宠爱有+应用程序设计规格说明

Ver:1.0

**目 录**

[1 引言 3](#_Toc301301027)

[1.1 目标 3](#_Toc301301028)

[1.2 文档范围 3](#_Toc301301029)

[1.3 术语和缩略语 3](#_Toc301301030)

[1.4 参考资料 3](#_Toc301301031)

[1.5 系统目标和约束 4](#_Toc301301032)

[2 系统设计 4](#_Toc301301033)

[2.1 系统架构概述 4](#_Toc301301034)

[2.2 对象模型 4](#_Toc301301035)

[2.3 接口](#_Toc301301036) 5

[2.4 特性实现](#_Toc301301037) 7

[2.5 错误代码](#_Toc301301038) 11

[2.6 错误日志](#_Toc301301039) 12

[2.7 部署视图](#_Toc301301040) 13

[3 数据库设计](#_Toc301301041) 14

[3.1 逻辑模型](#_Toc301301042) 14

[3.2 物理模型](#_Toc301301043) 14

[4 质量及其他方面](#_Toc301301044) 14

[4.1 可维护性](#_Toc301301045) 14

[4.2 安全性](#_Toc301301046) 15

[4.3 可扩展性](#_Toc301301047) 15

[4.4 可靠性](#_Toc301301048) 15

[4.5 可用性](#_Toc301301049) 15

[4.6 性能设计](#_Toc301301050) 15

[5 附录](#_Toc301301051) 16

[5.1 附件](#_Toc301301052) 16

[5.2 修过记录](#_Toc301301053) 16

# 引言

## 1.1目标

在完成软件开发前期的准备工作如项目需求等，结合《需求确认书》，项目小组提出了这份软件设计说明书。

此概要设计说明书对宠爱有+应用程序的功能分配，模块划分，程序的总体结构，输入输出和接口设计，运行设计，数据结构设计及出错设计等方面作了全面的概括性的说明，为软件详细设计奠定了基础，同时作为软件开发人员的阶段性总结和程序员进行开发及未来测试的重要文档资料。

## 1.2文档范围

本文档包括宠爱有+应用程序的架构设计、数据库设计、安全性、可靠性、可用性等方面的规划和设计。

## 1.3术语和缩略语

术语：

|  |  |
| --- | --- |
| 开发方 | XX |
| 用户 | 使用该APP的人群 |

## 1.4参考资料

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 文档名称 | 作者 | 日期 |
| 1 | 产品愿景和商业机会 | 张依梦 | 2020.3.13 |
| 2 | 用户分析 | 李炳文 | 2020.3.13 |
| 3 | 技术分析 | 王叶颖 | 2020.3.13 |
| 4 | 资源需求估计 | 马素隶 | 2020.3.13 |
| 5 | 风险分析 | 魏欣欣 | 2020.3.13 |

## 1.5系统目标和约束

系统目标：项目需要完成用户注册、商家注入、收养、寄养、与同样爱护小动物的人交流等

系统的约束：因为时间有限，该程序只实现最基本的功能。

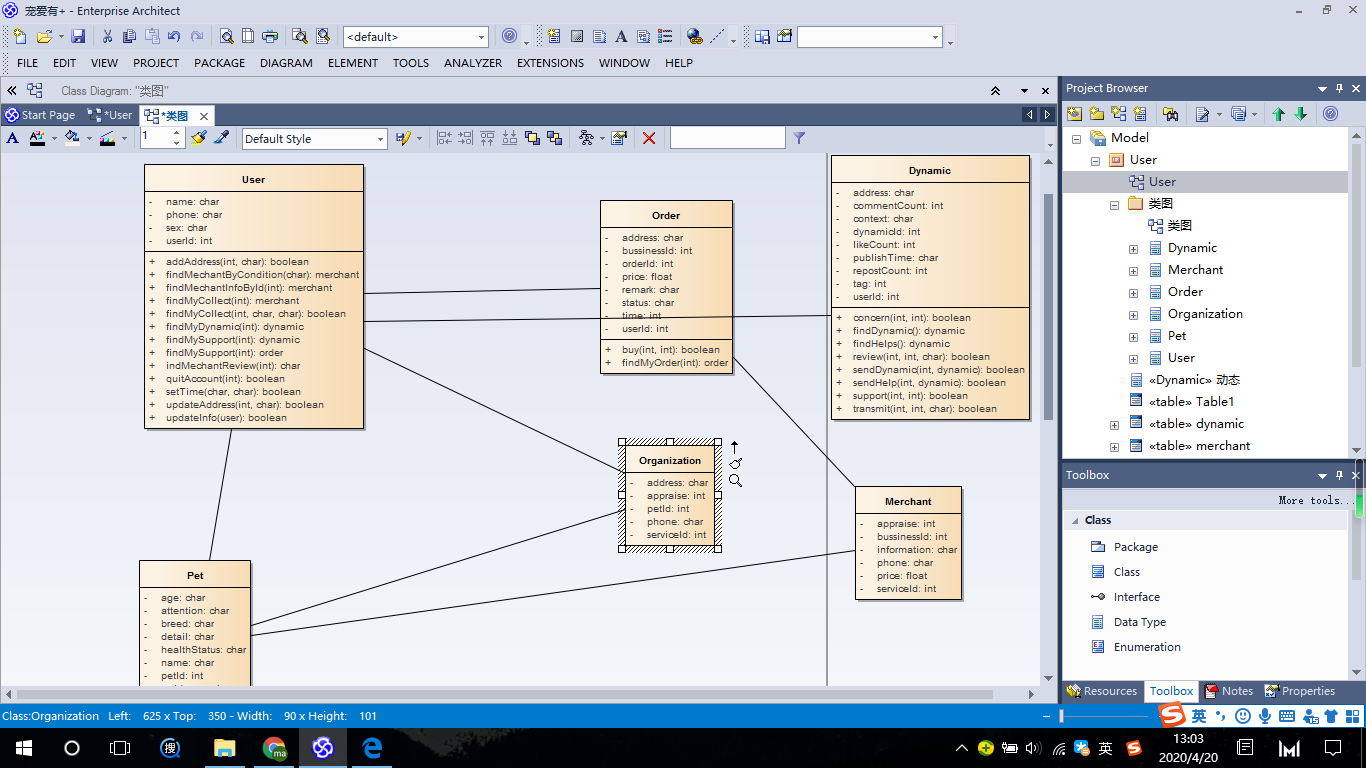
# 系统设计

## 2.1系统架构概述

该程序的架构采用MVC架构的模式：模型层（model）、视图层（view）、控制器层（controller）。采用面向对象的设计方法。

## 2.2对象模型

该系统采用面向对象的设计方法。



## 2.3接口

*[详细说明本系统内部每个接口的每个方法的定义。*

*注意：如果本系统需要和其他系统交互，则应该将交互的接口协议单独成册，而不是写入本节中。]*

com.adopt

1. 搜索 List<Pet> search(String key)
2. 筛选 List<Pet> choose(List<String> condition)
3. 查看宠物详情 Pet findPetInfoById(int id)
4. 进入收养机构 Organization findOrganizationInfoById(int id)
5. 建立客服对话 Boolean dialogue(int id)
6. 收藏 Boolean collect(int id)
7. 下单 Boolean buy(int userId,int petId)

com.dynamic

1. 查看全部动态 List<Dynamic> findDynamic()
2. 转发 Boolean transmit(int userId,int dynamicId,String content)
3. 评论 Boolean review(int userId,int dynamicId,String content)
4. 点赞 Boolean support(int userId,int dynamicId)
5. 关注 Boolean concern(int userId,int otherId)
6. 查看全部求助 List<Help> findHelps()
7. 发表动态 void sendDynamic(int userId,List<Dynamic> d)
8. 发表求助 void sendHelp(int userId,List<Help> help)
9. 定位城市 String country()
10. 查看评论 List<Review> findReview(int dynamicId)

com.center

1. 编辑资料 Boolean updateInfo(User user)
2. 查看自己发表的全部动态 List<Dynamic> findMyDynamic(int userId)
3. 查看自己的全部点赞 List<Dynamic> findMySupport(int userId)
4. 查看全部订单 List<Order> findMyOrder(int userId)
5. 查看我的收藏 List<Mechant> findMyCollect(int userId)
6. 进入夜间模式 void setMode()
7. 意见反馈 Boolean sentAdvice(int uesrId,String advice,String email)
8. 退出账号 Boolean quitAccount(int uesrId)

com.agency

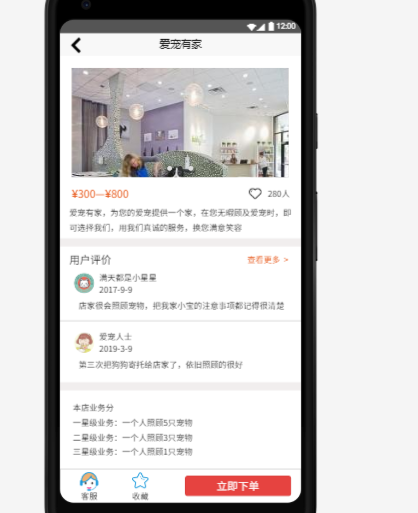
1. 筛选店铺 List<Mechant> findMechantByCondition(List<String> conditions)
2. 设置寄养时间 void setTime(String start,String end)
3. 查看店铺详情 Mechant findMechantInfoById(int id)
4. 查看店铺评论 List<String> findMechantReview(int mechantId)
5. 修改地址 Boolean updateAddress(int userId,Stirng address)
6. 添加地址 Boolean addAddress(int userId,Stirng address)

## 4 特性实现

1. 展示宠物店详细信息

a. 添加宠物店的详细信息

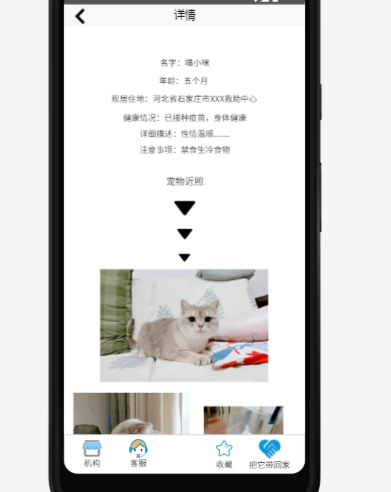
b. 查看自己所搜索的宠物店的详细信息



1. 用户查看收养宠物详情

a. 查看特定宠物

b. 查看特定的宠物类型的所有收养所的该宠物类型的所有宠物



1. 寄养

a. 用户选定宠物店下单



b.宠物店老板收到订单信息并进行处理

1. 收养

a. 用户选定宠物收养

b. 收养所收到订单信息进行处理

1. 查看求助信息

a. 点击主页中间动态

b. 点击右侧求助查看求助信息



1. 宠物店老板查看寄养宠物订单情况
2. 寄养日期到，用户取回宠物

a. 用户取回宠物，宠物店老板进行相应地修改订单状态操作

b. 用户选择性地进行评价

## 2.5 错误代码

*[详细列出错误处理机制所能处理的所有错误代码及其含义]*

# Java.lang.ArrayIndexOutOfBoundsException 数组下标越界，加一个if语句判断内容是否为空即可



没有找到sessionFactory，sessionFactory没有注入

## 2.6 错误日志

*[说明能够被处理和记录的错误类型和记录方式。]*

My sql

SZDB:/var/lib/mysql # tail SZDB.err

140906 22:06:45 InnoDB: Completed initialization of buffer pool

140906 22:06:45 InnoDB: highest supported file format is Barracuda.

140906 22:06:45  InnoDB: Waiting for the background threads to start

140906 22:06:46 InnoDB: 5.5.37 started; log sequence number 1605345

140906 22:06:47 [Note] Server hostname (bind-address): '0.0.0.0'; port: 3306

140906 22:06:47 [Note]   - '0.0.0.0' resolves to '0.0.0.0';

140906 22:06:47 [Note] Server socket created on IP: '0.0.0.0'.

140906 22:06:47 [Note] Event Scheduler: Loaded 0 events

140906 22:06:47 [Note] /usr/sbin/mysqld: ready for connections.

Version: '5.5.37-log'  socket: '/var/lib/mysql/mysql.sock'  port: 3306  MySQL Community Server (GPL)

Java



## 2.7 部署视图

1. 硬件环境：

a. 需要互联网

b. 至少需要一台服务器

1. 软件：

a. 需要mysql数据库。或者其他操作系统。

b. 需要Apache8作为WEB服务器。

c. 需要安装Android Studio软件

1. 配置：

a. 下载mysql驱动

b. 下载并安装mysql实现工具（如Navicat、MySQL-Front）

c. 下载并安装Android studio软件

1. 部署：

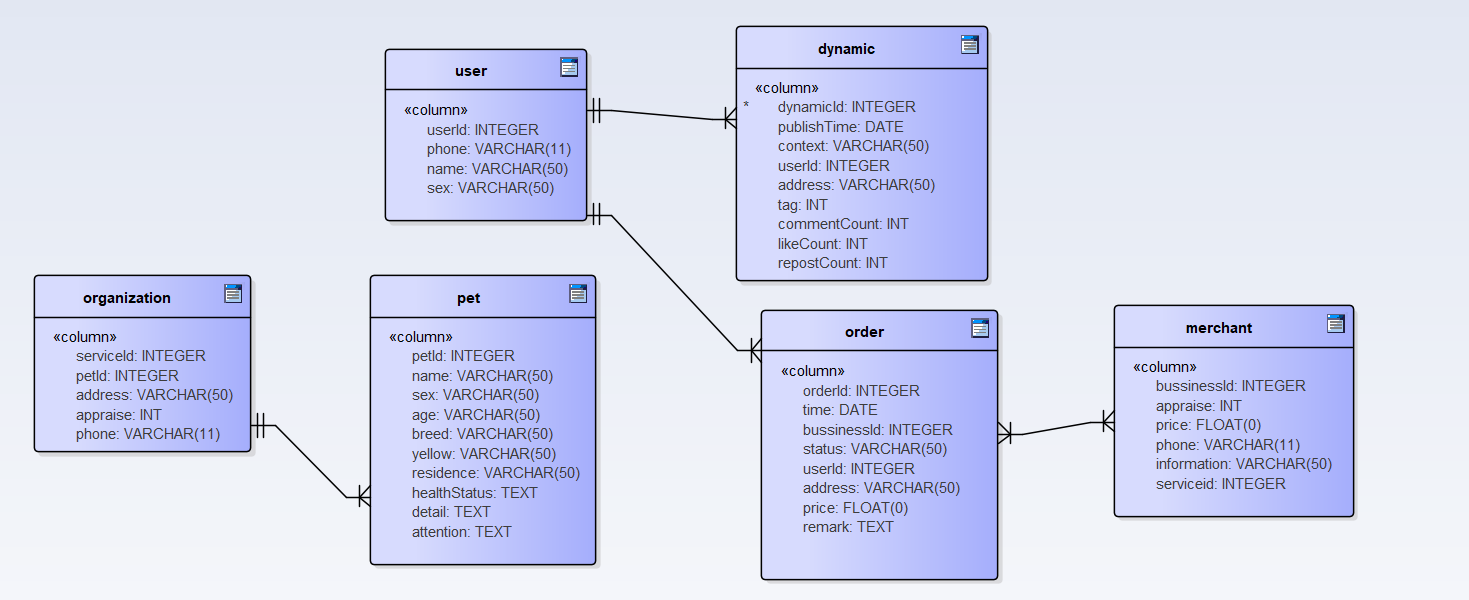
a. 配置spring.hibernate,关闭错误提示。

b. 连接My SQL的文件，修改IP地址、数据库的用户名和密码。

c. 将程序包放在Apache安装目录中得htdocs目录下。

# 数据库设计

## 3.1逻辑模型



## 3.2 物理模型

1. 为每个表初始设置300M的存储空间，以10%的大小扩展。
2. 为上传的图片预留空间。

# 质量及其他方面

## 4.1可维护性

1. 通过程序注释等方式增加代码的可读性和可维护性。
2. 编写易于修改的代码，即将代码模块化，将数据访问层分离，做成一个个函数，由其它层调用，以增加代码的可维护性。
3. 通过编写一段简短的的易于测试的代码来降低修复错误的成本。

4、数据库有日志记录，系统一旦出现故障有恢复到故障之前的信息和数据的能力

## 4.2安全性

1. 密码使用md5加密
2. 对用户的输入进行验证
3. 对用户的输入的特殊字符进行转义，防止sql注入攻击
4. 支持MAM（移动应用程序管理）和MDM（移动设备管理），以减少设备和应用程序相关的威胁
5. 加入一个API来加强移动应用程序的安全性。

## 4.3可扩展性

1. 可以通过增加硬件资源的方式提高系统的响应速度。
2. 可以通过修改代码，扩充系统的功能。
3. 使用分布式队列降低耦合性，减少模块之间的调用关系，从而实现新增和删除都对其他部分没有影响。

## 4.4 可靠性

1. 数据库按时备份，有日志记录
2. 软件的设计开发阶段，要进过充分讨论和评审，确保架构和编码的严谨。
3. 充分进行质量测试和性能测试。

## 4.5 可用性

1. 做好充分的前期工作，系统一旦投入使用，尽量减少宕机的次数和时间
2. 注意用户的反馈，并且主动征求用户意见。

## 4.6 性能设计

该软件中涉及到的输入、输出数据均在数据库存储着，数据类型和数据精度等详细信息可参见如上ER图和数据字典。

该软件的一些具体时间特性要求如下：

响应时间 <= 3000ms

更新处理时间 <= 2000ms

数据传送时间 <= 1000ms

数据计算时间 <= 20000ms

灵活性要求应用于Android手机客户端，方便、易携带

# 附录

## 5.1 附件

宠爱有+.eap

## 5.2 修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改人 | 修改时间 | 修改内容 | 核准 |
| 1.0 | 魏欣欣 | 2020-4-20 | 系统设计 |  |
| 2.0 | 王叶颖 | 2020-4-20 | 数据库设计 |  |