

项目说明文档

项目简介

“THU BBS”是一个面向学生的校园论坛APP，供同学们分享动态、交友聊天的平台。

关键功能设计与实现技术

用户部分

输入检查

注册登录账号、设置密码、修改个人信息时需要用户输入指定格式的字符串，例如邮箱、手机号有规定格式，密码有长度和字符限制等，我们使用[Ratifier](#)表格验证库进行对指定格式字符串的检查（本质是基于正则表达式），并实现了输入字符时，提示词自动消失，输入密码自动转为“*”遮挡状态。当用户输入不合法时，显示红色感叹号图标并提示正确的格式。若邮箱已经被注册或者密码不正确，弹出对话框提示用户。

跳转登录页自动填充

当用户完成注册后跳转进入登录页，此时我们自动帮用户填充邮箱和密码，用户就不需要二次输入。同时通过SharedPreferences存储用户名、邮箱、token、头像URL等信息。

头像圆形显示

使用Glide的apply(RequestOptions.circleCropTransform())加载图片为原形，以及[circleimageview](#)

Fragment/Activity之间的切换

同一Activity内的Fragment之间的切换通过 `FragmentManager` 完成，但是还有Fragment内部包含Fragment的情况，借助 `ViewPager` 和 `TabLayout` 完成，为了解决从Fragment的子Fragment切换后返回上一子Fragment不存在的问题，需要设置ViewPager为

```
viewPager.setSaveEnabled(false);
```

为了实现从某个Activity进入其他Activity内的Fragment（例如，从 `PostDetailActivity` 进入 `MainActivity` 下的他人主页Fragment），需要在StartActivity时，通过intent指定要进入的MainActivity的哪一个Fragment，而MainActivity需要在onCreate函数中判断intent参数中是否指定了要切换到的Fragment。

```
intent.putExtra("Fragment", "profileOtherUsers");
```

统一的信息展示模块

查看首页的帖子列表、分话题的帖子列表、用户自己的帖子列表、其他用户的帖子列表、收藏的帖子列表这几种模式，任务相同，都是要显示一个帖子列表，我们在前端将其统一为相同的基础模块，使用自定义的queryPost()函数从后端根据不同参数获取指定帖子列表（如筛选参数、排序参数、屏蔽参数等，详见后端部分），从而实现了“一次实现、多次使用”。

信息发布

定位

定位使用了腾讯位置服务提供的android定位SDK。我们先通过 `locationManager` 与 `locationListener` 获取设备所处的经纬度，然后将获取到的设备经纬度通过网络传给 `TencentLocationSdk`，获取到对应的地点名称。（注：因为 android studio 移动模拟器的定位默认是在美国，所以我们将“获取到的设备经纬度”先写成了主楼所在的位置）

从相册中选取或调用相机进行拍照或录像

首先调出选择对话框，让用户选择是从相册中选取文件还是调用相机进行拍摄。如果是调用相机进行拍照或录像，以视频为例，则要先创建一个以当前时间命名的视频文件名、声明一个相机的录像行为的意图、设置相机录像的输出路径为之前创建的视频文件，并将视频文件的Uri赋值给 `capturedVideoUri`，之后即可上传刚刚录制的视频。如果是上传已有的图片或视频文件，以图片为例，要获取已选择照片的路径对象，然后调用 `uploadImage()` 上传到后端，由后端图床返回一个url，在发布帖子时作为资源url统一上传。

信息查看

点赞动画

使用 `LikeButton` 实现点赞、收藏操作的流畅动画效果

图片加载

使用 `glide` 进行图片加载，显示评论区、帖子作者的圆形头像，以及帖子的图片内容

富文本显示

使用 `MarkdownView` 渲染富文本，使用github风格的markdown主题，支持加粗、斜体、六级标题、标记、代码、数学公式、emoji等基本markdown语法

视频加载

使用Google的 `ExoPlayer`，解决了 `videoview` 对视频格式要求过于严格导致经常无法播放的问题，并且使用了 `ExoPlayer` 自带的进度条和暂停播放的按钮，为用户提供了更便捷的视频播放服务。

聊天

可以从通知页面或者他人主页进入聊天界面，本质是 `RecyclerView`，通过消息的发出/接收ID来区分消息是用户发出的还是收到的，从而分别显示在左右两侧。

通知

我们自定义了 `NoticeModel` 这个类，包含了用户私信通知和系统通知两种类型，通过布尔变量 `isSystem` 区分，系统通知包括点赞、评论、被关注等类型，其详细信息是通过嵌套 `SystemNoticeDetail` 类来定义；私信通知可以之间点击进入聊天界面。

网络通信

我们使用 `retrofit2` 网络库发起网络请求和处理响应。使用 `retrofit2` 的 `GsonConverterFactory` 实现了自动解析后端发送来的JSON字符流为指定的类，简化了手工解析JSON的操作，并能安全处理异常。例如对于一个注册的Post请求，只需要在API中指定发起的请求的body类和接收响应的类即可，`GsonConverterFactory` 可以自动完成从JSON到类的序列化和反序列化操作。

动画

启动动画：通过自定义 `SplashActivity`，延时2秒播放完动画后进入主页

内容加载动画：使用 [Android-Loading-Animation](#)，但是这个库只能将动画显示为小部分，不能遮挡后面的整个页面，为了能遮挡后面页面的内容加载过程，我们设计了一个伴随动画展示而出现、在动画播放结束后消失的“幕布” (loadingOverlay)

```
// 显示加载动画
view.loadingOverlay = view.findViewById(R.id.loadingOverlay);
loadingOverlay.setVisibility(View.VISIBLE);
// 执行后台加载任务
queryPost(0);
handler.postDelayed(() -> {
    // 关闭加载动画
    loadingOverlay.setVisibility(View.GONE);
    // 显示后面的内容
    recyclerView.setVisibility(View.VISIBLE);
    rootLayout.setVisibility(View.VISIBLE);
}, 2000); // 2秒钟延迟关闭加载动画
```

页面切换动画：[Bungee](#)库的Zoom动画，原理是通过

```
((Activity) context).overridePendingTransition()
```

美化

按钮等组件使用Google的[material components](#)

其他

使用[butterknife](#)自动化绑定views

后端

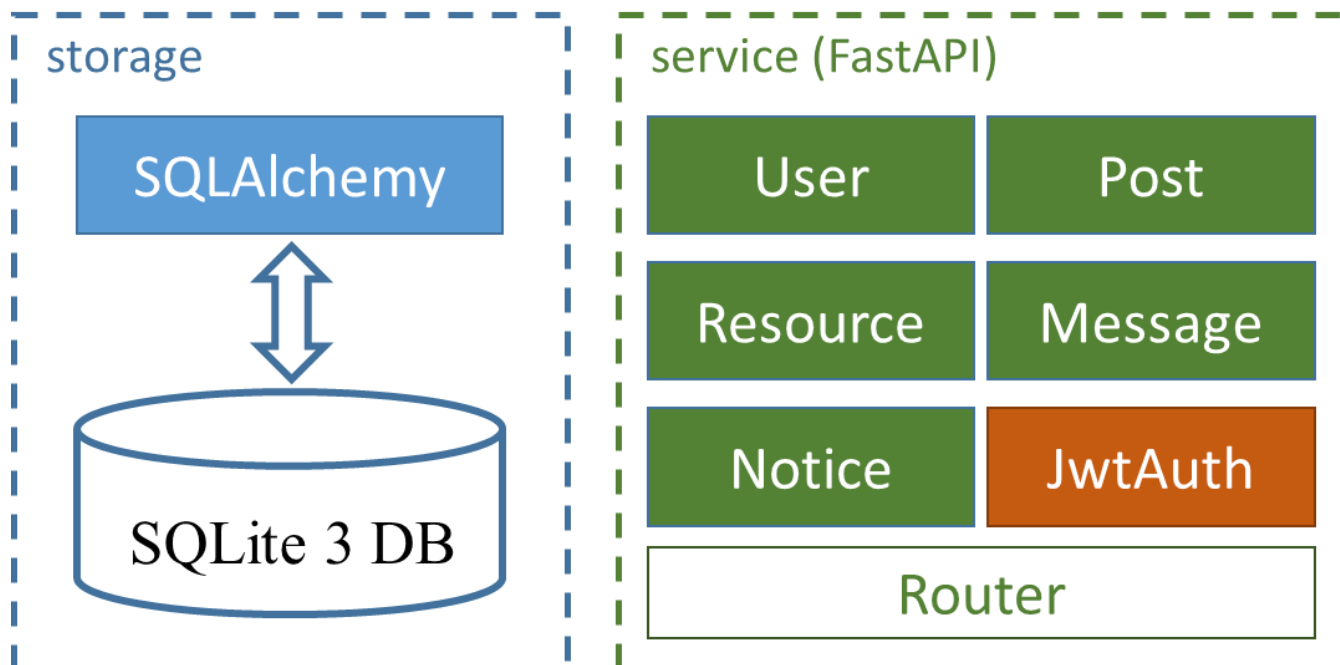
- 使用 *JSON Web Token* (JWT)处理鉴权
- 对视频文件进行压缩处理，提高通信效率
- 自建图床，可以稳定存储图片和视频
- 通过 `FastAPI` 自动生成API在线文档

服务框架：`FastAPI`

数据库：`SQLite3`

ORM框架：`SQLAlchemy`

部署方案：nginx 反代理，`uvicorn` 服务器



部署说明

详见 `src/frontend` 和 `src/backend` 目录下的 `readme.md` 文件

使用指南

详见操作指南文档 `Guide.pdf`

演示视频

详见 `THU BBS demo.mp4`