# 1、总览

项目开发文档这里主要分两部分：

* 数据字典
* 可重用类、方法及组件说明

它们的用处如下：

* 用于“丰富”最后提交的项目文档
* 用于合作时方便，相互调用接口、组件

首先切记不要误点到删除，因为这些文档我一般不定期进行本地备份，误删可能好多东西就真没了。

下面是对这两部分阅读方法的详细说明

### 数据字典

* 主要描述数据库中各表结构及对应字段的作用，编程中涉及对数据库的操作时要严格按照对应字段的作用和注释来进行；
* 对于由后端服务器进行维护的字段，不要显式地直接通过客户端代码进行随意更改、增删；
* 对于有默认值的字段，要能用默认值就用默认值，能省就省；
* 对于某些取唯一值的字段或字段组（这里可近似当做主键），在前端调用接口进行新记录插入时一定要先查询验证无重复，再插入。不要试图根据插入是否成功来判断有无重复，因为只要后端报错，后端接口统一就认为是服务器出错或者查询语句语法错误。约束性错误就交给前端自己解决。
* 部分字段的注释或者其他方面可能有谬误、表意不明或者有需要添加的字段，看的时候发现了可以直接更改，但记得在群里说一声。

### 可重用类、方法及组件说明

* 在写代码的时候一定会遇到一些类、方法或者自己通过代码xml文件等实现的特效等，感觉可以提取出来重用（比如搜索算法、数据库接口、自适应屏幕宽度的UI组件等），一定不要偷懒，及时提取出来（最好能测试两下），然后及时更新对应文档，这个工作量是必要的。具体格式没有统一要求，可以参考前面的同类文档，也可以加一些内容，总之要做到文档简介，而且能让其他人看了直接会用。
* 经其他人指出错误后，要及时修正源码，并且更新对应文档，一定要保持文档与源码的一致性。

# 2、Android Studio调试说明

* 构建报错在build窗口里看，运行报错在logcat窗口里看
* 不要惧怕英文，要学会用词典
* 报错优先看红字，报错内容优先看标题，异常trace优先看最顶端的异常类型
* 学会断点调试和Log方法+locat过滤器直观打印输出结果来测试，必要时新创一个测试activity加个button进行单元测试

# 3、代码书写规范

* 不要试图等到代码写完再整理规范，那是不可能做到的。写的时候就要注意规范，出现错误意识到了要立马修正，写完一小段要习惯性地扫一眼检查一下，顺便就把“尬眼”的代码就改了。
* java类名（文件名）、方法名、变量名一律采用标准Java驼峰命名法（忘记的赶紧查），后面不希望看到“尬眼的”、“诡异的”命名风格出现
* 增删代码后，剩余代码缩进要一并调好，不要出现“诡异的”缩进（这也对写python有巨大好处)
* 命名不准出现汉语拼音（除非查过词典，找不到很好的翻译）,生僻词一律写全，非生僻词可以缩写（比如image可以缩写成img)，对于含义过于复杂的变量，命名也要尽量贴切（比如标识一种“比率”至少得命名成“rate”），然后再加上注释进行解释。不准出现"a,b,c,x1,x2"这类毫无意义的命名
* xml文件名一律小写，单词间用下划线分隔，“activity"这个词要提到最前面（其实在创建activity的时候用下拉选项中的activity创建，把java命名好了生成的xml文件的命名也自动就是规范的了
* xml中组件id命名遵循java变量命名方法
* 样式xml文件一律放进'values'文件夹内
* 时刻留意代码的重用性、封装性，及时重构。不要“一个方法写到底把所有功能都实现”，这样没有逻辑层次、可读性差、也不易重用。实在觉得没必要分成多个方法或者进行抽象的代码块也要空行区分，并分模块注释
* 对于稍复杂的逻辑记得及时注释，这一切都是值得的，能够加快看这段代码的阅读速度也能方便测试时检查到底该模块有没有实现它应有的功能
* 除了少部分确实很可能会重新采用的代码，大多数注释掉的代码都是无用代码，及时删掉带来可读性提升是更值得的。
* 除了代码块的重用性还要留意变量的重用性，适当而及时地提升变量的作用域对理解代码结构、精简代码有很大帮助
* 除了测试，尽量避免使用常量，无论是xml属性值还是java代码。
* 代码不能只追求“功能实现”，一定要“符合逻辑结构”、“方便扩展”，这样测试或者重用才不会受阻。
* 类成员尽量使用private，尤其类变量，一律使用setter和getter方法来更新，在Android Studio中Alt+Insert快捷键可以快速生成代码模板
* 书写可重用方法时一定要记得多重载来减少参数，以便能够让其他使用者更简单地使用你的方法
* 涉及异步加载一律使用Thread.join来转化为同步

# 4、文档书写说明

* 该在线工具基本上是markdown语法，熟悉的可以全程不用鼠标写出一个布局良好的文档，不熟悉的可以看帮助，或者用鼠标点内容区上方的功能按钮来生成对应的样式代码（用几次记住了就直接敲出来了）
* 最外层新模块基本就不轻易变动，数据字典这个模块主要有何凌飞维护。而“可重用类、方法及组件说明”的模块由所有编程人员共同维护
* 如果其中新类要新创文件夹作为“可重用类、方法及组件说明”的子文件夹
* 一个方法一个文档，一个组件（包括xml和java的用法说明）一个文档，每个类文件夹下的第一个文档最好是“**类说明文档**”，简要说明下该类主要用来干什么
* 写开发文档的原则是“可重用”或者“需要给别人看”的才需要写文档，其他的“感觉基本确定只会用一次”，或者“从头到尾只有自己会用上”的不用写文档，实在怕后面维护测试的时候忘记了，直接在代码中加注释就好了。
* 开发文档写的时候顺便就把布局排版什么的调好，千万不要看上去凌乱或随意，不然就失去了写文档的初衷——让其他人“友好”、“舒适”地理解你的东西。
* 具体如何创建目录、文件、模板什么的，随便点点，看看帮助、百度两下showdoc用法基本就解决了。
* 必要时要加上图文说明，不要认为这是无用功，做的好，最后交作业可以直接把开发文档中的部分粘贴上去。

# 5、数据库-数据字典

## 5.1、users

* 用户表，储存用户信息

| **字段** | **类型** | **空** | **唯一** | **默认** | **注释** |
| --- | --- | --- | --- | --- | --- |
| telephone | varchar(10) | 否 | 是 |  | 登录账号 |
| id | varchar(225) | 否 | 是 |  | （供其他用户搜索的）用户标识，必须由后端生成填写 |
| password | varchar(50) | 否 | 否 |  | 登录密码(需检验：不含非数字、字母的字符，且不少于8位，且首字符必须为字母) |
| nick\_name | varchar(25) | 否 | 否 |  | 昵称 |
| head\_img | varchar(255) | 否 | 否 | <http://www.blacklighter.cn/TuCaoDict/imgs/head/normal.jpg> | 用户头像。客户端需调用相关方法上传图片，由后端对该字段进行更新 |
| gender | varchar(255) | 否 | 否 | '保密' | 性别。取值只能为{'男','女','保密'} |
| identity | varchar(255) | 否 | 否 | '普通用户' | 用户身份。取值只能为{'普通用户','特权用户','管理员'} |
| like\_num | int | 否 | 否 | 0 | 被赞数 |

## 5.2、entry\_kinds

* 词条类别表，储存目前词条所有的类别

| **字段** | **类型** | **空** | **唯一** | **默认** | **注释** |
| --- | --- | --- | --- | --- | --- |
| kind\_name | varchar(255) | 否 | 是 |  | 类别名 |
| kind\_img | varchar(225) | 否 | 否 |  | 类别图片。 客户端需调用相关的方法上传图片，由后端自动填写该字段 |

## 5.3、entry\_overview

* 词条总览表，储存词条总览信息

| **字段** | **类型** | **空** | **唯一** | **默认** | **注释** |
| --- | --- | --- | --- | --- | --- |
| entry\_name | varchar(255) | 否 | 是 |  | 词条名 |
| kind\_names | varchar(225) | 否 | 否 |  | 词条所属的所有分类名。需管理员审核，多个分类名间以单空格分隔 |
| telephone | varchar(255) | 否 | 否 |  | 词条创建人手机号 |
| hot\_num | int | 否 | 否 | 0 | 词条热度值 |
| state | varchar(255) | 否 | 否 | '待审核' | 词条状态。取值只能为{'待审核','通过','未通过'} |

## 5.4、history

* 浏览记录表，储存用户的词条浏览记录

| **字段** | **类型** | **空** | **唯一** | **默认** | **注释** |
| --- | --- | --- | --- | --- | --- |
| telephone | varchar(255) | 否 | 与entry\_name组合起来唯一 |  | 浏览者手机号 |
| entry\_name | timestamp | 否 | 与telephone组合起来唯一 |  | 浏览的词条名 |
| timestamp | varchar(255) | 否 | 否 | insert时的服务器时间 | 最近一次浏览某词条的服务器时间戳。调用相应方法，由后端填写、更新 |

## 5.5、collection

* 词条收藏表，储存所有用户的收藏记录

| **字段** | **类型** | **空** | **唯一** | **默认** | **注释** |
| --- | --- | --- | --- | --- | --- |
| telephone | varchar(255) | 否 | 与entry\_name构成的组合键唯一 |  | 收藏者手机号 |
| entry\_name | varchar(225) | 否 | 与telephone构成的组合键唯一 |  | 收藏的词条名 |

## 5.6、information

* 消息表，储存所有系统信息

| **字段** | **类型** | **空** | **唯一** | **默认** | **注释** |
| --- | --- | --- | --- | --- | --- |
| sender\_tele | varchar(255) | 否 | 与receiver\_tele、timestamp组合起来唯一 |  | 发送方手机号 |
| receiver\_tele | varchar(255) | 否 | 与sender\_tele、timestamp组合起来唯一 |  | 接收方手机号 |
| timestamp | varchar(225) | 否 | 与sender\_tele、receiver\_tele组合起来唯一 | insert时服务器时间戳 | 消息生成时的时间戳。只能由后端生成填写 |
| kind | varchar(255) | 否 | 否 | '系统通知' | 消息类别。取值目前只能为'系统通知' |
| heading | heading | 否 | 否 |  | 消息标题 |
| content | varchar(255) | 否 | 否 | '待审核' | 系统信息主体内容 |
| addition | varchar(255) | 是 | 否 | 额外说明 |  |
| state | varchar(255) | 否 | 否 | ‘未查看’ | 消息状态。取值目前只能为{‘已查看’，‘未查看’} |

## 5.7、module\_kinds

* 模块信息表，储存所有词条的所有模块的信息

| **字段** | **类型** | **空** | **唯一** | **默认** | **注释** |
| --- | --- | --- | --- | --- | --- |
| kind\_name | varchar(255) | 否 | 是 |  | 模块名 |
| kind\_formats | varchar(225) | 否 | 否 | 'text' | 模块允许的内容格式。取值为{'text','img','music','video'}中的一个或多个，多个时用单空格作为分隔符 |

## 5.8、modules

* 模块详情表，储存所有用户提交的所有词条的所有模块的具体内容

| **字段** | **类型** | **空** | **唯一** | **默认** | **注释** |
| --- | --- | --- | --- | --- | --- |
| entry\_name | varchar(255) | 否 | 与module\_kind，telephone组合起来唯一 |  | 词条名 |
| module\_kind | varchar(225) | 否 | 与entry\_name，entry\_name组合起来唯一 |  | 模块类别名 |
| telephone | varchar(255) | 否 | 与entry\_name，entry\_name组合起来唯一 |  | 模块内容提交者手机号 |
| timestamp | timestamp | 否 | 否 | 最近一次更改时的服务器时间戳 | 最近一次更改时的服务器时间戳。由服务器自动插入、更新 |
| content | varchar(255) | 否 | 否 | 模块具体内容。内容的格式在客户端根据module\_kinds进行检验后，方可插入、更新 |  |
| like\_num | int | 否 | 否 | 0 | 获赞数 |
| dislike\_num | int | 否 | 否 | 0 | 获踩数 |

# 6、可重用类、方法及组件说明