

移动应用软件开发：大作业说明文档

1. 代码说明

1.1 后端框架

框架: Python + Flask

数据库: MySQL

第三方库: fuzzywuzzy, scrypt

说明:

- 使用flask_sqlalchemy将数据库表作为类调用
- 使用fuzzywuzzy库实现模糊搜索
- 使用scrypt加密密码

使用说明:

- 运行forum_app.sql建立数据库表, 数据库端口号应为3306, 数据库用户名为forum, 密码为3306, 数据库名称为forum_app。也可在config.cfg里替换具体用户名等。
- 进入虚拟环境后, 在后端路径中运行run.bat设置参数
- 执行flask run启动后端

1.2 前端

第三方库: Okhttp, Firebase Cloud Messaging

说明:

- 使用Okhttp与后端连接
- 使用谷歌的Firebase Cloud Messaging实现通知接收与发送

使用说明:

- 将res/values/strings.xml中backend_url替换为对应的后端URL (如在虚拟机运行, 需替换为10.0.2.2)
- Android Studio里build后在虚拟机或真机运行即可

2. 成员分工

- a. 陈彦杨：动态查看，整体优化
- b. 霍建宇：后端，个人简介，通知，搜索
- c. 白承桓：动态发布，草稿，整体优化

3. 设计亮点

3.1 后端

- 用户密码
 - 所有密码在数据库中以密文形式保存
 - 使用Python的scrypt库中hash函数将密码转成一个Hash值，保存到数据库中
 - 用户登录时会将输入的密码用同样的方式转成Hash值后与数据库中保存的Hash对比
 - 这样就确保了安全性，数据库管理员无法得知用户密码，同时数据库泄漏也不会导致密码泄漏
- 模糊搜索
 - 使用Python的fuzzywuzzy库实现
 - 具体实现在utils.py/fuzzysearch中，输入要查找的key，options和一个默认为75的阈值（threshold）。通过fuzzywuzzy的process函数搜索“近似度”大于阈值的options返回。
- 通知发送
 - 使用Firebase Cloud Messaging
 - 在Firebase Cloud Messaging平台中注册了一个server，将生成的key等保存在前端中app/google-services.json
 - 后端收到点赞、评论或创建新动态时会向Firebase Cloud Messaging发送request，其body中包含了该通知的标题、内容和数据库中保存的目标用户的token
 - Firebase Cloud Messaging收到通知request后会向用户发送通知（详见3.2）

3.2 前端

- 登录
 - 用户登录后会将登录信息保存到SharedPreferences中，下次启动时可以自动登录
 - 用户点击“退出登录”后，会将登录信息从SharedPreferences中删除
- 通知
 - 用户登录后会生成token，并调用后端register-token接口将生成的token保存到数据库中
 - Firebase Cloud Messaging收到通知request后会向request body中的token发送该通知，该token所属的手机就会接收到这个通知
 - 用户点击“退出登录”后会调用后端unregister-token接口将保存的token从数据库中删除，这样一来就不会再收到通知
 - 如果不是“正常”的退出登录，如直接返回到登录页面登录的话，则token依然保存到数据库中，并会继续收到通知，这与大部分社交媒体如Facebook的设计一致
- 图片/音频/视频下载
 - 所有多媒体下载都使用Android的DownloadManager
 - 下载成功后DownloadManager会发送ACTION_DOWNLOAD_COMPLETE广播，activity接收到后再进行对应多媒体的替换
- 发布图片
 - 选择图片后可以在发布界面中预览要发布的图片，且利用ExifInterface.getAttributeInt获取图片的旋转角度，自动对进行旋转放置的图片进行旋转校正
- 草稿保存
 - 使用SharedPreferences对草稿进行自动key-mapping保存。在无后端环境下可以进行快速的对草稿进行操作
 - 当草稿继续编辑的动态发布时，通过根据时间生成的key对SharedPreference里对应的数据进行快速删除