺：所呈交的毕业论文（设计）是本人在指导教师的指导下，独立研究所取得的成果。本人在撰写毕业论文（设计）的过程中严格遵守学术规范，尽我所知，除致谢部分外凡文中引用的他人发表或未发表的学术成果均已注明出处，除引用部分外文中所有的数据及其他引证材料均真实客观。如有学术失范行为，本人愿意承担由此造成的一切后果。

论文（设计）作者签名：许家俊

日期： 2023 年 5 月 8 日

摘要 在我国社会的深入发展和我国经济的迅速崛起的过程之中，我国人民的收入和生活质量持续稳步向上，对生活的要求也有提高，宠物收养成为了一种趋势。宠物急速增加的数量必然会导致流浪动物大量出现。

本课题项目使用的是目前流行的前后端分离开发方式搭建的，摒弃了传统的前后端混合开发方式的同兼顾现实的实际需求，前端使用的是 Bootstrap 框架技术，数据库使用的 MySQL，后端框架使用的是 SSM，全套框架简化了冗余代码的编写，提高了开发效率。

宠物收养系统应用当前业界内的流行技术，可给予爱心人士较为全面的服务，同时还可以为爱心创业者提供一个创业平台，节约时间精力。系统根据实际需求，此系统分为用户管理，宠物管理，管理员管理，评论管理，活动日志管理，志愿者申请管理，收养申请管理等模块。通过本系统管理员可以高效、方便的管理系统内的各种信息，用户可以快捷的查看系统内的基本信息。

关键词 宠物收养；SSM；MySQL；Bootstrap

Abstract In the midst of our deepening social development and the rapid rise of our economy, our people's income and quality of life have continued to move steadily upwards and their demands on life have increased, with pet adoption becoming a popular trend. The rapid increase in pets inevitably leads to a large number of stray animals.

The project is built using the current popular front-end and back-end development method, abandoning the traditional mixed front-end and back-end development method and taking into account the actual needs of the reality, using Bootstrap framework technology for the front-end, MySQL for the database and SSM for the back-end framework, a full set of frameworks to simplify the writing of redundant code and improve development efficiency. The pet adoption system applies the current popular technology within the industry, which can give a more comprehensive service to caring people, and also provide a platform for caring entrepreneurs to start their own business and save time and energy. The system is divided into user management, pet management, administrator management, comment management, activity log management, volunteer application management, adoption application management and other modules according to actual needs. Through this system, administrators can efficiently and conveniently manage all kinds of information in the system, and users can quickly view the basic information in the system.

Keywords Pet adoption; SSM; MySQL; Bootstrap

目 录

| | |
|---------------------------|----------|
| 1 引言 | 1 |
| 1.1 研究的背景 | 1 |
| 1.2 研究意义 | 1 |
| 1.3 国内外应用现状 | 1 |
| 1.4 开发设计思路 | 2 |
| 1.5 论文架构 | 2 |
| 1.6 本章小结 | 2 |
| 2 开发技术简介 | 3 |
| 2.1 SSM | 3 |
| 2.2 MySQL | 3 |
| 2.3 Bootstrap | 3 |
| 2.4 JavaScript | 3 |
| 2.5 开发环境及其应用软件 | 4 |
| 2.6 本章小结 | 4 |
| 3 系统可行性和需求分析 | 5 |
| 3.1 系统可行性分析 | 5 |
| 3.1.1 技术可行性 | 5 |
| 3.1.2 经济可行性 | 5 |
| 3.1.3 市场可行性 | 5 |
| 3.1.4 时间可行性 | 6 |
| 3.2 系统用例分析 | 6 |
| 3.2.1 用户功能需求分析 | 6 |
| 3.2.2 管理员功能需求分析 | 7 |
| 3.3 非功能性需求分析 | 8 |
| 3.4 本章小结 | 8 |
| 4 系统概要设计 | 9 |
| 4.1 系统设计原则 | 9 |
| 4.1.1 可用性 | 9 |
| 4.1.2 安全性 | 9 |

| | |
|---------------------------|-----------|
| 4.1.3 稳定性 | 9 |
| 4.2 数据库设计 | 9 |
| 4.2.1 数据库的概念结构设计 | 10 |
| 4.2.2 数据库的逻辑结构设计 | 12 |
| 4.3 本章小结 | 14 |
| 5 系统详细设计 | 15 |
| 5.1 系统运行环境 | 15 |
| 5.1.1 硬件环境 | 15 |
| 5.1.2 软件环境 | 15 |
| 5.2 系统运行设计 | 15 |
| 5.2.1 运行模块组合 | 15 |
| 5.2.2 运行控制 | 15 |
| 5.3 系统结构设计 | 15 |
| 5.3.1 系统结构设计概述 | 15 |
| 5.3.2 系统架构的设计 | 15 |
| 5.3.3 系统功能模块图 | 16 |
| 5.4 系统管理端的功能模块设计 | 17 |
| 5.4.1 信息管理的模块设计 | 17 |
| 5.4.2 活动日志管理模块设计 | 19 |
| 5.4.3 收养审核模块设计 | 20 |
| 5.4.4 评论管理模块设计 | 22 |
| 5.5 系统用户端的功能模块设计 | 23 |
| 5.5.1 宠物收养模块 | 23 |
| 5.5.2 用户的评论模块 | 25 |
| 5.6 本章小结 | 26 |
| 6 系统功能实现 | 27 |
| 6.1 用户端 | 27 |
| 6.1.1 用户登录注册及首页 | 27 |
| 6.1.2 用户查看个人信息及收养宠物 | 28 |
| 6.1.3 用户评论 | 29 |
| 6.2 管理员管理 | 30 |
| 6.3 本章小结 | 36 |

| | |
|-----------------------|-----------|
| 7 系统功能测试 | 37 |
| 7.1 系统测试目的 | 37 |
| 7.2 系统测试内容 | 37 |
| 7.3 测试用例 | 37 |
| 7.4 测试总结 | 39 |
| 8 总结 | 40 |
| 8.1 系统总结 | 40 |
| 8.2 展望 | 40 |
| 参考文献 | 41 |

1 引言

1.1 研究的背景

伴随着中国社会全方位的更进一步发展,中国的经济和影响力也在高速增长,中国人的收入越来越高,生活的环境、质量也越来越好,对生活的要求也有提高,养宠物成为了一种流行趋势^[1]。宠物的增多不可避免地造成了城市流浪宠物的泛滥,造成流浪动物的成因是多方面的,主人的遗弃是造成流浪动物的重要原因之一,它们大多来自被遗弃的动物以及这些动物繁衍的后代^[2]。很多人因一时激情与爱养宠物而把它视为灵魂的慰藉,但激情退去后主人便在不知不觉中产生了关心小动物的厌倦情绪,从而导致了很多人被抛弃并且很多走失宠物仅能吃了上顿没下顿地过活的惨状。它们游走在人类居住区,给人们的生活带来了不小的困扰^[3]。相对于购买,一些人更倾向于收养这些宠物,让他们得到内心极大的满足、喜悦和成就感。

1.2 研究意义

而现今也存在由爱心人士组织的流浪动物救助站收养一些流浪狗,大多的城市都会出现一些由志愿者们自发组成的小型动物救助站,有爱心的志愿者们会把染病的和致残的流浪动物收留到爱心救助站里,有的善良的人会捐出善款来医疗得病和致残流浪动物,然后举行爱心活动来为这些倒霉的流浪动物寻找到一个新居。然而残酷的现实是有时候救助站里的宠物真的太多了,很难对付以至于许多被遗弃动物只能被实施“安乐死”这一结果^[4]。为解决目前流浪宠物难题,线下注册收养这一传统低效收养的方式逐步被取而代之,随之而来的是使用设备便可浏览访问的宠物收养管理系统。

1.3 国内外应用现状

国外的宠物收养行业已有较长的发展历史,相比起我国已较为成熟,特别是欧美国家,不仅对宠物的食物需求高,而且对宠物的护理打扮等产品的兴趣也是只增不减,而我国的宠物行业还存在市场无序竞争、繁育观念落后等问题。其实从宠物的饲养数量上来看:国内饲养宠物的人数已经比世界上任何一个国家都要多,主要因素是我国的人口基数大,也正因如此,弃养的宠物更是数不胜数^[5]。大部分人看来,宠物只是宠物,而宠物带来的负面影响会导致弃养,比如说宠物生病带来的费用难以承担就直接弃养。我国的宠物市场近几年才发展起来,很多方面并不完善,动物收养所便是重要一环。宠物行业涉及的范围较广,该行业内主要包括以下的两个行业,一个是

宠物服务，而另一个则为宠物产品。宠物服务包括宠物的医疗、宠物的美容、宠物的培训等^[6]；而宠物产品则包括宠物的交易、宠物的食品以及宠物的清洁用品等。在了解这些内容的情况下，很多人还是一种冲动的收养。而随着国内互联网技术的发展和成熟度越来越高，人们获取资讯以及了解信息的途径大部分都是通过网络进行。因此通过建立一个宠物收养系统能够实现包括宠物收养等功能，为宠物主人提供好的服务。

1.4 开发设计思路

本宠物收养系统主要采集合 Spring+SpringMVC+MyBatisplus 的 SSM 框架技术来实现 MVC 模式，结合前端技术 HTML、CSS 和 JavaScript 等语言，进一步使用 MySQL 数据库来存储数据来共同开发本系统。本系统的特点页面的模块化，即系统的接口是以模块化的方式设计的，将几个页面的共同部分整合到一个模块中；业务流程封装化即使业务逻辑处理层、用户界面层以及数据存储层的来设计^[7]；面向对象的设计是指用面向对象的思维，把各个部分的实体封装成其相应的数据类，然后再进一步把一个个的数据类开发成相对应一个个的操作类。

1.5 论文架构

本篇由八个章节组成，具体安排内容如下：

第 1 章主要讨论该课题的研究前提和研究学说，并简要介绍系统的设计思想。

第 2 章主要介绍系统开发所涉及的开发技术，并概述用于开发系统的技术。

第 3 章和第 4 章专门分析了各个方面，一个接一个，研究和分析需求，然后是系统架构图，假设系统的功能结构，最终完成数据库逻辑与物理设计。

第 5 章和第 6 章主要详细设计了系统的功能，增加了每个功能的详细实现效果图和展示关键的代码。

第 7 章基本总结了一般测试，解释了系统开发环境，并实现了测试用例的分析和实现测试等。

第 8 章是本文的最后一部分，它总结了工作，并展望了系统的未来。

1.6 本章小结

本章主要通过了解和分析市场现状，对宠物收养系统的研究背景、应用意义、应用现状作出了分析并对论文整体架构进行了简要的概述。

2 开发技术简介

2.1 SSM

框架是建立在设计模式、反射机制、面向对象程序设计语言以及算法基础上整合而成的一种“半成品”软件，SSM框架是一个结合 Spring、SpringMVC 与 Mybatis 框架的整合，如同一个大型工厂一样，Bean 在整个项目中被组装起来，配置文件里可指定特定的参数用调度实体类构造函数使实例化对象^{错误!未找到引用源。}。其中 Spring、MyBatis 是开源的框架，SpringMVC 是 Spring 中的一些内容。标准的 SSM 框架有四层，分别是 dao 层，service 层，controller 层和 View 层。运用 Spring 实现业务对象管理，运用 Spring 实现业务对象管理，运用 SpringMVC 进行请求转发和视图管理，再结合 mybatis 作为数据对象的持久化引擎。

2.2 MySQL

关于 MySQL，是目前 IT 开发人员最惯用的数据库管理系统，当今业界内最受欢迎的开源关系型的数据库管理系统，也是一个支持多线程高且并发多用户的数据库管理系统^[9]。此外，关联数据库将数据存储于不同的表中，而不是笼统地将所有的数据放在一个大的仓库中，通过这种独特的方式无论是灵活性还是其速度都得到了不小的提升。MySQL 最大的优势在于它是开源代码的数据库管理系统，给予了用户阅读、修改和优化源代码的极高权限^[9]。MySQL 可以在不同的操作系统下运行，也就是具有跨平台性，可以把在一个操作系统中实现的应用程序移植到其他操作系统，并且 MySQL 使用的是标准的 sql 语句，兼容其他系统运行，还支持多种编程语言，如 C 语言、C++等、Java 语言等。

2.3 Bootstrap

Bootstrap会向用户提供一套基于自身的特点较为完整的解决方案。框架的控制权在框架其本身，开发人员要按照框架所规定的一些规范进行开发^[10]。Bootstrap有许多预定义的CSS类，并提供了许多有用的插件，使开发者可以很容易地实现各种通用的这些插件使开发者可以轻松实现各种常见的功能。常见的前端框架有Vue、Bootstrap、React等，既可以在PC端开发，也可以在移动端开发。Bootstrap也是目前最流行的HTML、CSS、JS框架，用于开发响应式布局、移动优先的WEB项目等。它简单而灵活，使开发者可以很容易地进行Web开发。

2.4 JavaScript

JavaScript 简称 JS，是一种已经发展成为一种全面而多样的脚本语言。Javascript 解释器已经成为浏览器的一个基本工具，它被称为 Javascript 引擎，而且是在客户端使用。JavaScript 之所以被开发出来，是因为它被广泛用于网页的开发和构建。各种各样的复杂的计算问题都可由它来处理，其也可实现浏览器和网络服务器之间的互动，并且这个交互可以是动态的^[11]。此外，它在服务器端的作用随着技术的发展也日益明显。JavaScript 和 HTML 之间的交互允许页面行为和风格之间产生松散的耦合效应。在与浏览器交互时，通过监听器来绑定一个特定元素的处理程序，触发某元素对应的处理程序，接着触发对应处理程序来完成任务。

2.5 开发环境及其应用软件

1. 开发技术：Java, SSM, Bootstrap, MySQL。
2. 数据库：MySQL 8.0。
3. 开发平台：IntelliJ IDEA 2019 x64。
4. 开发语言：Java, JavaScript。

2.6 本章小结

本章主要介绍了该宠物收养系统之中所使用到的一些开发技术，开头就先介绍其主要的技术框架 SSM 的特点和优势，然后接着是介绍了 MySQL 数据库的特点以及主要的功能简洁概述，并且还介绍了前端框架 Bootstrap 的开发优点与开发特性等。这一些技术不仅仅被本系统使用，也为本系统的开发与实现都提供了重要的作用。

3 系统可行性和需求分析

3.1 系统可行性分析

在开发这个系统之前，首先可行性分析是以系统的方式进行的，主要是通过分析影响项目的各种因素，利用大量的数据来证明拟议的该项目的可行性，其开发难度的大小，其计划可不可以如期执行。系统可行性分析的主要目的是以最小的成本确定一个问题是否能在尽可能短的时间内得到解决。为了达到这个目的，重要的是要分析可能的解决方案的四个主要方面的优缺点——技术可行性、经济可行性、市场可行性和时间可行性，以进一步确定最初确定的系统规模和目标是否切合实际。

3.1.1 技术可行性

本系统采用 SSM 作为其基础框架，使用了当下行业内非常流行的前后端分离开发模式，使得在开发系统时实现解耦，实现在开发上的专一、高效，框架编写了大量的基础功能，程序员可以将工作重心放到业务逻辑上的实现。与此同时，为了兼容多个平台，本系统在前端开发方面选用了“易用、优雅、灵活、可扩展”的 Bootstrap 框架进行开发，符合了现代管理系统美感的同时又提高了系统的前端开发的效率。而在动态数据存储方面使用了关系型数据库 MySQL，进一步提高了本系统的性能。以上本系统所选取的主要框架技术都是当下较为流行的开发技术，都拥有详细的官方文档和完整的生态社区。

3.1.2 经济可行性

本系统的开发软件主要有 IntelliJ IDEA、SQLyog 等工具。都是开放免费使用。开源项目，满足为本系统客户端的基本功能，又可实现跨多个代码平台，大大减少了开发时对系统设备兼容方面的成本。在开发时所的软件、框架几乎都是免费的，由此分析可知，开发宠物收养平台在经济上是可行的。

3.1.3 市场可行性

通过对广州、东莞、湛江等多地实地考察，并结合网络上人们对宠物收养的需求现状，发现目前许多宠物收养的管理模式依旧停留在传统的管理模式，导致宠物收养资源没得到较好的利用，浪费了人力物力。随着全球互联网信息技术的发展，物联网、云计算、移动互联以及大数据等新兴热点技术的发展都为智慧化城市建设提供了技术基础。在全球的范围之内，智慧化已经成为当今世界不可逆转的历史潮流了。在面

对目前尚未完全成型的宠物收养市场，通过对传统市场的信息化建设，去实现在有限的资源下提高其效率，同时增强宠物收养的安全性和可靠性，进而节省大量的人力和物力资源。

3.1.4 时间可行性

本系统是基于快速原型模型，并通过在需求分析阶段对软件需求的初步但不完全的分析和定义，快速设计和开发软件系统原型，从而大大缩减了本系统的开发时间周期。本系统采用前后端分离的开发模式来实现解耦合并在开发上专一，充分提高了开发效率。本系统的预期开发周期为三个月，而且开发时在距离项目验收仍具有较充分的时间，因此，开发宠物收养系统从时间的可行性上来讲是可以的。

3.2 系统用例分析

通过结合现实生活中的宠物收养业务开发系统功能模块，从系统实际业务流程的角度与业务流程的调查信息进行了衔接，可提供进一步的分析。该系统预期目标为实现一个通过Bootstrap框架开发前端，Java作为后端的宠物收养系统的开发语言。同时将系统分为客户端和管理端两大模块进行需求分析，即用户通过网页进入系统，相应的功能使用在网页上完成，管理员通过后台可以对系统的数据进行分析和管理的。

3.2.1 用户功能需求分析

用户的功能模块主要包括注册登录、个人信息管理、评论、申请宠物收养、志愿者申请、查看活动日志、查看管理员简介。

在用户进入客户端后通过登录进入本系统或先进行注册后再登录进入本系统。登录之后，用户可以点击个人头像去查看个人信息，也可以对密码、性别等详细信息进行修改。用户在进入宠物中心后可以查看宠物的相关信息，并选择进入宠物的详细页面可以对宠物进行收养操作，浏览宠物信息并确定收养，选择性添加备注后再确认申请。用户可以对宠物进行评论发表自己的意见或看法，也可在其他用户或自己的评论下面对评论进行回复。用户在系统内，可以在活动日志页面查看系统内的活动发布情况，通过日期、参与人员和事件内容等详细信息了解活动内容。用户可以通过在志愿者申请模块提交申请，在申请表单内填写详细的个人信息，确认无误后点击提交数据就可以传送到系统后台。用户还可以在前端查看管理员的个人介绍，通过管理员的个人简介用户可以更好的了解系统管理员。用户的用例图如图 3.1 所示。

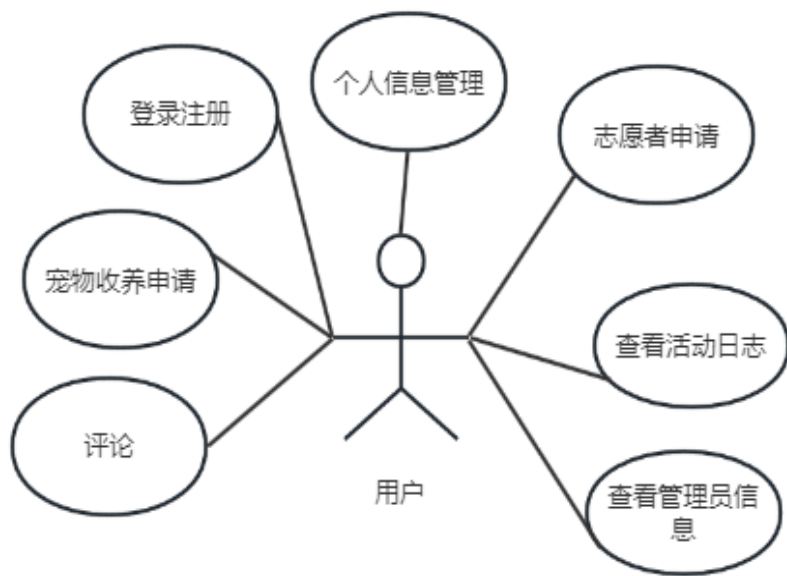


图 3.1 用户用例图

3.2.2 管理员功能需求分析

管理员的功能模块主要的一些包含登录、宠物信息管理、活动日志管理、用户信息管理、管理员信息管理、志愿者申请管理、宠物收养管理以及评论管理等功能管理模块。

管理员通过后台管理进入管理员登录界面后进行登录操作即可进入系统后台进行管理。用户信息管理模块包括系统用户信息查看、用户信息查看添加、用户信息删除和用户信息修改，管理员通过管理员登录，管理员可以查看和修改管理系统中用户的个人信息。宠物信息管理包含宠物信息的查看、宠物信息的增加、宠物信息的删除以及宠物信息的修改等，管理员通过查看宠物信息，来对宠物的相关信息进行操作。管理员信息管理包含对管理员信息的增加、管理员信息的查看、管理员信息的删除以及管理员信息的修改等，管理员通过查看管理员信息，以对管理员的详细信息进行管理操作。志愿者申请模块包括查询、修改和删除功能，用户在前台申请后管理员可以在后台对其进行管理操作。宠物收养审核模块包括同意收养功能、不同意收养功能，管理员通过制定相关审核规则，由用户提交申请后，管理员进行审核。活动日志管理模块包括活动上传功能、删除活动功能、增加活动功能和查找活动功能，以使管理员可对活动信息的查看和统一管理。评论管理模块包括查看评论的功能和删除评论的功能，对于网站内的所有评论，都可被管理员进行统一的管理和监控。管理员用例图如图 3.2 所示。

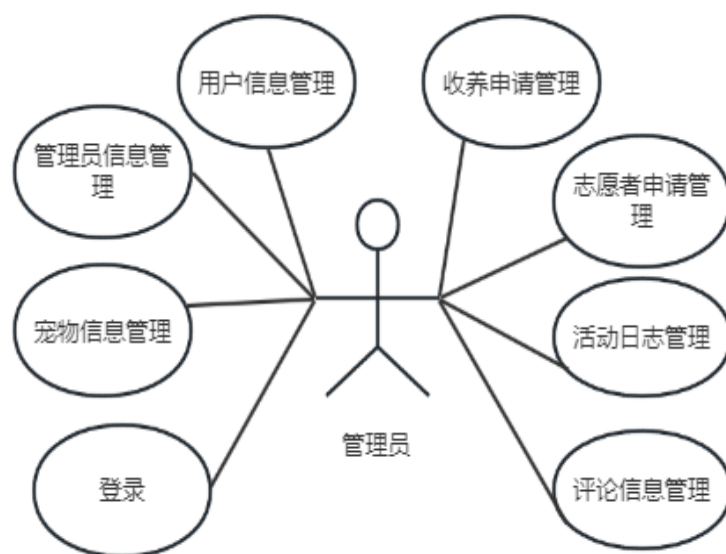


图 3.2 管理端用例图

3.3 非功能性需求分析

易用性：宠物收养管理平台十分注重用户在使用上的体验，图表化管理操作页面的客户端，大大提高了平台用户的易操作性。并且使用整体界面现代化的 UI 设计，并且使得用户界面美观，体现了易用性的三大原则：发现，易懂，效率。

稳定性：系统需要经过大量的测试来模拟上线场景，以确保本系统在正常使用的过程中没有明显的逻辑性错误漏洞导致系统运行异常，且满足系统最基本的业务需求，并使用友好的方式处理系统运行异常的情况。

高效性：高效性是在非功能需求中的重要指标，性能差的系统往往将会导致用户的使用体验变差。客户端在移除繁琐功能之后，减少用户在第一次使用客户端进行交互冗余时间，并且在对待管理端、客户端逻辑性能上，进行了全面优化，让所有来自用户的请求都得到了及时的回应，极大地改善了管理员和用户的操作体验。

3.4 本章小结

本章主要从系统的可行性和需求两大方面并结合本系统的实际需求进行分析和研究，不仅从经济可行性、技术可行性、时间可行性以及市场可行性等各方面对宠物收养管理系统进行多角度深入的分析阐述，并进一步确定了本选题的系统具有较高的可行性，而且也在功能需求的角度上对系统的设计进行了全面的分析和用例图的设计与描述。

4 系统概要设计

宠物收养系统分成用户端和管理端两个部分。在用户端主要向用户提供相关的基础服务，如查询宠物信息、评论和提交宠物收养申请等功能。在管理端主要向系统管理员提供基础服务，如用户信息管理、宠物信息管理、活动信息管理以及评论管理等功能。由客户端和管理端二者相结合，进一步优化了宠物收养的管理模式，意在建立一个高效、实用的宠物收养系统。

4.1 系统设计原则

为了保证宠物收养系统在系统可用性、安全性和稳定性等方面拥有良好的性能，在开发的过程中采用了当前主流、成熟的开发技术。并在系统的设计与开发过程中体现了以下原则^[12]：

4.1.1 可用性

系统以人为本，开发系统的最终目的是用来服务用户的。用户的体验是否友好，界面是否美观以及系统操作是否流畅成为了系统是否具有较好的可用性和亲和力。在现代化管理系统设计当中，开发者更应该站在系统用户的角度去考虑系统的功能、注重人机交互和系统界面设计等。

4.1.2 安全性

随着全球计算机网络技术的不断发展，加快了信息化建设的脚步，同时互联网安全等问题也日渐突出。在开发本平台系统时由于存在后台管理端以及不同角色之间的鉴权问题，鉴于本系统是前后分离端的开发模式，所以在后端的数据接口进行登录判断及鉴权拦截。用户处于未登录状态无法通过接口对管理端系统进行访问，为不同的角色分配了不同的管理权限。

4.1.3 稳定性

想要提高软件系统的稳定性，需要以运维服务是基于对软件系统的定期检查，以帮助客户发现软件中隐藏的问题，并尽可能地减少问题发生的频率。通过不断对本系统的进行测试、调试，尽量保证系统各服务可靠的基础之上，在系统出错的情况之下依然能够平稳的运行。

4.2 数据库设计

数据需求分析是从对数据进行组织与存储的角度，从用户视图出发，分析与辨别相应领域内所管理的各类数据项与数据结构^[13]，其构成了数据字典的主要内容。针对

宠物收养系统，分别对用户模块、管理员模块进行详细的分析，总结出如下的主要需求表格信息：用户表、管理员表、宠物表、评论表、回复表、日志表。他们分别的功能是：

1. 用户表：用来存储用户信息。
2. 管理员表：用来存储管理员信息。
3. 宠物表：用来存储宠物信息。
4. 评论表：用来存储评论所有信息以及状态。
5. 回复表：用来存储评论回复信息。
6. 活动日志表：用来存储活动信息。

4.2.1 数据库的概念结构设计

在对本系统的各个功能模块进行分析之后，再结合各功能数据可得知其实体-联系之间的关系模式，各个都管理员可以对多个评论进行管理的操作，每一个的管理员都可以对多个宠物信息进行管理操作，每个管理员都能够对多个用户进行管理操作，一位管理员可以对若干个的活动进行操作，其宠物收养系统的实体关系 E-R 图如图 4.1 所示。

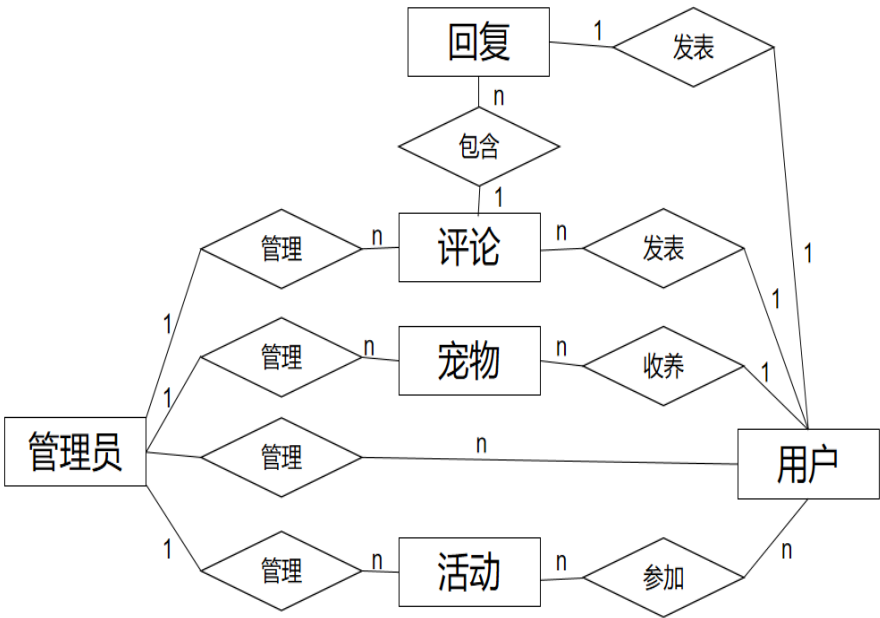


图 4.1 系统 E-R 图

每一个的管理员都有属于自己的个人信息，一个用户的信息可以被好几个管理员同时进行管理，而且一个宠物的信息也能被若干个管理员同时管理，一条评论可以被每个管理员管理，一个活动可以被所以管理员编辑发布。一个用户对应一个属于自

己的用户信息，一个用户又能同时对几个宠物进行评论，同时也可以对几个个评论同时回复，一个用户也能同时对好几个活动日志进行查看，一个用户可以对多个宠物进行领养，而一个宠物只能被一个用户领养。

管理员实体的属性主要为：管理员 id、邮箱、名字电话、生日、简介、密码、头像。

用户实体的属性主要为：用户 id、邮箱、头像、名字、电话、生日、简介、头像、密码。

评论实体的属性主要为：评论 id、用户 id、宠物 id、评论内容、评论时间。

活动实体的属性主要为：活动 id、人员、地点、时间、事件。

宠物实体的属性主要为：宠物 id、宠物生日、宠物类型、宠物状态、宠物备注。

管理员实体的主要设计就是对用户、宠物、评论、活动等实体进行管理，管理员可以对系统内多个实体的内容来管理操作，管理员关系 E-R 图如下图 4.2 所示。

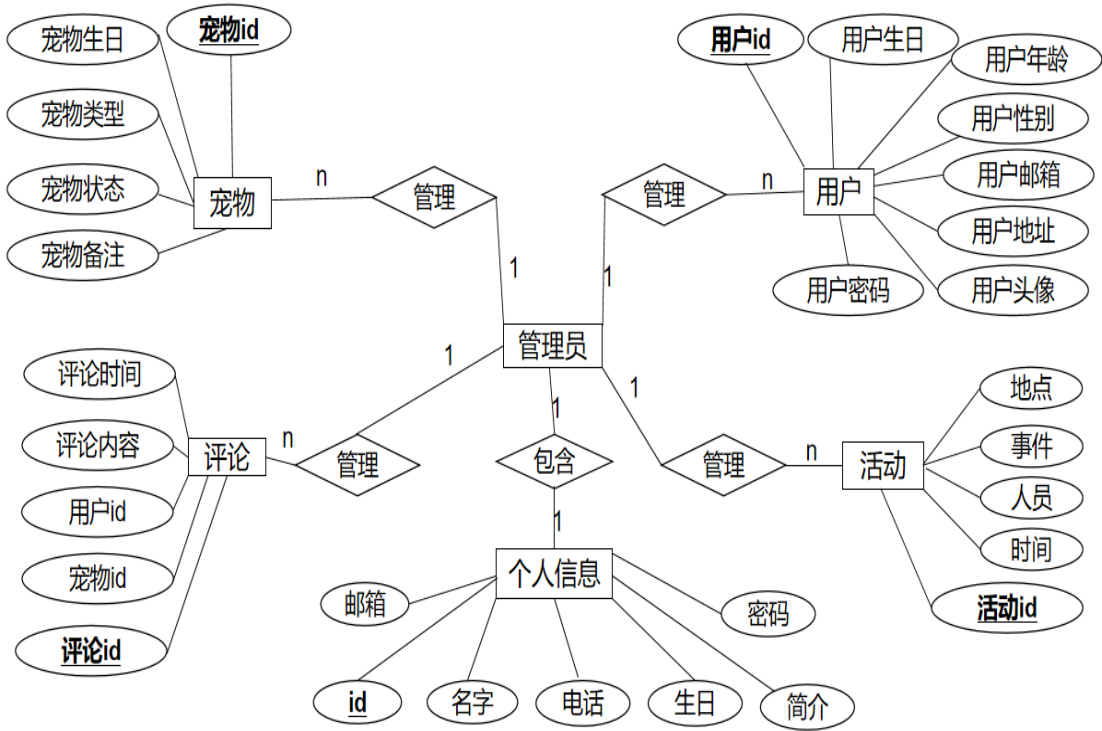


图 4.2 管理员关系 E-R 图

用户实体的设计主要就是对宠物、评论、活动等实体的进行相关操作，用户可对系统内的多个实体内容进行使用操作，用户关系 E-R 图如下图 4.3 所示。

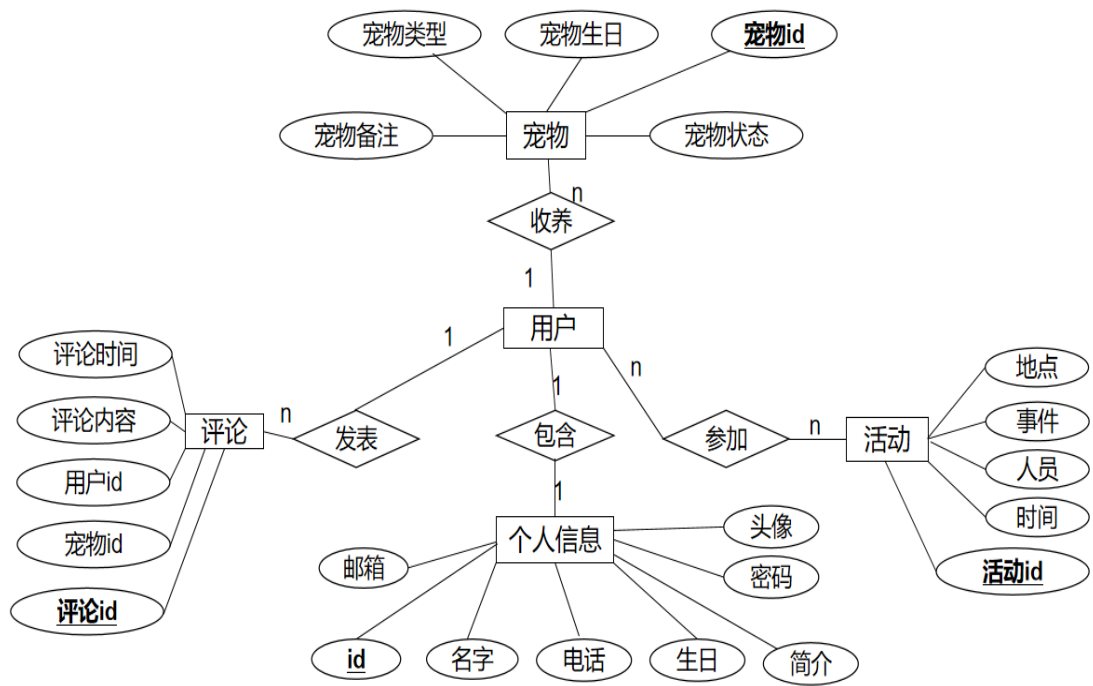


图 4.3 用户关系 E-R 图

4.2.2 数据库的逻辑结构设计

在本系统在设计、开发的过程里创建并使用的数据表较多，其中包含用户信息表等。本章仅用本宠物收养系统中几张较为主要的数据表来进行说明与阐述。

(1) User 表

User 表为用户表，包括 id、用户名、密码、性别、年龄，电话，邮箱，地址，头像，状态。管理用户表详细字段如表 4.1 所示。

表 4.1 user 表

| 列名 | 数据类型 | 长度 | 非空 | 主键 | 注释 |
|-----------|---------|-----|----|----|--------------------|
| id | int | 11 | √ | √ | 此为 id |
| userName | varchar | 20 | √ | | 此为用户姓名 |
| password | varchar | 30 | √ | | 此为用户密码 |
| sex | varchar | 2 | | | 此为性别 |
| age | int | 11 | | | 此为年龄 |
| telephone | varchar | 20 | | | 此为电话 |
| Email | varchar | 30 | | | 此为邮箱 |
| address | varchar | 50 | | | 此为地址 |
| pic | varchar | 100 | | | 此为头像 |
| state | Int | 11 | | | 0 是没有收养宠物，1 是有收养宠物 |

(2) Admins 表

Admins 表为管理员表，包括 id、管理员名、密码、性别、年龄、电话、生日、邮箱，地址，头像，标记。管理用户表详细字段如表 4.2 所示。管理员表详细字段如表 4.2 所示。

表 4.2 admins 表

| 列名 | 数据类型 | 长度 | 非空 | 主键 | 注释 |
|-----------|---------|-----|----|----|----------|
| id | int | 11 | √ | √ | 此为 Id |
| adminName | varchar | 20 | √ | | 此为管理员的名字 |
| adminPwd | varchar | 20 | √ | | 此为密码 |
| realName | varchar | 20 | √ | | 此为真实的名字 |
| telephone | varchar | 20 | √ | | 此为电话 |
| Email | varchar | 20 | √ | | 此为邮箱 |
| birthday | date | | √ | | 此为生日 |
| sex | varchar | 3 | | | 此为性别 |
| pic | varchar | 100 | √ | | 此为头像 |
| remark | varchar | 255 | | | 此为标记 |

(3) Pet 表

Pet 表为宠物信息表，包括 id、宠物名、宠物类型、性别、生日、头像、状态、备注。宠物表详细字段如表 4.3 所示。

表 4.3 pet 表

| 列名 | 数据类型 | 长度 | 非空 | 主键 | 注释 |
|----------|---------|-----|----|----|------------------------|
| id | int | 11 | √ | √ | 此为 Id |
| petName | varchar | 20 | √ | | 此为宠物的名字 |
| petType | varchar | 20 | √ | | 此为宠物类型 |
| sex | varchar | 3 | √ | | 此为性别 |
| birthday | date | | √ | | 此为生日 |
| pic | varchar | 20 | √ | | 此为头像 |
| state | int | 11 | √ | | 0 还未被收养，1 被申请收养 2 已被收养 |
| remark | varchar | 100 | | | 此为备注 |

(4) Comment 表

Comment 表为评论表，包括 id、用户 id、管理员 id、宠物 id、评论时间、评论内容。评论表详细字段如表 4.4 所示。

表 4.4 comment 表

| 列名 | 数据类型 | 长度 | 非空 | 主键 | 注释 |
|--------|------|----|----|----|---------|
| id | int | 11 | √ | √ | 此为 Id |
| userId | int | 11 | | | 此为用户 id |

续表 4.4

| | | | | | |
|-------------|---------|-----|---|--|----------|
| adminsId | int | 11 | | | 此为管理员 id |
| petId | int | 11 | | | 此为宠物 id |
| commentTime | date | | √ | | 此为评论时间 |
| content | varchar | 200 | √ | | 此为评论的内容 |

(5) Apply 表

Apply 表为回复表，包括 id、名字、邮箱、年龄、电话、消息、回复时间、状态。
回复表详细字段如表 4.5 所示。

表 4.5 apply 表

| 列名 | 数据类型 | 长度 | 非空 | 主键 | 注释 |
|-----------|---------|-----|----|----|--------|
| id | int | 11 | √ | √ | 此为 Id |
| name | varchar | 10 | √ | | 此为姓名 |
| email | varchar | 20 | √ | | 此为邮箱 |
| age | int | 11 | √ | | 此为年龄 |
| telephone | varchar | 12 | √ | | 此为电话号码 |
| message | varchar | 100 | √ | | 此为信息 |
| applyTime | date | | | | 此为回复时间 |
| state | int | 11 | | | 此为状态 |

(6) Blog 表

Blog 表为日志表，用于记录活动信息，包括 id、时间、地址、人员数、事件、标题。日志表详细字段如表 4.6 所示。

表 4.6 blog 表

| 列名 | 数据类型 | 长度 | 非空 | 主键 | 注释 |
|------------|---------|-----|----|----|-------|
| id | int | 11 | √ | √ | 此为 Id |
| actionTime | varchar | | √ | | 此为时间 |
| address | varchar | 100 | √ | | 此为地址 |
| peoples | int | 100 | √ | | 此为人员数 |
| event | varchar | 100 | √ | | 此为事件 |
| title | varchar | 20 | √ | | 此为标题 |

4.3 本章小结

本章主要于宠物收养系统进行需求分析后把所需要考虑的系统设计因素等进行了简洁的阐述，本宠物收养系统在设计与开发的过程中创建了若干张数据表，其中包含用户信息表和管理员信息表等，在本章在列举仅借用本系统中 6 张核心数据表进行说明，并且对本系统进行了系统设计和数据库设计，其中数据库设计包括概念结构设计、逻辑结构设计，而且详细清晰地阐述了系统设计原则。

5 系统详细设计

5.1 系统运行环境

5.1.1 硬件环境

在系统开发阶段所使用的操作系统为 Windows10 进行开发与调试，硬件为：处理器(i5-9300H)，GPU (GTX1650)，ROM(1TB SSD)，机带 RAM(16GB)。

5.1.2 软件环境

本系统在开发的过程之中所使用的开发环境为：关系型数据库 MySQL (5.7)，Java 开发工具 (JDK1.8)，Web 服务器 (Tomcat8.5)。

本系统在开发的过程之中所使用的开发工具为：SQlyog、IntelliJ IEDA、Visual Studio Code，浏览器为 Google Chrome。

5.2 系统运行设计

5.2.1 运行模块组合

此系统的开发框架使用 SSM (SpringMVC、Spring 以及 Mybatis) 进行开发，并且通过 Bootstrap 框架以实现前端页面搭建，总体采用工程模块化、组件化的新潮开发思想。数据传输是通过异步请求实现，页面内容的改变并不需要反复重新加载整个页面，避免了不必要的跳转和重复加载，给用户带来了较好的体验。工程模块化和组件化的思想，把相同模块封装为组件，增强了组件的复用性，大幅提高了系统的开发效率。

5.2.2 运行控制

在管理 PC 端对页面进行了现代化管理系统设计，在不过度影响网页性能的同时又添加了轮播图，提高了用户的体验。在客户端方面也通过对页面布局进行了设计，更加符合客户端用户操作逻辑。

5.3 系统结构设计

5.3.1 系统结构设计概述

此宠物收养系统的目的是设计实现一个效率较高和功能齐全的宠物收养系统。本系统可以实现用户的登录、评论和收养宠物等主要功能。管理员在系统里可以管理用户信息、宠物和其他信息并进行其他操作。

5.3.2 系统架构的设计

总体的逻辑设计是我们在系统的整体架构的设计中特别值得重视的一点，同时对系统使用的 B/S 架构的开发模式我们也需要关注，此外，值得关注的是应用层和业务层之间在数据对接上的一些问题。

系统在用户层有管理员和用户两个角色，两者皆是通过浏览器进入系统的前台或后台页面对系统进行操作，并通过 Ajax 请求在页面上对相关数据进行交互。本系统使用 SSM 框架进行设计开发，而且应用层以及业务层皆是基于 Java 语言的，并且在后端的数据的存储与管理上应用了 MySQL 数据库。系统总体的架构如图 5.1 所示。

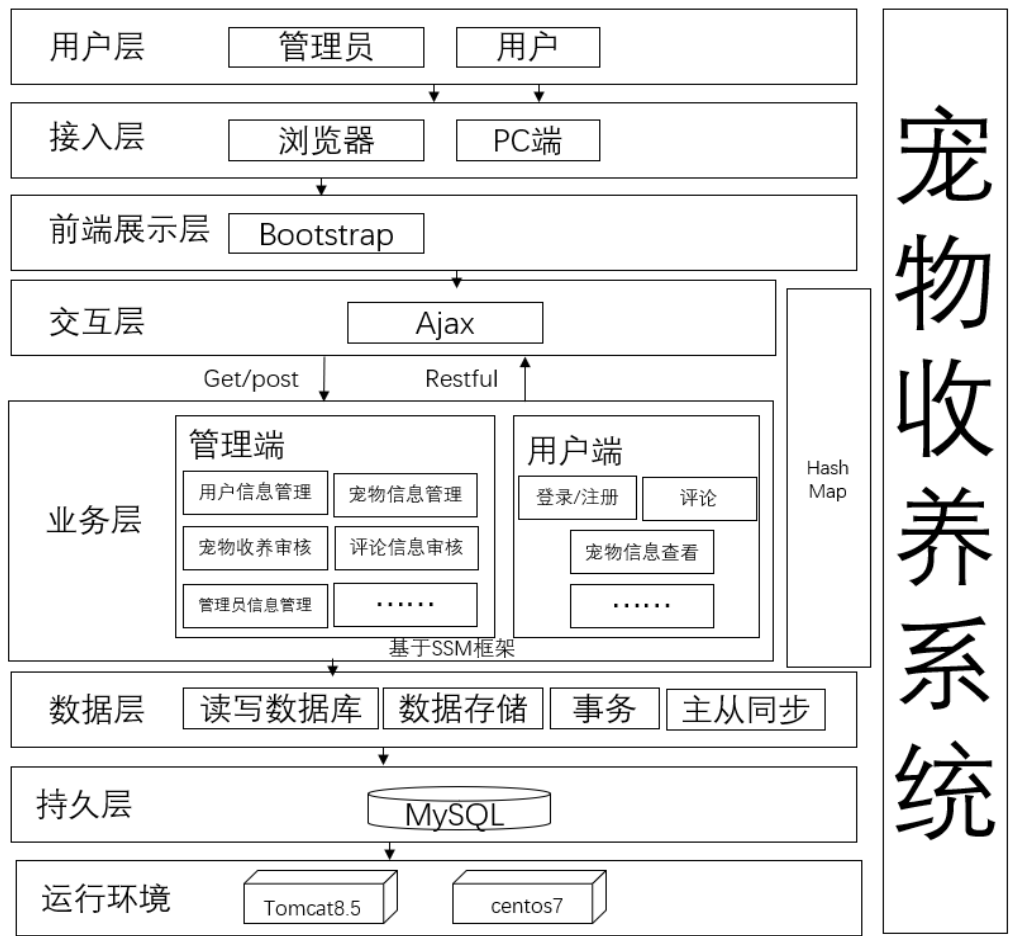


图 5.1 系统架构图

5.3.3 系统功能模块图

功能结构图将系统的功能模块分解成更小的部分来实现，而信息图则可根据功能的从属关系来表示。管理信息系统的子系统可以被看作是总系统下的功能，对于每一个功能，都能继续分解出子层次。

宠物收养系统的功能结构分为用户模块与管理模块，用户和管理员可以通过他们的账户登录，来进入各自的系统模块。用户只有在注册成功之后才能登录进行对宠物的浏览和收养，注册后的用户需要输入其用户名以及其设定的密码才可进行系

统的登录;管理员输入其管理员名及相对应的密码后进入系统进行管理操作,管理系统内的宠物信息与用户信息等。宠物收养系统的总体功能模块图如图 5.2 所示。

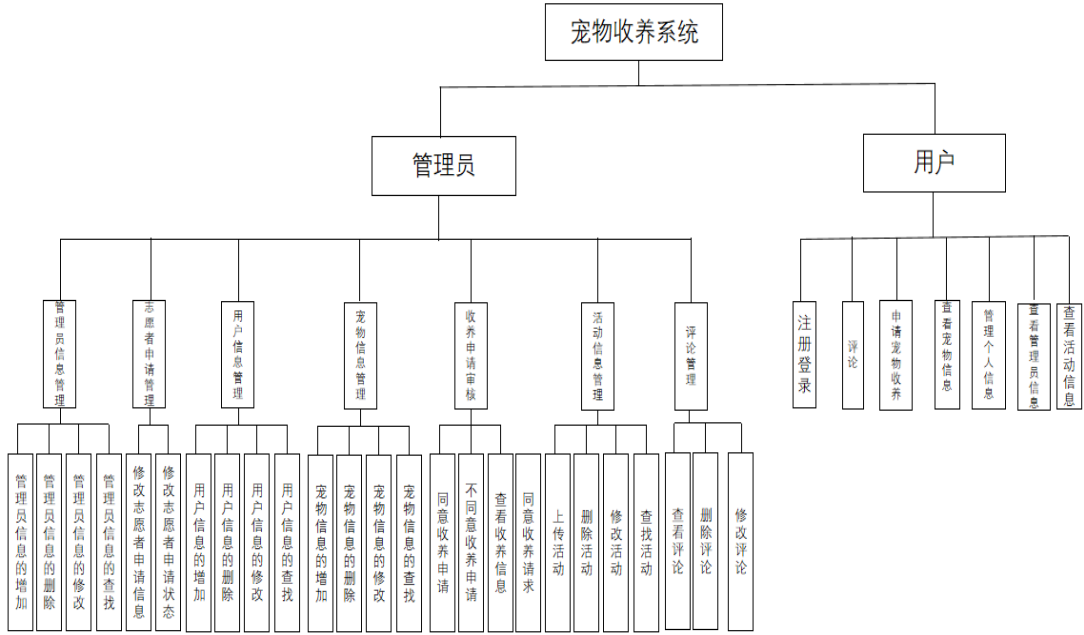


图 5.2 宠物收养系统的总体功能模块图

5.4 系统管理端的功能模块设计

5.4.1 信息管理的模块设计

信息管理的模块包含了用户管理、宠物管理、管理员管理这三个模块。

用户信息管理模块的功能实现了关于对系统内的用户们的信息管理操作。管理员可以在此模块对用户信息进行添加账号操作，修改用户相关信息操作，如修改姓名、生日等，删除用户账号信息与通过模糊查询来查找用户信息。用户管理的具体时序图如图 5.3 所示。

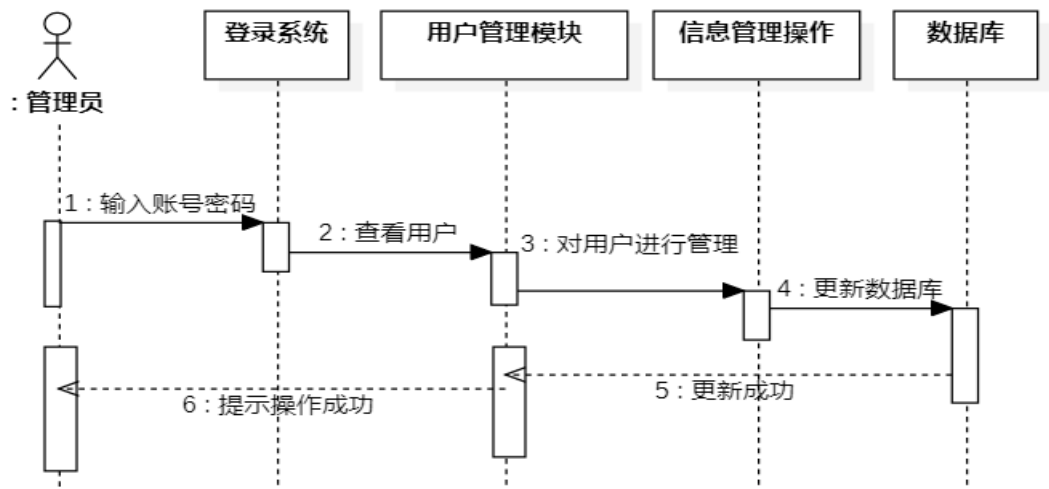


图 5.3 用户管理时序图

而宠物管理模块则与上述模块类似，实现了对系统内宠物信息的相关管理操作。管理员可在后端通过该模块对宠物信息进行增加，并可以对宠物信息如种类等进行修改，还可以删除该宠物，并能通过关键字查询来查找该宠物的信息。用户管理的具体时序图如图 5.4 所示。

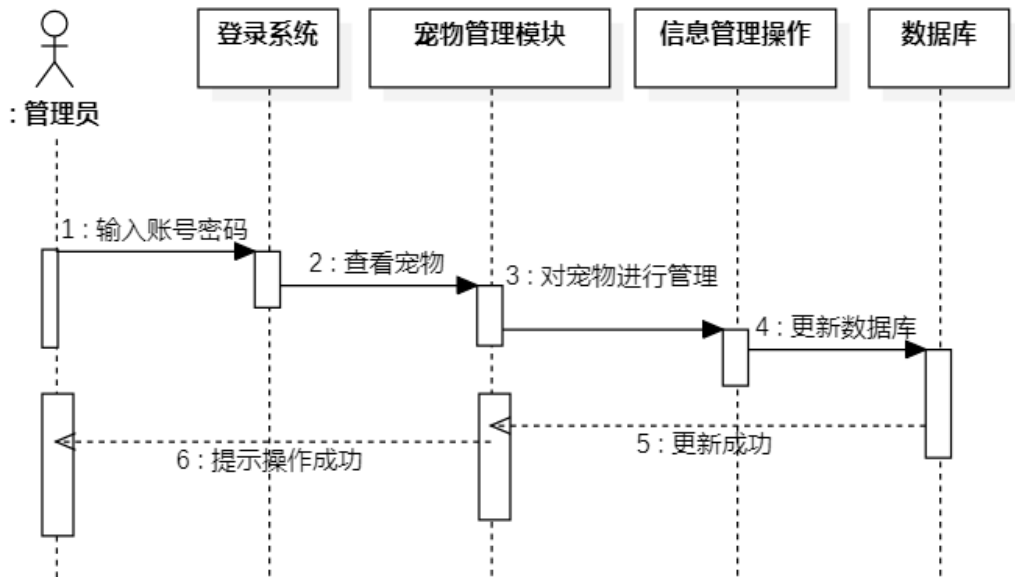


图 5.4 宠物管理时序图

管理员模块则是对管理员信息的增加、删除以及修改、查找功能操作的实现。用户管理的具体时序图如图 5.5 所示。

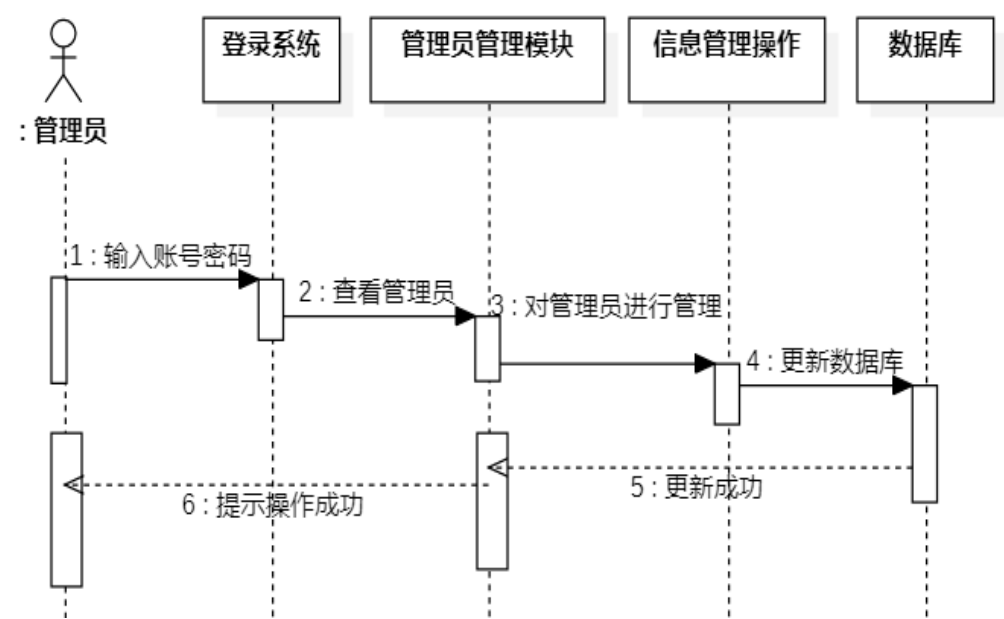


图 5.5 管理员管理时序图

信息管理的具体的流程图如图 5.6 所示。

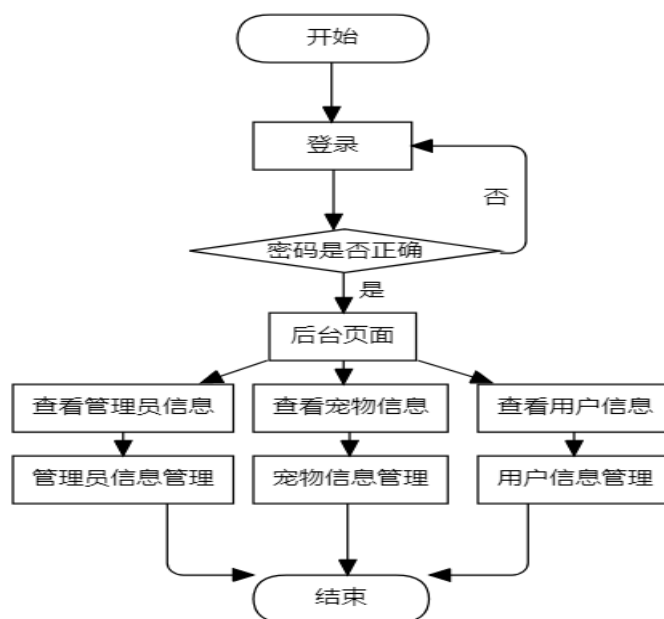


图 5.6 信息管理流程图

5.4.2 活动日志管理模块设计

管理员通过登录进入系统后台之后可以对活动日志进行相关的管理操作，修改操作如修改其地点、事件等数据，也可在此增加一项或多项活动，还能选择删除活动或通过日期对活动日志进行搜索。

具体的活动日志管理活动图如图 5.7 所示。

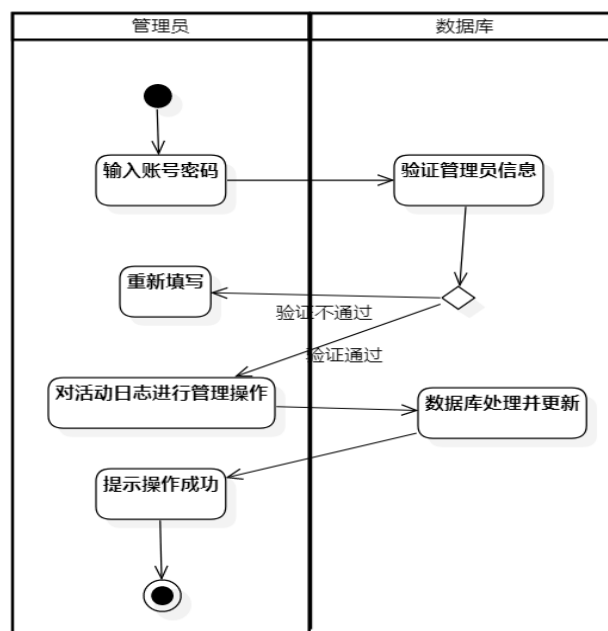


图 5.7 管理员活动日志活动

活动日志管理时序图如图 5.8 所示。

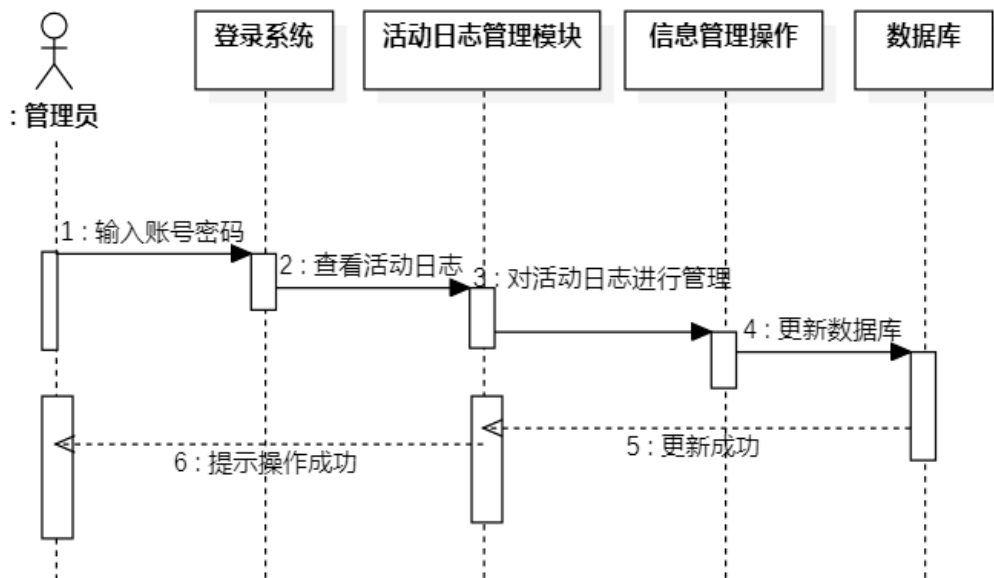


图 5.8 管理员活动日志时序图

活动日志管理流程图如图 5.9 所示。

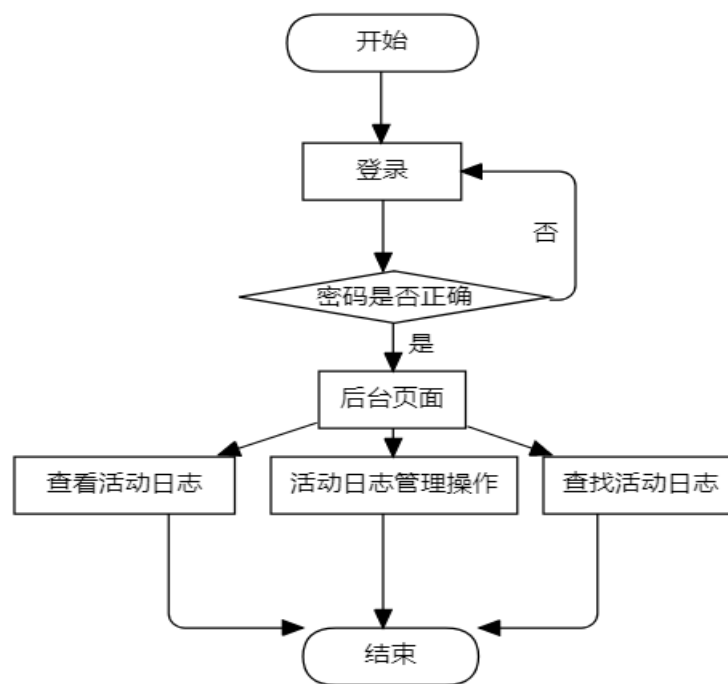


图 5.9 管理员活动日志流程图

5.4.3 收养审核模块设计

管理员可以由登录系统后台来管理收养申请的审核，用户在前台通过收养模块查看并挑选宠物，提交宠物的收养申请之后上传到数据库，然后系统将收养申请展示到系统后台，管理员可在本模块对收养申请进行查看后进行审核，选择同意申请或不同意申请。收养申请审核活动图如图 5.10 所示。

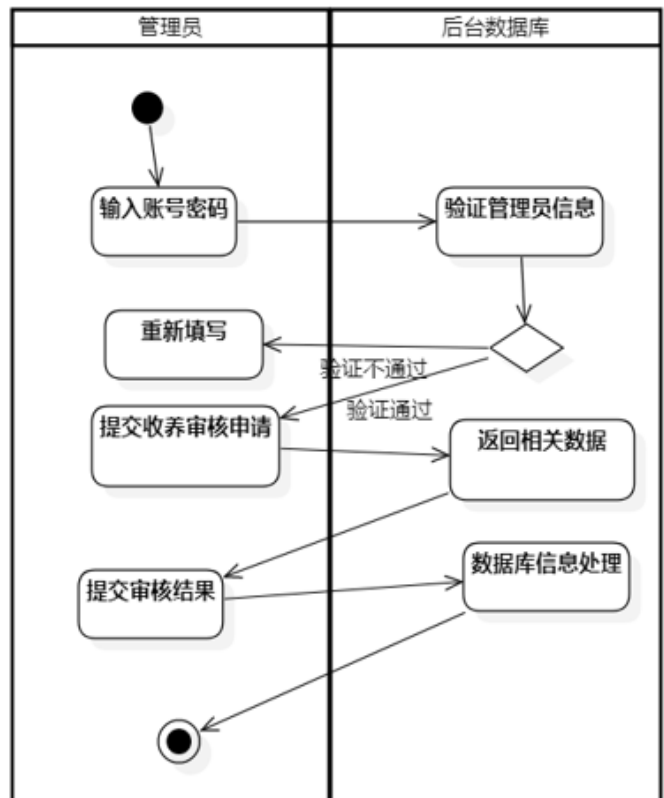


图 5.10 管理员收养审核活动图

收养申请审核时序图如图 5.11 所示。

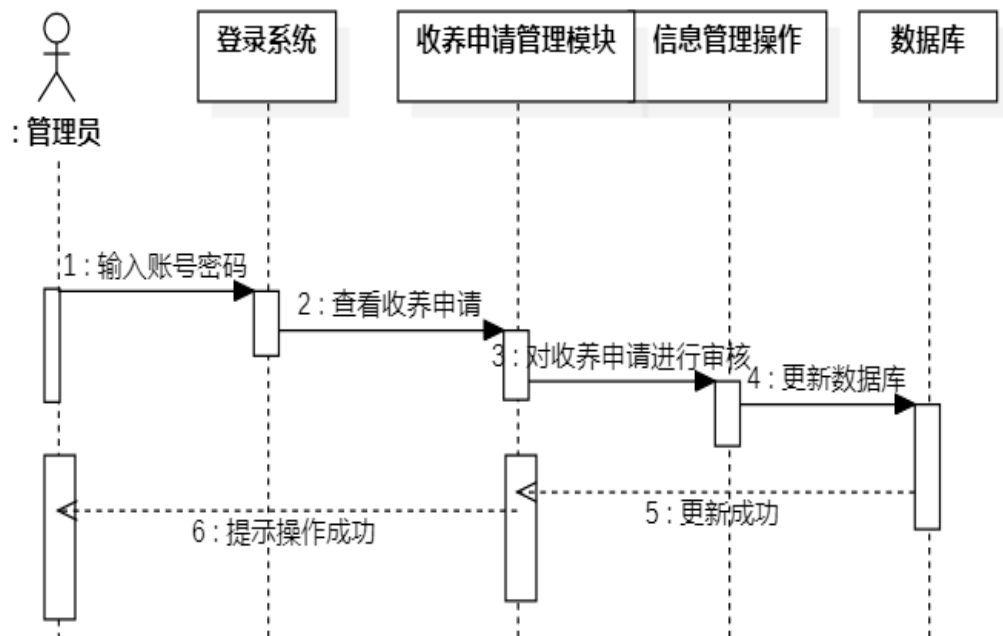


图 5.11 管理员收养审核时序图

收养申请审核流程图如图 5.12 所示。

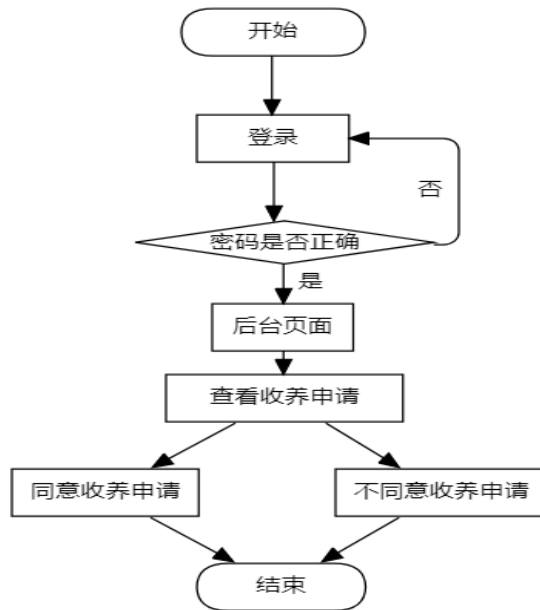


图 5.12 管理员收养审核流程图

5.4.4 评论管理模块设计

管理员在登录进入系统后台之后，可以在评论管理模块进行对评论相关的管理操作，修改操作在此可实现，如有影响言论环境的不良评论，还能选择删除该评论。

评论管理的活动图如图 5.13 所示。

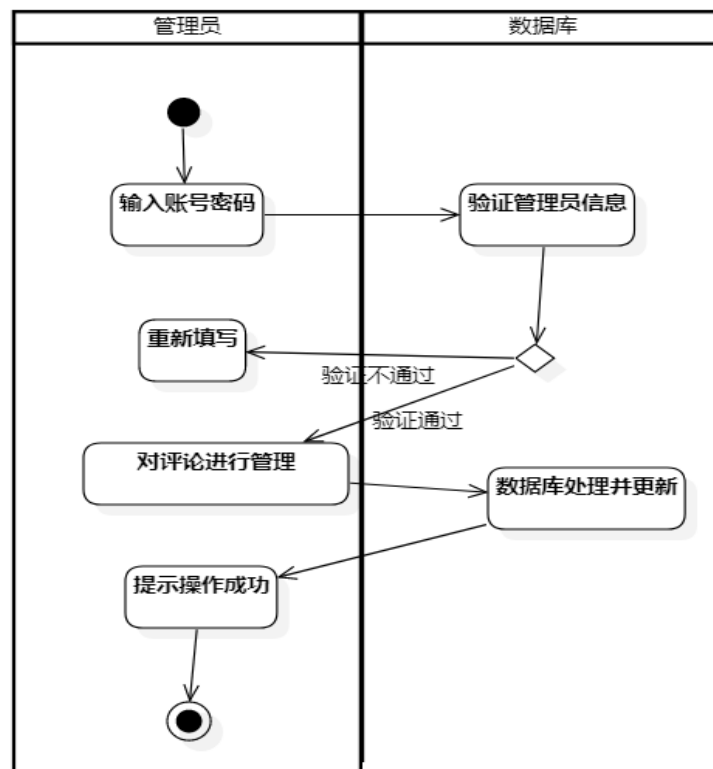


图 5.13 管理员评论管理活动图

评论管理的时序图如图 5.14 所示。

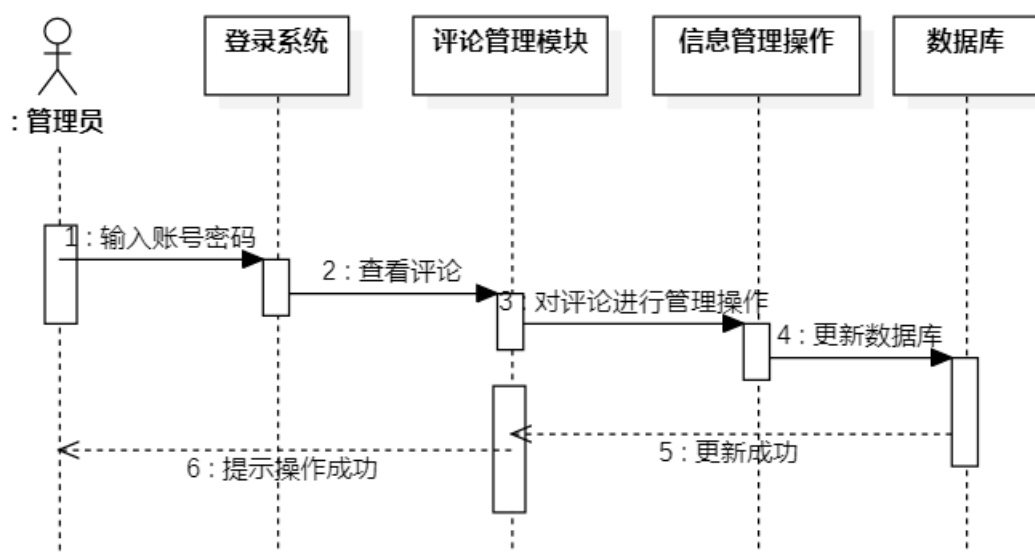


图 5.14 管理员评论管理时序图

评论管理的流程图如图 5.15 所示。

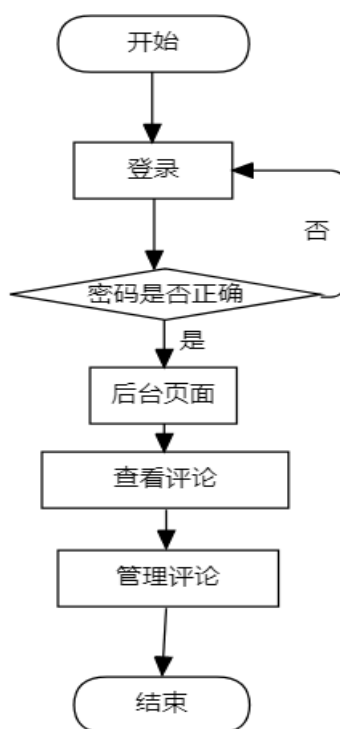


图 5.15 管理员评论管理流程图

5.5 系统用户端的功能模块设计

5.5.1 宠物收养模块

用户在登录进入用户页面之后,可以在宠物收养模块对宠物进行收养操作,在宠物中心查看浏览宠物并查看宠物的详细的信息之后对宠物进行收养申请提交到系统

后台。用户宠物收养申请活动图如图 5.16 所示。

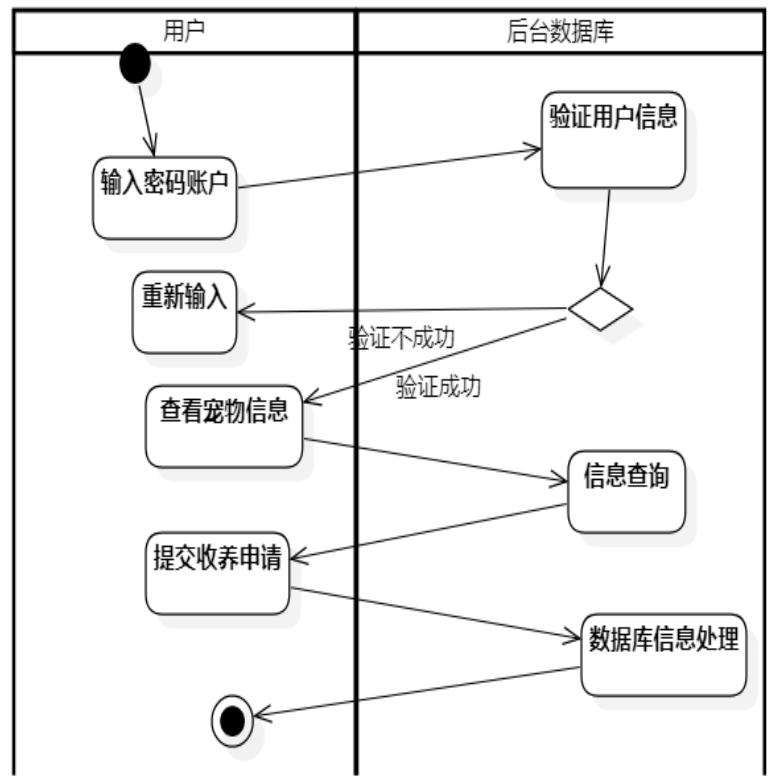


图 5.16 宠物收养申请活动图

用户宠物收养申请时序图如图 5.17 所示。

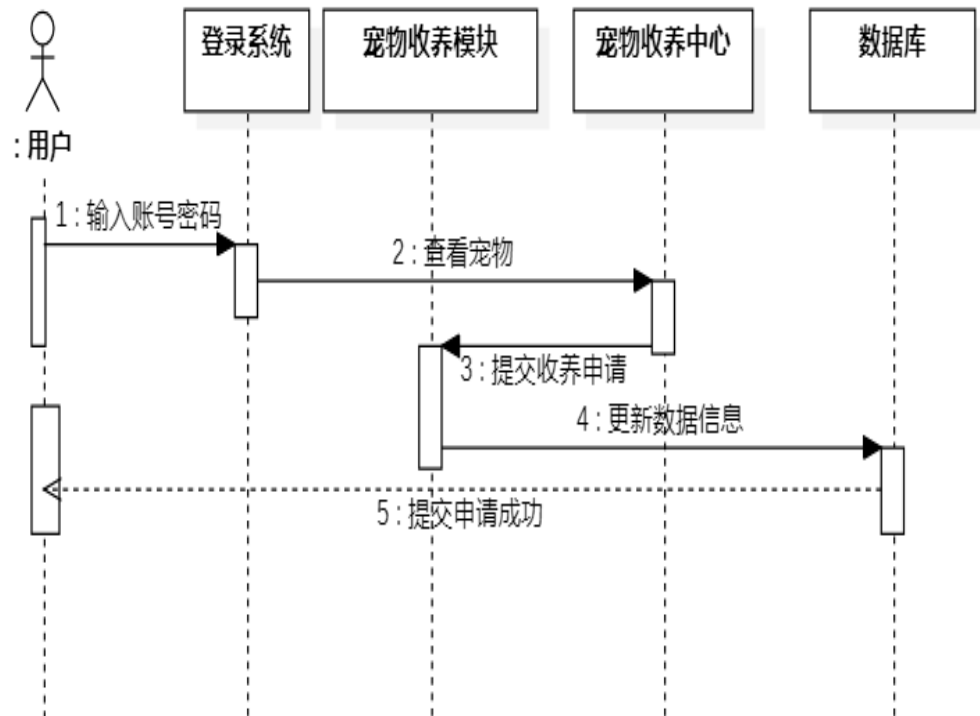


图 5.17 宠物收养申请时序图

用户宠物收养申请流程图如图 5.18 所示。

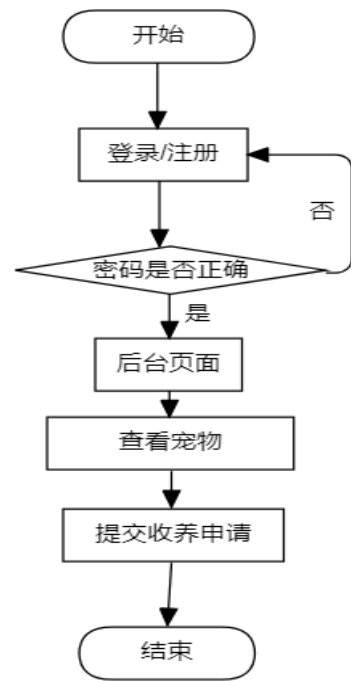


图 5.18 宠物收养申请流程图

5.5.2 用户的评论模块

用户可以在评论模块对宠物进行评论增加宠物的讨论热度为其他人进行答疑等，进入用户页面之后可以在宠物的详细页面下进行评论也可对他人的评论进行回复。用户评论的具体活动图如图 5.19 所示。

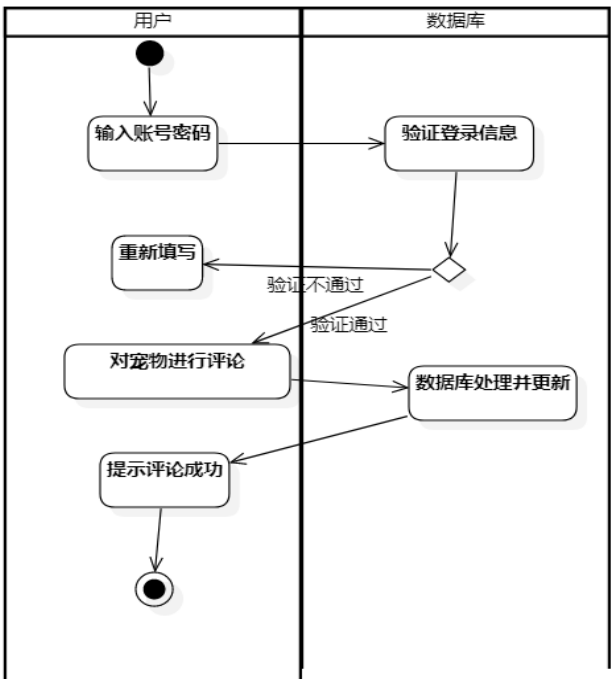


图 5.19 用户评论活动图

用户评论的具体时序图如图 5.20 所示。

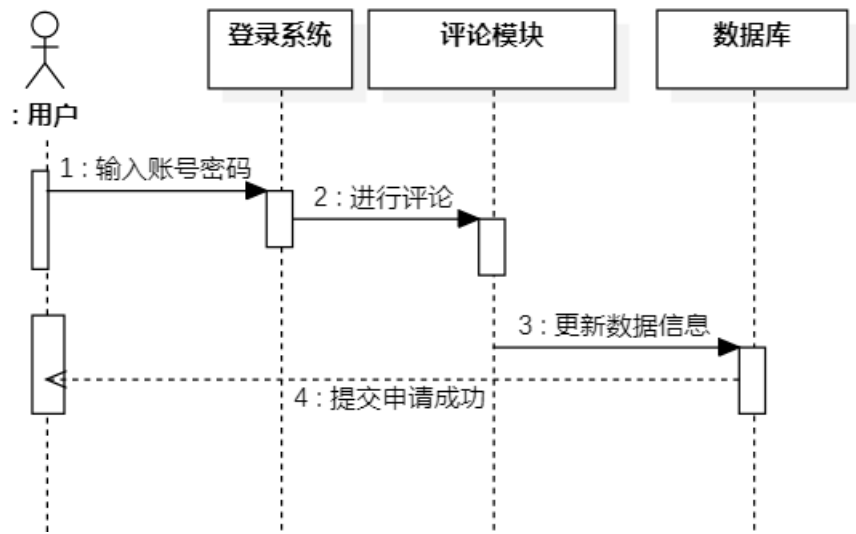


图 5.20 用户评论时序图

用户评论的具体流程如图 5.21 所示。

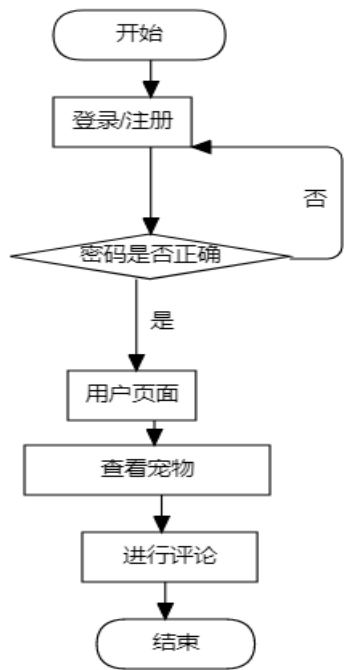


图 5.21 用户评论流程图

5.6 本章小结

在本章节中，对本宠物收养系统的详细设计进行说明性阐述，包括了整体系统开发环境和运行的部署环境，并且通过流程图的形式将此系统以功能模块的方式进行拆分设计和分析并对通过活动图与时序图对重要模块进行阐述。

6 系统功能实现

6.1 用户端

6.1.1 用户登录注册及首页

用户在用户端进行登录。登录功能的实现原理为：

首先，前端采用 JavaScript 编写对用户名及密码格式进行验证。验证通过后，传输到后端，让 service 对数据进行事务处理，然后再 mybatis 中通过 mapper 里面的 SQL 语句和采用 jdbc 驱动访问数据库的 user 表，将字段作为查询条件进行数据查询。在 service 中作出判断，若查询结果为空，则会提示用户重新输入。首页如图 6.1 所示，登录界面如图 6.2 所示。

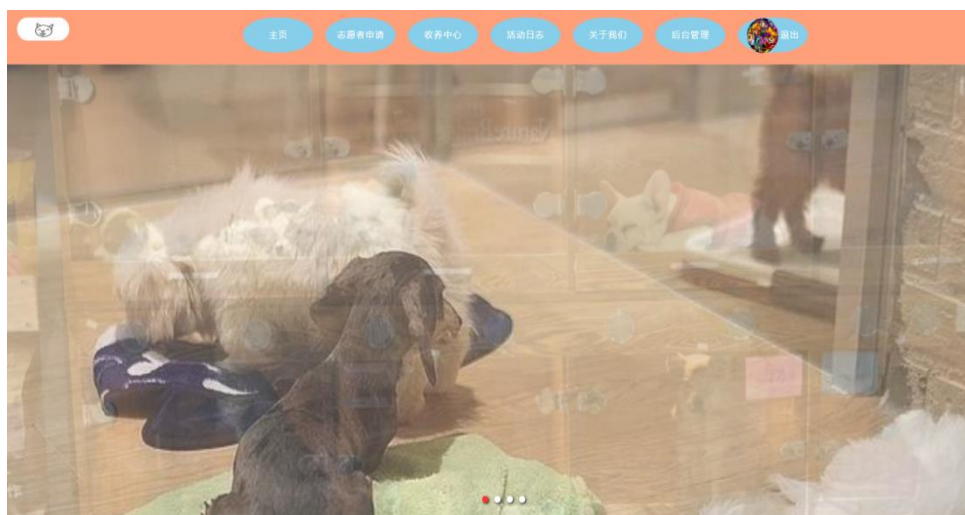


图 6.1 首页

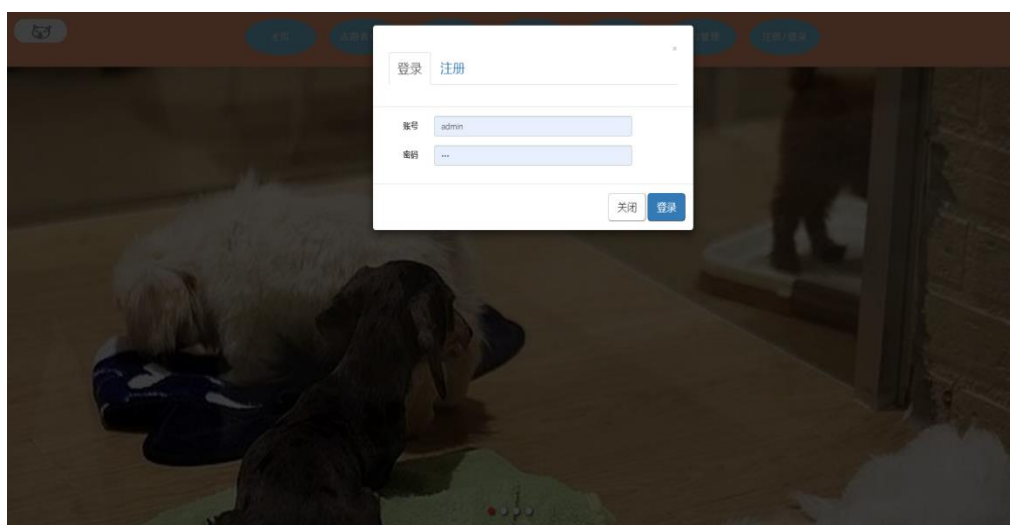


图 6.2 用户登录界面

关键代码实现如下：

```
public Message loginuser(HttpSession session, String
userName, String password){
    User user = userService.loginuser(userName, password);
    if (user != null){
        session.setAttribute("user",user);
        return Message.success().add("user",user);
    }else{
        return Message.fail();
    }
}
```

6.1.2 用户查看个人信息及收养宠物

用户登录成功进入系统后,可以点击头像查看个人信息并对用户名、密码、年龄、性别、电话号码等进行修改。个人信息界面如图 6.3 所示。



The image shows a user profile page. On the left is a circular, colorful anime-style avatar. To the right is a form titled '用户个人信息修改' (Edit User Personal Information). The form contains several input fields: '用户名:' (Username) with the value 'kun', '密码:' (Password) with three dots, '年龄:' (Age) with the placeholder '年龄', '性别:' (Gender) with a dropdown menu showing '男' (Male), '电话密码:' (Phone Password) with the placeholder '电话', 'Email:' with the placeholder 'Email', '地址:' (Address) with the placeholder '地址', and '有无收养经历:' (Whether there is a history of adoption) with a dropdown menu showing '有收养经历' (Has adoption history). At the bottom right of the form are two buttons: '提交修改' (Submit Modification) and '返回主页' (Return Home).

图 6.3 个人信息界面

关键代码实现如下：

```
public Message update(User user) {
    Integer update = userService.update(user);
    if(update > 0){
        return Message.success();
    }else {
        return Message.fail();
    }
}
```

用户在成功登录进入系统后，在收养中心的页面可以让用户在此浏览宠物信息，查看与宠物相关的数据，进而选出自己心仪的宠物，之后可以向系统申请收养。收养中心界面如图 6.4 所示。



图 6.4 收养中心界面

关键代码如下：

```
public Message adopts(@RequestParam(required = false) String
adoptTime, @RequestParam(required = false, value = "pn") Integer
pageNum) {
    int pageSize = 3 ;
    if(pageNum == null){
        pageNum = 1;
    }
    PageInfo<AdoptAnimal> pageInfo =
adoptAnimalService.all(adoptTime, pageNum, pageSize);
    return Message.success().add("pageInfo", pageInfo);
}
```

6.1.3 用户评论

用户在登录进入系统之后，用户可以在宠物等评论区进行评论，可以直接进行评论输入文字，也可在其他用户的评论下回复也可以回复其他用户的评论。评论界面如图 6.5 所示。



图 6.5 评论界面

关键代码如下：

```
@RequestMapping("/update")
@ResponseBody
public Message update(Comment comment) {
    int update = commentService.update(comment);
    if(update>0) {
        return Message.success();
    } else {
        return Message.fail();
    }
}
```

管理员进入系统的管理页面，管理员能够新增、信息和删除宠物信息以及用户，还可以对活动的设置，对评论进行管理，且页面与前端的页面不同。

6.2 管理员管理

管理员在通过首页的后台管理进入管理员登录界面之后，并进行管理员登录的验证，如果输入正确的账号与密码，就可以进入后台管理系统中的信息，否则管理员登录失败。管理员登录界面如图 6.6 所示。

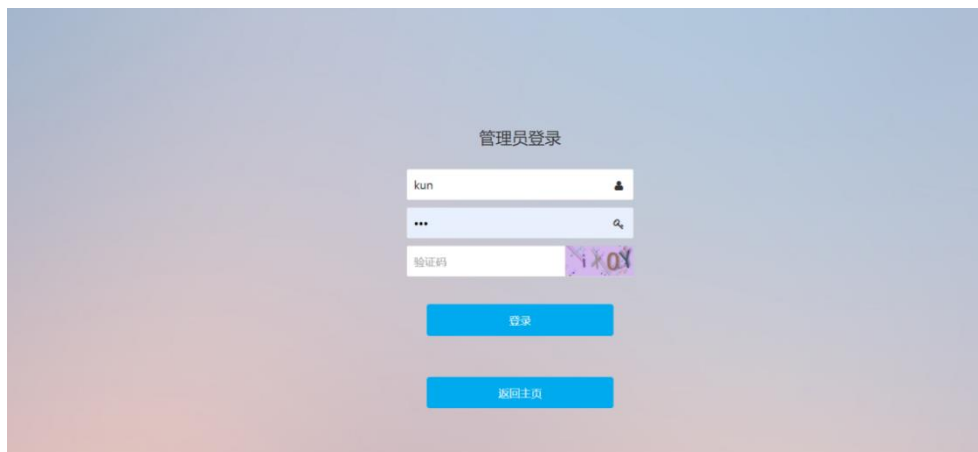


图 6.6 管理员登录界面

关键代码如下：

```
public String doLogin(HttpSession session, String
adminName, String adminPwd) {
    Admins login = adminService.login(adminName, adminPwd);
    if (login == null) {
        session.setAttribute("msg", "用户名或密码错误!");
        return "redirect:/admin/login";
    }
    session.setAttribute("admin", login);
    return "redirect:/admin/admin";
}
```

登录进入后端管理后，管理员可以通过后端对系统的管理员信息进行管理，这是宠物收养系统较为核心的模块，在这个模块实现了对系统内管理员信息的管理。管理员管理界面如图 6.7 所示。

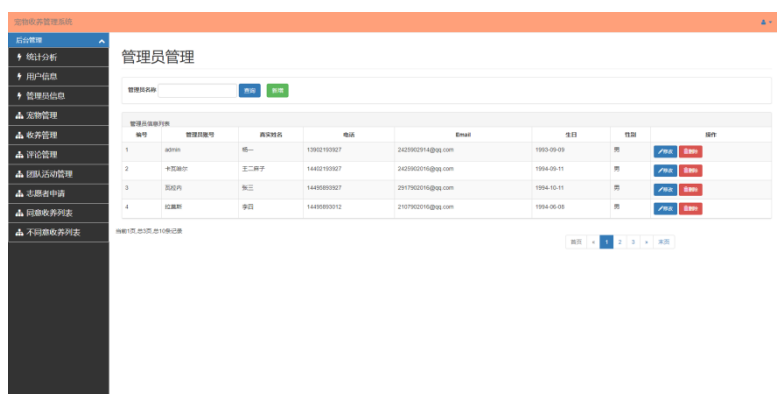


图 6.7 管理员管理界面

管理员一旦登录到后台管理，就可以通过后台进一步管理系统的用户信息，这同样也是宠物收养系统较为核心的模块，在这个模块实现了对系统内用户信息的管理。用户管理界面如图 6.8 所示。

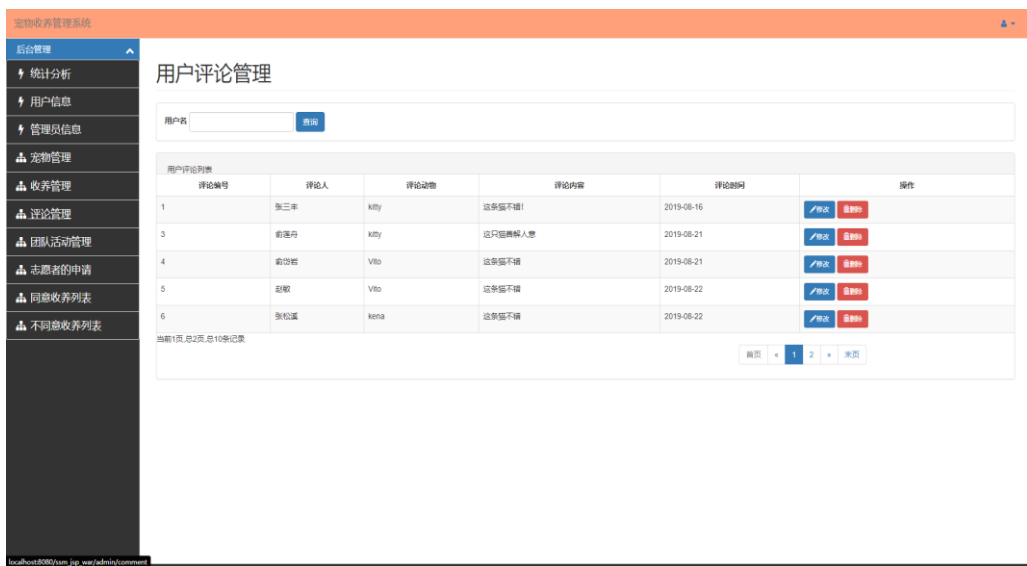


图 6.8 用户管理界面

管理员在登录访问后端的管理后，在后端对系统的宠物信息进行管理，这也是宠物收养系统比较重要的模块，宠物收养系统在这个模块实现了对系统内宠物信息的管理。宠物信息管理界面如图 6.9 所示。

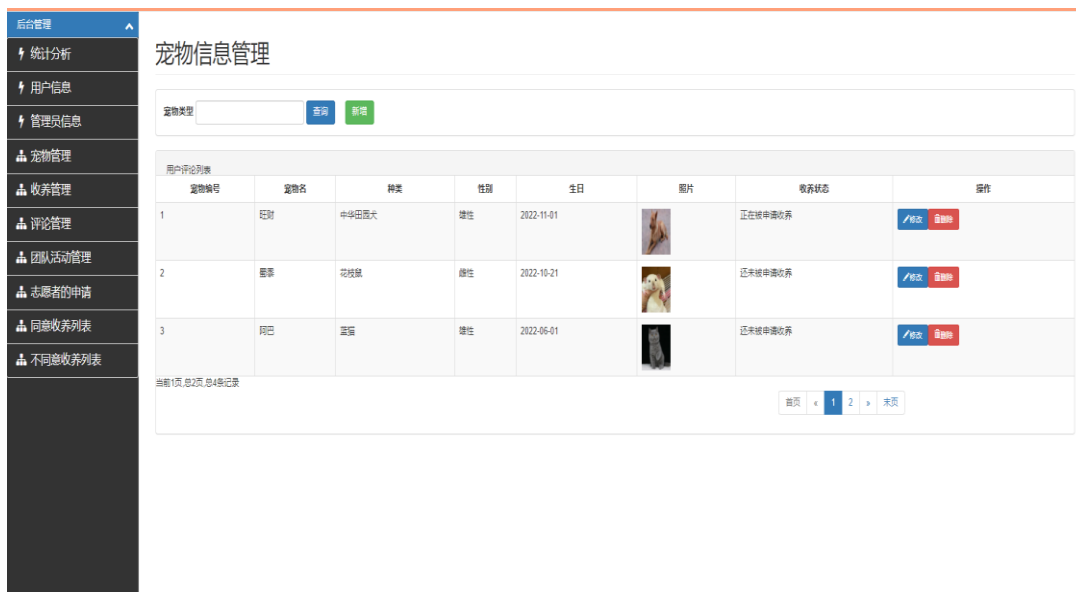


图 6.9 宠物信息管理界面

管理员可对用户的领养要求进行审核，这个模块是比较重要也是较为核心的模块之一，通过查看收养要求信息之后，如果同意则通过审核，否则不同意该申请。申请管理界面如图 6.10 所示。



图 6.10 申请管理界面

关键代码如下：

```
public Message agree(String userName, @RequestParam("pn") Integer
pageNum, Integer state) {
    Integer pageSize = 5;
    if (pageNum == 0) {
        pageNum = 1;
    }
    PageInfo<AdoptAnimal> pageInfo =
adoptAnimalService.allAdoptAnimal(userName, pageNum, pageSize, state);
    return Message.success().add("pageInfo", pageInfo);
}
```

管理员对用户的评论进行管理，通过查看评论内容，可以对不良的言论进行删除或修改，以实现平台评论更和谐的环境。评论管理界面如图 6.11 所示。

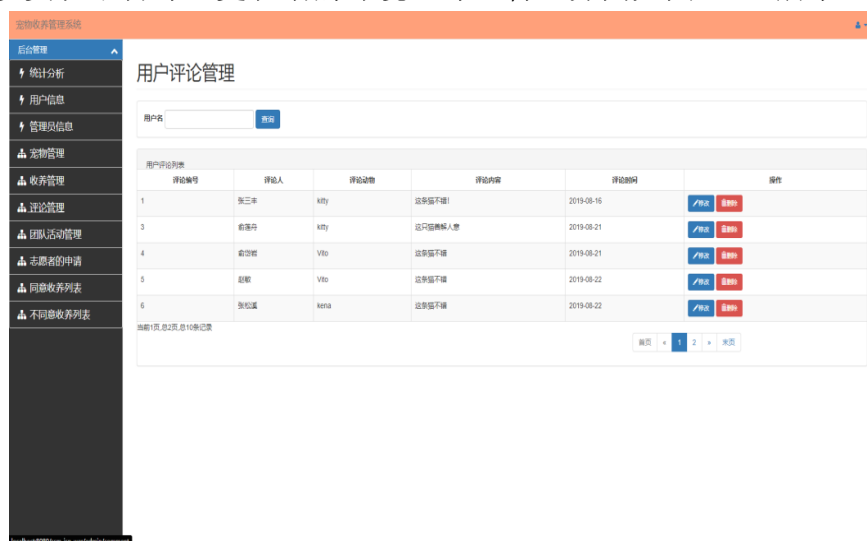


图 6.11 评论管理界面

关键代码如下：

```
public Message comments(@RequestParam(required = false) String
userName, Integer pn){
    Integer pageSize = 5;
    PageInfo<Comment> pageInfo = commentService.allComment(userName,
pn, pageSize);
    return Message.success().add("pageInfo", pageInfo);
}
```

管理员可对用户活动进行管理，查看用户的活动是否合理合规，如果活动的发布有不良信息则可对其进行删改。活动的管理界面如图 6.12 所示。

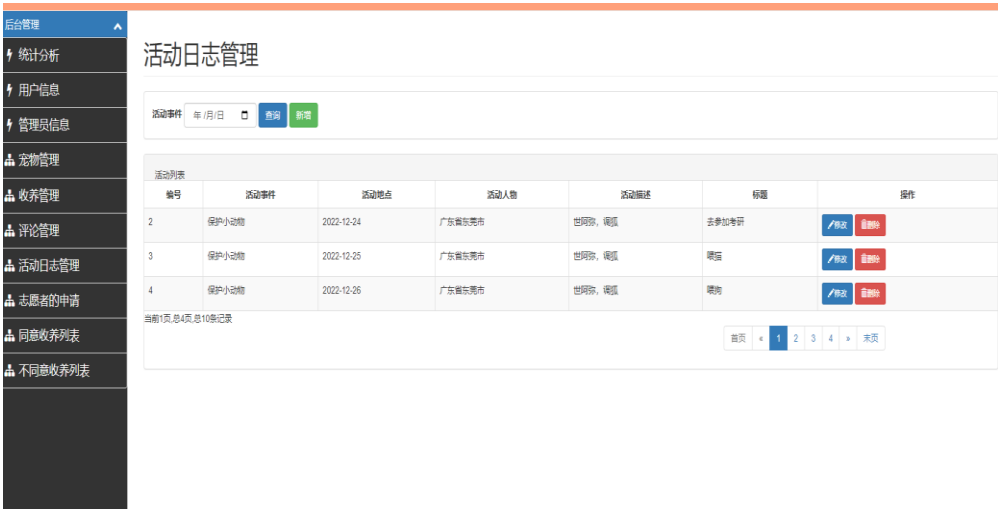


图 6.12 活动管理界面

关键代码如下：

```
public Message comments(@RequestParam(required = false) String
userName, Integer pn){
    Integer pageSize = 5;
    PageInfo<Comment> pageInfo =
commentService.allComment(userName, pn, pageSize);
    return Message.success().add("pageInfo", pageInfo);
}
```

管理员对应用信息记录的查看有助于管理员管理活动日志历史信息，也同时是管理员的管理需求，大大提高了系统管理的效率。如图 6.13、6.14 所示。

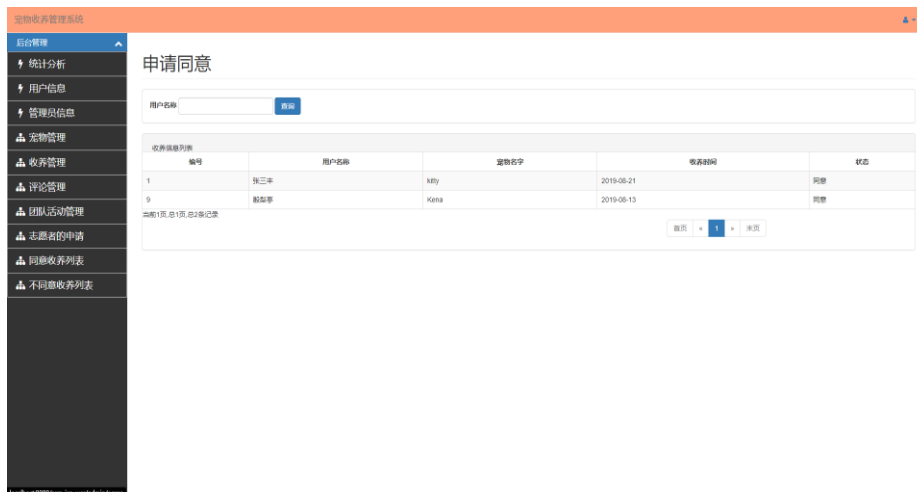


图 6.13 同意申请

关键代码如下：

```
public Message agree(Integer id){
    System.out.println(id);
    int update = adoptAnimalService.update(id, 2);
    if(update>0){
        return Message.success();
    }
    return Message.fail();
}
```



图 6.14 拒绝申请

关键代码如下：

```
public Message disagree(Integer id){
    int update = adoptAnimalService.update(id, 0);
    if(update>0){
        return Message.success();
    }
    return Message.fail();
}
```

此外，管理员还可以在登录进入后端系统后，通过查看具有统计分析功能扇形表

得知用户的性别比与宠物的性别比例来更好的应对市场情况从而提高效率。如图 6.15 所示。



图 6.15 统计分析功能扇形表

关键代码如下：

```
for (let i = 0; i < result.extend.pageInfo.length; i++) {
    console.log(result.extend.pageInfo[i])
    if(result.extend.pageInfo[i].sex=="男"){
        nan = nan+1

    }else{
        nv = nv+1
    }
    console.log(nan, nv)
}
setTimeout(function() {
    console.log(nan, nv, "nv, ci, ")
})
```

6.3 本章小结

本章对宠物收养系统的核心功能进行了简单的介绍，并且选取管理端和客户端的部分运行截图进行展示。系本统较为注重用户的视觉体验和使用体验，在前端页面的布局上使用较为简洁的风格，并且在页面配色方面使用了较为平淡的颜色作为系统的主色调。

7 系统功能测试

7.1 系统测试目的

系统测试是在开发完成后对管理端和客户端进行的较为全面的测试。其目的是检查所开发的系统功能是否符合要求，并找出系统中的漏洞和隐患^[14]，越是早发现软件中的 bug，投入使用后造成的损失就可以尽可能的减小。系统测试的主要任务是测试链接和测试目的，以及编写测试报告。系统测试的主要类型是单元测试、集成测试和系统测试。单元测试的范围相对较小，可以直接由开发人员来完成^[15]，集成测试的范围相对较大，目的是测试模块之间集成的功能，系统测试的目的在于系统的验收。对系统进行更为仔细的测试有助于提高系统的安全性和稳健性。

7.2 系统测试内容

测试系统功能的目的是为了尽可能地减少开发过程中出现的错误和缺陷。本文所设计的宠物收养系统将在用户和管理员模块进行测试。用户模块主要是管理员对用户信息模块的相关的管理操作。测试用例的合理性以及测试结果被用来编写相应的测试报告和评估所开发的系统。

7.3 测试用例

管理员在进入后台页面后进行添加用户的测试，统共分为两类，一类是输入合乎系统要求与输入不合乎系统要求的用户信息，在后台用户信息管理页面点击新增的按钮输入相关数据后，点击创建的按钮，以完成用户添加测试，其结果如表 7.1 所示。

表 7.1 用户添加测试表

| 操作序号 | 测试操作描述 | 预期输出 | 实际输出 |
|------|-----------------------------|--------|--------|
| 1 | 输入了合乎系统要求的用户相关信息，并点击了创建按钮。 | 创建成功 | 创建成功 |
| 2 | 输入了不合乎系统要求的用户相关信息，并点击了创建按钮。 | 提示错误 | 提示错误 |
| 3 | 输入了合乎系统要求的用户相关信息，并点击了关闭按钮。 | 无用户被创建 | 无用户被创建 |

管理员在进入后台页面后进行删除用户的测试，由系统后台页面的提示，在选择

要删除的用户后依次点击删除与确定或取消按钮，后完成该项测试。其删除用户结果如表 7.2 所示。

表 7.2 用户删除测试表

| 操作序号 | 测试操作描述 | 预期输出 | 实际输出 |
|------|---------------------------------|--------|--------|
| 1 | 选择要删除用户后，点击其旁边的删除按钮，弹出提示框后点击确定。 | 删除成功 | 删除成功 |
| 2 | 选择要删除用户后，点击其旁边的删除按钮，弹出提示框后点击取消。 | 用户未被删除 | 用户未被删除 |

用户信息进行修改的操作，管理员进入系统后台之后，在用户信息的管理模块选择要修改信息的用户，点击其右侧的修改的按钮，根据系统的提示来输入符合系统要求的修改信息或不符合系统要求的修改信息，最后点击保存修改或关闭，其修改用户信息结果如表 7.3 所示。

表 7.3 用户信息修改表

| 操作序号 | 测试操作描述 | 预期输出 | 实际输出 |
|------|---|-----------|-----------|
| 1 | 选择要修改信息的用户后，点击其旁边的修改按钮，弹出提示框后输入符合系统要求的修改后的信息。 | 用户信息更新成功 | 用户信息更新成功 |
| 2 | 选择要修改信息的用户后，点击其旁边的修改按钮，弹出提示框后输入不符合系统要求的修改后的信息。 | 提示错误 | 提示错误 |
| 3 | 选择要修改信息的用户后，点击其旁边的修改按钮，弹出提示框后输入符合系统要求的修改后的信息，并点击关闭。 | 用户的信息未被修改 | 用户的信息未被修改 |

对用户进行查询信息测试，在登录进入系统后台后，管理员在用户信息管理模块

在查询框内输入用户的名称对用户进行模糊查询，输入名称后点击查询完成测试，其用户信息查询测试结果如表 7.4 所示。

| 操作序号 | 测试操作描述 | 预期输出 | 实际输出 |
|------|---------------|------------------------|------------------------|
| 1 | 输入系统内已有的用户名称。 | 查询用户成功，用户信息页面展示出查找的结果。 | 查询用户成功，用户信息页面展示出查找的结果。 |
| 2 | 输入系统内没有的用户名称。 | 用户信息展示也显示没有用户。 | 用户信息展示也显示没有用户。 |
| 3 | 将输入信息设为空。 | 显示系统内所有的用户。 | 显示系统内所有的用户。 |

7.4 测试总结

按照测试用例对本系统进行了严格的测试工作，测试的过程中发现了登录注册功能存在的一些漏洞。根据测试中显示的一些漏洞，对相关的代码进行了修改，修改完毕后再对本测试用例的对应模块进行反复测试，测试结果已通过并符合预期，暂时无其他严重漏洞的存在。

8 总结

8.1 系统总结

对于本系统主要采用的 SSM 框架，它可以提高编码效率并减少工作量^[16]。每个计算机终端都需要一个相应的客户端，操作系统如果升级也需要提供相关的技术支持。本文利用 Java+SSM+MySQL 设计开发了一个宠物收养系统，以提高系统的管理水平。本论文的主要工作总结如下：

（1）尽可能的去获取宠物收养业务的情报，收集了一些普遍存在的信息，以及对业务流程进行了较为详细分析，并完成了关于系统的业务需求分析。

（2）根据当下状况总结出了宠物收养系统的大体方案。在前面提到的需求分析的基础之上，采用模块化设计思想设计了宠物收养系统的总体功能框架，并完成了数据库的设计。

（3）开发宠物收养系统。给出了用户登录、信息管理、评论和活动管理等主要模块的实施结果，成功地完成了宠物收养系统的开发。

（4）宠物收养系统已经部署并且还进行了系统的测试，测试结果合乎预料，能够满足实际的市场需求。

8.2 展望

虽然该宠物收养系统在实际的测试和应用过程中取得了良好的效果，但本文只是对基于 SSM 框架的宠物收养系统开发过程的基本相关研究，而且功能架构还能进一步完善和扩展，以及对系统和代码进行优化，从而可以减少代码的冗余度，以进一步的使代码的运行效率提升，进以提高系统的稳定性，减少数据输入的效率低下和错误率。进一步深化和分析宠物收养系统的业务流程。保留与需求相对应的程序的扩展性以及可移植性，而且可以再优化应用界面等。本系统的开发和研究将会持续伴随着“互联网+”时代的不停深化与宠物收养行业的需求的迭代。

参考文献

- [1]张译丹,贾峤.新时代宠物行业发展研究[J].现代商贸工业,2021,42(30):17-18.
DOI:10.19311/j.cnki.1672-3198.2021.30.008.
- [2]张贺雨.城市流浪宠物治理法律问题研究[D].淮北师范大学,2020.DOI:10.27699/d.cnki.ghbmt.2020.000140.
- [3]刘琴琴.我国流浪动物规范管理的法律研究[D].西北大学,2020.DOI:10.27405/d.cnki.gxbdu.2020.002158.
- [4]韩丽萍.中国宠物市场发展的的问题与对策研究[D].上海财经大学,2021.DOI:10.27296/d.cnki.gshcu.2021.000480.
- [5]张画仪,王楚桐,郑鑫楠等.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.
- [6]蔡佳文.“它经济”乱象待解[J].中国商界,2023(01):90-92.
- [7]赵泽铖,汤运凯,匡仙丹.基于SSM的前后端分离式藏文学习系统设计[J].信息与电脑(理论版),2022,34(17):105-109.
- [8]王江霖,黄彩霞,樊旭龙.基于SSM框架的网络资源共享平台的研究与实现[J].电脑知识与技术,2021,17(28):76-78.DOI:10.14004/j.cnki.ckt.2021.2999.
- [9]欧阳桂秀.基于Java和MySQL的数据库管理系统的设计与实现[J].信息记录材料,2022,23(09):240-242.DOI:10.16009/j.cnki.cn13-1295/tq.2022.09.007.
- [10]张文豪.Bootstrap模态对话框的开发与应用[J].福建电脑,2022,38(12):99-102.DOI:10.16707/j.cnki.fjpc.2022.12.020.
- [11]王艳梅,丁玲.JavaScript技术在Web网页中实践分析[J].自动化应用,2022(12):62-65.DOI:10.19769/j.zdhy.2022.12.015.
- [12]刘一农,刘子鸣.高校图书馆读者服务系统设计原则及模块实现[J].科技视界,2020,No.300(06):124-127.DOI:10.19694/j.cnki.issn2095-2457.2020.06.47.
- [13]王延武.浅谈数据库设计方法[J].山东工业技术,2016,No.224(18):139.DOI:10.16640/j.cnki.37-1222/t.2016.18.121.
- [14]Blanco Raquel,Trinidad Manuel,Suárez-Cabal María José,Calderón Alejandro,Ruiz Mercedes,Tuya Javier. Can gamification help in software testing education? Findings from an empirical study[J]. The Journal of

Systems & Software, 2023, 200.

[15]Kyungmin Sim, Jeong Yi, Haehyun Cho. SMINER: Detecting Unrestricted and Misimplemented Behaviors of Software Systems Based on Unit Test Cases[J].

Computers, Materials & Continua, 2023, 75(2).

[16]Design and Implementation of ERP System Based on SSM Framework[J]. , 2020, 440(5).

致谢

回顾整个大学时光，我的学生生涯就要结束了，离开这个校园，即将踏上新的人生旅程。此刻，百感交集。这是我人生中最后的学生时光——我生命中美丽的风景，我会珍惜这段记忆，好好记住它。衷心感谢陪伴我走过流金岁月的老师、家人和朋友，与你们的相遇也是我人生记忆这片美丽风景中重要的点缀。

重新站在广科的树荫之下，在一片片校园的嘈杂声之中，回忆起以往的点点滴滴，才意识到我的大学时光即将结束。一时感慨，只言片语难以言喻。三年的在校生活让我熟练掌握更多专业知识和相关技术，并且有幸在许多专业技能出众、教导有方的老师下学习，为我以后的工作学习打下了坚实的基础，虽然学习的过程充满了艰辛与痛苦，但是也在无形中磨砺了我的意志提高了我的计算机知识素养及实践能力。遥想 2019 年的秋季，到 2023 年的夏季，回忆涌上心头，历历在目，无法想象四年的求学生涯竟如此短暂，一路的欢笑和沮丧，让我受益良多。十分感谢我的毕业设计导师孟雨老师，无论是从论文的选题还是毕业论文最后的完成，都离不开孟雨老师平日里对我无微不至的教导，使得我对这个宠物收养系统有了更为深刻的认识。还有在大学里的各科老师以及辅导员，十分感谢他们对我悉心教导，让我在收获书本上的知识之余，还让我收获了处事做人的经验。承蒙各位导师们对我这个朽木的不离不弃，能够得到各位导师们的指导是我的荣幸，感谢母校提供的学习平台和优秀的学习氛围，感谢有幸能在 2019 级软件工程本科 9 班这个让我难以割舍的集体，大家同舟共济、互帮互助，在无数个日月中解的相互陪伴。感谢舍友们对我的帮忙和照顾，你们在我的青春中留下了浓厚的彩墨，构成了我的回忆中一幅美好大学生活的画卷。无论以后的路走到哪里，会在哪里驻足，广东科技学院永远都是我忘不掉的地方，一路风雨相伴，再次感谢大学相遇过的各位。

最后，感谢在百忙之中抽出宝贵时间对本论文进行评议和审阅的各位专家与教授。