

数字媒体资源管理

系统说明文档

Date : 2017-11-28

成员 1 315010**** *****

成员 2 315010**** *****

成员 3 315010**** *****

成员 4 314010**** *****

一、 系统介绍与设计

1. 系统介绍

BERRYFOLIO 是我们小组自主设计的一款线上作品集管理系统 Demo，针对数字媒体以及相关专业的同学，帮助他们更好地管理自己的作品。网站采用了友好的交互和符合同学们喜欢的风格，希望能够让大家在紧张的学习之余，不再为数据管理烦心。

系统为同学们提供一个存储作品、管理作品、下载作品和交流作品的平台，建立服务器为大家保管作品并做好加密工作，为大家尽量避免物理存储带来的意外事故。

2. 系统设计

A. 系统特色

a. 目录创建机制

我们设置了两种目录来方便文件存储。一种是父目录，意味着用户可以在该目录下增加新的子目录；一种是子目录，意味着用户可以在目录下上传文件。这样考虑的原因是，一般我们在硬盘里存储自己作品的目录都比较复杂，可能在目录下还有文件，文件的同级还存在目录，这对于作品集的生成是十分不利的（因为同学当然会想在制作作品集时有一个目录清晰的压缩文件），所以我们思考出了这种目录设置。并且和后端进行了联系。

b. 目录下载机制

我们采用选择下载目录的方式给用户看似不灵活但其实更加便捷的下载方式，只要选择了某一个目录（即使是一整个作品集），我们的系统就会将整个目录进行打包和压缩，由用户选择保存路径。

c. 登录和验证码机制

登录能够提供“记住当前用户”的服务。系统的验证码除了自动生成外，还会自动显示在拟人形象上，给系统增添了娱乐性，维持了风格的统一，验证码是区分大小写的。

d. 用户个性化设置

用户能够设置自己的头像、介绍和作品集的名称以及描述信息。登录能够提供“记住当前用户”的服务。系统的验证码除了自动生成外，还会自动显示在拟人形象上，给系统增添了娱乐性，维持了风格的统一，验证码是区分大小写的。

e. 作品浏览和目录索引

整体采用简洁的表格式设计，并添加了一些细节性的小动画。比如首页的图片自动播放和作品集主页的相册式动态展示。

B. 功能设计

系统包含下列核心功能：

a. 上传作品

用户在本地选取自己需要上传的文件，点击上传后就可以在个人作品集中找到相应的文件预览。

b. 浏览作品

用户能后在作品集页面看到自己上传的文件以及文件名称，并且能够在个人主页看到展示性的页面。

c. 打包下载

当用户选择想要下载的目录时，系统会自动形成压缩文件提供下载。

d. 数据入库

后端有完备的数据库用以存储用户信息和上传的文件信息。

e. 数据搜索

用户根据文件标签可以很方便地搜索到自己想要的文件。

除此之外，还有一些对应的扩展功能：

f. 用户登录注册系统

无需邮箱验证，但是用户的用户名会进行重名检测，一旦发现已经被注册，该用户就无法使用这个用户名，必须另外换一个。

g. 为登录系统添加验证码保护

验证码为随机生成的四位字母，显示在登录和注册界面的合适位置。

h. 为作品添加标题、描述性文字和标签

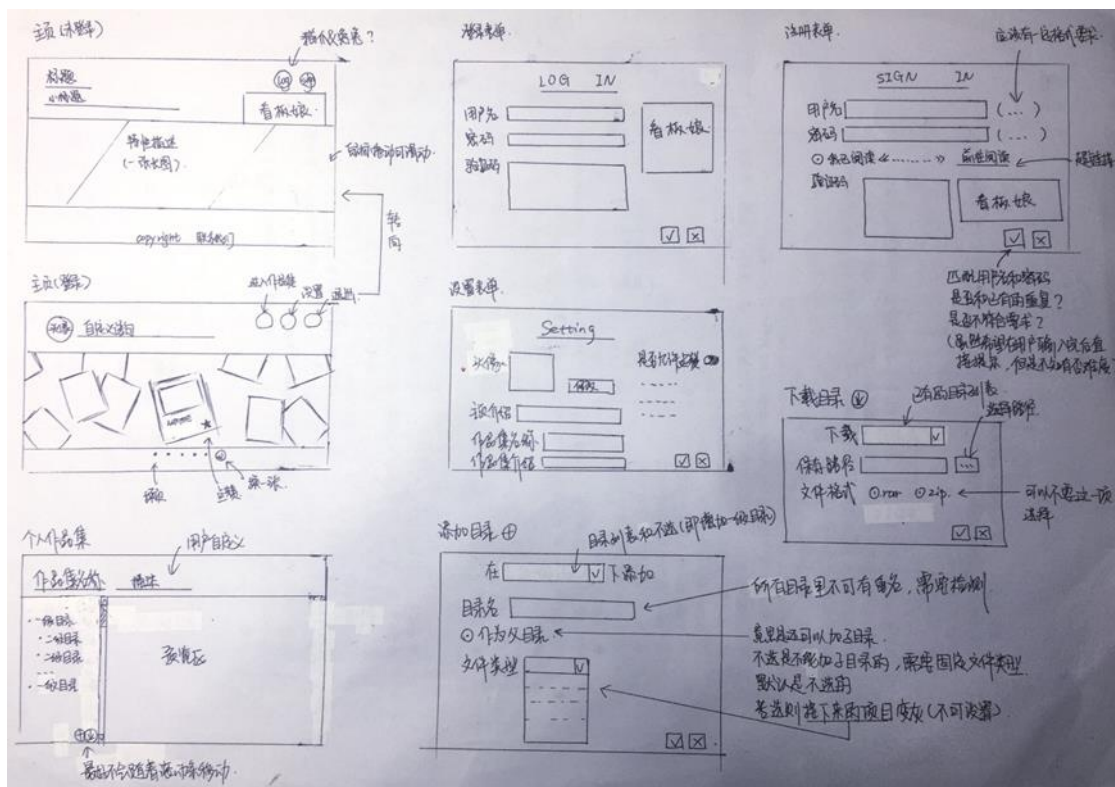
用户可以为作品添加 10 字以内的标题，25 字以内的描述以及上限 3 个标题。

i. 设计系统的拟人形象

可爱而有趣的 DAM 拟人形象，用于注册和登录界面。

C. 界面设计

a. 配色方案



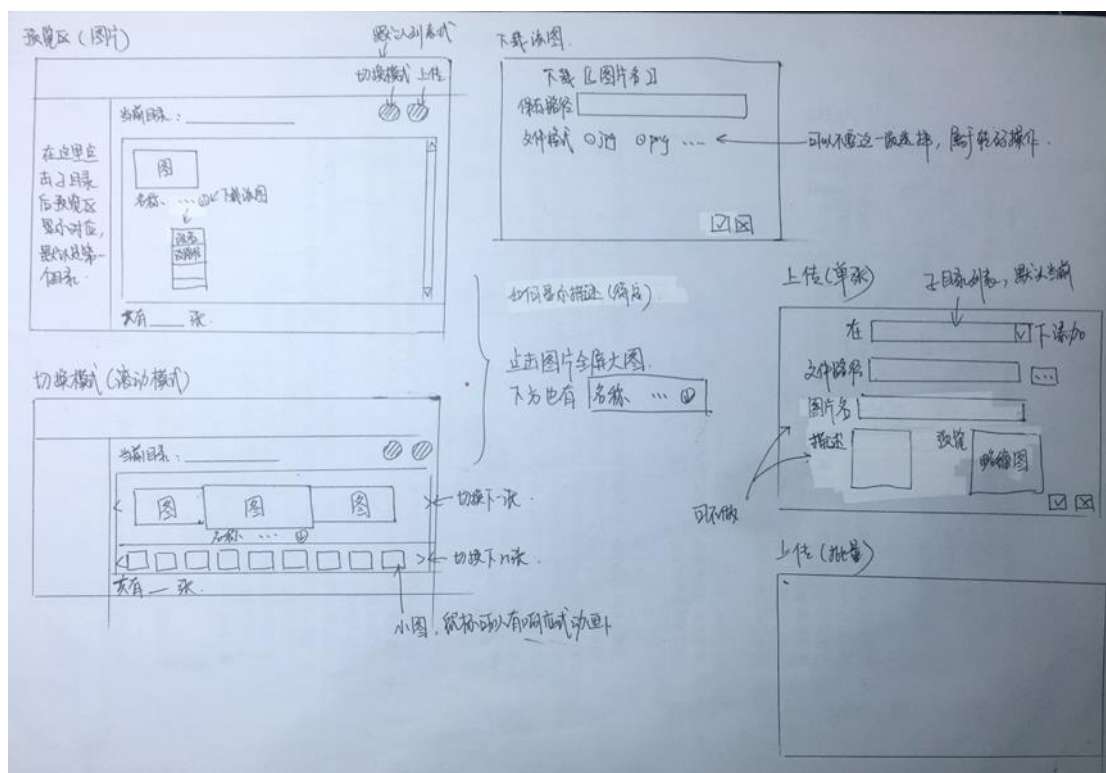


Figure 3 设计稿 (2)

二、系统关键技术介绍

1. 前端

本次前端的使用了 CSS(Bootstrap) + JS(jQuery)实现。

- 使用 CSS 和 Bootstrap 实现了前端的原创排版。
- 使用 JS 和 jQuery 制作了页面动效。
- 使用 JSON 格式进行前后端的数据传递。

A. 验证码的动态显示

<见 login.html 和 register.html>

使用了 HTML 5 的<canvas>标签在页面中加入画布，调用 JS 动态绘制吉祥物和验证码。

B. 动态显示文件夹

因为有增加文件夹的操作，所以需要动态更新文件夹，且需要保持文件夹的层级关系和展开收起的功能。

a. 文件夹初始化和层级关系的实现

在页面加载完毕后，调用初始化函数。向后端请求获得和文件夹有关的 JSON 数据串。数据串的格式如下：

```
{List: [1,2,3], type: [1,1,1], dName: ["DAM", "数字摄影", "计算机动画"]}
```

List 为该文件夹子文件夹的 id, type 为是否作为父文件夹 (1 是, 2 不是), dName 为文件夹的名字。

使用 jQuery 将从 JSON 数据串中解读出来的信息，填入 HTML 主体中。

如果存在为父文件夹的文件夹，递归重复此过程，直到不再存在父文件夹

通过和标签的正确嵌套保证父子文件夹间的层级关系。

b. 文件夹的展开和收起

使用 Bootstrap 的 collapse 组件+jQuery 实现内容的隐藏, 从而实现文件夹的收起和展开功能。关键在于正确地嵌套和关联。

通过 jQuery 判断 HTML 中的文本内容, 将文件夹前的符号▲ / ▼ 进行动态改变, 用不同符号代表文件夹当前的状态, 给用户更好的视觉化体验。

C. 显示不同文件夹中的内容

<见 portfolio.html>

通过 hash 锚定的功能动态展示不同文件夹下的内容。

每当锚值改变, 根据锚值 (和文件夹 id 关联) 向后端请求该文件夹下的文件数据, 获得 JSON 字符串, 格式如下:

```
{description: "图片介绍: 这本作业的一个成果展示图", filename: "图片名字", status: "success", path: "asset/source/1.jpg"}
```

使用 jQuery 动态插入 HTML 主体中。

D. 分页显示搜索结果

<见 search.html>

和显示不同文件夹中内容的逻辑类似, 故略去。通过锚定的变化动态向后端请求数据并且更改 HTML 内容。为按钮增加点击改变锚值的函数查看不同分页的内容。

E. 模态窗中动态生成目录

<见 portfolio.html>

使用了 Bootstrap 的组建模态窗减少页面跳转。

在初始化文件夹层级关系时, 将文件夹名称和 id 存入数组。每当读取完一个文件夹下所有普通文件夹后, 存入标识符以在目录中插入分割线。

在页面加载后插入 HTML。

2. 后端

A. 上传作品

上传用户个人作品是本系统的核心功能之一。上传文件功能我们使用 flask-uploads 库实现, 它为我们解决了文件名、文件类型和大小的问题和隐患, 使我们能将精力集中在处理上传成功的文件内容上而不是上传的细节。文件上传的接口为 POST 方法请求"/portfolio", 成功上传的文件将被赋予一个以当前时间的 md5 值编码的文件名存入临时目录, 随后文件 (目前实现的文件类型为图片类型) 将被添加数字水印并转存至用户文件夹, 同时图片的元数据会被提取。最后, 文件名、用户名、存储路径和所有相关的文件信息将被存入数据库。

B. 浏览作品

在线查看用户已上传的作品同样是核心功能之一。作品信息我们通过前后端的异步请求交互来传送而不是内嵌于 HTML 文档中, 其优点在于轻量、灵活, 随用随取。前端以文件 ID 请求路由"/query"后, 后端将从数据库中查找所请求的文件的相关信息并构造文件 URL, 以 JSON 格式回传给客户端。

C. 打包下载

用户能否快速地下载从而得到自己的作品集决定了用户使用本系统的满意程度。当用户请求下载自己的作品集时, 后端利用 shutil 库的 make_archive()函数将用户目录下的指定文件夹封入压缩文件。在此之前, 由于在服务器的文件系统中, 文件夹是按照该目录被创建时被分配得到的目录 ID 命名的, 我们需要将

该目录和子目录递归地重命名为用户在创建该目录时指定的文件夹名字。最后，叶子目录 (leaf) 中的文件的文件名也应被替换为具有实际意义的文件名。

D. 用户系统

本系统实现了简单的用户管理系统，新创建的用户获得默认的用户头像、个人介绍、作品集和主目录等内容并且可以自由修改。各个用户之间的数据相互隔离。

E. 标签和搜索

用户在上传个人作品的同时可以为作品添加至多 3 个标签信息，它们将随文件元数据一并存入数据库。本系统还提供了按标签搜索个人作品的功能，方便用户检索往期作品。

三、 用户操作指南

另请参阅《Berryfolio 用户使用指南》。

四、 测试报告

另请参阅《Berryfolio 测试报告》。

五、 其他

1. 数字水印

每一份上传至 Berryfolio 的作品都会被系统添加数字水印，算法为将输入图片的每个像素的 RGB 末 2 位用于存储水印信息。其流程如下图所示。

2. 统一格式与媒体转换

上传至 Berryfolio 的作品均会被转码为无损格式 PNG 并保存，做到无论是 JPEG、PNG 还是 BMP 格式的图片文件，都能被转换为统一格式，方便进行数字媒体资源管理。

3. 移动端适配

前端网页使用网格布局 (Grid System)，可以从容应对小屏幕、移动端的访问和使用。

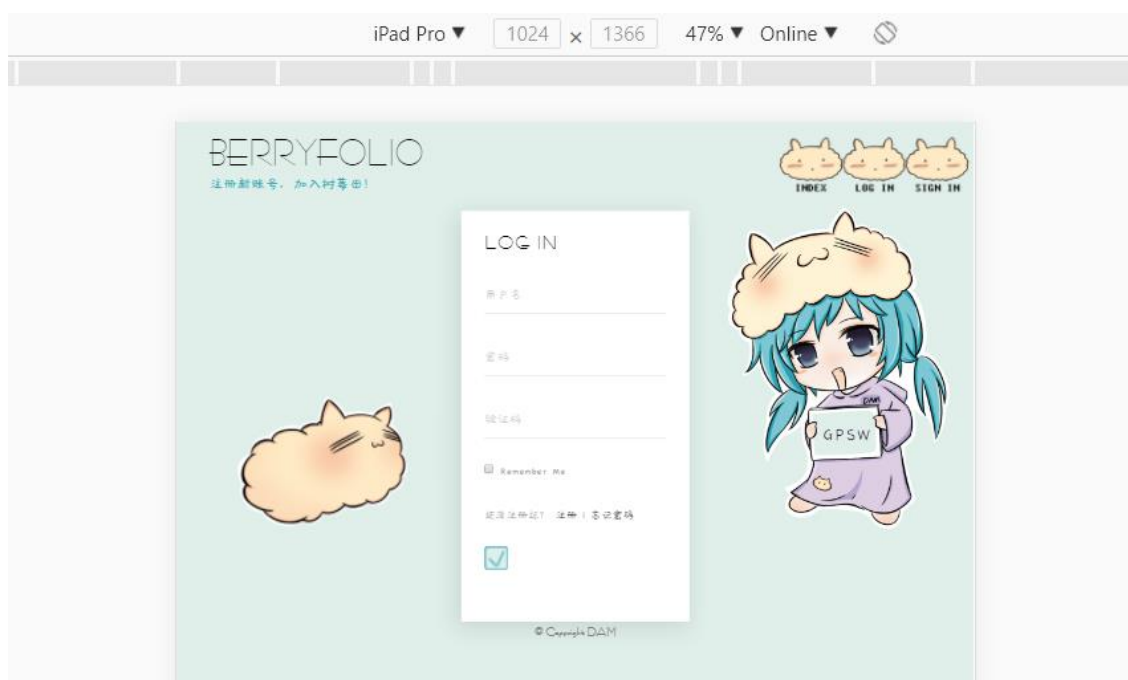


Figure 4 iPad Pro 访问



Figure 5 iPhone 6 Plus 访问

4. 流程管理与代码控制

Project 制作过程中, 我们进行了细致的流程管理。每个成员对自己的分工内容负责并且每日制定项目进度表。代码的版本控制使用 Git 实现并建立了远程仓库于 <https://gitee.com/YunzheZJU/DAMS>, 目前项

目已开源。项目的进度也可以从仓库的项目动态上看到。

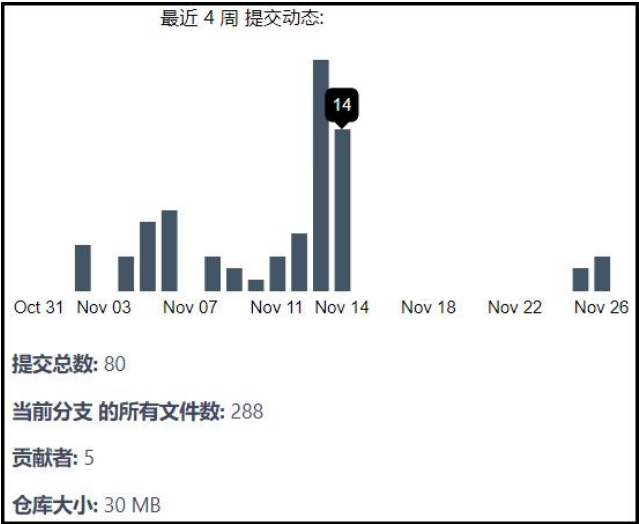


Figure 5 项目动态

六、 成员与分工

-  ***** : 主创、主美
-  ***** : 前端工程师
-  ***** : 数据库操作员
-  ***** : 后端工程师