

项目说明文档

项目技术

- 前端使用Java语言和Android SDK进行原生开发
- 后端API开发：使用Django框架和Django REST framework编写API服务
- 数据库：使用Django自带的sqlite3作为数据库
- 实时私信与通知：使用Firebase及其RealtimeDatabase实现

关键功能的设计与实现

用户

用户登录

用户登录模块复用了rest-framework的Token认证机制，客户端发送登录请求时，服务器便会生成一个Token并将此Token返回给客户端,作为客户端进行请求的唯一标识，以后客户端只需在http header中带上这个Token前来请求数据即可，用户登出时该Token被作废，此后客户端如想再访问API则必须重新登录。

重置密码

重置密码模块采用了基于邮箱的认证方式，用户请求重置密码时只需传入注册时填写的邮箱，后端根据该邮箱查找到对应用户，并生成一个基于该用户的一组独特uidb64和token，通过邮件发送至该用户的邮箱。用户收到邮件后只需将uidb64和token在前端填入，后端检查填入的token是否与基于该用户生成的token匹配，若匹配则允许此次修改密码操作。

关注列表

使用RecyclerView来实现，在关注列表中区分了三类：关注、粉丝以及黑名单，对其中每一类的每一个用户都有一个按钮，关注和粉丝列表为关注/取消关注，黑名单为屏蔽/取消屏蔽。由于ViewHolder与RecyclerView并不是一对一的关系，因此需要将对应的按钮点击事件设置为调用Adapter中的对应接口，如果直接处理会导致多个RecyclerView由于使用同一个ViewHolder绑定从而出现事件被替换掉的情况。

私信与通知

为了实现实时私信与通知，我们采用Firebase Realtime Database来实现。为APP创建一个Firebase项目，并将Realtime Database SDK 添加至应用，此后通过ValueEventListener对数据进行监听，可以实现数据修改后的实时响应。

对于私信，在Realtime Database中为每一对用户新建一个子项，其中包含该对用户的所有历史私信记录。对于通知，则根据需求，在点赞、评论、新发表动态、私信时为相应需要收到通知的用户的子项添加对应的通知类别和内容。这样配合实时响应就可以在前端APP中做出对应的更新。

此外，Firebase还提供了Cloud Message(FCM)功能用来实现消息推送，但是这需要Google Play商店应用或支持Google API的模拟器，因此放弃。我们选择在APP运行时通过前端用类似更新私信的方式发送系统通知。

信息发布

图片

通过 Intent 进入相册或者调用相机，但是默认的返回值是相对地址，利用 ContentResolver 获取绝对地址，否则无法上传视频

视频

和图片类似

富文本

利用第三方库Knife实现<https://github.com/mthli/Knife>

发布时将文本的格式转换为HTML，加载时通过HTML内容再解析为富文本

位置

利用 `LocationManager` 获取当前经纬度，然后通过 `Geocoder` 解析当前地址

信息查看与操作

动态浏览

采用Fragment+RecyclerView来实现，使得组件可以在需要浏览动态的地方重用，包括主页面的各个不同属性分类浏览的功能以及用户主页和“我的”页面中带有的该用户所有动态的浏览。只需要在对应的Adapter中放入不同的动态的列表即可灵活显示，这也自然实现了用户屏蔽、关注等操作后浏览动态的更新。

图片

图片的地址都是网络URL，但是ImageView只能设置本地URL，需要利用 `Picasso` 将网络图片加载到ImageView上

视频

视频的地址也是网络URL，但是VideoView只能设置本地URL，需要先将网络视频下载，然后才能加载；

利用第三方库Exoplayer实现<https://github.com/google/ExoPlayer>

动态分享

调用安卓默认的 `Intent.ACTION_SEND` 即可选择应用，通过 `Intent.putExtra` 设置分享内容

分类浏览

通过 `TabLayout` 和 `ViewPager2` 的绑定实现Fragment的切换，不同的Fragment设置为不同的内容

动画

初始在界面放置一个环形 `ProgressBar`，并使其覆盖整个屏幕，设置模式为不确定，那么效果就是一直转圈圈，当动态加载完毕之后将其设置为 `gone` 即可

API文档

本项目采用的是前后端分离的开发方式，因此后端API独立存在并可直接调用，API的具体形式可访问后端根目录url下的swagger-ui/和redoc/页面的url进行查看

组员的分工与贡献情况

在所有实现的功能中，我们小组的分工如下：

王政：前端用户相关内容和通知、私信以及Firebase部分，项目部署等

郭恩惠：前端信息发布和各类浏览操作以及搜索等

徐霁然：项目的后端等