

IMANGINE 需求分析规约

ID: GF-SRAS

小组成员

1352845 郭雨卿

1352878 王泽宇

1353005 杨安琪

1354361 刘林青

修订历史

编写日期	SEPG	版本	说明	作者	评审时间	参与人员	批准日期	确认人员
2015-12-15	GF	0.1	初稿	刘林青 杨安琪				
2015-12-20	GF	0.2	定稿	刘林青 杨安琪				

目 录

修订历史.....	202
1 引言.....	5
1.1 背景.....	5
1.1.1 项目特色.....	5
1.1.2 必要性.....	5
1.2 参考资料.....	5
1.3 假定和约束.....	5
1.4 用户特点.....	5
4.运行环境规定.....	6
4.1 设备.....	6
4.2 与第三方平台接口.....	6
4.3 普通接口.....	6
2. 功能需求.....	7
2.1 系统范围.....	7
2.2 系统总体流程.....	7
2.3 系统体系结构.....	9
2.4 需求分析.....	11
2.4.1 功能建模.....	11
2.4.2 数据建模.....	27
2.4.3 行为建模.....	29
3. 非功能需求.....	49
3.1. 性能要求.....	49
3.1.1. 用户体验.....	49

3.1.2. 时间特性要求	50
3.1.3. 输入输出要求	50
3.2. 数据管理能力要求	52
3.2.1 常量约定	52
3.2.2 问卷和记录个数	52
3.3.3 数据存储要求	52
3.3.4 负载	52
3.3. 安全及保密性要求	53
3.3.1 软件使用数据	53
3.3.2 用户账户数据	53
3.4. 灵活性要求	53
3.5. 其他专门要求	54
3.5.1 可维护性	54
3.5.2 灾难恢复	54
3.5.3 法律限制	54

1 引言

1.1 背景

1.1.1 项目特色

人们遇到喜欢的图片时会希望保存到某地以备将来使用,并且希望能够网罗相关主题的图片素材并且查看志同道合的朋友的图片开阔视野,这一需求对于前端开发人员或者设计师来说尤为重要。我们希望通过开发这个网站满足此类用户的需求,为用户提供分享和发现图片的平台。

1.1.2 必要性

市场上现有的搜索引擎针对文字搜索已经达到很高的精度,然而对于图片搜索的技术仍旧不是很令人满意,虽然已经有类似于“以图找图”的应用推广,也只能找到一部分图片。本软件可以把平时用户喜欢的图片完整地分门别类搜集起来,并且在一定程度上缩小了图片搜索的范围,更能为用户的使用提供方便。除此之外,与第三方平台的接口支持这些图片快速的转发传播,使得用户在网站之外的地方也能获取网站中的图片资源。

1.2 参考资料

1. Pressman R S. Software Engineering A Practitioner's Approach[M]. Seventh Edition. McGraw- Hill Higher Education, 2008.

1.3 假定和约束

发布时间: 2016 年 1 月 10 日发布 Alpha 测试版

浏览器要求: safari 6.2.1 及以上版本,chrome47.0.2526.107 及以上版本,Firefox 43.0.2.5833 及以上版本。

开发条件: 成员自备开发使用电脑,并配置相关开发环境

经费支持: 无经费支持

设计成本: 自行承担设计成本

开发效率: 采用用 SCRUM 敏捷开发模型,开发冲刺期间避免外界干扰

1.4 用户特点

1. 针对人群包括经常用到图片素材的前端开发人员和设计人员,热爱图片收藏和品鉴的年轻人等。

2. 用户会经常性的访问该网站，上传图片或收藏图片，每天会访问多次网站。
3. 用户之间的互动比较强，经常互访对方主页或收藏对方图片。

4.运行环境规定

4.1 设备

详情请见 SRS 2.2 节运行行环境部分。

1. 浏览器要求:safari 6.2.1 及以上版本,chrome47.0.2526.107 及以上版本,FireFox 43.0.2.5833 及以上版本。
2. CPU:AppleA4 以上,主频 1GHz 以上,ARM 架构
3. 主存:512M 以上
4. 系统要求:iOS 7.0 及以上版本

4.2 与第三方平台接口

1. 微信公众开发平台接口 api
2. 腾讯 QQ 接口 API
3. 新浪微博接口 api

4.3 普通接口

1. 账户授权协议:OAuth 2.0
2. 网网络访问协议:TCP/IP,HTTP
3. 数据压缩:objective-zip

2. 功能需求

2.1 系统范围

人们遇到喜欢的图片时会希望保存到某地以备将来使用，并且希望能够网罗相关主题的图片素材并且查看志同道合的朋友的图片开阔视野，这一需求对于前端开发人员或者设计师来说尤为重要。我们希望通过开发这个网站满足此类用户的需求，为用户提供分享和发现图片的平台。

2.2 系统总体流程

本系统拥有三个主要的业务流程，包括浏览业务、社交业务和个人主页管理业务，分以下三个业务流程图进行展示说明。

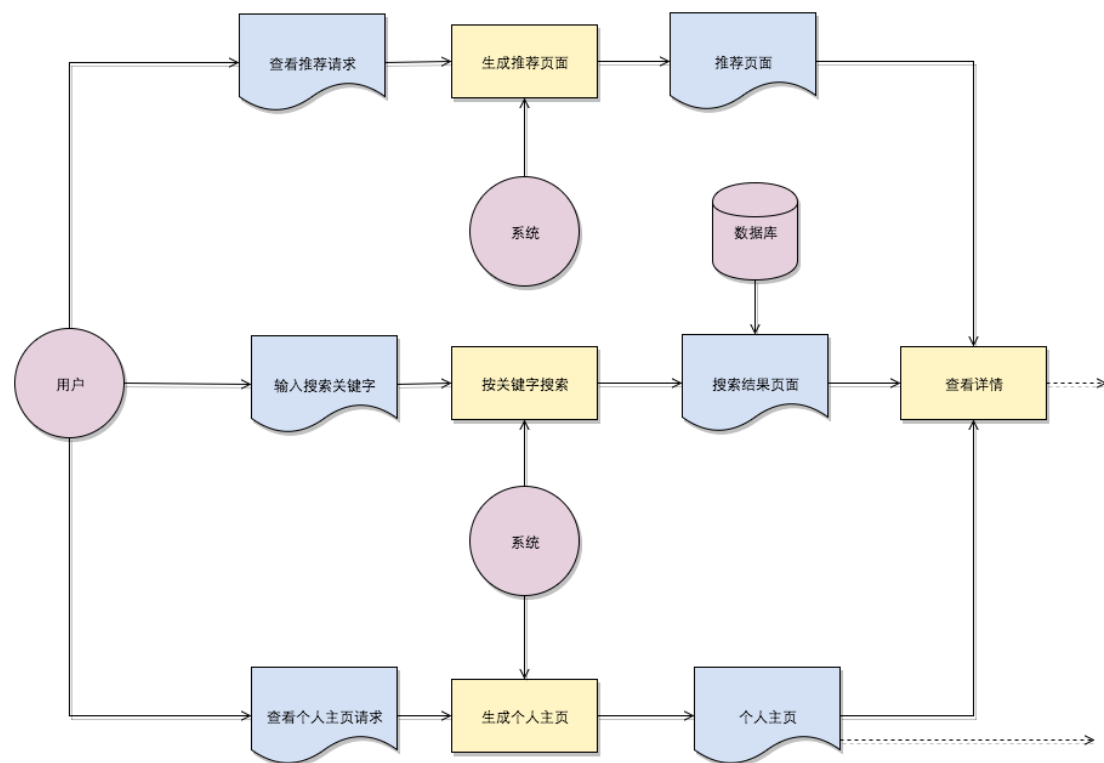


图1 系统业务流程图 - 浏览业务

浏览业务：

用户进入 IMENGINE 网站首页，即产生查看推送请求，系统将会根据推荐算法，为用户定向推送图片、专辑和小组，推荐页面以列表的形式展现，用户可以点击进一步查看详情；用户亦可以输入搜索关键字，系统将通过图片的标签进行精确、模糊搜索，并将搜索结果显示到网页；用户点击自己或他人头像，即产生了查看个人主页请求，系统将返回与该用户有关的图片、专辑和小组信息，用户可以在其个人主页上查看。

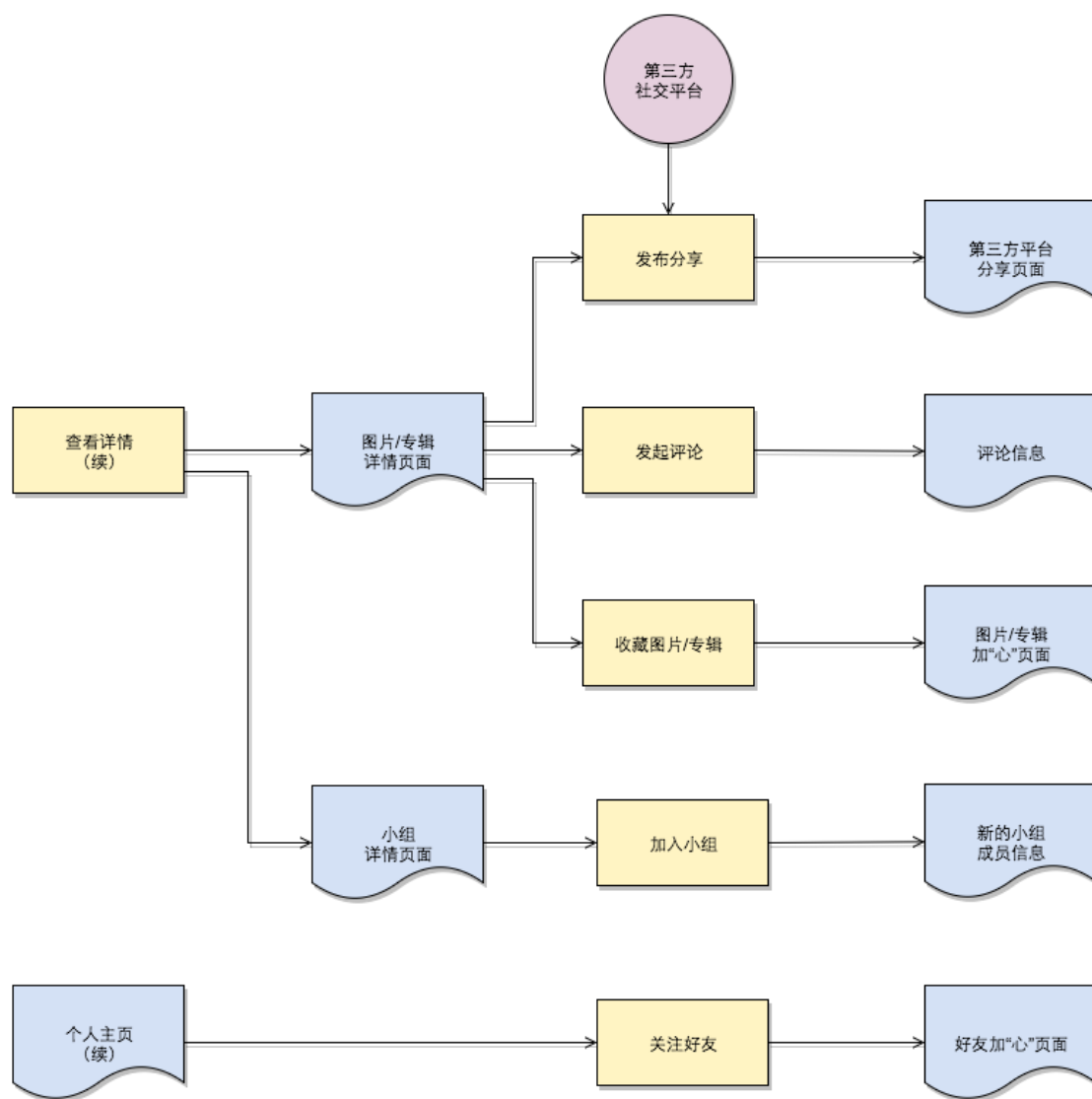


图2 系统业务流程图 -- 社交业务

社交业务：

用户进入图片或专辑的详情页面后，可以点击“分享”，将图片分享到第三方社交平台；可以发起评论；可以收藏该图片（专辑），收藏后图片页面上的红心变成实心，标志成功收藏；用户进入小组详情页面后，可以请求加入小组，页面将更新小组成员信息并显示在页面上；用户进入个人页面后，可以关注该用户，关注成为好友之后，页面的红心变成实心，标志关注成功。

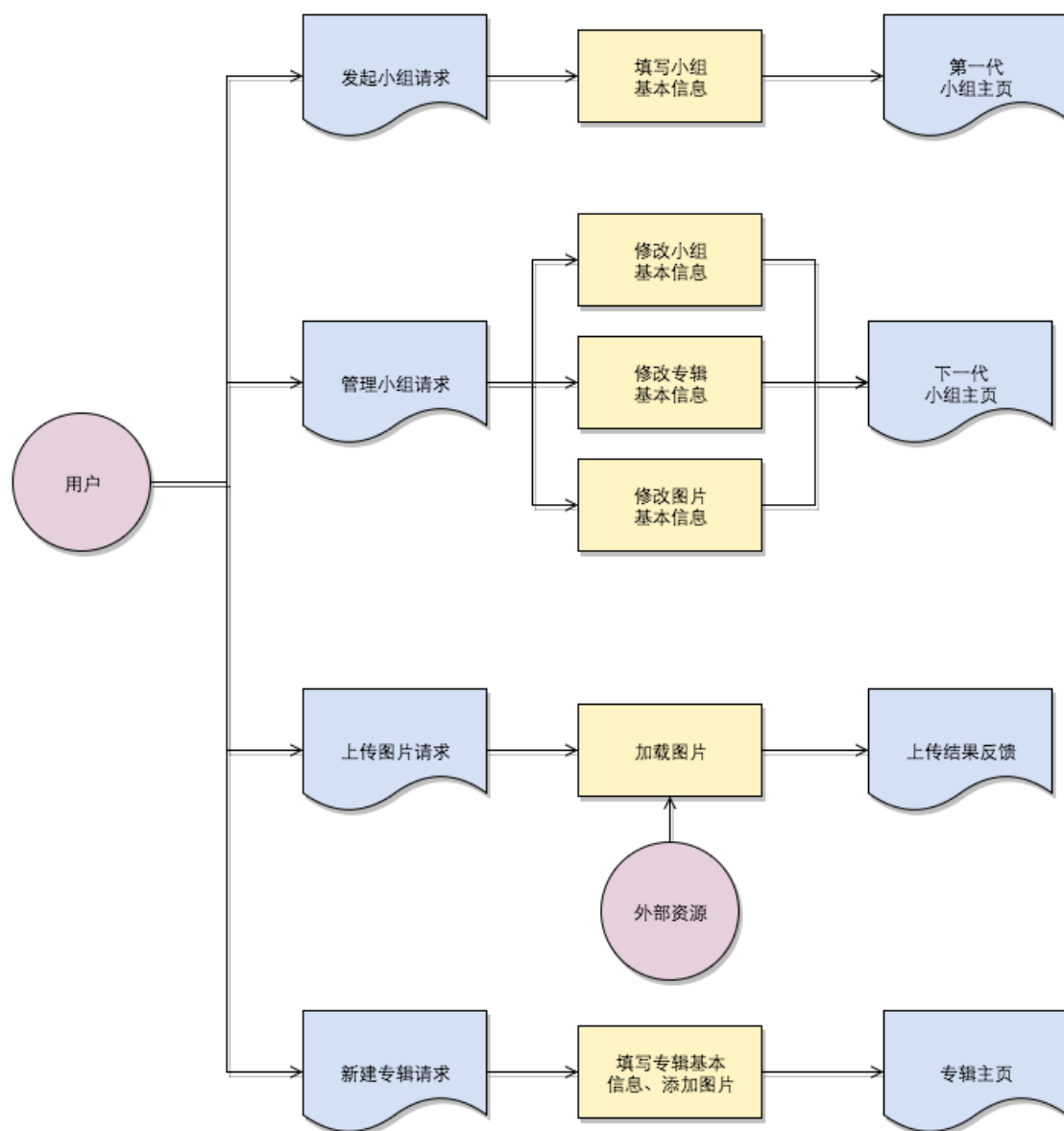


图3 系统业务流程图——个人主页管理业务

个人主页管理业务：

用户在其个人主页上，可以发起小组，填写小组基本信息之后，页面将显示新建小组的信息（第一代小组主页）；用户可以发起管理小组请求，用户可以修改小组基本信息、修改小组中的专辑的信息或修改小组中的图片的信息，管理完成后，页面将显示修改信息后的小组主页（下一代小组主页）；用户可以上传图片，从本地 PC 机或者其他网页链接加载成功后，成功上传图片；用户可以新建专辑，填写专辑基本信息、添加专辑封面图片之后，新建成功，页面显示专辑的主页。

2.3 系统体系结构

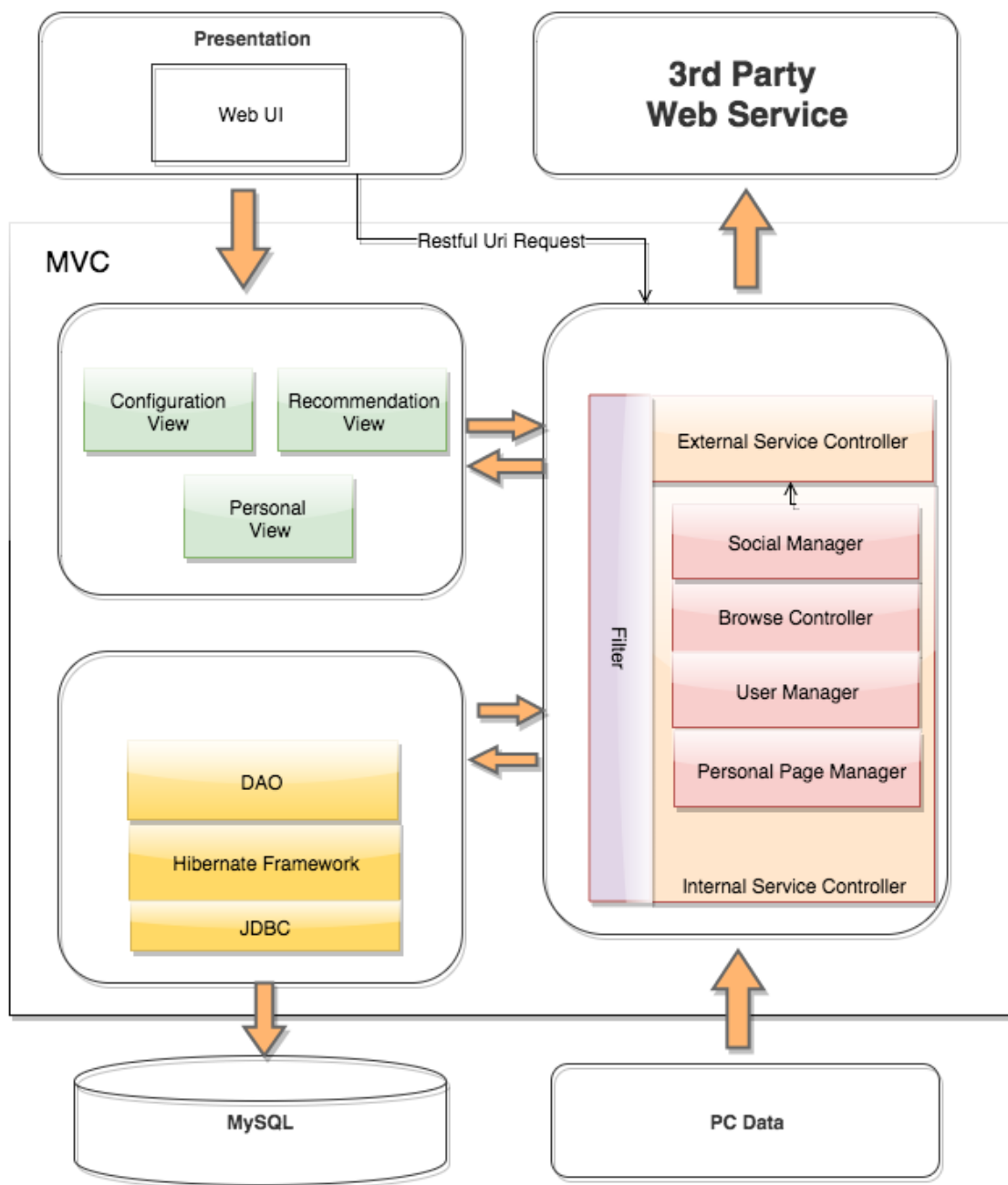


图 4 系统总体架构图

本系统由服务器端和客户端两部分组成。服务器端设计以实现各模块间的低耦合及模块内部的高内聚为导向，采用 MVC 作为总体架构；客户端向服务端的请求采用 Restful 设计风格，力求访问的灵活性；服务器端访问 MySQL 数据库进行数据交互；同时在系统层面设计了安全保障模块、缓存模块、日志和验证模块，为系统平台的安全和高效保驾护航。

2.3.1 数据模块

本系统作为基于 Web 的图片分享平台，对于大量的高清图片数据、专辑信息、小组信息、用户个人信息、用户访问记录等数据的大量存储和快速访问有着较高的要求。本系统使用 MySQL 5.6 关系型数据库存储信息，通过 JDBC 完成对数据库的访问。MySQL 数据库作为一个快速、多线程编程的数据库，不仅满足本系统对于访问速度的要求，而且具有强大的查询功能，

可以在同一查询中混用来自不同数据库的表，查询方便快捷。本系统支持用户上传本地图片，支持 PNG，BMP，JPEG，JPG 等常见图片格式，上传之后存入 MySQL 数据库。

2.3.2 模型层

模型层负责处理本系统中应用程序的数据逻辑部分，对数据对象进行存取操作。本模块通过 JDBC 访问 MySQL 数据库，采用 Hibernate 框架自动完成数据库表到 Java 类的映射。为了构建一个健壮的 J2EE 应用，本模块采用 DAO 作为接口，统一管理上层逻辑对于数据源的访问操作，使得上层逻辑无需关心数据访问的细节，同时扩展了系统的可维护性、可拓展性。

2.3.3 控制器层

控制器层是本系统处理用户请求的核心层，负责对用户的 http 请求作出响应，处理本系统的核心业务逻辑，转发用户请求。控制器层通过 Filter 过滤来自用户的请求，检查用户请求的数据格式及用户当前的登录状态，并根据判断结果对请求进行转发。用户 2 请求被转发到内部服务、外部服务两大模块，外部服务负责获取与第三方社交平台的接口，并将用户图片分享到第三方社交平台；内部服务分为社交管理、浏览管理、个人账户管理、个人主页管理四大部分，是本系统的核心功能逻辑。

- 社交管理：负责处理收藏、评论、分享、关注好友、加入小组等社交业务。
- 浏览管理：负责获取用户个人和系统推荐的图片、专辑、小组数据。
- 个人账户管理：负责登录、登出以及个人信息的配置管理业务。
- 个人主页管理：负责个人图片、专辑和小组的配置管理业务。

2.3.4 视图层

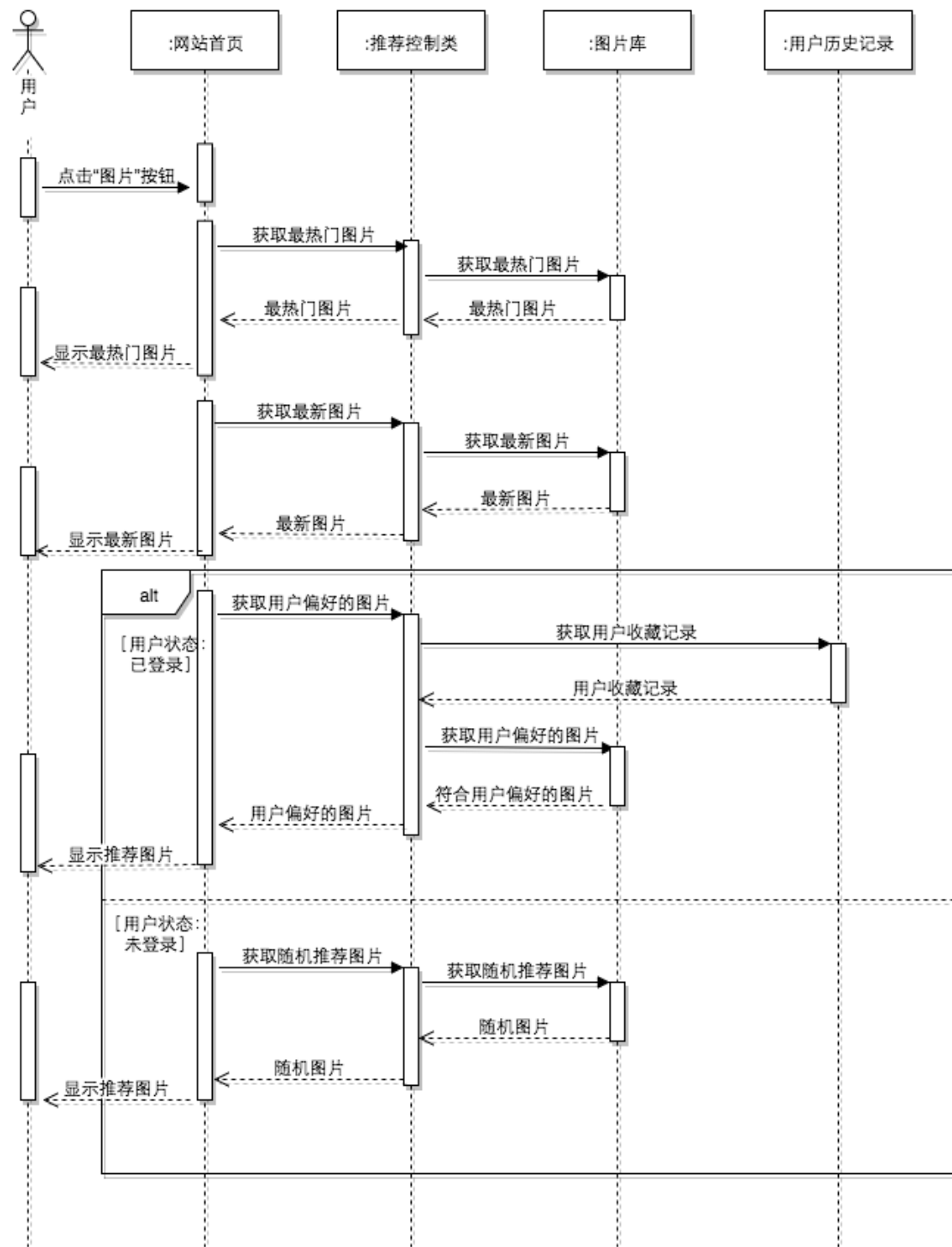
视图层是本系统的数据显示部分，是与用户交互的接口。负责系统的前端业务逻辑，支持 CSS，JavaScript，JSP 文件，将用户请求转发给 Controller 层，并显示经过 Controller 处理的业务。

2.4 需求分析

2.4.1 功能建模

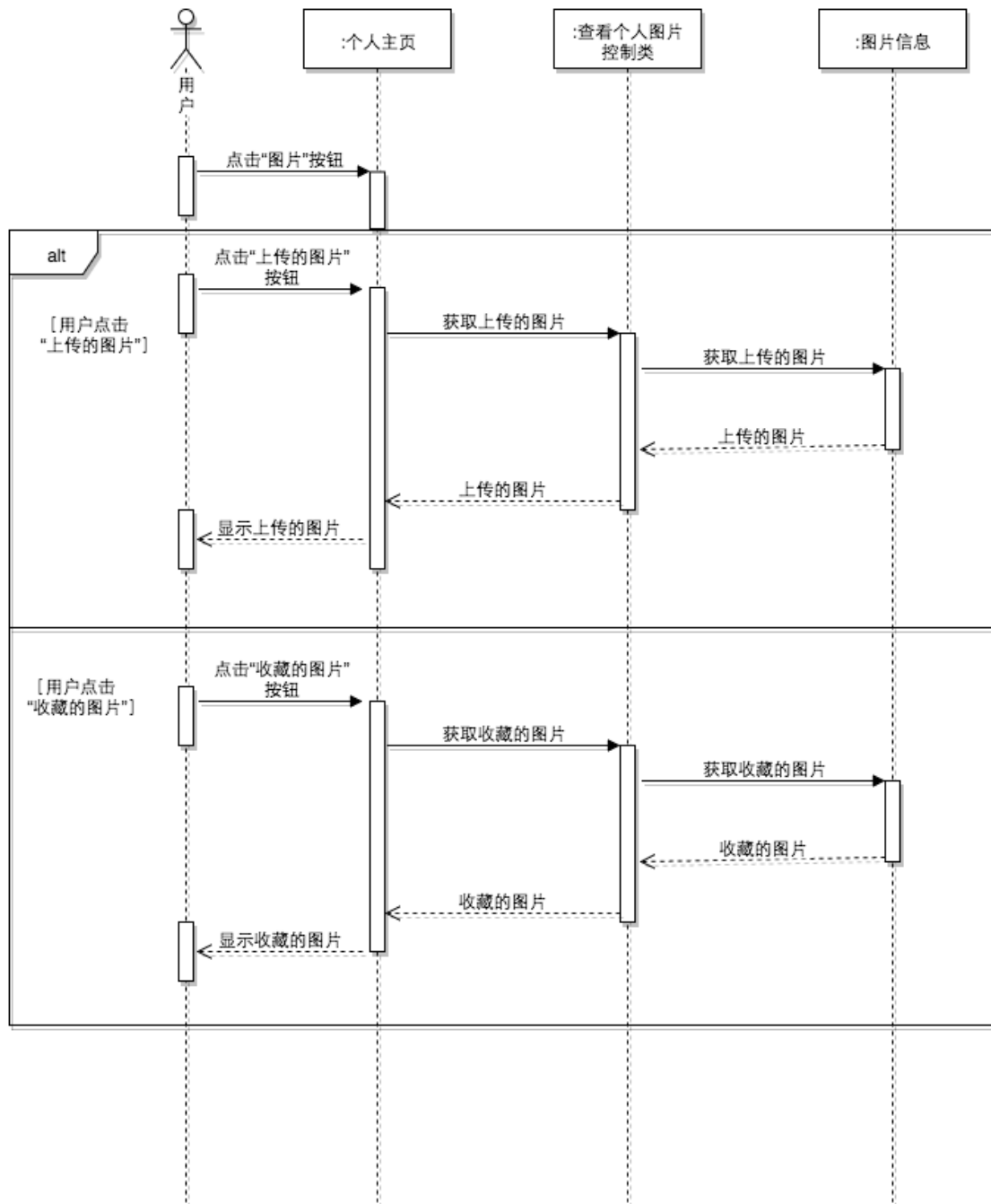
2.4.1.1 浏览模块

推荐图片：



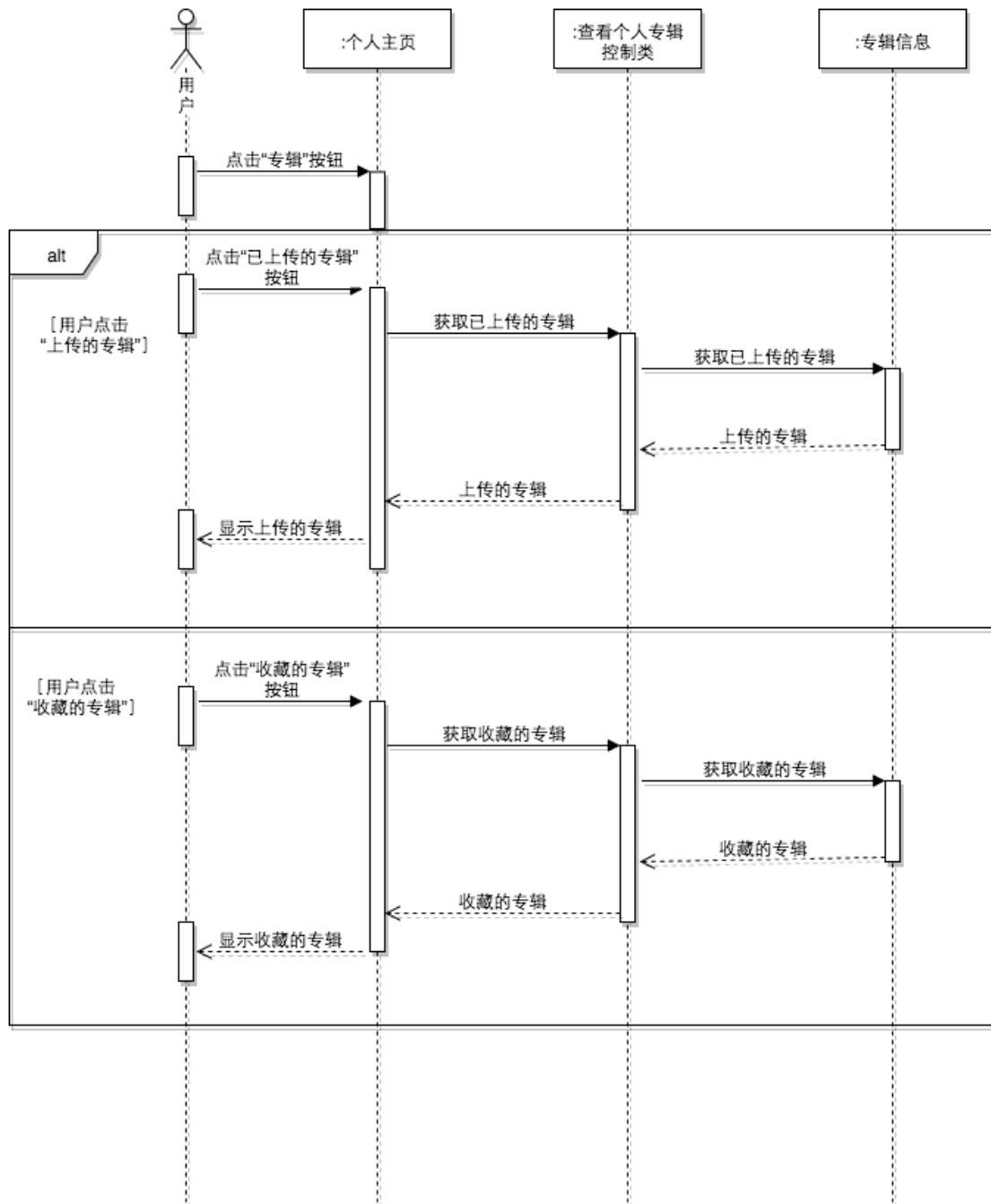
用户在网站首页上，点击“图片”按钮，系统将分别获取最热门图片、最新图片并显示给用户；若用户已登录，系统获取用户的收藏记录，依此判断用户的偏好，并获取符合用户偏好的图片，显示在页面上；若用户未登录，随机推荐图片，并显示在页面上。

查看个人图片：



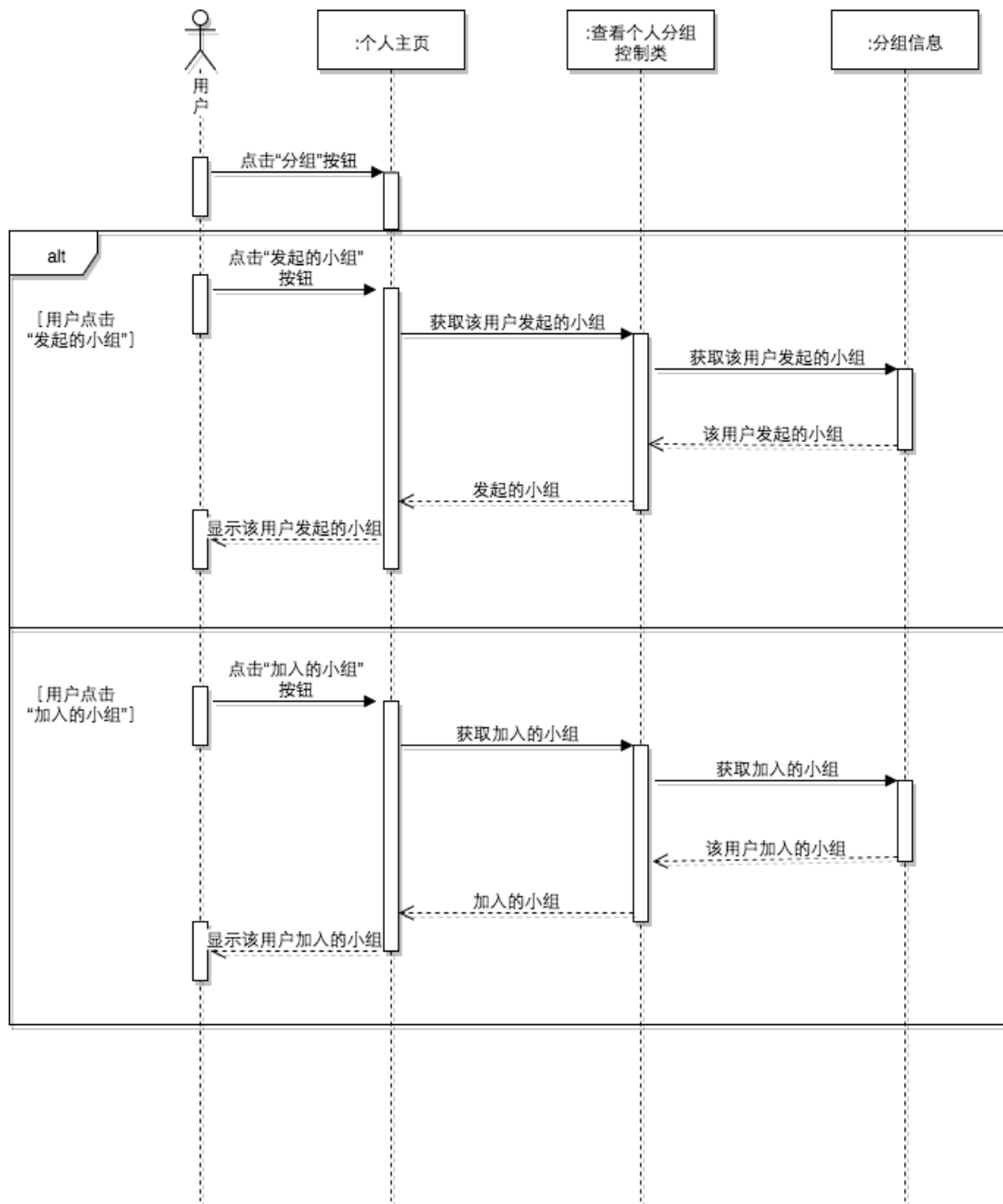
用户在个人主页上，点击“图片”按钮，如果选择查看该用户“上传的图片”，系统将获取并展示该用户上传的图片；如果选择查看该用户“收藏的图片”，系统将获取并展示该用户收藏的图片。

查看个人专辑：



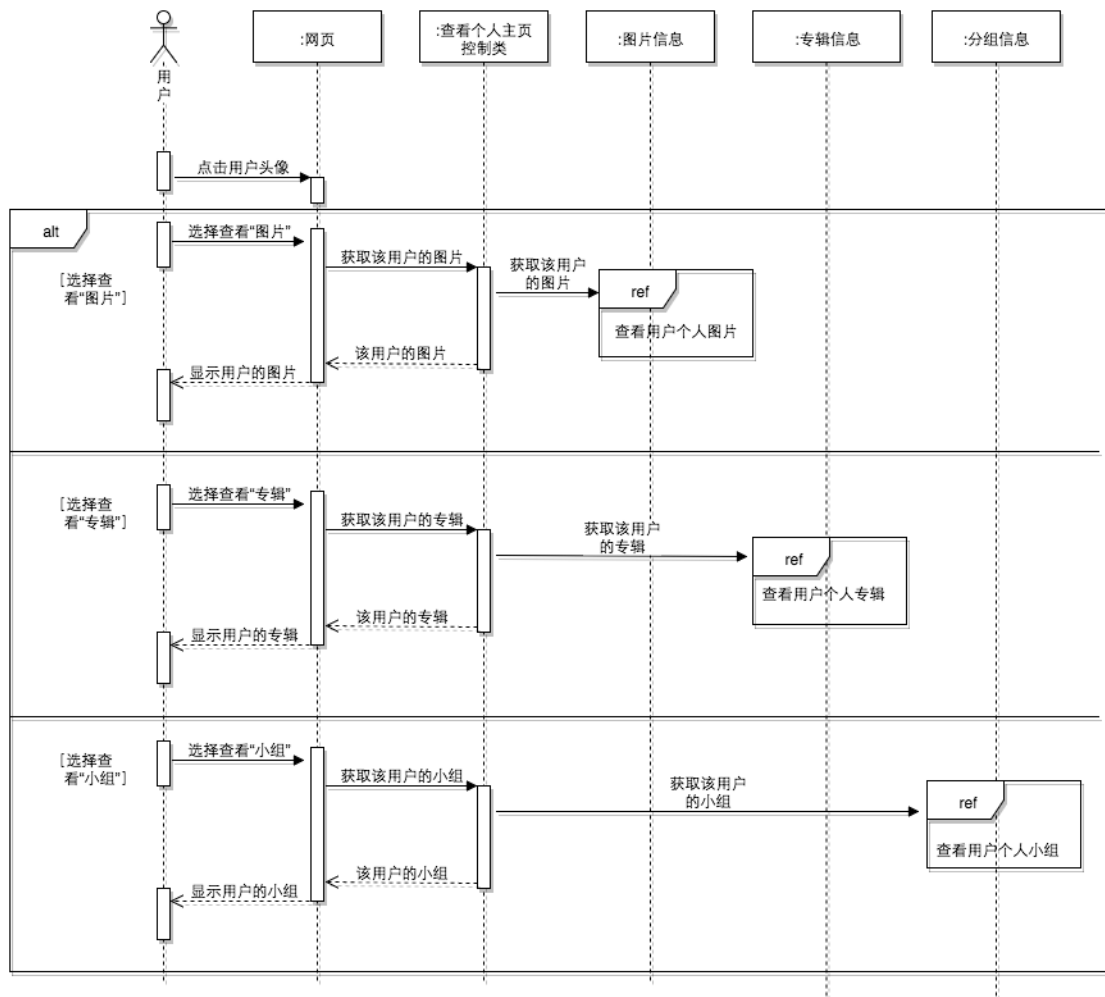
用户在个人主页上，点击“专辑”按钮，如果选择查看该用户“已上传的专辑”，系统将获取并展示该用户上传的专辑；如果选择查看该用户“收藏的专辑”，系统将获取并展示该用户收藏的专辑。

查看个人分组：



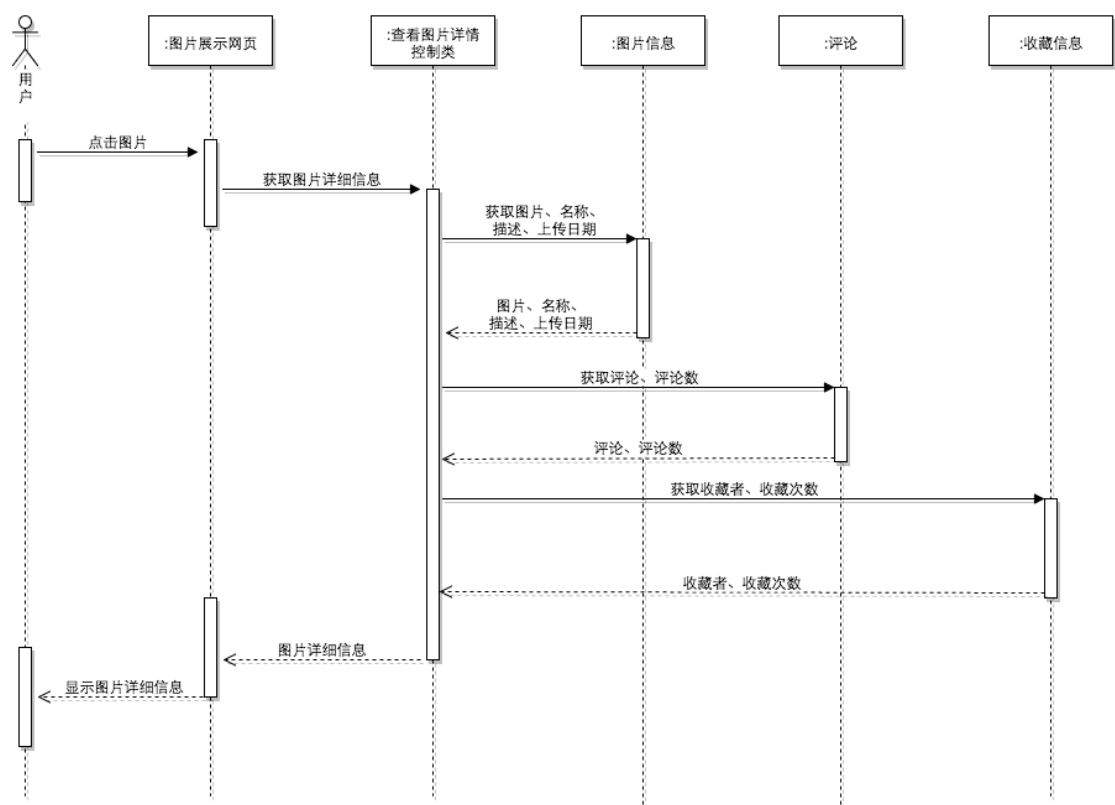
用户在个人主页上，点击“分组”按钮，如果选择查看该用户“发起的小组”，系统将获取并展示该用户发起的小组；如果选择查看该用户“加入的小组”，系统将获取并展示该用户加入的小组。

查看个人主页：



用户点击自己或他人的头像，跳转到该用户的个人主页上。如果用户选择查看该用户的“图片”，如“查看个人图片”的步骤；如果用户选择查看该用户的“专辑”，如“查看个人专辑”的步骤；如果用户选择查看该用户的“小组”，如“查看个人小组”的步骤；

查看图片详情：



用户点击图片，系统为用户分别获取该图片的高清版、名称、描述、上传日期、评论、评论数、收藏者、收藏次数等详细信息，并显示在网页上。

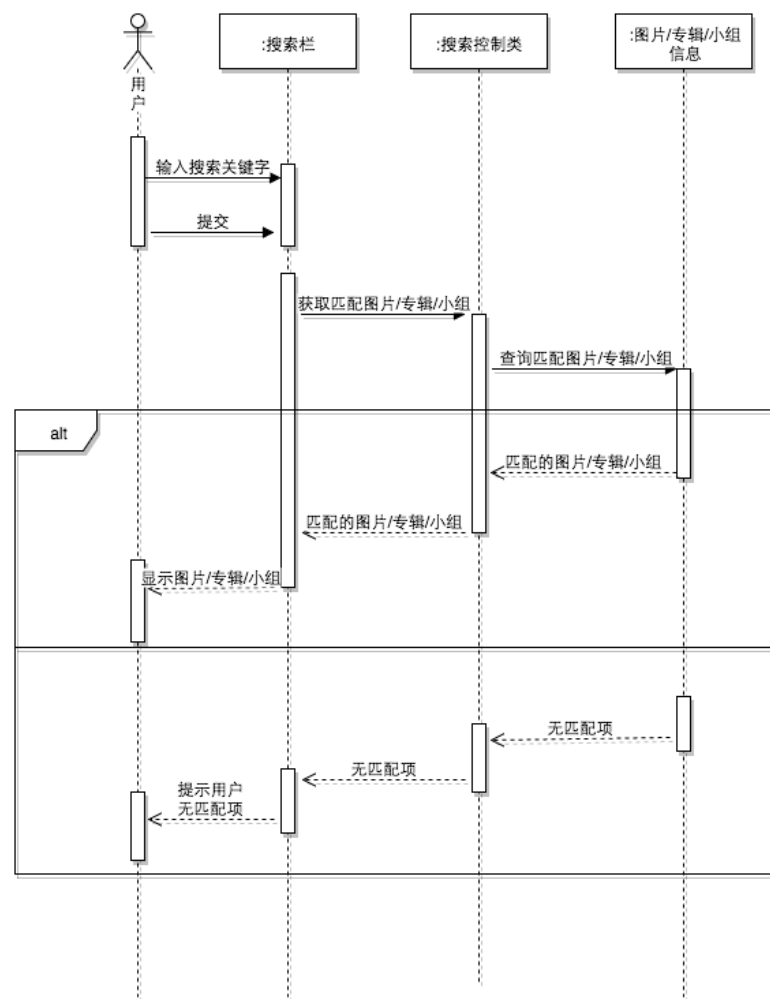
查看小组详情：

用户点击小组封面或名称，系统为用户分别获取该小组的名称、标签、发起人留言、上传日期、互动发言、互动次数、小组关注的专辑等详细信息，并显示在网页上。如果用户想查看小组中关注的专辑的详细信息，点击该专辑封面，如“查看专辑详情”的步骤。

查看专辑详情：

用户点击专辑封面或名称，系统为用户分别获取该专辑的名称、标签、标签、上传日期、专辑所属的图片、评论、评论次数、收藏者、收藏次数等详细信息，并显示在网页上。如果用户想查看专辑中的图片的详细信息，点击该图片，如“查看图片详情”的步骤。

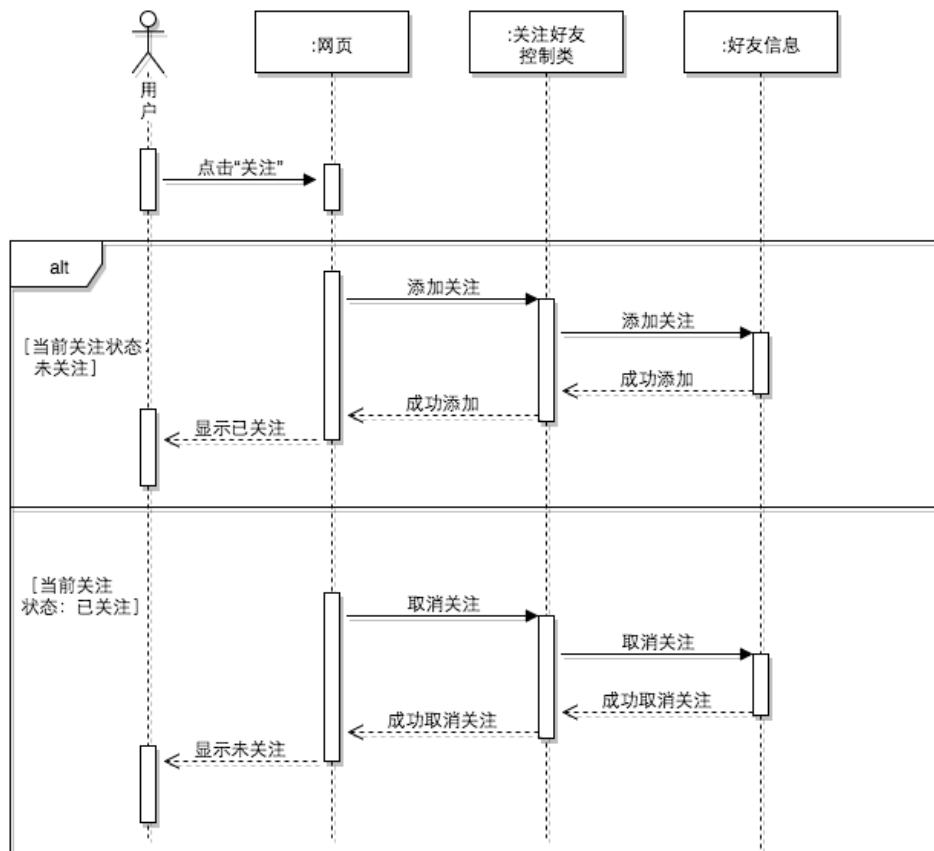
搜索：



用户输入搜索关键字并提交，系统通过搜索算法获取匹配的图片、专辑或小组，并显示给用户；若无匹配的搜索结果，显示无匹配项。

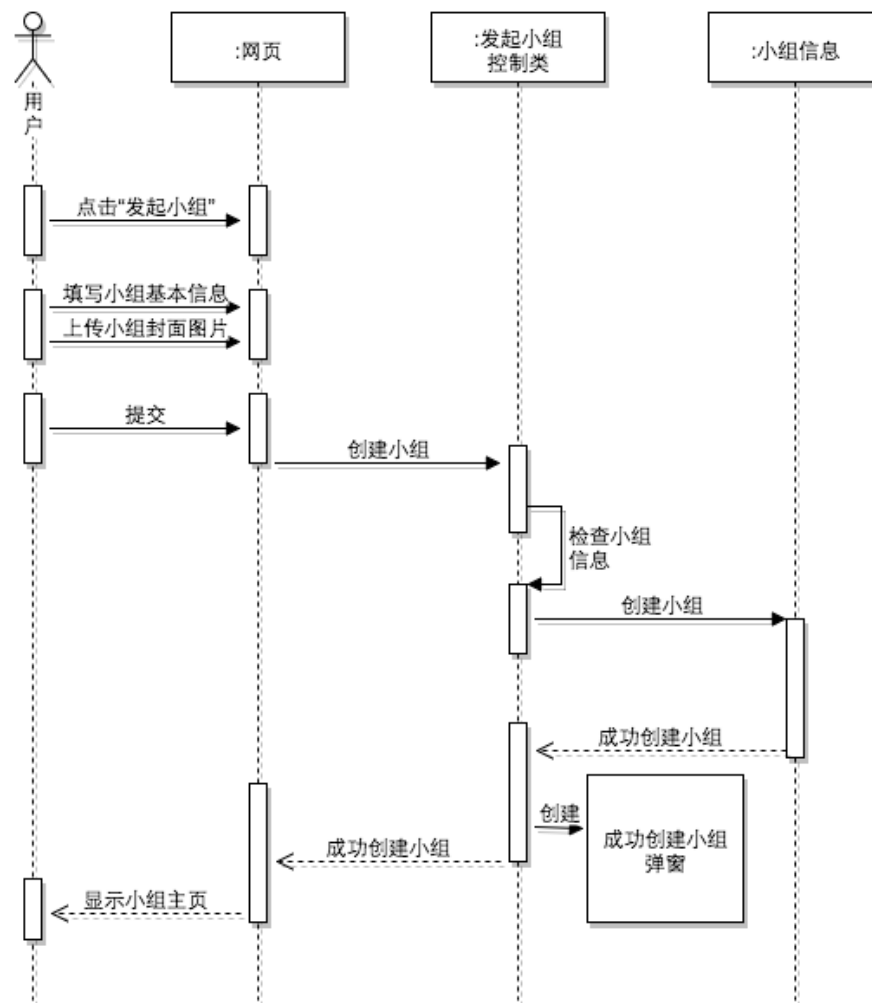
2.4.1.2 社交模块

关注好友：



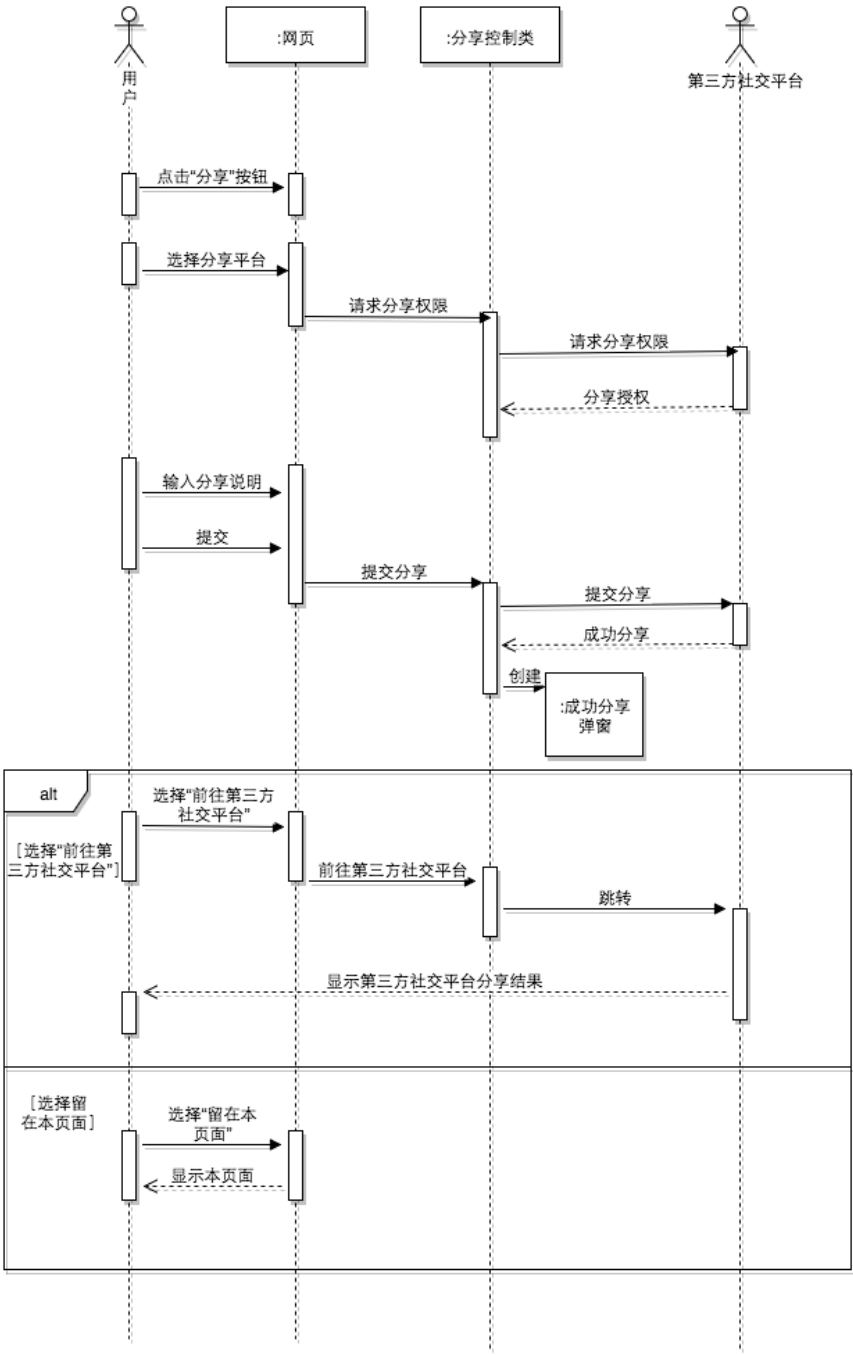
如果用户已关注过此用户，点击“关注”，取消关注；如果用户未关注过此用户，点击“关注”，添加关注。

发起小组：



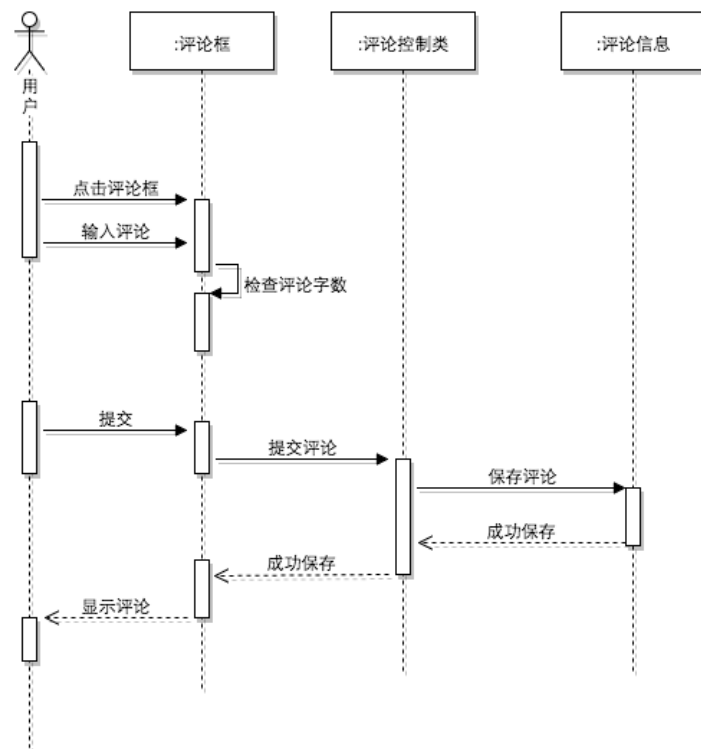
用户点击“发起小组”，并填写小组基本信息、上传封面图片，系统对小组信息进行检查，通过后弹出“成功创建小组”的弹窗，并进入新建小组的主页。

分享:



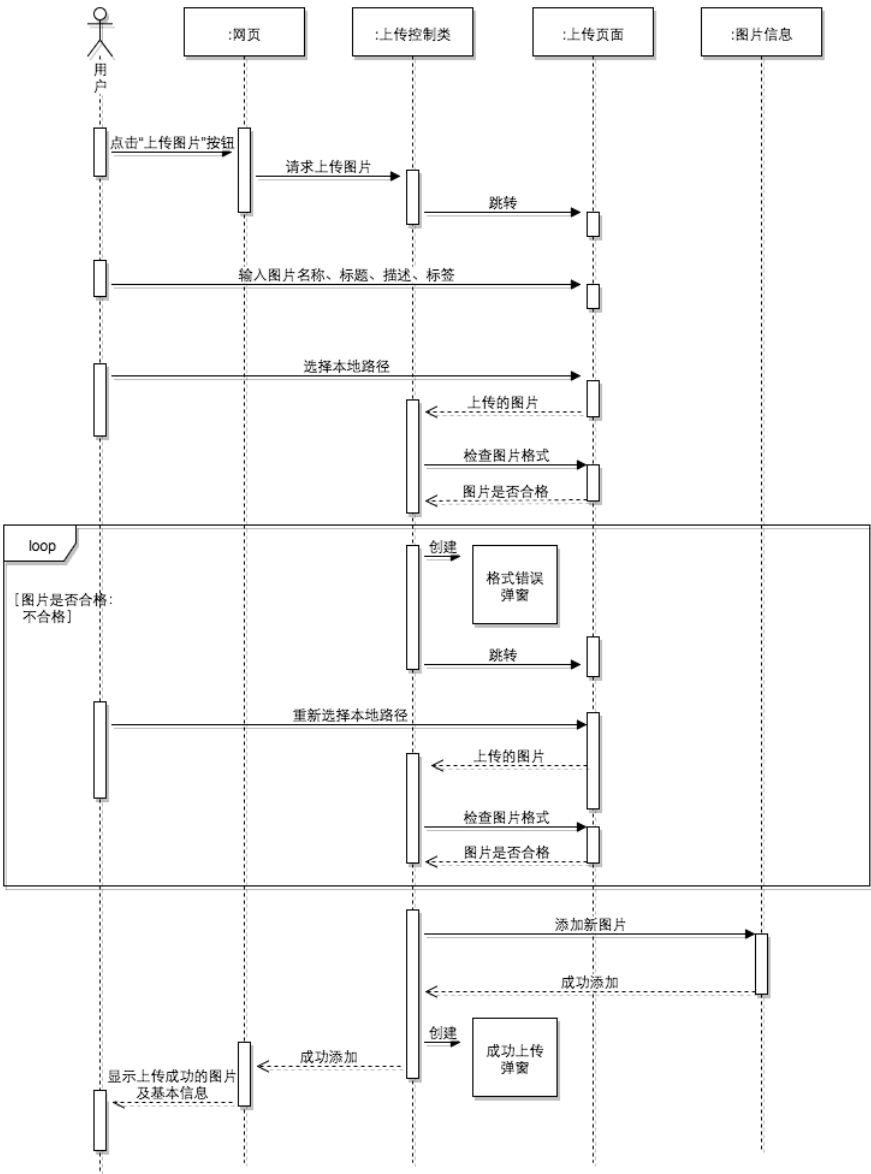
用户点击“分享”并选择分享平台，系统获取分享权限。用户输入分享说明并提交，分享成功后弹出“成功分享”弹窗。之后，用户可以选择前往第三方平台查看分享结果，或者留在本页面继续浏览。

评论：



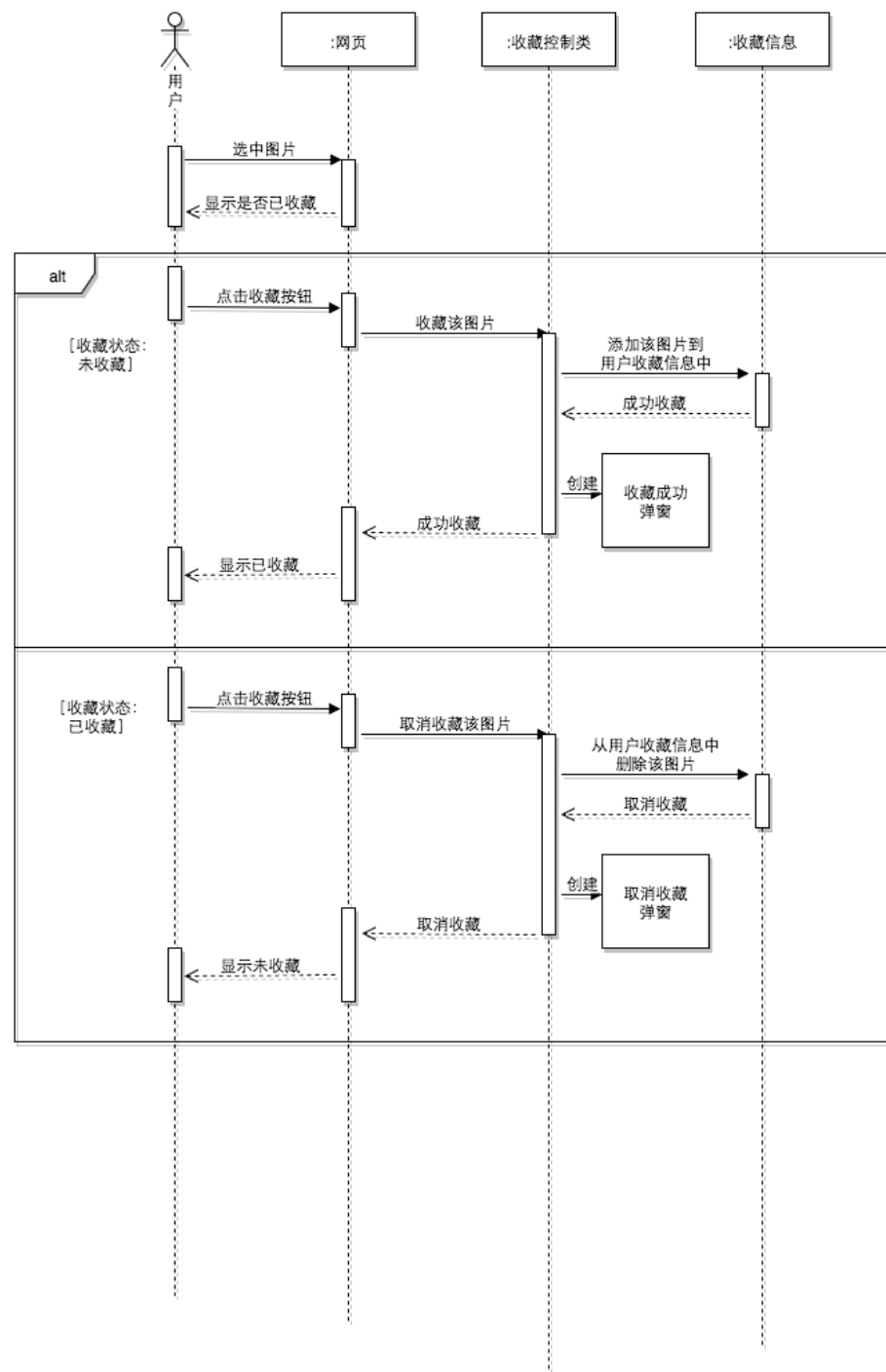
用户点击评论框，并输入评论，如果字数在限定范围内，则提交并保存评论，网页上将显示用户的评论。

上传图片：



用户选择“上传图片”，输入图片名称、标题、描述、标签，选择图片的本地路径，如果图片格式不符合要求，则重新选择上传路径；否则上传成功。

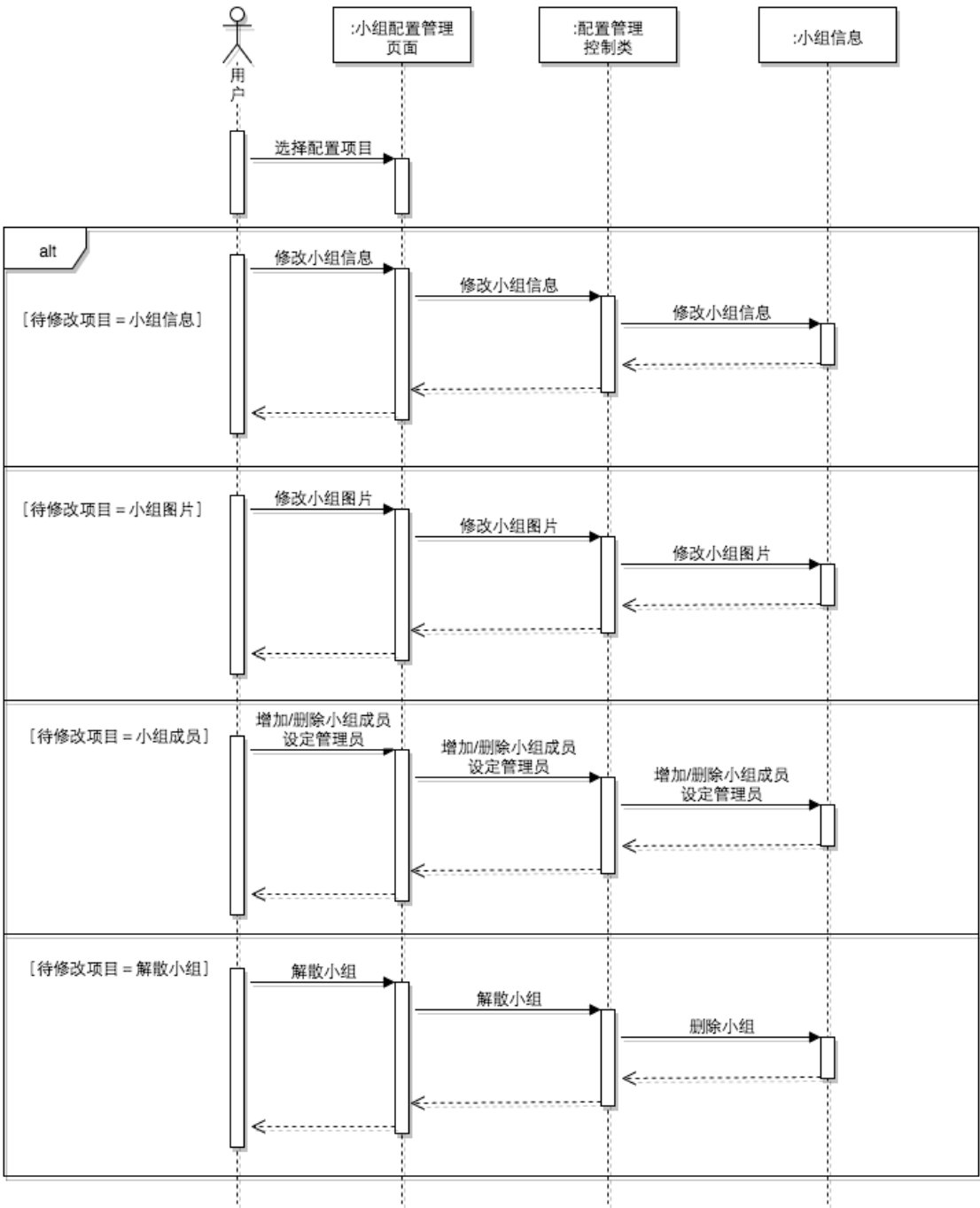
收藏：



用户选中某张图片，系统将显示该图片是否被收藏。若未收藏，点击“收藏”按钮，成功收藏；若一收藏，点击“收藏”按钮，取消收藏。

2.4.1.3 个人主页管理模块

管理小组：

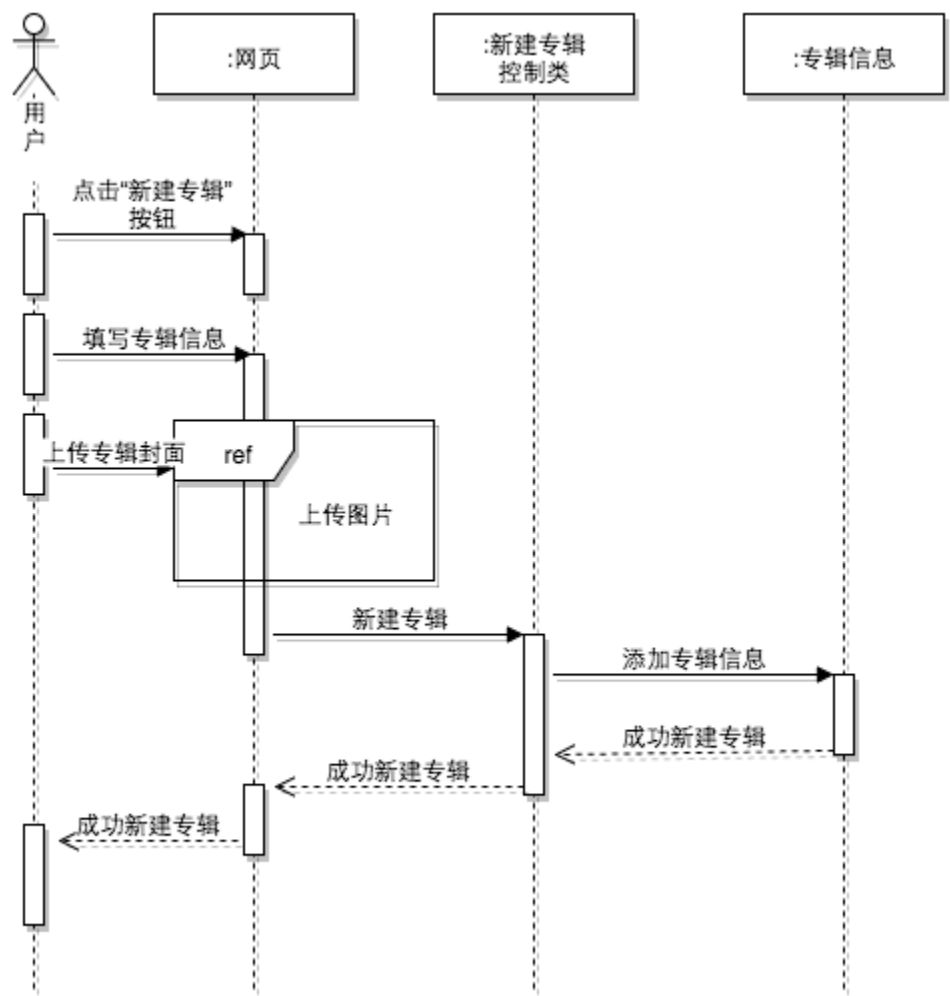


用户在小组配置管理页面上选择配置的项目：如果用户选择配置“小组信息”，用户修改当前小组信息并提交；如果用户选择配置“小组专辑”，选择某张专辑，如“管理专辑”的步骤；如果用户选择配置“小组成员”，用户可以增加、删除小组成员或设定管理员；如果用户选择“解散小组”，确认后将从小组信息中删除该小组。

管理专辑：

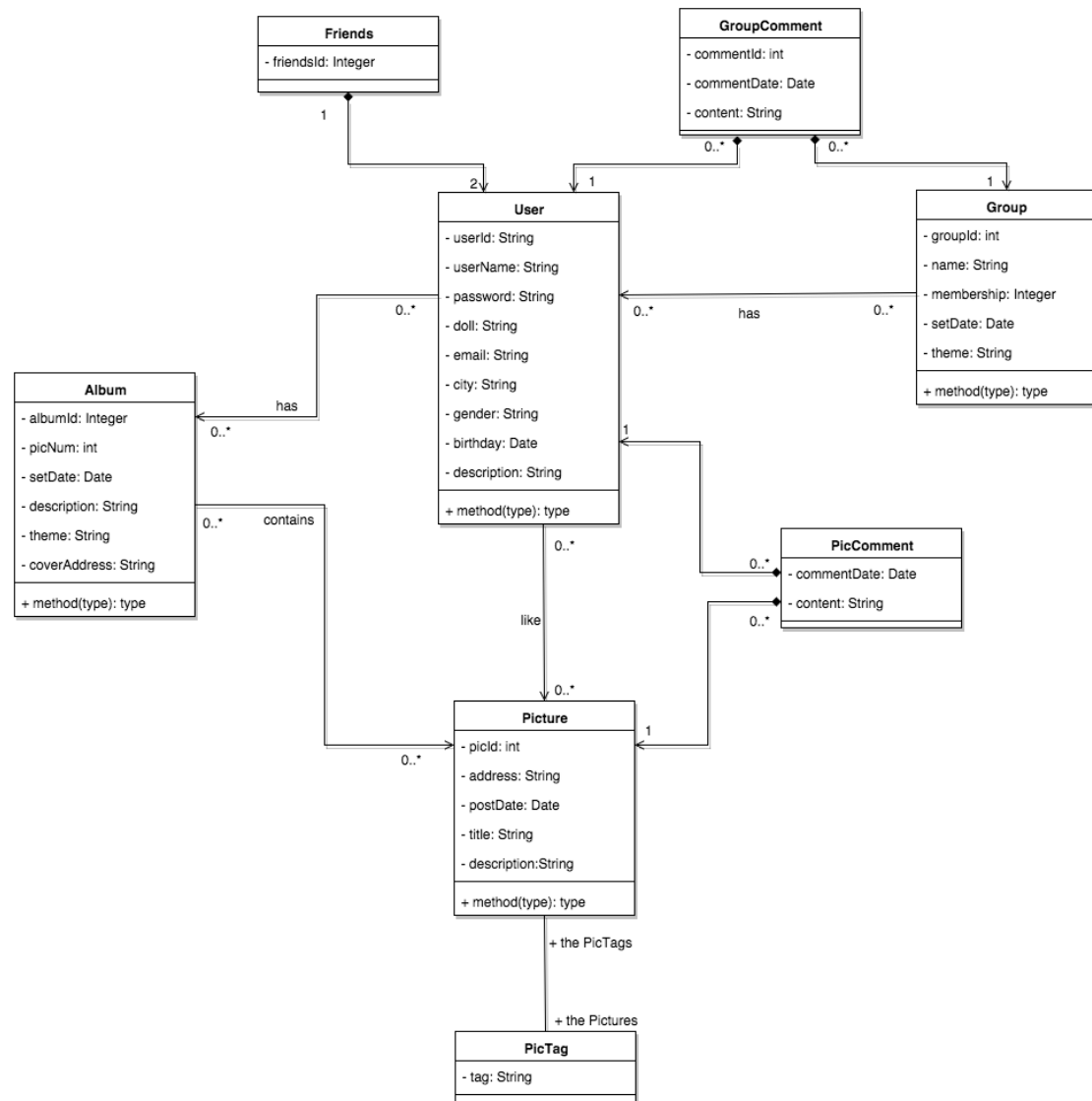
用户在专辑配置管理页面上选择配置的项目：如果用户选择管理专辑列表，则用户可以选择并且比批量删除小组中的专辑；如果用户选择管理选中的专辑，选择某张专辑，修改专辑的基本信息；如果用户选择为专辑添加图片，用户选择某张专辑，如“上传图片”中的步骤；如果用户选择管理专辑中的某张图片，如“管理图片”的步骤；如果用户选择管理专辑中的图片列表，用户可以选择并批量删除选中的图片。

新建专辑：



用户点击“新建专辑”，填写专辑信息，上传专辑封面图片并提交，页面将显示成功新建专辑的提示。

2.4.2 数据建模



2.4.2.1 用户类（User）

编号：CD1

职责：存储和管理用户基本信息。

属性：用户 ID，昵称，密码，头像，邮箱，所在城市，性别，生日，个人描述。

说明：用户类和图片类单向关联，一个用户可以收藏多张图片。

用户类的所有属性都是持久的，它存储了注册用户的基本信息。

2.4.2.2 小组类

编号：CD2

职责：存储和管理小组基本信息、小组成员。

属性：小组 ID，名称，创建日期，主题。

说明：小组类和用户类存在聚合关系，一个小组由多个用户组成。

小组类的所有属性都是持久的，它存储了小组及成员的信息。

2.4.2.3 小组评论类

编号：CD3

职责：存储用户在小组中的评论发言。

属性：评论 ID，评论日期，评论内容。

说明：小组评论类与用户存在关联关系，一个用户可以发表多条小组评论；小组评论类与小组存在关联关系，一个小组可以被多次评论。

2.4.2.4 好友类

编号：CD4

职责：存储用户之间的好友关系。

属性：好友 ID，用户 1ID，用户 2ID。

说明：好友类与用户类存在聚合关系，一个好友由两个用户组成，在类的实例化过程中，两个类所扮演的角色不同。

2.4.2.5 专辑类

编号：CD5

职责：存储专辑的基本信息。

属性：专辑 ID，图片数量，创建日期，描述，主题。

说明：用户类和专辑类存在关联关系，一个用户可以收藏或创建多个专辑。

2.4.2.6 图片类

编号：CD6

职责：存储图片基本信息

属性：图片 ID，地址，上传日期，标题，描述。

说明：图片类中的所有属性都是持久化的，它负责存储图片的基本信息。

2.4.2.7 图片评论类

编号：CD7

职责：存储用户对于某张图片的评论。

属性：评论日期，内容

说明：图片评论类和用户存在关联关系，一个用户可以发起多条评论。图片评论类和图片之间存在关联关系，一个图片可以被多次评论。

2.4.2.8 图片标签类

编号：CD8

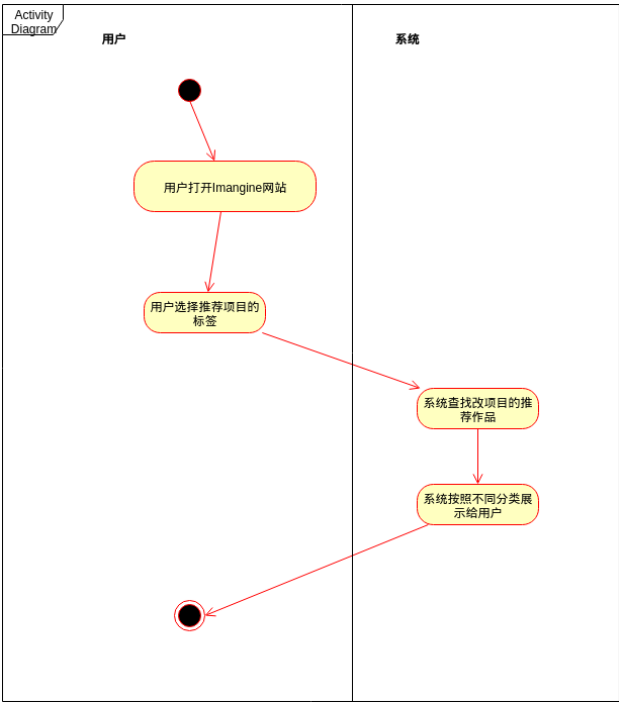
职责：存储图片的标签。

属性：标签内容

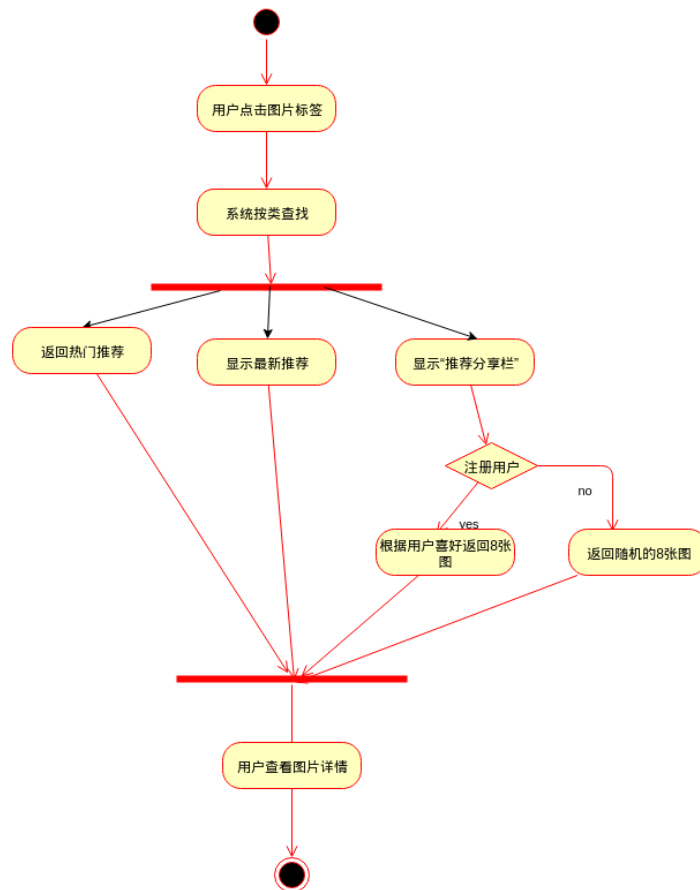
说明：标签类和图片之间存在关联关系，一个图片有多个标签，一个标签标记多个图片。

2.4.3 行为建模

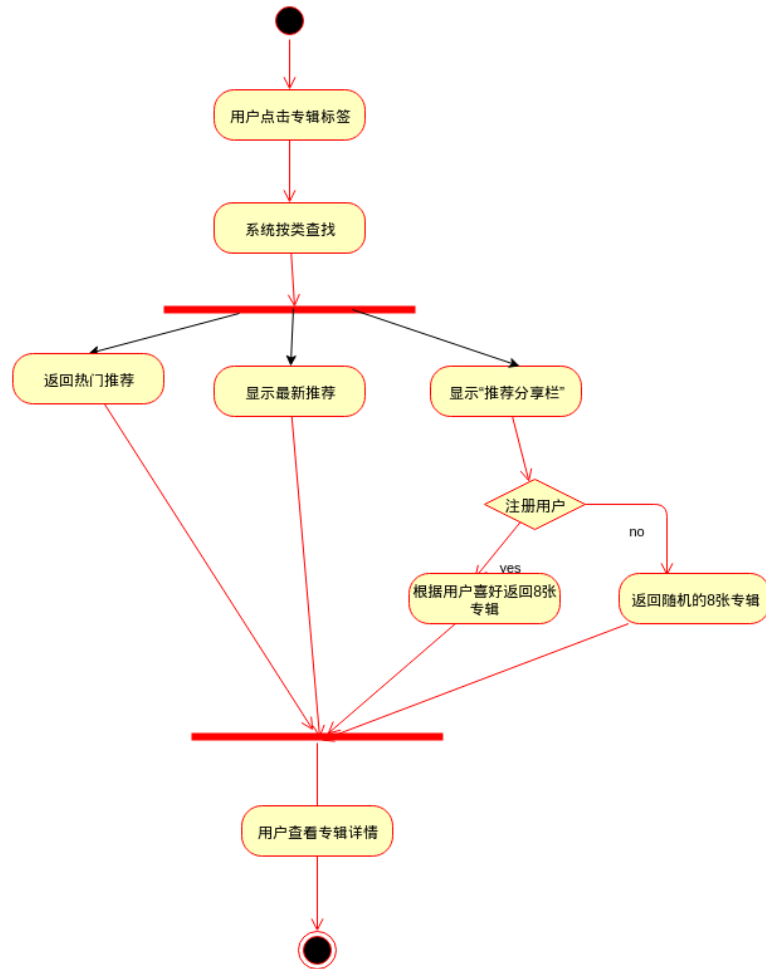
2.4.3.1 浏览模块



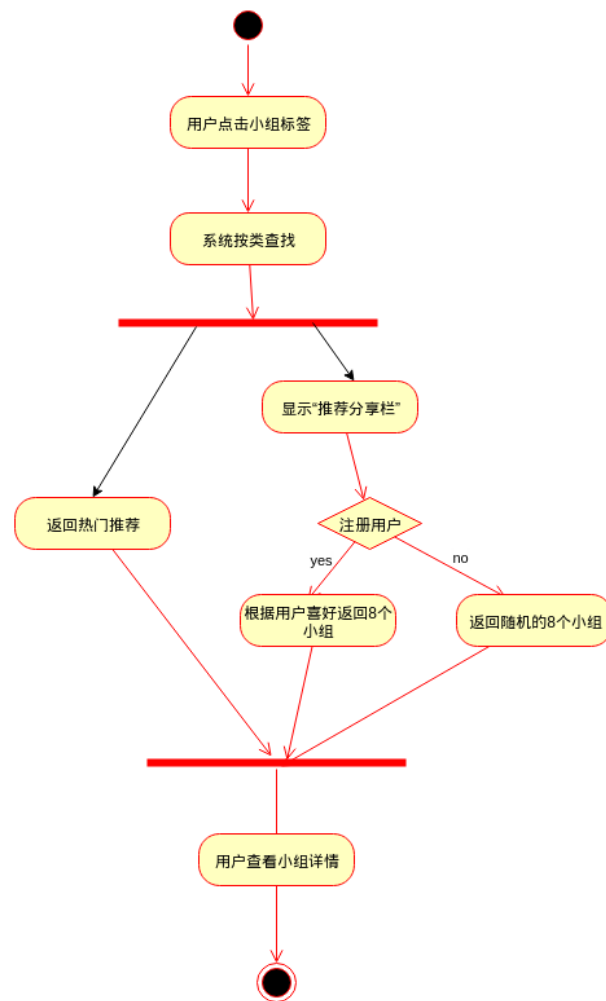
推荐：系统按照不同标准的分类(如最新、最热)为用户推荐艺术作品。



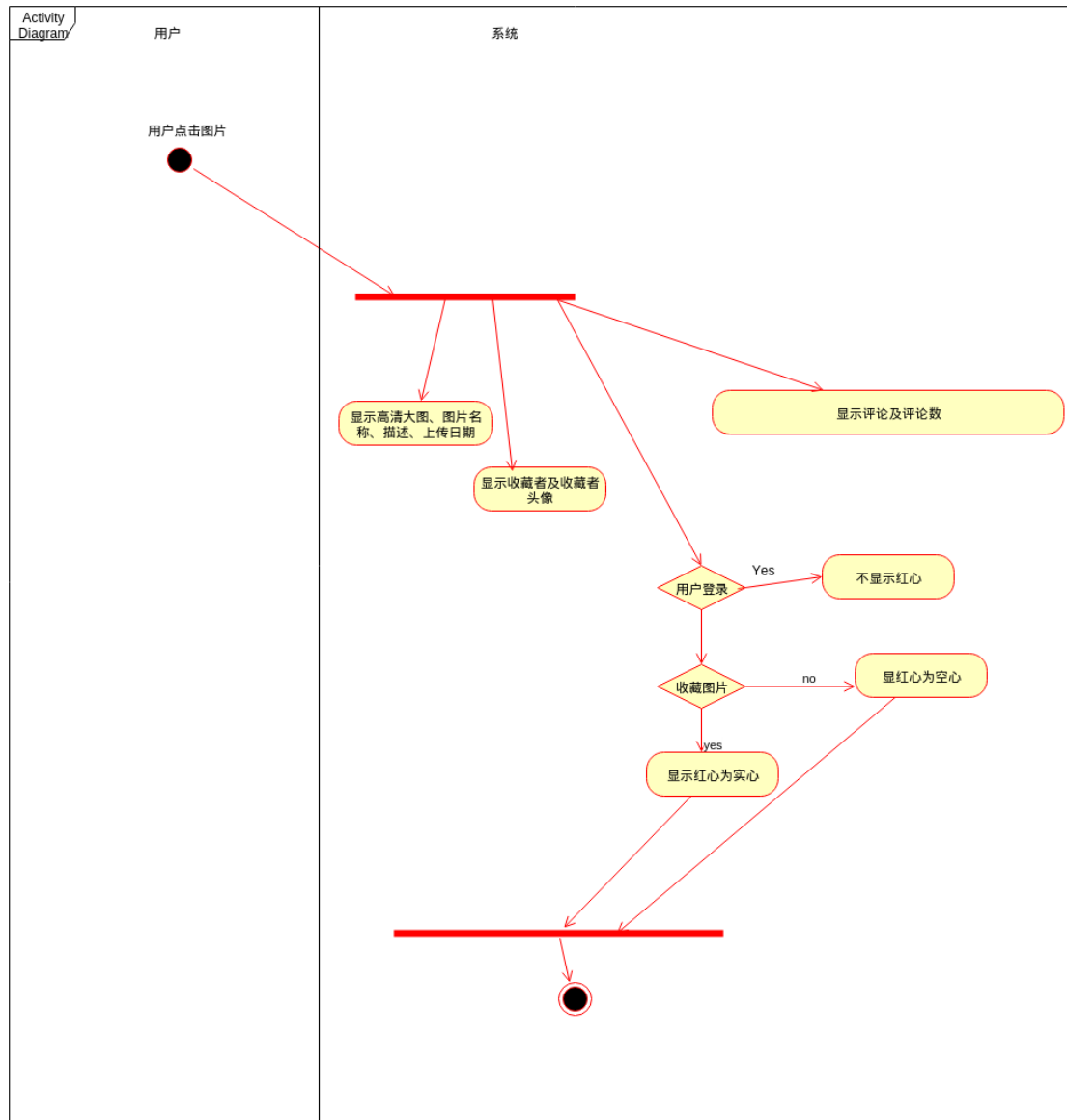
推荐图片：系统按照最热门、最新、用户个人偏好为用户推荐图片。



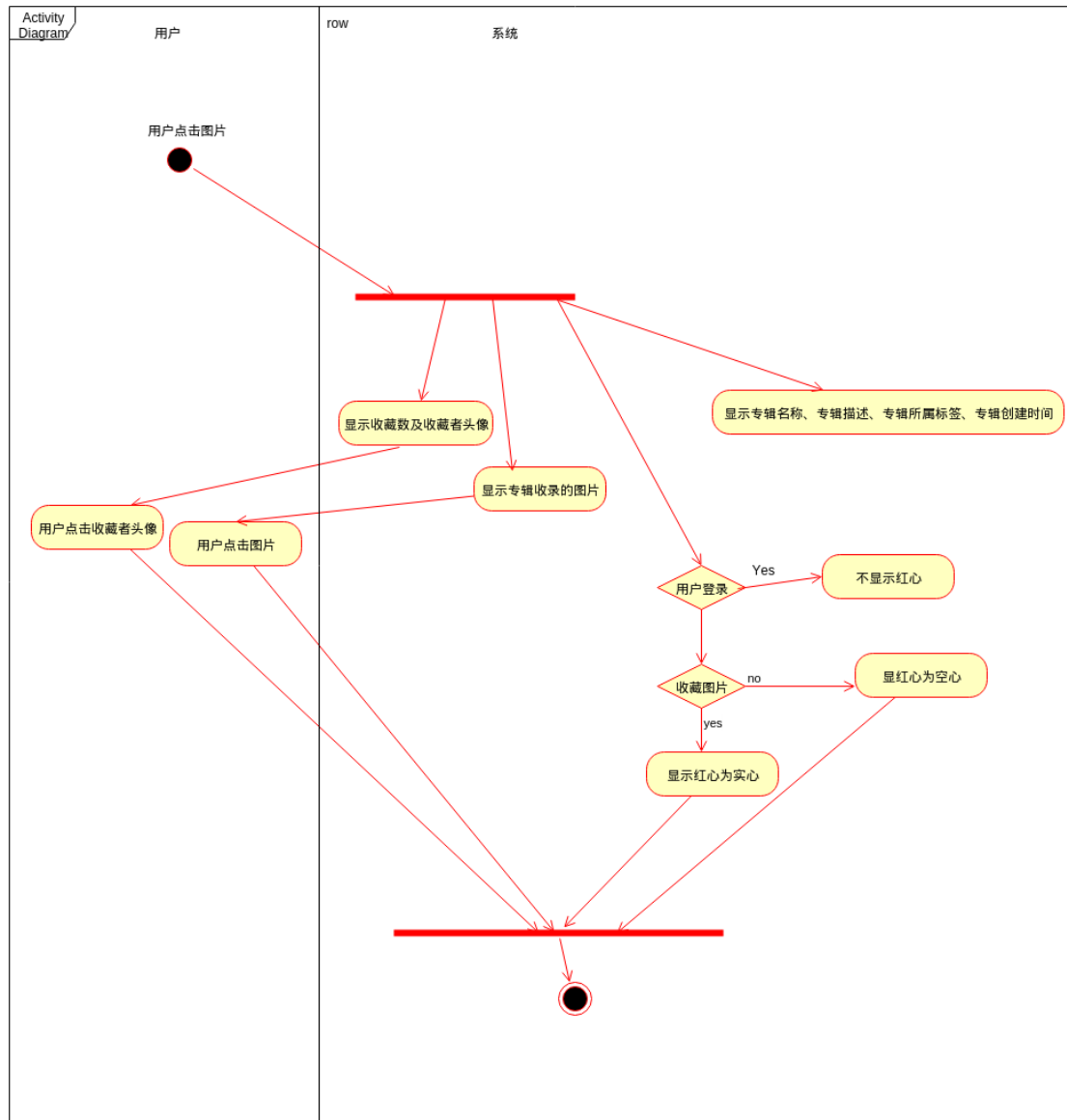
推荐专辑：系统按照最热门、最新、用户个人偏好为用户推荐专辑。



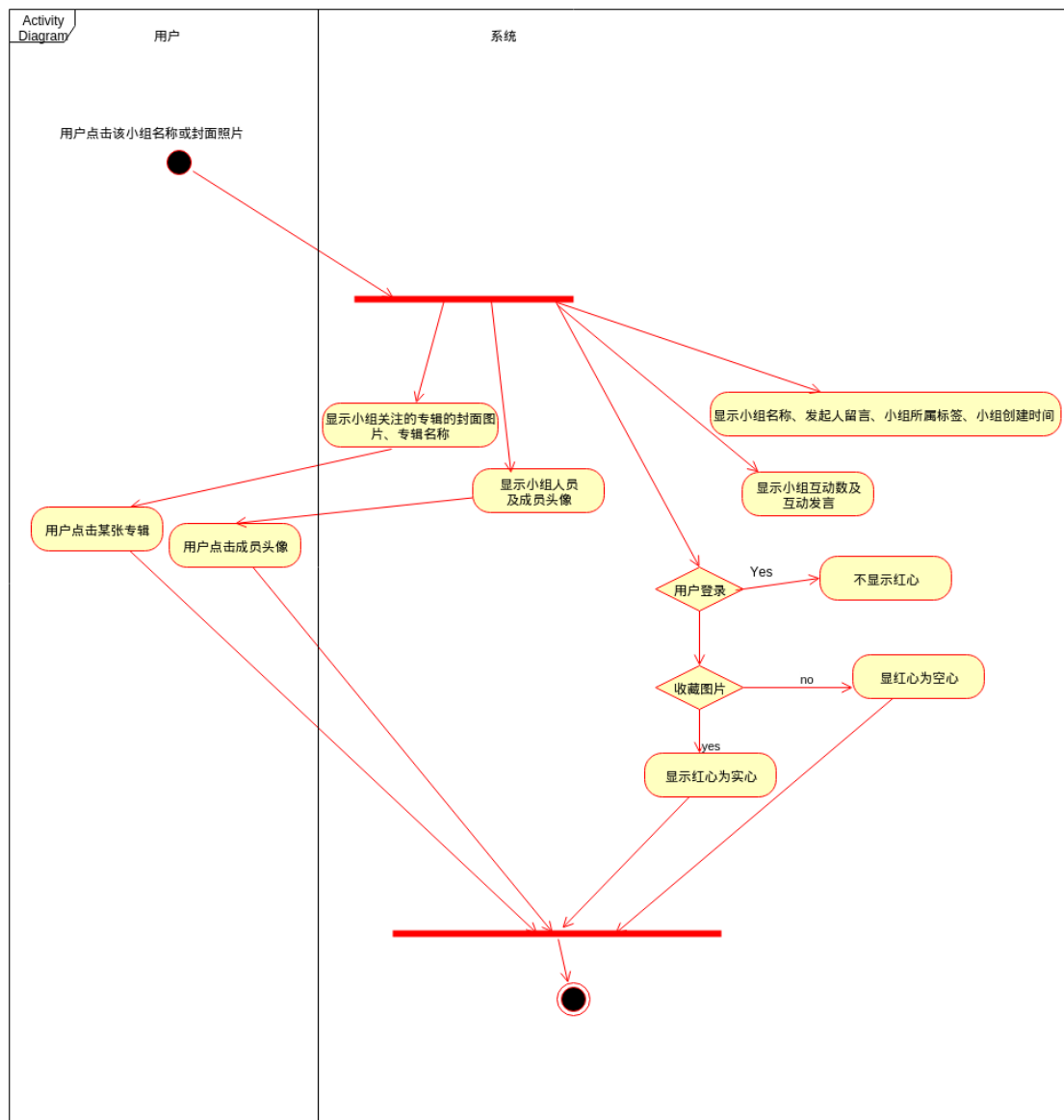
推荐小组：系统按照最热门(小组成员人数最多)、用户个人偏好为用户推荐小组。



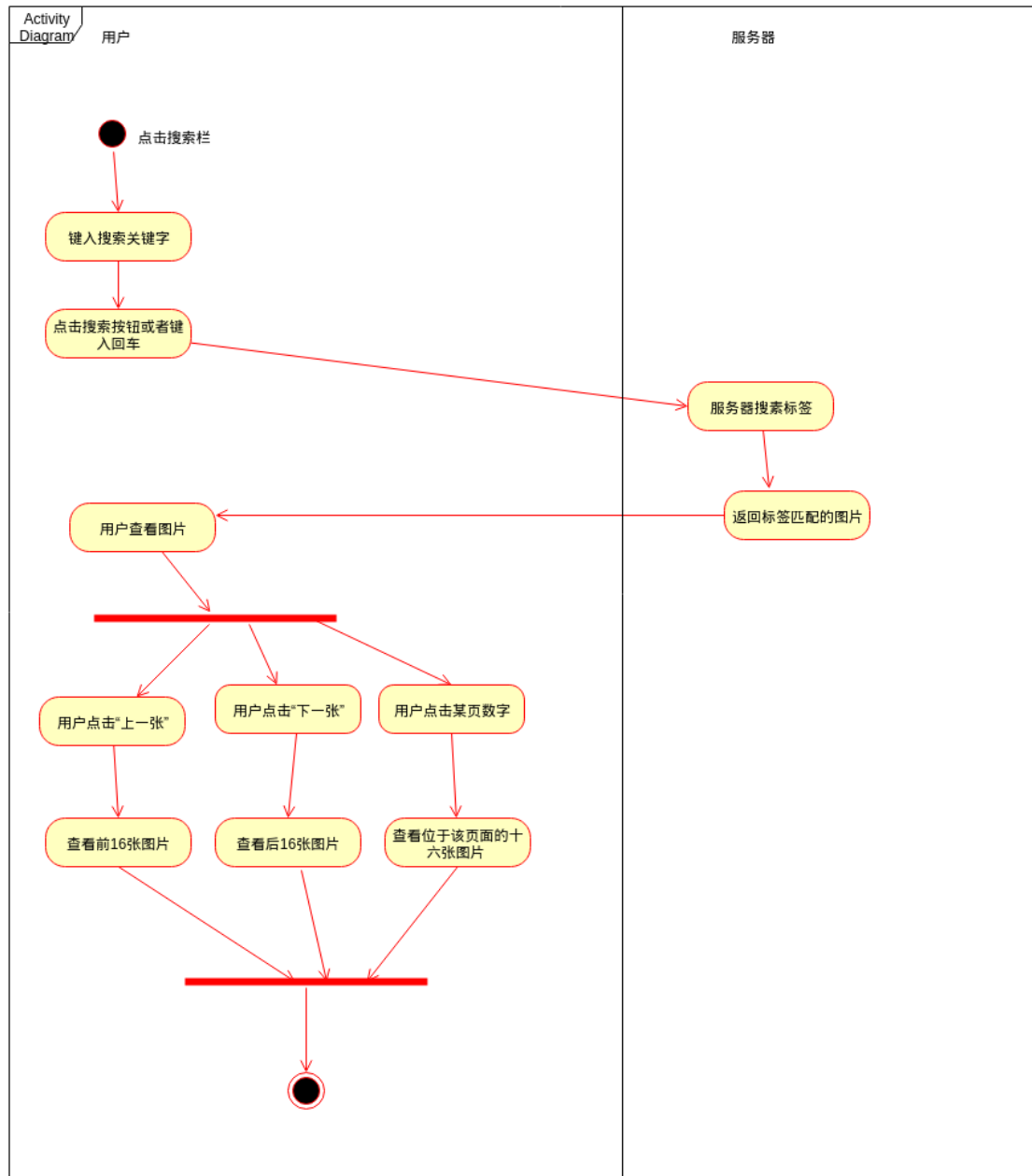
查看图片详情：用户查看大图, 以及图片详细信息。



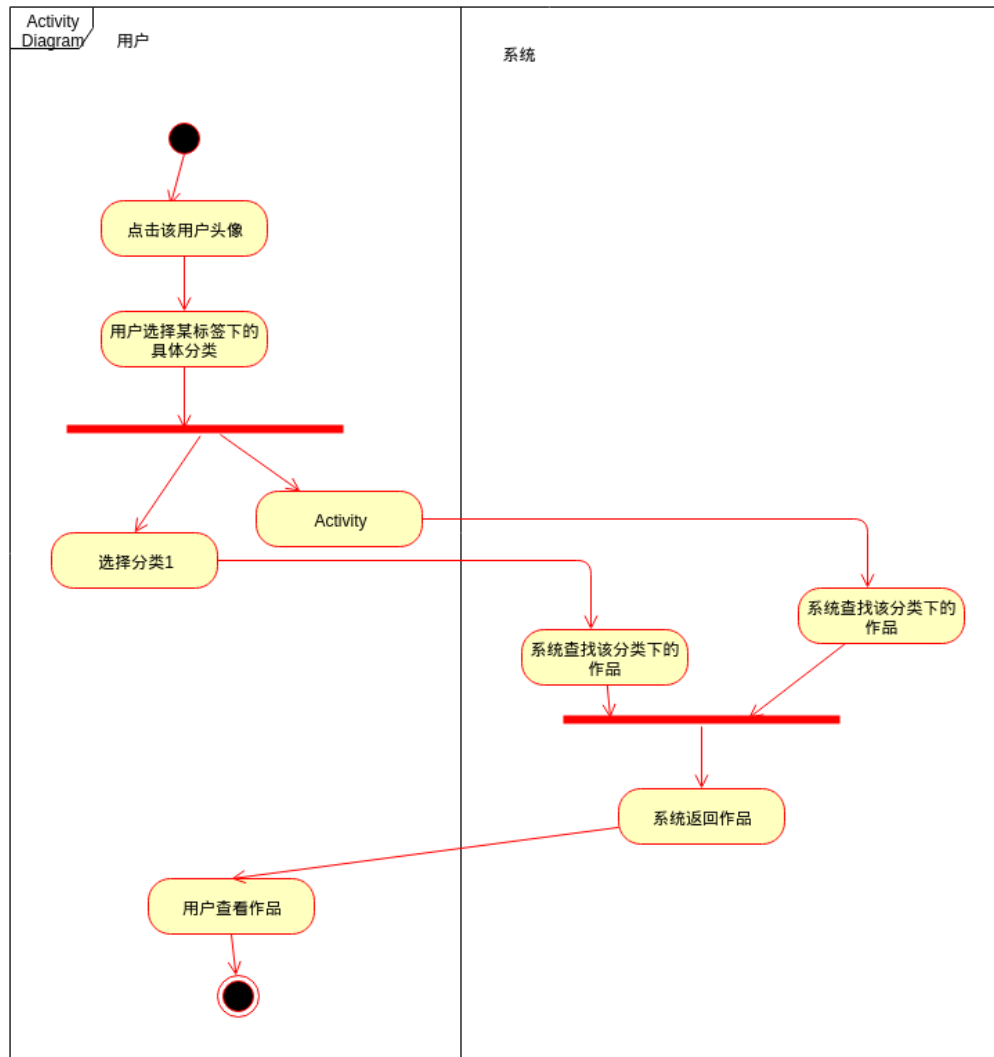
查看专辑详情：用户查看专辑有关的详细信息。



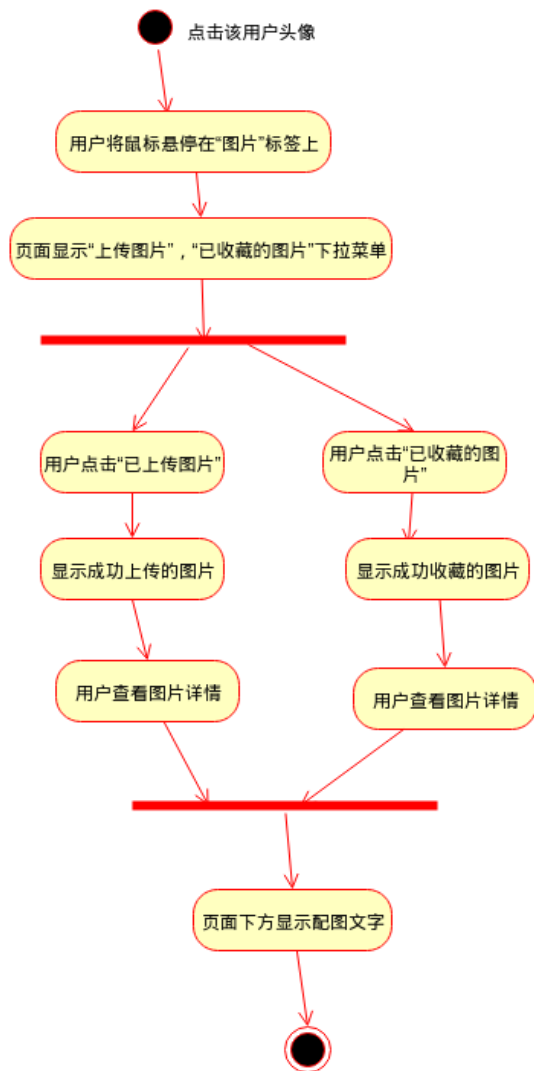
查看小组详情：用户查看小组有关的详细信息。



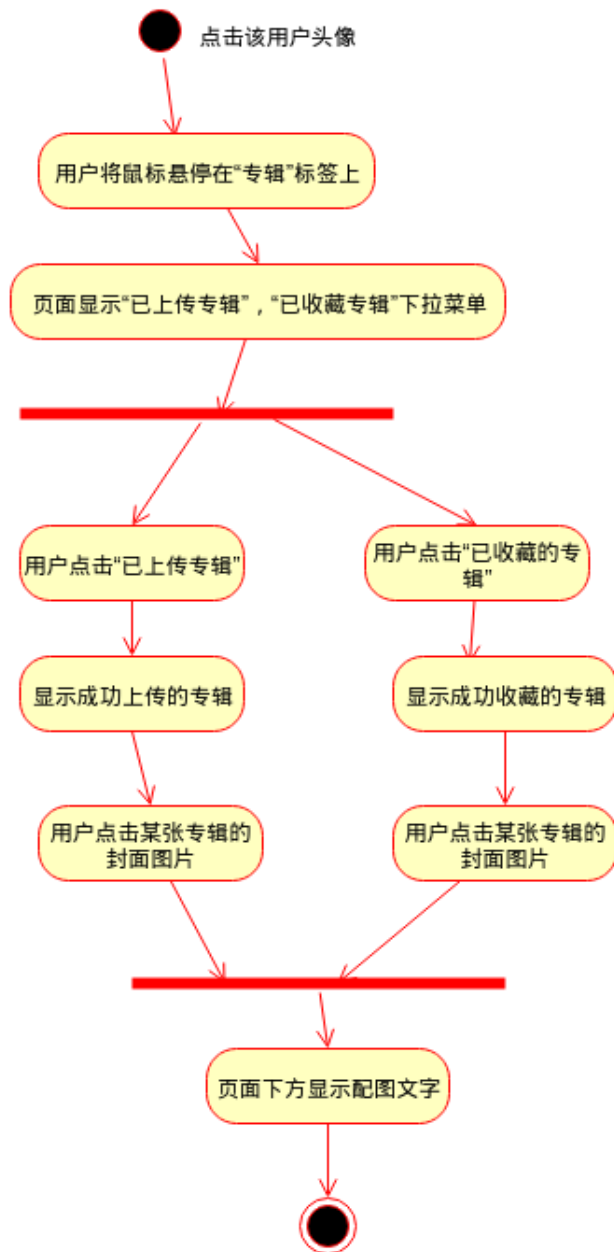
搜索：用户在搜索栏键入关键词，系统根据图片标签返回搜索结果。



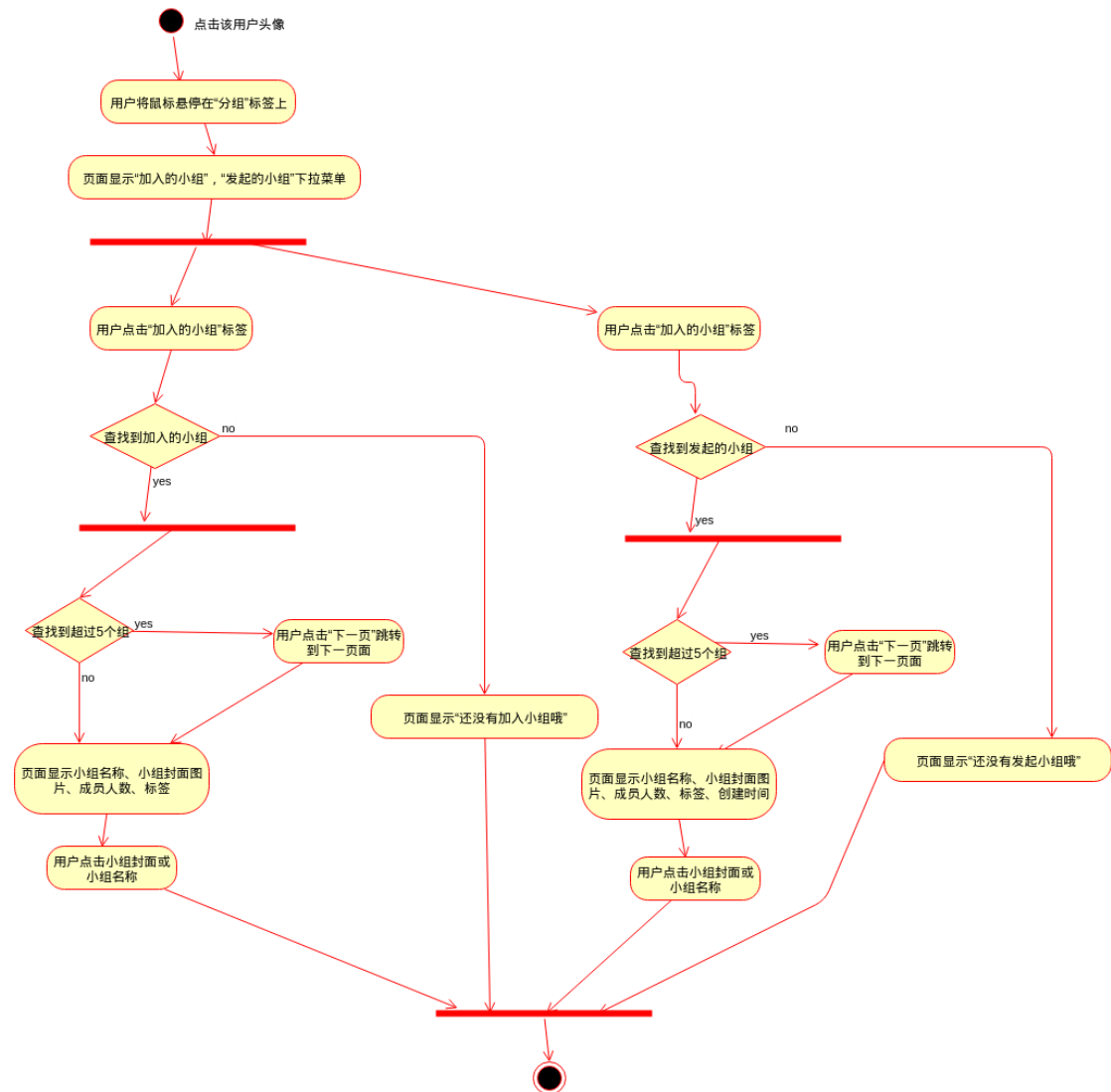
查看个人主页：用户(个人主页用户本人/其他用户)可以查看该用户主页内容。



查看个人图片：用户(个人主页用户本人/其他用户)可以查看该用户建立的专辑和被该用户收藏的 专辑。

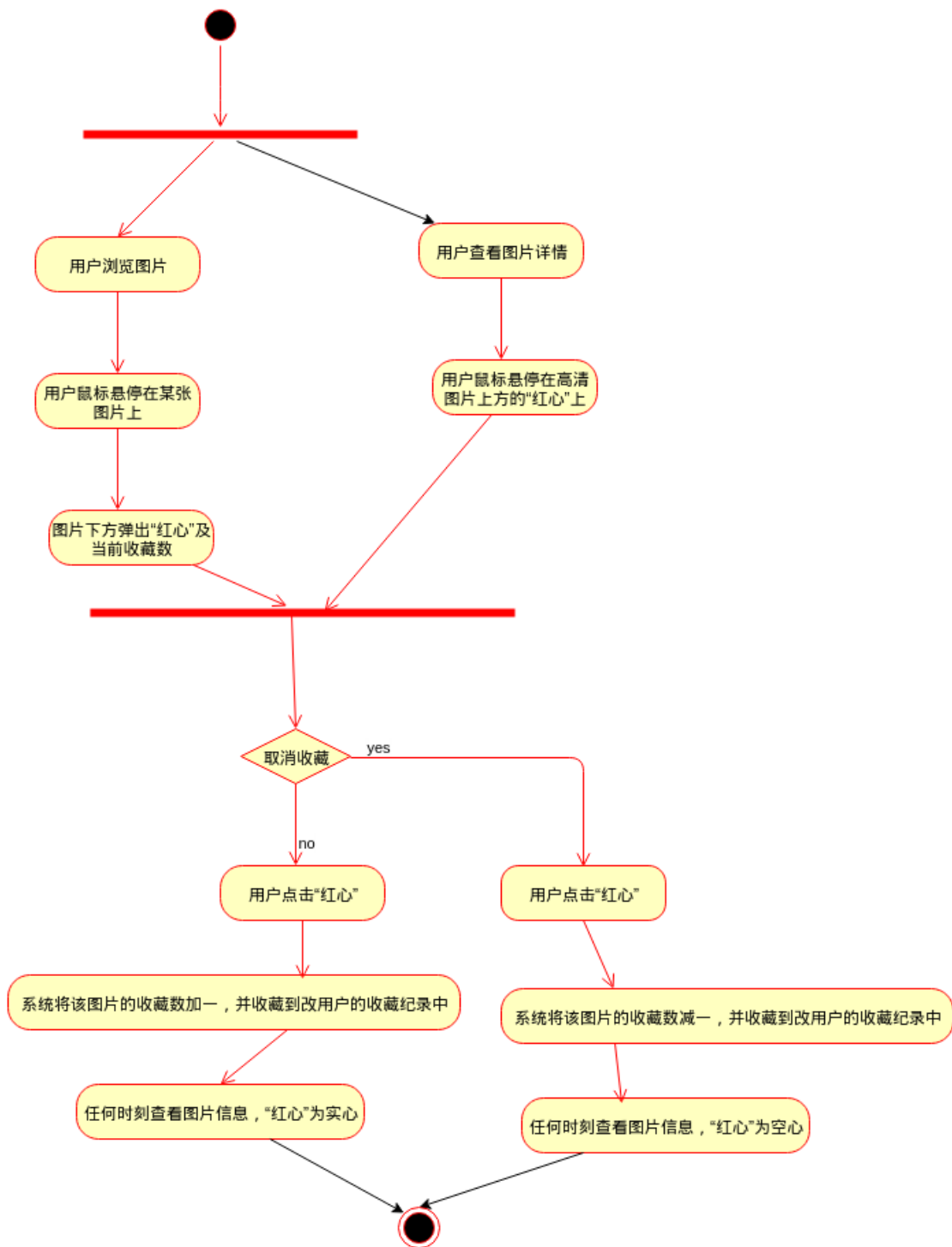


查看个人专辑：用户（个人主页用户本人/其他用户）可以查看该用户建立的专辑和被该用户收藏的 专辑。

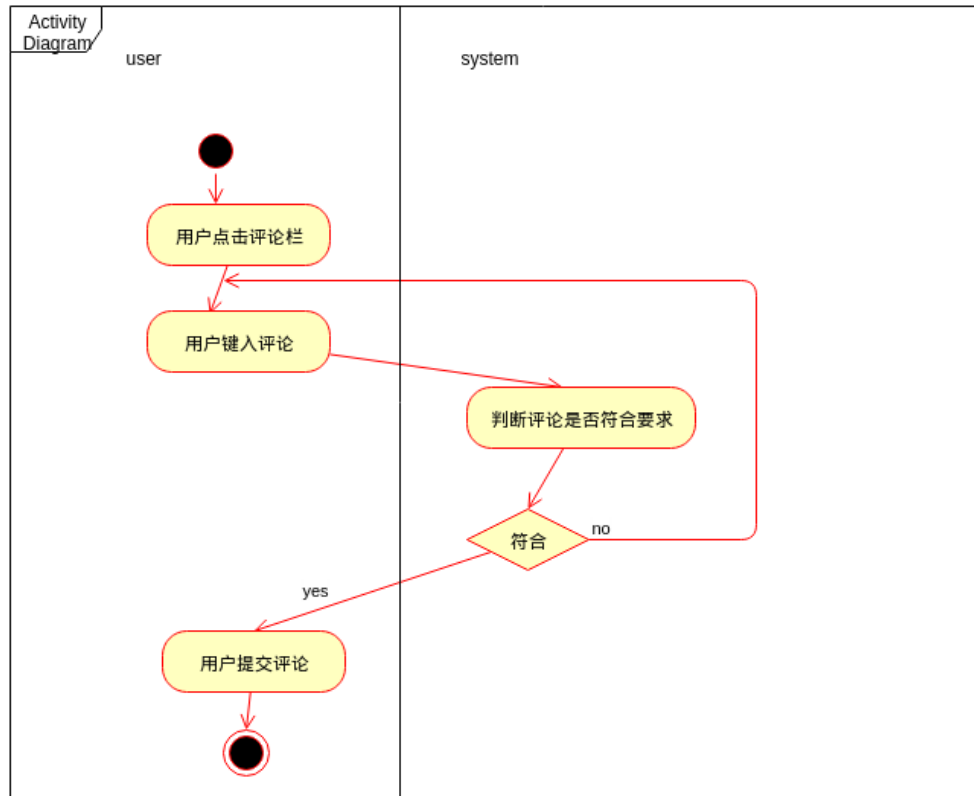


查看个人分组：用户(个人主页用户本人/其他用户)可以查看该用户加入和发起的小组

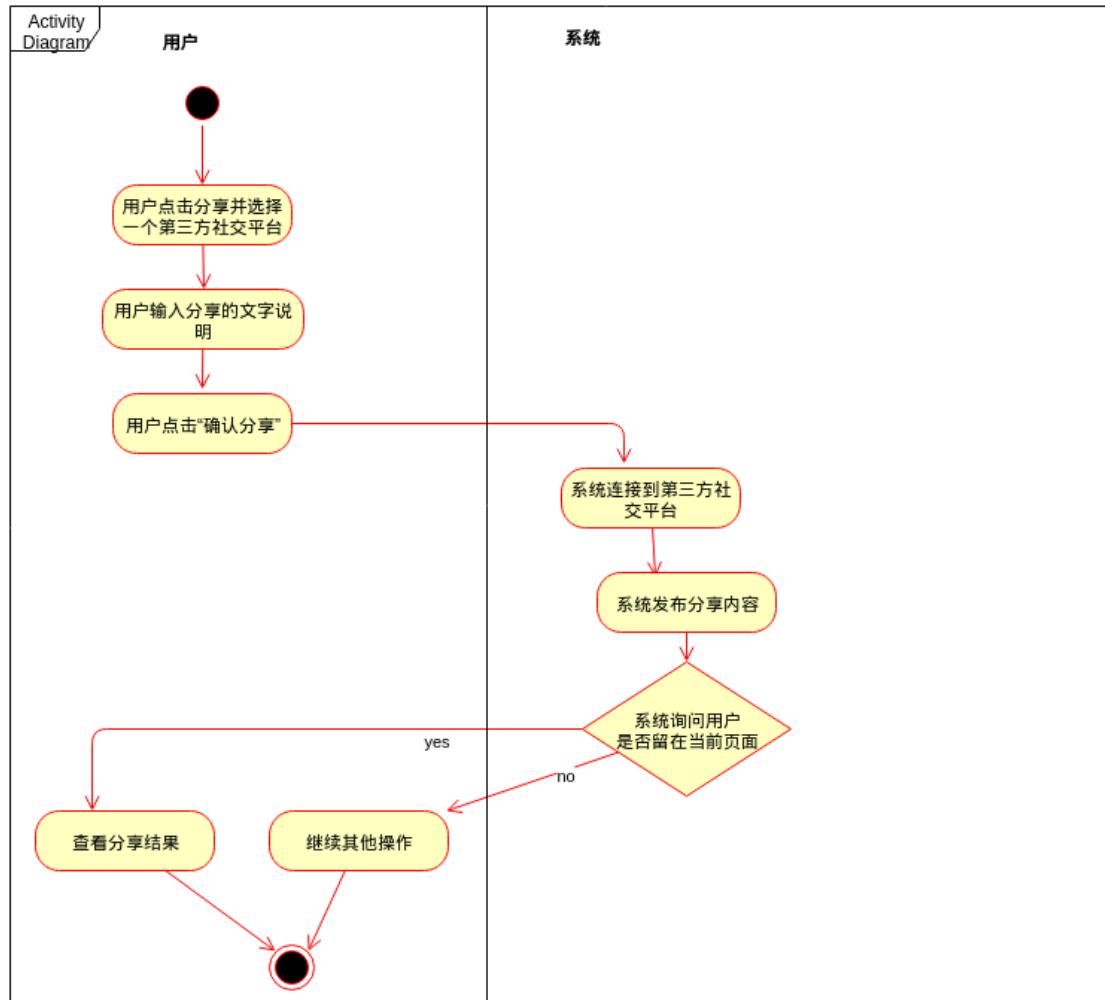
2.4.3.2 社交模块



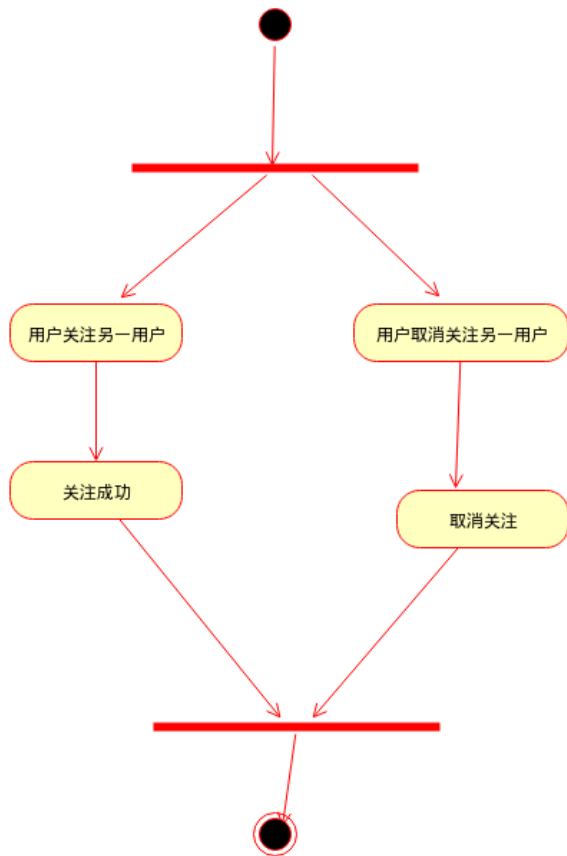
收藏： 用户点击“红心”，将该图片收藏。



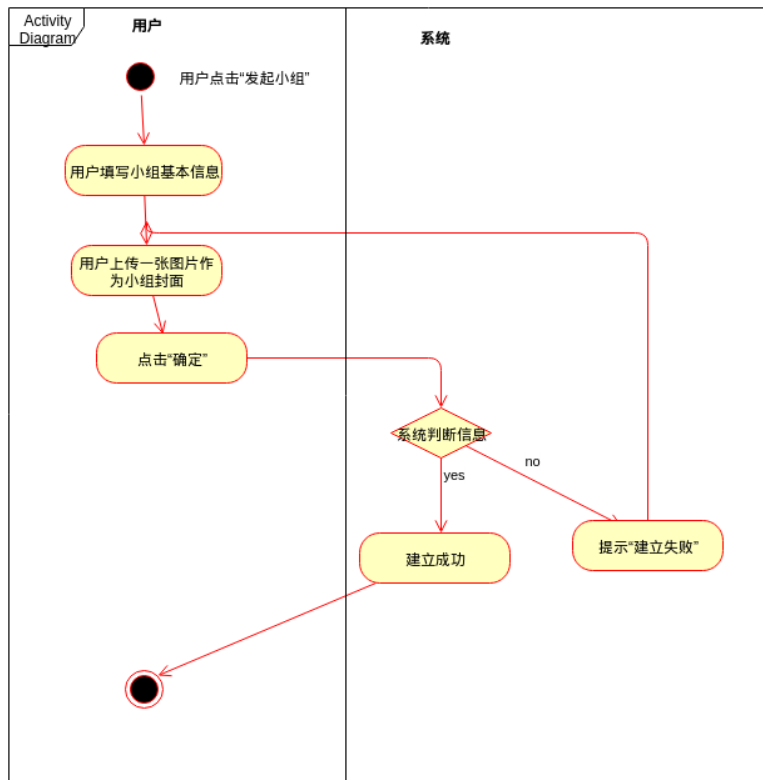
评论：用户在某图片详细信息页面下方的评论栏中,输入评论并提交。



分享：用户点击“分享”按钮并附上说明文字, 分享到第三方社交平台。

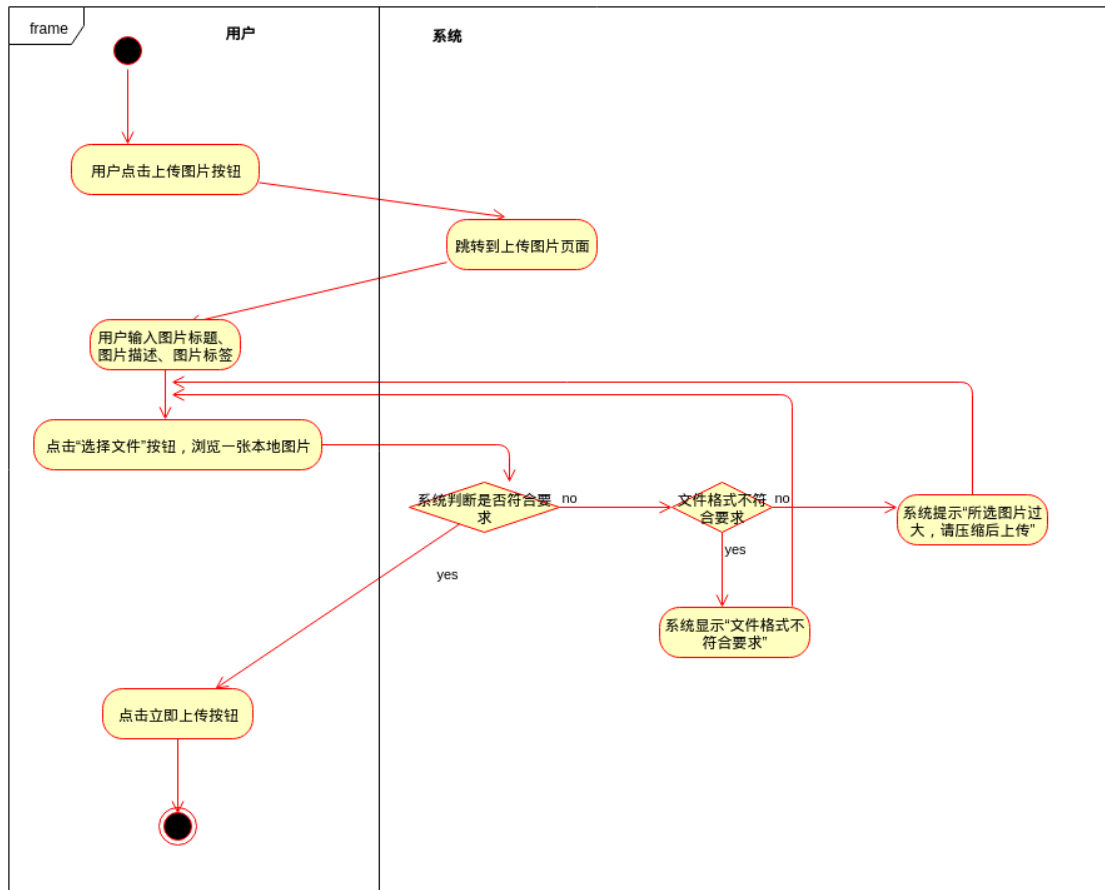


关注好友：关注好友,以便日后查阅该好友动态,建立自己的社交圈。

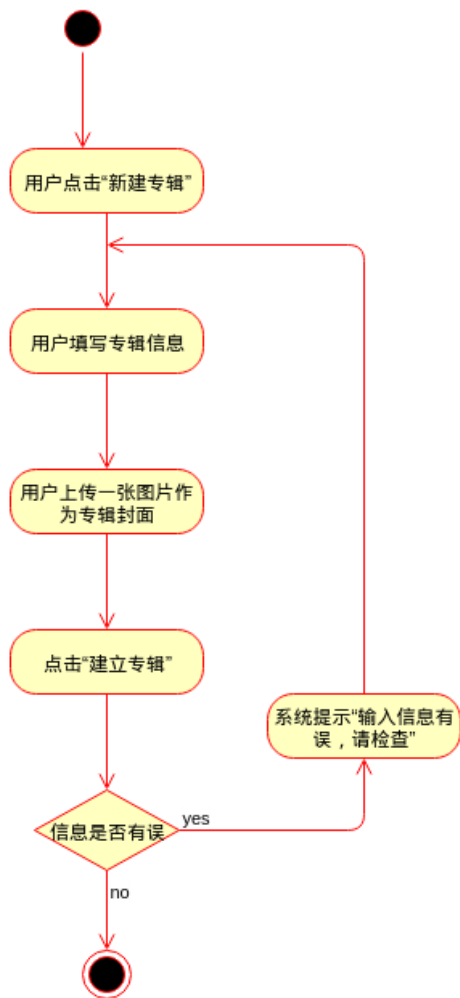


发起小组：用户发起一个小组, 以建立一个品味圈。

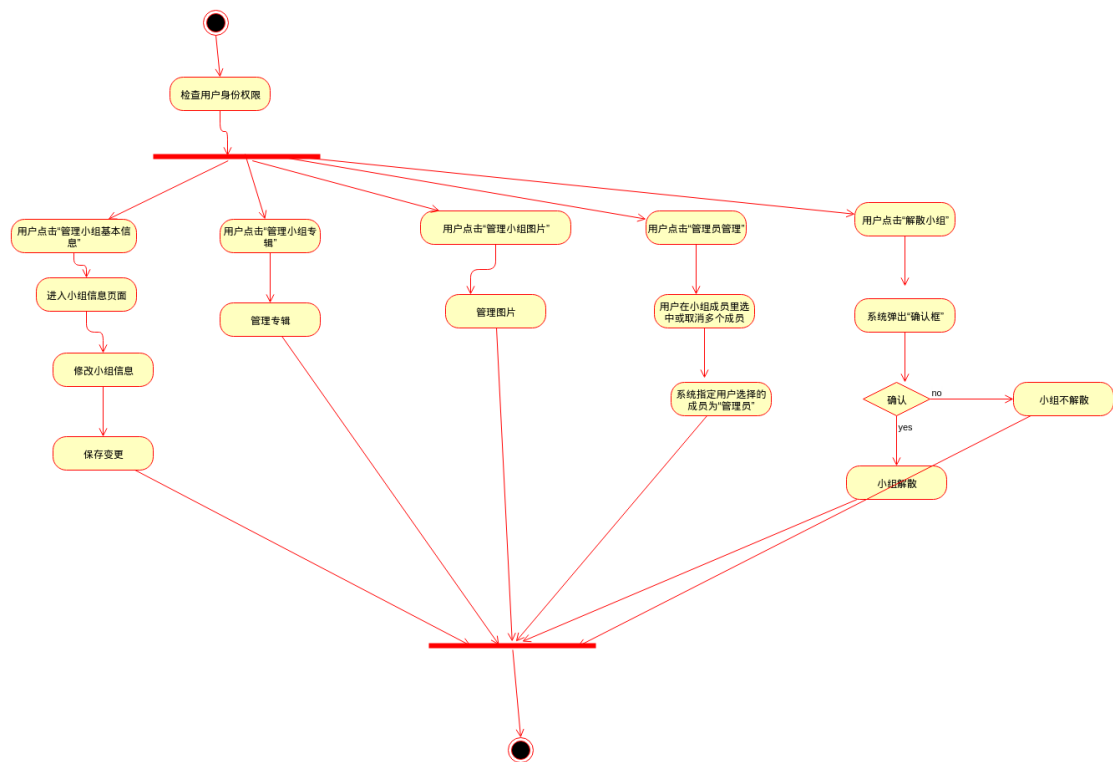
2.4.3.3 个人主页管理模块



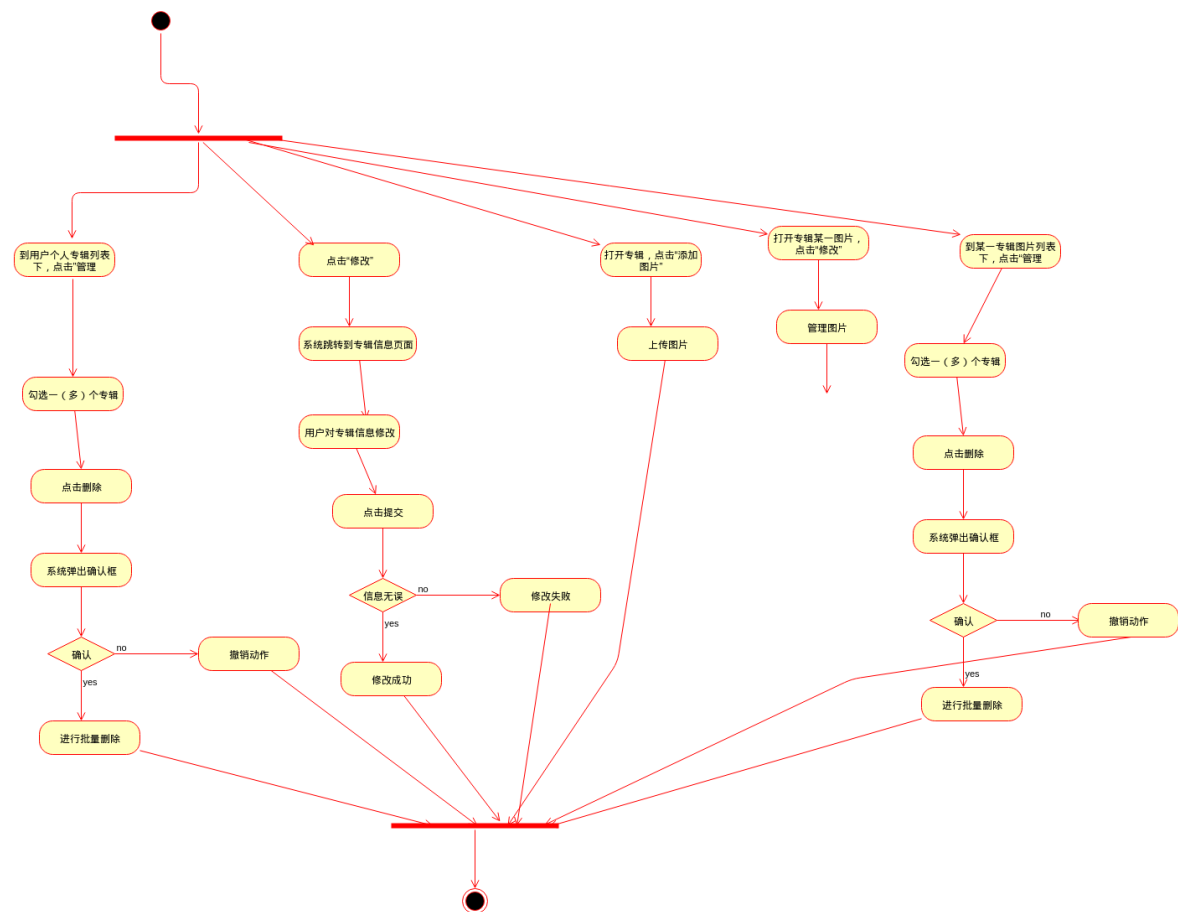
上传图片：用户上传一张本地的图片，作分享或展示用



新建专辑：用户新建一个专辑, 收录一类图片



管理小组：组长或者管理员, 对小组资料、小组成员以及小组所有的收录进行管理。



管理专辑：用户对已有的专辑进行管理, 比如修改专辑信息, 删除专辑, 向专辑中添加图片, 修改专辑中图片信息, 删除专辑中的图片等。

3. 非功能需求

3.1. 性能要求

3.1.1. 用户体验

本应用作为精美图片的分享平台, 应当关注平台的易用性、访问的流畅性和推荐的精确性。在界面设计上, 应当尽可能简洁明了、分类清晰, 能方便的进行功能性操作, 让用户能关注于图片浏览本身。在图片的加载和数据库查询上, 应尽可能高效, 避免卡顿造成的困扰。在

推荐算法的设计上，应充分利用和准确分析大数据，根据用户查询、收藏历史，为用户精确地定向推荐相关图片、专辑。

3.1.2. 时间特性要求

- 响应时间：

在网络状况良好的情况下，对于页面跳转的操作，响应时间应在 1~2s 之内，对于高清图的加载，响应时间不应超过 0.5s。涉及与第三方社交平台交互的部分，响应时间取决于与第三方平台的数据传输速度。

- 更新处理时间：

数据库的更新涉及网络数据传输，取决于网络传输速度。但正常情况下，数据库更新时间不应超过 1s，网络状况较差的情况下，不应超过 4s。采用消息中间件，异步更新，不影响主进程 UI 响应。

- 数据转换时间：

首次打开网站时需要进行大量的数据加载，时间较长，但不超过 3s。对于高并发、频繁访问的图片，浏览一次过后存入缓存数据库，以便再次浏览时迅速加载。

- 界面更新时间：

1. 界面加载采用异步加载的方式，优先加载文字。文字更新时间不应超过 0.1s，缩略图更新时间不应超过 0.2s。
2. 高清图加载采用先低清，再提升清晰度的方式。低像素图片加载不应超过 0.5s，高清图加载不应超过 1s。

3.1.3. 输入输出要求

1. 图片详情=图片基本信息+收藏情况+评论情况

- 1.1 图片基本信息=图片 id+图片+图片名称+描述+上传日期

- 1.1.1 图片 ID=*(0~100,000 之间正整)*

- 1.1.2 图片=*(png, jpeg, jpg, bmp 格式)*

- 1.1.3 图片名称=*(合法中文/英文字符)*

- 1.1.4 描述=*(任意长度的字符串)*

- 1.1.5 上传日期=年+月+日

- 1.2 收藏情况=收藏次数+收藏者

- 1.2.1 收藏次数=*(0~100,000 之间整数)*

- 1.2.2 收藏者=*(已注册的用户昵称)*

- 1.3 评论情况=评论次数+评论者

- 1.3.1 评论次数=*(0~100,000 之间整数)*

- 1.3.2 评论者=*(已注册用户的昵称)*

2. 专辑详情=专辑基本信息+收藏情况+评论情况

2.1 专辑基本信息=专辑 id+封面图片+专辑名称+描述+上传日期

2.1.1 专辑 ID=*(0~100, 000, 000 之间正整)*

2.1.2 专辑封面=*(png, jpeg, jpg, bmp 格式)*

2.1.3 专辑名称=*(合法中文/英文字符)*

2.1.4 描述=*(任意长度的字符串)*

2.1.5 上传日期=年+月+日

2.2 收藏情况=收藏次数+收藏者

2.2.1 收藏次数=*(0~100, 000 之间整数)*

2.2.2 收藏者=*(已注册的用户昵称)*

2.3 评论情况=评论次数+评论者

2.3.1 评论次数=*(0~100, 000 之间整数)*

2.3.2 评论者=*(已注册用户的昵称)*

3. 小组详情=小组基本信息+成员情况+互动情况

3.1 小组基本信息=小组 ID+小组名称+发起人留言+小组所属标签+小组创建时间

3.1.1 小组 ID=*(0~100, 000, 000 之间正整)*

3.1.2 小组封面=*(png, jpeg, jpg, bmp 格式)*

3.1.3 小组名称=*(合法中文/英文字符)*

3.1.4 发起人留言=*(任意长度的字符串)*

3.1.5 小组创建时间=年+月+日

3.1.6 小组所属标签=*(不多于 5 个中文字符或英文单词)*

3.2 成员情况=成员数量+成员名称

3.2.1 成员数量=*(0~500 之间整数)*

3.2.2 成员名称=*(已注册的用户昵称)*

3.3 互动情况=互动内容+发言人

3.3.1 互动内容=*(任意长度字符串)*

3.3.2 发言人=*(已注册用户的昵称)*

4. 搜索关键字=*(不得超过 32 个字符)*

5. 分享=分享图片+文字说明

6. 第三方平台授权=第三方社交平台用户名+密码

7. 上传图片=[jpg | jpeg | png | bmp | gif]

上传图片=*(一次上传图片大小不得超过 1M)*

3.2. 数据管理能力要求

3.2.1 常量约定

- 页面最多展示图片（专辑/小组）数量：16 个
- 用户最多翻页次数：100 次
- 每次查询返回条数：100 条
- 用户每日最大上传流量：4G
- 用户每日最大下载流量：4G

3.2.2 问卷和记录个数

本系统采用 mysql 数据库进行存储，共支持包括实体、关系数据库表格共 10 个，详细说明见（2.4.2 数据建模文字部分）。

3.3.3 数据存储要求

本应用采用 mysql 数据库进行数据存储，对于应对复杂的查询具有绝对优势，但是负载能力一般，为应对高负载的需求，本系统采用以下策略进行数据存储：

- 数据库表格的设计上，采用列维度和行纬度的拆分，以此提高数据的存储效率；
- 采用读写分离策略提高性能，读频率远大于写频率的数据单独存储；
- 对于系统中高并发、访问频繁的数据，采用缓存数据库对 mysql 进行隔离，防止 mysql 崩溃。
- 数据的备份采用单服务器集中备份的方法，保证数据的高安全性和高可靠性。

3.3.4 负载

本网站作为一个以图片分享为主题的分享应用，以保证访问的流畅性和应对高并发访问为导向，着重关注系统的负载能力。系统的负载能力由每秒请求数（Request per second）作为衡量指标，即每秒能成功处理的请求的个数。平均情况下，日 IP 量预计 25,000，人均页面浏览量预计 15 页，也就是说，每日的页面浏览量为：

$$PV_{daily} = IP_{daily} * n = 25,000 * 15 = 375,000$$

对于互联网应用，一天中，有 80%的请求发生在一天中的 40%时间内，由此计算单服务器情况下，平均每秒需要处理的请求个数：

每台服务器每秒处理请求的数量

$$\begin{aligned} &= ((80\% * PV_{total}) \div (24h * 60 * 60 * 40\%)) \div N_{server} \\ &= ((80\% * 375,000) \div (24h * 60 * 60 * 40\%)) \div 1 \\ &= 8.68 \text{ 个请求 / 秒} \end{aligned}$$

最大的挑战在于应对高峰期大量用户的并发访问。在压力测试情况下，高峰期的访问量是平均访问量的 2~3 倍，即服务器需要在每秒响应 17.36~26.04 个请求。

20%的功能会带来 80%的流量，对于 80%的功能，系统面对的请求压力是很小的，是没有必要进行过度设计的。但是对于另外 20%的功能则是着重进行负载均衡和优化，采用缓存、分布式存储和异步更新等策略。

3.3. 安全及保密性要求

3.3.1 软件使用数据

本应用存储的数据涉及用户上传的图片、专辑小组信息、用户的偏好信息等。一旦丢失将导致用户个人上传记录、收藏历史的遗失，推荐算法无法根据历史记录为用户定向推送，导致用户体验感急剧下降。因此本应用对于数据的安全性要求较高，要尽量保证用户的数据不会丢失。

1. 用户上传的图片采用多重备份的方式存储。采用分布式存储的方式，图片信息及其副本存放在不同的机器上，即使一台机器宕机，仍能保证数据的安全存储，防止由于掉电、机械故障等物理原因导致的数据丢失。

2. 用户的偏好信息由收藏记录、上传记录、小组及好友记录构成，为了保证数据的新鲜度，本应用对用户近 30 次的使用记录进行存储。由于特殊原因造成的个别记录丢失，并不影响系统按照用户偏好记录进行推送。

3.3.2 用户账户数据

1. 应用仅对用户在本系统的用户名、密码等进行保存，个人信息仅与本网站功能相关，不会对用户隐私造成任何不利影响。

2. 对于用户在本系统中对于第三方社交平台的授权信息，系统将进行加密存储，严格防止用户第三方社交平台的信息泄漏。应用不会利用用户第三方社交平台进行用户指定的分享之外的任何操作。

3.4. 灵活性要求

1. 采用敏捷开发 SCRUM 模型，拥抱变化，团队已一周为一个 sprint，定期开 SCRUM Daily 会议，并进行 Retrospective 总结会议。采用 iceScrum 辅助管理开发进程，依照每个 sprint 的实际 Velocity 制定下一个 sprint 的工作计划。整个团队采取自组织的工作模

式，所有成员均担当多方面任务，并根据个人能力主动选择 sprint 中的 backlog 进行工作，无需上级干预。

2. 采用测试驱动开发的开发模式（TDD），尽可能保证开发高效率和高准确度的完成代码编写。
3. 部署 Code Review 环节，尽可能提高每一次代码编写质量，提高工作效率。
4. 采用 Git 版本控制系统进行版本控制。
5. 软件架构采用 MVC 模式，使得界面和逻辑充分脱耦，提高灵活性。
6. 对于相同类型逻辑定义采用 Protocol 的方法进行抽象，同时注意类的高内聚和封装。

3.5. 其他专门要求

3.5.1 可维护性

可维护性是指在不影响系统其他部分的情况下修改现有系统功能中问题或缺陷的能力。开发人员创建和设计系统架构时，为了提高系统的可维护性，必须考虑以下几个方面的要素：低耦合、高内聚合系统文档记录？。本系统将采用严格的软件工程的规范进行开发，并采用良好的设计模式保证系统各模块之间的低耦合及模块之间的高内聚。

本系统的所有代码将会被详细注释，对于系统所有代码，我们会生成详尽的技术文档。对于系统开发过程可能出现的报错，我们将以文档的方式详细罗列报错码及对应的报错信息。

3.5.2 灾难恢复

由于本应用为服务器—客户端模式的网络应用，用户数据保存在服务器端，因此服务端负责数据备份并保证数据的安全性。数据的备份存储如（。）所述，即使发生灾难造成数据丢失，系统立即通过备份数据进行恢复，并且重新备份，保证系统中始终保持至少一份数据备份。

3.5.3 法律限制

3.5.3.1 版权限制

本网站致力于打造开放的高质量图片分享交流社区，不涉及任何盈利目的。我们鼓励用户充分利用本应用自由地张贴和共享自己的信息。用户可以自由张贴图片等内容，但这些内容必须位于公共领域内，或者用户本人拥有这些内容的使用权。同时，用户不应在自己的个人主页或社区中张贴其他受版权保护的内容。我们如果收到按下述程序提起的正式版权投诉，将会删除这些内容。用户对于其创作并在本应用平台上发布的合法内容依法享有著作权及其相关权利。

3.5.3.2 合法内容限制

用户在使用本应用提供的服务过程中，必须遵循国家的相关法律法规，不得利用本平台，发布危害国家安全、色情、暴力、凶杀、恐怖等非法内容；不得利用本平台发布含有虚假、有害、胁迫、侵害他人隐私、骚扰、侵害、中伤、粗俗、或其它道德上令人反感的内容。参见相关法规。

用户使用本服务的行为若有任何违反上述条款的情形时，本应用有权直接删除该等违反条款之内容。用户影响系统总体稳定性或完整性的操作可能会被暂停或终止，直到问题得到解决。

3.5.3.3 责任限制

互联网是一个开放平台，用户将图片等资料上传到互联网上，有可能会被其他组织或个人复制、转载、擅改或做其它非法用途，用户必须充分意识此类风险的存在。用户明确同意其使用本应用服务所存在的风险将完全由其自己承担；因其使用本应用服务而产生的一切后果也由其自己承担，本应用对用户不承担任何责任。