宠乐屋设计规格说明

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 3](#_Toc301301032)

[2 系统设计 4](#_Toc301301033)

[2.1 系统架构概述 4](#_Toc301301034)

[2.2 对象模型 5](#_Toc301301035)

[2.3 接口 5](#_Toc301301036)

[2.4 特性实现 6](#_Toc301301037)

[2.5 错误代码 12](#_Toc301301038)

[2.6 错误日志 15](#_Toc301301039)

[2.7 部署视图 15](#_Toc301301040)

[3 数据库设计 17](#_Toc301301041)

[3.1 逻辑模型 17](#_Toc301301042)

[3.2 物理模型 17](#_Toc301301043)

[4 质量及其他方面 18](#_Toc301301044)

[4.1 可维护性 18](#_Toc301301045)

[4.2 安全性 18](#_Toc301301046)

[4.3 可扩展性 18](#_Toc301301047)

[4.4 可靠性 18](#_Toc301301048)

[4.5 可用性 18](#_Toc301301049)

[4.6 性能设计 19](#_Toc301301050)

[5 附录 19](#_Toc301301051)

[5.1 附件 19](#_Toc301301052)

[5.2 修过记录 19](#_Toc301301053)

# 引言

## 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乐屋系统的功能分配，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，同时作为系统分析员工作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。

## 文档范围

本文档包括系统的架构设计、数据库设计、安全性、可靠性、可用性等方面的规划和设计。

## 术语和缩略语

PF – 宠乐屋

## 系统目标和约束

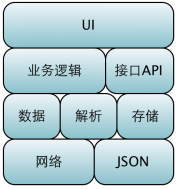
系统目标：项目需要完成宠物运动情况，宠物信息编辑，健康诊断，宠物用品购买，宠物购买，商品智能推荐，宠物社区，购物车等功能

系统的约束：因为时间有限，该系统只实现最基本的功能。

# 系统设计

## 系统架构概述

宠乐屋客户端系统按照逻辑划分，主要分为四层，基础类库层，数据层，业务逻辑层，UI展示层。每个层次由不同的模块组成。



基础类库层

包含的内容： 网络请求， JSON数据的解析

网络请求：网络请求采用Apache HttpClient 进行网络请求的支持。对于使用Volley类库的方式，就可以定义多种Request来完成Volley方式的请求。采用HttpClient可以更加灵活的定义各种网络接口请求，同时Android系统本身就包含HttpClient的实现。

JSON解析：JSON解析需要针对服务器返回的数据格式进行处理，只要能够将JSON解析成为对象，并且能够更新到UI界面即可。经过分析，接口返回的数据无法适应FastJSON/GSON的常用解析方式，因此最终确定 JSON 解析采用 org.json来进行。

数据层

数据层架构的六个统一，即统一数据规划，统一存储，统一计算，统一服务，统一接入，统一治理。

物理层：通过万兆链接核心交换区，实现网络高速交换，确保可靠性；各服务器均双线链接数据核心交换机，消除单点故障；结构清晰，层次分明。数据层要遵守整体性原则；标准化原则；安全与效率并重原则；系统功能与职责分工相适应原则；一致性原则；

业务逻辑与接口层

将在下文2.3中详细叙述

## 对象模型

Control 部门管理类

商家审核

商品审核

宠物信息审核

用户管理

后台数据库信息插入管理

后台数据库信息删除管理

后台数据库信息修改管理

Pet 宠物管理类

宠物信息查看

宠物信息修改

宠物身体状况查询

宠物个性化推荐

Business 商城管理类

用户信息查询

用户购买与付款

商家管理商品与店铺

个性化推荐商品

商城热门商品排行

## 接口

前端与后台数据库接口：基于flask框架设计交互的接口

数据接口：数据传输以及安全性保障

推荐系统与商城系统接口：热门商品和个性化推荐接口

## 特性实现

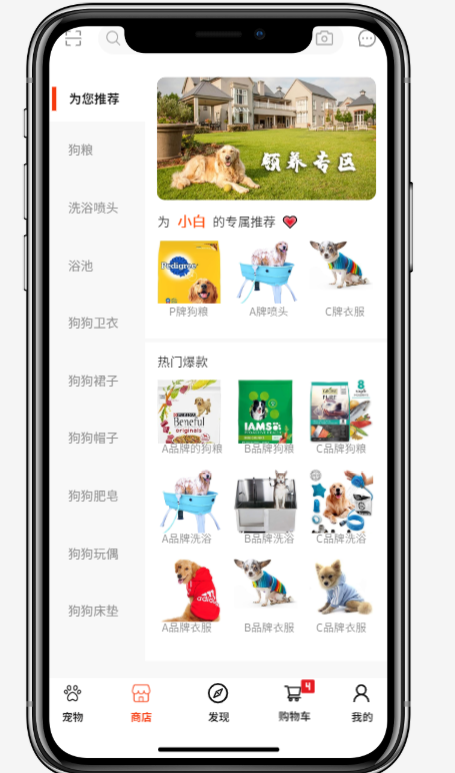
1. 用户登录与注册
   1. 用户注册页面
   2. 注册成功，可选择进入客户端主页面



1. 宠物信息页面
   1. 运动情况页面
   2. 宠物详细信息
   3. 健康诊断



1. 商店页面
   1. 搜索商品
   2. 查看推荐信息
   3. 商品分类
   4. 拍照上传

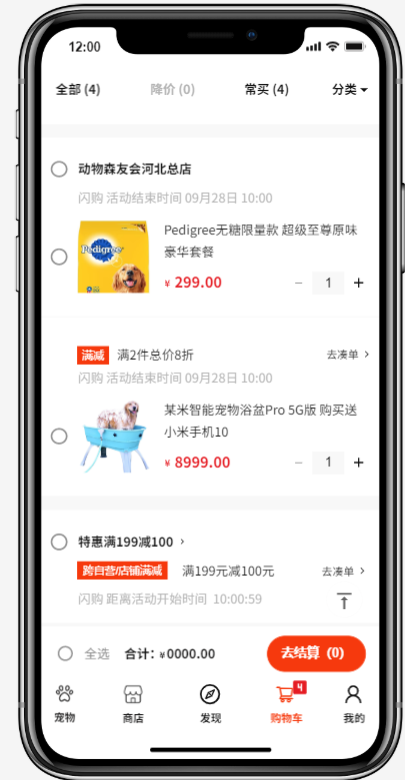


1. 发现页面

游客查看宠物周边信息



1. 购物车页面
   1. 全部商品
   2. 降价题型
   3. 结算账单



1. 我的信息
   1. 客户信息
   2. 商品管理
   3. 钱包管理
   4. 我的设备



## 错误代码

400 Bad Request：1、语义有误，当前请求无法被服务器理解。除非进行修改，否则客户端不应该重复提交这个请求。2、请求参数有误。

401 Unauthorized：当前请求需要用户验证。该响应必须包含一个适用于被请求资源的 WWW-Authenticate 信息头用以询问用户信息。客户端可以重复提交一个包含恰当的 Authorization 头信息的请求。如果当前请求已经包含了 Authorization 证书，那么401响应代表着服务器验证已经拒绝了那些证书。如果401响应包含了与前一个响应相同的身份验证询问，且浏览器已经至少尝试了一次验证，那么浏览器应当向用户展示响应中包含的实体信息，因为这个实体信息中可能包含了相关诊断信息。参见RFC 2617。

402 Payment Required：该状态码是为了将来可能的需求而预留的。

403 Forbidden：服务器已经理解请求，但是拒绝执行它。与401响应不同的是，身份验证并不能提供任何帮助，而且这个请求也不应该被重复提交。如果这不是一个 HEAD 请求，而且服务器希望能够讲清楚为何请求不能被执行，那么就应该在实体内描述拒绝的原因。当然服务器也可以返回一个404响应，假如它不希望让客户端获得任何信息。

404 Not Found：请求失败，请求所希望得到的资源未被在服务器上发现。没有信息能够告诉用户这个状况到底是暂时的还是永久的。假如服务器知道情况的话，应当使用410状态码来告知旧资源因为某些内部的配置机制问题，已经永久的不可用，而且没有任何可以跳转的地址。404这个状态码被广泛应用于当服务器不想揭示到底为何请求被拒绝或者没有其他适合的响应可用的情况下。出现这个错误的最有可能的原因是服务器端没有这个页面。

405 Method Not Allowed：请求行中指定的请求方法不能被用于请求相应的资源。该响应必须返回一个Allow 头信息用以表示出当前资源能够接受的请求方法的列表。

鉴于 PUT，DELETE 方法会对服务器上的资源进行写操作，因而绝大部分的网页服务器都不支持或者在默认配置下不允许上述请求方法，对于此类请求均会返回405错误。

406 Not Acceptable：请求的资源的内容特性无法满足请求头中的条件，因而无法生成响应实体。除非这是一个 HEAD 请求，否则该响应就应当返回一个包含可以让用户或者浏览器从中选择最合适的实体特性以及地址列表的实体。实体的格式由 Content-Type 头中定义的媒体类型决定。浏览器可以根据格式及自身能力自行作出最佳选择。但是，规范中并没有定义任何作出此类自动选择的标准。

407 Proxy Authentication Required:与401响应类似，只不过客户端必须在代理服务器上进行身份验证。代理服务器必须返回一个 Proxy-Authenticate 用以进行身份询问。客户端可以返回一个 Proxy-Authorization 信息头用以验证。参见RFC 2617。

408 Request Timeout:请求超时。客户端没有在服务器预备等待的时间内完成一个请求的发送。客户端可以随时再次提交这一请求而无需进行任何更改。

409 Conflict:由于和被请求的资源的当前状态之间存在冲突，请求无法完成。这个代码只允许用在这样的情况下才能被使用：用户被认为能够解决冲突，并且会重新提交新的请求。该响应应当包含足够的信息以便用户发现冲突的源头。冲突通常发生于对 PUT 请求的处理中。例如，在采用版本检查的环境下，某次 PUT 提交的对特定资源的修改请求所附带的版本信息与之前的某个（第三方）请求向冲突，那么此时服务器就应该返回一个409错误，告知用户请求无法完成。此时，响应实体中很可能会包含两个冲突版本之间的差异比较，以便用户重新提交归并以后的新版本。

410 Gone:被请求的资源在服务器上已经不再可用，而且没有任何已知的转发地址。这样的状况应当被认为是永久性的。如果可能，拥有链接编辑功能的客户端应当在获得用户许可后删除所有指向这个地址的引用。如果服务器不知道或者无法确定这个状况是否是永久的，那么就应该使用404状态码。除非额外说明，否则这个响应是可缓存的。

411 Length Required:服务器拒绝在没有定义 Content-Length 头的情况下接受请求。在添加了表明请求消息体长度的有效 Content-Length 头之后，客户端可以再次提交该请求。

412 Precondition Failed:服务器在验证在请求的头字段中给出先决条件时，没能满足其中的一个或多个。这个状态码允许客户端在获取资源时在请求的元信息（请求头字段数据）中设置先决条件，以此避免该请求方法被应用到其希望的内容以外的资源上。

413 Request Entity Too Large:服务器拒绝处理当前请求，因为该请求提交的实体数据大小超过了服务器愿意或者能够处理的范围。此种情况下，服务器可以关闭连接以免客户端继续发送此请求。

414 Request-URI Too Long:请求的URI 长度超过了服务器能够解释的长度，因此服务器拒绝对该请求提供服务。这比较少见，通常的情况包括：

本应使用POST方法的表单提交变成了GET方法，导致查询字符串（Query String）过长。

重定向URI “黑洞”，例如每次重定向把旧的 URI 作为新的 URI 的一部分，导致在若干次重定向后 URI 超长。

客户端正在尝试利用某些服务器中存在的安全漏洞攻击服务器。这类服务器使用固定长度的缓冲读取或操作请求的 URI，当 GET 后的参数超过某个数值后，可能会产生缓冲区溢出，导致任意代码被执行[1]。没有此类漏洞的服务器，应当返回414状态码。

415 Unsupported Media Type:对于当前请求的方法和所请求的资源，请求中提交的实体并不是服务器中所支持的格式，因此请求被拒绝。

416 Requested Range Not Satisfiable:如果请求中包含了 Range 请求头，并且 Range 中指定的任何数据范围都与当前资源的可用范围不重合，同时请求中又没有定义 If-Range 请求头，那么服务器就应当返回416状态码。

417 Expectation Failed:在请求头 Expect 中指定的预期内容无法被服务器满足，或者这个服务器是一个代理服务器，它有明显的证据证明在当前路由的下一个节点上，Expect 的内容无法被满足。

500 Internal Server Error:服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器端的源代码出现错误时出现。

501 Not Implemented:服务器不支持当前请求所需要的某个功能。当服务器无法识别请求的方法，并且无法支持其对任何资源的请求。

502 Bad Gateway：作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应。

503 Service Unavailable：由于临时的服务器维护或者过载，服务器当前无法处理请求。这个状况是临时的，并且将在一段时间以后恢复。如果能够预计延迟时间，那么响应中可以包含一个 Retry-After 头用以标明这个延迟时间。如果没有给出这个 Retry-After 信息，那么客户端应当以处理500响应的方式处理它。

504 Gateway Timeout：作为网关或者代理工作的服务器尝试执行请求时，未能及时从上游服务器（URI标识出的服务器，例如HTTP、FTP、LDAP）或者辅助服务器（例如DNS）收到响应。

505 HTTP Version Not Supported：服务器不支持，或者拒绝支持在请求中使用的 HTTP 版本。这暗示着服务器不能或不愿使用与客户端相同的版本。响应中应当包含一个描述了为何版本不被支持以及服务器支持哪些协议的实体。

507 Insufficient Storage：服务器无法存储完成请求所必须的内容。

509 Bandwidth Limit Exceeded：服务器达到带宽限制。

510 Not Extended：获取资源所需要的策略并没有被满足。（RFC 2774）

## 错误日志

**1、位置和内容**

错误日志无论在格式上还是在内容上都和访问日志不同。然而，错误日志和访问日志一样也提供丰富的信息，我们可以利用这些信息分析服务器的运行情况、哪里出现了问题。

错误日志的文件名字是error\_log，但如果是Win平台，则错误日志的文件名字是error.log。错误日志的位置可以通过ErrorLog指令设置：

ErrorLog logs/error.log

除非文件位置用“/”开头，否则这个文件位置是相对于ServerRoot目录的相对路径。如果阿帕奇采用默认安装方式安装，那么错误日志的位置应该在/usr/local/apache/logs下。但是，如果阿帕奇用某种包管理器安装，错误日志很可能在其他位置。

正如其名字所示，错误日志记录了服务器运行期间遇到的各种错误，以及一些普通的诊断信息，比如服务器何时启动、何时关闭等。

我们可以设置日志文件记录信息级别的高低，控制日志文件记录信息的数量和类型。这是通过LogLevel指令设置的，该指令默认设置的级别是error，即记录称得上错误的事件。

大多数情况下，我们在日志文件中见到的内容分属两类：文档错误和CGI错误。但是，错误日志中偶尔也会出现配置错误，另外还有前面提到的服务器启动和关闭信息。[1]

**2、文档错误**

文档错误和服务器应答中的400系列代码相对应，最常见的就是404错误——Document Not Found（文档没有找到）。除了404错误以外，用户身份验证错误也是一种常见的错误。[1]

**3、CGI错误**

错误日志最主要的用途或许是诊断行为异常的CGI程序。为了进一步分析和处理方便，CGI程序输出到STDERR（Standard Error，标准错误设备）的所有内容都将直接进入错误日志。这意味着，任何编写良好的CGI程序，如果出现了问题，错误日志就会告诉我们有关问题的详细信息。[1]

**4、查看日志文件**

下面看看如何方便地查看服务器日志文件。用telnet连接到服务器，然后输入下面的命令：

tail -f /usr/local/apache/logs/error\_log

## 部署视图

1. 硬件环境：
   1. 需要互联网
   2. 至少需要一台服务器
2. 软件：
   1. 需要Windows Server 2003或其他Server版的操作系统以安装SQL SERVER2005企业版。或者其他操作系统，但是使用SQL SERVER就不能使用企业版了，可以换成EXPRESS版。
   2. 需要Apache2.2作为WEB服务器。
   3. 需要安装PHP解释器
3. 配置：
   1. 因为PHP5.3以上的版本已经不支持mssql扩展，所以需要自行下载微软提供的The SQL Server Driver for PHP(SQLSRV20.EXE)
   2. 下载后解压缩，将所有的.dll文件拷贝到php安装目录的ext文件夹中，然后打开php.ini，添加以下语句打开php\_sqlsrv和php\_pdo\_sqlsrv扩展：

[PHP\_PDO\_SQLSRV]

extension=php\_pdo\_sqlsrv\_53\_ts\_vc6.dll

[PHP\_SQLSRV]

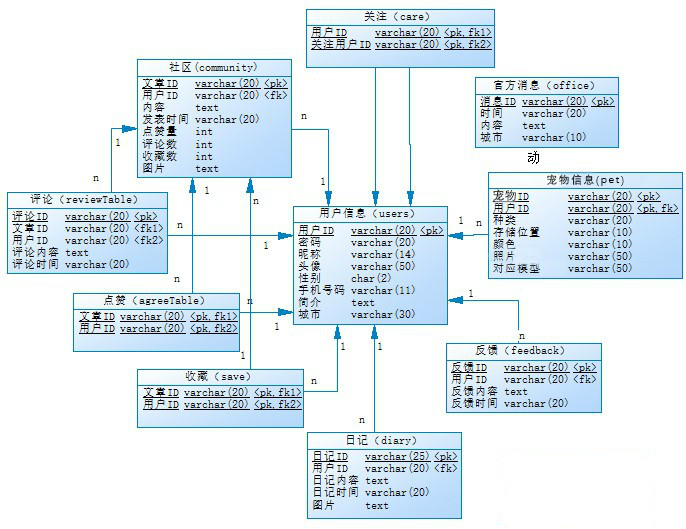
extension=php\_sqlsrv\_53\_ts\_vc6.dll

* 1. 安装Microsoft SQL Server 2008 R2 Native Client，可以去http://msdn.microsoft.com/en-us/library/cc296170(SQL.90).aspx下载安装，因为微软的这个扩展包需要它的支持。（注SQL SERVER 2005的客户端也不行，必须要2008的）

1. 部署：
   1. 配置PHP.ini,关闭错误提示。
   2. 配置PHP连接MS SQL的文件，修改IP地址、数据库的用户名和密码。
   3. 将程序包放在Apache安装目录中得htdocs目录下。

# 数据库设计

## 逻辑模型



## 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

## 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. 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. 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

## 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

## 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。

## 可靠性

1. 数据库按时备份，有日志记录

## 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

## 性能设计

1、准确度98%以上，确保信息准确，准确无误

2. 宠物运动中的GPS导航要求精确实时，和手环数据同步

3、实时性：能实现实时性要求，满足用户需要

# 附录

## 附件

墨刀原型网址：

<https://org.modao.cc/app/125408fa7254a3d5fd1a8c714c9de7fe2a8ed549?simulator_type=device&sticky>

## 修过记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改人 | 修改时间 | 修改内容 | 核准 |
| 1.0 | 申领东 | 2020-04-19 |  |  |
| 1.0 | 王盼 | 2020-04-19 |  |  |
| 1.0 | 盛蕾 | 2020-04-19 |  |  |