**基于Flask框架的艺术创作平台**

**——项目需求规格说明书**

2020/04/06

**版本记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 修改说明 | 参与者 |
| V1.0.0.200325 | 2020.3.25 | 完成第一版 | 马广洲，张祥国，潘安佶，牟秋宇，王康明，刘佳恒，常佳辉 |
| V2.0.0.200330 | 2020.3.30 | 更换项目，大幅修改说明书内容和结构 | 马广洲，张祥国，潘安佶，牟秋宇，王康明，刘佳恒，常佳辉 |
| V2.1.0.200404 | 2020.4.04 | 针对老师的点评，更改了“参与者”和第1.4节“参考资料” | 马广洲，张祥国，潘安佶，牟秋宇，王康明，刘佳恒，常佳辉 |
| V2.1.1.200406 | 2020.4.06 | 校对文字、术语，标题图表格式，用例图和RUCM | 马广洲，张祥国，潘安佶，牟秋宇，王康明，刘佳恒，常佳辉 |

目录

[一、引言 1](#_Toc37067215)

[1.1 编写目的 1](#_Toc37067216)

[1.2 背景 1](#_Toc37067217)

[1.3 术语定义 1](#_Toc37067218)

[1.4 参考资料 2](#_Toc37067219)

[二、任务概述 2](#_Toc37067220)

[2.1 目标 2](#_Toc37067221)

[2.2 用户特点 3](#_Toc37067222)

[2.3 假定与约束 3](#_Toc37067223)

[三、需求与设计 3](#_Toc37067224)

[3.1 需求 3](#_Toc37067225)

[3.1.1业务需求 3](#_Toc37067226)

[3.1.2功能需求 4](#_Toc37067227)

[3.1.3 非功能性需求 5](#_Toc37067228)

[3.2 用例图 7](#_Toc37067229)

[3.2.1 总体用例图 7](#_Toc37067230)

[3.2.2 用户注册与修改用例图 8](#_Toc37067231)

[3.2.3 用户发布作品用例图 9](#_Toc37067232)

[3.2.4 用户风格创建与修改用例图 10](#_Toc37067233)

[3.2.5 后端系统与算法模块交互用例图 11](#_Toc37067234)

[3.3 RUCM 11](#_Toc37067235)

[3.3.1 游客浏览页面 11](#_Toc37067236)

[3.3.2 用户注册 13](#_Toc37067237)

[3.3.3 用户更改个人信息 14](#_Toc37067238)

[3.3.4增添管理员账户 15](#_Toc37067239)

[3.3.5 用户创建作品 17](#_Toc37067240)

[3.3.6 用户上传作品 18](#_Toc37067241)

[3.3.7 用户评价作品 19](#_Toc37067242)

[3.3.8 用户创建风格 20](#_Toc37067243)

[3.3.9 用户修改已有风格 22](#_Toc37067244)

[3.3.10 后端系统请求运行算法 23](#_Toc37067245)

[3.3.11 算法模块调度运行队列 25](#_Toc37067246)

[3.3.12 后端系统请求预期运行时间 25](#_Toc37067247)

[3.3.13 算法模块通知运行结果 27](#_Toc37067248)

[四、运行环境规定 28](#_Toc37067249)

[4.1 设备 28](#_Toc37067250)

[4.2 支持软件 28](#_Toc37067251)

[4.3 接口 28](#_Toc37067252)

[4.3.1 硬件接口 28](#_Toc37067253)

[4.3.2 软件接口 29](#_Toc37067254)

[4.3.3 通信接口 29](#_Toc37067255)

[4.3.4 用户接口 29](#_Toc37067256)

# 一、引言

Flask是一个使用 Python语言编写的轻量级 Web 应用框架，本项目是基于Flask框架开发一个支持艺术绘画处理等功能的艺术创作平台，本需求规格说明书详细描述了所开发项目的业务需求及流程、功能性需求、非功能性需求、环境需求、技术架构及开发路线。

## 1.1 编写目的

本软件需求规格说明书，是为了软件设计、软件测试人员和用户所编写的。

本软件规格说明书的适用读者，包括参加能力验证的开发测试人员、Flask技术人员，以及项目的其他相关人员。

## 1.2 背景

软件名称：Flask

项目的组织机构：Flask开源项目开发组

项目的实施机构：Github网站上的568位贡献者

项目背景：本项目是一个支持多种艺术绘画处理功能的Web网站，使用者可以使用其对照片进行艺术化处理，同时可以同其他用户进行简单的社交。

## 1.3 术语定义

|  |  |
| --- | --- |
| Engine | 引擎，处理整个系统的数据流处理、触发事务，是整个框架的核心。 |
| Flask-login | Flask框架的插件，提供用户会话管理，处理常见的登录、退出和注册的任务。 |
| Neural Style transfer | 使用卷积神经网络将一张图片的语义内容与不同风格融合起来的过程 |
| Bootstrap | 一个非常受欢迎的Web前端框架，它是基于 HTML、CSS、JAVASCRIPT 的，用以快速开发响应式网页。 |
| Werkzeug | Werkzeug 是一个WSGI工具包，WSGI（Python Web Server Gateway Interface），是Python语言定义的Web服务器和Web应用程序或框架之间的一种简单而通用的接口。 |
| Jinja2 | Jinja2是基于python的模板引擎。 |

表1.3.1 术语定义

## 1.4 参考资料

[1] GB/T 9385-2008 计算机软件需求规格说明

[2] GB/T 8567-2006 计算机软件文档编制规范

[3] Bootstrap官方文档 https://getbootstrap.net/docs/getting-started/introduction/

[4] Flask官方文档 https://flask.palletsprojects.com/en/1.1.x/

[5] w3school flask学习教程：https://www.w3cschool.cn/flask/

[6]《Web安全攻防：渗透测试实战指南》

[7]《SQL注入攻击与防御(第2版)》

[8]《flask入门教程》：http://helloflask.com/tutorial/

[9] HelloFlask 论坛：https://discuss.helloflask.com/

# 二、任务概述

## 2.1 目标

Flask是目前十分流行的web框架，它的主要特征是核心构成比较简单，但具有很强的扩展性和兼容性，程序员可以使用Python语言快速实现一个网站或Web服务。Flask的基本模式为：在程序里将一个视图函数分配给一个URL，每当用户访问这个URL时，系统就会执行给该URL分配好的视图函数，获取函数的返回值并将其显示到浏览器上。

本项目是基于Flask框架开发的一个功能新颖的具有艺术绘画工具的艺术创作平台，它可以使每个人都只需单击几下即可创建和共享美丽的艺术照片，用户需要做的仅仅是上传照片并选择自己喜欢的风格，之后项目服务器将为用户渲染图稿并呈现出来。

除此以外，本网站项目还具备一定的社交功能，每个用户都可以很方便、快捷地评价其他用户的艺术照片作品以及回复评价等。

## 2.2 用户特点

本系统平台的使用者为喜欢分享、艺术化处理照片的网民，主要以6-40岁的青少年男女为主，用户年龄跨度大、社交需求高。大部分使用者具备独立的行事能力，少部分使用者需要适当的指导下进行操作。本软件无特殊情况是7\*24小时不间断运行。

## 2.3 假定与约束

（1）需求变更要严格按照需求变更流程。

（2）开发工作要在2020-5-30前完成。

（3）系统平台Web前端使用JavaScript, Html, CSS语言和Bootstrap框架进行开发，Web后端以及算法使用Python语言进行开发。

# 三、需求与设计

## 3.1 需求

### 3.1.1业务需求

艺术化处理照片曾经是熟悉PhotoShop、Adobe Illustrator等软件的专业相片处理人员的事情，但是随着人民生活水平和国家科技水平的提高，以手机为代表的移动设备几乎全员覆盖我国居民，而手机上的像素水平越来越高的相机功能也大大激发了普通用户后期处理所摄照片的需求。

普通用户如果想要使用后期处理照片的功能，以前往往需要下载高达几个GB的软件，再进行高深、困难的一系列操作。对于大多数普通用户而言，学习PhotoShop等专业软件是一件困难的事情，而本艺术创作平台项目旨在简化普通用户后期处理照片的步骤，用户只需上传自己所需处理的照片，再选择所需的艺术处理效果，然后进行提交，Web后台服务器即可自动生成对应的艺术化处理过后的照片，并且将其展示到网页前端上。用户可以将该照片下载到本地。用户可以出于保护隐私的目的仅仅是下载，也可以出于分享的心态，将其保存分享到Web网上，其他用户可以对其进行观看、点赞、评论等。

这样可以给广大用户节省很大的学习成本、工作成本，同时也获得更多拍照的乐趣。

本项目可以分为Web前端、Web后端和算法处理三大模块。

前端模块涉及游客浏览、用户注册和登录、用户修改账户资料、浏览现有风格、浏览现有作品、上传照片文件、选择艺术化处理风格、生成艺术照片作品、下载照片、分享照片、创建新的艺术化处理风格、修改现有的艺术化处理风格等等、评价作品、回复评价等功能。

后台模块主要负责账户管理、任务调度、预期完成时间反馈、通知运行结果、图片存储、分享记录、评价回复记录以及数据安全保护等功能；

算法模块主要负责编写、实现各种艺术化处理风格的人工智能算法，并对其进行系统本身自带的数据集和用户上传的数据集进行训练。

### 3.1.2功能需求

当用户登录本艺术创作平台网址时，可以以游客的身份进行浏览、使用部分功能，但系统前端会引导其进行注册或者登录，登录之后可以进行照片上传操作，上传完之后，可以选择不同的艺术化效果进行自动处理，处理完之后可以选择下载或者分享。用户还可以进行照片分享以及照片评论与回复，用户登录之后可以看到自己自从注册以来的分享的所有动态和所有评论。

我们组需求分析的用户分成3类：1是游客，2是普通用户，3是系统管理员。

我们系统支持游客浏览平台作品的功能。新用户在成为新用户之前，肯定是一个没有账户的游客，本系统允许游客浏览现有风格，然后创建作品，感受本艺术创作系统的效果，但是游客若想发布分享创建的作品，会提示游客进行注册。

游客注册、填写资料之后成为普通用户。对于普通用户，他可以修改账号资料，修改个人信息要求短信验证，只有通过短信验证，用户修改的资料才会生效。

对于系统管理员来说，它可以创建平台的管理员，平台的管理员对本平台的账户、一些非法的照片作品进行适当的管理。

基于以上业务需求的分析与相应的业务要求，我们可以认为软件应当具备如下所述的功能需求：

1. 游客浏览。新用户在成为新用户之前，肯定是一个没有账户的游客，本系统允许游客浏览现有风格，然后创建作品，感受本艺术创作系统的效果，但是游客若想发布分享创建的作品，会提示游客进行注册。
2. 注册/登录/修改账号资料——要有负责将游客转换成新用户的注册、资料填写功能，新、老用户登录本系统，以及让老用户修改账号资料。新用户在注册时，系统会要求新用户填写完整的个人信息，然后收到确认修改的短信验证码，通过短信验证后，才可以注册成为新用户。老用户想修改个人信息时也会要求短信验证，只有通过短信验证，用户修改的资料才会生效。
3. 作品创作与发布。该功能是本系统最核心的一部分，用户在未注册前可以浏览现有的艺术化处理风格，浏览现有的作品。用户在登录账号之后，可以创建自己的作品，创建作品首先要求用户从本地上传照片，上传完照片之后可以对照片进行一些简单的预处理，比如降低照片分辨率、剪裁照片、左右上下翻转等，上传完照片之后，可以使用绘图创作新的艺术风格，也可以被引导选择、确定其所需的现有的艺术风格，然后进行提交。后台人工智能算法根据特定的艺术效果进行处理后，会生成对应的照片，照片会被显示给用户预览，用户可以选择预览的窗口的大小，以及是否和原照片进行对比预览，选择是否下载照片，或者返回重新创作作品。用户可以选择“下载照片”，系统将负责提供用户下载照片和分享照片的功能，用户下载照片时，可以选择下载照片的质量，照片格式，以及文件保存的具体路径。用户如果选择“分享照片”，可以输入所需配备的文字，即可将作品发布分享出去。作品发布分享出去后，别的用户可以对该作品进行评价，该用户可以对这些评价进行回复。除此，该用户也可以看到其他用户所分享的作品，并对其进行评价。
4. 创建新风格。创建新的艺术处理风格也是本系统一个非常重要的功能，用户可以上传固定的一些照片，系统会使用这些照片进行数据集训练，从而生成新的风格，用户可以将新的风格保存到云端，同时他也可以修改已有的风格。

### 3.1.3 非功能性需求

本基于Flask开发的Web网站，需要支持以下非功能性需求以便用户更方便、更快捷地使用本网站。

1. 性能需求。要求系统能满足1000个人同时使用，总请求数能满足10000次每秒，普通页面反应时间不能超过1秒，请求响应时间不超过0.2秒，图片处理时间不超过20秒，吞吐量在50000次每秒，CPU资源利用率超过50%。
2. 可靠性。Web站点要求能够以7×24小时连续不间断地运行，年非计划的停机时间不能高于30小时。Web系统能够快速地进行重新部署，特别是在系统出现故障时，能够在半小时内切换到备用机。
3. 可维护性。系统的可维护性是衡量一个系统的可修复(恢复)性和可改进性的难易程度。所谓可修复性是指在系统发生故障后能够排除(或抑制)故障予以修复，并返回到原来正常运行状态的可能性。而可改进性则是系统具有接受对现有功能的改进，增加新功能的可能性。当系统出现问题时，要能快速定位并解决问题的成本，因为成本低则可维护性好。其次代码要容易被人理解，容易修改和增强功能。要求同一模块的代码按照驼峰式书写，变量名要语义相关，不发生代码重用现象，结构和表现要分离，HTML只用来形成网页的结构，网页的布局、表现方式用CSS完成。
4. 可访问性。该网站的上网用户中，可能存在有一些视力受损的人，也可能存在一些红绿色盲症患者，又或是一些高度近视的用户，他们需要屏幕阅读器将网页内容读出来。要求本网站能以语义化的HTML标签编写的网页可以，这样可以使屏幕阅读器知道网页的结构及各部分的大概功能，如<nav>标签表示导航，<aside>标签表示侧边栏等等，使使用者更便捷地与网页交互，更快地抓住网页主要表现内容。没有鼠标的用户要靠键盘或者触屏与网页交互，要充分考虑键盘操作和网页交互，以及要满足大量使用移动设备访问网页的用户。
5. 设备兼容性。使用本网站的用户，可能使用各式各样的操作系统、浏览器，可能使用笔记本、平板、手机等各种电子设备，要求网页界面的设计，能够与开发应当根据用户行为以及设备环境（系统平台、屏幕尺寸、屏幕定向等）进行相应的响应和调整，使网页内容能更清晰地展现出来。
6. 安全性。使用HTTPS传输协议，选用知名的、安全性和稳定性高的服务器，不能出现DNS被劫持、网站页面被挂上各种恶意广告的现象；保障用户隐私数据的安全性，保密的信息只有经授权许可的用户才能使用，用户修改用户自己的资料、数据时要跟用户关联起来，要求用户设置安全的密码，复杂度比如字符种类、字符长度要达到一定强度，每半年提醒用户定期修改密码。严格审查表单和URI，严格避免产生SQL注入攻击。
7. 防攻击性。能防止SQL注入、XSS跨站、Webshell上传、非授权访问等多种Web攻击，防止挂马、黑链入侵、DNS攻击，无视DDOS攻击。其中，由于本网站经常使用图片上传，所以需避免文件上传漏洞，文件上传攻击指的是恶意攻击者利用一些站点没有对文件的类型做很好的校验这样的漏洞, 上传了可执行的文件或者脚本，并且通过脚本获得服务器上相应的权利,或者是通过诱导外部用户访问或者下载上传的病毒或者木马文件，达到攻击目的。为了防范用户上传恶意的可执行文件和脚本，以及将文件上传服务器当作免费的文件存储服务器使用，需要对上传的文件类型进行白名单校验，并且限制上传文件的大小，上传的文件，需要进行重新命名，使攻击者无法猜测到上传文件的访问路径。
8. 易操作性。要求网站易理解，单凭观察，用户能否在几秒内就知道网站是做什么的，以及知道如何采取行动。对于新用户，要求新用户初次接触网站时，能够很轻松地完成任务，可以不用帮助文档就能使用网站。即使需要使用帮助文档，也能通过帮助文档，用户可以顺利解决遇到的问题。此外还要提高效率性，当用户熟悉网站功能或界面后，能够可以非常快速、高效率地完成任务。
9. 低出错率。要求在使用网站过程中，用户因操作而出现的错误尽量地少，不管所发生的操作错误有多严重，都能容易地从错误中恢复。
10. 高满意度。用户使用这个网站，要求具有较高的满意度；要求具有用户反馈建议的渠道。
11. 用户界面美观。要求网站的导航栏目不超过10个，只需要列出几个主要的页面，网站页面相互协调，整个页面所展示的视觉效果良好，最后形成一个和谐的整体。网页还要符合人们从左到右、从上到下的阅读习惯，对于部分较长的页面，应在底部设置导航。

## 3.2 用例图

### 3.2.1 总体用例图

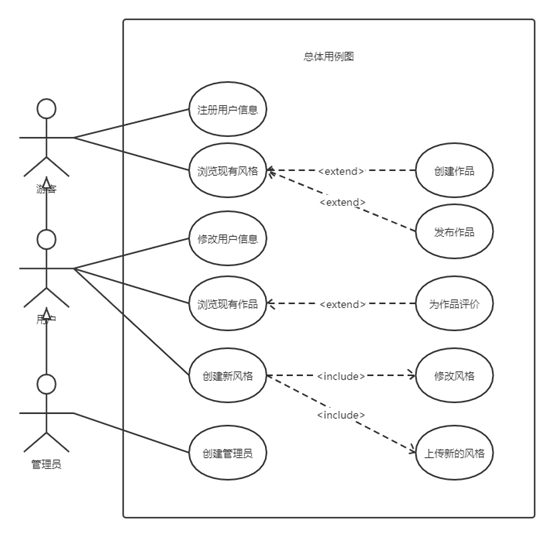


图3.2.1 总体用例图

该用例图描述了系统整体的用例与功能，包括用户的注册与信息修改、用户的作品发布与评论、用户增添新风格和修改已有风格等用例，而后端与算法模块的交互单独放在一个后面的用例图进行说明。

### 3.2.2 用户注册与修改用例图

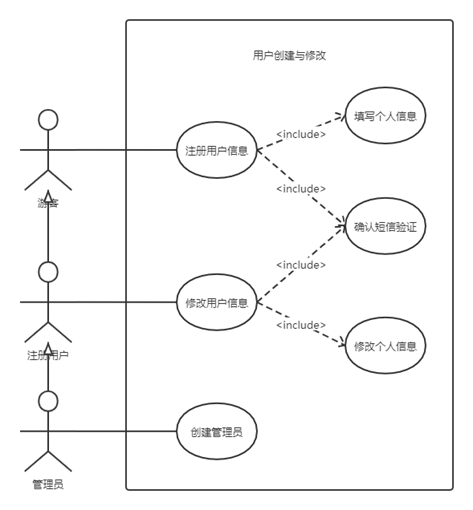


图3.2.2 用户注册与修改用例图

该用例图描述了新用户的注册、信息修改、管理员对管理账号的创建、短信验证等用例。游客角色需要进行注册才可以成为用户角色，并使用发布作品、评论等功能；而管理员角色可以对其他人的信息、作品、风格等进行修改。

### 3.2.3 用户发布作品用例图

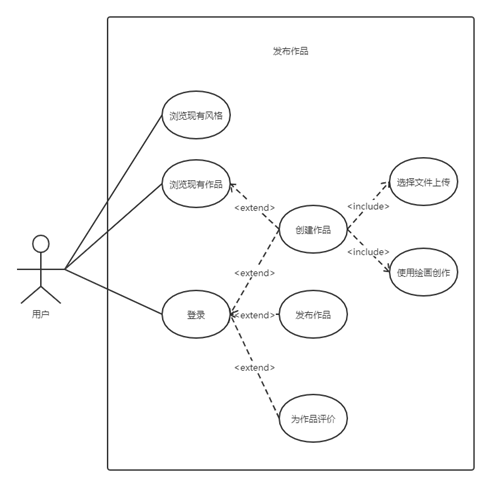


图3.2.3 用户发布作品用例图

该用例图描述了用户在登录后进行创建新的作品、发布作品、评论他人作品等用例，在用户创建作品的过程中，可以选择文件系统中的文件上传，或是使用系统提供的画布自行创造。

### 3.2.4 用户风格创建与修改用例图

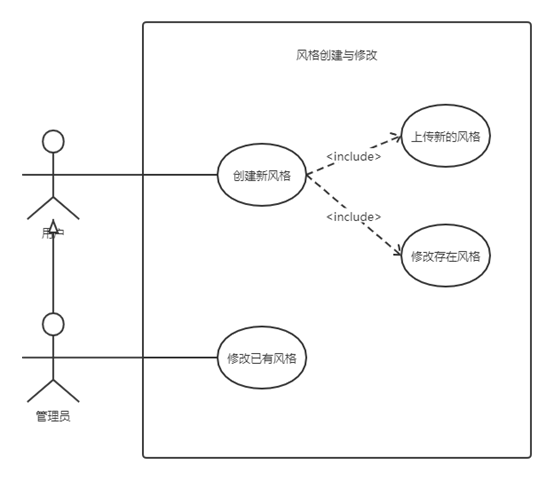


图3.2.4 用户风格创建与修改用例图

该用例图描述了用户及管理员创建、修改风格的用例图。用户可以选择自己的风格文件上传，以此创立新的风格，用户及管理员可以对这些风格进行修改与管理。

### 3.2.5 后端系统与算法模块交互用例图

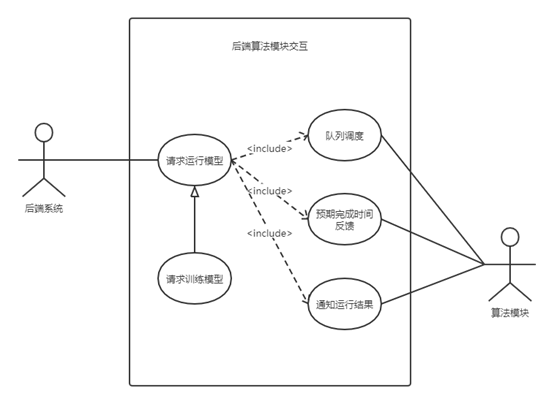


图3.2.5 后端系统与算法模块交互用例图

该用例图描述了后端系统与算法模块交互的用例，包括后端系统请求运行算法、请求预期完成时间；算法模块队列调度队列、通知运行结果等。

## 3.3 RUCM

### 3.3.1 游客浏览页面

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 浏览页面 | |
| Brief Description | 游客输入网址后进入网页浏览 | |
| Precondition | 游客获知网址 | |
| Primary Actor | 游客 | |
| Secondary Actors | 浏览器 | |
| Dependency | 无 | |
| Generalization | 无 | |
|  |  | |
| Basic Flow | Steps | |
|  | 1 | 游客连接网络 |
|  | 2 | 游客输入正确网址VALIDATAS THAT网址是否正确 |
|  | 3 | 服务器响应游客请求 |
|  | 4 | 服务器返回页面 |
|  | 5 | 游客浏览网页 |
|  | Postcondition | 游客成功获取网页并浏览 |
|  |  |  |
| Specific Alternative Flow | RFS2 | |
|  | 1 | 网址输入错误 |
|  | Postcondition | 网址有误，返回404 |
|  |  |  |
| Bounded Alternative Flow | RFS3-4 | |
|  | 1 | 服务器运行异常 |
|  | Postcondition | 服务器异常，无响应 |

表3.3.1 游客浏览页面RUCM

游客接入网络后，在浏览器中输入正确的网址后，向后端发送请求，服务器返回页面，游客可以在浏览器上浏览艺术创作平台页面。当网址输入错误时，游客无法获取艺术创作平台页面。

### 3.3.2 用户注册

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 注册正式用户 | |
| Brief Description | 游客注册成为正式用户 | |
| Precondition | 游客浏览页面 | |
| Primary Actor | 游客 | |
| Secondary Actors | 浏览器 | |
| Dependency | 无 | |
| Generalization | 无 | |
|  |  | |
| Basic Flow | Steps | |
|  | 1 | 1打开用户注册页面 |
|  | 2 | 2游客填写个人信息VALIDATAS THAT数据格式 |
|  | 3 | 3提交数据到服务器 |
|  | 4 | 4服务器启动短信验证码服务VALIDATAS THAT短信验证码 |
|  | 5 | 5数据库增添用户信息 |
|  | Postcondition | 游客成功注册成为正式用户 |
|  |  |  |
| Specific Alternative Flow | RFS | 2 |
|  | 1 | 数据输入有误 |
|  | Postcondition | 数据输入有误，刷新页面 |
|  |  |  |
| Specific Alternative Flow | RFS | 5 |
|  | 1 | 短信验证码发送有误 |
|  | Postcondition | 重新启动短信验证码服务 |
|  |  |  |
| Bounded Alternative Flow | RFS | 3 |
|  | 1 | 服务器运行异常 |
|  | Postcondition | 服务器异常，无响应 |

表3.3.2 用户注册RUCM

游客通过在用户注册界面填写必要的准确的信息，再经过手机短信验证后，注册为正式用户，注册过程中如果填写信息不符合要求，会刷新页面让游客重新填写，如果短信验证码发送有误，服务器则会重发验证码。

### 3.3.3 用户更改个人信息

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 修改用户信息 | |
| Brief Description | 注册用户修改用户信息 | |
| Precondition | 用户已登录 | |
| Primary Actor | 正式用户 | |
| Secondary Actors | 浏览器 | |
| Dependency | 用户已经注册 | |
| Generalization | 无 | |
|  | | |
| Basic Flow | Steps | |
|  | 1 | 打开用户信息修改界面 |
|  | 2 | 用户更改个人信息VALIDATAS THAT数据格式 |
|  | 3 | 提交数据到服务器 |
|  | 4 | 服务器VALIDATAS THAT Cookies |
|  | 5 | 数据库更改用户信息 |
|  | Postcondition | 正式用户更改个人信息成功 |
|  |  |  |
| Specific Alternative Flow | RFS | 2 |
|  | 1 | 数据输入有误 |
|  | Postcondition | 数据输入有误，刷新页面 |
|  |  |  |
| Specific Alternative Flow | RFS | 5 |
|  | 1 | Cookies过期 |
|  | Postcondition | 用户需要重新登录 |
|  |  |  |
| Bounded Alternative Flow | RFS | 3 |
|  | 1 | 服务器运行异常 |
|  | Postcondition | 服务器异常，无响应 |

表3.3.3 用户更改个人信息RUCM

已注册的用户可以在登录后进入用户个人信息修改页面，修改自己的个人信息，并提交到服务器。如果用户不处于登录态或cookies已经过期，则用户需要重新登录再进行个人信息修改的操作。

### 3.3.4增添管理员账户

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 增添管理员账户 | |
| Brief Description | 在后台管理系统中增添管理员账户 | |
| Precondition | 账户具有增添管理员权限 | |
| Primary Actor | 管理员 | |
| Secondary Actors | 浏览器 | |
| Dependency | 无 | |
| Generalization | 无 | |
|  |  |  |
| Basic Flow |  | Steps |
|  | 1 | 打开后台管理界面 |
|  | 2 | 输入管理员账号密码VALIDATAS THAT账号密码是否正确 |
|  | 3 | 打开增添管理员界面 |
|  | 4 | 输入增添管理员信息VALIDATAS THAT 信息格式是否正确 |
|  | 5 | 数据库中增添管理员账号 |
|  | Postcondition | 增添管理员账号成功 |
|  |  |  |
| Specific Alternative Flow | RFS | 2 |
|  | 1 | 账号密码输入有误 |
|  | Postcondition | 账号密码输入有误，刷新页面，提示重新输入 |
|  |  |  |
| Specific Alternative Flow | RFS | 4 |
|  | 1 | 增添管理员信息有误 |
|  | Postcondition | 管理员信息输入有误，刷新页面 |
|  |  |  |
| Bounded Alternative Flow | RFS | 5 |
|  | 1 | 服务器运行异常 |
|  | Postcondition | 服务器异常，无响应 |

表3.3.4 增添管理员账户RUCM

具有增添管理员权限的管理员账户可以在后台管理系统中增添管理员账户，首先管理员需要登录到后台管理系统中，再填写新增的管理员账户信息，增添管理员账户。如果新增管理员账户权限等信息输入不符合要求，会刷新页面，提示重新填写。如果管理员没有增添管理员权限，则无法进入新增管理员界面。

### 3.3.5 用户创建作品

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 创建作品 | |
| Brief Description | 已注册用户创建作品 | |
| Precondition | 注册用户已登陆 | |
| Primary Actor | 注册用户 | |
| Secondary Actors | 浏览器 | |
| Dependency | 无 | |
| Generalization | 无 | |
|  |  | |
| Basic Flow | Steps | |
|  | 1 | 打开作品创建界面 |
|  | 2 | 选择作品创建方式VALIDATAS THAT 创建方式 |
|  | 3 | 在画布上进行创作 |
|  | 4 | 保存画布上的内容 |
|  | 5 | 上传画布上的内容到服务器 |
|  | Postcondition | 作品上传成功 |
|  |  |  |
| Specific Alternative Flow | RFS | 2 |
|  | 1 | 选择文件上传方式 |
|  | Postcondition | 本地文件上传到服务器 |
|  |  |  |
| Bounded Alternative Flow | RFS | 5 |
|  | 1 | 服务器运行异常 |
|  | Postcondition | 服务器异常，无响应 |

表3.3.5 用户创建作品RUCM

注册用户可以在该艺术创作平台上进行创作，通过在作品创建页面选择创作方式，可以选择在网页端画布上绘画，也可以选择上传本地的作品到服务器上。

### 3.3.6 用户上传作品

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 发布作品 | |
| Brief Description | 注册用户发布作品 | |
| Precondition | 注册用户已登陆 | |
| Primary Actor | 注册用户 | |
| Secondary Actors | 浏览器 | |
| Dependency | 无 | |
| Generalization | 无 | |
|  |  | |
| Basic Flow | Steps | |
|  | 1 | 打开作品上传界面 |
|  | 2 | 选择已创建作品 |
|  | 3 | 添加上传作品标签 |
|  | 4 | 提交上传请求到服务器 |
|  | Postcondition | 注册用户作品上传成功 |
|  |  |  |
| Bounded Alternative Flow | RFS | 4 |
|  | 1 | 服务器运行异常 |
|  | Postcondition | 服务器异常，无响应 |

表3.3.6 用户上传作品RUCM

用户在网页上登录之后，可以选择已创建的作品进行上传操作。并且可以在上传时添加作品标签，上传之后，其他用户可以看到这个作品。

### 3.3.7 用户评价作品

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 评价作品 | |
| Brief Description | 注册用户网站上发布的作品 | |
| Precondition | 注册用户已登陆 | |
| Primary Actor | 注册用户 | |
| Secondary Actors | 浏览器 | |
| Dependency | 无 | |
| Generalization | 无 | |
|  |  | |
| Basic Flow | Steps | |
|  | 1 | 进入网页浏览已发布的作品 |
|  | 2 | 点击作品请求查看作品详情 |
|  | 3 | 服务器返回作品详情 |
|  | 4 | 用户编写用户评论 |
|  | 5 | 用户提交用户评论到服务器 |
|  | Postcondition | 用户评论成功 |
|  |  |  |
| Bounded Alternative Flow | RFS | 5 |
|  | 1 | 服务器运行异常 |
|  | Postcondition | 服务器异常，无响应 |

表3.3.7 用户评价作品RUCM

注册用户可以在登录后浏览平台上其他用户发表的作品，并可以通过点击作品请求查看作品的详情，

在作品的详情页，用户可以编写对该作品的评价，并提交到服务器端。作品的所有用户的评价可以展示在前端。

### 3.3.8 用户创建风格

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 创建新风格 | |
| Brief Description | 用户或管理员可以上传风格文件，以创建新的风格 | |
| Precondition | 用户角色登录 | |
| Primary Actor | 用户、管理员 | |
| Secondary Actors |  | |
| Dependency | 无 | |
| Generalization | 无 | |
| Basic Flow | **Steps** | |
| **1** | 用户选择“上传风格文件” |
| **2** | 在文件系统中选择需要的文件 |
| **3** | 系统VALIDATES THEN 确定训练风格 |
| **4** | 用户输入风格名称 |
| **5** | 系统创建新的风格 |
| **PostCondition** | 提示已创建完成 |
| Specific Alternative Flow | **RFS** | **3** |
| **1** | 上传文件不满足要求 |
| **PostCondition** | 不完成创建风格，返回文件上传 |
| Specific Alternative Flow | **RFS** | **4** |
| **1** | 风格已经存在 |
| **PostCondition** | 返回上一级 |
| Specific Alternative Flow | **RFS** | **4** |
| **1** | 用户选择返回 |
| **PostCondition** | 返回上传风格文件界面 |
| Global Alternative Flow | 用户选择退出系统 | |
| **1** | 退出系统 |
| **2** | **ABORT** |
| **PostCondition** | 系统重新回到登录界面 |

表3.3.8 用户创建风格RUCM

用户在使用该系统的时候，可以自己上传新的风格文件，以此采用自定义的风格。在选择“上传风格文件”后，用户需要从文件系统选择符合格式的风格文件进行上传，并确定风格的名称，在此之后系统便会创建新的风格。若是上传的风格文件不符合规范或是名称重复，系统会返回上一级。

### 3.3.9 用户修改已有风格

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 修改已有风格 | |
| Brief Description | 用户或管理员可以修改已存在风格 | |
| Precondition | 用户角色登录，已存在上传风格 | |
| Primary Actor | 用户、管理员 | |
| Secondary Actors |  | |
| Dependency | Use Case 查看风格 | |
| Generalization | 无 | |
| Basic Flow | **Steps** | |
| **1** | 用户选择“修改风格” |
| **2** | 选择自己需要修改的风格 |
| **3** | 系统VALIDATES THEN 确定需要修改的风格 |
| **4** | 选择需要修改的部分 |
| **5** | 系统修改完成风格 |
| **PostCondition** | 提示已修改完成 |
| Specific Alternative Flow | **RFS** | **3** |
| **1** | 需要修改的风格创建者不是用户 |
| **PostCondition** | 返回选择功能 |
| Specific Alternative Flow | **RFS** | **4** |
| **1** | 用户选择“更改名称” |
| **PostCondition** | 返回上一级 |
| Specific Alternative Flow | **RFS** | **4** |
| **1** | 用户选择“更改名称” |
| **2** | 用户输入新的风格名称 |
| **3** | 系统VALIDATES THEN完成名称修改 |
| **PostCondition** | 完成修改名称 |
| Specific Alternative Flow | **RFS** | **4** |
| **1** | 用户选择“删除名称” |
| **2** | 系统VALIDATES THEN完成删除 |
| **PostCondition** | 完成删除风格 |
| Global Alternative Flow | 用户选择退出系统 | |
| **1** | 退出系统 |
| **2** | **ABORT** |
| **PostCondition** | 系统重新回到登录界面 |

表3.3.9 用户修改已有风格RUCM

用户可以修改自己创建的风格，对其进行该名称或是删除。用户首先在界面选择“修改风格”，之后需要选择自己需要修改的风格，并决定需要对其进行改名或是删除操作，在完成操作后，系统便会对选定的风格进行修改。若是用户选定的风格无法满足操作需求，或是输入的名称重复，系统便会返回上一级。

### 3.3.10 后端系统请求运行算法

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 请求运行算法 | |
| Brief Description | 后端系统向算法模块传输输入数据，请求运行模型 | |
| Precondition | 用户角色登录 | |
| Primary Actor | 后端系统 | |
| Secondary Actors | 算法模块 | |
| Dependency | USE CASE 创建新作品 | |
| Generalization | 无 | |
| Basic Flow | **Steps** | |
| **1** | 后端向算法模块发送运行请求 |
| **2** | 算法模块VALIDATES THEN确认 |
| **3** | 后端向算法模块发送算法输入数据 |
| **4** | 算法模块VALIDATES THEN 开始运行 |
| **PostCondition** | 算法开始运行 |
| Specific Alternative Flow | **RFS** | **2** |
| **1** | 算法模块无法满足运行请求 |
| **2** | 算法模块返回后端通知信息 |
| **PostCondition** | 提示暂时无法运行 |
| Specific Alternative Flow | **RFS** | **4** |
| **1** | 发送数据数据不满足要求 |
| **2** | 算法模块返回后端通知数据不满足要求信息 |
| **PostCondition** | 提示输入数据不满足要求 |
| Global Alternative Flow | 中断请求过程 | |
| **1** | 退出系统 |
| **2** | **ABORT** |
| **PostCondition** | 系统回到请求前状态 |

表3.3.10 后端系统请求运行算法RUCM

在后端系统接收到用户的“创建作品”请求后，便会向算法模块发送运算请求，以运行算法模型。后端系统首先向算法模块发送运行请求和运行信息，在算法模块验证并回复后，后端系统向算法模块发送算法的输入数据，并开始算法运行。若算法模块获得的信息存在错误，或是输入数据存在异常，算法模块将向后端系统发送异常信息并返回请求前状态，以此通知用户。

### 3.3.11 算法模块调度运行队列

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 调度运行队列 | |
| Brief Description | 算法模块调度自身运行顺序 | |
| Precondition | 后端发送运行请求 | |
| Primary Actor | 算法模块 | |
| Secondary Actors | 无 | |
| Dependency | USE CASE 请求运行算法 | |
| Generalization | 无 | |
| Basic Flow | **Steps** | |
| **1** | 算法模块接收后端运行需求与输入数据 |
| **2** | 算法模块根据运行请求类型调度自身运行顺序 |
| **4** | 算法模块根据确定的运行顺序进行运行 |
| **PostCondition** | 算法开始运行 |

表3.3.11 算法模块调度运行队列RUCM

在算法模块得到后端的运行请求后，需要对自身的运行顺序队列进行调度。算法模块首先得到运行的状态信息和输入数据，之后根据请求的类型对自身运行队列进行调整，最后根据调度完成的顺序运行。

### 3.3.12 后端系统请求预期运行时间

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 请求运行时间 | |
| Brief Description | 后端系统向算法模块请求运行时间 | |
| Precondition | 开始算法运行 | |
| Primary Actor | 后端系统 | |
| Secondary Actors | 算法模块 | |
| Dependency | USE CASE 请求运行算法 | |
| Generalization | 无 | |
| Basic Flow | **Steps** | |
| **1** | 后端向算法模块发送运行时间请求 |
| **2** | 算法模块VALIDATES THEN确认请求 |
| **3** | 算法模块计算获得运行时间 |
| **4** | 算法模块将运算时间发送至后端系统 |
| **PostCondition** | 获得运行时间 |
| Specific Alternative Flow | **RFS** | **2** |
| **1** | 后端发送的运行请求不存在 |
| **2** | 算法模块通知后端系统异常信息 |
| **PostCondition** | 后端获得异常信息提示暂时无法运行 |
| Specific Alternative Flow | **RFS** | **3** |
| **1** | 无法计算获得运行时间 |
| **2** | 算法模块通知后端系统运行时间异常 |
| **PostCondition** | 后端获得异常信息 |
| Global Alternative Flow | 中断请求过程 | |
| **1** | 退出系统 |
| **2** | **ABORT** |
| **PostCondition** | 系统回到请求前状态 |

表3.3.12 后端系统请求预期运行时间RUCM

在用户需要得知自己的作品何时可以完成时，后端系统将向算法模块请求预期运行时间。后端系统首先向算法模块发送运行时间请求，在算法模块获得请求后，将根据自身的队列情况计算得到需要的大致运行时间，并向后端系统返回结果。若后端系统发送的运行请求不存在，或是计算出现问题，算法模块将向后端系统发送异常信息，以此进行提示。

### 3.3.13 算法模块通知运行结果

|  |  |  |
| --- | --- | --- |
| Use Case Specification | | |
| Use Case Name | 通知运行结果 | |
| Brief Description | 算法模块向后端系统发送运行结果 | |
| Precondition | 算法运行完成 | |
| Primary Actor | 算法模块 | |
| Secondary Actors | 后端系统 | |
| Dependency | USE CASE 请求运行算法 | |
| Generalization | 无 | |
| Basic Flow | **Steps** | |
| **1** | 算法模块完成运行 |
| **2** | 算法模块向后端系统发送运行结果 |
| **3** | 后端系统VALIDATES THEN获得运行结果 |
| **4** | 后端系统根据用户邮箱向用户发送邮件 |
| **PostCondition** | 完成算法运行完成通知 |
| Specific Alternative Flow | **RFS** | **3** |
| **1** | 算法运行结果异常 |
| **2** | 后端系统返回通知，重新开始算法运行 |
| **PostCondition** | 重新开始算法运行 |
| Specific Alternative Flow | **RFS** | **4** |
| **1** | 邮件发送失败 |
| **PostCondition** | 重新发送邮件 |
| Global Alternative Flow | 中断请求过程 | |
| **1** | 退出系统 |
| **2** | **ABORT** |
| **PostCondition** | 系统回到请求前状态 |

表3.3.13 算法模块通知运行结果RUCM

在算法模块完成运行请求后，将向后端系统发送通知，并返回运行结果。在后端系统获得到运行结果后，将会根据用户注册邮箱信息，向用户发送邮件提示。若后端系统获得的运行结果存在异常，将重新向算法模块发送运行请求；若发送邮件失败，将重新发送邮件。

# 四、运行环境规定

## 4.1 设备

　　操作系统系统：Linux、Mac OS X、Windows、Android、IOS

　　内存：256MB以上

硬盘：10M以上

用于开发基于flask框架的PC机或网络服务器

## 4.2 支持软件

　　Python3或2.7，基于Windows、Linux、Mac OS等平台

## 4.3 接口

### 4.3.1 硬件接口

　　无

### 4.3.2 软件接口

　　Python类库，浏览器Console控制台

### 4.3.3 通信接口

　　HTTP协议、TCP/IP协议、HTTPS协议

### 4.3.4 用户接口

IE浏览器（Internet explorer），Safari浏览器，Firefox浏览器，Opera浏览器，Chrome浏览器，360浏览器等。