# Dipper-图片展示系统需求分析及设计流程

[Dipper-图片展示系统需求分析及设计流程 1](#_Toc8804)

[一、 引言 2](#_Toc21786)

[1.1程序目的 2](#_Toc16669)

[1.2背景 2](#_Toc16915)

[1.3参考资料 2](#_Toc15331)

[二、 系统结构 2](#_Toc13644)

[三、 客户端模块设计说明 3](#_Toc15799)

[3.1 程序描述 3](#_Toc2801)

[3.2 功能 3](#_Toc25213)

[3.3 流程逻辑 3](#_Toc1565)

[四、 服务器模块设计说明 3](#_Toc9378)

[4.1 程序描述 3](#_Toc21594)

[4.2 功能 3](#_Toc27918)

[五、 设计流程 4](#_Toc3661)

[5.1 页面设计 4](#_Toc32683)

[5.2 DB设计 4](#_Toc9903)

[5.3 dao设计 5](#_Toc4349)

[5.4 service设计 5](#_Toc499)

[5.5 action设计 5](#_Toc30114)

[5.6 表单设计 5](#_Toc1107)

## 引言

### 1.1程序目的

1. 为java学习练习做基础
2. 展示摄影图片以及个人平面设计作品
3. 作为Dipper系统的子系统，进行先期设计

### 1.2背景

1. 作者：倪春军
2. 项目名称：Dipper-摄影展示系统

### 1.3参考资料

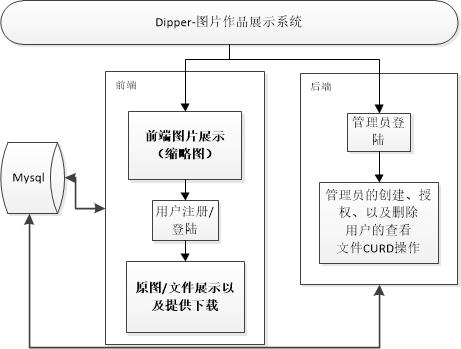
### 1.4 环境、命名规则

架构环境：

命名规则：

1. 页面命名：前台页面以 作用\_子系统.JSP命名，例如：”login.jsp”为登陆页面，但多个子系统间页面相同，所以仅为login.jsp即可，而不是login\_img.jsp。但图片页面为”type\_img.jsp”，img为图片子系统；后台页面以”a\*.jsp”命名，若首字母有a，则可以省略一个a；
2. 数据库命名：数据库命名为”dipper\_system”，管理员模块为”admin\_\*”，用户模块为”user\_\*”，图片模块为”img\_\*”，平面设计模块为”ps\_\*”，系统设置模块为”system\_”；

## 系统结构

图2.1 图片子系统结构图

## 客户端模块设计说明

### 3.1 程序描述

在前台展示图片：摄影作品和平面设计作品。用户在登陆状态下可以下载源文件。

### 3.2 功能

1. 用户能够进行登陆和注册；
2. 在前台展示两大分类的图片：摄影作品和平面设计作品；
3. 能够根据不同分类查看图片；
4. 能够根据图片名搜索指定的图片进行查看。
5. 在点击进入详情页后，能够看到更加清晰的图片；
6. 在用户在登陆状态下，可以免费下载图片或摄影作品的源文件。

### 3.3 流程逻辑

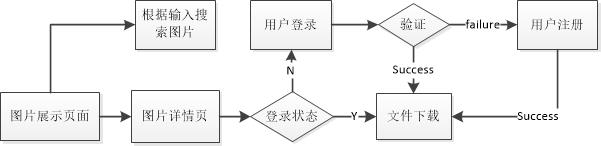


图3.3.1 客户端逻辑判断

## 服务器模块设计说明

### 4.1 程序描述

主要为用户、管理员的管理，图片的上传、下载以及图片的编辑。对应后端界面

### 4.2 功能

1. 用户的查看以及管理；
2. 新管理员的创建以及授权；
3. 文件的上传以及下载；
4. 图片属性的编辑：名称、类别、时间、地点
5. 图片的上传大小限制
6. 图片的上传、下载记录
7. 网站LOGO的更改

## 设计流程

### 5.1 页面设计

前端页面：

Index\_img.jsp 图片展示主页面

type\_img.jsp 图片分类页面

detail\_img.jsp 图片详情页

login.jsp 登陆页面

login\_error.jsp 登陆错误页面

register.jsp 注册页面

error.jsp 错误页面

Footer\_img.jsp 页脚页面

header\_img.jsp 页眉页面/导航栏

后端页面：

alogin.jsp 管理员登陆界面

aheader.jsp 页眉界面/导航

afooter.jsp 页脚界面

awest.jsp 菜单界面

asetting.jsp 系统设置界面

admin\_manage.jsp 管理员管理界面

auser\_manage.jsp 用户管理界面

admin\_register.jsp 管理员注册界面

aImg\_fileup.jsp 文件上传

aimg\_filedown.jsp 文件下载

aimg\_manage.jsp 文件管理

aimg\_detail.jsp 文件详情设置

aimg\_resule.jsp 文件上传下载统计

### 5.2 DB设计

1. 管理员模块

包括管理员信息模块

登陆验证模块

登陆状态过滤器，

权限管理模块

权限检查过滤器

1. 用户模块
2. 图片模块

Img-

Img\_type-

Img\_fileup-

Img\_filedown-

1. 设计文件模块
2. 系统模块

### 5.3 dao设计

方法：

//用户验证

public Admin\_im selectByNamePassword(Map<String,Object> map);

//查找所有管理员账户

public List<Admin\_im> selectAllAdmin();

//查找所有管理员级别

public List<Admin\_action> selectAllAction();

//插入管理员账户

public void insert(Admin\_im admin\_im);

//更新管理员账户

public void update(Admin\_im admin\_im);

//删除管理员账户

public void delete(int id);

//查询所有权限与管理员级别 对应表

//用以得到所有权限uri

public List<Authority\_url> selectAllAuthority\_url();

//根据查询得到的action的id查找此id的所有对应表中的数据

public List<Authority\_admin> selectAuthority\_adminByAdmin\_action\_id(int admin\_action\_id);

//根据查询到的对应表中的id查找对应的url

public Authority\_url selectAuthority\_url(int authority\_admin\_id);

/\*\*

\* 图片的相关方法 IMG

\*/

//上传图片

public void insertImg(Img img) throws Exception;

//在上传图片的同时增加图片的 上传记录

public void insertFileup(Img\_fileup img\_fileup) throws Exception;

//在上下载图片的同时增加图片的 下载记录

public void insertFiledown(Img\_filedown img\_filedown) throws Exception;

//更改图片属性

public void updateImg(Img img) throws Exception;

/\*\*

\* 查询IMG的方法

\* @return IMG对象

\* @throws Exception

\*/

//查询图片

// public List<Img> selectAllImg() throws Exception;

//根据 上传时间 排序查询

// public List<Img> selectAllImgGroupbyDate() throws Exception;

//根据 下载量 排序查询

// public List<Img> selectAllImgGroupbyFiledown() throws Exception;

//查询 下载量最多 的图片(应该先查询最大值，然后在根据最大值查询，防止并列)

public List<Img> selectImgByMaxFiledown() throws Exception;

//根据 图片标题 查询IMG (可能有多个图片标题是相同的,,,而且此处要使用模糊查询)

public List<Img> selectImgByName(String img\_name) throws Exception;

//根据 分类 查询图片

// public List<Img> selectImgByType(int img\_type\_id) throws Exception;

//根据 拍摄时间(day) 查询Img

public List<Img> selectImgByDate(Date date) throws Exception;

//根据ID查询IMG

public Img selectImgById(String img\_id) throws Exception;

//分页查询 1.所有 ；2.根据时间排序； 3.根据类型;4.下载量 (动态SQL，所以可以保留两个方法即可)

public List<Img> selectAllImgFY( Map<String,Integer> map ) throws Exception;

public List<Img> selectAllImgGroupbyDateFY(Date date,int start,int size) throws Exception;

public List<Img> selectImgByTypeFY(int img\_type\_id,int start,int size) throws Exception;

public List<Img> selectAllImgGroupbyFiledownFY(int start,int size) throws Exception;

/\*\*

\* 查询TYPE图片分类

\* @return 图片分类

\* @throws Exception

\*/

//查询所有分类

public List<Img\_type> selectAllType(int start,int size) throws Exception;

//查询 某一级别 的分类(图片、设计、网页，图片下的分类)

public List<Img\_type> selectAllFirstType(int img\_type\_level,int start,int size) throws Exception;

//根据分类名字查询

public Img\_type selectTypeByName(String img\_type\_name,int start,int size) throws Exception;

//根据id名字查询分类

public Img\_type selectTypeById(int img\_type\_id,int start,int size) throws Exception;

/\*\*

\* 查询上传记录

\*/

//查询所有的上传记录

// public List<Img\_fileup> selectAllFileup() throws Exception;

//分页查询

public List<Img\_fileup> selectAllFileupFY(int start,int size) throws Exception;

/\*\*

\* 查询下载记录

\*/

//查询所有的下载记录

// public List<Img\_filedown> selectAllFiledown() throws Exception;

//时间分页

public List<Img\_filedown> selcetFiledownByDateFY(Date date) throws Exception;

//分页查询

public List<Img\_filedown> selectAllFiledownFY(int start,int size) throws Exception;

/\*\*

\* 删除操作

\*/

//删除图片

public void deleteImg(int img\_id) throws Exception;

//清空上传记录（暂无需求，后续上线）

//清空下载记录（暂无需求，后续上线）

### 5.4 service设计

### 5.5 action设计

### 5.6 表单设计