

# 一. 引言

## 1.1 编写目的

本系统设计说明书是根据《i寻系统需求分析报告》所阐述的需求分析的基础上做出初步的系统设计。主要面向该系统的软件开发人员、设计人员以及用户，用于描述系统的功能层次和整体结构设计，也是最终验收产品的根据。

## 1.2 项目背景

软件名称：i寻  
开发者：福州大学至诚学院2021级软件工程专升本“薪火队”  
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，聚少成多，必然能够提高失物招领的效率。

## 1.3 参考资料

- [1] 《GB9385-2008 计算机软件需求规格说明规范》[S].2008
- [2] 邹欣.《构建之法(第三版)》[M].人民邮电出版社

## 1.4 用户

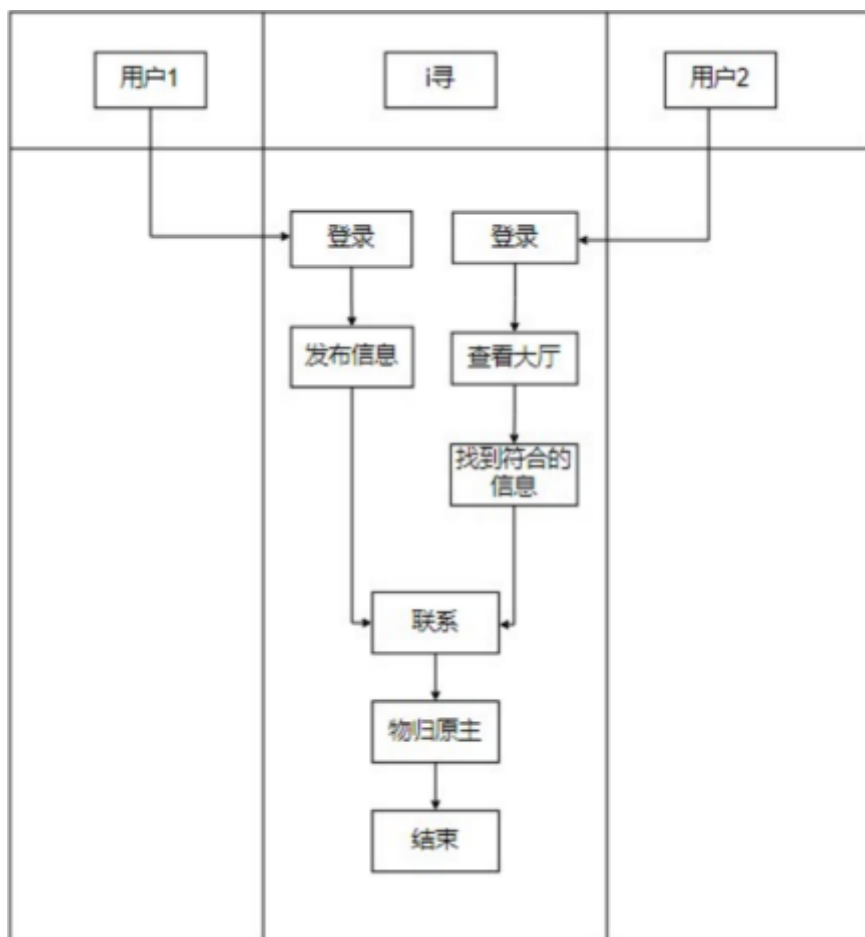
本产品现阶段主要面对福州大学至诚学院的工作人员，教师和学子。通过用户的反馈和后续的改进再进行进一步推广。

## 1.5 运行环境

存储空间5G、数据库容量2G/月的服务器兼容于已安装微信的手机

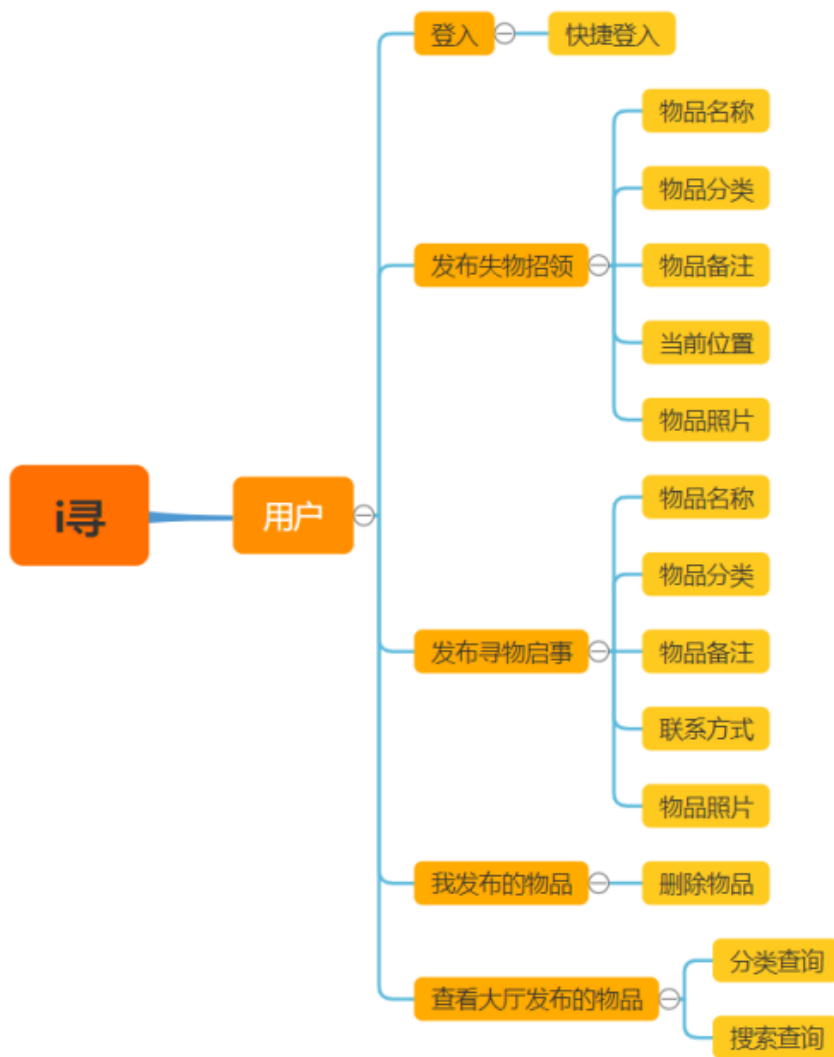
# 二. 总体系统设计

## 2.1 运行流畅图



## 2.2功能设计

### 2.2.1功能模块层次设计图



### 2.2.2功能模块说明

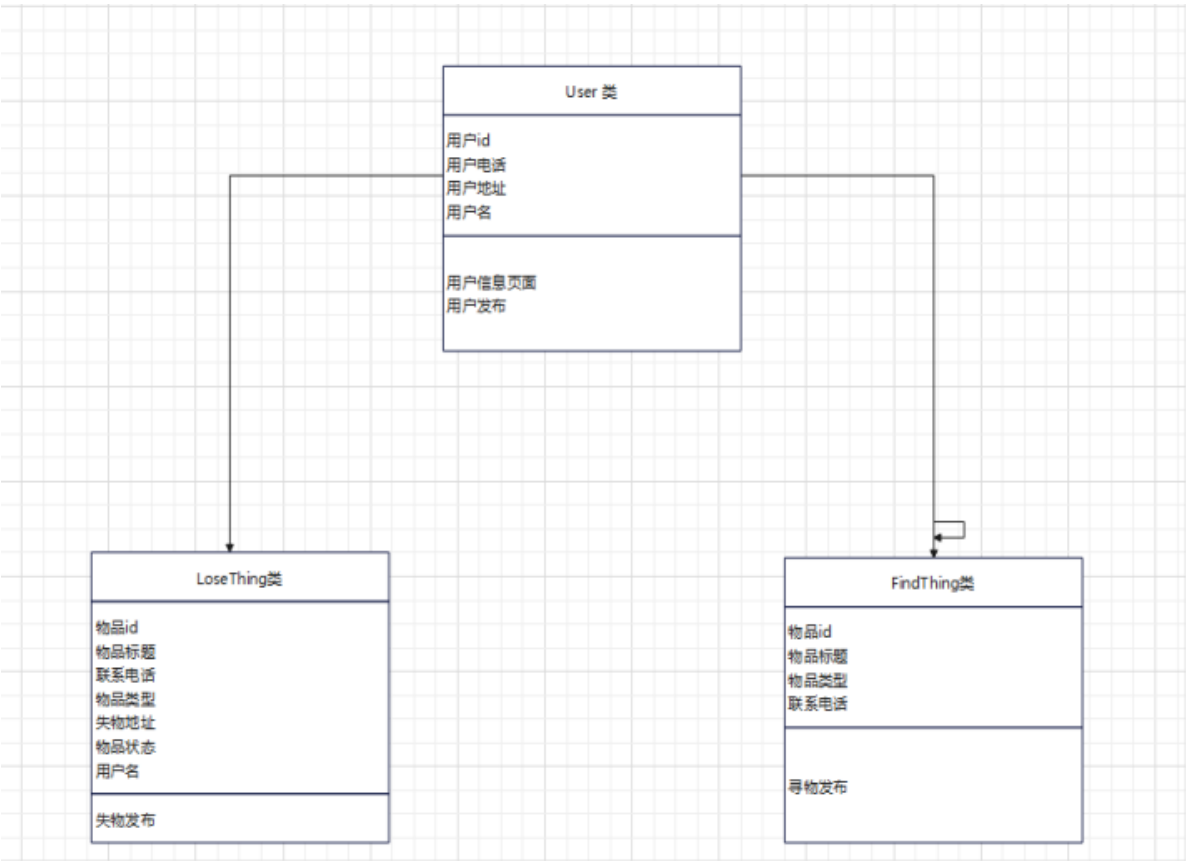
- 1.登录模块 用户可以在用户页面直接使用微信账号快捷登录。
- 2.发布失物招领模块 用户可以在首页页面填写丢失物品的详细信息和地址并发布失物招领。
- 3.发布寻物启事模块 用户可以在首页页面填写捡到物品的详细信息和地址并发布寻物启事。
- 4.我发布的物品模块 用户可以在用户页面中查看和删除自己发布的失物招领和寻物启事。
- 5.大厅查看发布的物品模块 用户可以在大厅页面查看他人发布的失物招领和寻物启事，也可以通过分类查询和搜索查询查看物品。

### 2.2.3功能设计思路

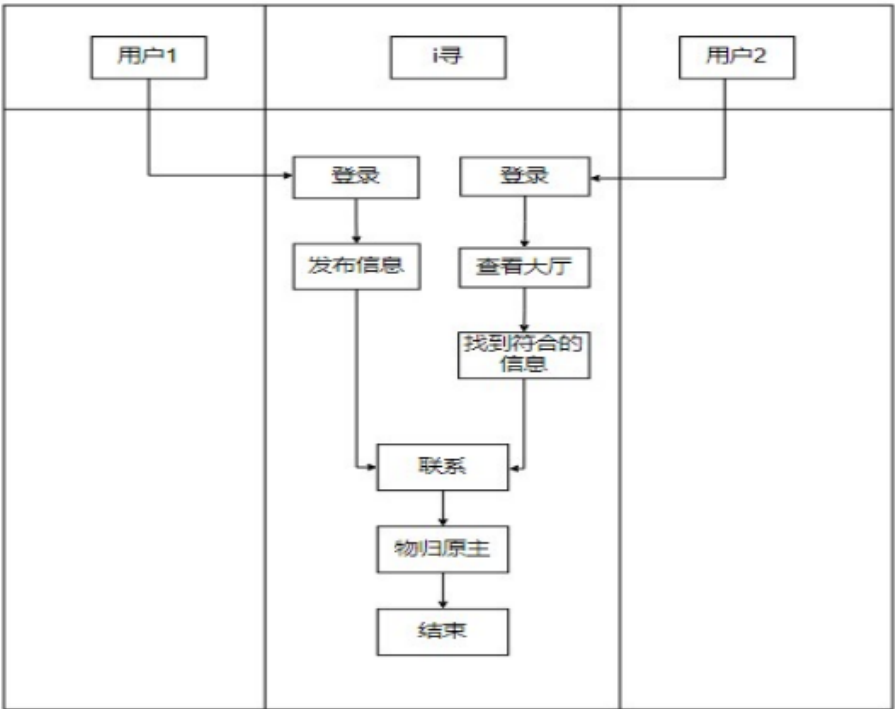
- 1.登入模块 卖家必须登入才可以发布和认领。
- 2.发布失物招领模块 必须填写满所有信息才可以发布。
- 3.发布寻物启事模块 必须填写满所有信息才可以发布。
- 4.大厅查看发布的物品模块 对物品进行分类，方便用户的快捷查询。

## 三.uml设计

### 3.1 类图



### 3.2 活动图



## 四.接口设计

接口名	接口说明	请求方法	请求数据	返回数据
api/good/losing/add	添加寻物	put	id	id
api/good/losing/delete	删除寻物	delete	id	id

接口名	接口说明	请求方法	请求数据	返回数据
api/good/seeking/delete	删除失物	delete	id	id
api/good/seeking/delete	添加寻物	put	id	id

# 五.ER分析与表结构设计

## 5.1ER分析



此小程序有用户表，寻物启事表，失物招领表。用户与寻物启事的关系为一对多，用户与失物招领的关系为一对多。

用户（用户id，记录id，头像图片，电话，学校，用户名，地址），用户id为主键

寻物启事（用户id，记录id，用户名，电话，头像图片，物品图片，标签，标题，详细信息，状态）主键为记录id

失物招领（用户id，记录id，学校，地址，代管情况，用户名，电话，头像图片，物品图片，标签，标题，详细信息，状态）主键为记录id

## 5.2表结构设计

User表

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"/>			

Losing表

对象 losething @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"/>			
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"/>			
schoolId	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"/>			

Seeking表

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"/>			

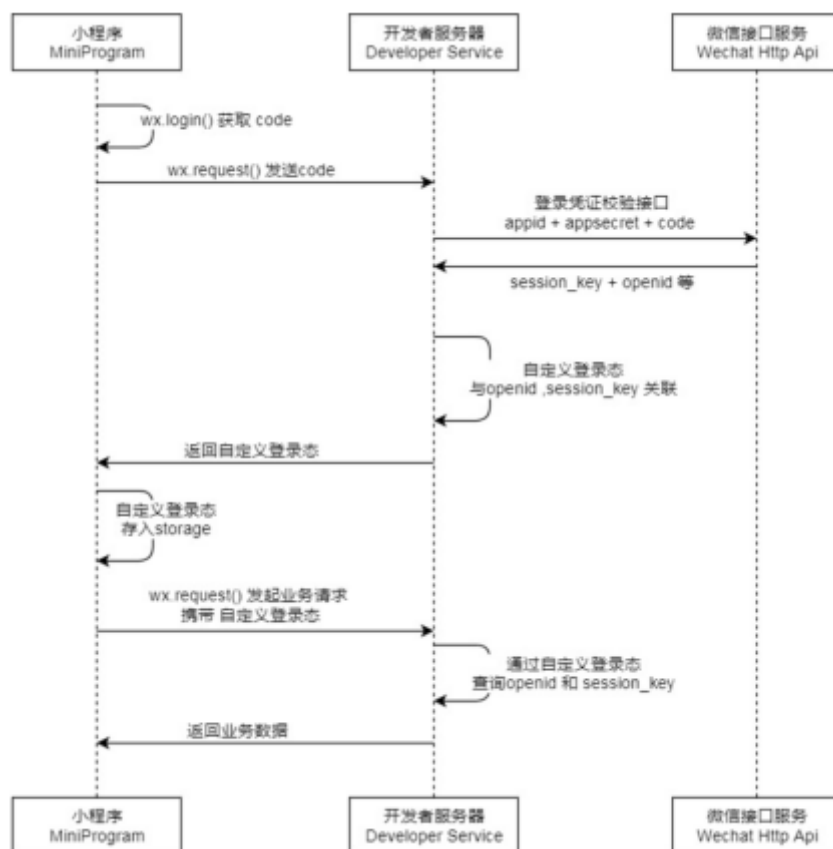
## 六. 系统安全

### 6.1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。

### 6.2登入安全设计

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。



## 6.3数据安全设计

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。