**系统架构指导**

* 1. 系统分层架构
* 表现层：用户交互界面，包括导入界面、预览界面、图片细节展示界面。前端技术上我们小组采用开发vue.js技术用于构建用户界面。
* 业务逻辑层：负责处理相片分类、文件操作、缓存管理等核心功能。计划的使用语言是Python，采用微服务架构，将不同模块（如读取删除模块、分类模块）分离成独立服务，方便后续扩展和维护。
* 数据层：负责持久化存储和读取文件的元数据及分类信息。计划本地采用MySQL来存储结构化元数据，并采用Gauss数据库进行云存储。
  1. 模块化设计
* 缓存管理模块：管理相片路径的缓存，支持批量导入、删除功能。提供接口与业务逻辑层交互。
* 分类模块：实现文件属性读取和日期分类功能。后续设计接入AI分类，支持更多分类方式（如按地点、标签分类）。
  1. 数据流设计
* 导入流程：用户通过导入接口上传图片，系统存储图片并记录元数据（文件大小、创建时间等），更新缓存管理模块。
* 分类流程：当用户请求分类时，分类模块根据元数据执行分类操作，将分类后的结果存储在数据库中，并更新预览界面。
* 预览和细节展示流程：预览界面从缓存或数据库中获取图片元数据，按需加载图片并展示。细节展示模块提供单张图片的详细信息，支持删除和重新分类操作。
  1. 计划安排

**第一周**：完成表现层、缓存管理模块的基础开发。

**表现层开发 (前端开发)**

* **任务**：
  + 使用Vue.js开发用户界面，重点是导入界面、预览界面和图片细节展示界面的基本布局。
  + 实现基础交互逻辑，如图片上传按钮、图片列表显示、基本的删除功能。

**缓存管理模块开发**

* **任务**：
  + 使用Python实现相片路径缓存管理功能，支持批量导入和删除操作。
  + 提供接口与表现层交互，返回上传图片的路径和相关元数据。

**第二周**：完成数据库的配置，实现分类模块开发与接口扩展。

**数据层初始化**

* **任务：**
  + **在本地使用MySQL初始化数据库，创建表结构以存储图片元数据。**
  + **配置Gauss数据库的云端存储，并确保与本地MySQL数据库的同步。**

**分类模块开发**

* **任务**：
  + 实现文件属性读取功能，提取图片的大小、创建时间等元数据。
  + 实现按日期分类的核心逻辑，将分类结果存储到MySQL数据库中。

**扩展表现层与业务逻辑层的接口**

* **任务**：
  + 在表现层添加分类功能的UI元素，并实现相关的前端逻辑。
  + 在业务逻辑层扩展接口，支持前端请求分类操作，并返回分类结果。

**第三周**：细节优化、性能测试与部署准备

**系统优化与细节完善**

* **任务**：
  + 优化缓存管理模块，提升批量操作的效率，减少处理时间。
  + 完善图片细节展示功能，增加更多元数据展示项（如文件大小、分类标签）。

**性能测试**

* **任务**：
  + 对系统进行负载测试，评估在大规模图片数据下的系统响应时间和性能。
  + 优化数据库查询性能，特别是分类查询和预览界面的加载速度。

**最终集成与用户测试**

* **任务**：
  + 进行最终的集成测试，覆盖所有功能模块。
  + 安排用户进行系统测试，收集反馈进行最后的调整和修复。