北京信息科技大学

**毕业设计（论文）**

**题 目：**基于iOS的宠物社交互动社区系统

的设计与实现

**学 院： 计算机学院**

**专 业：** 软件工程

**学生姓名：** **1501/2015011206**

**指导老师/督导老师：** 郝保水

**起止时间：**200 年 月 日 **至** 200 年 月 日

**目 录**

[摘要 5](#_Toc9172926)

[中文 5](#_Toc9172927)

[英文 5](#_Toc9172928)

[1引言 5](#_Toc9172929)

[1.1 课题背景及意义 5](#_Toc9172930)

[1.2 相关研究现状 5](#_Toc9172931)

[1.3 工作内容概述 5](#_Toc9172932)

[2PIGPEN需求分析 5](#_Toc9172933)

[2.1 系统功能需求 5](#_Toc9172934)

[2.2 业务流图和数据流图 6](#_Toc9172935)

[2.2.1 业务流图 6](#_Toc9172936)

[2.2.2 数据流图 7](#_Toc9172937)

[2.3 数据字典 9](#_Toc9172938)

[2.3.1 数据流分量 9](#_Toc9172939)

[2.3.2 数据流 9](#_Toc9172940)

[2.3.3 数据存储 9](#_Toc9172941)

[2.3.4 数据处理 9](#_Toc9172942)

[3PIGPEN系统设计 9](#_Toc9172943)

[3.1 总体设计 9](#_Toc9172944)

[3.2 客户端设计 9](#_Toc9172945)

[3.3 服务端设计 9](#_Toc9172946)

[3.4 数据模型设计 9](#_Toc9172947)

[3.4.1 模型概念设计（E-R 图） 9](#_Toc9172948)

[3.4.2 模型逻辑设计（关系模型） 9](#_Toc9172949)

[3.4.3 数据表 9](#_Toc9172950)

[3.5 系统开发与运行环境 9](#_Toc9172951)

[3.5.1 开发工具 9](#_Toc9172952)

[3.5.2 运行环境 9](#_Toc9172953)

[3.5.3 系统实现的关键技术 10](#_Toc9172954)

[4PIGPEN中间件模块 10](#_Toc9172955)

[4.1 Token鉴权 10](#_Toc9172956)

[4.2 HTTP请求方法和参数 10](#_Toc9172957)

[5PIGPEN用户管理模块 10](#_Toc9172958)

[5.1 用户管理模块结构设计 10](#_Toc9172959)

[5.1.1 用户管理模块功能简介 10](#_Toc9172960)

[5.1.2 用户管理模块功能图 10](#_Toc9172961)

[5.2 用户管理模块详细设计与实现 10](#_Toc9172962)

[5.2.1 页面设计 10](#_Toc9172963)

[5.2.2 用户登录 10](#_Toc9172964)

[5.2.3 注册用户 10](#_Toc9172965)

[5.2.4 搜索用户 10](#_Toc9172966)

[5.2.5 更新用户信息 10](#_Toc9172967)

[5.2.6 分享名片 10](#_Toc9172968)

[6PIGPEN即时通讯模块 10](#_Toc9172969)

[6.1 即时通讯模块结构设计 11](#_Toc9172970)

[6.1.1 即使通讯模块功能简介 11](#_Toc9172971)

[6.1.2 即使通讯模块功能图 11](#_Toc9172972)

[6.2 即时通讯模块详细设计与实现 11](#_Toc9172973)

[6.2.1 页面设计 11](#_Toc9172974)

[6.2.2 初始化及通讯密钥生成 11](#_Toc9172975)

[6.2.3 建立连接 11](#_Toc9172976)

[6.2.4 发送与接收消息 11](#_Toc9172977)

[6.2.5 获取会话列表 11](#_Toc9172978)

[6.2.6 获取消息未读数 11](#_Toc9172979)

[7宠物管理模块 11](#_Toc9172980)

[7.1 宠物管理模块结构设计 11](#_Toc9172981)

[7.1.1 宠物管理模块功能简介 11](#_Toc9172982)

[7.1.2 宠物管理模块功能图 11](#_Toc9172983)

[7.2 宠物管理模块详细设计与实现 11](#_Toc9172984)

[7.2.1 页面设计 11](#_Toc9172985)

[7.2.2 创建宠物 11](#_Toc9172986)

[7.2.3 搜索宠物 12](#_Toc9172987)

[7.2.4 关注宠物 12](#_Toc9172988)

[7.2.5 拉黑宠物 12](#_Toc9172989)

[7.2.6 更新宠物信息 12](#_Toc9172990)

[7.2.7 吃饭 12](#_Toc9172991)

[7.2.8 喝水 12](#_Toc9172992)

[7.2.9 遛狗 12](#_Toc9172993)

[7.2.10 撸猫 12](#_Toc9172994)

[8PIGPEN寻找附近猫狗模块 12](#_Toc9172995)

[8.1 寻找附近猫狗模块结构设计 12](#_Toc9172996)

[8.1.1 寻找附近猫狗模块功能简介 12](#_Toc9172997)

[8.1.2 寻找附近猫狗模块功能图 12](#_Toc9172998)

[8.2 寻找附近猫狗模块详细设计与实现 12](#_Toc9172999)

[8.2.1 页面设计 12](#_Toc9173000)

[8.2.2 处理用户GPS信息 12](#_Toc9173001)

[8.2.3 拉取附近猫狗位置信息 12](#_Toc9173002)

[9PIGPEN评分模块 13](#_Toc9173003)

[9.1 评分模块结构设计 13](#_Toc9173004)

[9.1.1 评分模块功能简介 13](#_Toc9173005)

[9.1.2 评分模块功能图 13](#_Toc9173006)

[9.2 评分模块详细设计与实现 13](#_Toc9173007)

[9.2.1 页面设计 13](#_Toc9173008)

[9.2.2 评分维度 13](#_Toc9173009)

[9.2.3 每日评分 13](#_Toc9173010)

[9.2.4 每周评分 13](#_Toc9173011)

[9.2.5 每月评分 13](#_Toc9173012)

[9.2.6 总评分 13](#_Toc9173013)

[9.2.7 评分重制 13](#_Toc9173014)

[10PIGPEN朋友圈模块 13](#_Toc9173015)

[10.1 朋友圈模块结构设计 13](#_Toc9173016)

[10.1.1 朋友圈模块功能简介 13](#_Toc9173017)

[10.1.2 朋友圈模块功能图 13](#_Toc9173018)

[10.2 朋友圈模块详细设计与实现 14](#_Toc9173019)

[10.2.1 文章创建 14](#_Toc9173020)

[10.2.2 文章删除 14](#_Toc9173021)

[10.2.3 评论创建 14](#_Toc9173022)

[10.2.4 评论删除 14](#_Toc9173023)

[10.2.5 图文混排引擎 14](#_Toc9173024)

[10.2.6 信息流 14](#_Toc9173025)

[10.2.7 点赞统计 14](#_Toc9173026)

[10.2.8 阅读统计 14](#_Toc9173027)

[10.2.9 缓存优化 14](#_Toc9173028)

[11PIGPEN系统评价 14](#_Toc9173029)

[11.1 特色与优点 14](#_Toc9173030)

[11.2 不足与改进 14](#_Toc9173031)

[11.2.1 系统存在的不足 14](#_Toc9173032)

[11.2.2 系统改进方案 14](#_Toc9173033)

[总结 14](#_Toc9173034)

[参考文献 15](#_Toc9173035)

# 摘要

## 中文

## 英文

# 引言

## 课题背景及意义

## 相关研究现状

## 工作内容概述

# PIGPEN需求分析

## 系统功能需求

宠物行业高速发展从侧面映射出了国内的一些社会现象，例如空巢老人和单身青年群体的持续增长等，从宠物主人的角度出发，大家从“饲养宠物”这件事情本身目前正在向“宠物陪伴”这一环节迈进，不再是之前单纯的养一只宠物而已，而是通过养宠物这一件事进行社交，找到适合自己加入的一些社群，通过养宠物这一件事进行锻炼，通过饲养宠物来培养自己发现生活中的美好事物，进而达到锻炼自己的感性心理状态。

目前宠物行业中主要的服务对象有：

1. 宠物

全国各地目前都出现了宠物服务店，例如提供宠物洗澡、宠物发型定制和售卖相关宠物用品的职能。

1. 宠物主人

当前宠物行业对宠物主人的服务非常有限，甚至没有。只能在一些如北京上海等稍大城市中见到寥寥几家宠物服务商会定期举办宠物主人社交活动。

从以上两点来看，可以得出当前宠物行业几乎都是围绕着宠物本身去服务，缺失了宠物主人这一环。而决定了“饲养宠物”这件事本身是否有意义、是否值得花费大量心思等问题的答案最终却是由宠物主人来决定的，故缺失了对宠物主人这一环节的考虑会降低“饲养宠物”的体验。

已经有一部分宠物服务提供商正在慢慢转为宠物公社。宠物公社在宠物行业的传统定义中不仅仅只是“宠物服务”提供商这一简单的角色，而是逐渐的承担起了连接宠物和其它宠物、宠物和宠物主人、宠物和其它宠物主人以及宠物主人和其它宠物主人这四大关系纽带的责任。在这四种关系纽带的建立过程中，宠物和宠物主人这一传统关系纽带已被市场挖掘得差不多，几乎已经很难找到差异化存在的地方。而其它三种关系纽带目前各大宠物服务提供商对其的认识都不一样，最终导致产出的服务产品种类和体验参差不齐。

PIGPEN在保证宠物和宠物主人这一传统关系的前提下，基于宠物和其它宠物、宠物和其它宠物主人和宠物主人和其它宠物主人这三种关系的理解上挖掘出核心产品逻辑，通过移动互联网的特有属性，结合大数据、人工智能等领域技术来进行宠物公社的建立和完善，帮助改善宠物和宠物主人这两个实体的关系。

## 业务流图和数据流图

### 业务流图

对业务详细调查的重点是对管理业务的流程进行描述，采用系统流程图中的部分图形工具来进行描述管理的活动，进行规范化的说明。PIGPEN的业务流程图符号说明如图1所示：



图 1业务流程图图例

对业务的详细调查之后，为开发本系统而绘制出的宠物社交互动社区的管理业务流程图如图2所示：



图 2PIGPEN宠物社交互动社区业务流程图

### 数据流图

数据流图对系统中数据运动进行了抽象。基于前期的调研，从整个系统的科学性、合理性以及数据的实际运动可行性展开。将系统对数据的处理能力以及数据实体和实体之间的联系通过自顶向下、逐层分解的方法，精确地从逻辑上描述了整个系统应该具备数据的加工、输入、输出、存储以及来源和去向（流向外部实体）等部分。对数据进行流程分析主要是分析了对数据的信息流动流动、信息传递、信息处理、信息存储等内容的分析。数据的流程分析实际目的是要尽可能早的发现并解决数据流向过程中的问题。分析系统现存的数据流程是目前大部分的做法是基于多层数据流程图（英文全称：data flow diagram，简称DFD）来实现的。实现多层数据流程图的做法是：按照系统的实际业务流程图整理出一套完整的业务流程实际业务处理顺序，接着将对应调查的过程中所掌握数据的处理过程，最终绘制成一套符合系统的整个数据流程图。

娱乐圈“遛狗/撸猫”模块的数据流图如图3所示：

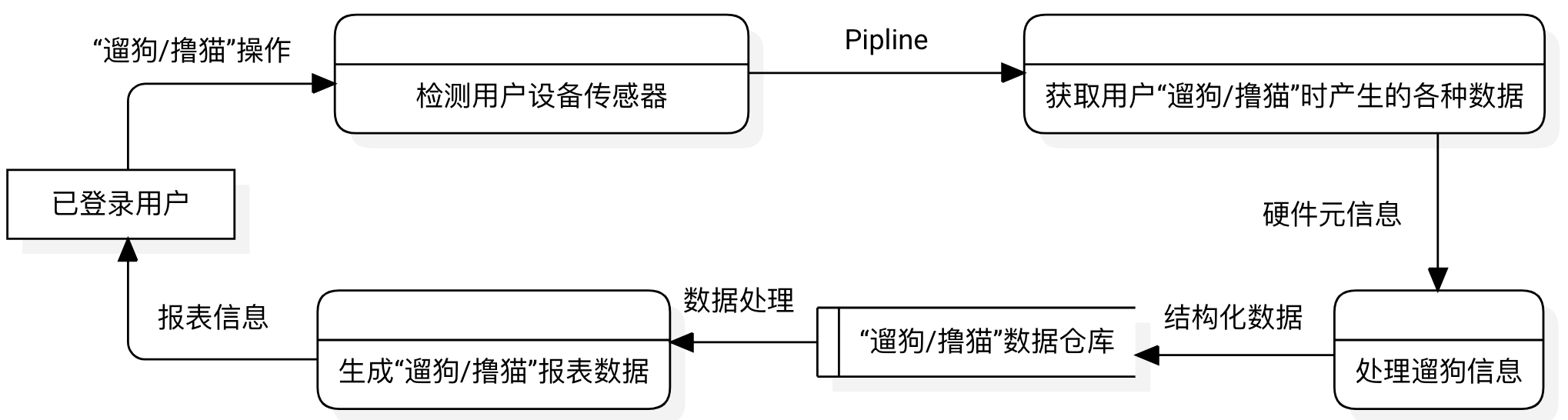


图 3娱乐圈“遛狗”模块的数据流图

朋友圈信息流模块的数据流图如图4所示：

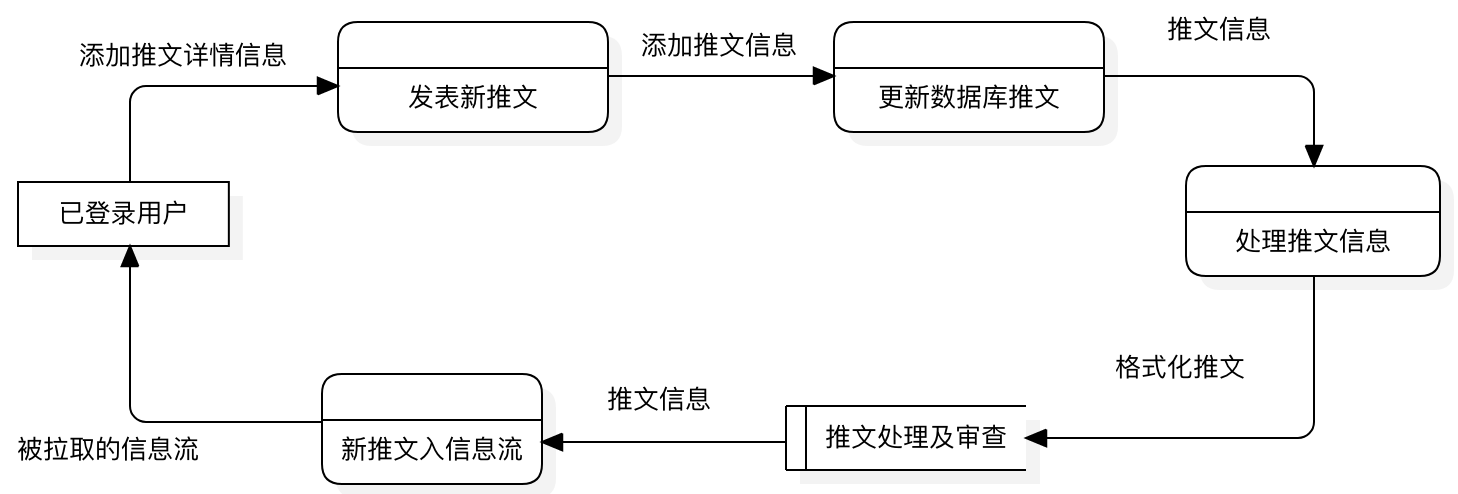


图 4朋友圈信息流数据流图

## 数据字典

数据字典是整个系统的数据信息集合，也即对数据流图中所有元素定义的集合。整个系统的逻辑模型由数据流图和数据字典二者构成，缺失了数据字典的数据流图就是一个不严格的数据流图，同样，缺失了数据流数据字典也难于发挥其真正的作用。只有将正确的数据流图中每个元素精确的定义放在一起，才能对这二者所描述的系统构成一个严格且准确的说明。

PIGPEN宠物社交互动社区的数据字典：

### 数据流分量

1.名字：用户信息

别名：无

描述：管理员用户和普通用户必须先在本系统进行登录后才能进入系统操作各个模块，这样在一定程度上保障了系统的安全性。

定义：用户信息=用户ID+用户名称+密码+手机号

位置：用来验证用户身份，分配用户权限

2.名字：密码

别名：无

描述：密码用于验证用户的身份和权限

定义：密码={MD5加密后字符串}

位置：用户表

3.名字：朋友圈信息流推文表

别名：无

描述：用户均可在朋友圈中发布新推文

定义：推文=推文ID+推文主体

位置：供用户发布新推文时使用

4.名字：发现附近的猫狗表

别名：无

描述：当创建了猫/狗的用户登录系统时，分享其GPS信息

定义：附近的猫狗=宠物ID+宠物最近出现GPS信息+宠物最近出现时间

位置：供用户查看附近的猫狗时对比查找一定距离范围的宠物

5.名字：宠物表

别名：无

描述：用户根据规则对照自己的真实宠物创建出自己的宠物

定义：宠物=宠物ID+宠物昵称+宠物生日+宠物体重+宠物进食量+宠物饮水量+宠物运动量+宠物所属用户ID

位置：供用户创建宠物时使用

6.名字：遛狗撸猫表

别名：无

描述：记录用户在遛狗和撸猫时尝产生的数据

定义：遛狗撸猫表单条数据=宠物ID +宠物所属用户ID+嬉戏类型+嬉戏开始时间+嬉戏主体修饰内容

位置：供用户进行遛狗和撸猫行为操作时使用

7.名字：评分表

别名：无

描述：根据用户每天在系统中产生的数据进行基于一定规则的评分。产出日评分、周评分、月评分和总评分

定义：评分=评分ID+用户ID+评分

位置：对用户产生的数据进行维度上的转换分析时使用

### 数据流

数据流的编号: DS01-01

数据流的名称: 朋友圈推文信息、用户、宠物管理等各种信息

简述: 对信息数据做必要的录入后得到的数据库表

数据流的来源: 录入

数据流的去向: 数据库

数据流的组成: 各类信息

数据流的流量: 10个/月

高峰期流量: 1000个/月

### 数据存储

数据存储的编号: F01-01

数据存储的名称: 数据库信息

简述: 存放的推文、宠物、用户等信息

数据存储的组成: 各类信息

关键字: 编号

相关联的处理: P1(对对应的数据表进行录入)

P2(对对应的数据表进行查询)

P3(对对应的数据表进行修改删除)

P4(对对应的数据表进行统计)

### 数据处理

1. 处理逻辑编号: P01-01

处理逻辑名称: 信息数据的录入

简述: 对各种信息的数据进行录入

输入的数据流: 已登录用户

处理过程: 对信息数据进行分类录入

输出的数据流: 各类的信息数据表

处理频率:2000个/月

1. 处理逻辑编号: P01-02

处理逻辑名称: 查询各类信息

简述: 根据条件查询所需的信息.

输入的数据流:信息来源于数据库

处理过程: 输入查询条件查询,得到符合条件的信息

输出的数据流: 查询得到的信息

处理频率:10个/分钟

1. 处理逻辑编号: P01-03

处理逻辑名称: 修改、删除各类信息数据

简述: 对数据做需要的处理修改后存入数据库中.

输入的数据流:数据库信息数据

处理过程: 对需要修改的信息数据做处理

输出的数据流: 修改或删除后得到的信息

处理频率: 10个/分钟

1. 处理逻辑编号: P01-04  
   处理逻辑名称: 统计评分数据信息

简述: 对信息做需要的评分统计后存入数据库中.

输入的数据流:宠物行为产生的数据

处理过程: 对需要统计评分的数据做统计

输出的数据流: 统计后得到的评分数据

处理频率: 全局/天

# PIGPEN系统设计

## 总体设计

通过对宠物社交分享社区平台系统的需求分析并结合实际情况分析，本系统是由宠物管理、用户管理、朋友圈信息流管理、娱乐圈模块、寻找附近猫狗模块、即时通讯和评分系统七个主模块构成。本系统的功能结构图如下

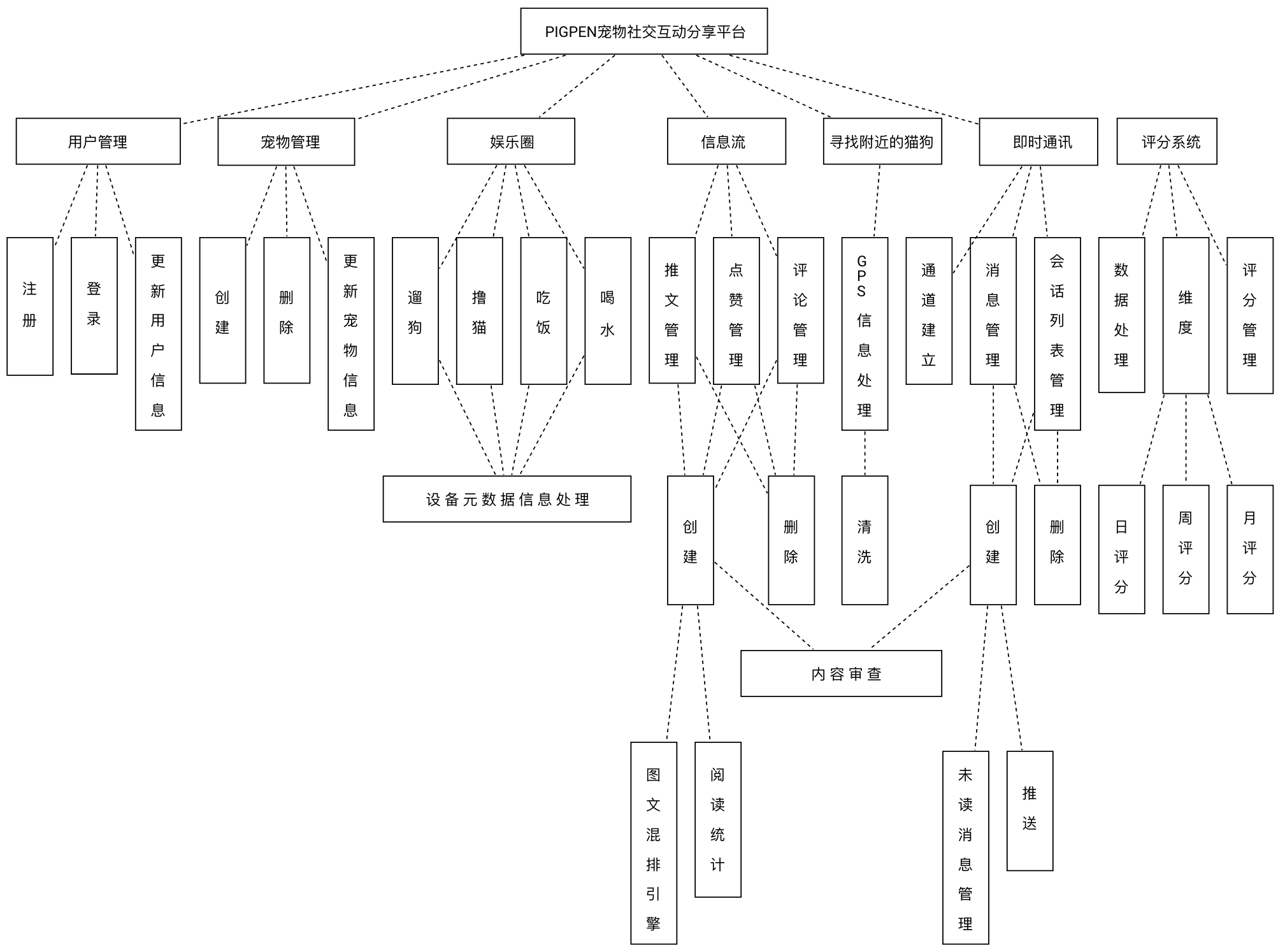


图 5PIGPEN 系统功能结构图

系统功能简介如下：

用户管理：提供基本用户管理功能，如用户的创建、登录、修改信息等。

宠物管理：提供宠物的创建、删除和更新宠物信息。

娱乐圈：对宠物实体抽象出了四个基本日常活动，如遛狗、撸猫、吃饭及喝水。遛狗和撸猫需要借助用户设备传感器进行活动，吃饭及喝水需要用户手动录入宠物当前吃饭和喝水的数据信息。

信息流：信息流基于PIGPEN的图文排版引擎使得用户可以随意搭配其推文格式。创建推文时需要将推文数据导流至内容审查中间件进行内容审查。其它用户可在朋友圈中对推文进行点赞和评论。

寻找附近的猫狗：需要结合其它用户上一次进入系统登录时携带的GPS信息数据，映射到地图中进行展示。

即时通讯：对其它用户发起一个聊天，需要先进行消息通道的建立，消息通道建立后双方产生聊天数据会经由APNS消息推送通道推送至对方设备中。聊天双方产生的消息数据可进行删除和导出操作。

评分：评分系统分为日评分、周评分和月评分三个维度进行，用户在未来180天内可进行一次评分的归零。最终产生的评分将作为衡量用户对宠物饲养是否符合PIGPEN标准的指标。

## 客户端设计

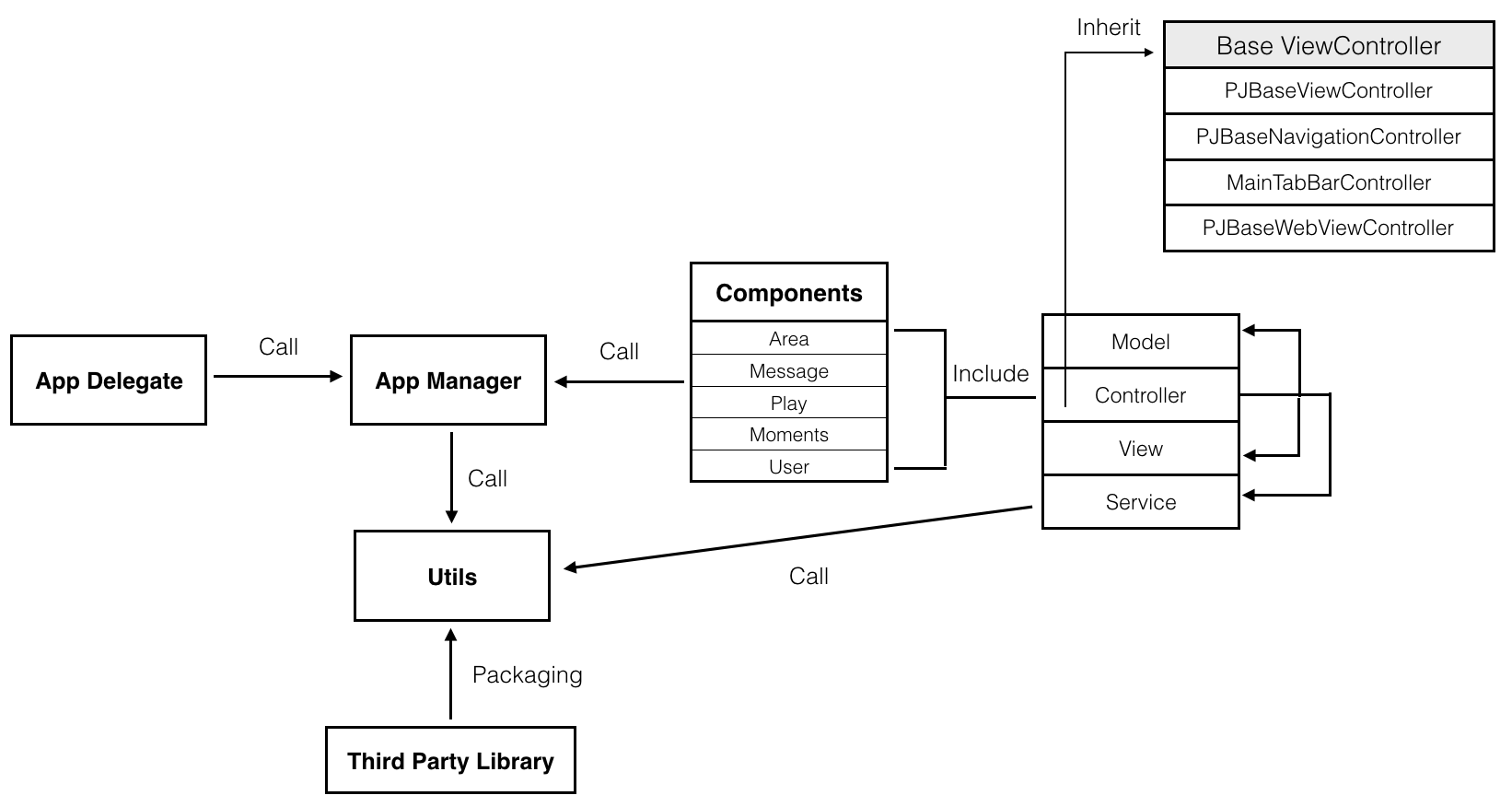


图 6PIGPEN 客户端设计

## 服务端设计

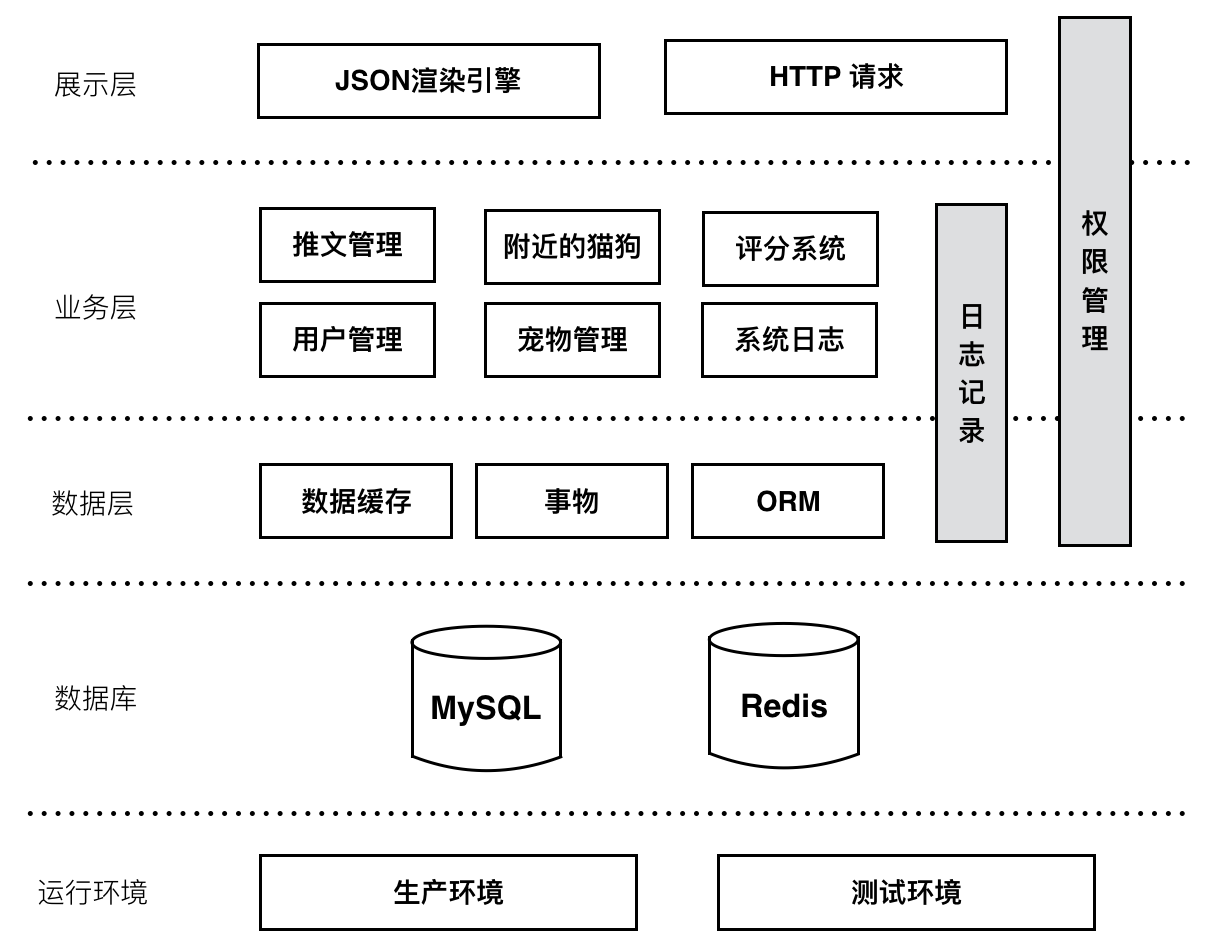


图 7PIGPEN服务端设计

## 数据模型设计

### 模型概念设计（E-R 图）

目前大部分的数据库系统基本上都是某种数据模型的实现。数据库的概念结构设计本质上是围绕着建立某种数据模型。在众多的数据模型中使用较为广泛的是概念模型，该模型可从用户角度来对数据进行建模，大多数用于数据库的设计。

概念模型的建立方法有很多，使用最广泛的是实体—联系图（Entity—Relationship Approach）。使用E-R图描述实际情况中的概念模型，又称为E-R模型。其提供了对数据实体、实体属性及描述二者关系的方法。数据实体使用矩形表示，矩形内为数据视图的标识符。属性用椭圆形表示,使用无向边将属性与对应的数据实体进行连接。关系使用菱形表示,菱形内为关系标识符,同样使用无向边联系数据实体和实体属性,并可在该无向边上标明关系的类型。

使用E-R图进行概念模型构建，先进行局部ER图的设计，接着把完成的各个局部ER图连接成一个全局E-R图，最后讲全局ER图进行完善和优化。

1. 设计局部E-R模式

实体和属性的定义：

用户（用户编号，手机号，昵称，密码，类型，性别，头像，创建时间，数据更新时间）

宠物（宠物编号，昵称，性别，类型，体重，绝育情况，感情状态，生日，品种，每日进食量，运动量，创建时间，数据更新时间）

遛狗（宠物编号，起点，终点，长度，是否完成，创建时间，数据更改时间）

撸猫（宠物编号，结束时间，是否完成，创建时间，数据更改时间）

吃饭（宠物编号，喂食量，消耗量，创建时间，预计消耗完成时间，数据更改时间）

喝水（宠物编号，喂水量，消耗量，创建时间，预计消耗完成时间，数据更改时间）

推文（推文编号，用户编号，推文内容，删除标志，发布时间，数据更新时间）

评论（评论编号，用户编号，评论内容，类型，发布时间，数据更新时间）

各个实体的E-R图如下所示：

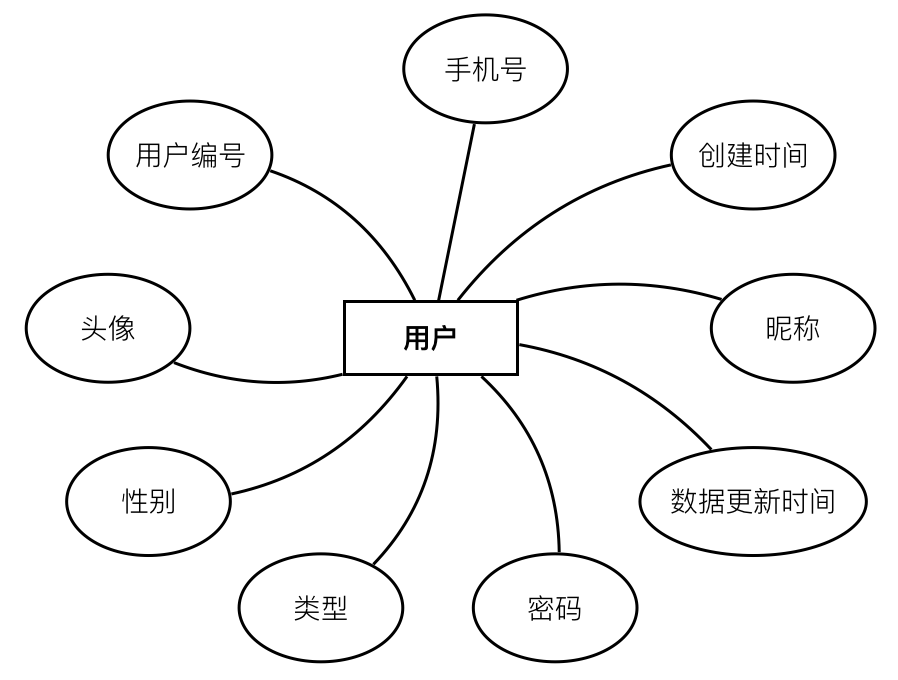


图 8用户 E-R图

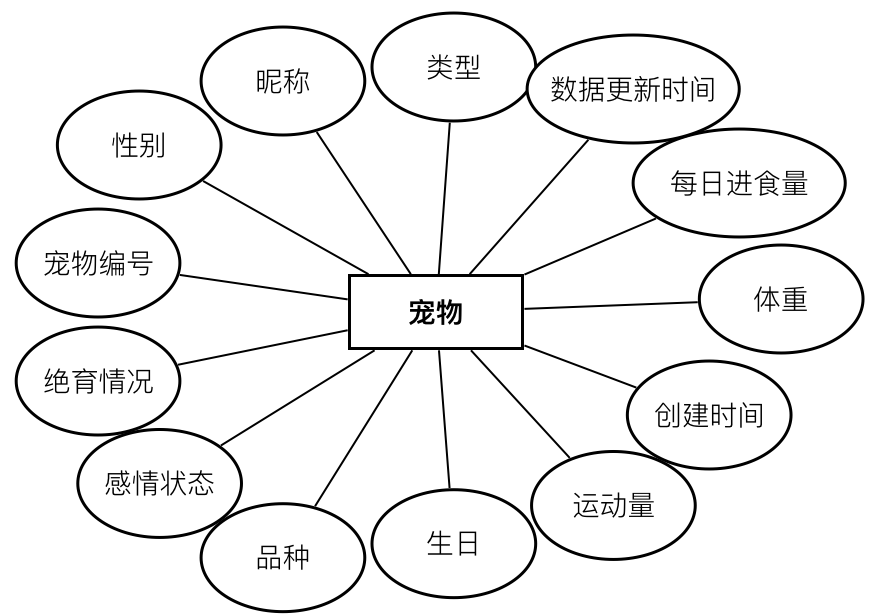


图 9宠物E-R图

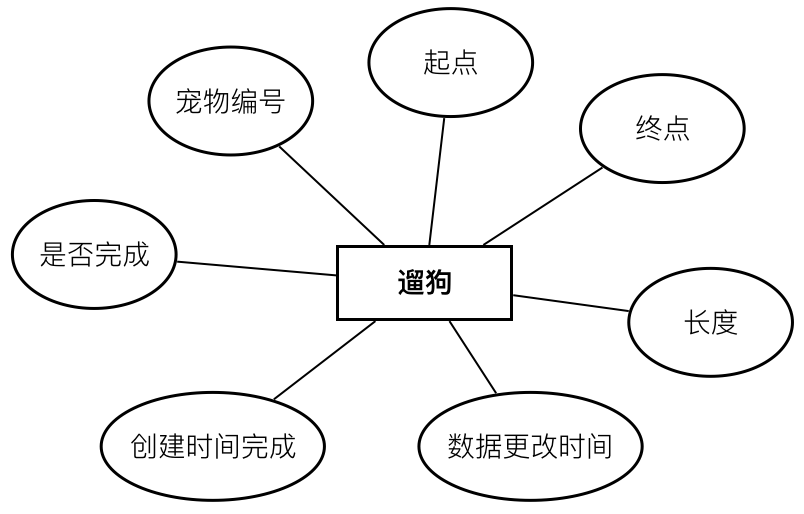


图 10遛狗E-R图

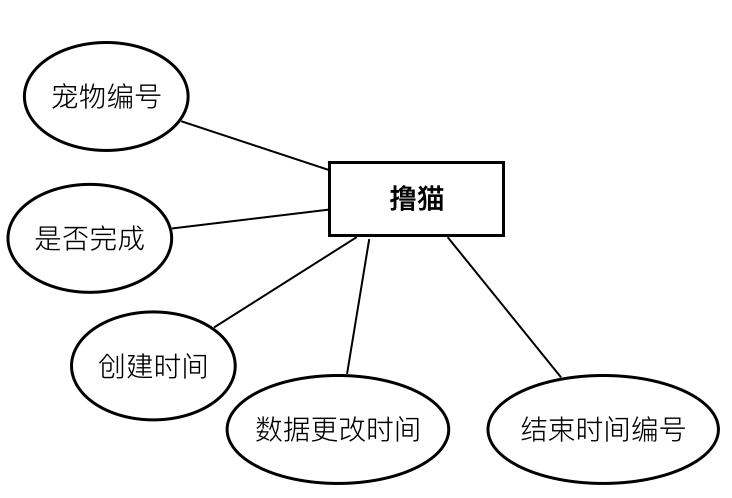


图 11撸猫E-R图

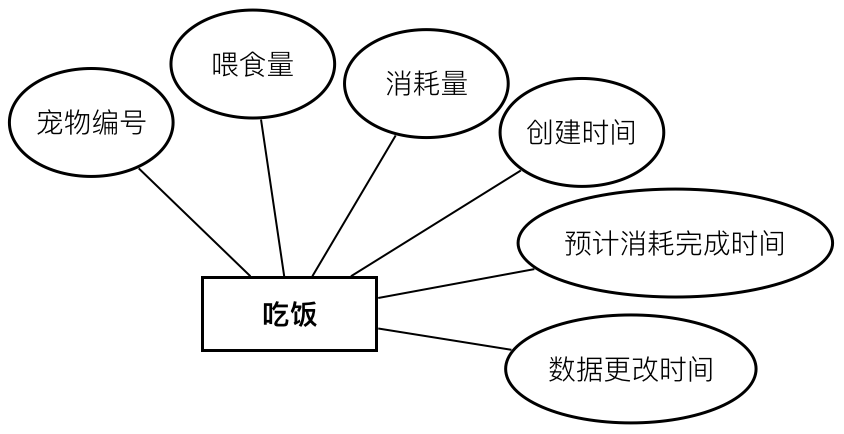


图 12宠物吃饭E-R图

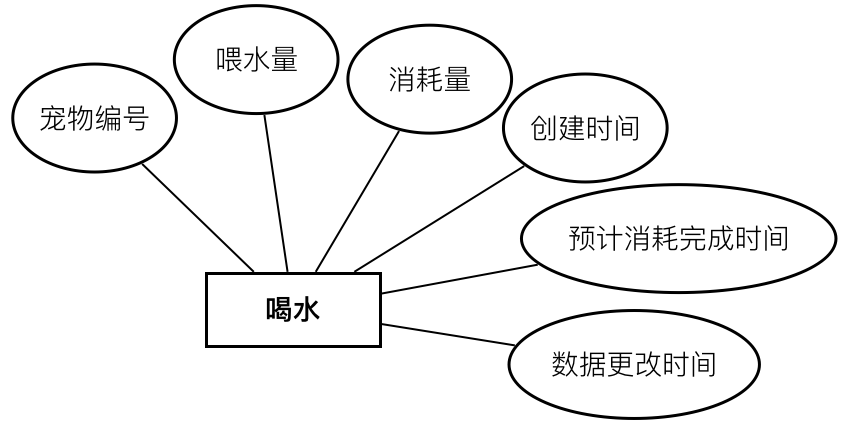


图 13宠物喝水E-R图

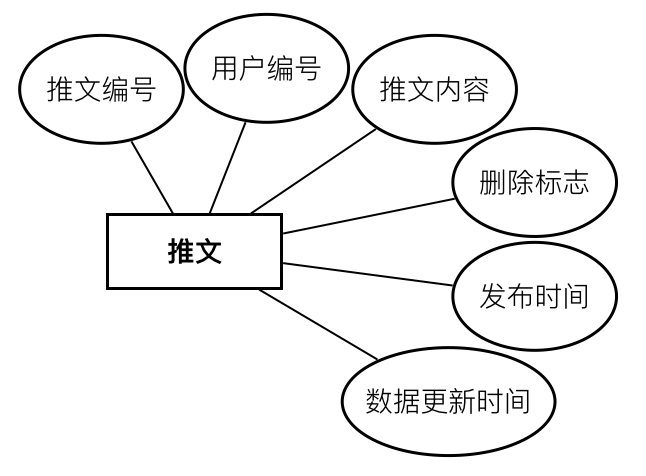


图 14推文E-R图

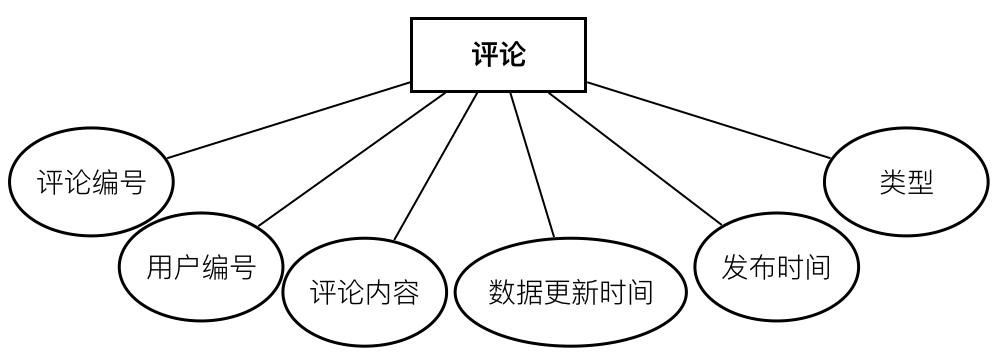


图 15评论E-R图

1. 全局E-R图如下所示：

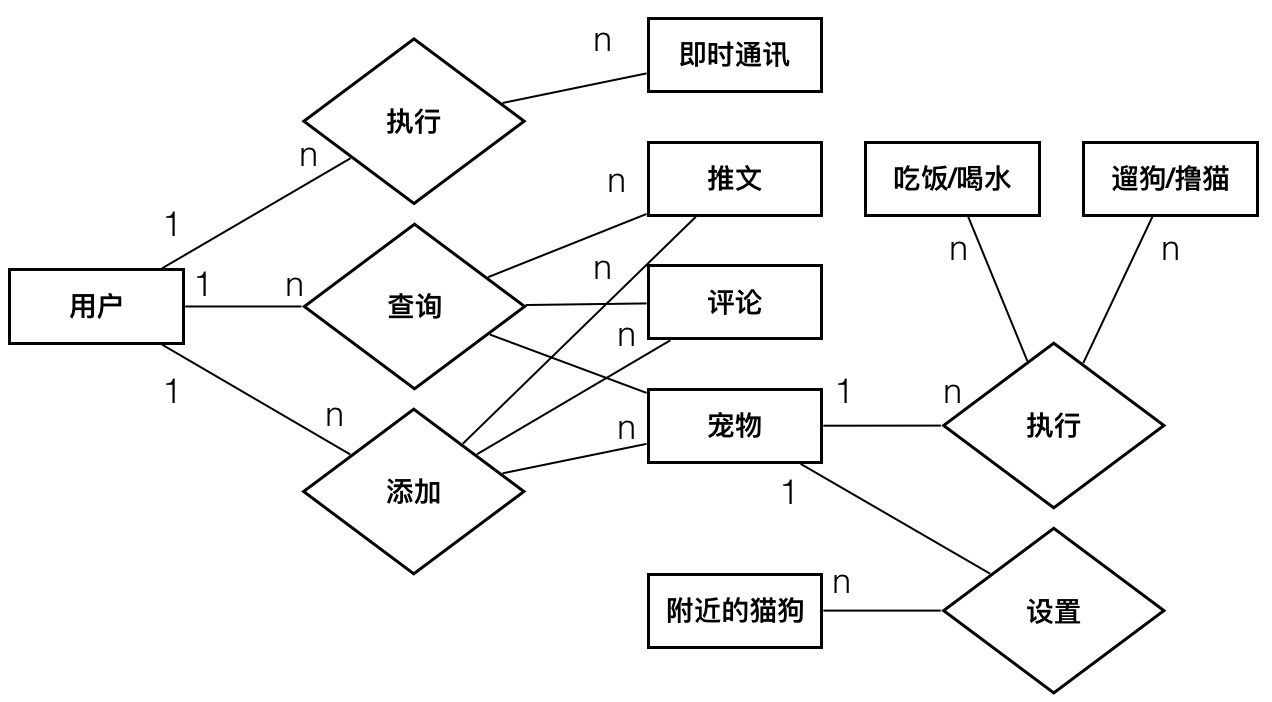


图 16PIGPEN全局E-R图

### 模型逻辑设计（关系模型）

关系的数据结构、关系的操作结合和关系的完整性约束组成关系模型。关系数据库系统的数据的组织方式使用关系模型进行描述，其实际的数据实体以及数据实体之间的联系都可使用用关系来描述。关系模型中数据在逻辑层中为二维表。关系模型里的关系操作包括选择、投影、除等增删改查操作两大部分。

可通过以下完整性的约束来定义关系完整性，分别是实体完整性、参照完整性和用户定义完整性，必须的完整性约束条件是实体完整性和参照完整性。ER图转换为关系模式的流程即为数据库逻辑结构设计。

1. 导出初始关系模式

用户（用户编号，手机号，昵称，密码，类型，性别，头像，创建时间，数据更新时间）

宠物（宠物编号，昵称，性别，类型，体重，绝育情况，感情状态，生日，品种，每日进食量，运动量，创建时间，数据更新时间）

遛狗（宠物编号，起点，终点，长度，是否完成，创建时间，数据更改时间）

撸猫（宠物编号，结束时间，是否完成，创建时间，数据更改时间）

吃饭（宠物编号，喂食量，消耗量，创建时间，预计消耗完成时间，数据更改时间）

喝水（宠物编号，喂水量，消耗量，创建时间，预计消耗完成时间，数据更改时间）

推文（推文编号，用户编号，推文内容，删除标志，发布时间，数据更新时间）

评论（评论编号，用户编号，评论内容，类型，发布时间，数据更新时间）

1. 子模式产生

子模式表现了出了其与初始模式之间的关系，且子模式是由初始模式导出的。

附近的猫狗（数据编号，宠物编号，GPS位置信息，创建时间，数据更新时间）

### 数据表

PIGPEN中数据表的主要表格设计如下表格所示，其中每一个表格表示其在数据库中一个独立的表。

表格 1用户表的字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 说明 |
| 1 | uid | bigint(20) | 用户唯一标识符 |
| 2 | phone\_number | varchar(11) | 用户手机号 |
| 3 | password | varchar(20) | 用户密码 |
| 4 | gender | int(11) | 用户性别 |
| 5 | avatar | int(11) | 用户头像索引 |
| 6 | created\_time | datetime(6) | 创建时间 |
| 7 | last\_updated\_time | datetime(6) | 更新时间 |

表格 2宠物表的字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 说明 |
| 1 | pet\_id | int(11) | 宠物唯一标识符 |
| 2 | nick\_name | varchar(18) | 宠物昵称 |
| 3 | gender | int(11) | 宠物性别 |
| 4 | pet\_type | int(11) | 宠物类型 |
| 5 | weight | int(11) | 宠物体重 |
| 6 | ppp\_status | int(11) | 宠物绝育状态 |
| 7 | love\_status | int(11) | 宠物感情状态 |
| 8 | birth\_time | int(11) | 宠物生日 |
| 9 | breed\_type | varchar(20) | 宠物品种 |
| 10 | food\_weight | int(11) | 宠物进食量 |
| 11 | created\_time | datetime(6) | 创建时间 |
| 12 | last\_updated\_time | datetime(6) | 更新时间 |
| 13 | user\_id | bigint(20) | 宠物主人标识符 |
| 14 | activity | int(11) | 宠物运动量 |

表格 3推文表的字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 说明 |
| 1 | id | int(11) | 推文唯一标识符 |
| 2 | content | longtext | 推文内容 |
| 3 | created\_time | datetime(6) | 创建时间 |
| 4 | last\_updated\_time | datetime(6) | 更新时间 |
| 5 | is\_deleted | int(11) | 是否删除标志位 |
| 6 | user\_id | bigint(20) | 推文发布者标识符 |

表格 4评论表的字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 说明 |
| 1 | id | int(11) | 评论唯一标识符 |
| 2 | is\_deleted | int(11) | 是否删除标志位 |
| 3 | created\_time | datetime(6) | 创建时间 |
| 4 | last\_updated\_time | datetime(6) | 更新时间 |
| 5 | user\_id | bigint(20) | 评论发布者标识符 |
| 6 | text | longtext | 评论内容 |
| 7 | parent\_id | int(11) | 父评论标识符 |
| 8 | root\_id | int(11) | 根评论标识符 |
| 9 | content\_type\_id | int(11) | 评论类型标识符 |

表格 5好友表的字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 说明 |
| 1 | id | int(11) | 记录唯一标识符 |
| 2 | status | int(11) | 好友状态标识符 |
| 3 | userA\_id | bigint(20) | 用户唯一标识符 |
| 4 | userB\_id | bigint(20) | 用户唯一标识符 |

## 系统开发与运行环境

### 开发工具

客户端的开发基于Xcode 集成开发环境进行。Xcode是Apple开发生态圈中的唯一官方指定集成开发环境，且提供大量的基础核心应用程序接口，Xcode中不仅仅只是一个代码编辑器和调试工具，其还集成了一个功能强大的应用视图分析工具，能够在应用程序运行时中截获当前应用程序的主窗口显示内容，并分析其视图层级，并提供分析功能。同时还可借用Xcode自身携带的各种分析应用程序性能的集成工具，例如分析应用程序视图滑动帧数工具、CPU占用分析工具、内存泄漏及循环引用工具等。

服务端开发基于PyCharm集成开发环境进行，其为jetbrains推出了众多编程语言集成开发环境之一，提供了优秀的代码编辑和调试工具。使用PyCharm还可直接通过配置连接上远程服务器进行开发，减少了远程服务器开发的痛点。

应用程序接口（即API）服务的测试工具使用PostMan进行，其前身为Chrome浏览器插件，后为提供更加便捷的使用体用独立出了一个完整的应用程序。PostMan提供完整的模拟HTTP请求方法，并可通过其进行请求的缓存和数据Mock测试，并可作为一个完整的web中转服务器。

### 运行环境

客户端

iOS 11.0及以上；

支持当前售卖的所有iPhone型号；

服务端

Django 2.0.7

Ubuntu. 16.04

Redis server 4.0.11

Mysql 5.6.0

Nginx。 1.4.7

开发工具

Xcode 10.1

PyCharm 2018.1.2

Postman 6.7.3

### 系统实现的关键技术

数据缓存。产品的属性定义了在日常的使用过程中会涉及到大量的数据交互，需要客户端提供一定的单机能力，在服务不可用等极端情况下需要保证良好的产品交互逻辑。在部分模块中会设计到大量的交互，甚至是重复性数据交互，需要服务端提供一定的“热数据”缓存能力，能够把请求频繁的数据进行缓存处理，降低服务端运行压力。

体征数据表现。对宠物体征数据进行归类计算并形成评分是产品的一大特色。关于如何良好的保证用户隐私的情况下通过特定指标数据结合自研公式计算出最终评分是决定用户留存率的一大关键问题。

宠物模拟产品涉及到众多模拟宠物真实生活习性的地方，如食物和水的消耗、遛狗撸猫等行为，需要结合宠物各个品种的标准值进行同等替换计算。

信息流信息流是构建产品社交属性的基本内容，而提供一个良好的信息流的展示是决定社交属性体现得好与坏的核心指标，产品将结合“推”“拉”两种信息流展示方式，持续优化信息流的展示。

# PIGPEN中间件模块

中间件是将具体业务和底层逻辑解耦的组件。其提供了对整个系统“化繁为简”的能力，善于管理和使用中间件能够提供工程效率和系统的鲁棒性。

## Token鉴权

使用Token作为用户鉴权的方式是目前工业界中最为流行的做法，考虑到移动应用无法使用传统Web应用中的Cookie和Session方式进行鉴权，再结合HTTP的无状态模式，从而衍生出了使用Token进行用户鉴权这一方式。

而使用Token进行用户鉴权时目前业界有不少做法，有把用户的用户名和密码或其它关键的标识信息经过一定的加密方式进行加密后放置于HTTP Header部分中在每一次进行HTTP请求时携带，也有一部分是Token的生成完全脱离用户管理体系，是一个完全独立的Token管理服务，不依托于系统的其它服务，当需要对用户或其它系统实体操作对象生成一个允许进入系统进行操作的全局唯一标识符时才调用该服务等方法。

在PIGPEN中，经过调研最后我们的选择是第二种方法，当用户登录时系统会将该请求脱离出完整的体系中，单独对其进行处理。检测用户此次提交的用户名和密码符合数据库中的某个用户时，通知基于Redis实现的Token管理服务生成加上了区别用户关键数据信息的JWT的格式Token字符串，用于标记该此次提交登录的用户信息有效，并在后续的HTPP请求中携带上该Token进行。

完整流程图如下图所示：

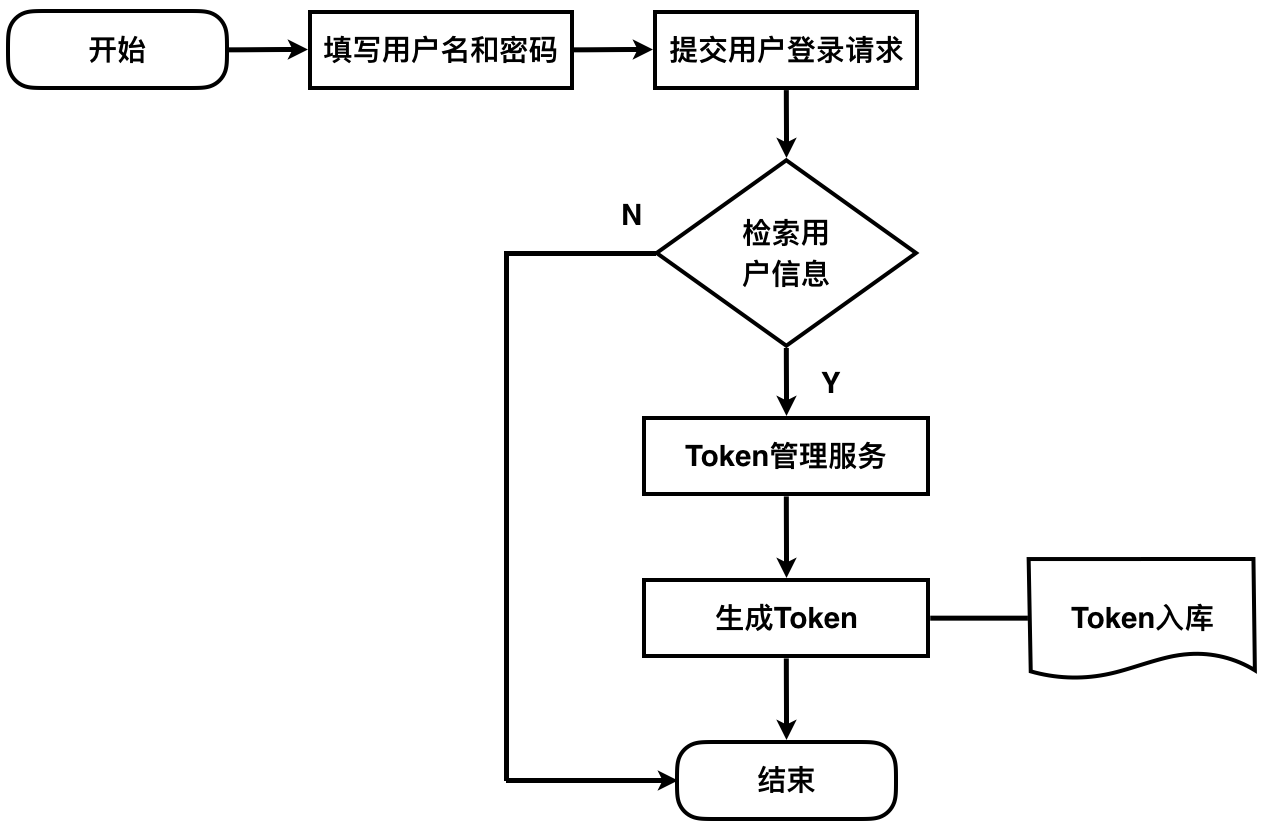


图 17Token生成流程图

Token中间件的实现代码如下所示：

图片包含 屏幕截图

描述已自动生成

图 18Token中间件实现

## HTTP请求方法和参数

HTTP是一个client-server协议，在开放式系统互联通信参考模型中位于应用层，也即请求通常是浏览器等客户端发起，在整个互联网系统中处在一个连接前端和后端的位置上，经过长时间的发展，HTTP已经成为了互联网服务的一个必不可少的工具，一个完整的HTTP请求分为起始行内容、HTTP Header和HTTP Body三部分组成。

在PIGPEN中，前后端进行了协调，HTTP请求方法分为了GET和POST两种方法，客户端需要PIGPEN系统的业务层中的各个方法时，需要根据业务层中所要求的HTTP请求方法进行提交对应的HTTP请求方法，否则将会被客户端所进行的HTTP请求将会以请求方法不一致而被拒绝。

在保证客户端的请求方法是符合系统业务层所要求的方法后，还要保证该业务层方法所要求的执行参数，只有方法的入餐保证了才能保证最后执行业务层的方法时的前置条件都是准确无误的。如果此次客户端提交的HTTP请求方法未能保证请求方法参数的完整性，同样系统会拒绝客户端此次的提交的业务层请求方法的执行。

完整的流程图如下所示：

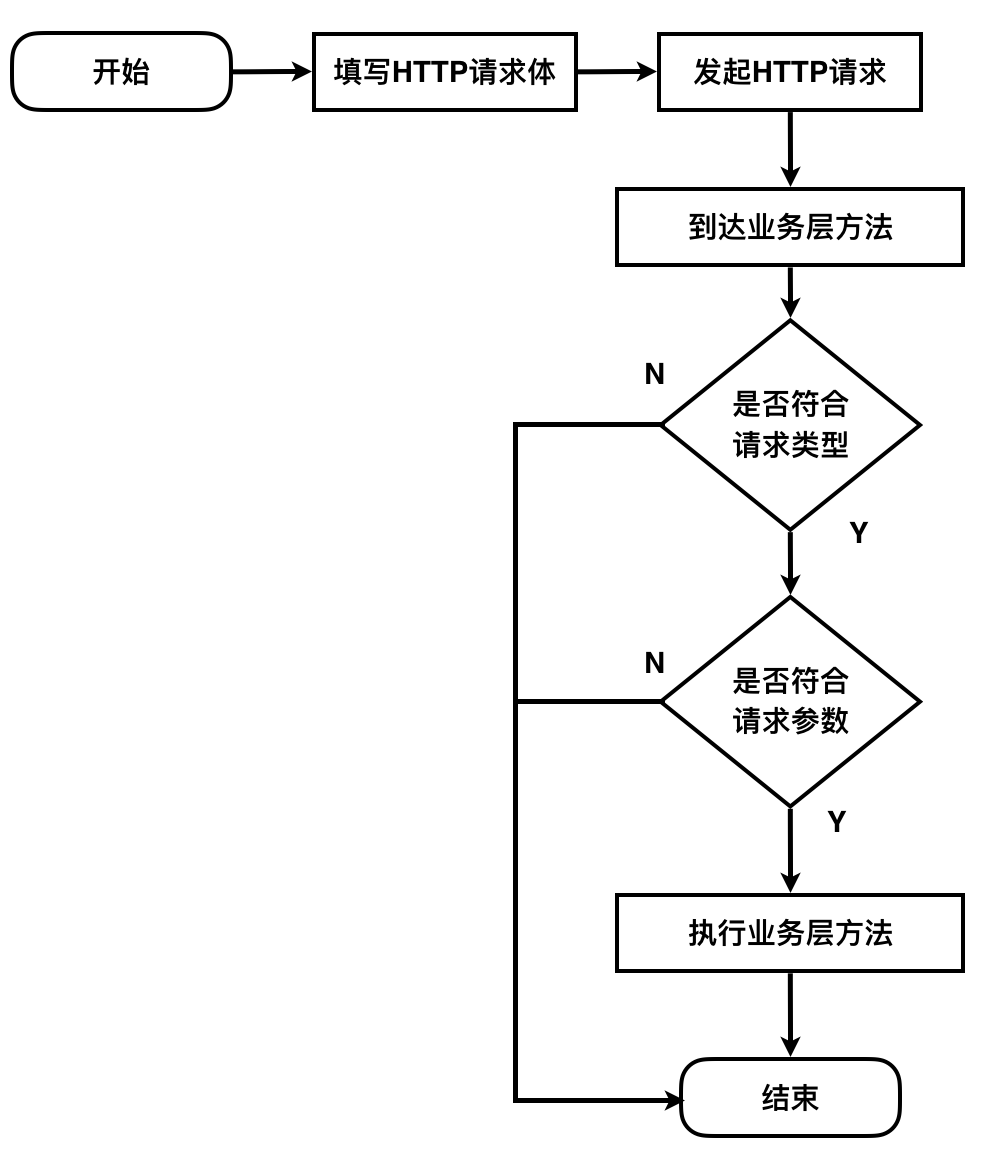


图 19HTTP请求方法和参数中间件流程图

HTTP请求方法和参数中间件的实现如下：

图片包含 屏幕截图

描述已自动生成

# PIGPEN用户管理模块

## 用户管理模块结构设计

### 用户管理模块功能简介

用户管理模块负责对使用本系统的用户进行管理，其负责的内容有用户的注册、用户的登录和更新用户信息等一系列需要使用到用户数据信息的操作。

如果是新用户使用本系统时，需要先进行用户注册，根据页面提示填写用户的个人信息如手机号，密码和昵称等，进而完成用户的注册。当用户在使用产品的过程中根据个人需求需要更新自己的用户信息数据，可进入用户信息更新页面进行更新用户的个人信息。客户端不支持多用户同时登录，当系统发现了同一用户灯登录系统时，将会对除当前用户登录的最后一台设备的所有其它设备进行强制性用户注销操作，废弃其通过登录操作获取到的用户鉴权Token。

### 用户管理模块功能图

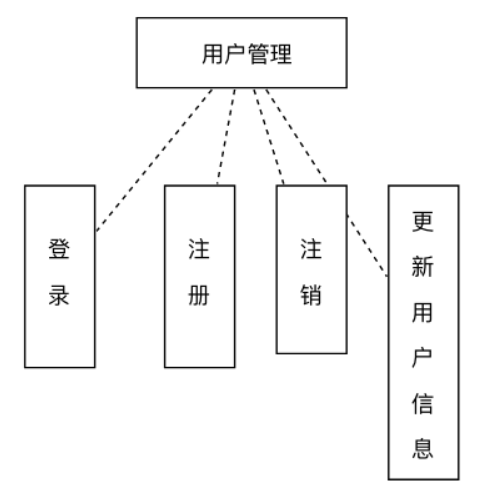


图 20用户管理功能结构图

## 用户管理模块详细设计与实现

### 用户登录

用户登录页面由PJUserLoginViewController.swift文件实现。该页面为客户端用户登录的主入口，用户只有通过该入门页面才能登录进入该页面，页面设计如下图所示：

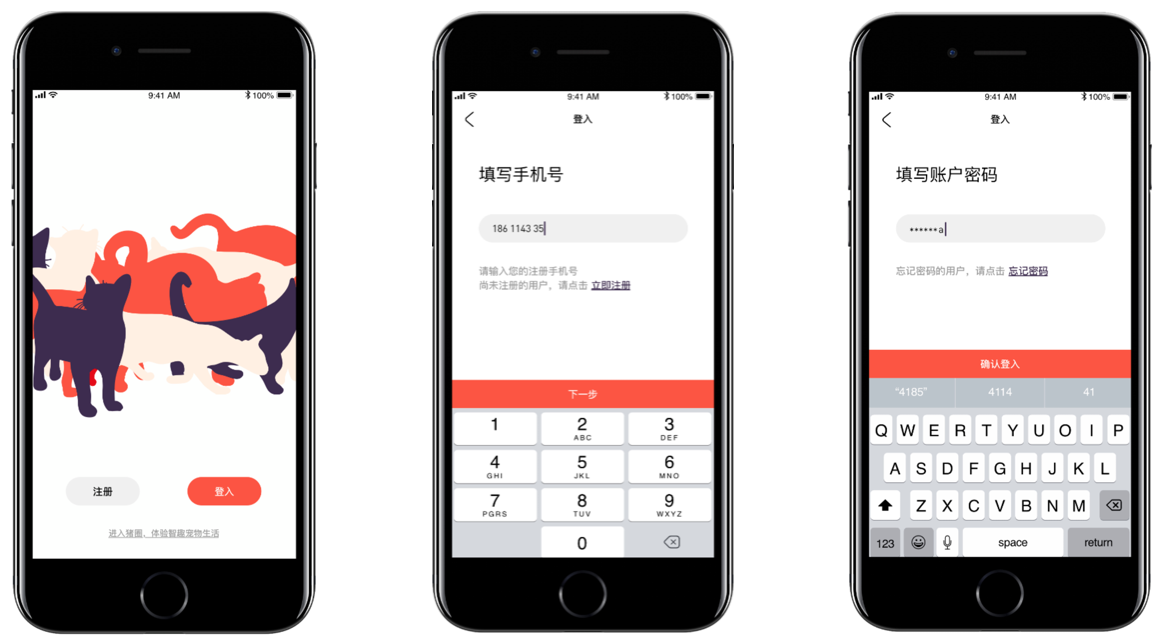


图 21用户登录页面设计

在用户的登录注册页面中，需要对用户输入的内容进行判断。对用户的手机号需要判断符合中国大陆手机号格式，且只能输入数字。对用户的密码需要限制其输入长度为18个32位长，可以输入设备支持的键入的所有字符，emoji表情符号除外。对用户输入的验证码限制可输入内容为4位，且只能输入数字。选择用户头像时为了保证在PIGPEN中宠物是第一公民的特点，对用户的头像进行了弱化，只允许用户选择我们的内置头像。

用户登录流程图如下所示：

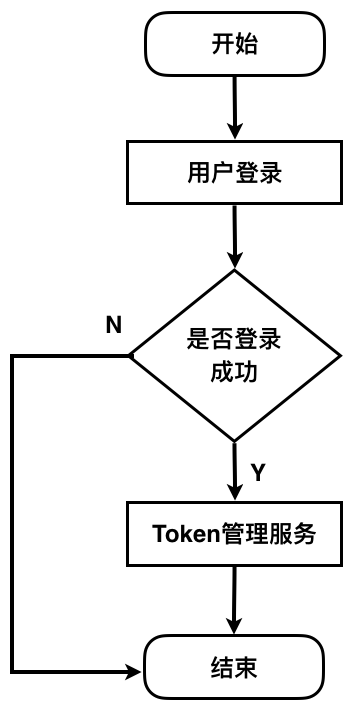


图 22用户登录流程图

用户在客户端的登录页面中输入完毕登录页面所要求的必须参数内容后，点击登录按钮，会出发登录页面对用户数据数据的检测方法，在该检测方法中会对用户输入的手机号和密码做有效性检测，比如检测用户输入的手机号是否符合中国大陆手机号规则，检测用户输入的密码是否长度超长等。

用户登录会调用客户端网络请求工具类单例发送HTTP的POST请求，在正式提交HTTP请求之前，会对用户输入的密码进行特殊规则拼接后使用MD5加密算法进行加密。为提高整个系统的安全性，服务端用户表中所存储的密码同样是存储加密过的字符串，在每次用户登录时都会去进行检测。

用户登录的核心代码如下所示：

图片包含 屏幕截图

描述已自动生成

图 23用户登录的客户端核心代码

登录数据提交到服务端后，服务端需要根据预先设定好的用户密码拼接规则对客户端发送的HTTP请求中携带上的用户名和密码参赛计算出最终的加密字符串，与数据库中的数据进行匹配。匹配完成后，进行JSON数据的渲染，最终把渲染完成后的用户登录数据返回给客户端。用户登录的服务端核心代码如下所示：

图片包含 屏幕截图

描述已自动生成

图 24服务端用户登录的核心代码

### 用户注册

新用户的添加目前有两种方式，可以通过客户端进行注册添加新用户，也可以通过系统后台进行添加。通过系统后台添加的用户一般都是管理员用户，而通过客户端注册的用户都是普通用户。在客户端中注册管理员用户是不安全的，会造成系统整个管理的混乱，且系统的数据安全性会受到影响，

以下将只介绍在客户端中进行普通用户注册的方法。考虑到在客户端注册普通用户需要填入手机号、验证码、密码和头像信息，可将页面设计如下图所示：



图 25用户注册页面的设计

用户注册页面使用了多个输入框，并在每个页面的确认按钮上都绑定了触发事件。在用户输入手机号注册框的输入事件中对用户输入的内容使用了正则匹配手机号。对用户输入的昵称和密码内容进行了非空判断和格式规则验证。

图片包含 屏幕截图

描述已自动生成

图 26用户手机号正则匹配代码实现

图片包含 屏幕截图

描述已自动生成

图 27用户注册页面输入框的有效性验证代码实现

获取验证码时客户端调用Mob SDK进行验证码的匹配，验证匹配通过后才能进行剩余数据的提交。当客户端把用户注册所需的数据通过HTTP方式POST请求发送到服务端时，服务端需要先判断当前用户表里是否存在相同手机号的用户，如果存在则返回“用户已存在”，反之在把用户数据写入数据库后，要通知Token管理服务进行Token的生成，并把相关的用户信息，如昵称等非敏感信息联合生产的Token渲染成JSON格式的数据返回给客户端。

服务端的用户注册核心代码如下所示：

图片包含 屏幕截图

描述已自动生成

图 28用户注册后端核心代码

### 搜索用户

搜索用户的数据由每一个用户创建时激发更新缓存，因为用户的搜索操作是一个只读操作，考虑到如果每次的搜索都需要进行用户表关键字段的全局扫描这势必会带来性能的损耗和响应时间的加长，因此将在系统中维护一套激发缓存策略，当用户注册时更新缓存和追加索引，缩短耗时。

根据搜索用户需要填写的数据和返回搜索的内容可对其页面做如下设计：

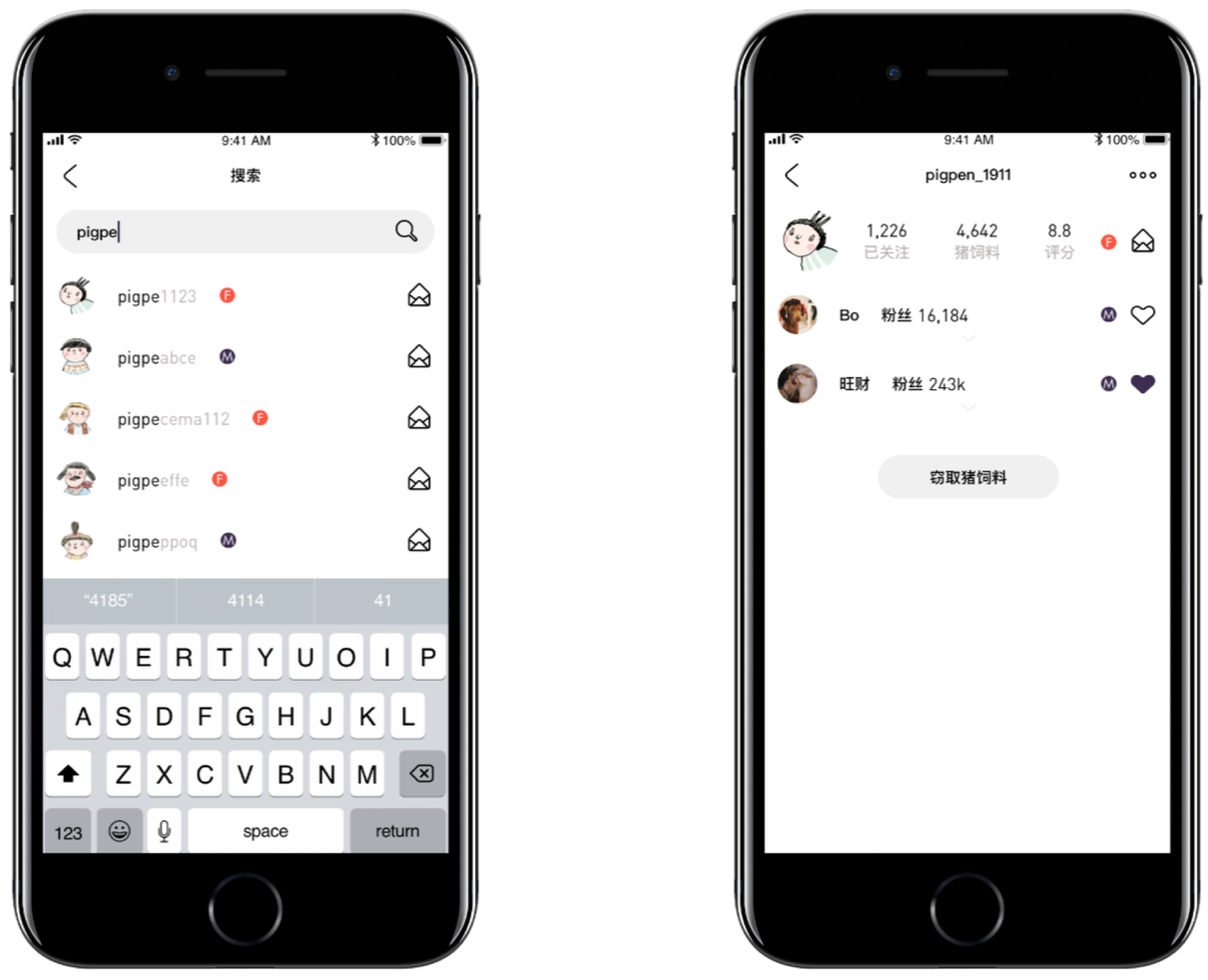


图 29搜索用户界面设计

用户注册的流程图如下所示：

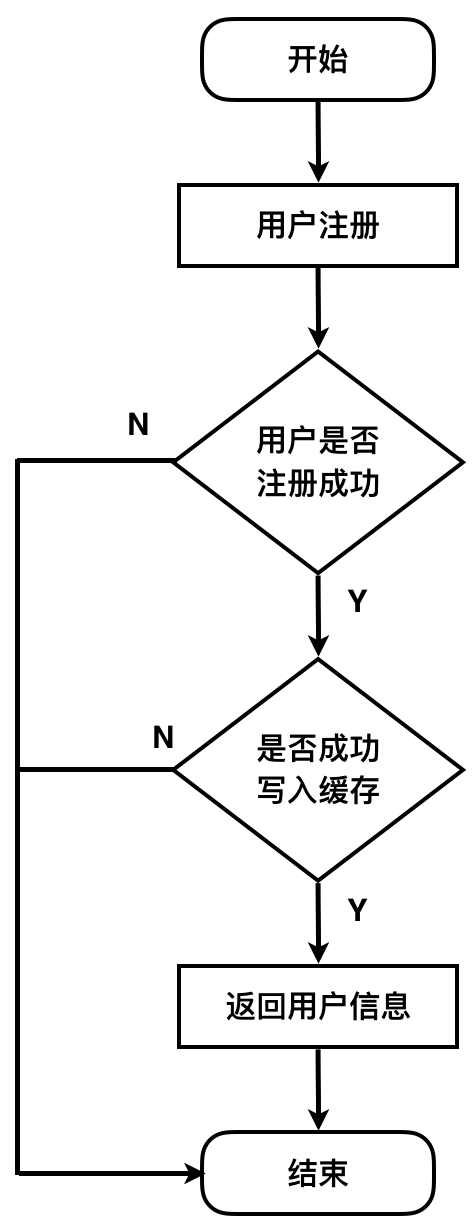


图 30用户注册流程图

经过一番调研后，得出系统不需要引入过于庞大的缓存组件，只需要按需增加需要的组件即可，因此使用了Whoosh组件作为用户数据的缓存。通过使用该组件可以达到对实时的获取搜索结果，在客户端中使用了复用了同一个TCP链接通道，报纸搜索结果的实时性。使用该缓存组件需要先在settings.py文件中注册并配置相关信息，配置的核心代码如下所示：

图片包含 屏幕截图

描述已自动生成

图 31缓存组件核心配置信息

在工程中对应的app文件夹下新建索引文件，并锁定用户模型的索引字段nick\_name进行缓存，当执行生成缓存的指令python manage.py rebuild\_index将触发缓存组件Whoosh重建索引的操作，最终在需要调用缓存的用户搜索接口中根据参数返回model数据和搜索集合数据。搜索好友的业务层代码实现如下图所示：

图片包含 屏幕截图

描述已自动生成

图 32搜索好友业务层核心代码

### 更新用户信息

更新用户信息需要在允许的时间间隔内，在该时间间隔内，用户可更新其头像等非核心信息。用户更新信息时，需要在客户端上进入更新信息页面PJUserUpdateViewController.swift中提交需要更新的信息，客户端做非空校验和数据可行性校验后通过HTTP的POST的方法提交到对应的后端服务中，后端服务解析提交的请求参数执行更新用户信息的方法，写入数据库，并返回更新用户信息完成的JSON格式信息返回给发起更新用户信息请求的客户端，客户端作出弹窗提示告知当前用户执行的更新用户信息操作已经完成，客户端刷新当前页面。

根据更新用户信息所需的字段，做出的设计图如下所示：

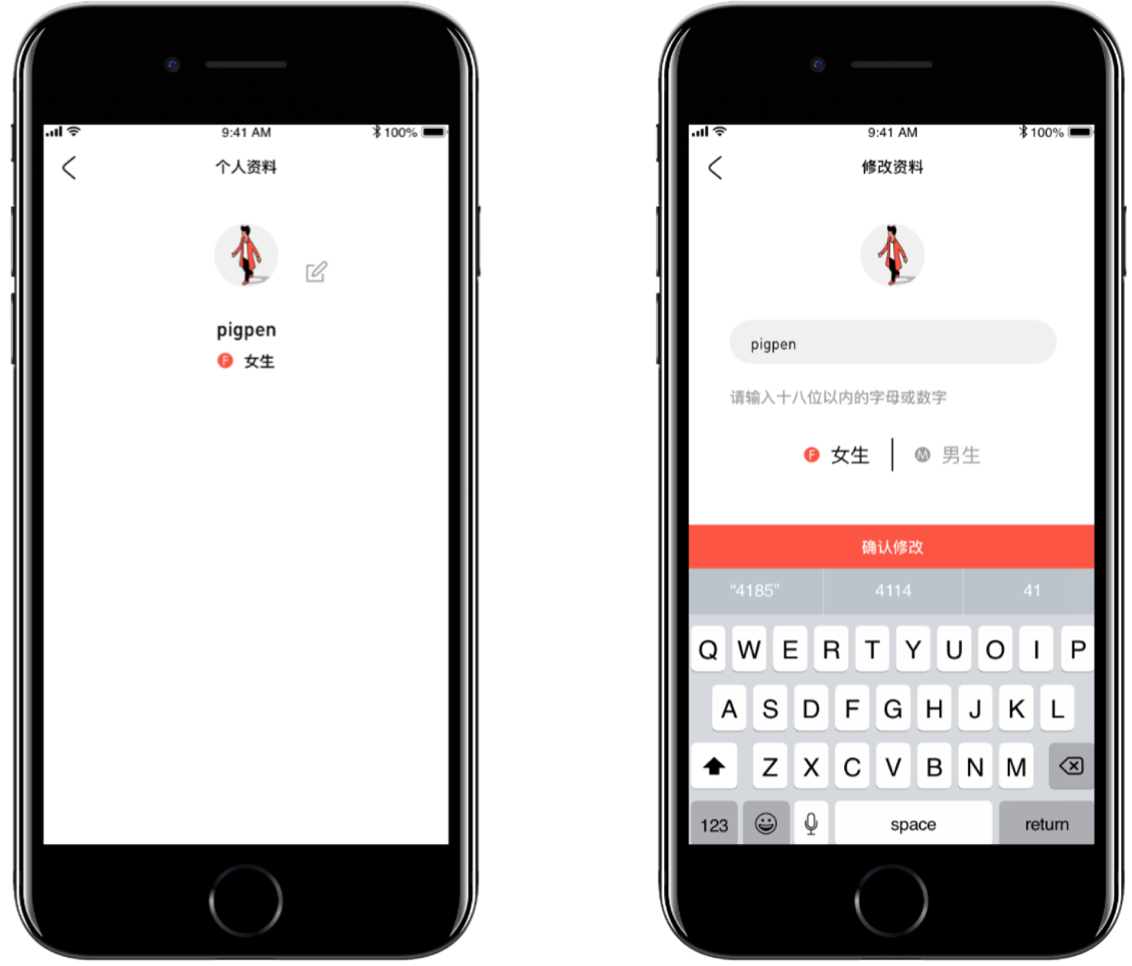


图 33修改用户信息页面设计

更新用户信息的服务端核心代码如下图所示：

图片包含 屏幕截图

描述已自动生成

图 34更新用户信息服务端核心代码

### 分享名片

为了对用户进行自发性传播产品的行为，需要对用户紧张价值分析，激励用户产生自主分析的意愿。对此PIGPEN设计了一套能够对用户产生自主分享欲望的机制，目前首先是通过用户自定义名片的分享，通过在名片上个人价值信息产生对自己的肯定，进而引发对各自用户拥有的社群进行传播，最终引发导流行为，把其它新用户引入产品中进行使用，达到一个完整的互联网用户产生闭环。

# PIGPEN即时通讯模块

## 即时通讯模块结构设计

### 即使通讯模块功能简介

即使通讯是目前具备社交属性产品都会具备的功能。对于一个即时通讯功能来说需要保证其高可用性和消息及时性，需要对用户与用户之间的消息内容进行缓存，具备管理与用户关联的会话列表，服务端需要维护一套完整的消息通道，对自定义消息体需要提供一定的拓展功能，以至于能够完成文字、图片、音频、视频、地图和自定义表情数据的传输。

经过调研，选择了融云作为即时通讯模块的基础聊天服务提供商，集成了融云的基础聊天服务，自定义即使通讯模块的相关UI，如会话列表、聊天界面等。

### 即使通讯模块功能图

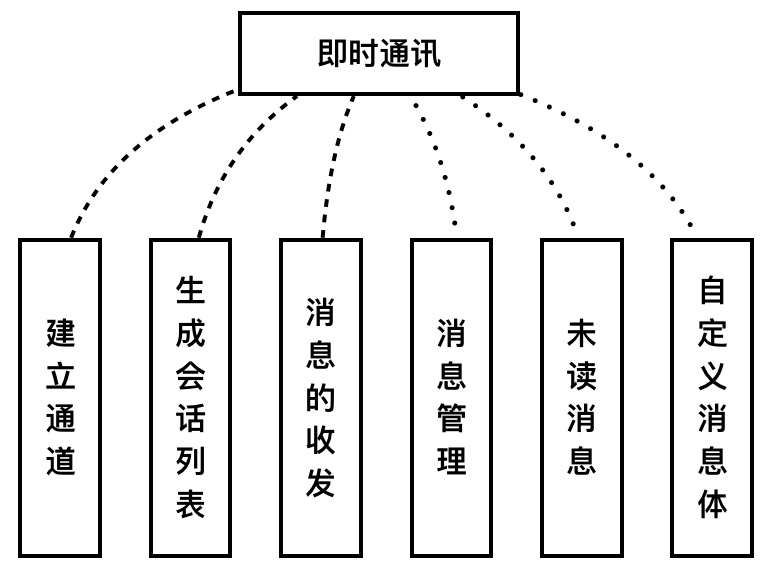


图 35即时通讯模块功能结构图

## 即时通讯模块详细设计与实现

### 页面设计

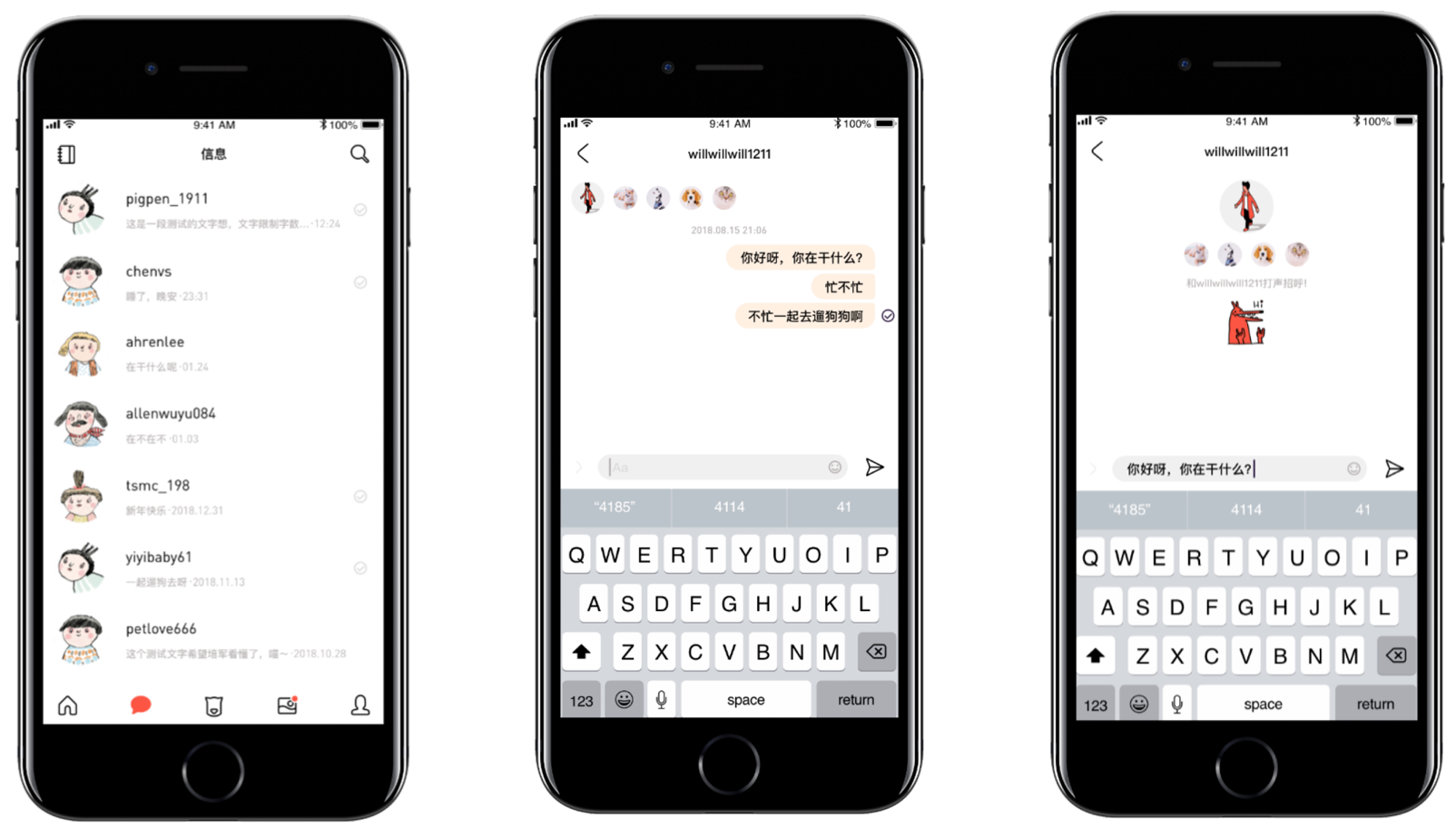


图 36即时通讯相关界面的设计

### 初始化及通讯密钥生成

在PIGPEN即时通讯中的通讯密钥被称之为Token ，也即用户令牌，相当于用户在PIGPEN客户端上连接PIGPEN即时通讯服务的的身份凭证。 在用户连接PIGPEN即使通讯服务器之前，需要先请求获取PIGPEN服务端获取Token的API获取 Token，客户端请求成功后解析服务端返回数据，获取Toke内容后即可进入下一步连接PIGPEN即时通讯服务器。

PIGPEN服务端生成通讯密钥的核心代码为：

图片包含 屏幕截图

描述已自动生成

图 37服务端生成通讯密钥Token的核心代码

### 建立连接

在 PIGPEN客户端的整个生命周期中，我们只需要调用一次建立与PIGPEN服务端连接的方法即可。之后不管是是网络异常还是PIGPEN应用程序的前后台的切换等问题，PIGPEN客户端会经由即时通讯持久化重连服务进行自动重连操作。 PIGPEN客户端的即时通讯持久化服务针对应用程序的前后台以及可能出现的各种网络情况，进行了即时通讯的连接和重新连接机制的改善。

图片包含 屏幕截图

描述已自动生成

图 38建立连接的客户端核心代码

### 发送与接收消息

一旦客户端和即时通讯服务连接成功之后，用户就可以在稳定的情况下进行消息的发送和接收，PIGPEN的即时通讯模块支持用户发送文本、图片、音频、视频、位置信息、emoji表情、自定义表情自定义消息体。考虑到多种消息体的内容不同，需要对不同消息体的内容做抽象，把不同消息体的内容都转化成了二进制流数据进行传输，进行可信传输，在客户端上分别进行消息体内容的压缩转码和解压转码。以下以PIGPEN即时通讯模块中的文本消息为例，展示消息的接收和发送流程。

图片包含 屏幕截图

描述已自动生成

图 39即时通讯模块中的发送文本消息的核心代码

图片包含 屏幕截图

描述已自动生成

图 40即使通讯模块中接收文本消息的核心代码

### 获取会话列表

为了降低服务端的压力，我们把与用户相关的会话列表存储在了用户设备中，并在本地使用基于sqlite数据库完成了会话列表数据的存储。用户主动通过搜索其它用户，并与其它用户进行了聊天操作，当聊天数推送到对方设备中时，PIGPEN认为该会话建立，并构建二者之间的消息通道，在用户各种的设备中进行会话列表的生成和存储。

但如果用户因其它原因把与之前其它建立起消息通道以及会话列表的用户数据进行了删除，例如删除了PIGPEN应用程序，则与该用于相关所有即时通讯数据均会被抹除，因为这些数据是存储在该用户的设备中。

图片包含 屏幕截图

描述已自动生成

图 41即时通讯模块中获取消息列表的核心代码

### 获取消息未读数

用户的某一个消息未读数是基于当前消息队列中剩余未读消息与已读消息的差值而生成的值，该值受用户当前设备中的接收到的所有的消息有关，且受限于用户当前设备所处的网络环境。

当有新消息推送到当前用户设备时，PIGPEN客户端会把该条消息拉取到本地消息队列中。PIGPEN应用程序会在用户下一次进入应用程序的消息页时对比已消耗消息队列和未消耗消息队列，把未消耗的消息增加到会话列表中。

用户点击新消息时，未消耗消息队列数据减一，已消耗消息队列数据加一，更新由这两个消息队列做差值而生成的消息未读数。获取消息未读数的核心代码如下所示：



图 42即时通讯模块中获取所有消息未读数

# 宠物管理模块

## 宠物管理模块结构设计

### 宠物管理模块功能简介

### 宠物管理模块功能图

## 宠物管理模块详细设计与实现

### 页面设计

### 创建宠物

### 搜索宠物

### 关注宠物

### 拉黑宠物

### 更新宠物信息

### 吃饭

### 喝水

### 遛狗

### 撸猫

# PIGPEN寻找附近猫狗模块

## 寻找附近猫狗模块结构设计

### 寻找附近猫狗模块功能简介

### 寻找附近猫狗模块功能图

## 寻找附近猫狗模块详细设计与实现

### 页面设计

### 处理用户GPS信息

### 拉取附近猫狗位置信息

# PIGPEN评分模块

## 评分模块结构设计

### 评分模块功能简介

### 评分模块功能图

## 评分模块详细设计与实现

### 页面设计

### 评分维度

### 每日评分

### 每周评分

### 每月评分

### 总评分

### 评分重制

# PIGPEN朋友圈模块

## 朋友圈模块结构设计

### 朋友圈模块功能简介

### 朋友圈模块功能图

## 朋友圈模块详细设计与实现

### 文章创建

### 文章删除

### 评论创建

### 评论删除

### 图文混排引擎

### 信息流

### 点赞统计

### 阅读统计

### 缓存优化

# PIGPEN系统评价

## 特色与优点

## 不足与改进

### 系统存在的不足

### 系统改进方案

# 总结

# 参考文献