

一、序言

1.1 编写目的

- (1) 本数据库设计说明书是用于指导“i寻”微信小程序的开发的云数据库设计；
- (2) 本数据库用于开发人员进行项目设计，并提示开发人员如何脱离传统繁杂的JDBC体系转而开始采用快速的云数据库开发。是对于设计中的数据库的所有标识、逻辑结构和物理结构做出具体的设计规定。；
- (3) 本数据库设计说明书，目标是建立一个合适的数据模型。能够再满足用户的同时也满足开发者的需求。

1.2编写背景

- (0) 数据库采用微信云开发数据库开发环境名（即数据库名）：cloud1-7gd3mlk6fe9855dd
- (1) 项目名：i寻
- (2) 开发团队：薪火队
- (3) 开发成员：林禹铭、杨仕金、张其津、张永朝、邱勇威、翁铭
- (4) 项目背景：即使现代安全技术已经十分先进，但物品遗失依然时有发生，很多时候物品丢失是不可避免的，虽然目前网络社交平台众多，但缺少针对失物招领和寻物启事的平台。虽然QQ学院墙、朋友圈等都有类似消息发布，但是发布的时间过长。失物招领、寻物启事的微信小程序将提供一个平台，统一将消息发布出来上，聚少成多，必然能够提高失物招领的效率。

1.3定义

序号	英文缩写	英文全称	中文全称	属性
1	SN	Software name	软件名称	i寻
2		Entity	实体	客观世界中存在的且可互相区别的事物
3		Contact	联系	客观事物中的彼此之间的关系
4		Attribute	属性	实体或者联系所具有的性质
5	DB	DataBase	数据库	用来保存系统数据的后台应用软件
6		Filed	字段	表中各个记录的名字
7		Table	表	不同字段汇总成的集合
8	MySQL	My Structured Query Language	数据库查询和程序设计语言	关系型数据库管理系统
9		JavaScript Object Notation DataBase	轻量级的数据交互格式数据库	文档型数据库管理系统

1.4参考资料

- (1)[大厂都在用的 MySQL 团队开发规范，太详细了！](#)
- (2)[微信云开发文档](#)

二、微信云开发指导

2.1必要的重点概念

2.1.1概念对应关系

数据库云开发提供了一个 JSON 数据库，顾名思义，数据库中的每条记录都是一个 JSON 格式的对象。一个数据库可以有多个集合（相当于关系型数据中的表），集合可看做一个 JSON 数组，数组中的每个对象就是一条记录，记录的格式是 JSON 对象。

关系型数据库和 JSON 数据库的概念对应关系如下表：

关系型	文档型
数据库 database	数据库 database
表 table	集合 collection
行 row	记录 record / doc
列 column	字段 field

2.1.2 储存

云开发提供了一块存储空间，提供了上传文件到云端、带权限管理的云端下载能力，开发者可以在小程序端和云函数端通过 API 使用云存储功能。

在小程序端可以分别调用 `wx.cloud.uploadFile` 和 `wx.cloud.downloadFile` 完成上传和下载云文件操作。下面简单的几行代码，即可实现在小程序内让用户选择一张图片，然后上传到云端管理的功能

2.1.3 云函数

云函数是一段运行在云端的代码，无需管理服务器，在开发工具内编写、一键上传部署即可运行后端代码。

小程序内提供了专门用于云函数调用的 API。开发者可以在云函数内使用 `wx-server-sdk` 提供的 `getWXContext` 方法获取到每次调用的上下文（appid、openid 等），无需维护复杂的鉴权机制，即可获取天然可信任的用户登录态（openid）。

2.1.4 云环境

一个环境对应一整套独立的云开发资源，包括数据库、存储空间、云函数等资源。各个环境是相互独立的，用户开通云开发后即创建了一个环境，默认可拥有最多两个环境。在实际开发中，建议每一个正式环境都搭配一个测试环境，所有功能先在测试环境测试完毕后再上到正式环境。以初始可创建的两个环境为例，建议一个创建为 test 测试环境，一个创建为 release 正式环境。同时，在 Network 面板中会输出各个云开发操作的请求详情，其中包括该调用所请求的环境 ID

三、外部设计

3.1 标识符和状态

数据库软件名称：微信云开发数据库（JSON 数据库）

数据库环境id：cloud1-7gd3mlk6fe9855dd

3.2 使用程序

本数据库在微信开发者中绑定“i寻”小程序直接使用

3.3 规范

3.3.1 命名规范

遵循微信云开发，尽可能每个集合（表）的命名采用大写驼峰方式命名

集合名尽可能体现了集合的内容，最好与相关的page或者事件的有关，便于项目开发者理解；

主键必须采用"_id"用于接收系统生成的id（微信自动生成不可自己修改）

3.3.2 设计规范

在本系统中，采用面向对象的设计方法，数据库的设计采用 PowerDesigner 进行，首先进行对象实体的设计，最后将对象持久化到数据库中，所有的表和表之间的关联(E-R 图)都采用标准的 ProcessOn 设计工具进行，这样能够将整个系统的设计和数据库设计有机的结合起来。

3.3.3支持软件

以下工具所使用的 MySQL 为 5.5 版本

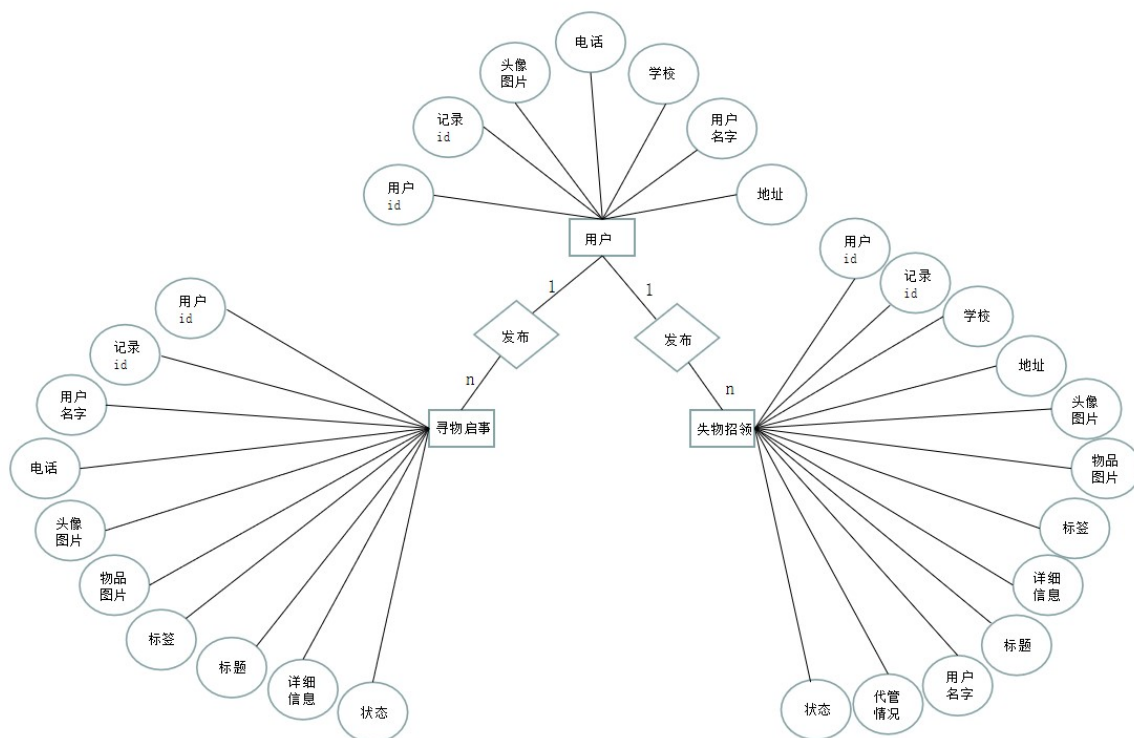
数据库设计工具：Navicat for MySQL

项目开发工具：微信开发者工具、HBuilderX 等

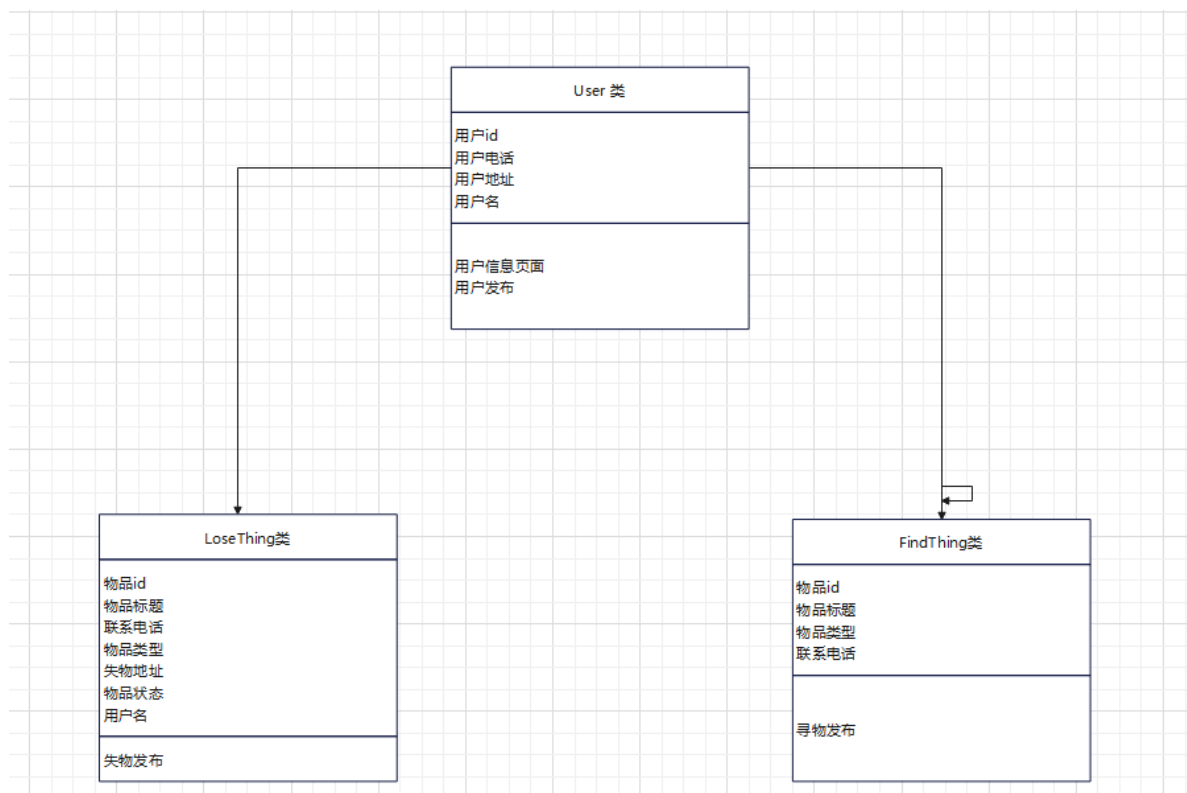
四、结构设计

4.1概念结构设计

4.1.1E-R图



4.1.2类图



4.2逻辑结构设计

4.2.1user表结构

用于存储用户信息

对象 user @ixun (ysj) - 表							
保存 添加字段 插入字段 删除字段 主键 上移 下移							
字段	索引	外键	触发器	选项	注释	SQL 预览	
名	类型	长度	小数点	不是 null	键	注释	
id	varchar	255	0	<input checked="" type="checkbox"/>	1		
_openid	varchar	255	0	<input type="checkbox"/>			
call	varchar	255	0	<input type="checkbox"/>			
headImg	varchar	255	0	<input type="checkbox"/>			
home	varchar	255	0	<input type="checkbox"/>			
schoolId	varchar	255	0	<input type="checkbox"/>			
userName	varchar	255	0	<input type="checkbox"/>			

4.2.2seeking表结构

失物招领信息

对象 seekthing @ixun (ysj) - 表							
保存 添加字段 插入字段 删除字段 主键 上移 下移							
字段	索引	外键	触发器	选项	注释	SQL 预览	
名	类型	长度	小数点	不是 null	键	注释	
id	varchar	255	0	<input checked="" type="checkbox"/>	1		
_openid	varchar	255	0	<input type="checkbox"/>			
callOrVx	varchar	255	0	<input type="checkbox"/>			
headImg	varchar	255	0	<input type="checkbox"/>			
seekImageBase64	varchar	255	0	<input type="checkbox"/>			
seekNowThing	varchar	255	0	<input type="checkbox"/>			
seekRemark	varchar	255	0	<input type="checkbox"/>			
seekTitle	varchar	255	0	<input type="checkbox"/>			
upshot	varchar	255	0	<input type="checkbox"/>			
userName	varchar	255	0	<input type="checkbox"/>			

4.2.3losething表结构

寻物启事信息

对象	losething @ixun (ys) - 表						
保存	添加字段	插入字段	删除字段	主键	上移	下移	
字段	索引	外键	触发器	选项	注释	SQL 预览	
名	类型	长度	小数点	不是 null	键	注释	
id	varchar	255	0	<input checked="" type="checkbox"/>	1		
_openid	varchar	255	0	<input type="checkbox"/>			
addRess	varchar	255	0	<input type="checkbox"/>			
headImg	varchar	255	0	<input type="checkbox"/>			
imgeBase64	varchar	255	0	<input type="checkbox"/>			
nowLocation	varchar	255	0	<input type="checkbox"/>			
offOn	varchar	255	0	<input type="checkbox"/>			
remark	varchar	255	0	<input type="checkbox"/>			
schooId	varchar	255	0	<input type="checkbox"/>			
title	varchar	255	0	<input type="checkbox"/>			
upshot	varchar	255	0	<input type="checkbox"/>			
userName	varchar	255	0	<input type="checkbox"/>			

4.3物理结构设计

本项目（i寻）采用微信云开发数据库（JSON数据库）进行数据设计，全部物理逻辑设计均由微信开发者工具自动完成。

4.4性能的规定

软件开发过程中采用迭代式开发模型，同时充分做好前期的准备工作，尽量降低软件后期变动，软件设计考虑可维护性，可拓展性，软件能够在最小变化范围内接受操作方式，数据精度等变化。

4.5故障处理要求

允许客户出错，调用微信云开发自处理，利用云管家的功能，将错误信息直接删除。无需担心负载，物理故障的问题

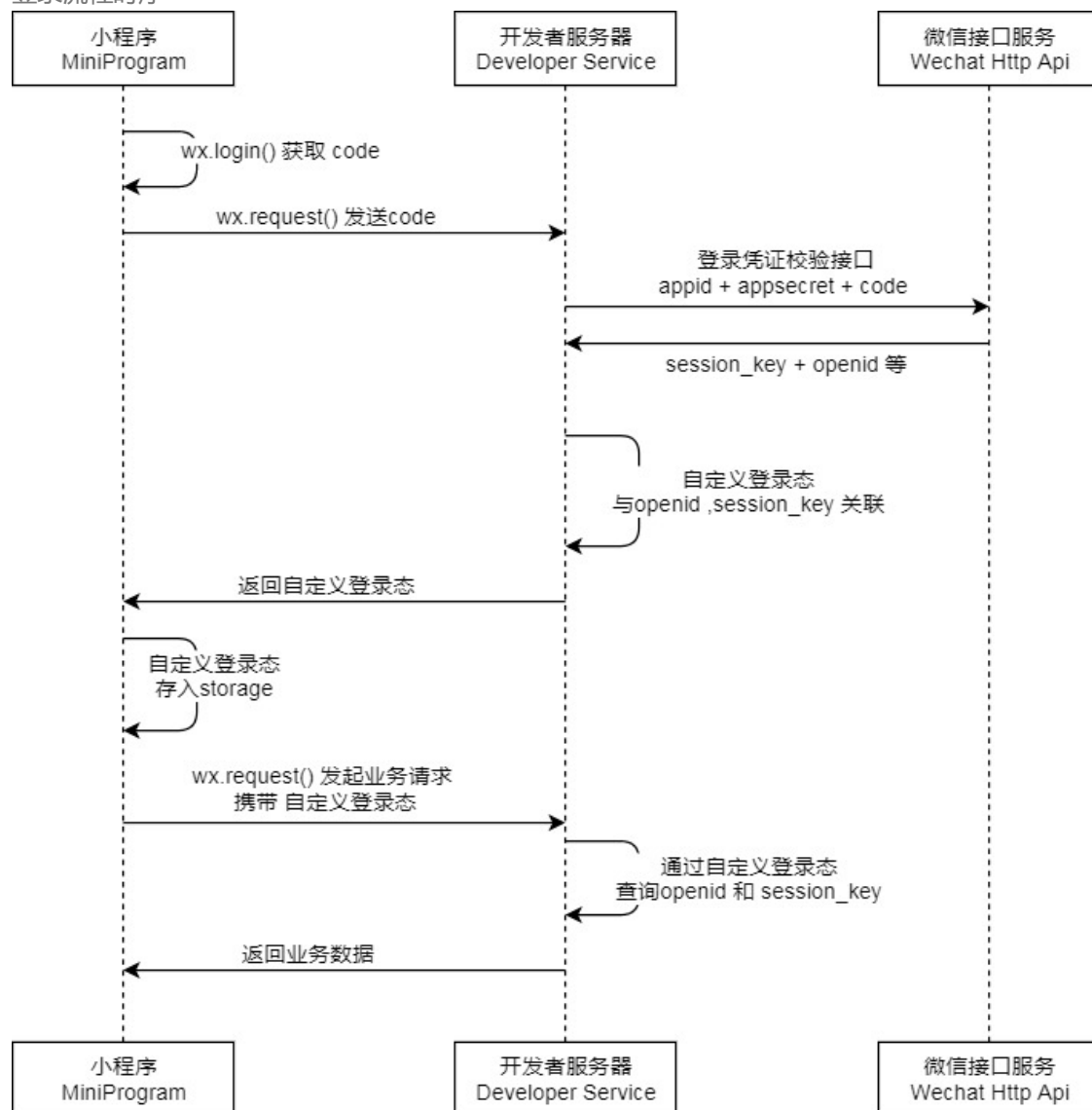
五、运用设计

5.1系统设计

5.1.1验证登录

小程序可以通过微信官方提供的登录能力方便地获取微信提供的用户身份标识，快速建立小程序内的用户体系

登录流程时序



解释说明：调用 wx.login() 获取 临时登录凭证code，并回传到开发者服务器。

特别注意：调用 auth.code2Session 接口，换取 用户唯一标识 OpenID、用户在微信开放平台帐号下的唯一标识UnionID（若当前小程序已绑定到微信开放平台帐号）和 会话密钥 session_key。之后开发者服务器可以根据用户标识来生成自定义登录态，用于后续业务逻辑中前后端交互时识别用户身份。

会话密钥 session_key 是对用户数据进行 加密签名 的密钥。为了应用自身的数据安全，开发者服务器不应该把会话密钥下发到小程序，也不应该对外提供这个密钥。

临时登录凭证 code 只能使用一次。

5.1.2数据安全

利用微信小程序的安全规则要求前端发起的查询条件必须是安全规则的子集，否则拒绝访问。比如定义一个读写访问规则是 `auth.openid == doc._openid`，则表示访问时的查询条件（doc）的 `openid` 必须等于当前用户的 `openid`（由系统赋值的不可篡改的 `auth.openid` 给出），如果查询条件没有包含这项，则表示尝试越权访问 `_openid` 字段不等于自身的记录，会被后台拒绝访问。

5.2.2 用户权限设计

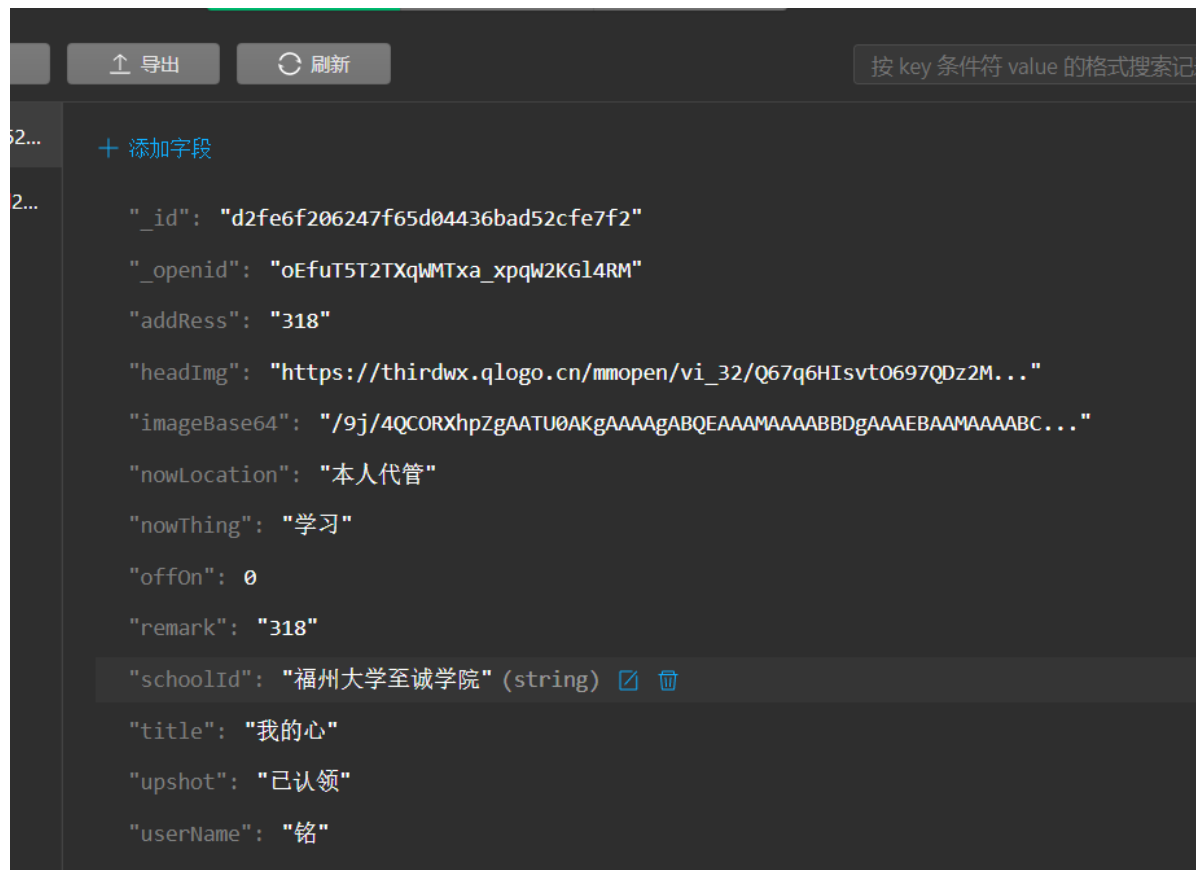
1. 系统通过用户所授权的微信账号来进行登录，保证了账号的唯一性。
2. 管理员为最高的用户权限，可以对用户所发布的信息进行审查。
3. 具体的系统配置将根据系统实际运行情况做进一步的调整。

5.3 数据库实施

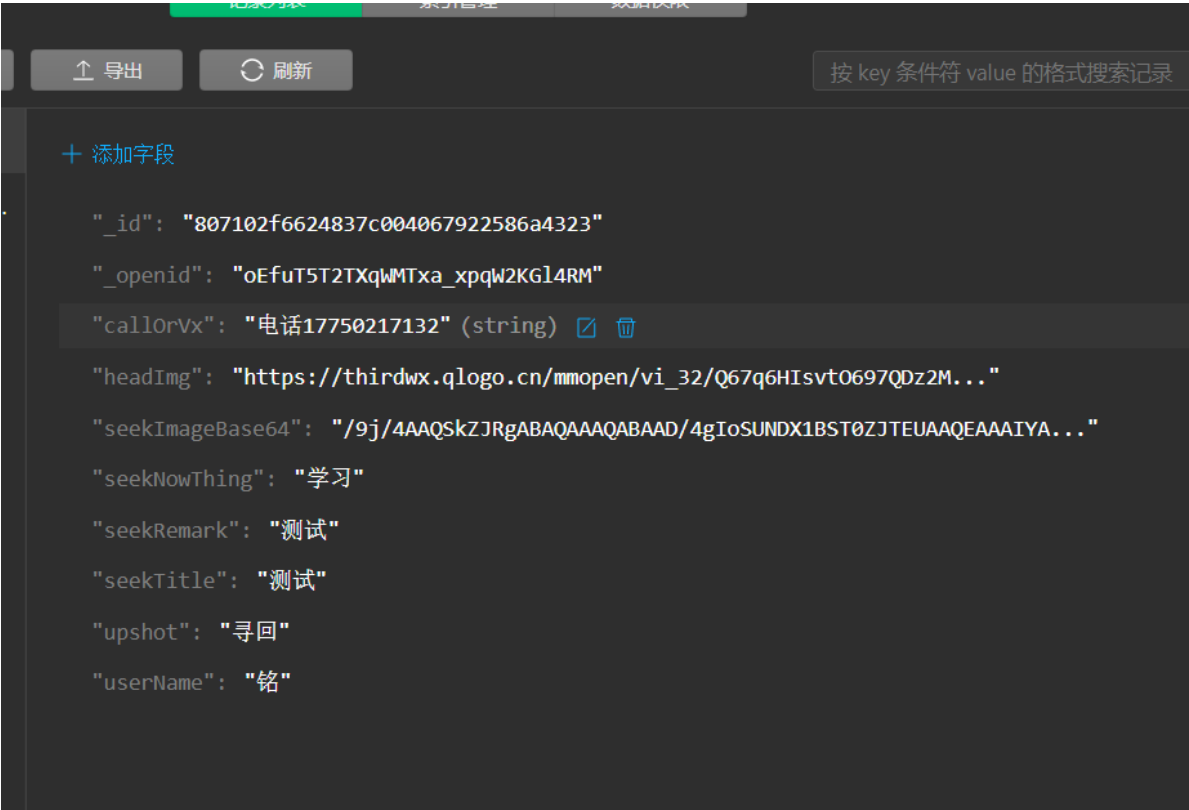
5.3.1 user



5.3.2 loseThing



5.3.3seekThing



六、数据库验证验收标准

功能	验收标准
数据的接受、更新	能够接受用户发布的寻物启事和失物招领的信息
数据的查询	能够看到他人和自己所有的寻物启事和失物招领的信息