**基于Android的论坛社交APP**

1. **系统的需求分析**

是开发人员经过深入细致的调研和分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求表述转化为完整的需求定义，从而确定系统必须做什么的过程。该部分的主要工作是确定系统必须完成哪些目标，也就是要对目标系统提出准确、清晰、具体、完整的要求。

**1.1系统目标**

采用Java编程语言和使用Android平台以及与数据库技术相结合，开发出一个具有良好用户界面，功能齐全，运行稳定，交互性良好的移动轻社交软件。本软件的定位为:社交，类似百度贴吧的功能。

**1.2功能需求**

开发一款软件，用户分析是必不可少的一个流程，因为软件最终面向的是用户，是需要用户来使用的，所以要从用户的角度分析，而用户的需求一般都有一些共同点：

界面友好：一款软件如果操作界面不友好，很容易让用户失去耐心，导致用户的流失。所以良好的交互界面是本软件的一个重要部分，这样可以增加用户黏性。

操作简便:用户很难愿意花费大量时间去学习如何使用一款软件，而且用户不是开发人员，程序的内部逻辑用户是不知道的，所以开发人员必须提供给一些封装好的便捷的操作接口来给用户使用。

运行稳定:流畅软件有可能在运行时出现不可预料的错误从而造成软件卡顿，崩溃。一款设计良好的软件必须避免这种情况，要避免用户体验差，造成一些用户对软件失去信心从而造成用户流失。所以软件的健壮性要强，要尽可能的将错误捕获，后台分析处理，一切都要站在用户的角度来看问题。

为了方便校内同学的交流，我们决定要做这么一个校园论坛，而对于论坛这样的数据流量特别大的网络管理系统， 必须要满足使用方便、操作灵活等设计需求。所以本系统在设计时应满足以下几个目标：

1.临时用户进入,可浏览帖子但不可发帖

2. 一个版面能显示所有的帖子

3. 具有登录模块，有用户的个人信息

4. 用户随时都可以查看自己发表的帖子

5. 有用户的消息的时间及时提醒，主要用于提示哪个用户 回复了自己的主题。 6.管理员权限可删除任意帖子，具有最大权限的管理功能

7. 对用户输入的数据，系统进行严格的数据检验，尽可能 排除人为的错误

8. 系统最大限度地实现了易维护性和易操作性

9. 系统运行稳定安全可靠

1. **系统的概要设计**

总体设计是开发软件时，根据需求分析和可行性论证，对系统进行整体设计,为系统确定整体框架结构的过程。本应用为一款移动社交应用，核心内容为社交，总体设计围绕着核心内容，具体分为：

核心功能：登录注册，状态圈，消息中心

次要功能：工具中心，上传下载中心，用户中心，设置中心

**2.1 概要设计**

本应用是一款完整的网络应用，包括客户端，服务器以及数据库，为了将模块功能细分，将其中的数据库部分又细分为客户端数据库(MYSQL)与服务区端数据库(SQLITE)，服务器端的实现也再次细分为Web服务器与Socket服务器。总体结构如图所示：

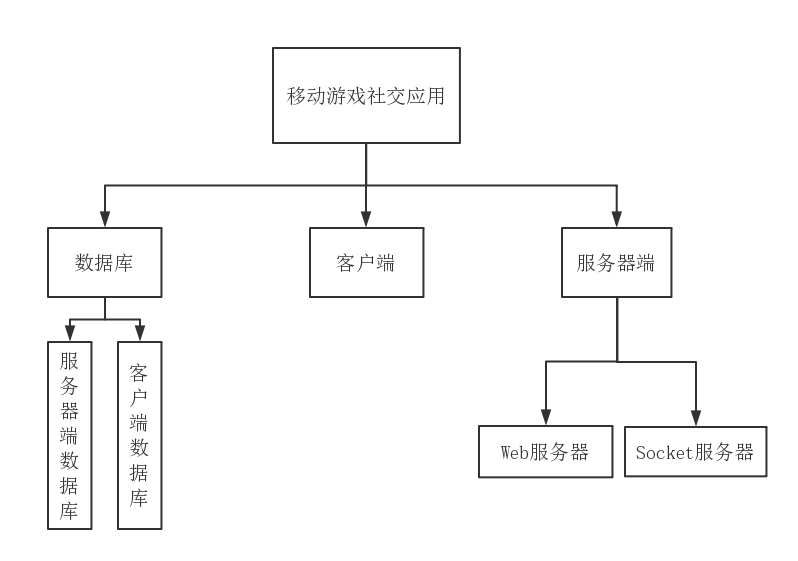


图2-1应用总架构图

**2.2 客户端的设计**

根据功能需求分析，本应用的客户端细分为九个模块，即登录注册，消息中心，下载上传中心，状态圈，工具中心，用户中心，设置中心，其系统总结构图如图所示。

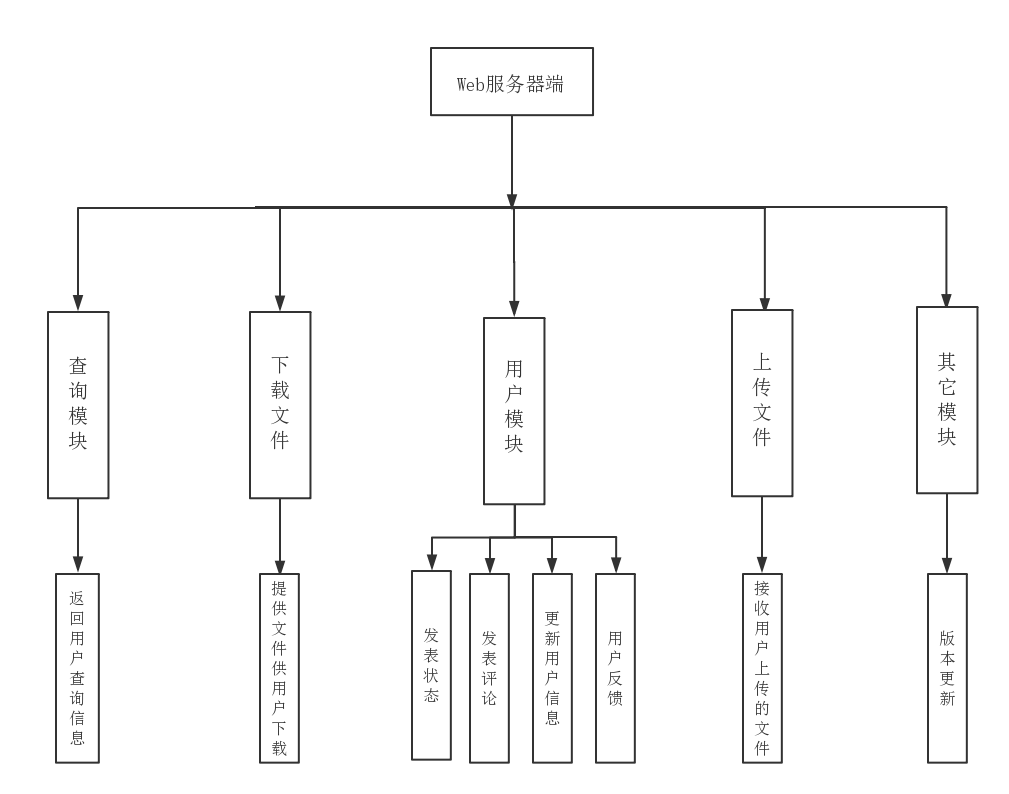
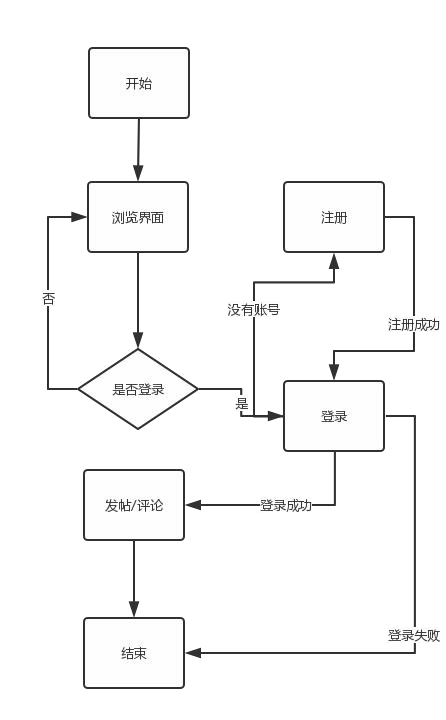


图2-2 ICG客户端架构图

**2.2.1 登录注册的设计**

登录注册模块要能够处理用户的登录，注册基本功能，还需要包括短信验证，密码找回等功能，其功能如图所示。

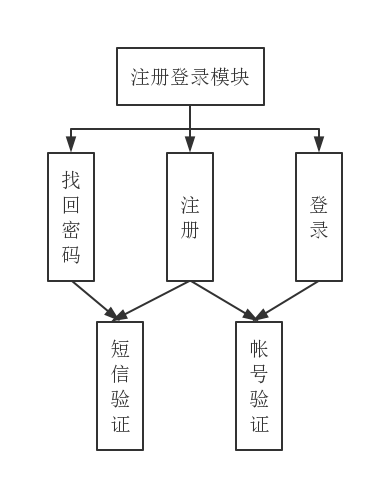


图2-2-1 登录注册模块功能图

**2.2.2 消息中心的设计**

消息中心模块可以查看系统消息或者好友发送的消息，可以查看好友状态以及随时向好友发送消息，可以随时与其它用户通信，包括匿名通信，实名通信，还可以随机寻友和按照需求信息匹配好友，以及包括摇一摇寻友和附近的人等功能．其功能如图所示。

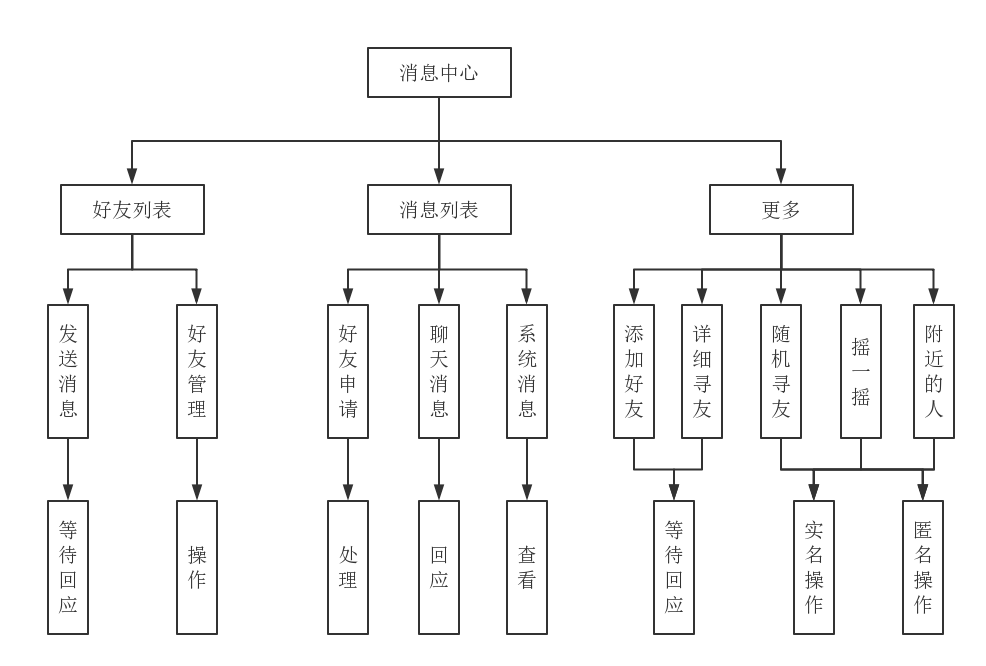


图2-2-2 消息中心功能图

**2.2.3 状态圈设计**

状态圈里可以查看自己或者朋友发的状态，包括发送状态，管理自己状态，点赞，评论等功能．其功能如图所示。

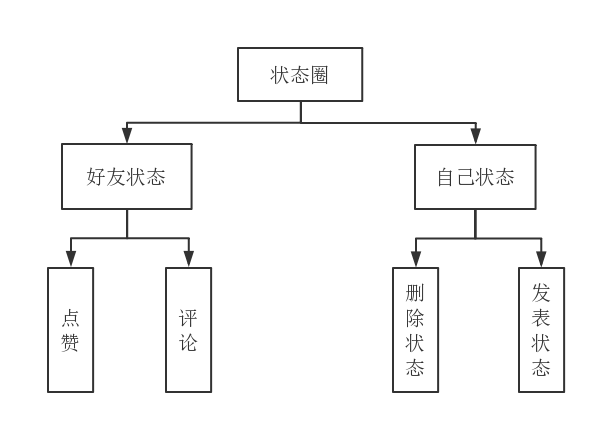


图2-2-3 状态圈功能图

**2.2.4 用户中心的设计**

用户中心包括一些用户会经常使用的功能，比如查看个人详细信息，修改用户资料，修改密码，反馈意见等功能，还包括摇一摇签到，升级会员等功能．其功能如图所示。

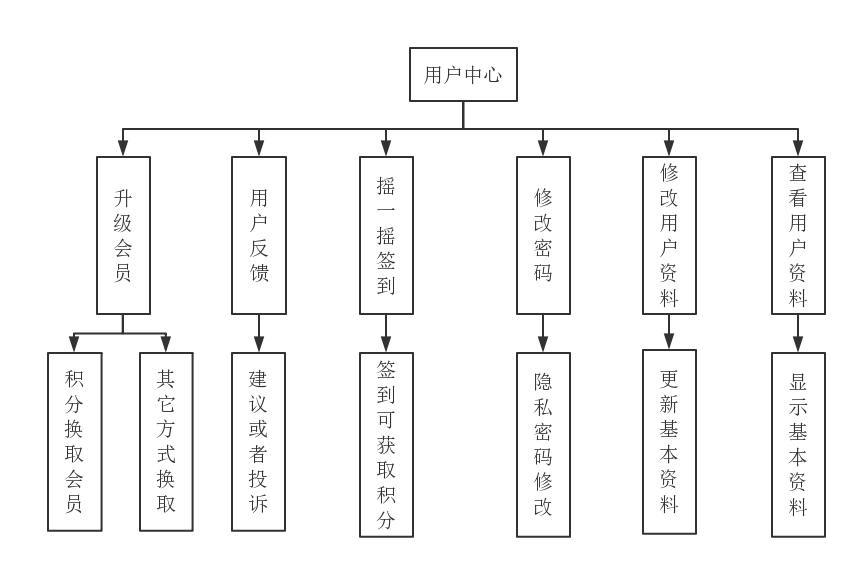


图2-2-4 用户中心功能图

**2.3数据库的设计**

一般在数据库设计中，最先设计的是概念模型，一般采用实体-联系图，即E-R图来描述概念模型，本应用中的实体联系图的实体主要包括用户，消息，定位信息，签到记录，反馈信息，状态日志，评论，上传信息，下载信息等实体。其中用户和消息是一对多的关系；用户和定位信息之间是一对多关系；用户和签到记录之间是一对多的关系；用户和用户反馈信息之间是一对多的关系；用户和状态日志是一对多的关系；用户和评论是一对多的关系；状态日志和评论是一对多的关系；用户和上传文件信息之间是一对多的关系；用户和下载信息是一对多的关系。实体联系图如图所示

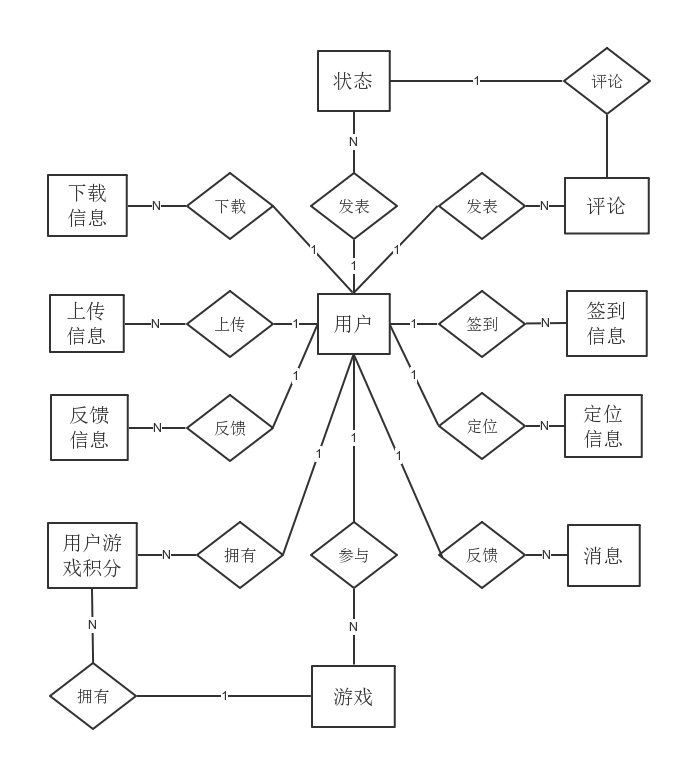


图2-3-1 E-R图

设计概念模型之后需要进行功能设计，本应用数据库的功能设计分为客户端端数据库部分和服务器数据库部分。

**2.3.1 服务器端数据库功能设计**

服务器端使用免费开源的MYSQL数据库，整个数据库分为几个模块，即用户模块(包括用户信息，好友关系，发状态，评论，签到，用户反馈)，消息模块(包括用户消息与系统消息)，，定位模块(包括用户的定位信息），其它模块(包括系统版本信息)，其系统总结构如图所示

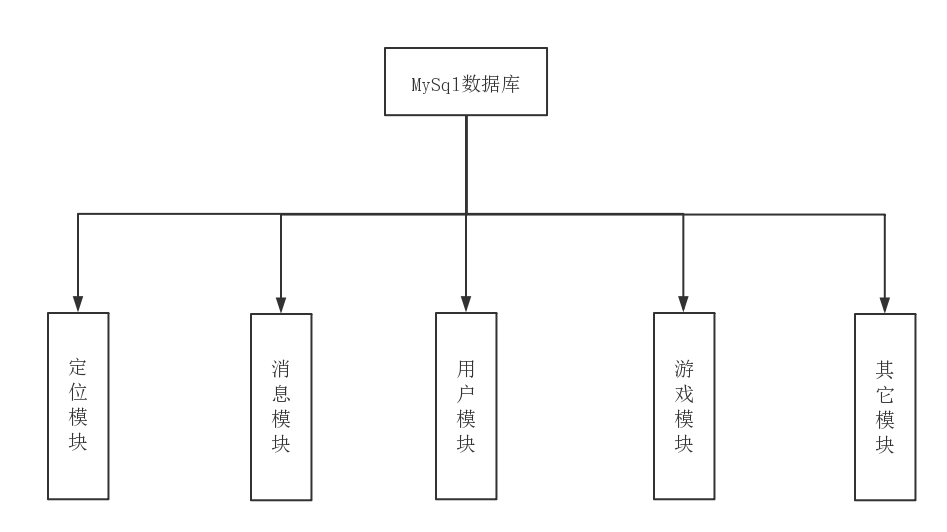


图2-3-1 MYSQL数据库总架构图

(1)用户模块设计

一个应用是围绕着用户来的，所以用户模块很重要，这里用户模块的结构如图所示．

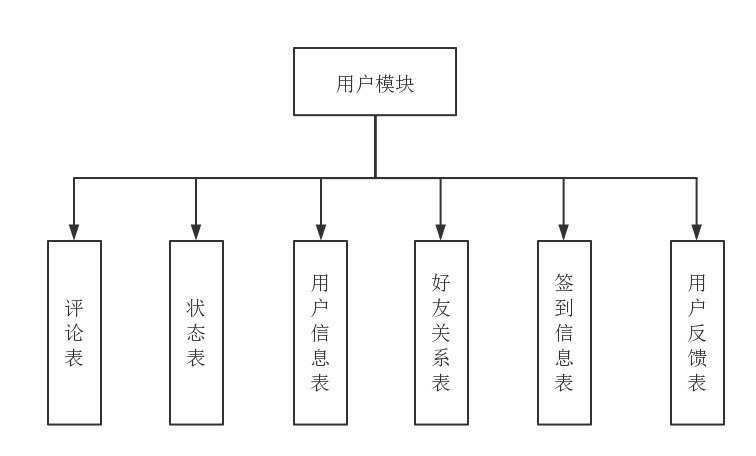


图2-3-1-1用户模块数据库架构图

(2)消息模块设计

消息模块目前主要是包括一个消息表，作用是当用户不在线时，用来存储系统和别的用户发送的离线消息，上线后再从消息表中获取未读取的消息．

**2.3.2 客户端数据库功能设计**

客户端使用Android自带的轻量级的Shared Preferences来保存一些常用配置，以及使用Android自带的SQLITE数据库来持久化一些数据。Shared Preferences，采用XML的轻量级存储方法，用来保存一些基本配置，例如皮肤选择路径，应用的设置信息等；SQLITE，用来持久化数据，主要分为几大模块，即用户信息模块(用来存储好友信息)，消息模块(包括持久化用户的聊天记录等功能)，上传下载记录模块(用来记录用户的上传下载信息，以及记录断点下载的进度和状态)，结构如图所示．

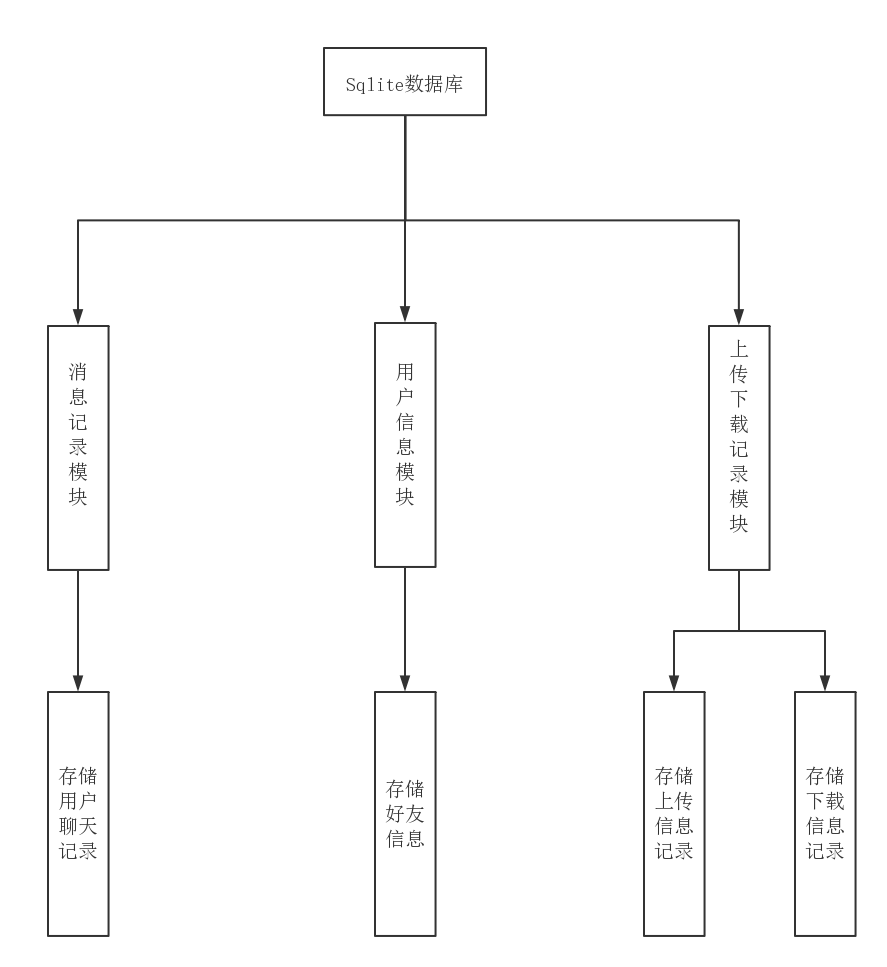


图2-3-2-1 SQLITE数据库架构图

1. **设计与实现部分**

**3.1 客户端设计**

3.1.1 登录注册模块

该模块主要负责处理用户的注册，登录，以及找回密码功能，处理流程如图所示．

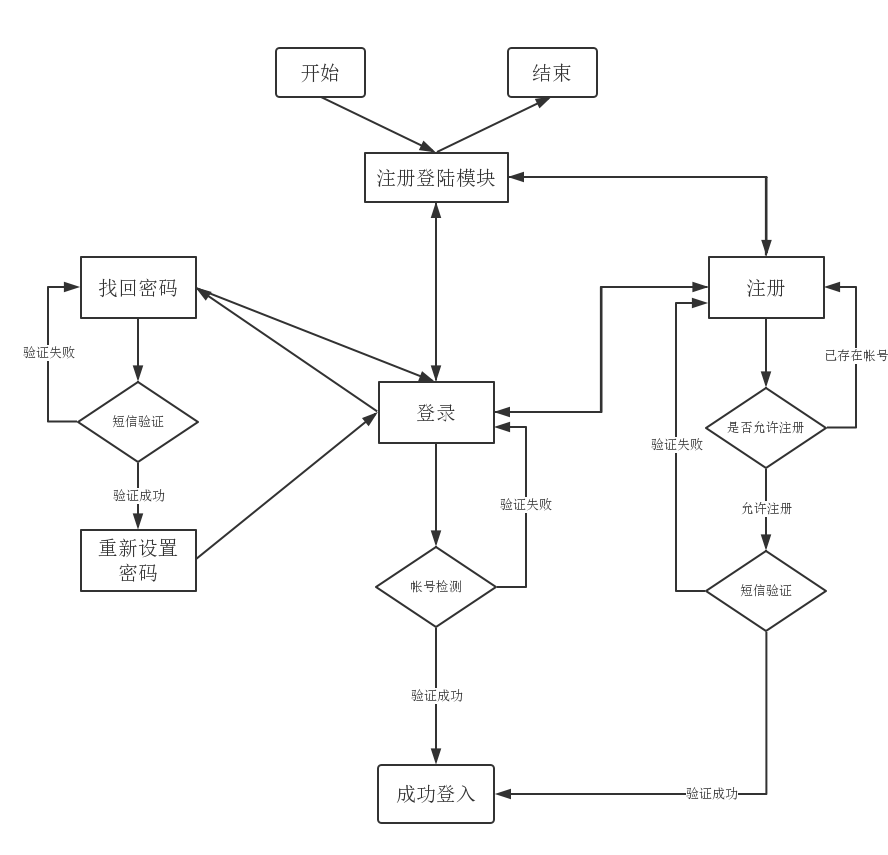


图3-1-1 登录注册模块流程图

3.1.2 消息中心模块

该模块主要处理即时通信，以及社交交友功能，包括添加好友，好友管理，处理好友申请，发送消息，查看系统消息，摇一摇寻友，以及查看附近的人等功能。处理流程如图所示．

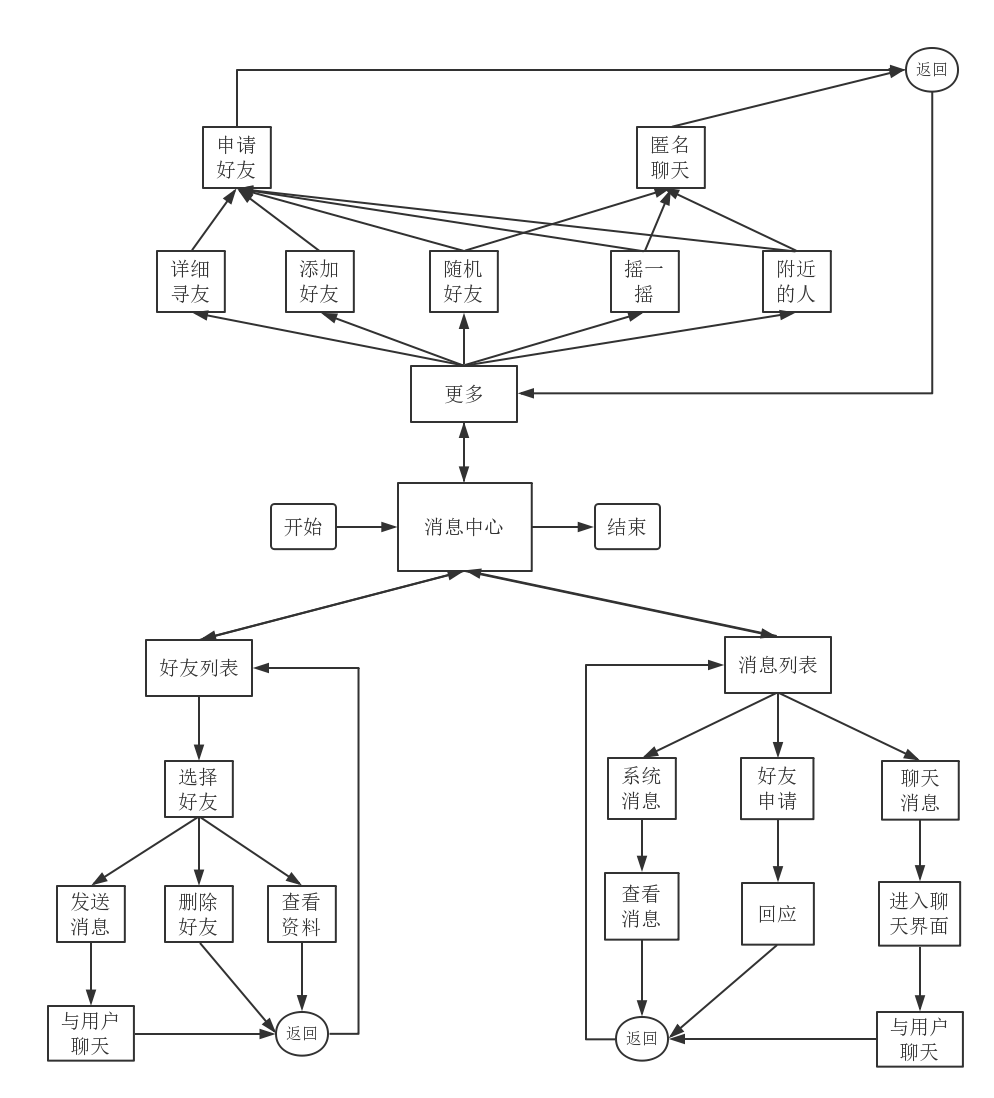


图3-1-2 消息中心流程图

**3.1.3 状态圈模块**

该模块主要是负责管理用户的状态，包括发表状态，删除状态，评论，点赞等功能。处理流程如图所示．

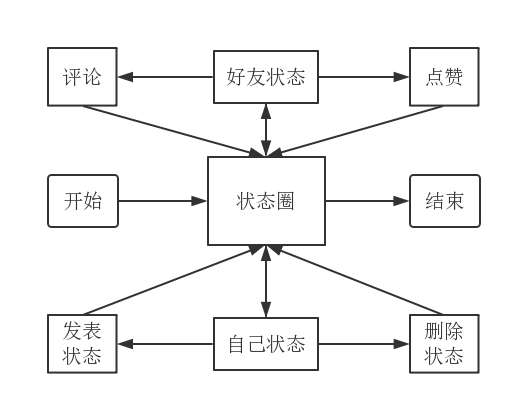


图3-1-3 状态圈流程图

**3.1.4 用户中心模块**

该模块的功能是围绕用户而来的，包括摇一摇签到，成为VIP，提交反馈，修改隐私密码，以及查看和更新用户基本资料等功能，处理流程如图所示。

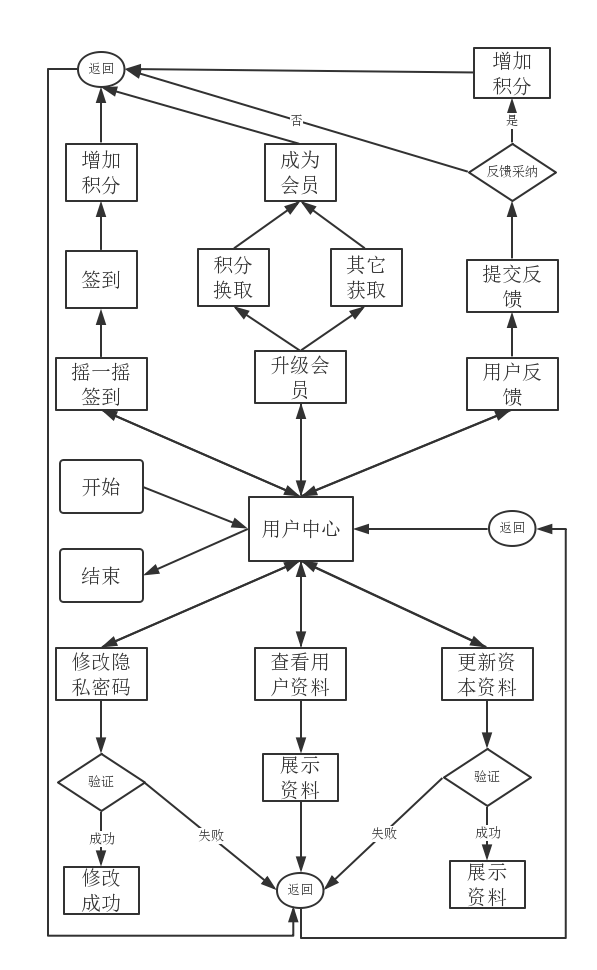


图3-1-4 用户中心流程图

**3.2 数据库设计**

数据库结构设计分为服务器端MYSQL数据库和客户端SQLITE数据库．

**3.2.1 服务器端数据库**

服务器端的数据表结构设计．

1. 用户表

表3-2-1-1 用户表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| id | 自增主键 | Int | 0 | 是 | 否 | 是 |
| phoneNumber | 手机号 | Varchar | 20 | 是 | 否 |  |
| icgId | IcgId号 | Varchar | 30 | 是 |  |  |
| password | 密码 | Varchar | 30 |  | 否 |  |
| access | 权限 | Int | 0 |  |  |  |
| star | 星级 | Int | 0 |  |  |  |
| vip | Vip等级 | Int | 0 |  |  |  |
| coin | 金币 | Int | 0 |  |  |  |
| nickName | 昵称 | Varchar | 30 |  |  |  |
| iconPath | 图像路径 | Text | 0 |  |  |  |
| sex | 性别 | Int | 0 |  |  |  |
| qrCodePath | 二维码路径 | Text | 0 |  |  |  |
| birthYear | 出生年 | Int | 0 |  |  |  |
| birthMonth | 出生月 | Int | 0 |  |  |  |
| birthDay | 出生日 | Int | 0 |  |  |  |
| email | 邮件 | Varchar | 30 | 是 |  |  |
| sign | 签名 | Text | 0 |  |  |  |
| netState | 在线状态 | Int | 0 |  |  |  |
| provinceId | 省份编码 | Varchar | 10 |  |  |  |
| cityId | 城市编码 | Varchar | 10 |  |  |  |
| createTime | 创建时间 | DateTime | 0 |  |  |  |
| modifyTime | 修改时间 | DateTime | 0 |  |  |  |

表3-2-1-3 消息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| senderPhone | 消息发送者手机号 | Varchar | 20 |  | 否 | 是 |
| receiverPhone | 消息接收者手机号 | Varchar | 20 |  | 否 | 是 |
| type | 消息类型 | Int | 0 |  | 否 |  |
| msg | 消息内容 | Text | 0 |  |  |  |
| createTime | 创建时间 | DateTime | 0 |  |  |  |

1. 好友关系表

其中masterPhone和friendPhone都是用户表里的外键引用．而且这两者的组合是一个唯一约束．unique Key (masterPhone,friendPhone)．

表3-2-1-4 好友关系表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| masterPhone | 所属人手机号 | Varchar | 20 |  | 否 | 是 |
| friendPhone | 好友手机号 | Varchar | 20 |  | 否 | 是 |
| createTime | 创建时间 | DateTime | 0 |  |  |  |

1. 定位信息表

其中masterPhone是用户表里的外键引用．

表3-2-1-5 定位信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| masterPhone | 所属人手机号 | Varchar | 20 |  | 否 | 是 |
| longtitude | 经度 | Double | 0 |  |  |  |
| latitude | 纬度 | Double | 0 |  |  |  |
| detailAddress | 详细地址 | Text | 0 |  |  |  |
| createTime | 创建时间 | DateTime | 0 |  |  |  |

1. 用户反馈表

这个表记录着每一个用户的反馈信息，用户可以匿名．

表3-2-1-8 用户反馈表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| id | 自增Id | Int | 0 | 是 | 否 | 是 |
| masterContact | 反馈人联系方式 | Varchar | 20 |  | 否 |  |
| createTime | 创建时间 | DateTime | 0 |  |  |  |
| masterName | 反馈人昵称 | Varchar | 20 |  |  |  |
| content | 反馈信息 | Text | 0 |  | 否 |  |

1. 用户状态表

这个表记录着每一个用户发表的状态信息,这里定义每一条状态最多发表6张图片,用户上传包含图片的信息时，服务器端会将图片保存在服务器本地地址，然后将图片地址存入数据库,其中masterPhone是用户表中的外键引用．

表3-2-1-9 用户状态表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| id | 自增Id | Int | 0 | 是 | 否 | 是 |
| masterPhone | 所属人手机号 | Varchar | 20 |  | 否 | 是 |
| createTime | 创建时间 | DateTime | 0 |  |  |  |
| content | 文字内容 | Text | 0 |  | 否 |  |
| imgOne | 第一张图片路径 | Text | 0 |  |  |  |
| imgTwo | 第二张图片路径 | Text | 0 |  |  |  |
| imgThree | 第三张图片路径 | Text | 0 |  |  |  |
| imgFour | 第四张图片路径 | Text | 0 |  |  |  |
| imgFive | 第五张图片路径 | Text | 0 |  |  |  |
| imgSix | 第六张图片路径 | Text | 0 |  |  |  |

1. 用户评论表

这个表记录着每一个用户发表的评论信息,其中commentMasterPhone是用户表中的外键引用，diaryId是状态表中的外键引用．

表3-2-1-10 用户评论表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| id | 自增Id | Int | 0 | 是 | 否 | 是 |
| diaryId | 所属状态Id | Int | 0 |  | 否 | 是 |
| commentMasterPhone | 评论发表人的手机 | Varchar | 20 |  | 否 | 是 |
| commentContent | 评论的内容 | Text | 0 |  |  |  |
| createTime | 创建时间 | DateTime | 0 |  |  |  |

1. 用户签到表

这个表记录着每一个用户的签到信息,其中masterPhone是用户表中的外键引用．

表3-2-1-11 用户签到表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| id | 自增Id | Int | 0 | 是 | 否 | 是 |
| masterPhone | 签到人的手机 | Varchar | 20 |  | 否 | 是 |
| content | 评论的内容 | Text | 0 |  |  |  |
| createTime | 创建时间 | DateTime | 0 |  |  |  |
| longitude | 签到的经度 | Double | 0 |  |  |  |
| latitude | 签到的纬度 | Double | 0 |  |  |  |
| detailAddress | 签到的详细地址 | Text | 0 |  |  |  |

1. 用户上传信息表

这个表记录着每一个用户的签到信息,其中masterPhone是用户表中的外键引用．

表3-2-1-12 用户上传信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| id | 自增Id | Int | 0 | 是 | 否 | 是 |
| masterPhone | 签到人的手机 | Varchar | 20 |  | 否 | 是 |
| filePath | 上传后的文件所在路径 | Text | 0 |  | 否 |  |
| createTime | 创建时间 | DateTime | 0 |  |  |  |
| fileName | 上传后的文件名字 | Varchar | 20 |  | 否 |  |
| detail | 详细信息介绍 | Text | 0 |  |  |  |

1. 版本信息表

这个表记录着应用的版本信息．

表3-2-1-13 版本信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| id | 自增Id | Int | 0 | 是 | 否 | 是 |
| versionName | 版本名称 | Varchar | 20 |  | 否 |  |
| versionCode | 版本代码 | Varchar | 20 |  | 否 |  |
| apkPath | 应用所在路径 | Text | 0 |  | 否 |  |
| detail | 详细信息介绍 | Text | 0 |  |  |  |

**3.2.2 客户端数据库**

客户端的数据表结构设计，采用Sqlite数据库．

1. 上传下载任务表．

这个表记录所有的上传和下载任务信息．

表3-2-2-1 上传下载任务表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| \_id | 自增Id | Integer | 0 | 是 | 否 | 是 |
| task\_name | 任务名称 | Varchar | 20 |  | 否 |  |
| updownload\_url | 上传路径或者下载路径 | Varchar | 60 | 是 | 否 |  |
| file\_path | 上传文件或者下载接收文件 | Varchar | 60 |  | 否 |  |
| file\_size | 文件总大小 | Integer | 0 |  |  |  |
| complete\_size | 完成大小 | Integer | 0 |  |  |  |
| type | 类型,上传还是下载 | Integer | 0 |  |  |  |
| finished | 是否完成标识 | Integer | 0 |  |  |  |

1. 上传下载详细信息表．

上传下载使用多线程断点，这个表记录每一个上传和下载任务的线程详细信息．

表3-2-2-2 上传下载线程信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| \_id | 自增Id | Integer | 0 | 是 | 否 | 是 |
| thread\_id | 线程Id | Integer | 0 |  |  |  |
| start\_pos | 开始位置 | Integer | 0 |  |  |  |
| end\_pos | 结束位置 | Integer | 0 |  | 否 |  |
| complete\_size | 完成大小 | Integer | 0 |  |  |  |
| complete\_size | 完成大小 | Integer | 0 |  |  |  |
| type | 类型,上传还是下载 | Integer | 0 |  |  |  |
| updownload\_url | 上传或下载路径 | Varchar | 60 |  |  |  |

1. 用户信息表．

为了节省网络请求数据量，Android客户端会在本地存储一些常联系的用户的信息．(不包括隐私信息)

表3-2-2-3 用户信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| id | 自增主键 | Integer | 0 | 是 | 否 | 是 |
| phoneNumber | 手机号 | Varchar | 20 | 是 | 否 |  |
| icgId | IcgId号 | Varchar | 30 | 是 |  |  |
| access | 权限 | Integer | 0 |  |  |  |
| star | 星级 | Integer | 0 |  |  |  |
| vip | Vip等级 | Integer | 0 |  |  |  |
| coin | 金币 | Integer | 0 |  |  |  |
| nickName | 昵称 | Varchar | 30 |  |  |  |
| iconPath | 图像路径 | Text | 0 |  |  |  |
| sex | 性别 | Integer | 0 |  |  |  |
| qrCodePath | 二维码路径 | Text | 0 |  |  |  |
| birthYear | 出生年 | Integer | 0 |  |  |  |
| birthMonth | 出生月 | Integer | 0 |  |  |  |
| birthDay | 出生日 | Integer | 0 |  |  |  |
| email | 邮件 | Varchar | 30 | 是 |  |  |
| sign | 签名 | Text | 0 |  |  |  |
| netState | 在线状态 | Integer | 0 |  |  |  |
| provinceId | 省份编码 | Varchar | 10 |  |  |  |
| cityId | 城市编码 | Varchar | 10 |  |  |  |

1. 离线聊天信息表．

客户端会为每两个用户之间的聊天记录创建一个聊天信息表,方便查询历史记录,使用两个用户的手机号作为一部分表名字,例如”chatInfo\_手机1\_手机2”作为表名．

表3-2-2-4 用户聊天记录表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| \_id | 自增Id | Integer | 0 | 是 | 否 | 是 |
| isCome | 是否接收的消息 | Integer | 0 |  |  |  |
| state | 发送状态,发送中还是发送完毕 | Integer | 0 |  |  |  |
| message | 聊天内容 | Text | 0 |  |  |  |
| date | 消息发送的时间 | Text | 0 |  |  |  |

1. 离线列表消息表．

客户端存储一些消息在本地，作为消息记录，例如存储系统通知，存储好友申请等消息．

表3-2-2-5 消息记录表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否唯一 | 是否允许为空 | 是否主键 |
| \_id | 自增Id | Integer | 0 | 是 | 否 | 是 |
| senderPhone | 消息发送人手机号 | Varchar | 20 |  | 否 |  |
| receiverPhone | 消息接受人手机号 | Varchar | 20 |  | 否 |  |
| message | 消息内容 | Text | 0 |  |  |  |
| type | 消息类型 | Integer | 0 |  |  |  |
| IsReaded | 是否已经读取 | Integer | 0 |  |  |  |
| isHandled | 是否已经处理 | Integer | 0 |  |  |  |
| date | 消息发送的时间 | Text | 0 |  |  |  |

Mainactivity.java代码如下：

package top.cuijinze.tetl;

import androidx.appcompat.app.AppCompatActivity;

import androidx.fragment.app.Fragment;

import androidx.viewpager2.widget.ViewPager2;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.TextView;

import com.gyf.immersionbar.ImmersionBar;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

ViewPager2 tab\_viewpager2;

//定义底部菜单栏按钮

private LinearLayout tab\_home,tab\_square,tab\_message,tab\_profile;

//定义底部菜单栏图片

private ImageView tab\_iv\_home,tab\_iv\_square,tab\_iv\_message,tab\_iv\_profile,ivCurrent;

//定义底部菜单栏文字

private TextView tab\_tv\_home,tab\_tv\_square,tab\_tv\_message,tab\_tv\_profile,tvCurrent;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

initPager();

initTabView();

//定义一些点击事件

ImmersionBar.with(this)

//.barColor(R.color.gray)//同时自定义状态栏和导航栏颜色，不写默认状态栏为透明色，导航栏为黑色

.fitsSystemWindows(true)

.statusBarColor(R.color.gray) //状态栏颜色，不写默认透明色

//.navigationBarDarkIcon(true) //导航栏图标是深色，不写默认为亮色

.navigationBarEnable(true) //是否可以修改导航栏颜色，默认为true

.statusBarDarkFont(true, 0.2f) //原理：如果当前设备支持状态栏字体变色，会设置状态栏字体为黑色，如果当前设备不支持状态栏字体变色，会使当前状态栏加上透明度，否则不执行透明度

.init();

}

private void initTabView() {

//获取底部菜单按钮

tab\_home=findViewById(R.id.tab\_home);

tab\_square=findViewById(R.id.tab\_square);

tab\_message=findViewById(R.id.tab\_message);

tab\_profile=findViewById(R.id.tab\_profile);

//获取底部导航栏图片

tab\_iv\_home=findViewById(R.id.tab\_iv\_home);

tab\_iv\_square=findViewById(R.id.tab\_iv\_square);

tab\_iv\_message=findViewById(R.id.tab\_iv\_message);

tab\_iv\_profile=findViewById(R.id.tab\_iv\_profile);

//获取底部导航栏文字

tab\_tv\_home=(TextView) findViewById(R.id.tab\_tv\_home);

tab\_tv\_square=(TextView)findViewById(R.id.tab\_tv\_square);

tab\_tv\_message=(TextView)findViewById(R.id.tab\_tv\_message);

tab\_tv\_profile=(TextView)findViewById(R.id.tab\_tv\_profile);

//设置初始进入app时的页面

tab\_home.setSelected(true);

ivCurrent=tab\_iv\_home;

tvCurrent=tab\_tv\_home;

//tvCurrent.setTextColor(R.color.tab\_selected);

//设置底部导航栏点击事件

tab\_home.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

changeTab(v.getId());

}

});

tab\_square.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

changeTab(v.getId());

}

});

tab\_message.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

changeTab(v.getId());

}

});

tab\_profile.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

changeTab(v.getId());

}

});

}

private void initPager() {

//获取Viewpager2

tab\_viewpager2=findViewById(R.id.id\_tab\_viewpager);

//设置viewpager禁止滑动

tab\_viewpager2.setUserInputEnabled(false);

//定义fragment数组

ArrayList<Fragment> tab\_fragments =new ArrayList<>();

tab\_fragments.add(new HomeFragment());

tab\_fragments.add(new SquareFragment());

tab\_fragments.add(new MessageFragment());

tab\_fragments.add(new ProfileFragment());

//初始化适配器

TabFragmenntPagerAdapter tab\_adapter=new TabFragmenntPagerAdapter(getSupportFragmentManager(),getLifecycle(),tab\_fragments);

//给viewpager2添加适配器

tab\_viewpager2.setAdapter(tab\_adapter);

//fragment选中事件

tab\_viewpager2.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {

@Override

public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {

super.onPageScrolled(position, positionOffset, positionOffsetPixels);

}

@Override

public void onPageSelected(int position) {

super.onPageSelected(position);

}

@Override

public void onPageScrollStateChanged(int state) {

super.onPageScrollStateChanged(state);

}

});

}

private void changeTab(int position) {

ivCurrent.setSelected(false);

switch (position) {

case R.id.tab\_home:

/\*点击时跳转到相应页面，并且取消过渡动画\*/

tab\_viewpager2.setCurrentItem(0, false);

case 0:

tab\_iv\_home.setSelected(true);

ivCurrent = tab\_iv\_home;

break;

case R.id.tab\_square:

tab\_viewpager2.setCurrentItem(1, false);

case 1:

tab\_iv\_square.setSelected(true);

ivCurrent = tab\_iv\_square;

break;

case R.id.tab\_message:

tab\_viewpager2.setCurrentItem(2, false);

case 2:

tab\_iv\_message.setSelected(true);

ivCurrent = tab\_iv\_message;

break;

case R.id.tab\_profile:

tab\_viewpager2.setCurrentItem(3, false);

case 3:

tab\_iv\_profile.setSelected(true);

ivCurrent = tab\_iv\_profile;

break;

}

}

}

1. **心得体会**

android开发是所有编程课里最贴近我们生活的一门课，做出来的东西更能理解。通过前四次的实验逐步的了解安卓开发的创建、使用控件、创建activity和连接数据库。最后再通过很多的地方查找，马马虎虎才完成了此次大作业。困难多多，收获也多多。

比如一开始的时候总是APP闪退，在模拟器上也是无法运行，原因有很多种，排除指针空和容量过大后，应该是onclick函数的问题，作业也很好的进行了完成，学到了很多安卓开发相关的知识。