1. 引言
   1. 编写目的

在本计划书中，确定应该如何具体地实现所要求的系统，从而在编码阶段可以把描述直接翻译成用具体的 程序语言书写的程序。主要的工作有：根据在《需求分析说明书》中所描述的数据、功能、运行、性能需求，并依照《软件开发计划书》所确定的处理流程、总体结构和模块外部设计，设计软件系统的结构设计、逐个模块的程序描述（包括各模块的功能、性能、输入、输出、算法、程序逻辑、接口等等）。

* 1. 命名规范

数据库表名，字段名用大写英文命名

私有变量用”\_”开头的小写英文命名，其他的变量用大写英文字母开头

所有函数使用驼峰表示法，类名首字母大写

测试采用W模型。

* 1. 术语定义

SQL：一种用于访问查询数据库的语言

SQL SERVER:系统服务器所使用的数据库关系系统（DBMS）。特指为MySQL

服务器：特指为Apache软件，Django框架，MySQL搭建的专用服务器

事务流：数据进入模块后可能有多种路径进行处理。

主键：数据库表中的关键域。值互不相同。

外部主键：数据库表中与其他表主键关联的域。

APP：若未特别指出，统指本图片社交软件。

广场：APP中按照一定顺序展现图片的页面。

标签：由上传图片者自定义的图片描述。

评论：对于某一张特定图片的评价。

缩略图：对原图片进行剪切压缩等处理后获得的适应不同终端的图片。

* 1. 参考资料

吕云翔，王昕鹏，邱玉龙.软件工程——理论与实践.北京：人民邮电出版社2012

图片社交软件Dolphin的《软件开发计划书》

图片社交软件Dolphin的《需求规格说明书》

* 1. 相关文档

图片社交软件Dolphin的《软件开发计划书》

图片社交软件Dolphin的《需求规格说明书》

图片社交软件Dolphin的《测试分析报告》

* 1. 版本更新信息

版本更新记录如表1所示。

表 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 创建者 | 创建日期 | 维护者 | 维护日期 | 维护记录 |
| V0.1.0 | 王浩然 | 2015-4-26 |  |  |  |
| V0.1.1 |  |  | 王浩然 | 2015-4-26 | 补全了功能模块清单 |
| v0.1.2 |  |  | 赵冯博 | 2015-5-5 | 补全了数据库表名和数据库表的详细清单 |
| v0.1.3 |  |  | 孙晓宇 | 2015-5-5 | 补全了功能模块设计的详细内容 |
| v0.1.4 |  |  | 谭梦辰 | 2015-5-18 | 添加了测试计划说明 |
| v0.1.5 |  |  | 赵冯博 | 2015-5-18 | 增加了TAG表，修改了用户表中错误的命名 |
| v0.1.6 |  |  | 赵冯博 | 2015-5-20 | 修改了ID的数据字典，增加了用户表的属性 |
| v0.1.7 |  |  | 王浩然 | 2015-5-21 | 加入了本地数据库 |
| v0.1.8 |  |  | 王浩然 | 2015-5-22 | 加入了本地数据库的存储过程设计 |
| v0.1.9 |  |  | 孙晓宇 | 2015-5-22 | 补充内部接口设计和系统错误设计 |
| V0.1.10 |  |  | 王浩然 | 2015-5-23 | 检查并修改了本地数据库的存储设计 |
| V0.1.11 |  |  | 崔永恩 | 2015-5-23 | 补全引言 |
| V0.1.12 |  |  | 赵冯博 | 2015-5-23 | 加入了数据库的存储设计 |
| V1.0 |  |  | 王浩然 | 2015-5-23 | 加入了总体设计，第一遍校阅 |
| V1.0.1 |  |  | 谭梦辰 | 2015-5-24 | 更改角色授权表，改正错别字，第二遍校阅 |
| V1.0.2 |  |  | 谭梦辰 | 2015-5-24 | 增加术语定义,更正模块内容，更正部分术语大小写错误 |
| V1.0.3 |  |  | 王浩然 | 2015-5-25 | 修改了本地数据库表的错误 |
| V1.0.4 |  |  | 王浩然 | 2015-5-26 | 增添了本地数据库 |

1. 总体设计
   1. 硬件运行环境
2. 服务器

处理器型号：AMD或Intel 2.00GHz以上

内存容量：2GB以上

网络配置：100M网卡

1. Android手机

处理器型号：1Ghz以上

内存容量：500M以上

* 1. 软件运行环境

1. 服务器

操作系统：Windows Server 2008及以上版本系统

数据库：MySQL

Web服务器:Apache

网络架构：Django framework

1. Android手机

操作系统：Android 4.3.1及以上

* 1. 开发环境

1. 硬件环境

本系统采用PC开发，配置如下。

处理器型号：AMD/Intel 2GHz及以上。

内存剩余空间：1GB及以上

外存剩余空间：2GB及以上

网络配置：100M网卡，串口。

1. 软件环境

服务器

操作系统：Windows Server 2008及以上版本系统

数据库：MySQL

Web服务器:Apache

网络架构：Django framework

* 1. 子系统清单

子系统清单如表2所示

|  |  |  |
| --- | --- | --- |
| 子系统编号 | 子系统名称 | 子系统功能简述 |
| SS1 | 登陆注册 | 1. 会员登陆系统时，对其身份进行检验和识别 2. 游客（新用户）可以进行注册 3. 注册用户可以修改密码和注销等 |
| SS2 | 图片管理 | 1. 注册用户可以拍摄、上传和删除图片 2. 注册用户可以添加和删除图片标签 |
| SS3 | 图片浏览 | 1. 游客和注册用户可以浏览图片广场中的图片 2. 注册用户可以查看图片详细信息 3. 注册用户可以下载图片 4. 注册用户可以查看自己和自己关注的人上传的所有图片 5. 注册用户可以按标签搜索图片 |
| SS4 | 图片评价 | 1. 注册用户可以评论图片 2. 注册用户可以修改自己的评论 3. 注册用户可以为图片点赞和取消赞 |
| SS5 | 社交功能 | 1. 注册用户可以按用户名搜索其他用户 2. 注册用户可以访问其他用户的主页 3. 注册用户可以关注其他用户，也可以取消关注 4. 已关注的人的新动态会推送给注册用户 5. 注册用户可以将其他用户加入黑名单，禁止这些用户关注自己，也可以从黑名单中删除某些用户 |

* 1. 功能模块清单

功能模块清单如表3所示

|  |  |  |
| --- | --- | --- |
| 模块编号 | 名称 | 模块功能描述 |
| SS1-1 | 用户注册 | 游客注册成为会员 |
| SS1-2 | 会员登陆 | 会员登陆系统 |
| SS1-3 | 用户注销 | 会员离开系统时，进行注销 |
| SS1-4 | 修改密码 | 会员修改个人密码 |
| SS2-1 | 拍摄图片 | 会员通过手机内置照相机拍摄图片 |
| SS2-2 | 上传图片 | 会员将图片上传分享 |
| SS2-3 | 删除图片 | 会员删除自己已上传的图片 |
| SS2-4 | 添加图片标签 | 会员为自己上传的图片添加标签 |
| SS2-5 | 删除图片标签 | 会员删除自己添加的标签 |
| SS3-1 | 浏览广场 | 游客和会员进入广场页面，可浏览服务器分配的图片 |
| SS3-2 | 查看图片详细信息 | 会员可查看图片的详细信息如上传者、上传日期、赞的数量以及评论 |
| SS3-3 | 下载图片 | 会员可将图片下载到本地 |
| SS3-4 | 查看自己的个人主页 | 会员可进入自己的个人主页，通过引导栏查看自己或已关注人的图片 |
| SS3-5 | 查看自己的所有图片 | 会员向上滑动屏幕可查看自己上传的图片 |
| SS3-6 | 查看已关注人的所有图片 | 会员向下滑动屏幕可查看自己已关注人的图片 |
| SS3-7 | 搜索图片标签 | 会员可在搜索框中输入要搜索的标签来搜索图片 |
| SS4-1 | 评论图片 | 会员可评论已关注人的图片 |
| SS4-2 | 修改自己的评论 | 会员可修改自己的评论 |
| SS4-3 | 为图片点赞 | 会员可以对图片点赞 |
| SS4-4 | 取消对图片的赞 | 会员可以取消已点赞图片的赞 |
| SS5-1 | 搜索用户名 | 会员可在搜索框中输入用户名来搜索用户 |
| SS5-2 | 访问用户主页 | 会员可以访问其他用户的主页，如果已关注该用户，则主页中有该用户的所有图片以及取消关注按钮，否则只有关注按钮 |
| SS5-3 | 关注用户 | 会员可关注某个用户 |
| SS5-4 | 取消关注 | 会员可取消对某个用户的关注 |
| SS5-5 | 查看所有关注的人的最新动态 | 系统会将会员关注的人的动态推送给会员 |
| SS5-6 | 查看已关注的名单 | 会员可查看自己已关注人的名单 |
| SS5-7 | 将其他用户加入黑名单 | 会员可将某些用户加入黑名单，禁止这些用户关注自己 |
| SS5-8 | 从黑名单中删除用户 | 会员可从黑名单中删除某些用户 |
| SS5-9 | 查看黑名单 | 会员可查看自己的黑名单列表 |

1. 数据库设计
   1. 数据库中的表名列表

数据库中表名列表格式如表所示

|  |  |  |
| --- | --- | --- |
| 编号 | 表名 | 表功能说明 |
| 001 | m\_USER | 注册用户表 |
| 002 | m\_IMAGE | 图片表 |
| 003 | m\_TAG | 标签表 |
| 004 | m\_COMMENT | 评论表 |
| 005 | m\_LIKE | 赞表 |
| 006 | m\_CONCERN | 关注表 |
| 007 | m\_BLACKLIST | 黑名单表 |

* 1. 数据库表的详细清单

m\_USER表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | U\_ID | 用户ID | string(32) | 否 | 主键 |
| 2 | U\_name | 用户名 | varchar(30) | 否 | unique |
| 3 | U\_password | 用户密码 | varchar(16) | 否 |  |
| 4 | U\_mail | 用户邮箱 | varchar(30) | 否 |  |
| 5 | Space\_remained | 剩余空间 | int | 否 |  |

m\_IMAGE表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | I\_ID | 图片ID | string(32) | 否 | 主键 |
| 2 | U\_ID | 上传用户ID | string(32) | 否 | 外键 |
| 3 | I\_space | 图片大小 | int | 否 |  |
| 4 | Like\_number | 赞的数量 | int | 否 |  |
| 5 | m\_Priority | 权值 | int | 否 |  |
| 6 | Update\_date | 上传时间 | date | 否 |  |

m\_TAG表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | T\_ID | 标签ID | string(32) | 否 | 主键 |
| 2 | U\_ID | 用户ID | string(32) | 否 | 外键 |
| 3 | I\_ID | 图片ID | string(32) | 否 | 外键 |
| 4 | tag | 标签内容 | varchar(10) | 否 |  |

m\_COMMENT表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | C\_ID | 评论ID | string(32) | 否 | 主键 |
| 2 | U\_ID | 评论者ID | string(32) | 否 | 外键 |
| 3 | I\_ID | 图片ID | string(32) | 否 | 外键 |
| 4 | Update\_date | 上传时间 | date | 否 |  |
| 5 | Comment | 评论内容 | varchar(256) | 否 |  |

m\_LIKE表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | U\_ID | 用户ID | string(32) | 否 | 主键、外键 |
| 2 | I\_ID | 图片ID | string(32) | 否 | 主键、外键 |

m\_CONCERN表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | U\_ID\_from | 关注者ID | string(32) | 否 | 主键、外键 |
| 2 | U\_ID\_to | 被关注者ID | string(32) | 否 | 主键、外键 |

m\_BLACKLIST表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | U\_ID\_from | 黑名单拥有者ID | string(32) | 否 | 主键、外键 |
| 2 | U\_ID\_to | 被添加者ID | string(32) | 否 | 主键、外键 |

1. 手机端数据库设计
   1. 数据库中的表名列表

数据库中表名列表格式如下表所示

|  |  |  |
| --- | --- | --- |
| 编号 | 表名 | 表功能说明 |
| 001 | M\_USER | 注册用户表 |
| 002 | M\_LASTUSER | 最后登录的注册用户表 |
| 003 | M\_IMAGE | 图片表 |
| 004 | m\_TAG | 标签表 |
| 005 | m\_COMMENT | 评论表 |
| 006 | M\_LOBBYIMAGE | 大厅图片表 |
| 007 | M\_IMAGECARED | 关注人图片表 |

* 1. 数据库表的详细清单

M\_USER表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | M\_USER\_ID | 注册用户ID | TEXT | 否 | 主键 |
| 2 | M\_USER\_PASSWORD | 注册用户PASSWORD | TEXT | 是 |  |

M\_LASTUSER表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | M\_LASTUSER\_ID | 最后登录的注册用户ID | TEXT | 否 | 主键、外键 |
| 2 | M\_LASTUSER\_PASSWORD | 最后登录的注册用户PASSWORD | TEXT | 否 | 外键 |

M\_IMAGE表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | M\_IMAGE\_IMAGEID | 图片ID | TEXT | 否 | 主键 |
| 2 | M\_IMAGE\_USERID | 上传用户ID | TEXT | 是 | 外键 |
| 3 | M\_IMAGE\_ISLIKE | 是否赞 | TEXT | 是 |  |
| 4 | M\_IMAGE\_LIKENUMBER | 赞的数量 | TEXT | 是 |  |
| 5 | M\_IMAGE\_UPDATEDATE | 上传时间 | DATETIME | 是 |  |

M\_TAG表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | M\_TAG\_ID | 标签ID | TEXT | 否 | 主键 |
| 2 | M\_TAG\_IMAGEID | 图片ID | TEXT | 否 | 外键 |
| 3 | M\_TAG\_NAME | 标签内容 | TEXT | 否 |  |

M\_COMMENT表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | M\_COMMENT\_COMMENTID | 评论ID | TEXT | 否 | 主键 |
| 2 | M\_COMMENT\_USERID | 评论者ID | TEXT | 否 | 外键 |
| 3 | M\_COMMENT\_IMAGEID | 图片ID | TEXT | 否 | 外键 |
| 4 | M\_COMMENT\_COMMETNTDATE | 上传时间 | DATETIME | 否 |  |
| 5 | M\_COMMENT\_CONTENT | 评论内容 | TEXT | 否 |  |

M\_LOBBYIMAGE表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | M\_LOBBYIMAGE\_RANK | 图片排位 | TEXT | 否 | 主键 |
| 2 | M\_LOBBYIMAGE\_IMAGEID | 图片ID | TEXT | 否 | 外键 |

M\_IMAGECARED表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 键 |
| 1 | M\_IMAGECARED\_ID | 关系ID | TEXT | 否 | 主键，自增ID |
| 2 | M\_IMAGECARED\_IMAGEID | 图片ID | TEXT | 否 | 外键 |
| 3 | M\_IMAGECARED\_UPDATEDATE | 图片上传日期 | TEXT | 否 |  |
| 4 | M\_IMAGECARED\_USERID | 上传者ID | TEXT | 否 | 外键 |

1. 功能模块设计

此项目中共涉及11个页面，分别是：

1. 登录页面：

参数：用户名，密码

调用背景：用户需要登录

页面组成：由用户名输入框，密码输入框和相应提示组成

调用描述：用户在需要登录的时候点击登录即可调出登录页面

1. 注册页面：

参数：用户名，密码，邮箱

调用背景：游客需要注册

页面组成：由用户名输入框，密码输入框，邮箱输入框和相应提示组成

调用描述：游客在登录页面点击注册

1. 主页（广场）:

参数：图片，用户名，

调用背景：用户希望查看广场

页面组成：由多个图片组成

调用描述：打开app默认打开主页（广场），在其他页面时点击返回或者点击广场调用该页面

1. 个人信息：

参数：用户名，邮箱，性别

调用背景：用户点击进入个人信息

页面组成：由用户的相关信息组成

调用描述：用户点击我的信息即可调用

1. 用户主页：

参数：图片，用户名

调用背景：用户希望查看用户的信息

页面组成：由该用户的用户名和许多张图片的缩略图组成

调用描述：用户点击希望查看的用户的用户名即可

1. 查看名单：

参数：用户名

调用背景：用户希望查看自己关注的人或自己拉入黑名单的人

页面组成：一个名单列表

调用描述：用户点击我关注的人或者黑名单

1. 动态：

参数：图片，用户名

调用背景：用户希望查看自己关注的人最近上传的图片

页面组成：由图片和用户名组成

调用描述：用户下滑主页面即可调用

1. 个人图片：

参数：图片，上传时间

调用背景：用户希望查看自己以往的图片

页面组成：图片和上传时间

调用描述：用户上划主页即可调用

1. 搜索：

参数：字符串

调用背景：用户希望搜索标签或者用户名

页面组成：由搜索框和搜索按钮组成

调用描述：用户点击搜索即可

1. 查看图片：

参数：图片，赞的数量，评论，评论数量

调用背景：用户希望查看图片的详细信息

页面组成：由图片，赞，评论及评论输入框组成

调用描述：用户点击图片

1. 查看大图：

参数：图片

调用背景：用户希望查看原图或者下载图片

页面组成：由图片及下载按钮组成

调用描述：用户在查看图片页面点击图片调用查看大图。

1. 数据库存储过程设计
   1. 涉及M\_USER表的存储过程
2. userInsert()

功能描述:将登陆的用户信息(用户名和密码)更新到数据库

入口参数:M\_USER\_ID, M\_USER\_PASSWORD

出口参数:无

伪语言实现:

CREATE PROCEDURE userInsert (@M\_USER\_ID char(32),@ M\_USER\_PASSWORD varchar(16))

AS

INSERT INTO M\_USER VALUES(@M\_USER\_ID,@ M\_USER\_PASSWORD);

RETURN;

1. userDelete()

功能描述:删除指定的用户

入口参数: M\_USER\_ID

出口参数:无

伪语言实现:

CREATE PROCEDURE userDelete (@M\_USER\_ID char(32))

AS

DELETE FROM M\_USER WHERE M\_USER\_ID=@ M\_USER\_ID;

RETURN;

1. userCheck()

功能描述:查找指定的用户在用户表中是否存在。

入口参数: M\_USER\_ID

出口参数:在用户表中是否存在

伪语言实现:

CREATE PROCEDURE userCheck (@M\_USER\_ID char(32))

AS

DECLARE left int;

SELECT count(\*) FROM M\_USER WHERE M\_USER\_ID=@ M\_USER\_ID;

RETURN;

1. userUpdatePassword()

功能描述:更新指定用户的Password

入口参数: M\_USER\_ID,NEW\_USER\_PASSWORD

出口参数:无

伪语言实现:

CREATE PROCEDURE userUpdatePassword (@M\_USER\_ID char(32),@ NEW\_USER\_PASSWORD varchar(16))

AS

UPDATE M\_USER SET M\_USER\_PASSWORD = @ NEW\_USER\_PASSWORD WHERE M\_USER\_ID =@ M\_USER\_ID;

RETURN;

* 1. 涉及M\_IMAGE表的存储过程

1. imageInsert()

功能描述:将新图片的信息更新到数据库

入口参数: M\_IMAGE\_IMAGEID,M\_IMAGE\_USERID ,M\_IMAGE\_LIKENUMBER,M\_IMAGE\_UPDATEDATE

出口参数:无

伪语言实现:

CREATE PROCEDURE imageInsert (@M\_IMAGE\_IMAGEID char(32),@M\_IMAGE\_USERID char(32),@ M\_IMAGE\_LIKENUMBER int,@ M\_IMAGE\_UPDATEDATE date)

AS

INSERT INTO M\_IMAGE VALUES(@M\_IMAGE\_IMAGEID,@M\_IMAGE\_USERID,@ M\_IMAGE\_LIKENUMBER,@ M\_IMAGE\_UPDATEDATE);

RETURN;

1. imageDelete()

功能描述:删除指定的图片

入口参数: M\_IMAGE\_IMAGEID

出口参数:无

伪语言实现:

CREATE PROCEDURE imageDelete (@M\_IMAGE\_IMAGEID char(32))

AS

DELETE FROM M\_IMAGE WHERE M\_IMAGE\_IMAGEID =@ M\_IMAGE\_IMAGEID;

RETURN;

1. userUpdateLikeNumber ()

功能描述:更新指定图片的赞数量

入口参数: M\_IMAGE\_IMAGEID,NEW\_ M\_IMAGE\_LIKENUMBER

出口参数:无

伪语言实现:

CREATE PROCEDURE userUpdateLikeNumber (@M\_IMAGE\_IMAGEID char(32),@NEW\_M\_IMAGE\_LIKENUMBER int)

AS

UPDATE M\_IMAGE SET M\_IMAGE\_LIKENUMBER = @NEW\_M\_IMAGE\_LIKENUMBER WHERE M\_IMAGE\_IMAGEID =@ M\_IMAGE\_IMAGEID;

RETURN;

1. userImageSelect ()

功能描述:根据指定用户查询图片集

入口参数: M\_IMAGE\_USERID

出口参数:指定用户的图片集

伪语言实现:

CREATE PROCEDURE userImageSelect (@M\_IMAGE\_USERID char(32))

AS

SELECT \* FROM M\_IMAGE WHERE M\_IMAGE\_USERID= @M\_IMAGE\_USERID;

RETURN;

* 1. 涉及m\_COMMENT表的存储过程

(1) commentInsert()

功能描述:将新评论的信息更新到数据库

入口参数: M\_COMMENT\_COMMENTID, M\_COMMENT\_USERID , M\_COMMENT\_IMAGEID, M\_COMMENT\_CONTENT, M\_COMMENT\_COMMETNTDATE

出口参数:无

伪语言实现:

CREATE PROCEDURE commentInsert (@M\_COMMENT\_COMMENTID char(32),@M\_COMMENT\_USERID char(32),@ M\_COMMENT\_IMAGEID char(32), M\_COMMENT\_CONTENT varchar(255),@ M\_COMMENT\_COMMETNTDATE DATETIME)

AS

INSERT INTO M\_COMMENT VALUES(@M\_COMMENT\_COMMENTID ,@M\_COMMENT\_USERID,@ M\_COMMENT\_IMAGEID, M\_COMMENT\_CONTENT ,@M\_COMMENT\_COMMETNTDATE);

RETURN;

(2) commentDelete()

功能描述:删除指定的评论

入口参数: M\_COMMENT\_COMMENTID

出口参数:无

伪语言实现:

CREATE PROCEDURE commentDelete (@M\_COMMENT\_COMMENTID char(32))

AS

DELETE FROM M\_COMMENT

WHERE M\_COMMENT\_COMMENTID=@M\_COMMENT\_COMMENTID;

RETURN;

(3) commentUpdateContent ()

功能描述:更新指定评论的内容

入口参数: M\_COMMENT\_COMMENTID,NEW\_ M\_COMMENT\_CONTENT

出口参数:无

伪语言实现:

CREATE PROCEDURE commentUpdateContent (@M\_COMMENT\_COMMENTID char(32),@NEW\_ M\_COMMENT\_CONTENT varchar(255))

AS

UPDATE M\_COMMENT SET M\_COMMENT\_CONTENT = @NEW\_M\_COMMENT\_CONTENT WHERE M\_COMMENT\_COMMENTID =@NEW\_ M\_COMMENT\_CONTENT ;

RETURN;

(4) imageCommentSelect ()

功能描述:根据指定图片查询评论集

入口参数: M\_COMMENT\_IMAGEID

出口参数:指定用户的评论集

伪语言实现:

CREATE PROCEDURE imageCommentSelect (@M\_COMMENT\_IMAGEID char(32))

AS

SELECT \* FROM M\_COMMENT WHERE M\_COMMENT\_IMAGEID =@M\_COMMENT\_IMAGEID;

RETURN;

* 1. 涉及M\_TAG表的存储过程

1. tagSelect()

功能描述:得到所有的标签集

入口参数:无

出口参数: 标签集

伪语言实现:

CREATE PROCEDURE tagSelect()

AS

SELECT \* FROM M\_TAG;

RETURN;

1. tagInsert()

功能描述:将新标签的信息更新到数据库

入口参数: M\_TAG\_ID,M\_TAG\_NAME, M\_TAG\_USERID,M\_TAG\_IMAGEID

出口参数:无

伪语言实现:

CREATE PROCEDURE tagInsert (@M\_TAG\_ID char(32),@ M\_TAG\_NAME varchar(255),@ M\_TAG\_USERID char(32),@M\_TAG\_IMAGEID TEXT)

AS

INSERT INTO M\_TAG VALUES ( @M\_TAG\_ID ,@M\_TAG\_NAME,@M\_TAG\_USERID,@M\_TAG\_IMAGEID)

RETURN;

* 1. tagDelete()

功能描述:删除指定的标签

入口参数: M\_TAG\_ID

出口参数:无

伪语言实现:

CREATE PROCEDURE tagDelete (@M\_TAG\_ID TEXT)

AS

DELETE FROM M\_TAG WHERE M\_TAG\_ID =@ M\_TAG\_ID;

RETURN;

1. tagUpdate ()

功能描述:更新指定标签的内容

入口参数: M\_TAG\_ID, NEW\_M\_TAG\_NAME

出口参数:无

伪语言实现:

CREATE PROCEDURE tagUpdate (@M\_TAG\_ID TEXT,@ NEW\_M\_TAG\_NAME TEXT)

AS

UPDATE M\_TAG SET M\_TAG\_NAME = @ NEW\_M\_TAG\_NAME WHERE M\_TAG\_ID =@ M\_TAG\_ID

RETURN;

1. tagImageSelect ()

功能描述:根据指定标签查询图片集

入口参数: M\_TAG\_NAME

出口参数:指定标签的图片集

伪语言实现:

CREATE PROCEDURE tagImageSelect (@M\_TAG\_NAME TEXT)

AS

SELECT \* FROM M\_TAG WHERE M\_TAG\_NAME= @M\_TAG\_NAME;

RETURN;

* 1. 涉及m\_BLACKLIST表的存储过程

1. blackListSelect()

功能描述：增加黑名单

入口参数：OWNER\_ID

出口参数：MEMBER\_ID集

伪语言实现：

CREATE PROCEDURE blackListSelect(@OWNER\_ID char(32))

AS

SELECT MEMBER\_ID FROM m\_BLACKLIST WHERE(OWNER\_ID = @OWNER\_ID)

RETURN;

1. blackListInsert()

功能描述：增加黑名单

入口参数：OWNER\_ID, MENBER\_ID

出口参数：无

伪语言实现：

CREATE PROCEDURE blackListInsert(@OWNER\_ID char(32), @MEMBER\_ID char(32))

AS

INSERT INTO m\_BLACKLIST VALUES(@OEWNER\_ID, MENBER\_ID)

RETURN;

1. blackListDelete()

入口参数：OWNER\_ID, MEMBER\_ID

出口参数：无

伪语言实现：

CREATE PROCEDURE blackListDelete(@OWNER\_ID char(32), @MEMBER\_ID char(32))

AS

DELETE FROM m\_BLACKLIST WHERE(OWNER\_ID = @OWNER\_ID AND MEMBER\_ID = @MEMBER\_ID)

RETURN;

* 1. 涉及m\_CONCERN表的存储过程

1. blackListSelect()

功能描述：增加黑名单

入口参数：OWNER\_ID

出口参数：MEMBER\_ID集

伪语言实现：

CREATE PROCEDURE blackListSelect(@OWNER\_ID char(32))

AS

SELECT MEMBER\_ID FROM m\_BLACKLIST WHERE(OWNER\_ID = @OWNER\_ID)

RETURN;

1. blackListInsert()

功能描述：增加黑名单

入口参数：OWNER\_ID, MENBER\_ID

出口参数：无

伪语言实现：

CREATE PROCEDURE blackListInsert(@OWNER\_ID char(32), @MEMBER\_ID char(32))

AS

INSERT INTO m\_BLACKLIST VALUES(@OEWNER\_ID,@ MENBER\_ID)

RETURN;

1. blackListDelete()

功能描述：删除黑名单

入口参数：OWNER\_ID, MEMBER\_ID

出口参数：无

伪语言实现：

CREATE PROCEDURE blackListDelete(@OWNER\_ID char(32), @MEMBER\_ID char(32))

AS

DELETE FROM m\_BLACKLIST WHERE(OWNER\_ID = @OWNER\_ID AND MEMBER\_ID = @MEMBER\_ID)

RETURN;

6.7涉及m\_LIKE表的存储过程

(1) like\_Check()

功能描述：是否赞过此图片

入口参数：U\_ID, I\_ID

出口参数：BOOL

伪语言实现：

CREATE PROCEDURE like\_Check(@U\_ID char(32), @I\_ID char(32))

AS

IF EXISTS (SELECT \* FROM m\_LIKE WHERE (@U\_ID = U\_ID AND @I\_ID = I\_ID))

RETURN TRUE

ELSE

RETURN FALSE;

1. likeListInsert()

功能描述：增加赞表

入口参数：I\_ID, U\_ID

出口参数：无

伪语言实现：

CREATE PROCEDURE likeInsert(@I\_ID char(32), @U\_ID char(32))

AS

INSERT INTO m\_LIKE VALUES(@I\_ID,@ U\_ID)

RETURN;

1. likeDelete()

功能描述：删除黑名单

入口参数：I\_ID, U\_ID

出口参数：无

伪语言实现：

CREATE PROCEDURE likeDelete(@I\_ID char(32), @U\_ID char(32))

AS

DELETE FROM m\_LIKE WHERE(I\_ID = @I\_ID AND U\_ID = @U\_ID)

RETURN;

1. 手机端本地数据库存储过程设计
   1. 涉及M\_USER表的存储过程
2. userSelect()

功能描述:得到所有的用户集

入口参数:无

出口参数:用户集

伪语言实现:

CREATE PROCEDURE userSelect()

AS

SELECT \* FROM M\_USER;

RETURN;

1. userInsert()

功能描述:将登陆的用户信息(只有用户名)更新到数据库

入口参数:M\_USER\_ID

出口参数:无

伪语言实现:

CREATE PROCEDURE userInsert (@M\_USER\_ID TEXT)

AS

INSERT INTO M\_USER VALUES(@M\_USER\_ID);

RETURN;

1. userInsert()

功能描述:将登陆的用户信息(用户名和密码)更新到数据库

入口参数:M\_USER\_ID, M\_USER\_PASSWORD

出口参数:无

伪语言实现:

CREATE PROCEDURE userInsert (@M\_USER\_ID TEXT,@ M\_USER\_PASSWORD TEXT)

AS

INSERT INTO M\_USER VALUES(@M\_USER\_ID,@ M\_USER\_PASSWORD);

RETURN;

1. userDelete()

功能描述:删除指定的用户

入口参数: M\_USER\_ID

出口参数:无

伪语言实现:

CREATE PROCEDURE userDelete (@M\_USER\_ID TEXT)

AS

DELETE FROM M\_USER WHERE M\_USER\_ID=@ M\_USER\_ID;

RETURN;

1. userCheck()

功能描述:查找指定的用户在用户表中是否存在。

入口参数: M\_USER\_ID

出口参数:在用户表中是否存在

伪语言实现:

CREATE PROCEDURE userCheck (@M\_USER\_ID TEXT)

AS

DECLARE left int;

SELECT count(\*) FROM M\_USER WHERE M\_USER\_ID=@ M\_USER\_ID;

RETURN;

1. userUpdatePassword()

功能描述:更新指定用户的Password

入口参数: M\_USER\_ID,NEW\_USER\_PASSWORD

出口参数:无

伪语言实现:

CREATE PROCEDURE userUpdatePassword (@M\_USER\_ID TEXT,@ NEW\_USER\_PASSWORD TEXT)

AS

UPDATE M\_USER SET M\_USER\_PASSWORD = @ NEW\_USER\_PASSWORD WHERE M\_USER\_ID =@ M\_USER\_ID;

RETURN;

* 1. 涉及M\_LASTUSER表的存储过程

1. lastUserSelect()

功能描述:得到最后登录的用户集

入口参数:无

出口参数: 最后登录的用户集

伪语言实现:

CREATE PROCEDURE lastUserSelect ()

AS

SELECT \* FROM M\_LASTUSER;

RETURN;

1. lastUserInsert()

功能描述:将最后登录的登陆的用户信息(用户名和密码)更新到数据库

入口参数:M\_USER\_ID, M\_USER\_PASSWORD

出口参数:无

伪语言实现:

CREATE PROCEDURE lastUserInsert (@M\_LASTUSER\_ID TEXT,@ M\_LASTUSER\_PASSWORD TEXT)

AS

INSERT INTO M\_LASTUSER VALUES(@M\_LASTUSER\_ID,@ M\_LASTUSER\_PASSWORD);

RETURN;

1. lastUserDelete()

功能描述:删除最后登录的用户

入口参数:无

出口参数:无

伪语言实现:

CREATE PROCEDURE lastUserDelete ()

AS

DELETE FROM M\_LASTUSER

WHERE M\_LASTUSER\_ID =(SELECT M\_LASTUSER\_ID FROM M\_LASTUSER);

RETURN;

1. lastUserUpdateid ()

功能描述:更新最后登录的用户的信息

入口参数: NEW\_M\_LASTUSER\_ID,NEW\_M\_LASTUSER\_PASSWORD

出口参数:无

伪语言实现:

CREATE PROCEDURE lastUserUpdateid (@NEW\_M\_LASTUSER\_ID TEXT,@ NEW\_M\_LASTUSER\_PASSWORD TEXT)

AS

UPDATE M\_LASTUSER SET M\_LASTUSER\_ID = @NEW\_M\_LASTUSER\_ID, M\_LASTUSER\_PASSWORD=@NEW\_M\_LASTUSER\_PASSWORD

WHERE M\_LASTUSER\_ID =(SELECT M\_LASTUSER\_ID FROM M\_LASTUSER);

RETURN;

1. lastUserUpdatePassword ()

功能描述:更新最后登录的用户的密码

入口参数: NEW\_M\_LASTUSER\_PASSWORD

出口参数:无

伪语言实现:

CREATE PROCEDURE lastUserUpdatePassword (@ NEW\_M\_LASTUSER\_PASSWORD TEXT)

AS

UPDATE M\_LASTUSER SET M\_LASTUSER\_PASSWORD=@NEW\_M\_LASTUSER\_PASSWORD

WHERE M\_LASTUSER\_ID =(SELECT M\_LASTUSER\_ID FROM M\_LASTUSER);

RETURN;

* 1. 涉及M\_IMAGE表的存储过程

1. imageSelect()

功能描述:得到所有的图片集

入口参数:无

出口参数: 图片集

伪语言实现:

CREATE PROCEDURE imageSelect ()

AS

SELECT \* FROM M\_IMAGE;

RETURN;

1. imageInsert()

功能描述:将新图片（大图）的信息更新到数据库

入口参数: M\_IMAGE\_IMAGEID,M\_IMAGE\_USERID , M\_IMAGE\_ISLIKE,M\_IMAGE\_LIKENUMBER,M\_IMAGE\_UPDATEDATE

出口参数:无

伪语言实现:

CREATE PROCEDURE imageInsert (@M\_IMAGE\_IMAGEID TEXT,@M\_IMAGE\_USERID TEXT,@ M\_IMAGE\_ISLIKE TEXT,@ M\_IMAGE\_LIKENUMBER TEXT,@ M\_IMAGE\_UPDATEDATE DATETIME)

AS

INSERT INTO M\_IMAGE VALUES(@M\_IMAGE\_IMAGEID,@M\_IMAGE\_USERID, @ M\_IMAGE\_ISLIKE ,@ M\_IMAGE\_LIKENUMBER,@ M\_IMAGE\_UPDATEDATE);

RETURN;

1. imageInsert()

功能描述:将新图片（缩略图）的信息更新到数据库

入口参数: M\_IMAGE\_IMAGEID

出口参数:无

伪语言实现:

CREATE PROCEDURE imageInsert (@M\_IMAGE\_IMAGEID TEXT)

AS

INSERT INTO M\_IMAGE VALUES(@M\_IMAGE\_IMAGEID);

RETURN;

1. imageDelete()

功能描述:删除指定的图片

入口参数: M\_IMAGE\_IMAGEID

出口参数:无

伪语言实现:

CREATE PROCEDURE imageDelete (@M\_IMAGE\_IMAGEID TEXT)

AS

DELETE FROM M\_IMAGE WHERE M\_IMAGE\_IMAGEID =@ M\_IMAGE\_IMAGEID;

RETURN;

1. userUpdateLikeNumber ()

功能描述:更新指定图片的赞数量

入口参数: M\_IMAGE\_IMAGEID,NEW\_ M\_IMAGE\_LIKENUMBER

出口参数:无

伪语言实现:

CREATE PROCEDURE userUpdateLikeNumber (@M\_IMAGE\_IMAGEID TEXT,@NEW\_M\_IMAGE\_LIKENUMBER TEXT)

AS

UPDATE M\_IMAGE SET M\_IMAGE\_LIKENUMBER = @NEW\_M\_IMAGE\_LIKENUMBER WHERE M\_IMAGE\_IMAGEID =@ M\_IMAGE\_IMAGEID;

RETURN;

1. userUpdateIsLike ()

功能描述:更新指定图片的是否赞过

入口参数: M\_IMAGE\_IMAGEID,NEW\_ M\_IMAGE\_ISLIKE

出口参数:无

伪语言实现:

CREATE PROCEDURE userUpdateLikeNumber (@M\_IMAGE\_IMAGEID TEXT,@ NEW\_ M\_IMAGE\_ISLIKE TEXT)

AS

UPDATE M\_IMAGE SET M\_IMAGE\_ISLIKE= @ NEW\_ M\_IMAGE\_ISLIKE WHERE M\_IMAGE\_IMAGEID =@ M\_IMAGE\_IMAGEID;

RETURN;

1. userImageSelect ()

功能描述:根据指定用户查询图片集

入口参数: M\_IMAGE\_USERID

出口参数:指定用户的图片集

伪语言实现:

CREATE PROCEDURE userImageSelect (@M\_IMAGE\_USERID TEXT)

AS

SELECT \* FROM M\_IMAGE WHERE M\_IMAGE\_USERID= @M\_IMAGE\_USERID;

RETURN;

* 1. 涉及M\_COMMENT表的存储过程

1. commentSelect()

功能描述:得到所有的评论集

入口参数:无

出口参数: 评论集

伪语言实现:

CREATE PROCEDURE commentSelect ()

AS

SELECT \* FROM M\_COMMENT;

RETURN;

1. commentInsert()

功能描述:将新评论的信息更新到数据库

入口参数: M\_COMMENT\_COMMENTID, M\_COMMENT\_USERID , M\_COMMENT\_IMAGEID, M\_COMMENT\_CONTENT, M\_COMMENT\_COMMETNTDATE

出口参数:无

伪语言实现:

CREATE PROCEDURE commentInsert (@M\_COMMENT\_COMMENTID TEXT,@M\_COMMENT\_USERID TEXT,@ M\_COMMENT\_IMAGEID TEXT, M\_COMMENT\_CONTENT TEXT,@ M\_COMMENT\_COMMETNTDATE DATETIME)

AS

INSERT INTO M\_COMMENT VALUES(@M\_COMMENT\_COMMENTID ,@M\_COMMENT\_USERID,@ M\_COMMENT\_IMAGEID, M\_COMMENT\_CONTENT ,@M\_COMMENT\_COMMETNTDATE);

RETURN;

1. commentDelete()

功能描述:删除指定的评论

入口参数: M\_COMMENT\_COMMENTID

出口参数:无

伪语言实现:

CREATE PROCEDURE commentDelete (@M\_COMMENT\_COMMENTID TEXT)

AS

DELETE FROM M\_COMMENT

WHERE M\_COMMENT\_COMMENTID=@M\_COMMENT\_COMMENTID;

RETURN;

1. commentUpdateContent ()

功能描述:更新指定评论的内容

入口参数: M\_COMMENT\_COMMENTID,NEW\_ M\_COMMENT\_CONTENT

出口参数:无

伪语言实现:

CREATE PROCEDURE commentUpdateContent (@M\_COMMENT\_COMMENTID TEXT,@NEW\_ M\_COMMENT\_CONTENT TEXT)

AS

UPDATE M\_COMMENT SET M\_COMMENT\_CONTENT = @NEW\_M\_COMMENT\_CONTENT WHERE M\_COMMENT\_COMMENTID =@NEW\_ M\_COMMENT\_CONTENT ;

RETURN;

1. imageCommentSelect ()

功能描述:根据指定图片查询评论集

入口参数: M\_COMMENT\_IMAGEID

出口参数:指定用户的评论集

伪语言实现:

CREATE PROCEDURE imageCommentSelect (@M\_COMMENT\_IMAGEID TEXT)

AS

SELECT \* FROM M\_COMMENT WHERE M\_COMMENT\_IMAGEID =@M\_COMMENT\_IMAGEID;

RETURN;

* 1. 涉及M\_TAG表的存储过程

1. tagSelect()

功能描述:得到所有的标签集

入口参数:无

出口参数: 标签集

伪语言实现:

CREATE PROCEDURE tagSelect()

AS

SELECT \* FROM M\_TAG;

RETURN;

1. tagInsert()

功能描述:将新标签的信息更新到数据库

入口参数: M\_TAG\_ID,M\_TAG\_NAME,M\_TAG\_IMAGEID

出口参数:无

伪语言实现:

CREATE PROCEDURE tagInsert (@M\_TAG\_ID TEXT,@ M\_TAG\_NAME TEXT, @M\_TAG\_IMAGEID TEXT)

AS

INSERT INTO M\_TAG VALUES ( @M\_TAG\_ID ,@M\_TAG\_NAME ,@M\_TAG\_IMAGEID)

RETURN;

1. tagDelete()

功能描述:删除指定的标签

入口参数: M\_TAG\_ID

出口参数:无

伪语言实现:

CREATE PROCEDURE tagDelete (@M\_TAG\_ID TEXT)

AS

DELETE FROM M\_TAG WHERE M\_TAG\_ID =@ M\_TAG\_ID;

RETURN;

1. tagUpdate ()

功能描述:更新指定标签的内容

入口参数: M\_TAG\_ID, NEW\_M\_TAG\_NAME

出口参数:无

伪语言实现:

CREATE PROCEDURE tagUpdate (@M\_TAG\_ID TEXT,@ NEW\_M\_TAG\_NAME TEXT)

AS

UPDATE M\_TAG SET M\_TAG\_NAME = @ NEW\_M\_TAG\_NAME WHERE M\_TAG\_ID =@ M\_TAG\_ID

RETURN;

1. tagImageSelect ()

功能描述:根据指定标签查询图片集

入口参数: M\_TAG\_NAME

出口参数:指定标签的图片集

伪语言实现:

CREATE PROCEDURE tagImageSelect (@M\_TAG\_NAME TEXT)

AS

SELECT \* FROM M\_TAG WHERE M\_TAG\_NAME= @M\_TAG\_NAME;

RETURN;

* 1. 涉及M\_LOBBYIMAGE表的存储过程

1. lobbyImageSelect()

功能描述:得到大厅的图片集

入口参数:无

出口参数: 最后大厅的图片集

伪语言实现:

CREATE PROCEDURE lobbyImageSelect ()

AS

SELECT \* FROM M\_LOBBYIMAGE;

RETURN;

1. lobbyImageInsert()

功能描述:将大厅的图片信息更新到数据库

入口参数:M\_LOBBYIMAGE\_IMAGEID, M\_ LOBBYIMAGE\_RANK

出口参数:无

伪语言实现:

CREATE PROCEDURE lobbyImageInsert (@M\_LOBBYIMAGE\_IMAGEID TEXT,@ M\_ LOBBYIMAGE\_RANK TEXT)

AS

INSERT INTO M\_ LOBBYIMAGE VALUES(@M\_LOBBYIMAGE\_IMAGEID,@ M\_ LOBBYIMAGE\_RANK);

RETURN;

1. lobbyImageDelete()

功能描述:删除指定的大厅的图片

入口参数: M\_LOBBYIMAGE\_IMAGEID

出口参数:无

伪语言实现:

CREATE PROCEDURE lobbyImageDelete (@M\_LOBBYIMAGE\_IMAGEID TEXT)

AS

DELETE FROM M\_ LOBBYIMAGE WHERE M\_LOBBYIMAGE\_IMAGEID =@M\_LOBBYIMAGE\_IMAGEID;

RETURN;

1. lobbyImageUpdate ()

功能描述:更新指定的大厅的图片的信息

入口参数: M\_LOBBYIMAGE\_RANK, NEW\_M\_LOBBYIMAGE\_IMAGEID

出口参数:无

伪语言实现:

CREATE PROCEDURE lobbyImageUpdate (@M\_LOBBYIMAGE\_RANK TEXT, @NEW\_M\_LOBBYIMAGE\_IMAGEID TEXT)

AS

UPDATE M\_LOBBYIMAGE SET M\_LOBBYIMAGE\_IMAGEID = @NEW\_M\_LOBBYIMAGE\_IMAGEID WHERE M\_LOBBYIMAGE\_RANK =@ M\_LOBBYIMAGE\_RANK;

RETURN;

* 1. 涉及M\_IMAGECARED表的存储过程

1. imageCaredInsert()

功能描述:将新关注关系更新到数据库

入口参数:M\_IMAGECARED\_IMAGEID, M\_ IMAGECARED \_USERID,M\_IMAGECARED\_UPDATEDATE

出口参数:无

伪语言实现:

CREATE PROCEDURE imageCaredInsert (@M\_IMAGECARED\_IMAGEID TEXT,@ M\_ IMAGECARED \_USERID TEXT,@ M\_IMAGECARED\_UPDATEDATE DATETIME)

AS

INSERT INTO M\_ IMAGECARED VALUES(@M\_IMAGECARED\_IMAGEID,@ M\_ IMAGECARED \_ USERID, @ M\_IMAGECARED\_UPDATEDATE);

RETURN;

1. imageCaredDelete()

功能描述:删除指定的大厅的图片

入口参数: M\_IMAGECARED\_IMAGEID, M\_ IMAGECARED \_USERID

出口参数:无

伪语言实现:

CREATE PROCEDURE imageCaredDelete (@M\_IMAGECARED\_IMAGEID TEXT,@ M\_ IMAGECARED\_USERID TEXT)

AS

DELETE FROM M\_ IMAGECARED WHERE M\_ IMAGECARED\_IMAGEID =@M\_ IMAGECARED\_IMAGEID AND M\_IMAGECARED\_USERID=@M\_IMAGECARED\_USERID;

RETURN;

1. userImageCaredSelect ()

功能描述:根据指定用户查询该用户关注人的图片

入口参数: M\_ IMAGECARED \_USERID

出口参数:指定用户的图片集

伪语言实现:

CREATE PROCEDURE userImageCaredSelect (@M\_ IMAGECARED \_USERID TEXT)

AS

SELECT \* FROM M\_ IMAGECARED WHERE M\_ IMAGECARED \_USERID = @ M\_IMAGECARED\_USERID;

RETURN;

1. 接口设计
   1. 用户接口

基于Android平台，提供清晰简洁易用美观的用户界面。

* 1. 外部接口

数据存储：数据库通过Django与MySQL交互

手机端本地数据库由Android和Java语言与SQLite进行交互。

* 1. 内部接口

本系统各模块均通过Django与数据库服务器进行交互，涉及本地缓存时通过Android和Java语言与手机端本地数据库进行交互。

页面跳转时，通过Intend传递图片，字符串等参数。

1. 角色授权设计

角色授权表如表所示

|  |  |  |
| --- | --- | --- |
| 模块 | 游客 | 注册用户 |
| 登陆注册 | **·** | **·** |
| 图片管理 |  | **·** |
| 图片浏览 | **·**（部分功能） | **·** |
| 图片评价 |  | **·** |
| 社交功能 |  | **·** |

注：本软件管理员即为数据库管理员，拥有所有权限。

1. 系统错误设计
   1. 出错信息
2. 对没有权限的用户（游客）进行查看限制以防止错误产生。
3. 对用户的输入进行检查，如果输入错误，系统进行提示并中断操作，防止不合法输入导致的错误或异常发生。
4. 对执行过程中可能发生异常的代码进行异常处理。
5. 运行初期检查设备状态（摄像机是否能正常工作，网络是否通畅等），如果设备不能正常运行软件则进行相应的友好提示并中断软件。
   1. 故障预防与补救

以统一的机制对用户权限进行管理，对程序中用到的数据尽量进行加密，定期对数据库中的数据进行海量备份和增量备份(手机本地数据库由用户选择是否备份)。

* 1. 系统维护设计

1. 编码实现时应采用模块化和分层的思想，提高模块内部的内聚，减少模块间的耦合，使系统逻辑结构清晰，从而增强可读性和可维护性。
2. 面向数据与面向对象的结合，模块划分符合面向对象思想。
3. 在编码过程中注意标识符命名的意义，添加适量注释。
4. 测试计划

测试计划作为本系统的测试指导，测试人员按计划和规定进行测试。本系统的测试将从模块测试、子系统测试、集成测试和验收测试四个方面来设计