# 单元测试设计

## base64编解码测试：

@Test  
public void base64Test01() throws Exception{  
 InputStream inputStream=new FileInputStream("E:\\MicrosoftEdgeDownloads\\timg.jpg");  
 OutputStream outputStream=new FileOutputStream("E:\\MicrosoftEdgeDownloads\\a.jpg");  
  
 //文件读入缓存并编码  
 byte[] buf=new byte[inputStream.available()];  
 inputStream.read(buf);  
 //编码  
 String s=new String(Base64.getEncoder().encode(buf));  
 System.out.println(s);  
 //解码，并写入文件  
 byte[] buf1= Base64.getDecoder().decode(s);  
 outputStream.write(buf1);   
  
 outputStream.close();  
 inputStream.close();  
}

## 使用okhttp进行网络请求设计：

@Test  
public void test03() throws Exception{  
 String path = "E:\\MicrosoftEdgeDownloads\\timg.jpg";  
 OkHttpClient client = new OkHttpClient();  
 // 上传文件域的请求体部分  
 RequestBody fileBody = RequestBody.create(new File(path), MediaType.parse("image/jpeg"));  
 // 整个上传额请求体部分(普通表单+文件上传域)  
 RequestBody requestBody = new MultipartBody.Builder()  
 .setType(MultipartBody.FORM)  
 .addFormDataPart("title","kk kk")  
 //filename:avatar,originname:abc.jpg  
 .addFormDataPart("avatar","timg.jpg",fileBody)  
 .build();  
 // 构造网页请求  
 Request request = new Request.Builder()  
 .url("http://localhost:8000/upload")  
 .post(requestBody)  
 .build();  
 Response response = client.newCall(request).execute();  
 System.out.println(response.body().string());  
}

## 解析json数据测试

@Test  
public void parseJson(){  
 String jsonStr = "{\n" +  
 " \"result\": {\n" +  
 " \"face\_num\": 1,\n" +  
 " \"face\_list\": [\n" +  
 "\t{\n" +  
 " \"beauty\": 55.61,\n" +  
 " \"liveness\": {\"livemapscore\": 0.33},\n" +  
 " \"angle\": {\n" +  
 " \"roll\": 32.28,\n" +  
 " \"pitch\": 11.45,\n" +  
 " \"yaw\": -52.64\n" +  
 " },\n" +  
 " \"face\_token\": \"cdb14ebb8928e9d8bb5aa85e2e8d3251\",\n" +  
 " \"location\": {\n" +  
 " \"top\": -5.35,\n" +  
 " \"left\": 342.29,\n" +  
 " \"rotation\": 28,\n" +  
 " \"width\": 329,\n" +  
 " \"height\": 315\n" +  
 " },\n" +  
 " \"face\_probability\": 1,\n" +  
 " \"age\": 23\n" +  
 " }]\n" +  
 " },\n" +  
 " \"log\_id\": 8445056575996,\n" +  
 " \"error\_msg\": \"SUCCESS\",\n" +  
 " \"cached\": 0,\n" +  
 " \"error\_code\": 0,\n" +  
 " \"timestamp\": 1572956790\n" +  
 "}";  
 try {  
 JSONObject object = new JSONObject(jsonStr);  
 JSONObject rusultObj = object.getJSONObject("result");  
 JSONArray face\_list = rusultObj.getJSONArray("face\_list");  
 JSONObject face\_list\_0 = face\_list.getJSONObject(0);  
 double beauty = face\_list\_0.getDouble("beauty");  
 String face\_token = face\_list\_0.getString("face\_token");  
 int age = face\_list\_0.getInt("age");  
 System.out.println(beauty+"=="+face\_token+"=="+age);  
  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
}

## 人脸对比测试

@Test  
public void compare(){  
 try {  
 InputStream is = new FileInputStream("E:\\MicrosoftEdgeDownloads\\test1.jpg");  
 String str1 = FIleBase64Str.getBase64Str(is);  
 is = new FileInputStream("E:\\MicrosoftEdgeDownloads\\test1.jpg");  
 String str2 = FIleBase64Str.getBase64Str(is);  
 FaceOpreationUtils.faceCompare(str1,str2);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
}

# 类设计

## UI类设计

* 1. loginButton类：自定义button，为button添加动画效果。

### MyImageView 类：自定义ImageView，实现了在ImageView上画框和写字的功能。

### 1.3 MyImageView2类：自定义ImageView，在MyImageView基础上扩展了可以绘制多个画框和文字功能。

## Utils类设计

### 2.1AipFace\_Instance类：

### 单例获取AIpFace对象。

### 2.2FaceOpreationUtils类：

### 调用百度ai人脸识别JDK实现人脸注册，人脸检测，人脸对比，人脸搜索的功能。

### 2.3FIleBase64Str类：

### 将Bitmap和InputStream对象转化为base64编码并压缩图片工具类。

### 2.4ImgLocation类：

自定以MyImageView中绘制的画框个人脸信息包装类。

### 2.5ParseJson类：

处理人脸检测，人脸注册，人脸识别，人脸搜索返回的Json信息，提取其中所需要展示在UI界面上的信息。

## Activity类设计

### 3.1MainAcivity类：

App主界面，可以跳转至各个子功能界面.

### 3.2RegisterByCamera类:

通过拍照上传人脸界面,其中包括跳转至通过选择照片上传人脸界面切换按钮.

### 3.3RegisterByImage类:

通过选择本地照片上传人脸界面.

### 3.4DetectByCamera类:

通过拍照检测人脸界面,其中包括跳转至通过选择照片检测人脸界面切换按钮.

### 3.5DetectByImage类:

通过选择本地照片检测人脸界面.

### 3.6CompareByImage类:

选择两张本地照片进行人脸对比,返回比对结果.

### 3.7SearchByCamera类:

通过拍照搜索人脸界面,其中包括跳转至通过选择照片搜索人脸界面切换按钮.

### 3.8SearchByImage类:

通过选择本地照片搜索人脸界面.

# 软件安装及使用说明

## 软件安装:

由于本软件为打包apk,因此只能通过andorid studio运行源文件,通过android studio进行安装.

## 使用说明:

安装完毕后,点击图标进入主界面,选择相应的功能进行下一步操作.

1. 人脸注册:

通过拍照或选择照片进行人脸注册,注册时需要填写自己的学号和姓名.

### 人脸检测:

通过拍照或选择照片进行人脸检测,检测成功会返回对应的年龄和美丽值评分.

### 人脸对比:

选择两张本地的图片进行人脸比对,返回两张图片的对比相似结果.

### 人脸搜索:

使用人脸搜索功能,需要先进行人脸注册,注册成功后便可以通过拍照或选择照片进行人脸搜索,返会对应的结果.