Souvenirs需求分析文档

1. 项目概述
2. 项目概述

见project.docx。

1. 用户

学生及其周围群体。

1. 参与者

用户、小组。

1. 名词约定

* 自我评论：对应项目文档的comment；
* 相册：逻辑上表示用户照片存储的位置，物理上对应用户目录下的一个文件夹；
* 毕业照：指通过用户上传的照片按照一定的模板生成的照片组合；
* 小组：又称“组”，是多个注册用户所组成的一个集体。用户可以分享自己的照片到小组中，小组中的所有成员都可以看到其他用户分享的照片。用户可以经由小组生成带有其他用户分享照片的毕业照；
* 实际存在：指的是小组存在的意义，从时间和空间两个角度来看：时间上，一个小组应该在一定的时间内保持一定的活跃度；空间上，一个小组至少应由两人组成。满足这两条的小组，被称为实际存在的小组。

1. 项目配置要求

|  |  |  |  |
| --- | --- | --- | --- |
|  | 名称 | 版本 | 架构 |
| 操作系统 | Win10 | Build 1607 | X64 |
| 开发环境 | Eclipse | Mars | X86 |
|  | Mysql Workbench | 6.3 | x64 |
| Java | JDK | 1.8 | X64 |
| 服务器 | Tomcat | 8.0 | X64 |
| 数据库 | Mysql | 5.6 | X86 |
| CSS | Bootstrap | 3 | ---- |
| 连接件 | Apache File Upload | 1.3.3 | ---- |
|  | Apache DBCP | 2.1.1 | ---- |
| 浏览器 | Chrome | 53.0.2785.143 m | X64 |
|  | IE | 11.447.14393.0 | X64 |
|  | Safari(for iPad) | 6 | ---- |
|  | UC Browser(for Android) | 10.4.2.585 | ---- |

1. 架构

MVC+DAO

1. 功能需求描述
2. 用户注册

Souvenirs只向注册用户提供服务。注册的业务组成基本是填写相关注册信息(用户名、密码、密码确认)后提交表单，后台根据提交的信息在用户数据表中插入一条数据。

在提交登录信息之前需要对注册信息的合法性进行检查：

|  |  |  |
| --- | --- | --- |
| 检查域 | 检查条件 | 解释 |
| 用户名 | 3~50个字符，只可以由半角英文字符和数字组成，用户名不与已有用户冲突。 | 不含其它字符以避免SQL注入 |
| 密码 | 6~50个字符 | 小于6位的密码安全性太低 |
| 密码确认 | 应该与密码的内容完全一致 | 保证用户确认注册的密码 |

在注册成功后，系统会为用户分配一个唯一的ID。ID的格式为#XXXXXXXXXX(共11位)，其中#为引导提示符，用来与用户名区分。注册成功后，用户有一个默认相册。

在后续增量中会加入绑定邮箱的信息。

1. 用户登录

用户在进行所有操作之前需要登录，登录的方式为传统的输入用户名/ID、密码和验证码的方式。使用同一用户的用户名和ID可以得到相同的结果。

登陆成功后会进入用户主页，用户可以在该页面查看自己的个人资料、查看相册、选择相关操作(如上传文件、生成毕业照、修改账户信息等)。如果登陆失败会停留在登录界面上，显示登录错误信息。错误信息有两种：一是用户名或密码错误[[1]](#footnote-0)；二是验证码错误。用户名或密码错误的情况下，系统将清空密码框，并且重新加载验证码；验证码错误的情况下，系统将保留密码框中内容，只重新加载验证码。

登录分为多个增量开发：第一个增量只完成基本的登录；第二个增量完成记住用户操作；第三个增量结合绑定邮箱完成密码重置操作。

1. 用户管理自己的账户

用户可以修改自己的用户名和密码，但是不可以修改ID。修改的内容也需要检查合法性，详情参考功能1。

在后续增量中会添加修改绑定邮箱操作。

1. 用户上传自己的照片并自我评论

上传照片的操作在个人主页上完成。用户在个人主页上选择待上传的照片(一张)，然后在下方添加评论以及选择相册，最后选择“上传照片及评论”完成上传。

上传的照片会按照“/data/用户名/相册名/时间戳\_照片名”的路径格式存在服务器中。

1. 用户制作毕业照

用户选择“制作毕业照”选项进入制作页面。用户首先选择一个合适的模板，然后根据模板中的空缺选择填入的照片。照片添加完成后选择“完成制作”，系统会打开一个新的标签页显示生成的毕业照。(通过打印的方式导出)

1. 用户管理自己的相册

用户选择“相册管理后”会进入管理界面：用户可以添加一个相册；删除一个相册(相册中内容将一并删除)；修改相册的名字；修改相册的简介；修改相册的查看权限；对相册中照片重命名；添加一张照片；删除一张照片；修改照片的评论。

所有操作都要完成在数据表中的映射；其中添加/删除/更名相册、添加/删除/更名照片还涉及文件系统的操作。

1. [for task1]用户完善自己的资料(待完善)

用户可以添加自己的个人资料来展示自己，如：性别、年龄、职业、兴趣、头像等等。用户的某项个人资料可以设成私密，这样别人访问到自己的主页就不会显示出来。

1. 用户新建小组

所有的注册用户都可以创建一个小组，用户可以为小组创建名称和简介。每个小组都有唯一标识ID，这个ID是查找、加入、退出小组所必需的。创建的小组是逻辑概念，没有物理实现。系统通过数据表来维护小组实体。

1. 用户加入一个已存在的小组

用户可以搜索小组的ID(或名字，高级功能)来加入一个小组。当有用户加入小组的时候，小组的每个用户将收到一条消息，以提示有人加入了该小组。

1. 用户向一个加入的组分享照片

用户首先进入自己的照片页面，然后选择“分享到小组”，选择想要分享的小组后，该照片就可以被小组中的所有人查看了。小组中一个成员分享的照片不能被其他用户再次分享，分享的照片只是多了一个读取权限但是文件位置没有任何变动。

1. 用户解除分享自己的照片

用户在小组内选择自己分享的照片，然后选择“不再分享照片”，此时该照片便不能再被小组内用户所用。注意的是，如果有用户使用了已撤销分享的照片作为自己的毕业照，重新生成的时候该照片将不可用。

1. 升级功能5：用户可以将组内的照片用作自己的毕业照

基本的生成方式与生成个人毕业照相似，用户可以选择更多的照片：个人照片和通过小组分享的照片。

1. 用户退出所在的小组

用户在想要退出的小组中选择“退出小组”，即可退出。退出后所有已分享到该小组的照片都会解除分享；并且用户也不能看到已退出小组的分享照片。

1. 检查小组实际存在性

在用户退出的时候，系统对退出的小组进行空间实际存在性的检查，如果没有成员将删除小组；定期进行全部小组的实际存在性检查，对于长时间没有活跃或者只有一个用户的小组，向组内成员发送提示消息。一段时间后(如7天)仍然不满足实际存在条件的，将删除该小组；对于没有用户存在的小组，检测到则直接删除。

1. [进一步]防止SQL注入

通过限制用户名的格式，对密码中敏感字符的转义解决SQL注入问题。

1. [进一步]用户密码传输过程中加密

在客户端使用SHA1对用户输入的密码进行编码，然后在服务器端采用PBKDF2加盐值的方法编码，在很大程度上保证用户密码难以被破解。

1. [进两步]用户邮箱认证与密码找回

在注册的时候需要填写验证邮箱，然后系统向对应的邮箱发一封验证邮件，里面包含用户的验证信息。邮件有效期一般设为15~30分钟。用户登录验证邮箱后，点击该链接，跳转到验证页面。系统对验证请求进行校验，校验成功则该邮箱称为用户的验证邮箱；不论是否验证成功，验证链接都将失效。

1. [进两步]用户免密码登录

在用户登录的时候如果选择“记住我”，系统会生成一个验证cookies给浏览器，并且设定其过期时限为30天。在有效期内，用户打开主页的时候回传递cookies给系统，系统会从cookies中解析出用户信息并直接跳转到登录界面。

1. [进三步]生成毕业照的导出

用户可以将毕业照到处成jpg格式，下载到本地。

1. 分析图表
2. 项目进度安排

1. 从安全角度看，为了降低暴力破解的可能性，不能给出只有密码错误的提示。如果这样做会无形中透露给攻击者“他所尝试的用户是存在的”的信息。 [↑](#footnote-ref-0)