**休闲生活**

**软件设计说明书**

# 引言

## 编写目的

通过梳理软件需求，明确软件需要实现的功能后，需要通过软件设计来确定软件实现方案，从而使开发人员能够按照方案进行开发，从而实现相关功能。

## 术语表

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | 接口 | 对协定进行定义的[引用类型](https://baike.baidu.com/item/%E5%BC%95%E7%94%A8%E7%B1%BB%E5%9E%8B/2889743) |
| 2 | 框架 | 对组成系统的各部分进行搭配和安排 |
| 3 | JSON | 一种轻量级的数据交换格式 |

# 系统设计

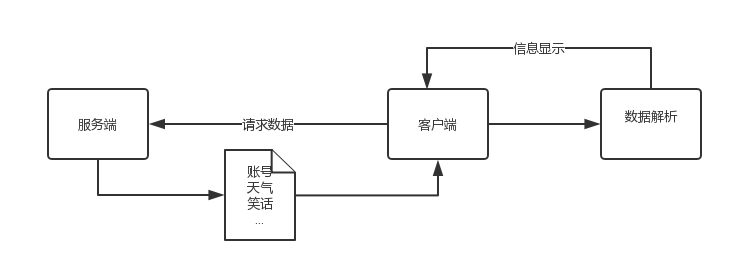


图1

如图1所示，客户端通过网络请求获取账号、天气、笑话等信息，再将得到的信息通过数据解析模块进行数据格式化，最后客户端得到其能够识别的数据格式，并将相应的信息显示出来。

# 模块设计

## 网络请求模块

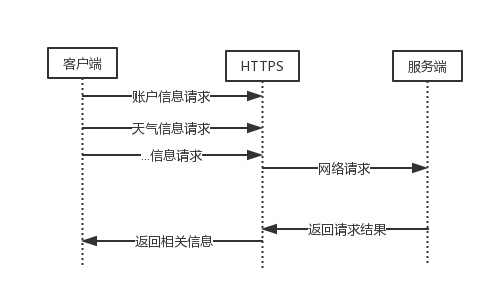


图2

如图2所示，其为网络请求模块处理流程

1. 客户端通过HTTS发起账号、天气等信息请求
2. HTTPS通过网络连接服务端进行请求
3. 服务端响应请求，返回相关信息

## 数据解析模块

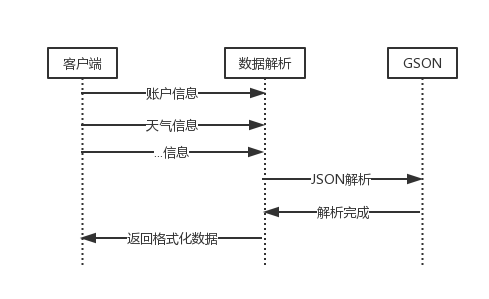


图3

如图3所示，其为数据解析模块处理流程

1. 客户端将账号、天气等信息发送给解析模块进行解析
2. 解析模块调用GSON框架进行JSON数据格式解析
3. GSON解析完成，返回相应的格式化数据

## 数据展示模块

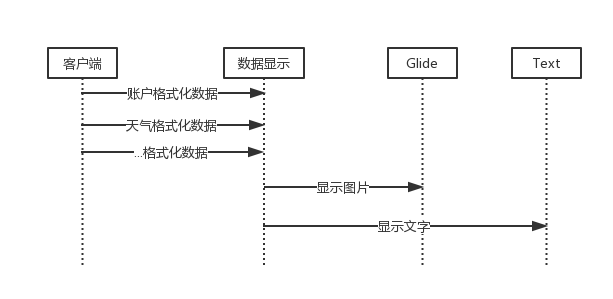


图4

如图4所示，其为数据展示模块处理流程

1. 客户端将账号、天气等格式化数据发送给展示模块进行展示
2. 展示模块调用Glide框架进行图片处理，并把处理后的图片显示出来
3. 展示模块调用Text进行文件处理，并把处理后的文字显示出来

# 接口设计

## 账户处理接口

public Observable<UserInfoResponseBean> register(String user\_operator, String user\_info)

接口名称: register

接口参数: user\_operator账户操作类型(0创建账户、1登录、2注销)

user\_info 用户信息

返回值 : UserInfoResponseBean账户信息格式化数据

## 获取城市列表接口

public Observable<CityInfoListResponseBean> getCityInfoList()

接口名称: getCityInfoList

接口参数: 无

返回值 : CityInfoListResponseBean城市列表信息格式化数据

## 获取天气接口

public Observable<TodayWeatherResponseBean> getZHTianQiByCity(String city)

接口名称: getZHTianQiByCity

接口参数: city城市名称

返回值 : TodayWeatherResponseBean天气信息格式化数据

## 获取笑话接口

public Observable<QSBKElementList> getQSBK(int page, String user\_id)

接口名称: getQSBK

接口参数: page获取笑话的起始页

user\_id账户ID

返回值 : QSBKElementList笑话列表信息格式化数据

# 数据结构

## 账户信息UserInfoResponseBean

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **描述** |
| 1 | user\_id | 用户ID |
| 2 | user\_name | 用户名 |
| 3 | pass\_word | 密码 |
| 4 | phone\_number | 手机号 |
| 5 | nick\_name | 昵称 |
| 6 | sex | 性别 |
| 7 | birthday | 生日 |

## 城市列表信息CityInfoListResponseBean

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **描述** |
| 1 | province | 省名称 |
| 2 | city\_code | 城市编码 |
| 3 | city\_name | 城市名称 |
| 4 | city\_py | 城市拼音 |
| 5 | city\_head | 城市拼音首字母 |

## 天气信息TodayWeatherResponseBean

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **描述** |
| 1 | address\_info | 地址信息 |
| 2 | city\_name | 城市名称 |
| 3 | today\_weather | 天气总述 |
| 4 | today\_weather\_detail | 天气详情 |

## 天气总述信息TodayWeather

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **描述** |
| 1 | now\_time | 当前时间 |
| 2 | temperature | 温度 |
| 3 | wind\_direction | 风向 |
| 4 | air\_quality | 天气质量 |
| 5 | humidity | 湿度 |
| 6 | wind\_value | 风力 |
| 7 | limit\_content | 限行描述 |
| 8 | simple\_content | 天气总述 |
| 9 | is\_w | 是否有风向、风力值 |
| 10 | is\_h | 是否有湿度值 |
| 11 | is\_pol | 是否有空气质量值 |
| 12 | is\_limit | 是否有限行描述 |
| 13 | temperature\_icon\_css | 温度图标样式 |
| 14 | humidity\_icon\_css | 湿度图标样式 |
| 15 | wind\_icon\_css | 风向图标样式 |
| 16 | air\_quality\_icon\_css | 天气质量图标样式 |
| 17 | limit\_icon\_css | 限行描述图标样式 |

## 白天、夜晚天气详情信息TodayWeatherDetail

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **描述** |
| 1 | temperature | 温度 |
| 2 | title | 白天、夜晚 |
| 3 | wind\_direction | 风向 |
| 4 | weather\_desc | 天气描述 |
| 5 | weather | 天气 |
| 6 | sun\_time | 日出、日落 |
| 7 | wind\_value | 风力值 |
| 8 | weather\_icon\_css | 天气图标样式 |
| 9 | wind\_icon\_css | 风力图标样式 |
| 10 | sun\_icon\_css | 日出、日落图标样式 |

## 笑话信息QSBKElementList

public int current\_page = 0;

public int total\_page = 0;

public List<QSBKElement> result = new ArrayList<>();

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **描述** |
| 1 | current\_page | 当前页 |
| 2 | total\_page | 总页数 |
| 3 | result | 笑话信息列表 |

## 笑话详情信息QSBKElement

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **描述** |
| 1 | author\_id | 笑话ID |
| 2 | author\_sex | 作者性别 |
| 3 | thumb | 笑话图片 |
| 4 | author\_age | 作者年龄 |
| 5 | has\_thumb | 是否有笑话图片 |
| 6 | comment\_number | 评论数 |
| 7 | author\_name | 作者名称 |
| 8 | content | 笑话内容 |
| 9 | author\_head\_img | 作者头像 |
| 10 | is\_anonymity | 作者信息是否开放 |
| 10 | vote\_number | 好笑数 |
| 10 | is\_collect | 是否已收藏 |