## 数据库设计

### 1.Photo

{

\_id:String

name:String,

content:String,

createTime:Number

}



### Face

{

\_id:String

groupId:String,

userId:String,

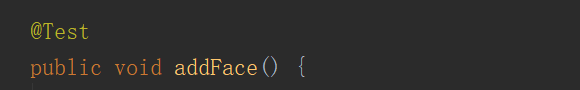
userInfo:String

}

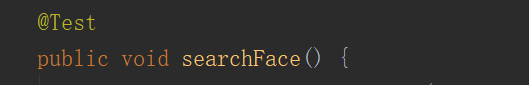


## 2.单元测试设计

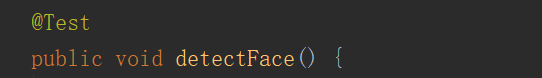
### 1.人脸库添加



### 2.人脸搜索



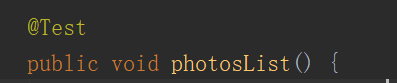
### 3.人脸检测

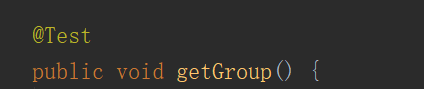


### 4.人脸对比



### 5.获取图片列表、用户组列表





### 6.获取Token



## 3. 类设计

### 1. MainActivity:

创建四个主要功能模块：图片上传、图片列表、人脸识别、其他识别，每个模块属于一个Fragment

### 2. IndexFragment

上传照片模块：包括相机拍照和相册选择照片上传

### 3. ListFragment

图片列表模块：采用recycleView展示图片列表，且可支持大图预览展示，侧滑转入识别、删除功能。

### 4. FaceFragment

人脸识别模块：人脸添加、人脸搜索、人脸检测、人脸对比

### 5. LastFragment

其他识别模块：动物识别、植物识别

### 6. ImageAdapter

图片列表recycleView的适配器

### 7. PhotoLoader

图片预览功能中的图片加载方式

### 8.Face、Photo

APP中的人脸和图片的实体类

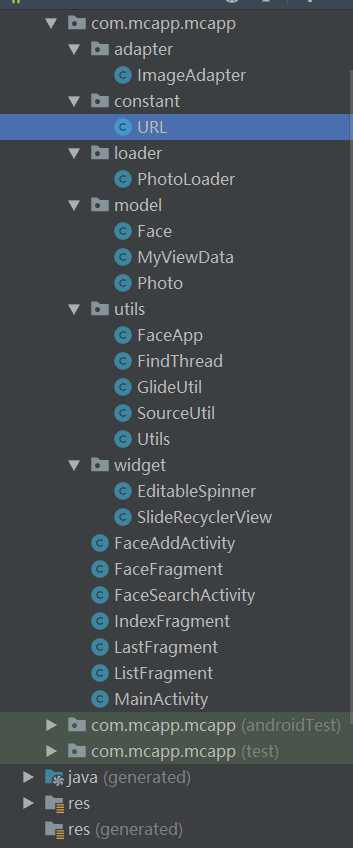
### 9. FaceApp、FindThread、GlideUtil、SourceUtil

人脸识别线程实现

### 10．URL

APP中接口链接声明

### 整体类设计图片



## 4. 本地接口设计

### 1.创建主功能模块

TabViewChild tabViewChild01 = new TabViewChild(R.drawable.*tab01\_sel*, R.drawable.*tab01\_unsel*, "首页", indexFrag);  
TabViewChild tabViewChild02 = new TabViewChild(R.drawable.*tab03\_sel*, R.drawable.*tab03\_unsel*, "列表", listFrag);  
TabViewChild tabViewChild03 = new TabViewChild(R.drawable.*tab05\_sel*, R.drawable.*tab05\_unsel*, "人脸识别", faceFrag);  
TabViewChild tabViewChild04 = new TabViewChild(R.drawable.*tab02\_sel*, R.drawable.*tab02\_unsel*, "其他识别", lastFrag);

### 2.1拍照接口

//拍照  
public void photo(View view) throws Exception{

#### 2.2打开相册

//打开相册,进行照片的选择  
private void openGallery() {

#### 2.3申请权限

//申请权限调用相册  
private void gallery() {

#### 2.4申请权限回调

@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

#### 2.5选择图片回调

//拍照、相册回调  
@Override  
public void onActivityResult(int requestCode, int resultCode, Intent data) {

#### 2.6图片压缩

//图片压缩  
public static Bitmap compressImage(Bitmap image) {

#### 2.7片上传

//文件上传的处理  
public void upload(View view) {

### 3.1初始化图片列表数据

private void initData() {

public void displayImage(Object src, ImageView imageView, LoadCallback callback) {

#### 3.2添加图片列表数据更新监听

public void addListener() {

#### 3.3图片列表添加数据

//添加图片  
public void addPhoto(Photo p) {

#### 3.4图片列表删除数据

//删除图片  
public void delete(final String id) {

### 4.1设置图片源、imageview添加bitmap

public void setImg(Photo photo) {

public void setBitmapImg(Bitmap bitmap) {

#### 4.2脸库添加

FaceApp.*addFaceWithBase64*(imageBase64,groupId,userId,userInfo,this);

#### 4.3人脸搜索

FaceApp.*searchFaceWithBase64*(imageBase64,groupId,this);

#### 4.4人脸检测

new FindThread(photo.getImagesByte(), 0, this).start();

#### 4.5人脸对比

new FindThread(SourceUtil.*imagesByte*,photo.getImagesByte(), 3, this).start();

### 5.1动植物识别

new FindThread(photo.getImagesByte(), 1, this).start();

new FindThread(photo.getImagesByte(), 2, this).start();

## 5. restful网络接口设计

### 1.图片接口：

|  |  |  |
| --- | --- | --- |
| 描述 | URL | Method |
| 图片增加 | /upload | post |
| 图片删除 | /upload/id | delete |
| 图片列表 | /upload | get |

### 2.用户接口：

|  |  |  |
| --- | --- | --- |
| 描述 | URL | Method |
| 人脸增加 | /face | post |
| 组列表 | /face | get |