

ImageS需求分析报告

|  |  |
| --- | --- |
| 班级： | 计科1604 |
| 小组成员： | 赖思羽 16281165  王晓云 16281174  扎西拉姆 16281179  郑晖 16281183 |
| 日期： | 2019年5月12日 |

目录

[1. 引言 4](#_Toc11166711)

[1.1. 编写目的 4](#_Toc11166712)

[1.2. 参考资料 4](#_Toc11166713)

[2. 功能需求 4](#_Toc11166714)

[2.1. 系统范围 4](#_Toc11166715)

[2.2. 系统体系结构 4](#_Toc11166716)

[2.2.1. 数据模块 4](#_Toc11166717)

[2.2.2. 模型层 5](#_Toc11166718)

[2.2.3. 控制器层 5](#_Toc11166719)

[2.2.4. 视图层 5](#_Toc11166720)

[3. 外部接口需求 6](#_Toc11166721)

[3.1. 用户接口 6](#_Toc11166722)

[3.2. 软件接口 7](#_Toc11166723)

[3.3. 硬件接口 7](#_Toc11166724)

[4. 数据模型 8](#_Toc11166725)

[4.1. 用户类（User） 8](#_Toc11166726)

[4.2. 小组类（Group） 9](#_Toc11166727)

[4.3. 小组评论类（GroupComment） 9](#_Toc11166728)

[4.4. 好友类（Friends） 9](#_Toc11166729)

[4.5. 专辑类（Album） 9](#_Toc11166730)

[4.6. 图片类（Picture） 10](#_Toc11166731)

[4.7. 图片评论类（PicComment） 10](#_Toc11166732)

[4.8. 图片标签类（PicTag） 10](#_Toc11166733)

[5. 分析模型 10](#_Toc11166734)

[5.1. 数据字典 10](#_Toc11166735)

[5.2. 数据对象ER图 15](#_Toc11166736)

[6. 非功能需求 17](#_Toc11166737)

[6.1. 性能要求 17](#_Toc11166738)

[6.1.1. 用户体验 17](#_Toc11166739)

[6.1.2. 时间特性要求 17](#_Toc11166740)

[6.1.3. 输入输出要求 18](#_Toc11166741)

[6.2. 数据管理能力要求 19](#_Toc11166742)

[6.2.1. 常量约定 19](#_Toc11166743)

[6.2.2. 数据存储要求 20](#_Toc11166744)

[6.3. 安全及保密性要求 20](#_Toc11166745)

[6.3.1. 软件使用数据 20](#_Toc11166746)

[6.3.2. 用户账户数据 20](#_Toc11166747)

[7. 版本历史 21](#_Toc11166748)

# 引言

## 编写目的

文档旨在对“ImageS”项目进行详细的需求分析，保证软件开发的质量、需求的完整与[可追溯性](https://www.baidu.com/s?wd=%E5%8F%AF%E8%BF%BD%E6%BA%AF%E6%80%A7&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)，编写此文档。通过此文档，以保证业务需求提出者与需求分析人员、开发人员、测试人员及其也相关利益人对需求达成共识。

## 参考资料

1. 《软件工程导论（第六版）》，张海藩、牟永敏编著，清华大学出版社。

# 功能需求

## 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。

## 系统体系结构

### 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。

### 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。

### 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。

### 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。

# 外部接口需求

## 用户接口

|  |  |
| --- | --- |
| 社交管理模块 | 评论图片接口 |
| 评价小组接口 |
| 评价专辑接口 |
| 收藏图片接口 |
| 收藏专辑接口 |
| 关注好友接口 |
| 加入小组接口 |
| 分享到第三方平台接口 |
| 浏览模块 | 最热图片推荐接口 |
| 最热小组推荐接口 |
| 最热专辑推荐接口 |
| 最近图片推荐接口 |
| 最近小组推荐接口 |
| 最近专辑推荐接口 |
| 按照喜好图片推荐接口 |
| 按照喜好小组推荐接口 |
| 按照喜好专辑推荐接口 |
| 查看图片详情 |
| 查看小组详情 |
| 查看专辑详情 |
| 个人主页查看上传图片接口 |
| 个人主页查看收藏图片接口 |
| 个人主页查看新建专辑接口 |
| 个人主页查看收藏专辑接口 |
| 个人主页查看发起小组接口 |
| 个人主页查看用户所在小组接口 |
| 搜索接口 |
| 个人账户管理模块 | 登录接口 |
| 注册接口 |
| 登出接口 |
| 个人信息修改接口 |
| 个人主页管理模块 | 修改小组信息接口 |
| 删除小组接口 |
| 发起小组接口 |
| 删除专辑接口 |
| 新建专辑接口 |
| 修改专辑信息接口 |
| 上传图片接口 |
| 修改图片信息接口 |
| 删除图片接口 |

## 软件接口

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

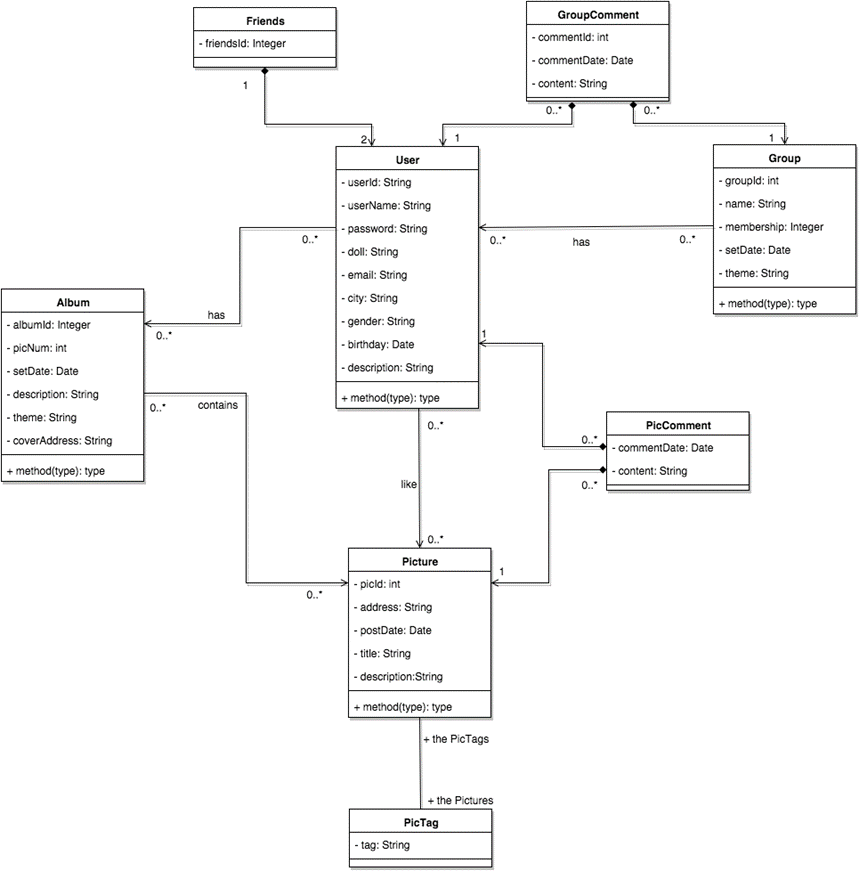
## 硬件接口

1. 账户授权协议:OAuth 2.0

2. 网络访问协议:TCP/IP,HTTP

3. 数据压缩:objective-zip

# 数据模型



## 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。

## 小组类（Group）

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。

## 小组评论类（GroupComment）

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。

## 好友类（Friends）

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。

## 专辑类（Album）

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。

## 图片类（Picture）

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。

## 图片评论类（PicComment）

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。

## 图片标签类（PicTag）

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。

# 分析模型

## 数据字典

Album（专辑）

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 描述 |
| albumId | Integer | 专辑 ID，主键 |
| user | User | 新建专辑的用户 |
| picNum | int | 专辑拥有的图片数目 |
| setDate | Date | 专辑的创建日期 |
| description | String | 专辑的描述 |
| theme | String | 专辑主题 |
| coverPath | String | 封面图片的本地路径 |

AlbumIncluded（专辑－图片）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| albumId | int | 专辑，主键 1 |
| picId | Integer | 专辑包含的图片，主键 2 |

AlbumLiked（专辑－用户关系集：收藏该专辑的用户）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| albumId | int | 专辑，主键 1 |
| userId | Integer | 收藏专辑的用户，主键 2 |

Friends（用户－用户）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| friendId | Integer | 好友用户，主键 1 |
| myId | int | 用户本人，主键 2 |

GroupComment （小组－评论）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| commentId | Integer | 评论 ID，主键 |
| group | Group | 被评论的小组 |
| user | User | 发起评论的用户 |
| commentDate | Date | 评论日期 |
| content | String | 评论内容 |

Group（小组）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| groupId | Integer | 小组 ID，主键 |
| user | User | 发起小组的用户 |
| name | String | 小组名称 |
| membership | Integer | 小组成员数 |
| setDate | Date | 发起小组的日期 |
| theme | String | 小组的主题 |

GroupMem （小组－用户关系集：小组及其成员）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| groupId | Integer | 小组，主键 1 |
| userId | Integer | 小组包含的用户，主键 2 |

User（用户）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| userId | Integer | 用户 ID |
| userName | String | 用户姓名 |
| password | String | 用户密码 |
| avatar | String | 用户头像 |
| email | String | 用户邮箱地址 |
| city | String | 用户所在城市 |
| gender | String | 用户性别 |
| birthday | Date | 用户生日 |
| discription | String | 用户个人描述 |

Pictures（图片）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| picId | Integer | 图片 ID，主键 |
| user | User | 上传图片的用户 |
| path | String | 图片的本地地址 |
| uploadDate | Date | 上传日期 |
| title | String | 图片标题 |
| description | String | 图片描述 |

PicComment （图片评论）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| commentId | Integer | 关系集 ID，主键 |
| pictures | Pictures | 被评论的图片 |
| user | User | 发起评论的用户 |
| commentDate | Date | 评论日期 |
| content | String | 评论内容 |

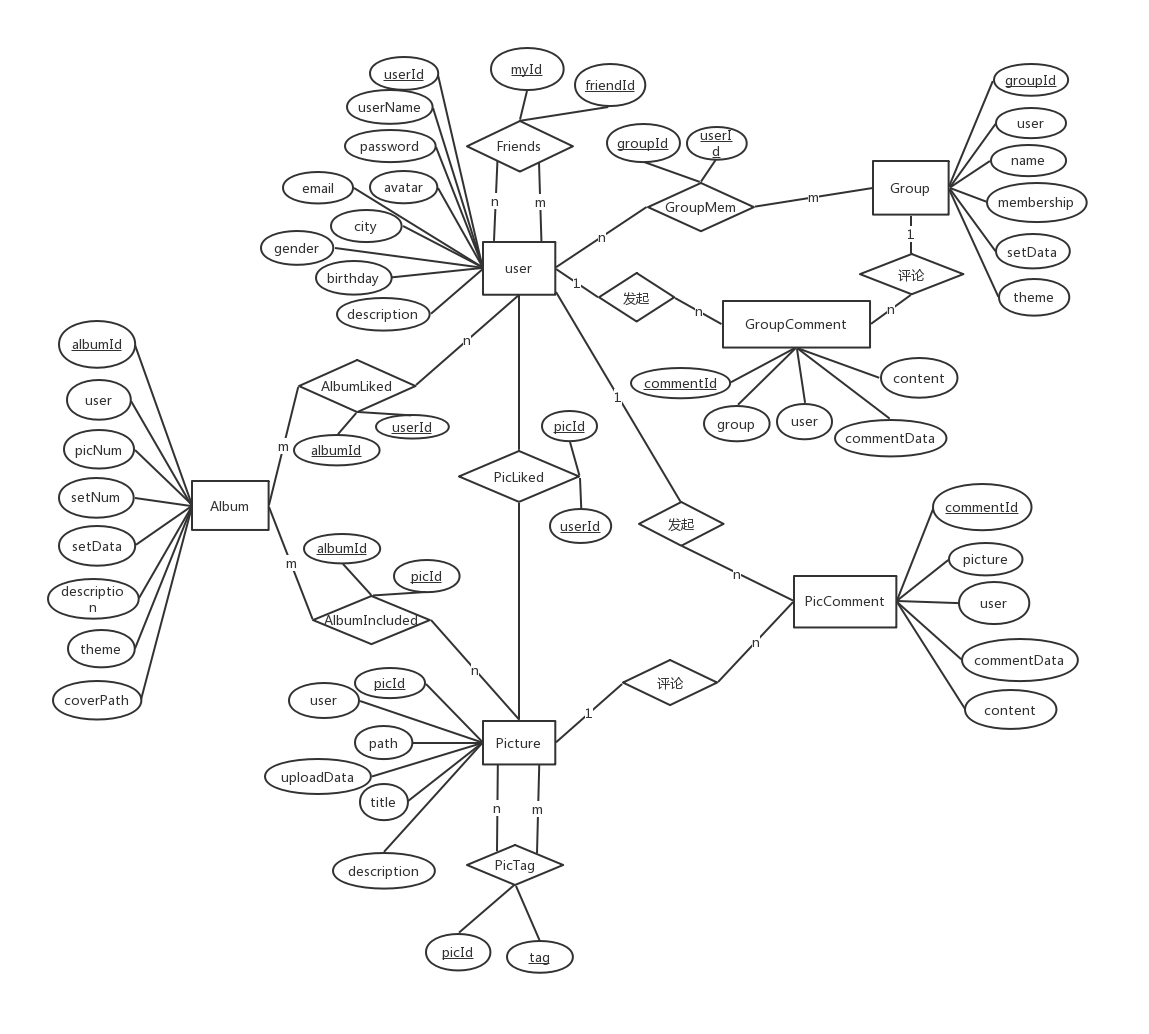
PicLiked （图片－用户关系集：图片与收藏它的用户）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| picId | Integer | 被收藏的图片，主键 1 |
| userId | Integer | 收藏图片的用户，主键 2 |

PicTag （图片－标签关系集：图片所拥有的标签）

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| tag | String | 图片对应的标签，主键 1 |
| pictureId | Picture | 图片，主键 2 |

## 数据对象ER图



一个专辑可以被多个用户收藏，一个用户可以收藏多个专辑，故Album（专辑）与User（用户）是多对多的关系。因此建立AlbumLiked关系集，描述被收藏的专辑与收藏该专辑的用户之间的关系。

一张图片可以被多个用户收藏，一个用户可以收藏多张图片，故 Picture（图片）与 User （用户）是多对多的关系。因此建立 PicLiked 关系集，描述被收藏的图片与收藏该图片的用户 之间的关系。

一个用户可以加入多个小组，一个小组有多个成员用户，故 Group（小组）与 User（用户）是多对多的关系。因此建立 GroupMem 关系集，描述小组及其成员用户之间的关系。

一个用户可以关注多个好友，一个用户也可以被多个其他用户关注，故用户（User）与用 户（User）之间是多对多的关系。因此建立 Friends 关系集，描述用户与用户之间的好友关系。

一个专辑可以包含多张图片，一张图片可以属于多个专辑，故 Picture（图片）与 Album （专辑）是多对多的关系。因此建立 AlbumIncluded 关系集，描述专辑及其所包含的图片之间 的关系。

一张图片可以有多个标签，一个标签可以标记多多幅图片，因此 Tag（标签）与 Picture （图片）是多对多的关系，但是 Tag（标签）实体只有标签名这一个属性，故将该实体合并到 标签-图片关系集中，标签-图片关系集记为 PicTag，描述图片及其所属标签的关系。

一个用户可以多次发起小组评论，而一条评论只能有一个发起者，故 User（用户）和 GroupComment 之间是一对多的关系。

一个小组可以被多次评论，而一条评论只能针对一个小组，故 Group（小组）和 GroupComment 之间是一对多的关系。

一个用户可以多次发起图片评论，而一条评论只能有一个发起者，故 User（用户）和 PicComment 之间是一对多的关系。

一张图片可以被多次评论，而一条评论只能针对一张图片，故 Picture（图片）和 PicComment 之间是一对多的关系。

# 非功能需求

## 性能要求

### 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。

### 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，对于高清大图的加载，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. 高清图片加载采用先低清，再提升清晰度的方式。低像素图片加载不应超过 0.5s，高清图片加载不应超过1s。

### 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

图片基本信息＝图片 id＋图片＋图片名称＋描述＋上传日期

图片 ID＝\*(0～100, 000, 000 之间正整)\*

图片＝\*(png, jpeg, jpg, bmp 格式)\*

图片名称＝\*(合法中文/英文字符)\*

描述＝\*(任意长度的字符串)\*

上传日期＝年＋月＋日

收藏情况＝收藏次数＋收藏者

收藏次数＝\*(0～100，000 之间整数)\*

收藏者＝\*(已注册的用户的昵称)\*

评论情况＝评论次数＋评论者

评论次数＝\*(0～100，000 之间整数)\*

评论者＝\*(已注册用户的昵称)\*

1. 专辑详情＝专辑基本信息＋收藏情况＋评论情况

专辑基本信息＝专辑 id＋封面图片＋专辑名称＋描述＋上传日期

专辑 ID＝\*(0～100, 000, 000 之间正整)\*

专辑封面＝\*(png, jpeg, jpg, bmp 格式)\*

专辑名称＝\*(合法中文/英文字符)\*

描述＝\*(任意长度的字符串)\*

上传日期＝年＋月＋日

收藏情况＝收藏次数＋收藏者

收藏次数＝\*(0～100，000 之间整数)\*

收藏者＝\*(已注册的用户的昵称)\*

评论情况＝评论次数＋评论者

评论次数＝\*(0～100，000 之间整数)\*

评论者＝\*(已注册用户的昵称)\*

1. 小组详情＝小组基本信息＋成员情况＋互动情况

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

小组 ID＝\*(0～100, 000, 000 之间正整)\*

小组封面＝\*(png, jpeg, jpg, bmp 格式)\*

小组名称＝\*(合法中文/英文字符)\*

发起人留言＝\*(任意长度的字符串)\*

小组创建时间＝年＋月＋日

小组所属标签＝\*(不多于 5 个中文字符或英文单词)\*

成员情况＝成员数量＋成员名称

成员数量＝\*(0～500 之间整数)\*

成员名称＝\*(已注册的用户的昵称)\*

互动情况＝互动内容＋发言人

互动内容＝\*(任意长度字符串)\*

发言人＝\*(已注册用户的昵称)\*

1. 搜索关键字＝\*(不得超过 32 个字符)\*
2. 分享＝分享图片＋文字说明
3. 第三方平台授权＝第三方社交平台用户名＋密码
4. 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)\*

## 数据管理能力要求

### 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

### 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。

## 安全及保密性要求

### 软件使用数据

* 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。
* 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。

### 用户账户数据

* 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。
* 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。

# 版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 编写日期 | 说明 | 参与人员 |
| 0.1 | 2019-05-11 | 初稿 | 赖思羽、王晓云 |
| 0.2 | 2019-05-12 | 数据字典 | 王晓云 |
| 0.3 | 2019-05-12 | 修改定稿 | 郑晖、扎西拉姆 |
| 0.4 | 2019-06-11 | 添加版本历史 | 郑晖 |