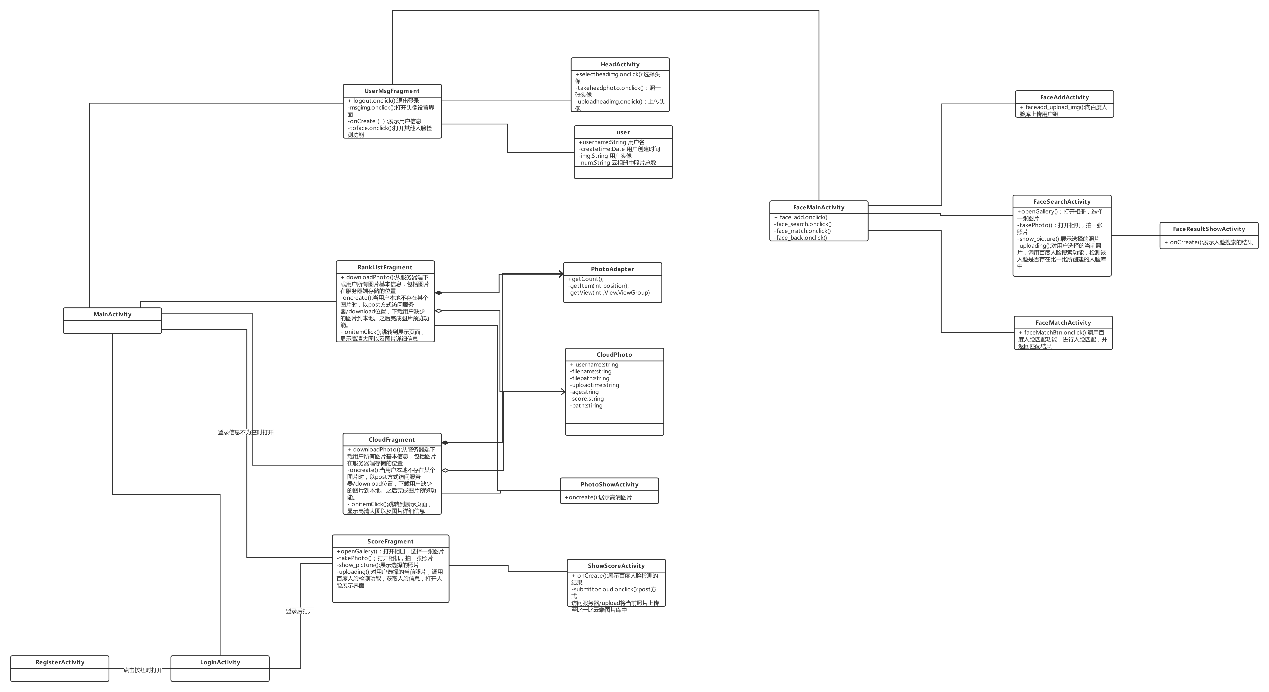
easy项目说明文档

1. 数据库设计
2. 用户表设计（users）
   1. 用户名username:String
   2. 用户密码password:String
   3. 用户创建时间createtime:Data
   4. 加密口令salt:String
   5. 用户头像img:String
3. 用户图片表设计（images）
   1. 用户名：username String
   2. 文件名：filename String
   3. 文件路径：filepath String
   4. 文件上传时间: uploadtime String
   5. 预测图片年龄：age String
   6. 预测图片颜值：score String
4. 单元测试设计

1测试web服务器返回的JsonArray图片信息

2测试百度API接口返回Json图片检测结果信息

1. 类设计



1. 本地接口设计
2. public static String getFilePathByUri(Context context, Uri uri)

根据Uri来获取文件存储的实际路径

1. public static Bitmap SampleRateCompress(String filepath)

图片压缩算法

1. takePhoto();

在activity中打开Android的相机功能

1. openGallery();

在activity中打开Android的相册功能

1. convertToBytes

将inputStream变为byte[]

1. UriToBase()

将uri的图片变为base64编码

1. public static String add(final String assess\_token, String img, String userinfo, String user)

将当前人脸信息发送到百度人脸库中，建立新的人脸信息

1. public static String faceDetect(final String assess\_token,String img)

将当前照片发送到百度人脸库中，检测照片中的人脸

1. public static String faceMatch(final String assess\_token, String img1, String img2)

将当前照片发送到百度人脸库中，匹配照片中的人脸

1. public static String faceSearch(final String assess\_token, String img)

将当前照片发送到百度人脸库中，搜索照片中的人脸

1. Restful网络接口设计
2. changehead.js:
   1. 发送get请求时，获取用户头像
   2. 发送post请求时，更改用户头像
3. login.js：

发送get请求时，查询用户名与密码是否匹配

1. Register.js

发送post请求时，将用户注册信息存贮到数据库

1. upload.js
   1. 发送get请求时，获取当前图片信息
   2. 发送post请求时，上传图片至服务器
2. sort.js

发送post请求，获取排行榜界面需要的图片信息

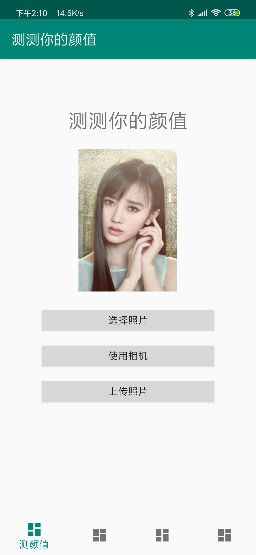
1. getphoto.js

发送post请求，获取云相册界面需要的图片信息

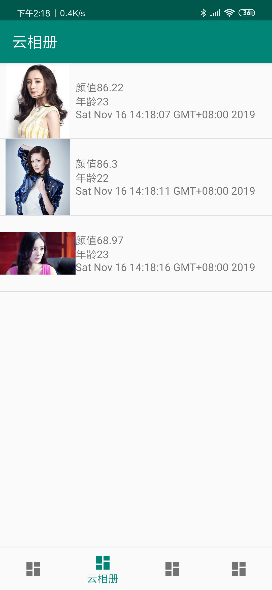
1. 使用说明
2. 登录&注册功能：点击app，进入app登录界面，登录界面可切换至注册界面



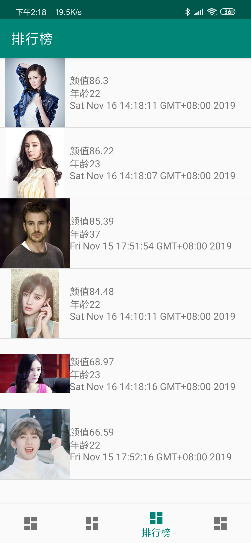
1. 人脸检测功能介绍；点击测颜值界面，进入测颜值界面上传图片后返回颜值得分与预测年龄，并可选择上传文件至云相册



1. 云相册功能介绍：点击云相册，进入云相册界面



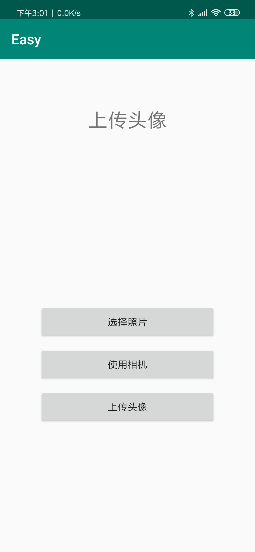
1. 排行榜功能介绍：点击排行榜，进入排行榜界面



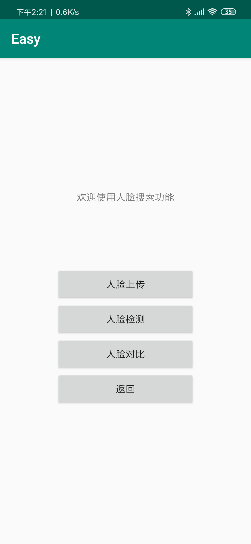
1. 个人中心功能介绍：点击个人中心，进入个人中心界面



1. 更换头像功能介绍：点击头像，进入上传头像界面

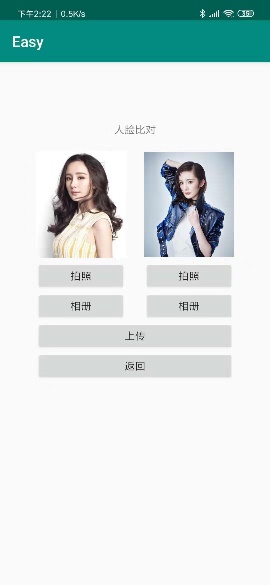
1. 人脸搜索界面：个人中心中点击人脸搜索，进入人脸搜索界面



1. 人脸添加功能介绍：从相册中选择照片或者使用相机拍照,输入用户名信息和备注信息，点击上传图片，即将图片和对应信息存入到数据库中。



1. 人脸检测功能介绍：从相册中选择照片或者使用相机拍照,选中两张图片,点击上传图片，传送到后台数据库中，调用百度API接口检测两张照片上的人物，并把是否是同一人和相似度的结果返回到界面上。



1. 人脸匹配功能介绍：从相册中选择照片或者使用相机拍照,点击上传图片，传送到后台数据库中，调用百度API接口和数据库中的用户照片做对比，测试出照片是哪个用户的，并把用户信息和相似度的结果返回到界面上。

