**基于WEB的宠物领养平台的设计与实现**

|  |  |  |  |
| --- | --- | --- | --- |
| 学 院：  专 业：  姓 名：  指导老师： | 计算机学院 | | |
| 计算机科学与技术 | | |
| 唐瑾 | 学 号：  职 称： | 150201101892 |
| 王庆娟 | 讲师 |

中国·珠海

二○一九年四月

**诚信承诺书**

**本人郑重承诺：**本人承诺呈交的毕业设计《基于WEB的宠物领养平台的设计与实现》是在指导教师的指导下，独立开展研究取得的成果，文中引用他人的观点和材料，均在文后按顺序列出其参考文献，设计使用的数据真实可靠。

本人签名：

日期： 年 月 日

**基于WEB的宠物领养平台的设计与实现**

**摘 要**

国内经济的飞速发展让人们的生活水平普遍提高，随之提高的还有人们对于心理精神上的需求,喂养宠物逐渐成为他们缓解心理压力和体验精神享受的一种方式，拥有宠物的人多了，因为各种原因流离失所的宠物数量也在逐年攀升。为了让那些无家可归的小宠物们能够找到对他们真正负责的主人，避免宠物拐卖诈骗的行为，本论文设计并实现了一种具有第三方“爱宠”机构参与监督和审核的宠物领养系统。

本设计使用Atom编辑器，运用Node.js，Mysql,Vue等技术为爱宠人士搭建了“爱宠”宠物领养网站。“爱宠”宠物领养网站为爱宠人士提供了一个可靠的宠物领养平台，让他们能够及时浏览领养信息，选择申请希望领养的宠物，同时为他们提供一个更加便利正规的网络领养环境。

经过对系统的测试，测试结果表明用户可以在网站上成功进行领养和申请等操作，此外，管理员也能对领养信息进行有效的审核还有修改，该宠物领养平台的成功运行，可以有效的减少流浪宠物的数量。

**关键词：**宠物领养；网站；Mysql；Vue

**Design and Implementation of a Pet Adoption Platform**

**Based on WEB**

**Abstract**

The rapid development of the domestic economy has generally improved people's living standards, and there has been a rise in people's psychological and spiritual needs. Feeding pets has gradually become a way for them to relieve psychological stress and experience spiritual enjoyment. The number of pets displaced for various reasons is also rising year by year. In order to allow those homeless pets to find the owners who are really responsible for them and avoid the behavior of pet trafficking fraud, this paper designs and implements a pet adoption with a third-party “love pet” agency to participate in supervision and review system.

This design uses the Atom editor, using Node.js, Mysql, Vue and other technologies to build a "love pet" pet adoption website for the pet lovers. The “Pet Pet” pet adoption website provides a reliable pet adoption platform for pet lovers, allowing them to browse adoption information in a timely manner, choose to apply for pets they wish to adopt, and provide them with a more convenient and formal online adoption environment.

After testing the system, the test results show that the user can successfully carry out the adoption and application operations on the website. In addition, the administrator can also effectively review and adopt the adoption information, and the successful operation of the pet adoption platform can be effective. Reduce the number of stray pets.

**Keywords:**  Pet adoption; website; Mysql;vue

**目 录**

[1. 引言 1](#_Toc6354681)

[1.1. 项目开发背景和意义 1](#_Toc6354682)

[1.2. 项目技术 1](#_Toc6354683)

[1.2.1. Mysql介绍 1](#_Toc6354684)

[1.2.2. Node.js介绍 1](#_Toc6354685)

[1.2.3. Vue介绍 2](#_Toc6354686)

[1.2.4. Html,css介绍 2](#_Toc6354687)

[1.2.5. Javascript介绍 2](#_Toc6354688)

[1.2.6. Express框架介绍 2](#_Toc6354689)

[1.3. 可行性分析 3](#_Toc6354690)

[1.3.1. 社会可行性分析 3](#_Toc6354691)

[1.3.2. 技术可行性分析 3](#_Toc6354692)

[1.4. 论文结构内容 3](#_Toc6354693)

[2. 系统需求分析 4](#_Toc6354694)

[2.1. 功能需求分析 4](#_Toc6354695)

[2.1.1. 用户端 4](#_Toc6354696)

[2.1.2. 管理端 4](#_Toc6354697)

[2.2. 用例图 5](#_Toc6354698)

[2.3. 详细用例说明 6](#_Toc6354699)

[2.3.1. 用例UC1：注册账号 6](#_Toc6354700)

[2.3.2. 用例UC2：登录账号 7](#_Toc6354701)

[2.3.3. 用例UC3：修改密码 8](#_Toc6354702)

[2.3.4. 用例UC4：申请发布领养信息 8](#_Toc6354703)

[2.3.5. 用例UC5：选择领养人 9](#_Toc6354704)

[2.3.6. 用例UC6：申请领养 10](#_Toc6354705)

[2.3.7. 用例UC7：查看领养信息 10](#_Toc6354706)

[2.3.8. 用例UC8：查看个人主页信息 11](#_Toc6354707)

[2.3.9. 用例UC9：修改领养信息 11](#_Toc6354708)

[2.3.10. 用例UC10：审核领养信息 12](#_Toc6354709)

[2.3.11. 用例UC11：删除领养信息 13](#_Toc6354710)

[2.3.12. 用例UC12：查看已完成的领养信息 13](#_Toc6354711)

[2.4. 领域模型 14](#_Toc6354712)

[3. 概要设计 15](#_Toc6354713)

[3.1. 逻辑视图 15](#_Toc6354714)

[3.2. 开发视图 15](#_Toc6354715)

[3.3. 物理视图 16](#_Toc6354716)

[4. 数据库设计 18](#_Toc6354717)

[4.1. 数据库ER图 18](#_Toc6354718)

[4.2. 数据库表设计 20](#_Toc6354719)

[4.3. 数据库逻辑视图 22](#_Toc6354720)

[5. 网页设计与代码实现 24](#_Toc6354721)

[5.1. 用户端 24](#_Toc6354722)

[5.1.1. 登录账号 24](#_Toc6354723)

[5.1.2. 注册账号 25](#_Toc6354724)

[5.1.3. 修改密码 26](#_Toc6354725)

[5.1.4. 申请发布领养信息 27](#_Toc6354726)

[5.1.5. 查找领养信息 28](#_Toc6354727)

[5.1.6. 选择领养人 29](#_Toc6354728)

[5.1.7. 隐藏已经成功领养的领养信息 30](#_Toc6354729)

[5.2. 管理员端 31](#_Toc6354730)

[5.2.1. 登录账号 31](#_Toc6354731)

[5.2.2. 显示未审核的领养信息 31](#_Toc6354732)

[5.2.3. 审核领养信息 32](#_Toc6354733)

[5.2.4. 修改领养信息 33](#_Toc6354734)

[5.2.5. 删除领养信息 34](#_Toc6354735)

[6. 安装与测试 36](#_Toc6354736)

[6.1. 运行环境简介 36](#_Toc6354737)

[6.2. 测试内容 36](#_Toc6354738)

[6.3. 测试操作步骤 36](#_Toc6354739)

[6.4. 测试用例 36](#_Toc6354740)

[6.4.1. 登录账号 36](#_Toc6354741)

[6.4.2. 注册账号 37](#_Toc6354742)

[6.4.3. 修改密码 37](#_Toc6354743)

[6.4.4. 查找领养信息 37](#_Toc6354744)

[6.4.5. 申请发布领养信息 37](#_Toc6354745)

[6.4.6. 申请领养 38](#_Toc6354746)

[6.4.7. 审核领养信息 38](#_Toc6354747)

[6.4.8. 修改领养信息 39](#_Toc6354748)

[6.4.9. 删除领养信息 39](#_Toc6354749)

[7. 结论 40](#_Toc6354750)

1. **引言**
   1. **项目开发背景和意义**

在信息科技不断发展的今天，随着人们生活水平的提高，喂养宠物成了人们释放工作压力跟享受生活的一种方式。宠物店快速发展起来，养宠物的人越来越多的同时，因为各种原因流离失所的宠物数量也在逐年攀升，为了私人利益而进行不正当的宠物交易，宠物拐卖还有宠物诈骗的行为屡见不鲜。而通过宠物领养平台，爱宠人士能够更加方便快捷的浏览领养信息，只有通过网上提交申请，审核成功之后才能领养宠物，提高了宠物领养的安全性，网站领养成为了越来越多人的选择。

但是，现如今网上的宠物领养信息真假难辨，如何能够让无家可归的宠物们找到真正对自己负责的主人？如何避免网上宠物诈骗拐卖行为的发生？如何让宠物领养真正公益化？以上问题需要一个专门验证审核领养信息和领养人信息的宠物领养平台来解决。开发本宠物领养平台就是为了帮助四处流浪，居无定所的宠物们找到能对他们一生负责的主人，避免网上宠物诈骗拐卖行为的发生，通过宠物领养网站，可以更加方便迅速的查找宠物信息，及时有效的过滤虚假的领养信息，宠物们能够通过更加安全正规的途径被爱心人士收养，领养人可以通过注册登录网站，浏览查找相关的领养信息，选择自己心仪的宠物申请领养，通过审核之后，便可以与领养信息发布者联系领养宠物，流程操作简单，具有良好的用户体验。

* 1. **项目技术**

本系统采用b/s架构，系统设计开发分为前端和后端开发，用mysql建立数据库，运用node.js技术的express框架搭建后台，前端方面用vue搭建前台，用html,css,javascript进行开发，前后端用json的数据格式进行传输数据，ajax异步加载数据。

* + 1. **Mysql介绍**

Mysql是一个多线程，多用户的关系型数据库管理系统。它可以运行在多种不同的操作系统平台，具有良好的兼容性。同时，mysql也支持常见的sql语句规范，安装mysql不需要占用很大的内存空间，安装之后具有良好的运行效率，同时也可以处理拥有上千万条记录的大型数据。Mysql相对于其他的大型数据库来说，具有调试，管理，优化简单的特点。

* + 1. **Node.js介绍**

Node.js是一个javascript的运行环境，依赖于chrome V8引擎进行代码解释，它轻量，可伸缩，适用于实时数据交互应用。Node.js不会为每一个客户连接创建一个新的线程，而仅仅使用一个线程，当由用户连接的时候就触发一个内部事件，节省内存。另外，node.js中采用了非阻塞型I/O机制，因此在执行了访问数据库的代码之后，将立即转而执行之后的代码，将数据库返回结果的处理代码放在回调函数中，从而提高了程序的执行效率。

* + 1. **Vue介绍**

Vue.js是一个构建数据驱动的web界面的渐进式框架。它是一个轻巧，高性能，可组件化的MVVM库，而MVVM框架就是把MVC(Model(模型)+View(视图)+controller(控制器))里的controller和MVP(Model+View+Presenter)里面的Presenter改成了ViewModel，view是Html文本的js模板，它的变化会自动更新到业务逻辑层ViewModel，ViewModel的变化也会自动同步到View上显示，Model数据层是对数据的处理。View.js的目标是通过尽可能简单的API的实现响应的数据绑定和组合的视图组件，它的核心是一个响应的数据绑定系统。

* + 1. **Html,css介绍**

HTML(HyperText Markup Language)是一个超文本标记语言，它定义了网页内容的含义还有结构。HTML使用“标记”来注明文本，图片还有其他内容，以便于在web浏览器当中显示。Css(Cascading style sheets)，是一种样式表语言，用来描述HTML或者XML文档的呈现。Css主要是让页面更加好看，更加生动吸引用户，让用户使用网站时具有良好的用户体验。

* + 1. **Javascript介绍**

JavaScript是一种基于对象和事件驱动并且具有安全性能的脚本语言，使用JavaScript结合html还有css可以实现丰富的页面交互效果。有的页面效果html或css能实现的，javascript也能实现，有时候使用javascript还比html还有css更加方便，简洁。

* + 1. **Express框架介绍**

Express是一个简洁灵活的node.js的web应用框架，提供了一系列强大特性帮助开发者创建各种web应用还有丰富的Http工具，使用express可以快速的搭建一个完整功能的网站，可用于单页面，多页面和混合web应用。

* 1. **可行性分析**
     1. **社会可行性分析**

近年来，生活水平的提高促使越来越多的人开始养宠物，同时需要关怀的流浪宠物的数量也在逐年攀升，网上的宠物诈骗行为屡禁不止。此外，国内外虐待宠物的新闻频频被曝，有些宠物的原主人和领养人之间经常因为领养宠物的问题纠纷不断，让爱宠人士们感到分外寒心。当下非常需要一个宠物领养平台来缓解流浪宠物无家可归的问题，另外，宠物领养平台还可以给原主人跟领养人之间做领养凭证用以解决不必要的纠纷同时也避免了宠物诈骗行为的发生。

* + 1. **技术可行性分析**

技术方面，本设计将会用到html，css，javascript，ajax，node.js，vue.js，mysql等新兴技术，利用这些技术来完成整个网站可以使网页界面交互性更加友好，使得整体代码更加简洁，确保网站运行的安全稳定。

* 1. **论文结构内容**

首先，论文在第二章开始会对宠物领养网站进行系统需求分析，它分别从用户端还有管理员端对领养网站来进行功能需求分析，接下来使用用例图来更加简洁的展现系统的功能需求，然后再进行详细的用例说明。在对每个用例进行详细的说明之后，在第三章分别从逻辑视图，开发视图还有物理视图来描述对系统的概要设计。之后的第四章用数据库ER图来描述实体间的关系，列出所有的数据库的表，画出数据库的逻辑视图，详细的写出系统的数据库的设计。在第五章就是分别从用户端还有管理员端详细写写用例实现。第六章会写系统的安装还有测试，这一章会从运行环境，系统运行操作步骤还有测试这三个部分展开，最后就是论文的总结部分。

1. **系统需求分析**
   1. **功能需求分析**

宠物领养平台根据参与者角色可以分成两个子系统：用户端和管理端，分别面向用户跟管理员。

* + 1. **用户端**

用户端面向的用户是想要领养宠物的人，这个系统的主要功能是申请领养，具体功能需求分析如下：

1. 注册登录

只有通过注册登录之后，用户才能使用网站上的所有功能，包括申请领养。

1. 修改密码

修改密码的功能是为那些忘记密码的用户提供的，通过该功能能让用户重新设置密码。

1. 申请发布领养信息

用户登录之后，选择申请发布领养信息，管理员审核通过后，领养信息就可以在网站上显示出来了，发布人可以获取申请领养的用户信息。

1. 选择领养人

发布人可以在申请领养人当中选择一个最适合领养宠物的领养人，其他领养人的申请则不被通过。

1. 申请领养

用户选择了自己想要领养的宠物之后，就可以提交领养申请，通过发布人审核通过之后就可以进行领养了。

1. 查找领养信息

用户可以根据自己的需要查找领养信息。

1. 查看个人主页信息

用户可以通过查看自己的个人主页知道自己的领养申请是否通过，也可以查看到自己发布的领养信息是否通过了管理员审核。

* + 1. **管理端**

管理端面向的是网站管理员，主要功能就是让管理员审核申请发布的领养信息。具体功能分析如下：

1. 审核领养信息

管理员审核申请发布的领养信息，确认无误后即可允许发布申请，只有通过审核，网页上才能显示出通过审核的领养信息。

1. 删除领养信息

管理员可以删除已经成功领养或者其他发布者的领养信息。

1. 修改领养信息

对于信息有错误的领养信息，管理员有权利修改领养信息，并将新修改的信息替换原来出错的信息。

1. 查看已完成领养信息

管理员登录之后也可以查看已经成功领养的信息。

* 1. **用例图**

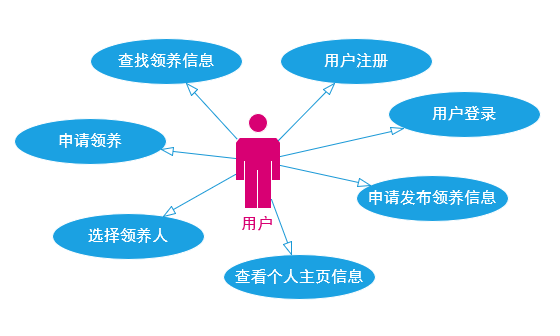


图2.1用户用例图

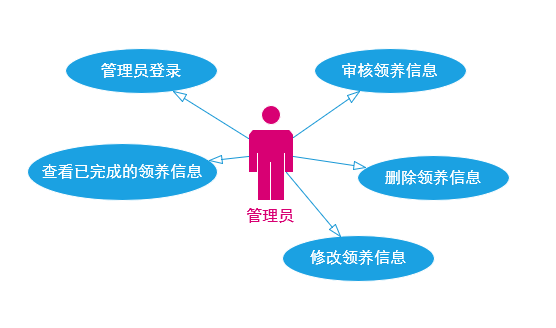


图2.2管理员用例图

* 1. **详细用例说明** 
     1. **用例UC1：注册账号**

**范围：**整个宠物领养网站

**主要参与者：**用户

**涉众及其关注点：**

未注册的用户希望注册的步骤方便快捷，而且个人信息不会泄露。

**前置条件：**

1. 用户还没有在该网站注册账号
2. 用户从未在该网站上登录自己的账号

**成功保证：**

用户注册之后登录账号成功，跳转到相应的主界面。

**主成功场景：**

1. 用户点击宠物领养网站进入注册页面。
2. 用户填写账号，密码，邮箱，联系方式，点击注册按钮提交。
3. 系统校验信息，检验成功之后跳转到登录页面。

**扩展：**

2a.注册信息

1. 当用户漏填其中任意一项注册信息的时候，网站会提示“请填写完整的用户信息”。
2. 当联系方式里填写的手机号码不足11位数或多于11位时，网页提示“请填写正确的手机号码”。

3a.系统校验信息

1. 当系统校验信息不通过时，会提示相应的错误信息，以便用户重新填写。
2. 当系统检测到注册账号跟数据库内有重复，则提示“该账号已注册，请重新输入”，之后再次返回到注册界面，用户重新注册。
3. 用户填写邮箱跟数据库有重复，则提示“该邮箱已被注册，请重新输入”
4. 之后再次填写注册信息才能继续注册。
5. 若用户填写联系方式跟数据库内数据重复，则网页提示“该联系方式已经被注册，请重新输入”，之后再次返回注册页面重新填写信息进行注册。
6. 校验通过之后，用户数据插入数据库，网页跳转至用户登录界面进行重新登录。

**发生频率：**高

* + 1. **用例UC2：登录账号**

**范围：**整个宠物领养网站

**主要参与者：**用户，管理员

**涉众及其关注点：**

用户，管理员希望通过填写相应的账号跟密码之后成功登录。

**前置条件：**

1. 用户已经在网站上注册成功
2. 用户，管理员还没有登录网站

**成功保证：**用户，管理员成功登录，页面跳转到相应主界面。

**主成功场景：**

1. 用户，管理员进入到登录界面之后，填写账号跟密码。
2. 点击登录按钮，系统对用户，管理员输入的账号密码进行校验。
3. 校验成功，进入网站主界面。

**扩展：**

2a.系统验证登录信息

1. 如果用户或管理员填的账号或密码信息为空，则提示用户“请输入账号密码”。
2. 如果验证账号密码不对应数据库的数据，则提示“账号或密码输入错误，请重新输入”，并且返回登录界面进行重新登录操作。

**发生频率：**高

* + 1. **用例UC3：修改密码**

**范围：**整个宠物领养网站

**主要参与者：**用户

**涉众及其关注点：**

用户能够在想要修改密码的情况下，能够以方便快捷的方式修改密码。

**前置条件：**

用户已经在网页上注册并登陆。

**成功保证：**

用户可以把自己原来的密码改成自己想要记住的新密码。

**主成功场景：**

1. 用户登录之后，点击进入个人主页。
2. 在个人主页点击修改密码按钮。
3. 在填写了原密码还有重复填写两次新密码之后，点击确认修改按钮。
4. 系统检验原密码是否正确，两次新密码是否一致。
5. 检验成功，修改密码成功，返回个人主页。

**扩展：**

2a.填写原密码跟新密码

1. 用户原密码与数据库里面的密码不一致，提示“原密码错误，请重新输入”，并且重新返回修改密码页面重新填写原密码。
2. 两次新密码不一致，提示“两次密码不一致，请重新输入”，并且重新返回修改密码页面重新填写两次新密码。

**发生频率：**低

* + 1. **用例UC4：申请发布领养信息**

**范围：**整个宠物领养网站

**主要参与者：**用户

**涉众及其关注点：**用户希望能快速的发布领养信息。

**前置条件：**用户在网站上已经注册，并且成功登录网站。

**成功保证：**

用户在编写好领养信息，点击申请按钮之后，返回个人主页，发送的申请会等待管理员审核。

**主成功场景：**

1. 用户在主界面点击导航栏“个人主页”下拉菜单的“我的申请”，跳转到申请页面之后点击“申请发布领养信息”按钮，网页跳转到编写申请界面。
2. 用户填写宠物名称，上传宠物照片，编写宠物的身体健康状况，还有联系方式，点击提交按钮。
3. 后台管理员接收申请信息，审核通过，发布领养信息到主页。

**扩展：**

3a.审核申请发布的领养信息

1. 用户提交申请之后，申请在被管理员审核之前是“待审核”状态。
2. 如果管理员审核不成功，则该申请的审核状态会变成“未通过”状态。
3. 如果管理员审核成功，则申请的审核状态就变成了“通过”状态。

**发生频率：**中

* + 1. **用例UC5：选择领养人**

**范围：**整个宠物领养网站

**主要参与者：**用户

**涉众及其关注点：**用户希望用方便方式选择申请领养宠物的人。

**前置条件：**

1. 用户已经在网站上注册登录成功。
2. 提交了申请发布领养信息，并且申请已经通过管理员审核。
3. 并且有一名或者多名用户申请领养宠物。

**成功保证：**

领养发起用户成功选择其中一名用户作为领养人，并且能够顺利与领养人取得联系进行领养宠物。

**主成功场景：**

1. 用户登录成功之后，在导航栏上点击“我的申请”，选中下拉菜单当中的“申请发布领养帖”，点击之后进入申请页面。
2. 在申请页面选中已经通过的申请，显示出来的申请内容下方有申请领养的用户列表，选中一个最合适的领养人，点击“通过”，其他用户则显示成“未通过”状态。

**扩展：**

2a.选择领养人

1. 选择了其中一个领养人之后，其他领养人的申请领养都会变成不通过的状态，系统将会发送通知给全部领养人，告知他们的申请结果是否通过。

**发生频率：**中

* + 1. **用例UC6：申请领养**

**范围：**整个宠物领养网站

**主要参与者：**用户

**涉众及其关注点：**

用户希望能够通过简单的申请流程领养到自己心仪的宠物。

**前置条件：**

用户已经在网页上注册并登陆成功。

**成功保证：**

用户选择好自己想要领养的宠物，点击“申请领养”跳转至相应界面，编辑好领养申请，并把申请发送给管理员，页面跳转到“我的申请”页面查看领养状态。

**主成功场景：**

1. 用户登录之后，点击自己想要领养宠物的信息进入宠物介绍界面。
2. 点击宠物信息页面里面的“申请领养”，跳转至申请领养信息填写界面。
3. 编辑申请领养的信息，包括家庭环境是否合适养宠物，家里是否还养了另外的宠物，经济条件是否能够承担养宠物的费用，工作是否繁忙，能不能照顾到领养的宠物等信息。
4. 编辑完成，点击“提交申请”按钮，领养发起人将会收到“领养申请”，发起人审核通过申请，申请人收到申请领养通过通知。

**扩展：**

4a.申请不通过

系统发送申请不通过通知给申请人。

**发生频率：**中

* + 1. **用例UC7：查看领养信息**

**范围：**整个宠物领养网站

**主要参与者：**用户

**涉众及其关注点：**

用户能够通过搜索快速的查找到关键的宠物领养信息。

**前置条件：**

用户已经在网页上注册并登陆。

**成功保证：**

用户在搜索框内输入关键词搜索，在页面成功显示出用户想要的搜索结果，并且搜索结果按时间排序。

**主成功场景：**

1. 用户登录之后，在主界面的查找输入框内填写想要查找的关键词，点击搜索。
2. 网站列出含有相应关键词的搜索结果。

**扩展：**无

**发生频率：**高

* + 1. **用例UC8：查看个人主页信息**

**范围：**整个宠物领养网站

**主要参与者：**用户

**涉众及其关注点：**

用户能够方便快捷的看到自己的主页信息，包括领养信息。

**前置条件：**

用户已经在网页上注册并登陆。

**成功保证：**

用户可以看到自己的个人信息。

**主成功场景：**

1. 用户登录之后，点击主页面的“个人主页”下拉菜单的“我的主页”按钮，进入个人主页查看个人信息。

**扩展：**

1a.用户查看个人信息

1. 用户只能查看自己的个人信息，用户之间不能互相看对方的个人信息，也就是进入不到对方的个人信息界面。

**发生频率：**中

* + 1. **用例UC9：修改领养信息**

**范围：**整个宠物领养网站

**主要参与者：**管理员

**涉众及其关注点：**

管理员能够方便快捷的修改宠物领养信息的内容。

**前置条件：**

管理员已经在网站上登录。

**成功保证：**

用户可以成功修改领养信息的内容，让领养信息更加简洁明了。

**主成功场景：**

1. 管理员登录之后，在主界面点击想要修改的领养信息。
2. 进入领养信息页面之后，点击修改领养信息按钮。
3. 修改需要更正的内容，点击保存，修改成功。

**扩展：**无

**发生频率：**低

* + 1. **用例UC10：审核领养信息**

**范围：**整个宠物领养网站

**主要参与者：**管理员

**涉众及其关注点：**

管理员能够在登录之后，能够方便快捷的审核领养信息。

**前置条件：**

管理员已经登陆网站。

**成功保证：**

管理员成功审核申请，点击通过或者不通过都跳转到主界面。

**主成功场景：**

1. 管理员登录之后，点击“审核领养信息”按钮，跳转至待审核的领养信息列表。
2. 审核申请信息是否为虚假诈骗信息。
3. 审核完成，点击“通过审核按钮”，领养信息会显示在主界面的领养信息列表上。
4. 系统提示用户申请已经通过，请留意申请领养宠物的爱宠人士。

**扩展：**

2a.审核信息失败

1. 点击“审核不通过”按钮，系统提示用户审核不通过，请再次检查信息真实性之后再次提交。

3a.显示领养信息

1. 在主界面显示的通过审核的领养信息则会以通过审核的时间为准来排列先后顺序。

**发生频率：**高

* + 1. **用例UC11：删除领养信息**

**范围：**整个宠物领养网站

**主要参与者：**管理员

**涉众及其关注点：**

管理员能够成功删除已经成功领养宠物的领养信息或者其他不必要或者不正规的领养信息。

**前置条件：**

管理员已成功登录网页。

**成功保证：**

管理员可以成功删除领养信息。

**主成功场景：**

1. 管理员登录之后，点击“选择删除领养信息”按钮，跳转到相应页面。
2. 在领养信息列表中，选中想要删除的领养信息。
3. 点击删除按钮，领养信息成功删除。

**扩展：**无

**发生频率：**中

* + 1. **用例UC12：查看已完成的领养信息**

**范围：**整个宠物领养网站

**主要参与者：**管理员

**涉众及其关注点：**

管理员希望查看已经完成的领养信息。

**前置条件：**

管理员已经成功登录网页。

**成功保证：**

管理员可以成功查看已完成的领养信息。

**主成功场景：**

1. 管理员登录网站之后，点击“已完成领养”按钮，跳转到已完成领养信息列表的页面。
2. 在管理员登陆之后的主界面，在搜索框输入“已完成领养”，网页显示已完成领养信息的列表。
3. 管理员点开想要查找的领养信息查看详情。

**扩展：**无

**发生频率：**低

* 1. **领域模型**

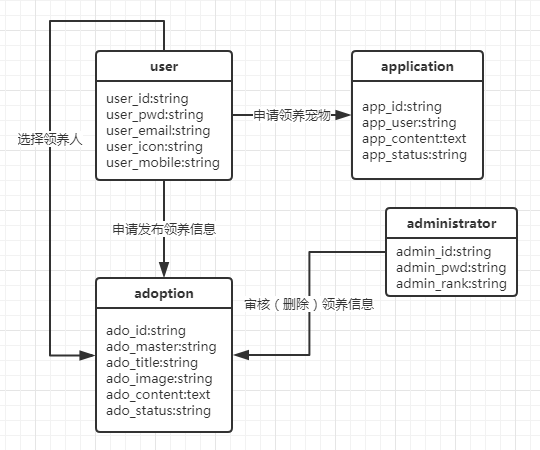


图2.1领域模型图

整个系统的用户分为两种，用户User还有管理员Administrator。其中，用户user里面有5个属性，分别是用户账号user\_id，密码user\_pwd，邮箱user\_email，头像user\_icon，联系方式user\_mobile。管理员Administrator有3个属性：管理员账号admin\_id，密码admin\_pwd，管理员等级admin\_rank。用户User可以通过填写申请发布领养帖Adoption来让爱心人士发现流浪宠物并且领养它，申请发布的领养帖有6个属性：申请号码ado\_id，申请人ado\_master，申请主题ado\_title，宠物图片ado\_image，申请内容ado\_content，申请状态ado\_status，管理员Administrator有权利去审核用户填写的申请Adoption并且决定申请是否可以通过审核。用户可以通过填写申请Application来申请领养宠物，申请Application有4个属性，申请号码app\_id，申请用户app\_user，申请内容app\_content，申请状态app\_status，领养帖的发起人可以选择其中一个申请人领养宠物。

1. **概要设计**
   1. **逻辑视图**

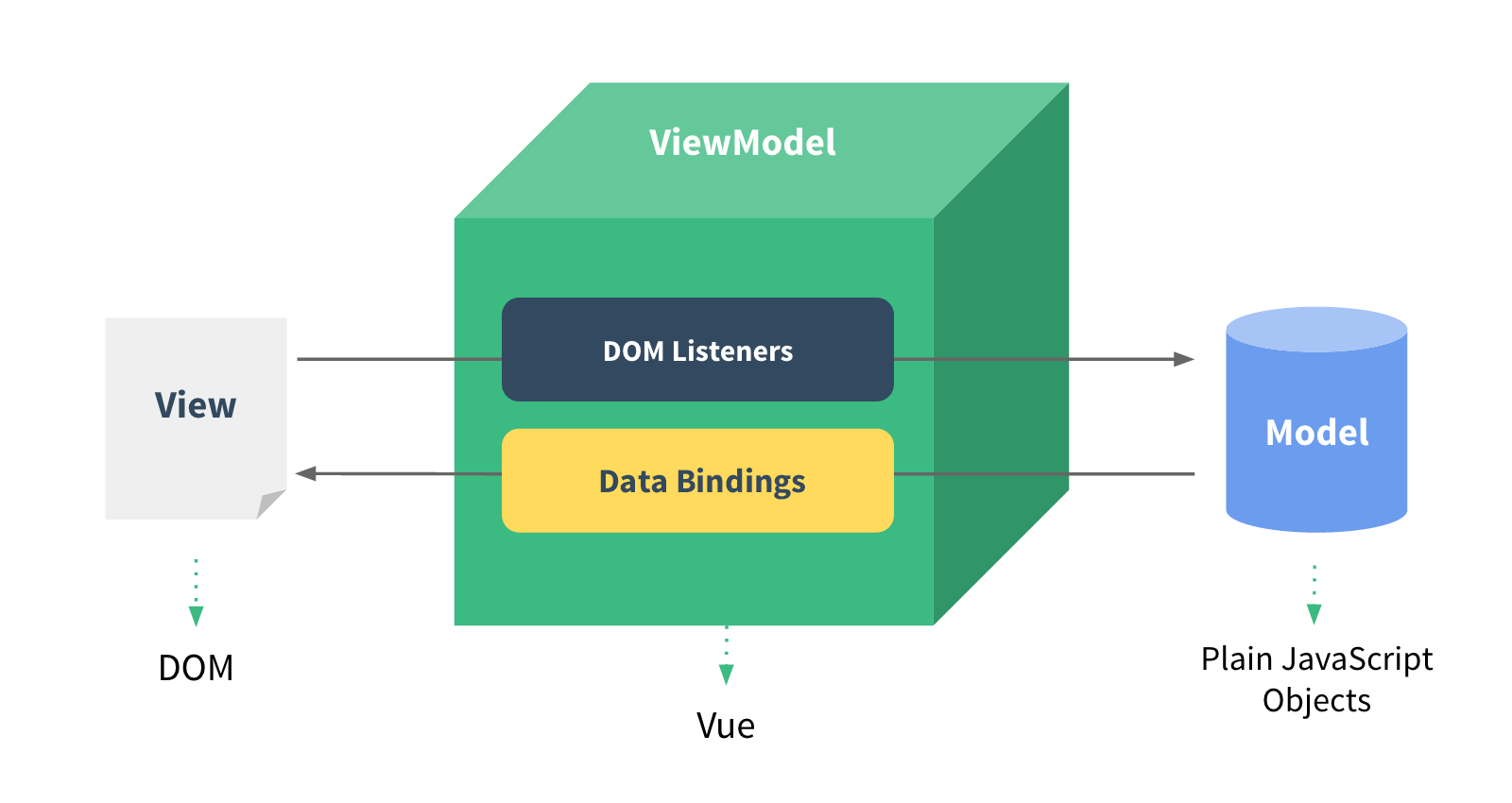


图3.1系统逻辑视图

本毕业设计采用前后端分离的模式，使用的是MVVM架构模式，它分别是由模型(Model)，视图(View)，还有视图模型(ViewModel)，绑定器(Binder)组成。其中，模型(Model)主要是用于对数据库的数据进行业务逻辑相关的操作,也用于描述系统里实体间的关系；视图(View)是用来展现界面的设计结构还有整体的外观；视图模型(ViewModel)里面有绑定器(Binder)，绑定器可以在视图(View)还有数据绑定器之间进行通讯；MVVM采用的是数据的双向绑定，View层如果有什么变化会直接反映到ViewModel上，相反，如果ViewModel上出现什么变化， View也会出现相应的改变。因为采用这种架构模式，所以本设计的前端后端是分离的。

* 1. **开发视图**

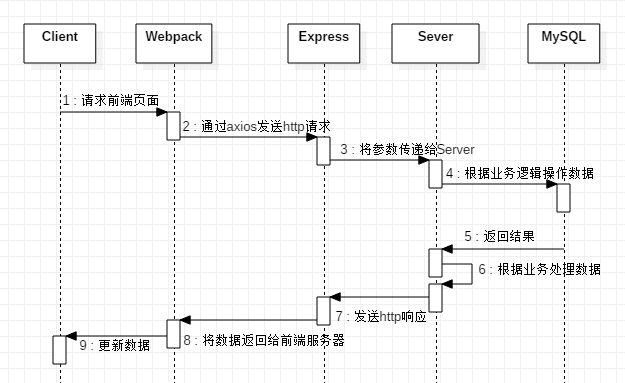


图3.2开发视图

用户端向用vue-cli搭建的前端服务器发送请求前端页面，之后，服务器会通过axios的方法向后端发送http请求，后端服务器由express框架搭建，前端服务器将参数传递给后端服务器Server，然后，后端服务器根据业务逻辑来操作数据库的数据，操作完成之后，数据库返回结果，后端服务器会根据业务对数据进行相应的处理，之后，后端服务器发送http响应给express，之后，express会把数据返回给前端的服务器，前端服务器接收到数据之后，会在用户端页面的相应数据位置进行数据更新。

* 1. **物理视图**

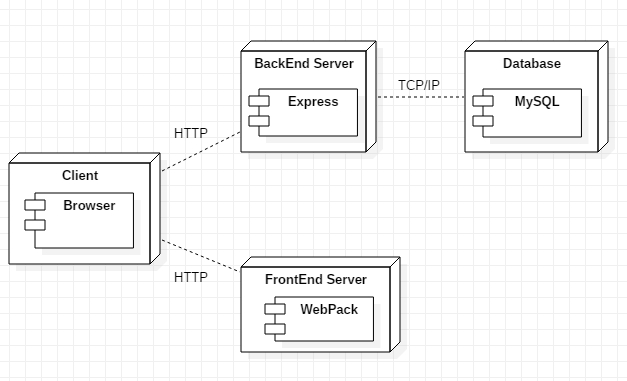


图3.3物理视图

本系统用的是前后端分离的架构，宠物领养网站整体分为前端服务器还有后端服务器两种，前端服务器使用vue框架，主要负责通过接收后端服务器发送过来的数据，对数据进行解析之后把内容展现到网页上，向用户提供页面还有图片等静态信息，后端服务器使用express框架，主要负责接收前端发送过来的数据，对数据库里面的信息进行增删改查操作。用户通过Http请求的形式与前端和后端服务器进行通信，后端服务器通过TCP/IP的方式连接数据库，详细过程如上图所示。

1. **数据库设计**
   1. **数据库ER图**

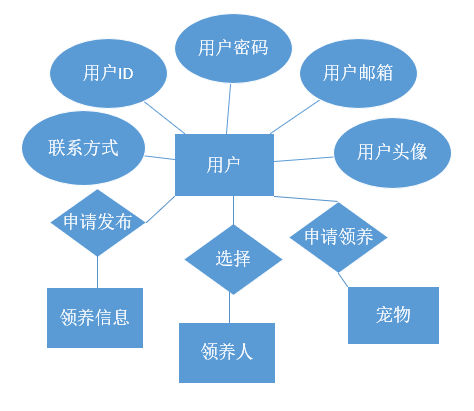


图4.1用户ER图

用户ER图关系到四个实体，分别是用户，领养人，宠物，领养信息。用户具有5个属性：用户ID，用户密码，用户邮箱，用户头像，用户联系方式。另外，用户与其他三个实体的关系如下：用户可以申请发布领养信息，选择宠物的领养人，申请领养流浪宠物。具体实体关系图如上图所示。

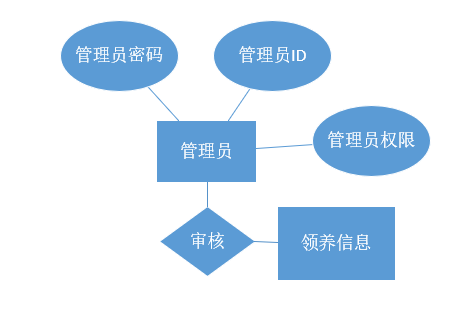


图4.2管理员ER图

管理员ER图涉及到两个实体，管理员还有领养信息，管理员有三个属性：管理员ID，管理员密码，管理员权限登记，管理员与领养信息的关系是管理员可以审核领养信息。

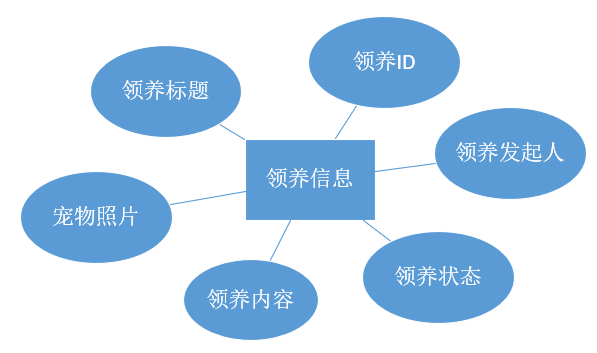


图4.3领养信息ER图

领养信息ER图只涉及一个实体领养信息，实体领养信息有6个属性，分别是领养ID，领养发起人，领养标题，宠物照片，领养内容，领养状态。

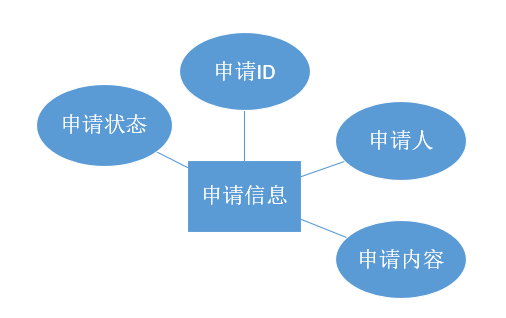


图4.4申请信息ER图

申请信息ER图的实体是申请信息，申请信息实体有4个属性，分别是申请ID申请人，申请人，申请内容，申请状态。

* 1. **数据库表设计**

表4.1用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | User | | | |
| 字段 | 数据类型 | 是否为空 | 含义 | 约束条件 |
| user\_id | varchar | not null | 用户ID | 主键 |
| user\_pwd | varchar | not null | 用户密码 |  |
| user\_email | varchar | null | 用户邮箱 |  |
| user\_icon | varchar | null | 用户头像 |  |
| user\_mobile | varchar | null | 联系方式 |  |

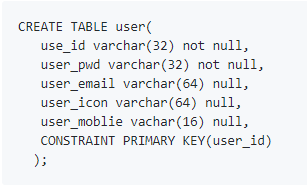


图4.5创建用户表语句

表4.2管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | Administrator | | | |
| 字段 | 数据类型 | 是否为空 | 含义 | 约束条件 |
| admin\_id | varchar | not null | 管理员ID | 主键 |
| admin\_pwd | varchar | not null | 管理员密码 |  |
| admin\_rank | varchar | not null | 管理员权限 |  |

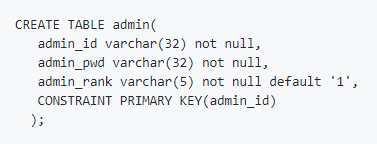


图4.6创建管理员表语句

表4.3领养信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | Adoption | | | |
| 字段 | 数据类型 | 是否为空 | 含义 | 约束条件 |
| ado\_id | varchar | not null | 领养ID | 主键 |
| ado\_master | varchar | not null | 领养发起人 | 外键(user\_id) |
| ado\_title | varchar | not null | 领养标题 |  |
| ado\_image | varchar | null | 宠物照片 |  |
| ado\_content | text | not null | 领养内容 |  |
| ado\_status | varchar | not null | 领养状态 |  |

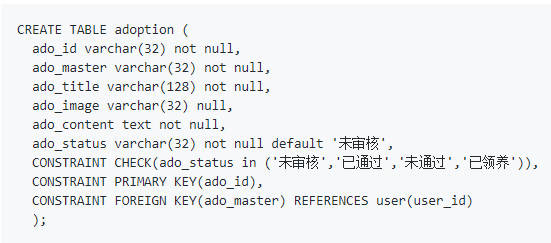
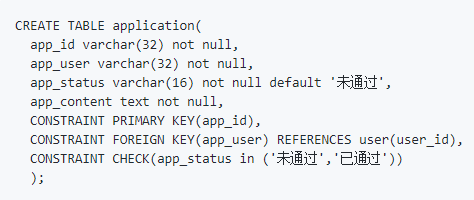


图4.7创建领养信息表语句

表4.4申请领养表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | Application | | | |
| 字段 | 数据类型 | 是否为空 | 含义 | 约束条件 |
| app\_id | varchar | not null | 申请ID | 主键 |
| app\_user | varchar | not null | 申请人 | 外键(user\_id) |
| app\_content | text | not null | 申请内容 |  |
| app\_status | varchar | not null | 申请状态 |  |

  
图4.8创建申请领养表语句

* 1. **数据库逻辑视图**



图4.5数据库逻辑视图

1. **网页设计与代码实现**
   1. **用户端**
      1. **登录账号**

用户在需要登录账号的时候，点击“登录”按钮，输入账号密码之后，页面将账号还有密码数据以http请求的形式发送到服务器上，服务器根据请求当中的url为其分配登录账号的业务逻辑，根据业务逻辑对MySQL数据库中的用户表进行查询，如果在数据库查找到与其中一个用户的账号密码匹配一致，则将成功登录的结果逐层返回，页面提示登陆成功，并且跳转到主界面。

用户登录的时序图和界面图如下图所示：

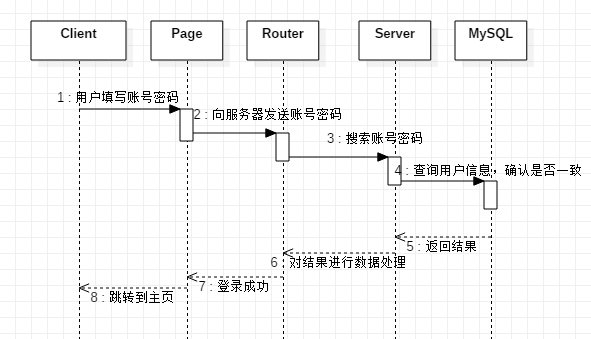


图5.1登录账号时序图



图5.2登录界面

* + 1. **注册账号**

在用户注册账号的时候，先点击页面的注册按钮进入注册页面之后，填写用户名，密码，确认密码，邮箱，联系方式，设置头像了之后，点击注册按钮，网页会把相关的注册信息以http请求的方式发送给服务器，服务器会根据请求的url为其分配注册账户的业务逻辑，之后，网站会在数据库查找是否有相同用户名的用户，若查找出来，则返回到注册页面提示用户，用户名已经存在，请重新设置用户名，若用户名在数据库没有重复出现，则把用户信息插入用户表，之后把结果返回到注册页面上，提示用户注册成功，之后跳转到登录界面进行登陆操作。

注册用户的时序图还有注册界面图如下图所示：

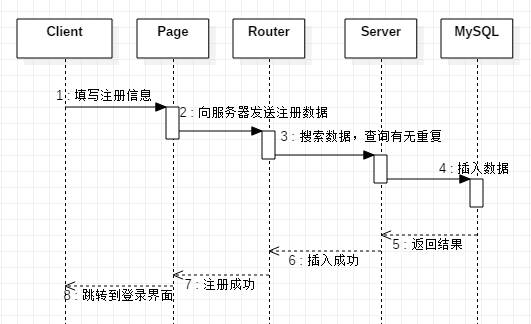


图5.3用户注册时序图



图5.4用户注册界面图

* + 1. **修改密码**

当用户想要进行修改密码操作时，必须先登录成功之后才能进行密码修改。登陆成功之后，进入个人主页，点击修改密码按钮，分别填写原密码，重复填写两次新密码，点击确定按钮，新密码会以http请求的形式发送到服务器，之后服务器会根据里面请求的url分配修改密码的业务逻辑，随后业务逻辑会把用户的旧密码改成后来修改的新密码，数据重新更新，页面返回结果并且提示修改密码成功，页面跳转到个人主页界面。

修改密码的时序图，修改密码页面如图所示：

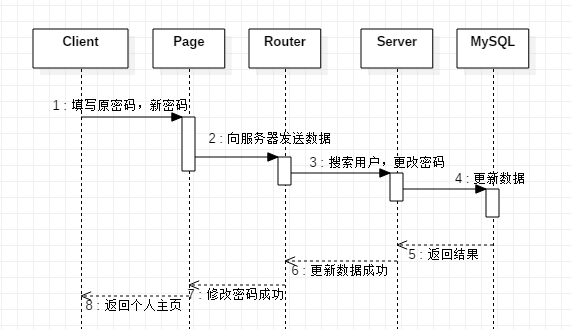


图5.5修改密码时序图

* + 1. **申请发布领养信息**

用户如果想要通过发布领养帖来寻找想要领养宠物的爱心人士 ，可以选择导航栏上“我的申请”下拉菜单当中的“申请发布领养帖”按钮，进入编辑界面，填写好宠物的介绍以及健康状况，联系方式还有地址等基本情况，上传宠物照片之后，点击申请按钮。网页会以http请求的形式把这些数据发送到服务器，服务器根据请求的url分配相关的业务逻辑，根据业务逻辑会把申请信息插入数据库的领养信息表当中，审核状态默认为 “未审核”状态，结果从数据库逐层返回到申请页面，随后页面提示用户申请成功请等待管理员审核。

用户申请发布领养帖的时序图和界面图如下图所示：

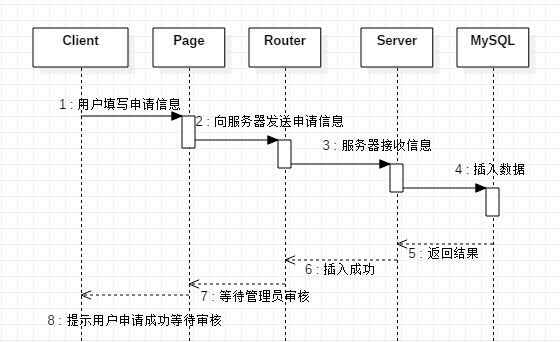


图5.6申请发布领养帖之时序图

* + 1. **查找领养信息**

如果用户想要根据自己的经济条件领养合适自己的宠物，可以通过在查找输入框填写关键词，点击搜索之后，页面把搜索信息用http请求发送到服务器，服务器根据请求当中的url为其分配搜索领养信息的业务逻辑，根据业务逻辑在数据库中对领养信息表进行查找，找出符合关键词要求的领养信息数据，数据库返回查找出来的领养信息数据，并且逐层返回至页面，页面上显示所有符合条件的领养信息给用户浏览。

查找领养信息时序图如图所示：

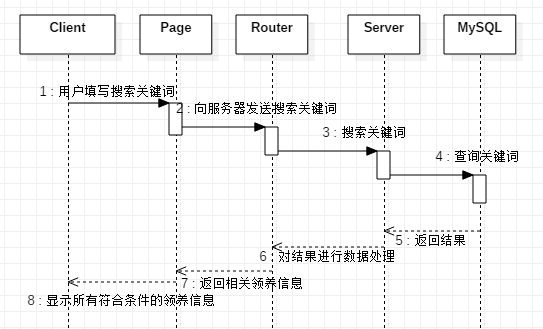


图5.7查找领养信息时序图

* + 1. **选择领养人**

已经发布领养信息的发起人如果想从申请领养的用户当中选择一个作为领养人，可以通过以下步骤选择：首先发起人先在导航栏，在“我的申请”当中选择“申请发布领养帖”，进入页面之后，可以浏览到所有申请领养的用户的记录，选中其中一个用户，选择“同意申请”按钮，则其他用户为“不同意”，页面将用户选择的信息以http请求的方式发送给服务器，服务器接收到之后根据请求的url分配的业务逻辑在数据库当中对领养信息表当中相对应的领养信息记录进行更改，把领养状态更改为“领养成功”。之后数据库把更新后的信息发送给服务器，服务器再返回到页面，提醒发起人“成功选择领养人”，同时也会通知被选择用户“申请领养宠物成功”，未被选中用户则会被提醒“申请领养失败”。

选择领养人的时序图如图所示：

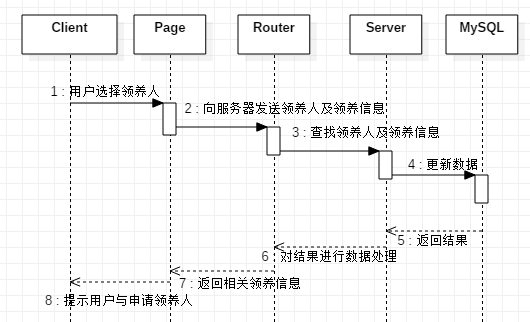


图5.8选择领养人时序图

* + 1. **隐藏已经成功领养的领养信息**

当领养信息的“领养状态”显示“领养成功”之后，也就是当用户成功选择领养人之后，页面会发送http请求把领养成功的信息发送给服务器，服务器根据请求的url分配领养成功的业务逻辑，根据业务逻辑将从数据库搜索相关的领养信息，并且对其进行页面隐藏，即在罗列领养信息的界面不显示已经成功领养的领养信息。

隐藏已经成功领养的领养信息时序图如图所示：

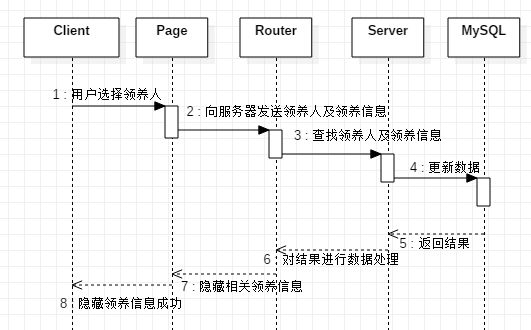


图5.9隐藏成功领养的领养信息时序图

* 1. **管理员端**
     1. **登录账号**

管理员需要登录账号的时候，直接进入管理员端的登录网页，填写账号和密码，点击登录按钮，页面会以http请求的方式向服务器发送登录信息，服务器收到之后，根据请求的url分配管理员登录的业务逻辑，根据登录的业务逻辑在数据库里搜索管理员的用户名和密码，对比查看是否一致，若一致则返回结果，页面提示登录成功，之后跳转至管理员主界面。

管理员登录的时序图如图所示：

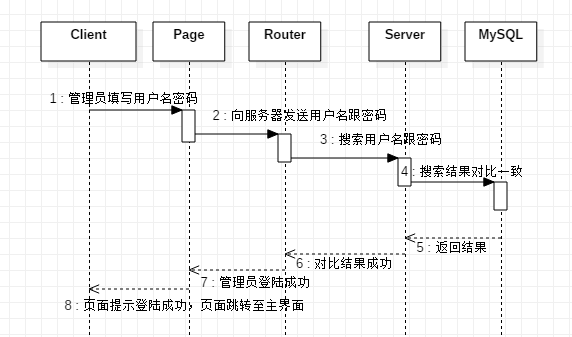


图5.10管理员登录时序图

* + 1. **显示未审核的领养信息**

管理员登录之后，点击“审核领养信息”按钮，页面以http请求的形式发送到服务器，服务器根据请求的url分配审核领养信息的业务逻辑，根据业务逻辑，数据库从领养信息表当中搜索未经过审核的领养信息，提取领养信息，将搜索结果返回给服务器，服务器返回未经过审核的领养信息，并且在页面上显示出来。

显示未审核领养信息的时序图如图所示：

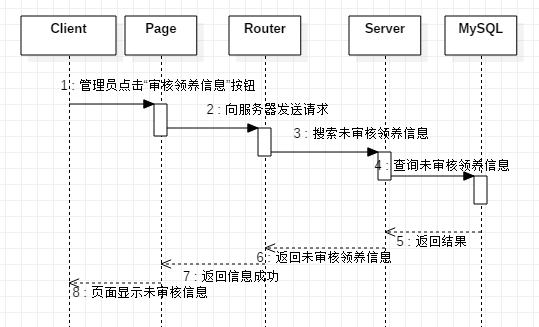


图5.11显示未审核领养信息时序图

* + 1. **审核领养信息**

管理员点击“审核领养信息”按钮，网页跳转到相应界面，显示出所有未审核领养信息，之后管理员可以点击其中任意一个领养信息查看详情，在核实了领养信息的真实性还有正确性之后，管理员点击“审核通过”按钮，网页以http请求的形式发送到服务器，服务器分配相应的审核信息业务逻辑，业务逻辑对数据库的领养信息表当中的对应信息的审核状态进行修改，把该领养信息审核状态修改为“审核通过”，数据库返回结果，结果逐层返回到网页，给用户浏览的领养信息当中会显示出之前刚刚通过审核的领养信息。

审核领养信息时序图如图所示：

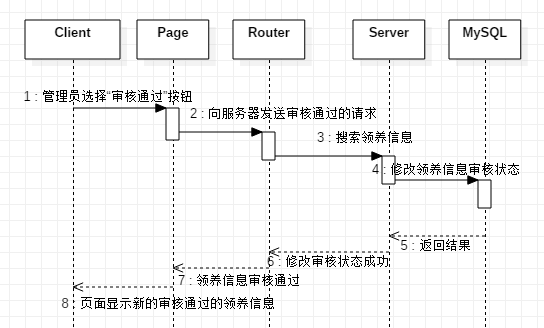


图5.12审核领养信息时序图

* + 1. **修改领养信息**

管理员点击“修改领养信息”按钮，进入管理页面，页面显示出所有通过审核的领养信息，选中需要修改的领养信息，点击之后进入对应页面，重新编辑好需要修改的信息内容，点击“确定“按钮，网页会以http请求的形式把修改好的领养信息发送到服务器，服务器会根据请求的url分配修改信息的业务逻辑，根据业务逻辑对数据库的领养信息表的对应领养信息进行修改操作，之后返回结果，修改成功，原来的领养信息更新成修改后的领养信息，页面提示管理员修改信息成功。

修改领养信息时序图如下图所示：

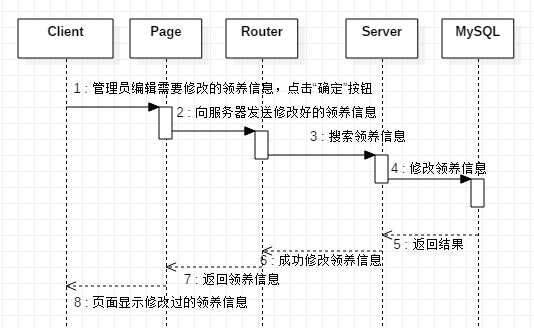


图5.13修改领养信息时序图

* + 1. **删除领养信息**

当管理员想要删除领养信息方便对其他的领养信息进行管理的时候，点击“删除领养信息”按钮，页面会显示全部的领养信息，选中想要删除的领养信息，点击“删除”按钮，页面提示“是否确定删除该领养信息”，选择“是”按钮，则页面会向服务器发送http请求，服务器根据请求的url分配删除领养信息的业务逻辑，根据业务逻辑在数据库中搜索对应的领养信息，执行删除操作，之后删除成功，逐层返回结果，页面提示“删除成功”，则用户界面还有管理员管理领养信息界面都不会显示删除的领养信息。

删除领养信息的时序图如下图所示：

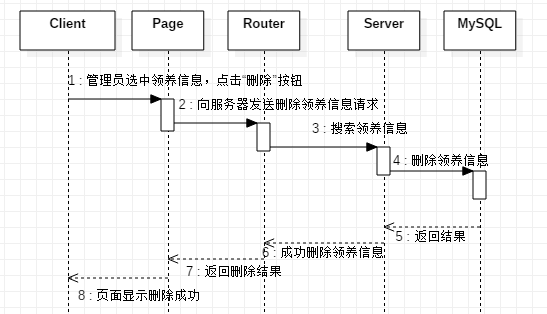


图5.14删除领养信息时序图

1. **安装与测试**
   1. **运行环境简介**
2. 浏览器兼容：chrome73.0(64位)；
3. 开发环境：ATOM；
4. 数据库环境：MYSQL5.7.21；
5. 其他：Vue.js、Node.js；
   1. **测试内容**

结合本宠物领养网站的实际功能，我进行了如下测试：

1. 准确性：本宠物领养网站进行操作后的结果是否正确，界面显示是否完整。
2. 易用性：对本宠物领养网站的操作是否简单，是否适合用户操作。
3. 容错性：当网站出现错误的时候，是否能够友好的提示用户。
4. 功能性：网站功能是否完整，功能是否能正常运行。
   1. **测试操作步骤**

对宠物领养网站测试的操作步骤如下所示：

1. 先从github上下载全部程序；
2. 点开项目文件夹，进入BackEnd文件夹，打开命令行窗口，输入命令node app.js回车之后，后端模块启动；
3. 进入项目文件夹的FontEnd文件夹，点开project文件夹，打开命令行窗口，输入命令npm start即可启动前端模块，
4. 前后端模块同时启动之后，在网页上输入<http://localhost:8080/views/home>就可以进入网站的主界面；
5. 根据网站的提示进行测试；
   1. **测试用例**
      1. **登录账号**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例1：登录账号 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 选中“登录”按钮 | 系统跳转到  登录界面 | 成功进入登录界面 |
| 2 | 填写用户名和密码，点击“登录”按钮 | 系统提示登陆成功，跳转到主界面 | 成功登录，页面跳转到主界面 |

* + 1. **注册账号**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例2：注册账号 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 选中“注册”按钮 | 系统进入注册界面 | 页面成功跳转到注册界面 |
| 2 | 填写用户名，两次密码，邮箱，联系方式，设置头像，点击注册 | 网页提示注册成功，页面跳转到登录界面 | 注册成功 |

* + 1. **修改密码**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例3：修改密码 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 点击“修改密码”按钮 | 页面跳转到修改密码界面 | 成功跳转 |
| 2 | 填写原密码，再填写两次新密码，点击“确定” | 系统提示修改密码成功 | 密码修改成功 |

* + 1. **查找领养信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例4：查找领养信息 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 用户进入主界面 | 进入主界面 | 成功进入主界面 |
| 2 | 在搜索框里填写关键词，点击“搜索”按钮 | 搜索结果显示出相应领养信息 | 显示所有符合条件的领养信息 |

* + 1. **申请发布领养信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例5：申请领养发布信息 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 用户进入主界面 | 进入主界面 | 成功进入主界面 |
| 2 | 点击顶端导航栏的“我的申请”当中的“申请发布领养帖” | 页面跳转到编辑申请页面 | 系统成功跳转到编辑申请界面 |
| 3 | 编辑宠物的基本信息，上传宠物照片，点击“申请”按钮 | 申请递交，提示用户已上传申请 | 页面提示申请成功，请等待审核结果 |

* + 1. **申请领养**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例6：申请领养 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 进入主界面 | 进入主界面 | 进入主界面 |
| 2 | 点击顶部导航栏“我的申请”当中的“申请领养” | 系统跳转到编辑申请领养界面 | 系统成功跳转到申请领养界面 |
| 3 | 填写基本信息，联系方式，点击“申请” | 系统提示“申请成功，请等待回复” | 递交申请成功，系统提示申请成功 |

* + 1. **审核领养信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例7：审核领养信息 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 管理员进入主界面 | 进入主界面 | 成功进入主界面 |
| 2 | 点击“审核领养信息”按钮 | 跳转到审核信息页面 | 进入审核信息页面 |
| 3 | 选择领养信息进行审核，决定信息是否通过审核 | 审核通过，主界面显示相应的领养信息，否则，领养信息审核状态为“未通过” | 可以成功对领养信息进行审核操作 |

* + 1. **修改领养信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例8：修改领养信息 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 管理员进入管理员主界面 | 进入管理员主界面 | 进入管理员主界面 |
| 2 | 点击“修改领养信息”按钮 | 页面跳转到修改领养信息页面 | 页面成功跳转 |
| 3 | 点击想要修改的领养信息的页面，对需要改正的内容信息进行修改，点击“确认”按钮 | 领养信息成功被修改，页面提示修改成功 | 成功修改领养信息，页面提示修改成功 |

* + 1. **删除领养信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例9：删除领养信息 | | | |
| 步骤 | 操作描述 | 预测结果 | 运行结果 |
| 1 | 管理员进入主界面 | 进入主界面 | 成功进入主界面 |
| 2 | 点击“删除领养信息”按钮 | 页面跳转到显示所有领养信息的页面 | 页面成功跳转到显示所有领养信息的页面 |
| 3 | 点击想要删除的领养信息，选择“删除”按钮 | 领养信息成功被删除 | 成功删除领养信息 |

1. **结论**

通过这次毕业设计，我完整的体验了一次项目开发的全过程，从需求分析到概要设计，再到数据库设计，项目设计实现，每一步都需要认真思考学习。在开始项目设计之前，我用vue-cli搭建前端服务器用于开发前端页面，用express搭建后端服务器，搭建环境的时候，我了解到了本地还有全局的区别，慢慢的去消化使用两个框架所要掌握的知识。在我开始写界面的时候，我接触到了get还有post方法，学会使用了前端路由器，运用了axios的方法来让前后端进行交互。期间遇到了好多奇奇怪怪的错误，有的时候页面显示不出来，都不知道到底是哪里出现了问题，好不容易解决了之后，会觉得自己还是太菜了，还需要努力学习啊。

本来还觉得，其实html还有css还挺好学，w3s刷几遍应该就掌握了，后来发现自己太过天真，写前端的时候才发现自己的浅薄，有一些页面显示出来的效果跟自己逻辑当中的有差别，要用css调整好久，感觉在页面美化方面还是很欠缺的。希望在今后的学习过程当中能够扎扎实实的把基础打好，每天进步一点点。

**参考文献**

［1］邱俊涛．《轻量级web应用开发》［M］．人民邮电出版社，2015年7月第1版.  
［2］Nicholas C.Zakas. 《JavaScript高级程序设计》［M］. 人民邮电出版社，2012年3月第1版.  
［3］Douglas Crockford. 《JavaScript语言精粹》［M］. 电子工业出版社，2012年9月第2版.  
［4］Jeremy Keith，Jeffrey Sambells. 《JavaScript DOM编程艺术》［M］. 人民邮电出版社，2011年4月第1版.

［5］陈火旺，徐建波. 《Web设计原理与编程艺术》［M］.中南大学出版社，2005版.  
［6］梁灏. 《Vue.js实战》［M］. 清华大学出版社，2017.10.  
［7］忽如寄. 《Node.js开发实战》［M］. 清华大学出版社，2017.10.  
［8］程杰. 《大话设计模式》［M］. 清华大学出版社，2007.12.

［9］朴灵. 《深入浅出Node》［M］. 人民邮电出版社，2013.12.

［10］李雁翎. 《数据库技术及应用》［M］. 高等教育出版社，2007.8.

**谢 辞**

本次毕业设计的完成离不开王庆娟老师的指导还有帮助，对我来说，她就像姐姐一样很有耐心，很有亲切感。在大学四年的时光里，我深知自己的笨拙，很感谢在这四年时间里面遇到的老师还有同学给予我的耐心的帮助还有理解，每当在我无助还有遇到学习瓶颈的时候，感谢我的同学兼好友少君安慰我，提供给我学习方法，教我如何解决学习问题；也很感谢好友玉敏在我每次有烦恼的时候跟我一起分担；当然，我还要特别感谢的就是我的男朋友，感谢他在跟我在一起的时间里没有嫌我笨，每一次都极其耐心还有认真的教我学习，虽然我很多次都惹他生气，但是他最后还是很有耐心教我，让我学习了一些课堂之外的前端方面的知识，引导我打开前端世界的大门。时间真的过得好快，转眼之间就要毕业了，有好多遗憾的事情，也有值得开心的回忆，带上对大家的感谢还有对未来的期待，希望未来的我能比现在的自己更好。

**附 录**