# **软件需求说明书**

## **一、引言**

### **1.1 项目名称及内涵**

***SOFT Workplace***

S(Sharing): 代表共享

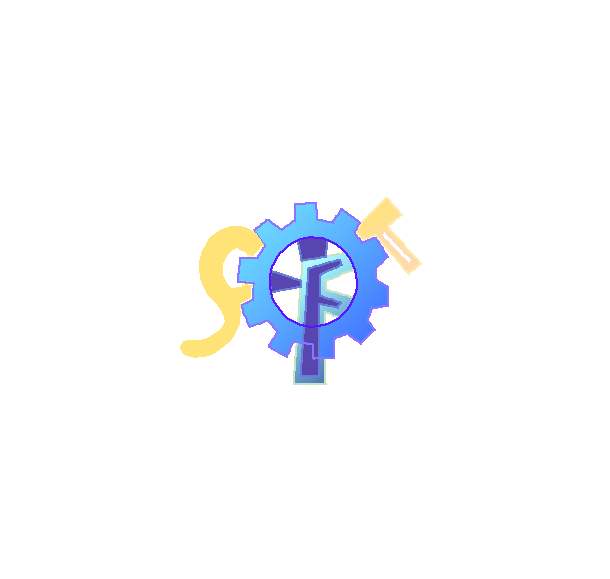
O(Open Source): 代表开源

F(Facilitate): 代表便利

T(Team): 代表团队

**SOFT 代表着我们设计的是一个让团队开发更为便利的开源合作平台。**

### **1.2 项目LOGO**



在LOGO中，S是扳手，O是齿轮，F是支架，T是锤子。SOFT组成了一个Workplace

### **1.3 背景**

当今社会的工作项目离不开团队协作开发，一个针对团队开发项目并且能够方便使用的平台可以说是令人目盼心思。然而，就市场上已有的平台，并不能很好的满足团队开发的要求，基本上都存在一些问题，例如国内服务器访问国外平台出现网络连接不畅，无法登陆；文件保存时间受限；平台占用大量内存；权限混乱，查看任务不便；使用界面新手不友好等。

### **1.4 项目简介**

SOFT Workplace主要面向团队项目开发者服务，协助用户设定项目开发目标，实时管理开发项目，附带教程，实现项目开发新手也能快速融入团队进行项目开发。用户在该平台上创建项目，通过链接在社交平台邀请其他成员；同团队成员可以上传、下载、在线编辑团队项目文件；项目管理者有最高权限并可以修改团队其他成员权限。

### **1.5 项目面向用户**

我们团队预期平台最终用户有一定编程水平，能够熟练使用markdown纯文本格式编写文档，能利用不同格式文件参与团队项目开发，能利用本平台进行项目管理。本平台的亮点在于组织管理系统，日程编写，github查找，格式优化以及自动打包功能，相较于已有平台，大大提升用户使用感。

### **1.6 小组成员简介**

组长：陈彧齐（需求分析，项目规划，项目分配）

组员：周源韬（后端） 杨啟超（产品测试，视频剪辑）

罗喆（前端） 黄涛（后端）

袁帅（软件架构） 胡文静（前端）

李玮杰（原型设计，用户界面UI） 朱亚城（前端）

赵睿言（前端）

## **二、需求分析**

### **2.1 目标**

**1. 制作一个让团队开发更为便利的开源合作平台。**

**2. 实现项目管理，团队管理，文件共享和编码优化功能**

**3. 将开发的成果进行小规模测试并投入使用**

### **2.2 用户的特点**

1. 年龄层面为20——40岁
2. 有一定的编程水平及项目合作经验
3. 能熟练使用markdown纯文本格式文档
4. 具有需要上传多种格式文件的需求
5. 具有文件管理的需求

### **2.3 假定与约束**

时间约束：11.20需要提交α版本软件

### **2.4 需求分析表**

项目与团队的建立功能

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 模块名称 | 功能描述 | 操作方法（实现功能） |
| 1 | 项目与团队的建立 | 用户可以根据自己的需要建立团队与项目，类似GitHub的organization和repository。 | 1. 团队与项目的命名与重命名 2. 通过QQ微信号邀请团队成员 |
| 2 | 对已有团队与项目的查看 | 用户可以在自己的个人页面寻找到加入或创建的团队项目 | 1. 根据需要查看团队项目 2. 通过搜索框查找项目 |

文件的共享上传，实时编辑

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 模块名称 | 功能描述 | 操作方法（实现功能） |
| 1 | 文件上传 | 用户可以根据自身需求上传**各类格式**文档，并创建文件夹保存文档 | 1. 用户新建文件夹 2. 用户上传各类文件 3. 文件的命名与重命名 |
| 2 | 文件共享 | 同一项目和团队里的成员能够看见，下载，观看文件内容(暂定只能看代码，文档，表格，PPT) | 1. 团队成员下载文件 2. 成员在线观看文件内容 |
| 3 | 文件实时编辑 | 用户可以像腾讯文档一样，合作实时编辑文件，系统会对编辑以后的内容实时保存 | 1. 实时编辑 2. 合作编辑 3. 自动保存 |

常用函数的整合，使用教程

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 模块名称 | 功能描述 | 操作方法（实现功能） |
| 1 | 整合函数 | 用户可以根据自身需求，将常用的函数整合上传，使用的时候参考vscode的自动补全功能，生成一个函数使用模板 | 1. 系统自带一部分常用函数 2. 用户上传常用函数 3. 调用函数时生成模板 |
| 2 | 使用教程 | 对于部分复杂的函数，用户可以自行制作函数的教程，在调用函数时可以参考 | 1. 用户上传教程 2. 使用函数时弹出教程 |

文件的检索

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 模块名称 | 功能描述 | 操作方法（实现功能） |
| 1 | 文件搜索 | 用户可以通过搜索框进行搜索，用于查找团队项目中的某一个文件 | 使用搜索框搜索 |

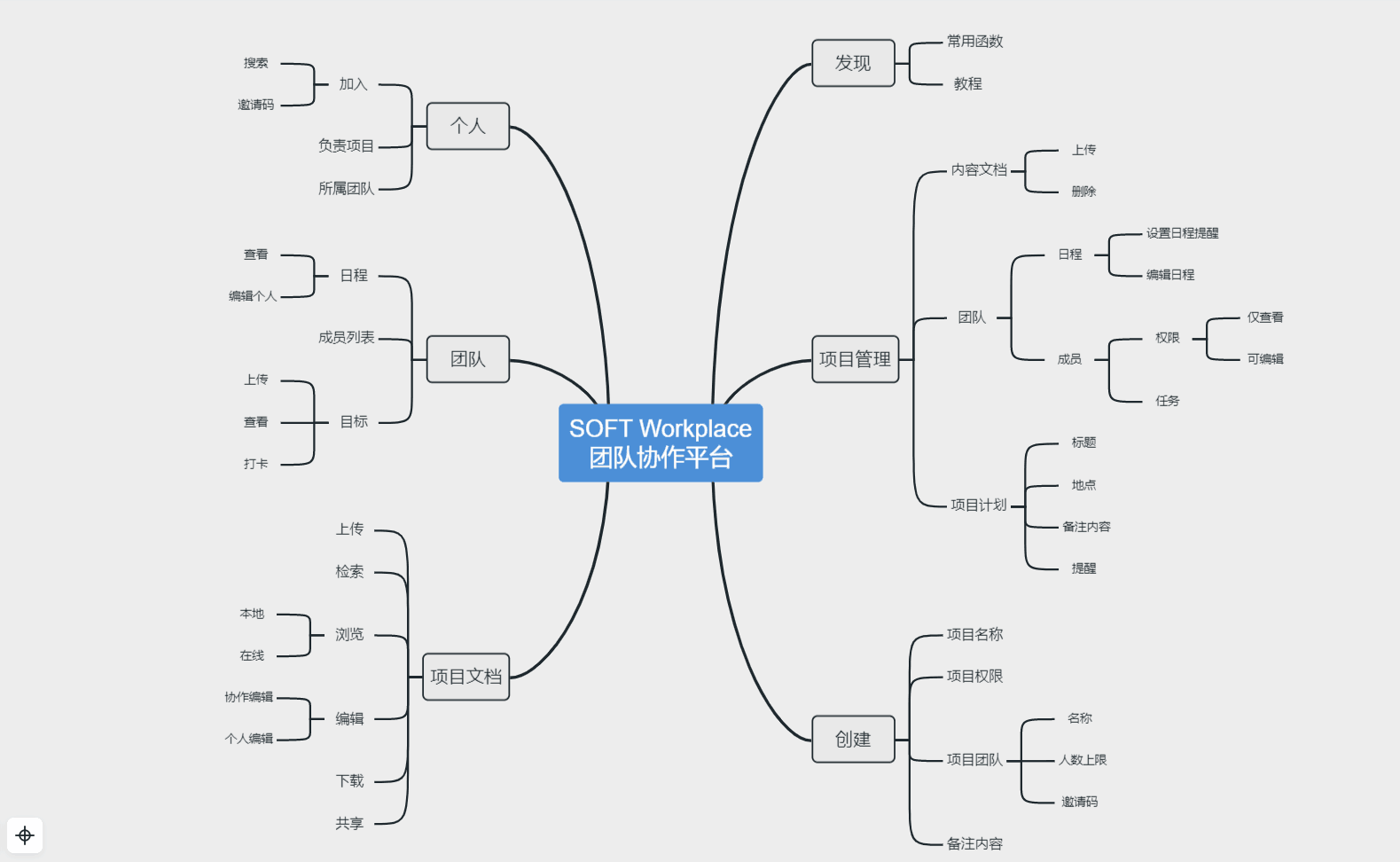
项目的管理

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 模块名称 | 功能描述 | 操作方法（实现功能） |
| 1 | 权限管理 | 项目创建者拥有最高权限，可以改变团队中其他成员的权限。权限设置为：1.只能观看 2.可以编辑修改 3.可以改变其他成员权限 | 待补充 |
| 2 | 日程编写 | 具有专门的模板进行日程的编写，小组成员可以看到整个团队里的人员日程，包括开会请假等相关事务。同时会设置一个机器人来提醒日程的安排。 | 1. 提供模板 2. 提供一个安排提醒的机器人 |
| 3 | 组织管理系统 | 每个团队成员都可以给自己定目标，自己的目标对齐总目标，用户可以根据自己的需求，定义需要的目标。值得一提的是，目标制定是完全透明的。 | 1. 提供目标模板 2. 目标可视化 |

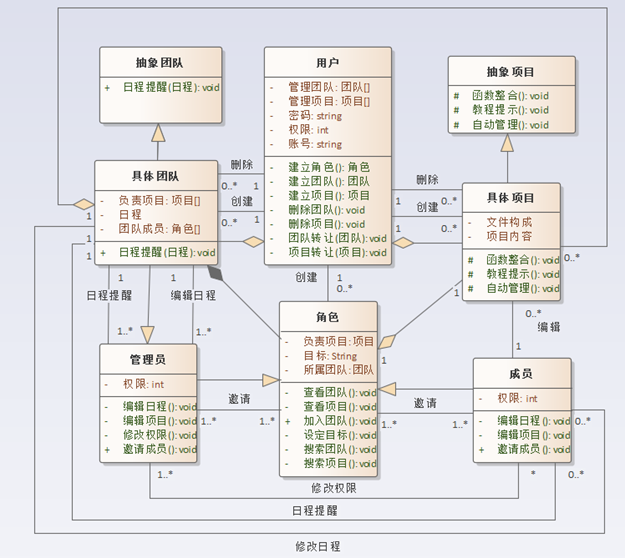
后端的实现

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 模块名称 | 功能描述 | 操作方法（实现功能） |
| 1 | 存储用户信息的数据库 | 后端具有一个数据库，用于存储用户创建的项目，文件以及收集的教程的信息 | 1. 创建一个管理信息的应用数据库 |
| 2 | 管理员界面 | 安排一个管理员页面，可以删除用户上传的违法违规文件 | 1. 创建一个管理员登录路口 2. 赋予管理员最高权限 3. 管理员可以查看加密的项目 |
| 3 | 接口定义 | 搭建给前端的接口Sever | 待补充 |

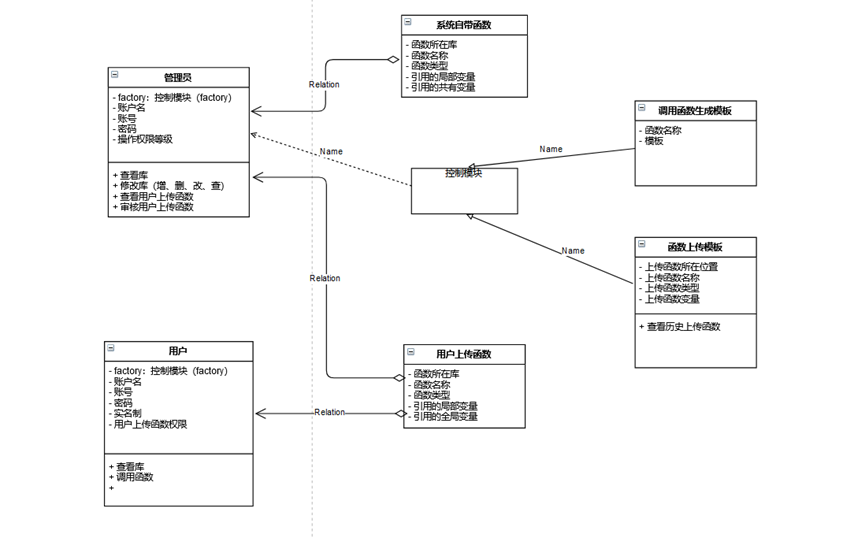
### **2.5 需求功能思维导图**



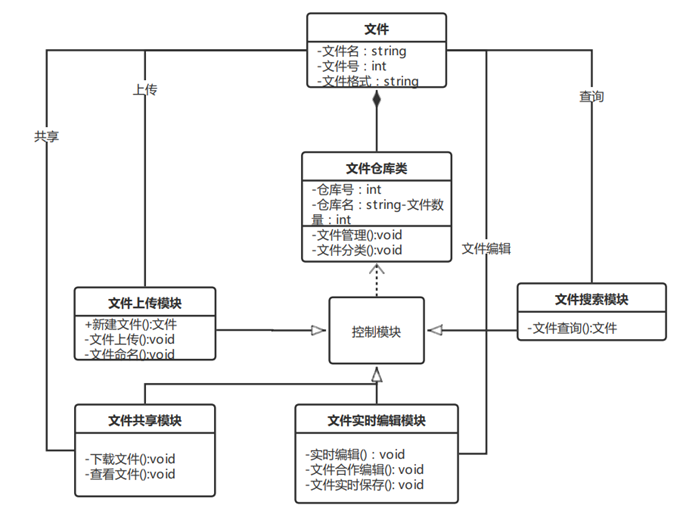
### 2.6 需求功能实现类图

****

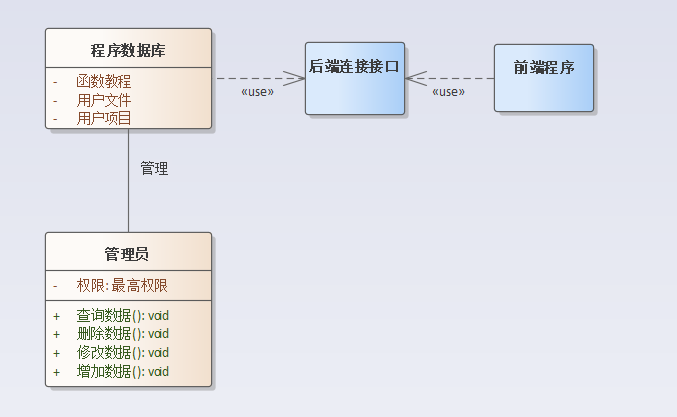
上图为项目与团队管理功能的类图



上图为编码优化功能的类图



上图为文件共享功能的类图

上图为后端实现的类图

## **三、需求规定**

### **3.1 项目验收标准**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 模块功能 | 使用操作 | 预期结果 |
| 1 | 项目与团队的建立 | 1. 分享邀请链接邀请成员加入  2. 通过邀请码系统邀请加入 | 1. 设计出一个较为完整的创建团队与项目的流程  2. 实现通过邀请码加入团队的功能 |
| 2 | 文件的共享上传，实时编辑 | 在主界面选择项目，并点击上传按钮进行文件更新 | 实现上传功能，并归档 |
| 3 | 常用函数的整合，使用教程 | 通过用户自主上传 | 1. 要求预置人工审核功能，能对用户上传内容进行过滤 2. 要求能实现已上传函数与教程历史记录的保存功能 3. 能较为稳定的运行 |
| 4 | 文件的检索 | 在各界面的搜索栏进行模糊搜索 | 能通过关键词搜索到相应结果 |
| 5 | 项目的管理 | 项目的创建者/创建者指定的管理员可以对项目内文件进行增删改操作 | 1. 允许创建者对其项目成员发放权限 2. 要求设置的权限范围不可逾越至项目外 |
| 6 | 后端的实现 | 1. 设计一个管理员模块，实现对数据的增删改查 | 1. 能用较为稳定的完成前后端数据的传输 2. 使管理员具有对数据库内数据增删改查的最高权限 3. 前后端的接口应能稳定使用 |

#### **3.1.2 界面效果的验收**

1. 界面元素的名称、标签应该易懂，用词准确，避免使用模楞两可的字眼，要与同一界面上的其他元素易于区分，能望文知意最好。理想的情况是用户不用查阅教程就能知道该界面的功能并进行相关的正确操作。
2. 软件界面在布局上应足够合理;在界面的视觉效果上应尽量简约，以保证软件的方便性，同时对加载的图片和皮肤的处理上也应显得大方整洁。

**3.1.3 界面效果的验收**

1. 功能上的稳定性:要在保证数据处理准确的同时确保多任务、数据定位和数据查找等功能运行正常且稳定。
2. 软件本身的稳定性:要确保软件不出现崩溃、卡死等情况;在对软件窗口进行处理时，软件界面不会出现断纹、控件错位等不统一的情况。

### **3.2 规范化制定**

#### **3.2.1 文档的规范化**

本次项目文档需要采用Markdown的自动排版格式书写，通过腾讯文档平台上传，上传者分享给其余组员时应根据实际情况打开观看与编辑权限。

#### **3.2.2 迭代/更新日志的规范化**

以天为单位进行撰写，如果当天有版本更新，写上当日最后一个版本的更新日志。需要标记版本号，写明更新内容（如有必要，附上更新该功能的原因）。

关于版本号：测试版以V0.X为准，如V0.3，V0.11，一次更新数字+1。如是通过测验的，可正常运行的版本，以V1.0开始，后续更新编号规则同上。

迭代更新表的格式如下：

|  |  |  |  |
| --- | --- | --- | --- |
| SOFT Workplace 迭代/更新日志 | | | |
| 版本号 | 更新日期 | 迭代内容 | 提交人员 |
| v0.3 | 11.1 | xxxxx | xxx,xxx |

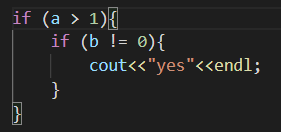
#### **3.2.3 编码的规范化**

为了让不同成员编写的代码便于互相查看，编码需要统一一个书写规范。允许有一定的差异，但大体上尽量保持一致，应具有可读性。超过10行的代码必须按一定规则遵守规范。

##### 3.2.3.1 缩进

次一级的代码需进行缩进，用tab而不用空格

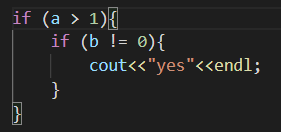
例如：



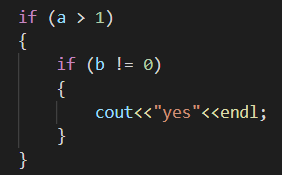
##### 3.2.3.2 括号对齐

后括号须与if/else/while等对齐

例如：



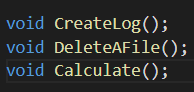
或



##### 3.2.3.3 函数命名

函数名和类名使用驼峰命名法，每个单词的首字母全部大写

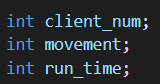
例如：



##### 3.2.3.4 常量命名

全部小写，单词之间采用下划线连接

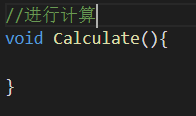
例如：



##### 3.2.3.5 注释

使用 // 进行注释 在函数声明前一行进行注释表明功能，重要功能与函数需要带注释

例如：



#### 3.2.3 接口的规范化

使用接口前，预先和相关负责人交流好命名内容。函数和变量的命名规则按3.2.1《编码的规范化》为准。详细接口名称需要先行交流，并做好记录。

#### 3.2.4 项目文件上传的规范化

保存文件时，选择UTF-8编码格式。上传文件时，上传至Github自己工作项目的文件夹内，并按照文件功能命名（如：随机数发生器V1.0）。发生更新时，在Readme中写上更新日志（修复了什么bug，或是增加了什么功能等），并增加版本号（如：随机数发生器V1.1）

### **3.3 小组工作成绩评定标准**

会议分：

我们小组采用有事开会，无事不开的原则，但是每次开会原则上都要参加。

具体计分细则如下：

（1）无故缺会并未汇报： 参与度0分

（2）有事请假，有汇报： 参与度1分，贡献度4分

（3）迟到5分钟以上： 参与度1分，贡献度5分

（4）迟到10分钟以上： 参与度1分，贡献度4分

（5）正常参加： 参与度1分，贡献度6分

（6）完成会议要求（或会议无要求） 贡献度4分

（7）部分完成会议要求 贡献度1~3分

（8）未完成会议要求 贡献度0分

工作分：

我们小组采用能者多劳，多劳多得的原则，会给予部分工作量大的小组成员加分补偿（可以超出10分上限，上不封顶）

具体计分细则如下：

（1）未在规定时间内完成分配的任务: 参与度0分

（2）完成的工作内容与实际项目不符： 参与度1分，贡献度0.5

（3）在工作截至24小时以上，告知无法完成分配的核心任务 参与度1分，贡献度5分

（4）在工作截止6~24小时中，告知无法完成分配的核心任务 参与度1分，贡献度4分

PS.上述两项需提供已完成的部分（编译不通的源代码也可以），工作量过少或无，不加分

（5）只完成了核心部分，其余部分未完成 参与度1分，贡献度7分

（6）按时完成所有分配的任务 参与度1分，贡献度10分

（7）在完成自己的任务后，分担别人的任务，每10行代码或者其他相应工作量的，一次加0.3分，上不封顶

（8）当某位成员被分配的工作量远超其他成员时，该成员在所得分的基础上额外再加2分（可超上限）