首先是项目后端开发采用ssm架构，也就是mapper层、service层和controller层，具体的开发框架采用的是springboot，并在本地持久化上采用的技术是mybatis及mysql。结合接口文档和实际开发需要，现设计以下关于后端的数据结构：

首先是**实体类**：

* **ImageInfo**: 存储图片信息的实体类。
  + 属性:
    - url: 照片路径 (String)
    - id：照片id（int）
    - name: 照片名 (String)
    - photoTime: 拍摄时间 (Date)
    - uploadTime：上传时间（LocalDateTime）
    - size: 照片大小 (String)
    - place: 拍摄地点 (String)
    - device: 设备信息 (String)
* **Classification**: 存储分类信息的实体类。
  + 属性:
    - name: 类别名 (String)
    - images: 该类别中的照片列表 (List)
* Result：用于返回前端数据。
  + 函数体：public class Result<T>{ }
  + 属性：
    - private Integer code;//业务状态码 0-成功 1-失败
    - private String message;//提示信息
    - private T data;//响应数据

然后是关于**Mapper层**，该层负责与数据库的交互打交道，包括：

* **ImageMapper**: 与数据库交互的接口，负责图片信息的增删查改。
  + 方法:
    - insertImage(ImageInfo image): 插入图片信息
    - deleteImage(String url): 删除指定路径的图片
    - selectImages(): 查询所有图片信息
    - selectImagesByTime(): 按时间顺序查询图片信息
    - selectImagesByOther(): 按其他方式查询图片信息 (根据实际需求设计)

在**Service层**上，负责业务逻辑处理。包括：

**Service接口**

* **ImageService**: 业务逻辑处理接口，封装对图片信息的操作。
  + 方法:
    - importImages(MultipartFile image): 导入图片
    - sortImages(List<String> urls, String option): 排序图片
    - deleteImage(int id): 删除图片
    - getClassifications(String option): 获取分类信息
    - searchImages(String key): 搜索图片

以及**Service实现类**：

* **ImageServiceImpl**: 实现 ImageService 接口，完成具体业务逻辑。
  + 需要调用 ImageMapper 接口进行数据库操作。
  + 根据文档描述，需要实现图片导入、排序、删除、分类预览和搜索等功能。

在Controller层上，用于处理前端请求并返回响应，包括有：

* **ImageController**: 接收前端请求，调用 Service 层方法，返回响应数据。包括但不限于：
  + public Result addImage(MultipartFile image) //上传图片
  + public Result deleteImage(@RequestParam int id) //删除图片

具体的后端实现将在本周内完成。

这里附有关于该后端的总体架构设计导图：



