

# 卓越软件开发基础 Project 项目文档

题目 旅游图片分享网站

学号 19302010021

姓名 张皓捷

定稿日期： 2020 年 07 月 28 日

# 目 录

<b>1</b>	<b>项目概况.....</b>	<b>1</b>
1.1	项目背景.....	1
1.2	项目完成情况.....	1
<b>2</b>	<b>项目详情.....</b>	<b>2</b>
2.1	导航栏.....	2
2.2	主页.....	2
2.3	搜索页.....	2
2.4	登录页.....	2
2.5	注册页.....	2
2.6	我的收藏页.....	2
2.7	上传页.....	3
2.8	我的照片页.....	3
2.9	加好友页.....	3
2.10	我的好友页.....	3
<b>3</b>	<b>项目总体结构.....</b>	<b>4</b>
3.1	前端.....	4
3.2	后端.....	4
3.2.1	后端总体实现情况.....	4
3.2.2	数据库结构.....	4
3.2.3	实体类.....	4
3.2.4	DAO 层.....	4
3.2.5	Service 层.....	5
3.2.6	Servlet.....	5
3.2.7	用户访问网站时经历了什么.....	5
<b>4</b>	<b>关键功能的实现.....</b>	<b>6</b>
4.1	注册页的前端展示.....	6
4.2	单点登录.....	6

4.3	我的足迹 .....	6
4.4	好友系统的设计 .....	7
4.5	验证码 .....	8

# 1 项目概况

## 1.1 项目背景

随着经济社会发展，人们出行旅游的需求不断增多，人们分享自己旅行所见所闻的需求意愿也越来越强。为满足人民日益增长的精神文化需要，【Daddy 旅行社】委托世界一流大学--复旦大学的同学们充分运用所学知识，完成一个旅游图片分享的交流平台

## 1.2 项目完成情况

完成了全部的基础功能。Bonus 中完成了图片局部放大功能，登录与注册验证码功能，用户对图片的评论功能和部署服务器：

项目演示地址: <https://photoshare.miaowumiaowu.buzz/index>

## 2 项目详情

### 2.1 导航栏

导航栏在大多数页面中都存在，可以导航到主页和搜索页。当用户未登录时，导航栏下拉菜单处显示未登录。点击下拉菜单可以跳转到登录页面。当用户正确登录后，下拉菜单包括我的收藏，上传图片，我的图片，加好友，我的好友，退出登录等选项，可以跳转到对应的页面。

### 2.2 主页

主页中间有轮播显示最热门（收藏数最多）的三张图片，下方显示了最新上传的三张图片。

### 2.3 搜索页

搜索页中提供了两组单选框。第一组设定搜索方式，用户可以根据标题或主题进行模糊搜索。第二组设定搜索结果的排序方式，用户可以根据发布日期和收藏数进行倒序排序。搜索结果支持分页。

### 2.4 登录页

用户可以在登录页输入账号，密码和验证码进行登录。登录页有为未注册过账号的用户设计的跳转到注册页面的链接按钮。

### 2.5 注册页

用户可以在注册页输入用户名，电子邮箱，密码和验证码进行注册。其中用户名必须在 4-15 位之间，密码不得小于 6 位，也不得大于 15 位，电子邮箱格式必须正确，密码和确认密码的输入必须一致。以上都可以在用户输入时进行实时检测，并马上给用户反馈。

### 2.6 我的收藏页

用户可以在该页设置自己的好友是否可以看到自己的收藏。

用户可以在该页看到自己的浏览足迹。最多显示 10 张图片的浏览足迹。点击图片足迹中的标题可以跳转到对应的图片详情页面。

用户可以在该页查看到自己收藏过的图片，并可以取消收藏。

## 2.7 上传页

用户可以在该页上传自己的图片。

## 2.8 我的照片页

用户可以在该页看到自己上传的所有图片。并可以对图片进行修改和删除操作

## 2.9 加好友页

用户可以在这个页面根据用户名进行模糊搜索，找到其他用户，并可以向指定的用户发出好友请求。

## 2.10 我的好友页

用户可以在该页看到别人向自己发送的好友申请，并可以选择接受请求或拒绝请求

用户可以在该页看到与好友有关的系统消息。主要是“xxx 已同意你的好友请求”，“xxx 已删除你好友”之类的提示。

用户可以在该页看到自己的好友列表。点击好友用户名可以跳转到“别人的收藏”页面。如果目标用户允许被好友查看收藏，则系统会显示目标用户的收藏。否则系统会提示目标用户不允许被好友查看收藏

## 3 项目总体结构

### 3.1 前端

本项目主要通过了 Bootstrap, JavaScript 和 jQuery 实现了前端的功能。

本项目为大多数页面设计了一个单独的 JavaScript 类, 这些文件都以\*\*\*Page.class.js 结尾, 存在于/js/class 目录下, 如 SearchPage.class.js、UploadPage.class.js 等。这些类用于封装页面的各个 HTML 元素, 页面属性和页面上的各种操作, 同时可以向后台发送 Ajax 请求, 得到后台返回的数据。

### 3.2 后端

#### 3.2.1 后端总体实现情况

本项目使用了 JavaBean, Servlet 和 JSP 实现了后端逻辑。数据库使用了 MySQL。

#### 3.2.2 数据库结构

在原有的数据库基础上, 我对数据库结构进行了以下改动:

(1).在 traveluser 表中添加了 salt, sessionID 和 canBeSeenFavor 三个字段。其中添加 sessionID 字段是为了实现单点登录(即: 同一个用户只能在一处登录, 如果在另一处登录, 之前的登录状态会消失)

(2).在 travelimage 表中增加了 dateReleased 字段, 用来存放该图片的发布日期

(3).增加了 friendrequest 表, sysmessage 表, travelfriendrecord 表, travelimagecomment 表, travecommentfavor 表, 分别用于存放好友请求, 系统消息, 好友记录, 对图片的评论和对图片评论的点赞的相关信息。

#### 3.2.3 实体类

本项目中编写了以下的实体类:

City 类, Comment 类, CommentFavor 类, Country 类, FriendRecord 类, FriendRequest 类, Image 类, ImageFavor 类, SysMessage 类和 User 类。

#### 3.2.4 DAO 层

本项目中编写了以下的 DAO 层，主要负责对对应的数据表进行增删改查的操作。

CityDAO 类，CommentDAO 类，CommentFavorDAO 类，CountryDAO 类，FriendRecordDAO 类，FriendRequestDAO 类，ImageDAO 类，ImageFavorDAO 类，SysMessageDAO 类和 UserDao 类。

### 3.2.5 Service 层

CommentService 类，FriendRecordService 类，FriendRequestService 类，ImageService 类，SysMessageService 类和 UserService 类。

### 3.2.6 Servlet

本项目为每个页面都单独制作了一个 Servlet，放在了 com.haojie.servlet.pageServlet 包中。此外 com.haojie.servlet.otherServlet 中还有另几个 Servlet。他们是：

UseServlet 类，ImageServlet 类，CommentServlet 类，CountryCityServlet 类：分别用来处理前端发出的和用户，图片，评论，国家城市有关的 Ajax 请求。

AuthImageServlet 类：主要用于生成验证码，并把验证码写到 session 中。

### 3.2.7 用户访问网站时经历了什么

用户访问本项目的网站时，主要有两种可能

(1).用户直接访问 Servlet，Servlet 进行相关的业务逻辑操作，获得数据，并将数据转发到 JSP 中，JSP 将数据填充到对应的模板中，生成 HTML，最后发回给用户

(2).用户点击页面上的按钮，由 JavaScript 发送 Ajax 请求到后台 Servlet，Servlet 进行相关的业务逻辑操作，将返回的数据编码成 JSON 格式，发回给前台。前台再由 JavaScript 解析 JSON 数据，通过 DOM 操作将数据呈现在前台页面上。



## 4 关键功能的实现

### 4.1 注册页的前端展示

注册页中为相关的 input 元素添加了 oninput 类型的事件。只要用户在输入框中输入数据，事件就会被触发，检测用户的输入是否符合要求，并立刻通过 DOM 操作向用户给出提示。

关于密码强弱性的展示：我将字符分为了四类：(a).数字 (b).大写字母 (c).小写字母 (d).其他字符。创建一个描述密码得分的变量。当用户密码长度小于 6 或大于 15 位时，密码得分为 0 分。否则，上述(a)(b)(c)(d)四类字符，密码中每出现一种，可以得到 25 分，最多得到 100 分。最终密码得分情况会在前台通过进度条予以实时显示。

### 4.2 单点登录

本项目中实现了用户的单点登录（即：同一个用户最多只能在一处登录自己的账号。如果用户已经在 A 处登录了自己的账号，这时又在 B 处登录自己的账号，则 A 处的登录状态会自动取消）。实现思路是：一个用户最多只能对应一个 sessionId。因此，我在 traveluser 类中添加了 session ID 字段，当用户通过登陆页面(/login)手动登录时，若登录成功，系统会更新数据库中对应的 sessionId，并将用户名写入到 session 中。当访问其他页面时，系统先根据 sessionId 和 session 中存储的用户名，在数据库中查找对应的用户。如果能找到则显示用户的信息，如果不能找到，则显示未登录。如果用户在多地重复登录，则数据库中的 sessionId 会被更新为他最后一次登录时的 sessionId。因此，当前面一处页面刷新时，就会找不到对应的用户而显示未登录，这样就实现了单点登录。

此外，用户每次登录后，系统都会更换 sessionId 以防止固定会话攻击。

### 4.3 我的足迹

本项目采用了后端+cookie 的方式实现我的足迹功能。首先在后端建立了 BrowseRecord 和 SingleBrowseRecord 类。在访问图片详情页面时，若系统在数据库中找到了图片，则使用 BrowseRecord 中的 addRecord 方法添加一条包含了图片 imageID 和标题中的记录。addRecord 方法是经过定制的。当已经存在的记录小于 10 时，则直接添加记录。当已经存在的记录等于 10 时，该方法会将第 2 至第 10 条记录向前移动一位（这

样第 1 条记录就没有了)，再将第 10 条记录设成本次新的浏览记录。此后，将 `browseRecord` 对象 JSON 化。再新建一个 cookie，name 为用户的 uid，value 为 JSON 化的 `browseRecord` 对象。再将该 cookie 添加到 response 中发回给前端。此后当用户再浏览一张图片，则根据用户的 UID，找到 cookie 中 JSON 化的 `browseRecord` 对象，将其反 JSON 化得到真正的 `browseRecord` 对象，再向其添加记录，再 JSON 化放到回 cookie 里。

当用户浏览我的收藏页面时，通过 JavaScript，读取 cookie 里 JSON 化的 `browseRecord` 对象，反 JSON 化后遍历里面的记录，利用 DOM 在页面上显示，并设置好相应的跳转即可。

此外，我的收藏页还提供了一个清空自己的足迹的按钮。点击该按钮后，会通过 JavaScript，将存储足迹用的 cookie 设为过期的。这样足迹就被清空了。

在实现这个过程中遇到的问题：(1).JSON 化后的 `browseRecord` 对象可能含有空格等 cookie 中不允许有的字符。解决方法是对 JSON 化后的字符串进行 URL 编码，使用的时候再解码即可。(2). Cookie 对存储的内容有 4KB 的大小限制。本次项目需求是显示 10 条足迹，只要图片标题不是特别特别长，4KB 一般能满足需求。但如果项目需求更改为需要显示 100 条或更多足迹，或图片标题存在特别特别长的情况，则可能需要考虑别的方法实现。

## 4.4 好友系统的设计

和好友相关的主要有两个页面。一个是“加好友”页面，另一个是“好友列表”页面。用户可以在“加好友”页面中搜索好友，并向其发送好友申请。项目建立在数据库中了一个 `friendrequest` 表，发送好友请求时在其中插入一条数据，包含有好友请求发出者的 uid 和好友请求接收者的 uid。另一个页面是“好友列表”页面。这个页面上会显示我收到的好友请求，并可以选择同意或拒绝。当用户同意时，系统会在 `friendrecord` 表中添加两条记录。一条记录的 uidA 为我的 uid，uidB 为对方的 uid，另一条记录 uidB 为我的 uid，uidA 为对方的 uid。这样两个用户就成为了好友。不论是同意还是拒绝，都会在 `friendrecord` 里删除对应的好友请求记录并给请求的发出者发送一条对方同意了或是拒绝了提示（利用 `sysmessage` 表）。当用户选择删除好友时，系统会删除 `friendrecord` 里对应的两条好友记录，并给被删好友的用户发出提示（利用 `sysmessage`

表)。

## 4.5 验证码

实现思路很简单。后台随机生成 6 位长的字母字符串，生成一个具有干扰作用的图片，再将字母字符串存到 `session` 里。用户登录或注册的请求提交后，在后台验证用户填写的验证码是否和 `session` 里存储的一致即可。此外，每次用户登录和注册的请求提交后，不论是否成功，都应该更新验证码，即：一个验证码只能使用一次。