

系统架构如图所示，为C/S架构，分为展示层（APP）、传输层、安全层、服务层、存储层，各层详细内容如下所示：

1. 展示层：
2. 非风格转化处理：滤镜处理和图片贴纸处理均在展示层（APP）上进行。
3. 风格转化预处理：图像风格转化之前的用户预处理工作，选区、裁剪、人像轮廓识别均在展示层（APP）上进行。
4. 管理用户信息：包括登录注册、查看分享内容。
5. 社交：包括点赞、关注、收藏。
6. 传输层：采用HTTP1.1协议进行通讯，主要使用协议中的POST、GET、PUT、DELETE方法。
7. 安全层：使用OAuth2.0协议作为用户登录的认证。用户登录网站获取一个令牌，每粗发送请求时将令牌放至Header中，以供服务器进行验证身份。
8. 服务层：
9. API服务：系统架构为C/S架构，所有请求服务均建立在HTTP之上，系统制定并编写一套符合Restful规则的API接口，以供客户端获取资源使用，API服务使用Python的Flask框架进行开发。
10. 云推送服务：使用第三方推送平台，当用户风格转化完成之后，系统调用第三方推送系统将消息推送至客户端，以提醒用户完成转化。
11. GPU计算：图像风格转化需要进行大量计算，光靠CPU进行计算远远不够，因此需要GPU进行计算，GPU在图像矩阵方面的计算速率大约是同等级别CPU的60倍左右，因此图像风格转化选用GPU计算。由于每次计算时间至少在1分钟以上，并且每次风格转化均为独占设备，所以不能实时响应用户的转化需求，故需要队列等候。GPU计算所使用的计算框架为基于Tensorflow计算引擎的Keras框架。
12. 存储层：
13. MySQL：MySQL用于存储关系型数据，使用SQL语句对数据库进行增、删、改、查等操作。
14. 非结构化资源存储：系统将会保存大量图像资源，因此需要专门的存储空间对图像资源进行存储，转化为静态资源，访问图像资源则是以URL的形式进行访问。