

**更改履历**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 更改时间 | 更改人 | 更改章节 | 状态 | 更改描述 |
| 1 | V1.0 | 2020-6-29 | 全组成员 |  | 新建 |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

状态：新建、增加、修改、删除。

目 录

[1 引言 3](#_Toc225322385)

[1.1 目的 3](#_Toc225322386)

[1.2 背景 3](#_Toc225322387)

[1.3 参考资料 3](#_Toc225322388)

[1.4 术语 3](#_Toc225322389)

[2 任务概述 目标 3](#_Toc225322390)

[3 需求规定 3](#_Toc225322391)

[3.1 一般性需求 3](#_Toc225322392)

[3.2 功能性需求 3](#_Toc225322393)

[3.3 从本节开始为具体功能点需求 3](#_Toc225322394)

[需求描述 3](#_Toc225322395)

[业务流程描述](#_Toc225322396) *[此处画出该功能的业务流程图](#_Toc225322396)* [4](#_Toc225322396)

[数据描述 4](#_Toc225322397)

[3.4 系统安全性的要求 4](#_Toc225322398)

[4 运行环境规定 4](#_Toc225322399)

[4.1 运行环境 4](#_Toc225322400)

[4.2 接口 4](#_Toc225322401)

[5 遗留问题 4](#_Toc225322402)

[6 项目非技术需求 4](#_Toc225322403)

# 引言

## 目的

5人团队在15天内开发出一个个人空间系统。该用户可以为用户提供登录注册，博客管理的基本功能，并提供数据分析方面的拓展功能如数据推送等。

为明确软件，明细项目的数据流向和数据结构，为设计阶段打下坚实基础；确定系统功能，设计时应遵守的原则与约束撰写需求规格说明。

本需求规格说明书是为了开发个人空间系统而编写，主要面向系统分析员、程序员、测试员、部署程序员和最终用户。

本需求规格说明书作为整个软件开发依据，对以后阶段的工作起指导作用，也是系统验收的依据。

## 背景

所开发的系统名称：个人空间系统

开发者：五人组成的取经小队

目标用户：blog爱好者

本文档面向对象：

1. 项目经理：据此文档了解产品预期功能，并依此进行系统设计和项目管理。
2. 设计员：对需求进行分析，了解系统功能，据此设计系统，并完成数据库设计
3. 程序员：了解系统功能进行开发，并编写用户手册
4. 用户：了解产品的功能和性能

## 参考资料

Bootstrap-vue：https://bootstrap-vue.org/docs

Vue-cli：<https://cli.vuejs.org/zh/guide/>

Vue-axios - npm：<https://www.npmjs.com/package/vue-axios>

SM.MS v2 API Docs：https://doc.sm.ms/

《Spring IN ACTION (4th)》 --Craig Walls

## 术语

# 任务概述

目标：5人团队在15天内开发出一个个人空间系统。该用户可以为用户提供登录注册，博客管理的基本功能，并提供数据分析方面的拓展功能如数据推送等。

# 需求规定

## 一般性需求

在系统工程及需求工程中，一般性需求是指依一些条件判断系统运作情形或其特性，而不是针对系统特定行为的需求。本项目的一般性需求主要是体现在易用性、软件可测试性、可维护性、可扩展性等和软件系统结构及开发过程有关的品质。

易用性：个人空间系统的界面美观、整洁、大方，在保证功能完整性和丰富性的前提下方便用户的使用，操作简便且易于上手。用户只需要通过注册便可以享受到本项目带来的快捷和高效，很好地体现出该软件项目的易用性。

软件可测试性：我们项目采用前后端完全分离的技术和思想，在编写代码的时候可以分别对前后端的部分分别进行模块化的测试，以功能模块为单位，将大的问题转化为小的问题去测试，大大提高了测试的效率。同时我们也有专门进行测试的小组成员，通过第三方的测试也可以提高测试的稳定性。

软件可维护性：在编写代码的时候我们规定了所有人都要加入注释，并提交自己的功能部分的文档说明。这样在后续的开发中，方便了我们对于项目本身代码的修改，同时更方便了组内成员的学习交流。整体代码清晰明了，有着很好的可维护性。

软件可扩展性：对于每一个需求或功能来说，我们都预留出很多软件接口，这些接口经过测试后都能够进行正常的使用，同时注明接口的具体说明，便于后续对项目的需求或功能扩展。

## 功能性需求

对于项目中具体涉及到的需求和功能，我们在项目开发的前期也进行了充分的讨论，目的就是为了在后续项目开发中减少对于需求分析的一个修改和变更，这样也是保证我们组内效率一个非常重要的因素，那么下面我们将会简要说明一下项目的功能性需求分析。

登录注册部分：用户通过注册账号和登录后可以进入到个人空间系统进行一系列的操作，对于这部分功能来说首先是数据库要对用户的密码进行一个安全性的保障，在登录的时候采用单点登录可以让用户账号变得更加安全。

博客部分：因为个人空间涉及到个人博客的功能，所以我们的项目中会有一个Markdown文档编写的部分，用户编写好自己的文档后提交便可以进行自己文档文件的上传。同时，用户可以对自己上传的文档设置公开和私密，这样为用户自身提供了操作的多样性。我们也会显示用户当前的访问量等数据，这些数据能够很好地反映一篇博客的质量。同时增加的搜索功能，可以让用户快速找到自己喜欢的博客。

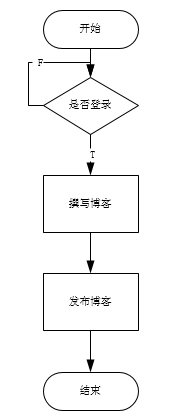
数据分析部分：除了这些需求外，我们还增添了一个对于数据分析和处理的功能。首先对想要获得的数据进行爬取，然后对爬取的数据进行构建模型后的训练和处理，并在页面上显示出来。这些信息能够提供对于用户的便利，同时也丰富了个人空间的功能。

## 从本节开始为具体功能点需求

## 需求描述

|  |  |
| --- | --- |
| 功 能 需 求 | |
| 功能名称 | 撰写博客 |
| 优先级 | 重要 |
| 业务背景 | 博客的出现才是近几年的事情，但是现在个人空间博客已经是一种娱乐潮流，甚至现在流行微型博客--微博。个人空间集娱乐，交友，游戏，甚至新闻，信息沟通及广告的功能于一身。 |
| 功能说明 | 用户可以在个人空间撰写博客并发布。 |
| 约束条件 | 需要登录 |
| 其他需求 | 无 |
| 裁剪说明 | 不可裁剪 |

业务流程描述



### 数据描述

Blog ID：String

PostTime：String

watchTimes：int

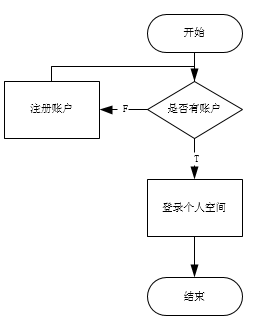
userID：String varchar()

username：String varchar()

tags：String

|  |  |
| --- | --- |
| 功 能 需 求 | |
| 功能名称 | 登录注册 |
| 优先级 | 重要 |
| 业务背景 |  |
| 功能说明 | 用户通过注册获得账户，通过登录账户进入个人空间。 |
| 约束条件 | 注册需要按要求填写注册信息；登录需要个人账户。 |
| 其他需求 | 无 |
| 裁剪说明 | 不可裁剪 |

业务流程描述



### 数据描述

userID：String varchar()

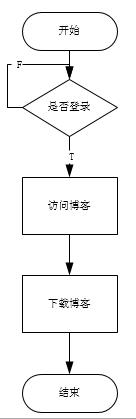
username：String varchar()

password：String varchar()

email：String varchar()

|  |  |
| --- | --- |
| 功 能 需 求 | |
| 功能名称 | 博客访问控制 |
| 优先级 | 比较重要 |
| 业务背景 | 通过博客使个人自由表达和出版，与他人深度交流沟通，知识过滤与积累 |
| 功能说明 | 用户可以访问自己和他人撰写的博客，可以下载博客。 |
| 约束条件 | 需要登录 |
| 其他需求 | 无 |
| 裁剪说明 | 不可裁剪 |

业务流程描述



### 数据描述

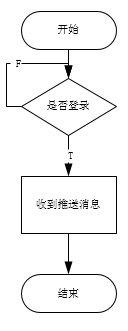
blog name:String

userID：String varchar()

username：String varchar()

|  |  |
| --- | --- |
| 功 能 需 求 | |
| 功能名称 | 消息推送 |
| 优先级 | 比较重要 |
| 业务背景 | 在网络信息碎片化的今天，各类信息越来越多，通过消息推送可以把用户关注的信息进行精准推送。 |
| 功能说明 | 用户登录个人空间可以收到系统推送的各类信息 |
| 约束条件 | 需要登录 |
| 其他需求 | 无 |
| 裁剪说明 | 不可裁剪 |

业务流程描述



## 3.4系统安全性的要求

### 3.4.1数据存储安全

数据存储上，数据的存储已经成为了人们日常生活与工作中必须要做的一项任务。随着人们对数据的依赖程度越来越严重，逐渐的开始对数据存储安全重视起来。当前，很多企业面临的挑战是如何找到安全与支出之间的平衡，当整个企业都在努力降低成本的时候，IT管理员要如何说服公司投资安全工具呢?人为错误通常是企业存储环境面临的最重要的存储安全错误，随着2009年网络犯罪和身份盗窃的不断增加，企业需要更加警惕防御抵制因为人为因素而导致的钓鱼攻击和社会工程攻击。

我们将数据安全主要分为两部分一是访问控制，二是文件加密。

同过对数据尽心访问控制可以有效的保证无关人员接触不到我们的核心数据，自然就能够减少很多数据安全的因素，另一方面，通过一些基本的机密方法，可以时得用户不能够通过一些推测就直接推断出文件再服务器的位置，这样可以防止一些不良人员想要通过越权直接访问。

### 3.4.2访问控制安全

访问控制是数据安全的一个基本组成部分，它规定了哪些人可以访问和使用服务器的信息与资源。通过身份验证和授权，访问控制策略可以确保用户的真实身份，并且拥有访问服务器数据的相应权限。访问控制还适用于限制对园区、建筑、房间和数据中心的物理访问。

访问控制上，我们将会结合java的jwt去进行用户鉴权，根据博客的可见性，来决定访问者能够访问到该页面。同时根据用户的鉴权，我们让只有正在登陆的该用户来修改他的博客，其他的用户不能够越权去修改

### 3.4.3网络传输安全

网络传输安全是决定系统安全的一大重要因素，对于用户来说，如果发送的的请求在发送途中就被劫持了，尤其是當发送的数据完全明文的时候，那么中间层可以轻易的将数据捕获到。从而获取用户的密码以及一些非常重要的信息。

关于网络传输安全，我们目前打算使用md5加密法，通过使用md5的不可逆性来进行保密，然后服务器端和客户端。同时如果后期时间充足，我们将打算使用非对称加密来进行数据加密。

### 3.4.4应用系统审计

审查应用程序的两个目的，一个是測试应用控制系统的符合性；二是通过检查程序运算

和逻辑的正确性以达到实质性瀏试目的。测试应用控制的符合性是指对嵌人应用程序中的控

制措施进行测试，看它们是否按设计要求在运行和起作用。

### 3.4.5系统约束

系统方面，我们要尽量保证系统是安全的，也就是说，一些不法的黑客不能够轻易的通过一些常规手段就拿到我们后端服务器的权限，要确保系统在绝大部分时候都是安全的，尽最大可能性保护用户的隐私与安全。不要乱开不需要的端口，一些关键的数据也不能乱放，关于用户的使用权限也要尽量保证严格规范。

### 3.4.6其他专门要求

故障处理是系统安全性的一个重要方面。如果在系统出现故障的时候不能及时处理，很可能影响系统的稳定性或是其他安全约束得不到保障。

关于系统故障处理，我们将会将系统故障分成不同等级，进行不同层次的处理，并保证做好日志记录，及时发现处理系统故障。

# 运行环境规定

## 4.1运行环境

### 4.1.1软件环境

使用 Linux 作為開發環境，Git 作為版本控制。

前端部份：npm、vue-cli

後端部份：JDK8、Tomcat、Python

使用開發工具：VScode、IntelliJ IDEA

讓我們舒適進行網頁開發。

### 4.1.2硬件环境



以上環境提供我們良好的開發環境，避免開發時 Memory 不足，在測試前端的階段也能夠開端口進行調試，前後端對接測試時，也可以前後端各開一個端口在本地測試。

## 4.2接口



1. 用户登录 post
2. 用户注册 post
3. 文件上传 post
4. 文档列表 get
5. 文档信息 get
6. 搜索接口 get
7. 数据监听 get
8. 消息推送 post

# 遗留问题

# 项目非技术需求

个人空间系统的界面美观、整洁、大方，在保证功能完整性和丰富性的前提下方便用户的使用，操作简便且易于上手。用户只需要通过注册便可以享受到本项目带来的快捷和高效，很好地体现出该软件项目的易用性。项目采用前后端完全分离的技术和思想，在编写代码的时候可以分别对前后端的部分分别进行模块化的测试，以功能模块为单位，将大的问题转化为小的问题去测试，大大提高了测试的效率。同时我们也有专门进行测试的小组成员，通过第三方的测试也可以提高测试的稳定性。

在编写代码的时候我们规定了所有人都要加入注释，并提交自己的功能部分的文档说明。这样在后续的开发中，方便了我们对于项目本身代码的修改，同时更方便了组内成员的学习交流。整体代码清晰明了，有着很好的可维护性。对于每一个需求或功能来说，我们都预留出很多软件接口，这些接口经过测试后都能够进行正常的使用，同时注明接口的具体说明，便于后续对项目的需求或功能扩展。

文档的完备也是本项目的的非技术需求之一。本项目文档记录了组内各成员的分工，同时对自己负责的功能模块进行详细说明，尤其是对于整体框架在文档上面会记载地非常清晰。