 **肇庆学院**

**计算机科学与软件学院、大数据学院**

**作业： 设计与开发文档**

**老师： 岑宇森**

**项目名： 动物之家**

**组员： 郑锦铭（201624133236）**

**林凯城（201624133215）**

**张康泽（201624133242）**

**评分：**

目录

[1、引言 2](#_Toc5870316)

[1.1、团队介绍 2](#_Toc5870317)

[1.2、背景 2](#_Toc5870318)

[1.3、定义 2](#_Toc5870319)

[2、需求分析 3](#_Toc5870320)

[2.1、功能介绍 3](#_Toc5870321)

[2.1.1、“宠物去哪” 3](#_Toc5870322)

[2.1.2、“带我回家” 3](#_Toc5870323)

[2.1.3、“问我问我” 4](#_Toc5870324)

[2.1.4、“宠物圈” 4](#_Toc5870325)

[2.1.5、“萌宠商城” 5](#_Toc5870326)

[2.2、可行性分析 5](#_Toc5870327)

[2.2.1、技术可行性分析 5](#_Toc5870328)

[2.2.2、经济可行性分析 6](#_Toc5870329)

[2.2.3、社会可行性 7](#_Toc5870330)

[2.2.4、业务可行性分析 7](#_Toc5870331)

[2.3、数据流图 8](#_Toc5870332)

[2.3.1、顶层数据流图 8](#_Toc5870333)

[2.3.2、0层数据流图： 11](#_Toc5870334)

[2.3.3、1层数据流图 18](#_Toc5870335)

[2.4、IPO表 25](#_Toc5870336)

[2.5、实体关系图 31](#_Toc5870337)

[2.6、数据字典 41](#_Toc5870338)

[3、详细设计 56](#_Toc5870339)

[3.1、词汇表 56](#_Toc5870340)

[3.2、硬件组成. 58](#_Toc5870341)

[3.3、软件架构与开发环境 60](#_Toc5870342)

[3.3.1、软件架构 60](#_Toc5870343)

[3.3.2、开发环境 60](#_Toc5870344)

[3.4、系统接口表 61](#_Toc5870345)

[3.5、编程规范及要求 67](#_Toc5870346)

[3.6、进度计划 70](#_Toc5870347)

[4、其它 72](#_Toc5870348)

[4.1、特色与优势. 72](#_Toc5870349)

[4.1.1、软硬结合 72](#_Toc5870350)

[4.1.2、受众范围广 72](#_Toc5870351)

[4.1.3、“刚需品” 72](#_Toc5870352)

[4.1.4、可持续发展 73](#_Toc5870353)

[4.2、未来与发展 73](#_Toc5870354)

# 1、引言

1.1、团队介绍

郑锦铭（201624133236），

林凯城（201624133215），

张康泽（201624133242）

1.2、**背**景

当今，流浪动物横生，一是人类的弃养，二是流浪动物之间的繁殖。它们数量众多，并且在这种现代化城市中难以生存。许多动物都病衰残疾，不但可怜，而且影响环境。而各地的动物保护组织的作用也并不明显，因为它们给予的帮助是有限的。

还有就是，每个饲主最最最担心的问题，莫过于自家宠物突如其来的疾病或走失，当宠物生病时，网上查找的方法鱼龙混杂，让人更加无从下手；想问人却求之无门，身旁没有与之相关的朋友；想上医院却不知附近哪里有宠物医院。。。

比起生病，宠物们毫无征兆地走失才是令人痛苦，饲主和宠物们的感情堪比亲情友情，而宠物的走失，犹如一把钝刀在你心头上反复割锯。

以上种种问题，该如何解决呢？

1.3、定义

本产品是基于物联网的一站式宠物管理平台。

2、需求分析

2.1、功能介绍

2.1.1、“宠物去哪”

该功能结合一款定位项圈，当宠物莫名走失时，只需打开网页，立刻查看宠物位置，重新找回它。

该功能还能记录宠物每天的行走路线，看看你家宠物是不爱出门的死宅，还是到处游荡的浪子。

项圈上装有震动板，你只需稍微训练一下宠物，当线圈发出震动时，就是提示它该回家吃饭了。

2.1.2、“带我回家”

许多饲主都有这样困惑，家里的宠物太会繁衍了，每胎都是好几只，想送人却没有门路，自己养却有心无力；或家里发生变迁，宠物无法继续饲养时，这些宠物何去何从？

而对于城市来说，流浪动物是一项困扰，由于无人照管，他们极易病死饿死于街头，极大影响城市市容；

因此，必须有一个平台，能够将这些被遗弃的，待遗弃的，无力饲养的动物的信息，集中并发布到这上面来，以此来寻找那些想要养宠物的爱心人士。这样，就能更好地解决城市的流浪动物问题，给待遗弃宠物提供了一条新的去处。

2.1.3、“问我问我”

对于每个铲屎官而已，宠物生病时令人焦灼，网上查的方法良莠不齐，让人更加无从下手。

因此，“动物之家”提供了一个权威的“健康资料”数据库，里面的每一条资料都经过专家的审核，而不是以讹传讹的假数据。让铲屎官们“有资料可查“。

除此之外，我们还提供了“在线医生”功能，当资料查不出解决方法时，几十位专家们在线帮助你，他们将会在接收到求助信息后，第一时间给你解决方案，做到“有专家可问”。

如果以上都无法解决宠物的病情时，或病情十分严重时，最好的方案是立即送往医院。“附近医院”功能，就能快速规划好最近最好的宠物医院路线，不让病情耽误一刻，做到“有医院可治”。

2.1.4、“宠物圈”

以上功能都只是“需时才用”，即用户只有在想要用的时候才会打开该网站，因此需要有一个功能来增加用户流量。

“宠物圈”即能满足该需求，他是一个以动物为主题的论坛，用户可以在上面发布各种动物趣图，或晒出自家宠物，让浏览者忍不住地往下刷；并且还能在宠物圈上交好友，约附近的人一起出来遛猫遛狗，晒宠物，其乐融融。

2.1.5、“萌宠商城”

除了以上功能，我们还需要一个功能来维持我们项目的资金链，“萌宠商城”则负责这一关键点。它除了卖我们自主开发的“宠物定位项圈”，还有其它各式各样的宠物产品，比如食物，玩具，宠物衣服，宠物生活用品等。以此来提供我们的收入。

2.2、可行性分析

2.2.1、技术可**行性分**析

#### 技术风险及规避方法:

产品使用springboot+bootstrap+thymeleaf开发动态响应式网站，数据存储、安全管理、权限验证、任务列表等技术等使用springboot框架的。

为了进行开发，小组成员将进行为期两周的springboot和thymeleaf技术学习，提高技术储备，同时在网上寻找一份bootstrap前端模板套用。

暂时无法解决的问题:集成到硬件上的gps定位功能

解决方法:基本功能完成后，购买硬件接口集成到本产品中

#### 产品环境依赖要求

适配IE6以上，火狐、谷歌、360等绝大部分浏览器

匹配不同屏幕大小的要求

设配win98/win2002/win7/win10

2.2.2、经济可行性分析

本软件产品是小组成员自主开发，成本预算=硬件成本+人工开发成本+调研成本+后期维护成本+服务器

|  |  |  |
| --- | --- | --- |
| 类别 | 预算 | 用途 |
| 硬件成本 | 300 | 购买GPS定位器 |
| 人工开发成本 | 劳动成本 | 开发整个软件系统 |
| 调研成本 | 300 | 购买宣传单 |
| 后期维护成本 | 劳动成本 | 系统维护 |
| 服务器 | 180 | 3个阿里云学生服务器半年 |

总体成本=780元+劳动成本

收益：

广告费+跟第三方合作费用+产品销售费用+产品功能的服务费

隐性价值:良好的口碑和庞大数量的用户群体

2.2.3、社会可行性

* **道德方面:**

迎合当前背景下人们对宠物的喜爱,为宠物寻找一个温暖的怀抱

* **法律方面:**

本平台上的宠物是免费领养的，没有把动物商业化，宗旨是为了救助了流浪动物以及改善城市环境。是正能量的平台

* **社会影响力:**

大批的用户参入到我们的平台上，会形成一股能量，让爱护动物的风气在社会上盛行

* **自有资源:**

平台上庞大的用户群体

2.2.4、业务可行性分析

本软件产品设计信息发布、安全管理、权限认证、gps定位、产品购买5大模块的功能。

就目前技术来说，经过前期的技术储备之后，除了产品购买和GPS功能暂未实现后，其他功能皆可以实现。

2.3、数据流图

2.3.1、顶层数据流图





















2.3.2、0层数据流图：





































2.3.3、1层数据流图



























2.4、IPO表

|  |  |  |  |
| --- | --- | --- | --- |
| 元素编号 | 名称 | 类型 | 说明 |
| IPO1.1 | 数据管理 | 处理 | 数据管理操作 |
| IPO1.1.1 | 用户管理 | 处理 | 管理员可以用户管理 |
| IPO1.1.2 | 商品管理 | 处理 | 管理员可以商品管理 |
| IPO1.1.3 | 健康资料管理 | 处理 | 管理员可以健康资料管理 |
| IPO1.1.4 | 宠物圈管理 | 处理 | 管理员可以宠物管理 |
| IPO1.1.5 | 领养动物管理 | 处理 | 管理员可以领养动物管理 |
| IPO2.1 | 健康资料管理 | 处理 | 医生可以进行健康资料管理 |
| IPO2.1.1 | 个人健康资料列表显示 | 处理 | 医生可以查看个人商品列表 |
| IPO2.1.2 | 健康资料发布 | 处理 | 医生可以发布健康资料 |
| IPO2.1.3 | 健康资料删除 | 处理 | 医生可以删除资料 |
| IPO2.1.4 | 健康资料编辑 | 处理 | 医生可以编辑资料 |
| IPO2.2 | 健康资料查询 | 处理 | 用户可以健康资料查询 |
| IPO2.3 | 健康资料收藏 | 处理 | 用户可以进行健康资料收藏 |
| IPO3.1 | 商品管理 | 处理 | 商家可以对商品进行管理 |
| IPO3.1.1 | 个人商品列表显示 | 处理 | 商家可以查看个人商品列表 |
| IPO3.1.2 | 商品发布 | 处理 | 商家可以发布商品 |
| IPO3.1.3 | 商品下架 | 处理 | 商家可以下架商品 |
| IPO3.1.4 | 商品删除 | 处理 | 商家可以删除商品 |
| IPO3.1.5 | 商品编辑 | 处理 | 商家可以编辑商品 |
| IPO3.2 | 商品购买 | 处理 | 用户可以购买商品 |
| IPO3.2.1 | 判断库存 | 处理 | 系统判断商品库存 |
| IPO3.2.2 | 计算价格 | 处理 | 系统计算商品价格 |
| IPO3.2.3 | 确定购买 | 处理 | 用户确认购买商品 |
| IPO3.3 | 商品评论 | 处理 | 用户可以对商品进行评论 |
| IPO4.1 | 领养动物管理 | 处理 | 用户可以领养动物管理 |
| IPO4.1.1 | 个人发布列表显示 | 处理 | 用户可以查看自己发布的领养动物列表 |
| IPO4.1.2 | 领养动物发布 | 处理 | 用户可以发布领养动物信息 |
| IPO4.1.3 | 领养动物删除 | 处理 | 用户可以删除掉领养动物信息 |
| IPO4.1.4 | 领养动物编辑 | 处理 | 用户可以编辑发布的领养动物信息 |
| IPO4.2 | 申请领养动物 | 处理 | 用户可以申请领养动物 |
| IPO5.1 | 个人宠物圈管理处理 | 处理 | 用户可以进行个人宠物圈管理 |
| IPO5.1.1 | 查询个人宠物圈 | 处理 | 用户可以查看个人宠物犬 |
| IPO5.1.2 | 修改某条宠物圈 | 处理 | 用户可以修改个人宠物圈 |
| IPO5.1.3 | 删除宠物圈 | 处理 | 用户可以删除个人宠物圈 |
| IPO5.1.4 | 发表宠物圈 | 处理 | 用户可以发表个人宠物圈 |
| IPO5.2 | 好友宠物圈事务处理 | 处理 | 用户可以对好友宠物圈进行实物处理 |
| IPO5.2.1 | 查询好友宠物圈 | 处理 | 用户可以查询好友宠物圈 |
| IPO5.2.2. | 评论宠物圈 | 处理 | 用户可以评论好友宠物圈的信息 |
| IPO5.2.3 | 收藏宠物圈 | 处理 | 用户可以收藏宠物圈 |
| IPO5.2.4 | 点赞宠物圈 | 处理 | 用户可以点赞宠物圈 |
| IPO6.1 | 动物位置定位 | 处理 | 用户可以对动物进行位置定位 |
| IPO6.1.1 | 动物位置定位 | 处理 | Gps在动图上对动物进行定位 |
| IPO6.1.2 | 显示到达路线 | 处理 | 系统在地图上显示路线 |
| IPO7.1 | 注册事务处理 | 处理 | 游客可以通过注册成为用a户 |
| IPO7.1.1 | 注册信息验证 | 处理 | 系统对游客注册信息验证 |
| IPO7.1.2 | 激活码发送到邮箱 | 处理 | 系统把激活码发送到游客邮箱 |
| IPO7.1.3 | 游客输入激活码 | 处理 | 游客输入激活码 |
| IPO7.1.4 | 激活码验证 | 处理 | 系统进行激活码验证 |
| IPO7.2 | 登录事务处理 | 处理 | 用户可以登录访问本网站 |
| IPO7.2.1 | 登录信息验证 | 处理 | 系统进行登录信息验证 |
| IPO8.1 | 收藏列表处理 | 处理 | 用户可以对个人收藏列表进行管理 |
| IPO8.2 | 个人信息处理 | 处理 | 用户可以进行个人信息处理 |
| IPO8.3 | 购物车处理 | 处理 | 用户可以对个人购物车进行管理 |
| IPO8.4 | 订单处理 | 处理 | 用户可以对个人订单进行处理 |
| IPO9.1 | 发起审核申请 | 处理 | 用户可以发起成为医生或者商家的审核请求 |
| IPO9.2 | 审核事务处理 | 处理 | 管理员对用户的审核请求进行审核 |
| IPO9.3 | 用户权限赋予 | 处理 | 管理员对用户赋予权限 |
| IPO10.1 | 聊天信息处理 | 处理 | 用户可以聊天 |
| IPO10.1.1 | 接收并封装信息 | 处理 | 系统接受并封装信息 |
| IPO10.1.2 | 判断聊天目标是好友还是群聊 | 处理 | 判断聊天目标是好友还是群聊 |
| IPO10.1.3 | 判断用户是否在线并发送聊天信息 | 处理 | 判断用户是否在线并发送聊天信息 |
| IPO10.1.4 | 向在线的群员发送聊天信息 | 处理 | 向在线的群员发送聊天信息 |
| IPO11.1 | 好友分组管理处理 | 处理 | 用户可以管理好友分组 |
| IPO13.1 | 添加群组事务处理 | 处理 | 用户可以申请添加群组 |
| IPO13.1.1 | 接收加群信息 | 处理 | 系统加收加群信息 |
| IPO13.1.2 | 将加群信息发送给群组管理员审核 | 处理 | 用户将加群信息发送给群组管理员审核 |
| IPO13.1.3 | 根据加群信息处理结果判断是否添加群员信息 | 处理 | 系统根据加群信息处理结果判断是否添加群员信息 |
| IPO14.1 | 退群事务处理 | 处理 | 用户可以申请退群 |
| IPO15.1 | 建群事务处理 | 处理 | 用户可以申请建群 |
| IPO15.1.1 | 接收建群申请数据 | 处理 | 系统接收用户的申请加群信息 |
| IPO15.1.2 | 判断建群是否合法 | 处理 | 系统判断用户的建群是否合法 |
| IPO15.1.3 | 存储群组头像 | 处理 | 系统存储群组头像 |
| IPO16.1 | 更新群组分组事务处理 | 处理 | 系统自动更新群组信息 |
| IPO17.1 | 添加好友事务处理 | 处理 | 用户可以添加好友 |
| IPO17.1.1 | 接收好友申请消息 | 处理 | 系统接收用户的申请添加好友信息 |
| IPO17.1.2 | 将好友申请消息发送给目标审核 | 处理 | 系统将好友申请消息发送给目标审核 |
| IPO17.1.3 | 根据好友申请消息处理结果判断是否添加好友信息 | 处理 | 系统根据好友申请消息处理结果判断是否添加好友信息 |
| IPO18.1 | 删除好友事务处理 | 处理 | 用户可以删除好友 |
| IPO19.1 | 解散群组处理 | 处理 | 用户可以解散群组 |
| IPO19.1.1 | 接收解散群组信息 | 处理 | 系统接收群组用户发送的解散好友信息 |
| IPO19.1.2 | 获取并删除群组成员 | 处理 | 系统获取并删除群组成员 |
| IPO19.1.3 | 向所有群组成员发送群组解散信息 | 处理 | 系统向所有群组成员发送群组解散信息 |
| IPO19.1.4 | 删除群组 | 处理 | 系统删除群组 |
| DS1 | 用户信息 | 数据存储 | 保存用户个人信息 |
| DS2 | 商品信息 | 数据存储 | 保存商品信息 |
| DS3 | 健康资料信息 | 数据存储 | 保存医生发布的资料信息 |
| DS4 | 宠物圈信息 | 数据存储 | 用户发布的宠物圈信息 |
| DS5 | 领养动物信息 | 数据存储 | 保存用户发布的营养动物信息 |
| DS6 | 订单信息 | 数据存储 | 保存用户订单 |
| DS7 | 消息盒子信息 | 数据存储 | 保存未读消息 |
| DS8 | Gps用户地址信息 | 数据存储 | 保存用户地址信息 |
| DS9 | 动物地址信息 | 数据存储 | 保存动物地址 |
| DS10 | 激活码信息 | 数据存储 | 保存激活码信息 |
| DS11 | 健康资料收藏信息 | 数据存储 | 保存用户收藏的健康资料信息 |
| DS12 | 商品评论信息 | 数据存储 | 保存商品的评论信息 |
| DS13 | 领养动物申请信息 | 数据存储 | 保存领养动物申请信息 |
| DS14 | 购物车信息 | 数据存储 | 保存购物车信息 |
| DS15 | 审核信息 | 数据存储 | 保存审核信息 |
| DS16 | 商家信息 | 数据存储 | 保存商家信息 |
| DS17 | 医生信息 | 数据存储 | 保存医生信息 |
| DS18 | 聊天信息 | 数据存储 | 保存聊天信息信息 |
| DS19 | 好友分组信息 | 数据存储 | 保存好友分组信息 |
| DS20 | 群组成员信息 | 数据存储 | 保存群组成员信息 |
| DS21 | 群组信息 | 数据存储 | 保存群组信息 |
| DS22 | 好友信息 | 数据存储 | 保存好友信息 |

2.5、实体关系图

用户实体描述



医生实体描述



商家实体描述



用户权限表：



商品实体描述



订单实体描述



订单商品列表



商品评论实体描述



宠物圈实体描述



宠物圈评论表



资料实体描述



领养信息实体描述



领养动物申请实体描述



审核实体描述



好友聊天记录表



好友分组表



群组表



群员信息表



消息盒子表



用户登陆记录实体描述



宠物圈收藏表



资料收藏表



商品购物车表



领养动物信息收藏表



激活码表:



2.6、数据字典

用户数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS1 af\_user | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| u\_id | int |  |  | NOT NULL | 用户id |
| u\_name | varchar | 50 |  | NOT NULL | 用户名 |
| u\_account | int | 20 |  | NOT NULL | 用户账号 |
| u\_password | varchar | 50 |  | NOT NULL | 密码 |
| u\_phone | varchar | 15 |  |  | 手机号 |
| u\_email | varchar | 30 |  | NOT NULL | 邮箱 |
| u\_sex | int |  |  | 2 | 性别 0男性/1女性/2保密 |
| u\_register\_time | datetime |  |  | 自动生成 | 注册日期 |
| u\_introduce | text |  |  |  | 自我介绍 |
| u\_headimage | varchar | 100 |  | 有默认值 | 头像图片路径 |
| u\_state | int |  |  | 0 | 状态 0不可用（激活邮件后变成正常），1正常 ，2注销 |
| u\_login\_status | varchar | 20 |  | online | 登陆状态offline/hide/online |

好友聊天记录数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS2 im\_chatlog | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| ic\_id | int |  |  | NOT NULL | 唯一标识 |
| ic\_from\_id | int |  |  | NOT NULL | 发送者id |
| ic\_to\_id | int |  |  | NOT NULL | 接受者id |
| ic\_group\_id | int |  |  |  | 群组id 若为好友聊天则为空 |
| ic\_content | text |  |  | NOT NULL | 消息内容 |
| ic\_sendTime | datetime |  |  | 自动生成 | 发送时间 |
| ic\_type | varchar | 10 |  | NOT NULL | 群聊(group)还是好友(friend) |
| ic\_status | int |  |  | 0 | 状态:好友(接收者收到1/未收到0), 群组(发出者2/接收者收到1/未收到0) |

健康资料数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS3 health\_data | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| hd\_id | int |  |  | NOT NULL | 资料id,唯一标识 |
| hd\_title | varchar | 50 |  | NOT NULL | 资料标题 |
| hd\_titleimg | varchar | 100 |  | 有默认值 | 资料标题图路径 |
| hd\_describe | text |  |  | NOT NULL | 资料描述 |
| hd\_user | int |  |  | NOT NULL | 发布者id |
| hd\_publish\_time | datetime |  |  | 自动生成 | 发布时间 |
| hd\_state | int |  |  | 1 | 状态 0不可见 1可见 2删除 |

商品信息数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS4 goods | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| goods\_id | int |  |  | NOT NULL | 商品id |
| goods\_name | varchar | 50 |  | NOT NULL | 商品名称 |
| goods\_price | double |  |  | NOT NULL | 商品价格 |
| goods\_introduce | text |  |  |  | 商品介绍 |
| goods\_img | varchar | 100 |  | 有默认值 | 图片路径 |
| goods\_sales | int |  |  | 0 | 销量 |
| goods\_publish\_time | datetime |  |  | 自动生成 | 发布时间 |
| goods\_state | int |  |  | 1 | 状态 0下架，1上架，2删除 |
| goods\_sends\_address | varchar | 255 |  | NOT NULL | 发货地址 |
| goods\_inventory | int |  |  | 0 | 库存 |
| goods\_label | varchar | 50 |  |  | 商品标签 |
| goods\_user | int |  |  | NOT NULL | 商家id |
| good\_describe\_small | varchar | 255 |  | NOT NULL | 商品简介 |

领养信息数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS5 adopt\_animal | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| adoa\_id | int |  |  | NOT NULL | 领养信息id |
| adoa\_adress | varchar | 255 |  | NOT NULL | 地址 |
| adoa\_kind | varchar | 50 |  | NOT NULL | 种类 |
| adoa\_label | varchar | 50 |  |  | 标签 |
| adoa\_title | varchar | 50 |  | NOT NULL | 标题 |
| adoa\_title\_img | varchar | 100 |  | NOT NULL | 标题图路径 |
| adoa\_describe | text |  |  |  | 信息 |
| adoa\_user | int |  |  | NOT NULL | 发布者id |
| adoa\_state | int |  |  | 默认0 | 0未被领养,1被领养,2删除 |
| adoa\_publish\_time | datetime |  |  | 自动生成 | 发布时间 |

领养动物申请表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS6 apply\_animal | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| appa\_id | int |  |  | NOT NULL | 申请id，唯一标识 |
| appa\_user\_id | int |  |  | NOT NULL | 申请者id |
| appa\_adopt\_animal | int |  |  | NOT NULL | 领养动物信息id |
| appa\_state | int |  |  | 0 | 状态 0未审核，1通过，2未通过 |
| appa\_dispose\_time | datetime |  |  |  | 处理时间 |
| appa\_apply\_time | datetime |  |  | 自动生成 | 申请时间 |

订单数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS7 orders | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| order\_id | int |  |  | NOT NULL | 订单id |
| order\_user | int |  |  | NOT NULL | 买家id |
| order\_receive\_adress | varchar | 255 |  | NOT NULL | 收货地址 |
| order\_time | datetime |  |  | 自动生成 | 下单时间 |
| order\_state | int |  |  |  | 状态0未支付，1已支付，2删除 |

宠物圈信息数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS8 pet\_circle | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| pc\_id | int |  |  | NOT NULL | 宠物圈id |
| pc\_title | varchar | 50 |  | NOT NULL | 宠物圈标题 |
| pc\_title\_img | varchar | 100 |  | NOT NULL | 宠物圈标题图片 |
| pc\_describe | text |  |  | NOT NULL | 宠物圈信息描述 |
| pc\_describe\_small | varchar | 255 |  | NOT NULL | 宠物圈信息简介 |
| pc\_publish\_time | datetime |  |  | 自动生成 | 发布时间 |
| pc\_user | int |  |  | NOT NULL | 发布者id |
| pc\_like\_num | int |  |  | 0 | 点赞数 |
| pc\_collect\_num | int |  |  | 0 | 收藏数 |
| pc\_state | int |  |  | 1 | 状态 0不可见 1可见 2删除 |

宠物圈评论表数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS9 pet\_circle\_comment | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| pcom\_id | int |  |  | NOT NULL | 宠物圈评论id |
| pcom\_pet\_circle | int |  |  | NOT NULL | 宠物圈id |
| pcom\_user | int |  |  | NOT NULL | 评论者id |
| pcom\_reply\_comment\_id | int |  |  |  | 回复评论id(为空为初始评论) |
| pcom\_state | int |  |  | 1 | 评论状态 1可见 2删除 |
| pcom\_builds | int |  |  | 0 | 评论楼数 |
| pcom\_message | varchar | 255 |  | NOT NULL | 评论内容 |
| pcom\_time | datetime |  |  | 自动生成 | 评论时间 |

商品评论表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS10 goods\_comment | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| gcom\_id | int |  |  | NOT NULL | 商品评论id |
| gcom\_goods | int |  |  | NOT NULL | 商品id |
| gcom\_user | int |  |  | NOT NULL | 评论者id |
| gcom\_reply\_comment\_id | int |  |  |  | 回复的评论id(为空为初始评论) |
| gcom\_state | int |  |  | 1 | 评论状态 1可见 2删除 |
| gcom\_message | varchar | 255 |  | NOT NULL | 评论内容 |
| gcom\_time | datetime |  |  | 自动生成 | 评论时间 |
| gcom\_level | double |  | 1~5 | 0 | 商品评级 0未评级，1~5等级 |

商家

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS11 business | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| business\_id | int |  |  | NOT NULL | 商家表唯一标识 |
| business\_user | int |  |  | NOT NULL | 商家用户id |
| business\_license\_img | varchar | 100 |  | NOT NULL | 营业执照路径 |

医生数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS12 doctor | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| doctor\_id | int |  |  | NOT NULL | 医生表唯一标识 |
| doctor\_user | int |  |  | NOT NULL | 医生用户id |
| doctor\_label | varchar | 100 |  | NOT NULL | 标签 |
| doctor\_describe | varchar | 255 |  | NOT NULL | 医生简介 |
| doctor\_img | varchar | 100 |  | NOT NULL | 医生照片 |
| doctor\_license\_img | varchar | 100 |  | NOT NULL | 执照路径 |

订单商品列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS13 order\_goods\_list | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| oglist\_id | int |  |  | NOT NULL | 订单商品列表唯一标识 |
| oglist\_order | int |  |  | NOT NULL | 订单id |
| oglist\_goods | int |  |  | NOT NULL | 商品id |
| oglist\_buynumber | int |  |  | NOT NULL | 购买数量 |

审核表数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS14 application | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| app\_id | int |  |  | NOT NULL | 审核id |
| app\_user\_id | int |  |  | NOT NULL | 发起人id |
| app\_type | varchar | 10 |  | NOT NULL | 审核类型医生,商家 |
| app\_apply\_time | datetime |  |  | 自动生成 | 申请时间 |
| app\_manager\_id | int |  |  |  | 审核者id |
| app\_check\_time | datetime |  |  |  | 审核时间 |
| app\_result | int |  |  | 0 | 审核结果 0未审核1通过 2未通过 |

群组信息数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS15 im\_group | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| ig\_id | int |  |  | NOT NULL | 群号 唯一标识 |
| ig\_groupname | varchar | 20 |  | NOT NULL | 群名字 |
| ig\_avatar | varchar | 100 |  | NOT NULL | 群头像 |
| ig\_notice | varchar | 255 |  |  | 群公告 |
| ig\_creater\_id | int |  |  | NOT NULL | 群主id/用户id |
| ig\_approval | int |  |  | 0 | 是否开启验证（0不开启/1开启） |

好友信息数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS16 im\_friend | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| if\_id | int |  |  | NOT NULL | 唯一标识 |
| if\_ifg\_id | int |  |  | NOT NULL | 用户分组id |
| if\_user\_id | int |  |  | NOT NULL | 用户id,也就是这个用户是属于这个分组的 |
| if\_nickName | varchar | 20 |  |  | 好友的昵称 |

好友分组数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS17 im\_friend\_group | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| ifg\_id | int |  |  | NOT NULL | 唯一标识 |
| ifg\_user\_id | int |  |  | NOT NULL | 所属用户id |
| ifg\_name | varchar | 20 |  | NOT NULL | 分组名称 |

群员信息数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS18 im\_group\_user | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| igu\_id | int |  |  | NOT NULL | 唯一标识 |
| igu\_group\_id | int |  |  | NOT NULL | 群组id |
| igu\_user\_id | int |  |  | NOT NULL | 用户id |
| igu\_nickName | varchar | 20 |  |  | 该用户的群昵称 |

消息盒子数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS19 im\_msgbox | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| im\_id | int |  |  | NOT NULL | 消息盒子id 唯一标识 |
| im\_user\_id | int |  |  | NOT NULL | 目标用户id |
| im\_from\_id | int |  |  |  | 来源用户id 为空则为系统消息 |
| im\_from\_group | int |  |  |  | 来源用户选择的分组 |
| im\_type | int |  |  | NOT NULL | 消息类型 |
| im\_content | varchar | 255 |  |  | 消息描述 |
| im\_remark | varchar | 255 |  |  | 附言 |
| im\_href | varchar | 255 |  |  | 未知 |
| im\_read | int |  |  | 0 | 是否已读 0未读1同意2拒绝 |
| im\_time | datetime |  |  | 自动生成 | 消息添加时间 |

用户权限表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS20 user\_authority | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| ua\_id | int |  |  | NOT NULL | 用户权限表id,唯一标识 |
| ua\_user\_id | int |  |  | NOT NULL | 用户id |
| ua\_authority | varchar | 20 |  | NOT NULL | 授权名称 用户/医生/商家/管理员 |

用户登陆记录数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS21 user\_login\_info | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| uli\_id | int |  |  | NOT NULL | 唯一标识 |
| uli\_user\_id | int |  |  | NOT NULL | 用户id |
| uli\_login\_time | datetime |  |  | 自动生成 | 登陆时间 |
| uli\_login\_ip | varchar | 25 |  | NOT NULL | 登陆ip |
| uli\_login\_address | varchar | 50 |  | NOT NULL | 登陆地点 |

宠物圈收藏表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS22 pet\_circle\_collect | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| pcc\_id | int |  |  | NOT NULL | 宠物圈收藏id唯一标识 |
| pcc\_user\_id | int |  |  | NOT NULL | 收藏者id |
| pcc\_pc\_id | int |  |  | NOT NULL | 宠物圈id |
| pcc\_time | datetime |  |  | 自动生成 | 收藏时间 |

领养动物信息收藏表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS23 adopt\_animal\_collect | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| aac\_id | int |  |  | NOT NULL | 领养动物信息收藏id唯一标识 |
| aac\_user\_id | int |  |  | NOT NULL | 收藏者id |
| aac\_adoa\_id | int |  |  | NOT NULL | 领养动物信息的id |
| aac\_time | datetime |  |  | 自动生成 | 收藏时间 |

购物车表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS24 cart | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| c\_id | int |  |  | NOT NULL | 购物车id唯一标识 |
| c\_user\_id | int |  |  | NOT NULL | 用户id |
| c\_goods\_id | int |  |  | NOT NULL | 商品id |
| c\_num | int |  |  | NOT NULL | 商品数量 |

资料收藏表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS25 health\_data\_collect | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| hdc\_id | int |  |  | NOT NULL | 资料收藏id唯一标识 |
| hdc\_data\_id | int |  |  | NOT NULL | 资料id |
| hdc\_user\_id | int |  |  | NOT NULL | 收藏者id |
| hdc\_time | datetime |  |  | 自动生成 | 收藏时间 |

激活码表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DS26 user\_active\_code | | | | | |
| 名称 | 类型 | 长度 | 域值 | 初值 | 备注 |
| uac\_id | int |  |  | NOT NULL | 激活id,唯一标识 |
| uac\_user\_id | int |  |  | NOT NULL | 用户id |
| uac\_active\_code | varchar | 40 |  | NOT NULL | 激活码 |

3、详细设计

3.1、词汇表

|  |  |
| --- | --- |
| 名称 | 含义 |
| B/S结构 | B/S结构（Browser/Server，[浏览器](https://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8/213911" \t "_blank)/[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)模式），是[WEB](https://baike.baidu.com/item/WEB" \t "_blank)兴起后的一种网络结构模式，WEB浏览器是[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF/101081" \t "_blank)最主要的[应用软件](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E8%BD%AF%E4%BB%B6/216367) |
| SpringBoot | *Spring Boot*是由Pivotal团队提供的全新框架，其设计目的是用来简化新*Spring*应用的初始搭建以及开发过程。 |
| Thymeleaf | Web 和独立环境的现代服务器端 Java 模板引擎 |
| bootstrap | *Bootstrap*是Twitter推出的一个用于前端开发的开源工具包 |
| Spring Security | Spring Security致力于为Java应用提供认证和授权管理。它是一个强大的，高度自定义的认证和访问控制框架。 |
| MyBatis | MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。 |
| MySQL | MySQL 是最流行的[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511" \t "_blank)之一， |
| Redis | Redis是一个开源的使用ANSI [C语言](https://baike.baidu.com/item/C%E8%AF%AD%E8%A8%80)编写、支持网络、可基于内存亦可持久化的日志型、Key-Value[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)，并提供多种语言的API。 |
| Tomcat | *Tomcat*服务器是一个免费的开放源代码的Web应用服务器。 |
| Druid | Druid首先是一个数据库连接池。Druid是目前最好的数据库连接池。 |
| Git | Git(读音为/gɪt/。)是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。 |
| gitHub | *gitHub*是一个面向开源及私有软件项目的托管平台，因为只支持git 作为唯一的版本库格式进行托管，故名*gitHub*。 |
| GitKraken | GitKraken是一款基于Electron构建的Git图形化工具，可以在Windows、[Mac](http://www.xue51.com/zt/macxzgjdq/)和Linux桌面系统上运行。 |
| Douker | Docker 是一个开源的应用容器引擎，基于 [Go 语言](http://www.runoob.com/go/go-tutorial.html) 并遵从Apache2.0协议开源。 |
| WebSocket | WebSocket是一种在单个[TCP](https://baike.baidu.com/item/TCP" \t "_blank)连接上进行[全双工](https://baike.baidu.com/item/%E5%85%A8%E5%8F%8C%E5%B7%A5" \t "_blank)通信的协议 |
| Layui | Layui（谐音：类UI) 是一款采用自身模块规范编写的前端 UI 框架 |
| Ajax | Ajax 即“Asynchronous Javascript And XML”（异步 JavaScript 和 XML），是指一种创建交互式[网页](https://baike.baidu.com/item/%E7%BD%91%E9%A1%B5/99347" \t "_blank)应用的网页开发技术。 |
| jQuery | jQuery是一个快速、简洁的JavaScript框架 |
| UEditor | UEditor是由[百度](https://baike.baidu.com/item/%E7%99%BE%E5%BA%A6" \t "_blank)WEB前端研发部开发的所见即所得的开源富文本编辑器，具有轻量、可定制、用户体验优秀等特点。 |
| 高德地图API |  |
| IPO图 | IPO是指结构化设计中变换型结构的输入、加工、输出。IPO图是对每个模块进行详细设计的工具，它是输入加工输出图的简称， |
| E-R图 | *E-R图*也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型 |

3.2、硬件组成.

|  |  |
| --- | --- |
| 参数名 | 参数值 |
| 云服务器平台 | 阿里云 |
| 公网IP地址 | 119.23.201.230 |
| CPU核心数 | 1核 |
| 内存 | 2GB |
| 硬盘储存空间 | 40GB |
| 每月流量限制 | 1000GB |
| 网络带宽 | 5Mbps |
| 操作系统 | CentOS7.3 |

|  |  |
| --- | --- |
| 参数名 | 参数值 |
| 云服务器平台 | 阿里云 |
| 公网IP地址 | 39.108.171.10 |
| CPU核心数 | 1核 |
| 内存 | 2GB |
| 硬盘储存空间 | 40GB |
| 每月流量限制 | 1000GB |
| 网络带宽 | 5Mbps |
| 操作系统 | CentOS7.3 |

|  |  |
| --- | --- |
| 参数名 | 参数值 |
| 云服务器平台 | 阿里云 |
| 公网IP地址 | 120.79.151.63 |
| CPU核心数 | 1核 |
| 内存 | 2GB |
| 硬盘储存空间 | 40GB |
| 每月流量限制 | 1000GB |
| 网络带宽 | 10Mbps |
| 操作系统 | CentOS7.3 |

|  |  |
| --- | --- |
| 参数名 | 参数值 |
| 云服务器平台 | 腾讯云 |
| 公网IP地址 | 203.195.224.186 |
| CPU核心数 | 1核 |
| 内存 | 2GB |
| 硬盘储存空间 | 50GB |
| 每月流量限制 | 无限制 |
| 网络带宽 | 1Mbps |
| 操作系统 | CentOS7.3 |

3.3、软件架构与开发环境

3.3.1、软件架构



3.3.2、开发环境

IDE: IntelliJ IDEA 2018.2.5 x64

JDK版本: 1.8

系统环境: win10

数据库: MySQL

3.4、系统接口表

|  |  |
| --- | --- |
| AdoptAnimalService | |
| void  insertAdoptAnimal  (AdoptAnimal adoptAnimal); | 添加一条领养动物信息 |
| AdoptAnimalExample transformAdoptAnimalToAdoptAnimalExample  (AdoptAnimal adoptAnimal); | 将对象转换为查询对象 |
| List<AdoptAnimal>  selectAdoptAnimalList  (AdoptAnimalExample adoptAnimalExample,  Integer currentPage, Integer pageSize); | 根据查询对象查询(不包含大类型) |
| List<AdoptAnimal>  selectAdoptAnimalListWithBLOBs  (AdoptAnimalExample adoptAnimalExample,  Integer currentPage,  Integer pageSize); | 根据查询对象查询(包含大类型) |
| int  countAdoptAnimal  (AdoptAnimalExample adoptAnimalExample); | 根据查询对象统计 |
| LayuiAdoptAnimal  renderAdoptAnimal  (AdoptAnimal adoptAnimal); | 渲染一条AdoptAnimal对象 |
| List<LayuiAdoptAnimal>  renderAdoptAnimals  (List<AdoptAnimal> adoptAnimals); | 渲染一组AdoptAnimal对象 |
| AdoptAnimal  getAdoptAnimalByAdoaId  (int adoaId); | 根据id获取AdoptAnimal对象 |
| boolean  delAdoptAnimalByAdoptId  (int adoaId); | 根据id逻辑删除数据 |
| boolean  updataAdoptAnimalByAdoptId  (AdoptAnimal adoptAnimal); | 根据id修改数据 |

|  |  |
| --- | --- |
| DoctorlService | |
| List<Doctor>  selectDoctorList  (Integer currentPage, Integer pageSize); | 直接查询所有数据 |
| LayuiDoctor  renderDoctor(Doctor doctor); | 渲染一条Doctor对象 |
| List<LayuiDoctor>  renderDoctors  (List<Doctor> doctors); | 渲染一组Doctor对象 |
| Doctor  getDoctorByAdoaId  (int adoaId); | 根据id获取Doctor对象 |

|  |  |
| --- | --- |
| FileService | |
| FileData  uploadFile  (MultipartFile multipartFile); | 上传文件 |

|  |  |
| --- | --- |
| HealthDataService | |
| void  insertHealthData  (HealthData healthData); | 添加一条健康资料信息 |
| HealthDataExample  transformHealthDataToHealthDataExample  (HealthData healthData); | 将对象转换为查询对象 |
| List<HealthData>  selectHealthDataList  (HealthDataExample healthDataExample,  Integer currentPage,  Integer pageSize); | 根据查询对象查询(不包含大类型) |
| List<HealthData>  selectHealthDataListWithBLOBs  (HealthDataExample healthDataExample,  Integer currentPage,  Integer pageSize); | 根据查询对象查询(包含大类型) |
| int  countHealthData  (HealthDataExample healthDataExample); | 根据查询对象统计 |
| LayuiHealthData  renderHealthData  (HealthData healthData); | 渲染一条HealthData对象 |
| List<LayuiHealthData>  renderHealthDatas  (List<HealthData> healthDatas); | 渲染一组HealthData对象 |
| HealthData  getHealthDataByAdoaId  (int hdId); | 根据id获取HealthData对象 |
| boolean  delHealthDataByAdoptId  (int hdId); | 根据id逻辑删除数据 |
| boolean  updataHealthDataByAdoptId  (HealthData healthData); | 根据id修改数据 |

|  |  |
| --- | --- |
| LayimChatlogService | |
| void  addChatLog  (String message, int status); | 插入聊天记录 |
| void  addChatLog  (ImChatlog imChatlog); | 插入聊天记录 |
| ImChatlog  transformMessageToImChatlog  (String message, int status); | 组装聊天信息 |
| LayimChatMessageResult  getChatMessage  (String message); | 组装消息结果信息 |
| List<LayimChatMessageResult>  getUnreadMessage  (int userId); | 获取所有未读信息 |
| void  updateUnreadMessageStatus  (int fromId, int toId, String type, int status); | 更新好友未读信息为已读 |
| List<ImChatlog>  getFriendHistoryChatlogs  (int userId, int id); | 获取历史消息 |
| List<ImChatlog>  getGroupHistoryChatlogs  (int id); | 获取历史消息 |
| List<LayimChatlogResult>  renderHistoryChatlogs  (List<ImChatlog> imChatlogs); | 渲染历史消息 |

|  |  |
| --- | --- |
| LayimGroupService | |
| ImGroupExample  transformImGroupToImGroupExample  (ImGroup imGroup); | 将ImGroup对象转换为ImGroupExample对象 |
| List<ImGroup>  selectImGroups  (ImGroup imGroup, Integer currentPage, Integer pageSize); | 根据模板执行查询 |
| List<ImGroup>  selectImGroups  (ImGroupExample imGroupExample,  Integer currentPage,  Integer pageSize); | 根据模板执行查询 |
| int  countImGroups  (ImGroup imGroup); | 根据模板执行统计 |
| int  countImGroups  (ImGroupExample imGroupExample); | 根据模板执行统计 |
| ImGroupUserExample  transformImGroupUserToImGroupUserExample  (ImGroupUser imGroupUser); | 将ImGroupUser对象转换为ImGroupUserExample对象 |
| List<ImGroupUser>  selectImGroupUsers  (ImGroupUser imGroupUser,  Integer currentPage,  Integer pageSize); | 根据模板执行查询 |
| List<ImGroupUser>  selectImGroupUsers  (ImGroupUserExample imGroupUserExample,  Integer currentPage,  Integer pageSize); | 根据模板执行查询 |
| int  countImGroupUsers  (ImGroupUser imGroupUser); | 根据模板执行统计 |
| int  countImGroupUsers  (ImGroupUserExample imGroupUserExample); | 根据模板执行统计 |
| List<Integer>  getGroupUserId  (int groupId); | 通过群组id查找群的群员的用户id列表 |
| List<AfUser>  getGroupUsers  (int groupId); | 通过群组id查找群的群员的用户列表 |
| List<ImGroup>  selectImGroupsByUId  (int userId); | 通过userId查找所有所属的群组列表 |
| List<ImGroup>  selectMyGroups(int userId); | 通过userId查找所有属于该用户的群组 |
| ImGroup  getImGroupById  (int groupId); | 通过群组id获取群组对象 |
| void  groupAddUser  (int userId, int groupId); | 用户加入群组 |
| Result  addGroup  (ImGroup imGroup); | 插入群组 |
| Result  updateGroup  (ImGroup imGroup); | 编辑群组信息 |
| Result  deleteGroup  (int groupId); | 解散群组 |
| Result  exitGroup  (int userId, int groupId); | 用户退出群组 |

|  |  |
| --- | --- |
| LayimMessageBoxService | |
| int  addMessageBox  (ImMsgbox imMsgbox); | 添加一条消息盒子信息  @return 返回的是状态值 1-添加 2-更新 |
| void  addSystemMessageBox  (int userId, int type, String content); | 添加一条系统消息信息 |
| void  updateMessageBox  (ImMsgbox imMsgbox); | 更新消息盒子信息 |
| ImMsgboxExample  transformImMsgboxToImMsgboxExample  (ImMsgbox imMsgbox); | 将消息盒子对象转换为查询模板对象 |
| List<ImMsgbox>  selectImMsgboxs  (ImMsgboxExample imMsgboxExample,  Integer currentPage,  Integer pageSize); | 根据查询模板对象进行查询 |
| Integer  countImMsgboxs  (ImMsgboxExample imMsgboxExample); | 根据查询模板对象进行统计 |
| void  updateImMsgboxImRead(int imMsgboxId, int status); | 更新消息盒子状态 |
| void  updateAddFriendMsgboxRead  (int userId, int fromId, int readStatus); | 更新添加好友消息盒子状态 |
| void  updateAllSystemMessageRead  (int userId); | 更新用户所有系统消息状态为已读 |
| ImMsgbox  getImMsgboxById  (int imMsgBoxId); | 根据消息盒子Id获取消息盒子消息 |

|  |  |
| --- | --- |
| LayimService | |
| LayimInitResult  getLayImInitResult  (Integer userId); | 获取LayIM初始化信息 |
| LayimChatMessageResult  getUserState  (int userId); | 获取用户登录状态 |

|  |  |
| --- | --- |
| SuperService | |
| AfUserExample  transfromAfUserToAfUserExample  (AfUser afUser); | 将用户对象转换为查询模板对象 |
| List<AfUser>  selectAfUsers  (AfUser afUser, Integer currentPage, Integer pageSize); | 根据查询模板对象查询对象 |
| List<AfUser>  selectAfUsers  (AfUserExample afUserExample,  Integer currentPage,  Integer pageSize); | 根据查询模板对象查询对象 |
| int countAfUsers(AfUser afUser); | 根据查询模板对象统计条数 |
| int countAfUsers(AfUserExample afUserExample); | 根据查询模板对象统计条数 |
| AfUser getUserByUId(int userId); | 根据userId获取用户 |
| AfUser getUserByUPhone(String uPhone); | 通过手机号查找用户 |
| AfUser getUserByUPhone(String uPhone, boolean includeDel); | 通过手机号查找用户 |
| AfUser getUserByUAccount(String account); | 通过账号查找用户 |
| AfUser getUserByUAccount(String account, boolean includeDel); | 通过账号查找用户 |
| AfUser getUserByUEmail(String email); | 通过邮箱查找用户 |
| AfUser getUserByUEmail(String email, boolean includeDel); | 通过邮箱查找用户 |
| void updateAfUser(AfUser afUser); | 更新用户信息 |
| Result updateUserStatus(Integer userId, String status); | 修改用户登录状态 |
| Result updateUserStatus(String uPhone, String status); | 修改用户登录状态 |
| void updateUserSign(int userId, String newSign); | 修改用户签名 |
| void insertAfUser(AfUser afUser); | 插入用户 |
| void deleteUserEntirely(int userId); | 完全删除用户 |

|  |  |
| --- | --- |
| UserService | |
| Result userRegister(AfUser afUser); | 用户注册 |
| Result userActive(String activeCode); | 用户激活 |
| Result insertUserLoginInfo(int userId, String IP, String address); | 插入用户登录信息 |
| UserActiveCode getUserActiveCodeByUserId(int userId); | 通过用户id查找激活码 |
| UserActiveCode  getUserActiveCodeByActiveCode  (String activeCode); | 通过激活码查找激活码对象 |

3.5、编程规范及要求

(1)所有锁定范围的数据必须使用enum来控制如status等,enum中的所有的枚举值名都是全大写的

(2)所有全局通用值必须使用常量类来控制

(3)书写代码时应该注意缩进与空格, if与小括号 左括号前 右括号与下一个字母 双斜杠注释后面都需要插入一个空格

if (num == 1) {

// .....

} else if (num == 2) {

// ....

}

(4)过长的行需要换行编码,并保持与起始行一个缩进的地方继续编码

(5)多打注释

(6)在同一个方法中不相关的代码块要用一个空行分离

/\*\*

\* 该方法用于获取SS列表与Friend列表

\*/

public stiatc Result test(HttpSession httpSession, HttpServletRequest httpServletRequest,

HttpServletResponse httpServletResponse) {

// 创建一个对象(创建对象如果没有特殊情况是不需要注释的, 这里只是要演示一下)

MyTest test = new MyTest();

//获取SS列表并计算条数

List<SS> sss = service.getSSList();

Integer count = sss.size();

//获取Friend列表并计算条数

List<Friend> friends = service.getFriendList();

Integer count2 = friends.size();

//封装结果

Result result = new Result();

result.setSssList(sss);

result.setFriendList(friends);

return result;

}

(7)当方法传递的基础类型参数可以为null的时候, 使用该基础类型的包装类, 当方法传递的基础类型参数不可为null时, 使用基础类型

(8)数据库查询方法命名规范

单个查询使用getXxx()

多个查询使用listXxx()

统计查询使用countXxx()

插入使用saveXxx()或insertXxx()

删除使用removeXxx()或deleteXxx()

更新使用deleteXxx()

(9)方法名要有意义

(10)要多打注释但是不能滥用注释

(11)git开发要刷新全局列表->如果当前版本高于develop版本则继续开发, 否则要同步自身版本到develop版本再进行下一步的开发, 开发时只更新自己的开发支线, 每次提交前都要检查是否是提交到自己的开发支线, 提交时要写清楚本次开发要点, 发生错误要报告

3.6、进度计划

2018年6月份-9月，小组已经有做好了前期的需求调研工作和最初版本的开发（未部署到服务器）。

在今年的3月2号开始，预计花费45天进行全部升级开发，然后一周的时间进行测试。了保证项目圆满完成，分阶段进行进度控制。同时加强软件质量管理，以保障项目按工期规定顺利交付。

（以下内容描述的是升级版本的开发）

|  |  |  |  |
| --- | --- | --- | --- |
| 项目阶段 | 时间 | 工作内容 | 成果 |
| 需求调研 | 2019.2.12-2019.2.22 | 线上制作需求调研表单宣传，线下在肇庆市各个地方做现场调研，并且咨询指导老师的意见。 | 系统需求规格说明书 |
| 系统设计 | 2019.3.2-2019.3.16 | 在需求调研的基础上,对系统架构，系统功能，数据库设计 | 系统设计说明书  数据库 |
| 系统开发 | 2019.3.16-2019.3.24 | 完成各个子功能的设计和单元测试 | 不同迭代版本的可运行系统 |
| 系统集成 | 2019.3.24-2019.3.26 | 完成系统的集成并且完成集成测试 | 测试报告，形成完整的可运行系统 |
| 系统初验 | 2019.3.26-2019.3.28 | 项目初验 | 初验报告 |
| 系统试运行 | 2019.3.28-2019.4.4 | 1：平台持续运行  2：系统优化 | 升级版本的可运行系统，并且部署到服务器 |
| 项目终验 | 2019.4.5- | 项目终验 | 验收报告 |

4、其它

4.1、特色与优势.

4.1.1、软硬结合

相比于其它web开发，我们功能之一“宠物去哪”是软硬件结合的产物，它由“定位项圈”硬件提供精准的动物位置，再传送到前端页面及时显示数据。

4.1.2、受众范围广

一个成功的产品背后肯定有一群感兴趣的人群，而“动物之家”的起始目标人群是那些“动物爱好者”，“宠物饲主”，“动物保护者”等，不管在哪个年龄阶段都存在着大量的目标人群。

因此，我们产品起步的受众人群广，易于后续的发展。

4.1.3、“刚需品”

对于宠物饲主来说，毫无章法地搜索资料，看着鱼龙混杂的数据完全无法很好地处理宠物问题，因此一个具有权威性的资料库，一个能够提供在线咨询的平台已成为“刚需品”。

并且如何处理好城市流浪动物和待遗弃动物问题，更是体现本产品的“刚需”。

4.1.4、可持续发展

我们提供了一个主打宠物产品商城，有效地保证了资金来源。

4.2、未来与发展

（1）我们将会与各地“动物爱好者协会”，“流浪动物保护组织”等机构合作，结合我们项目“带我回家”功能，积极做到让每一只被遗弃的动物，都有一个温暖的家。

（2）我们将会找相关厂家合作，批量生产“动物定位项圈“，将该产品作为我们项目的特色收入；并且欢迎各大厂家入驻，以售卖宠物用品为主要收入。

（3）我们将会聘请专业的宠物医生，他们不仅可以提供“在线咨询”服务，而且可以提供具有权威性的宠物文章，以完善我们“健康资料”的数据库。