**PAAS平台总体设计说明文档**

小组成员：吴祥生 孙毅安 黄帆

张军 李萍 吴超

指导老师：丁波 闵建

15软件卓越班2018年暑期实训项目

2018年6月27日

**目录**

[1.引言 3](#_Toc19640_WPSOffice_Level1)

[1.1编写目的 3](#_Toc24106_WPSOffice_Level2)

[1.2项目背景 3](#_Toc2087_WPSOffice_Level2)

[1.3范围 3](#_Toc23646_WPSOffice_Level2)

[1.4 术语定义 4](#_Toc26792_WPSOffice_Level2)

[1.5 开发环境 4](#_Toc24174_WPSOffice_Level2)

[1.6参考资料 4](#_Toc19680_WPSOffice_Level2)

[2.总体结构设计 5](#_Toc2087_WPSOffice_Level1)

[2.1系统约束 5](#_Toc29487_WPSOffice_Level2)

[2.2系统物理结构设计 5](#_Toc18561_WPSOffice_Level2)

[2.3数据结构与程序的关系 5](#_Toc22588_WPSOffice_Level2)

[2.4系统前端界面设计 5](#_Toc167_WPSOffice_Level2)

[2.5系统管理员界面设计 5](#_Toc11398_WPSOffice_Level2)

[2.6系统后端设计 6](#_Toc30133_WPSOffice_Level2)

[2.7系统安全设计 6](#_Toc16181_WPSOffice_Level2)

[2.8系统总体架构 7](#_Toc4418_WPSOffice_Level2)

[3.数据库表设计 8](#_Toc23646_WPSOffice_Level1)

[4.通信协议及接口设计 8](#_Toc26792_WPSOffice_Level1)

[5.系统权限设计 9](#_Toc24174_WPSOffice_Level1)

[6.安全保密设计 10](#_Toc19680_WPSOffice_Level1)

# 1.引言

## 1.1编写目的

本文档为PAAS平台的总体设计说明文档，明确软件的需求，安排项目规划与进度、项目整体结构框架，为整个小组的设计人员、开发人员提供参考。

## 1.2项目背景

软件名称：PAAS平台

项目简介：本项目是基于Docker容器技术构建Paas平台服务，方便用户申请相关的容器服务（如java环境、mysql、nginx、tomcat等），并进行相应的信息配置（容器名、mysql密码设置等），最终运行Paas平台服务，进行测试。

## 1.3范围

本文作为总体设计方案，为以后的更加详细的设计提出了开发思路，具体内容包括：

（1）系统体系结构及各个模块的划分，各个模块的功能说明。

（2）数据库的总体设计思路，便于进行进一步的数据系统设计。

（3）通信协议设计和系统各个部分的接口设计方法

（4）安全保密设计

（5）系统权限设计

## 1.4 术语定义

PAAS平台：PAAS平台即(Platform-as-a-Service：平台即服务)，把应用服务的运行和开发环境作为一种服务提供的商业模式。

Docker：Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。

## 1.5 开发环境

操作系统：Windows、Centos7

前端开发工具：Axure、Webstorm

后端开发工具：IDEA

数据库：MySQL

## 1.6参考资料

（1）综合实训整体要求

（2）需求分析说明书

（3）详细设计说明书

（4）数据库结构设计说明书

（5）软件工程

# 2.总体结构设计

## 2.1系统约束

本系统遵循java和HTTP协议的要求；

本系统运行于Web端，兼容各个浏览器平台，数据库采用MySQL；

本系统界面友好，容易使用。

2.2系统物理结构设计

所有的数据都将保存在服务器的数据库里，数据库采用MySQl存取方法、存储结构都按MySQl数据库的方式来存储。在数据库里根据需要建立多张不同的表，存取数据直接对表进行操作.

2.3数据结构与程序的关系

程序需要的一切信息都从数据库中查询，存储的一切信息也从数据库中获得。 

## 2.4系统前端界面设计

框架:Vue Amaze UI

系统前端界面包括首页、登录与注册界面、主页、docker容器关联界面、账号信息界面、用户网盘界面、站内镜像信息界面、后台管理员界面、集群服务管理界面、远程终端登录操作界面等。

## 2.5系统管理员界面设计

数据可视化：Echars

框架：Vue Amaze UI

后台管理员的网站信息包括平台用户信息管理（如用户资源配置和权限审核管理）、docker容器的信息展示（如容器的运行状态、配置信息、以及所属用户信息等），并且需要实现对用户容器的创建、启动、删除等操作界面，以及对用户镜像管理、用户集群服务管理等操作，并对相关统计数据用Echars可视化出来显示。

## 2.6系统后端设计

后端框架：SpringBoot

数据库工具：MyBatis Plus

消息队列：Active MQ

缓存处理：Redis

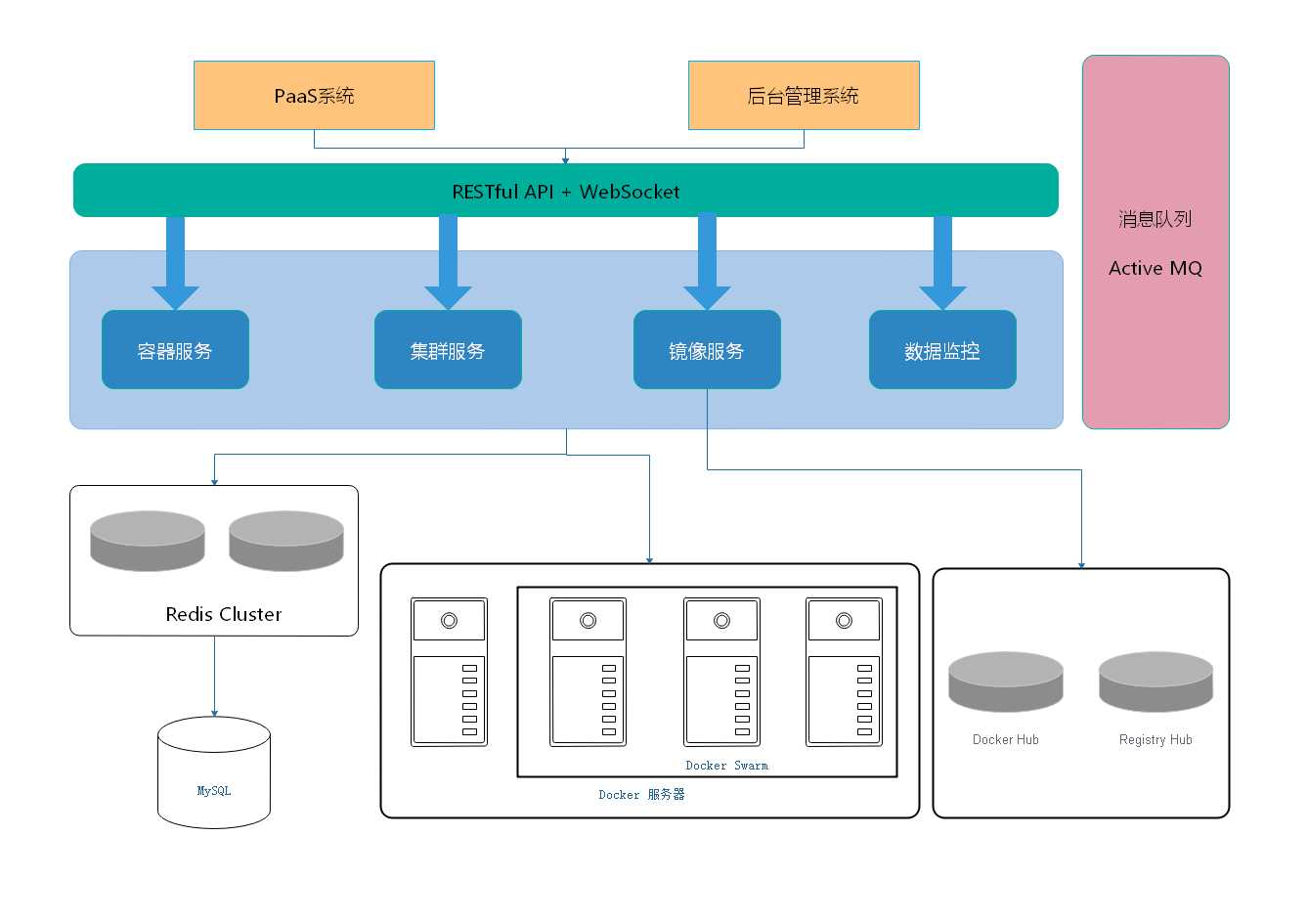
后端以SpringBoot 框架为基础，结合MyBatis Plus进行数据库管理，主要分为登录注册模块、镜像管理模块、容器管理模块、集群服务模块、网络管理模块等几个模块实现，其中，还运用了Redis缓存技术加快数据获取的速度，以及Active Mq实现通知管理功能。

## 2.7系统安全设计

安全框架：Spring Security

Spring Security是一个能够为基于Spring的企业应用系统提供声明式的安全访问控制解决方案的安全框架。它提供了一组可以在Spring应用上下文中配置的Bean，充分利用了Spring IoC，DI（控制反转Inversion of Control ,DI:Dependency Injection 依赖注入）和AOP（面向切面编程）功能，为应用系统提供声明式的安全访问控制功能，减少了为系统安全控制编写大量重复代码的工作。

## 2.8系统总体架构



本系统主要分为前台系统、Paas服务系统和后台管理系统；数据库使用mysql；镜像仓库分为Docker Hub和本平台搭建的Self Hub两类，用来方便用户镜像的拉取和上传管理；其中本系统采用了消息队列和缓存技术来优化性能。

本系统主要分登录/鉴权服务、容器服务、仓储服务和镜像服务四个模块。

登录/鉴权服务：实现用户注册、用户登录、用户权限鉴定；

容器服务：实现用户对java环境、mysql、nginx等容器服务的申请、设置容器名、mysql密码等相应容器信息配置和运行容器等操作；

仓储服务：实现对用户创建的容器及相关信息（如容器的运行转台、配置信息等）进行存储；

镜像服务：实现用户从远端拉取镜像以及对用户上传的镜像进行储管理服务。

# 数据库表设计

|  |  |
| --- | --- |
| 表名 | 功能说明 |
| 表sys\_login | 主要存储用户信息 |
| 表user\_project | 主要存储用户项目信息 |
| 表project\_log | 主要存储用户项目日志信息 |
| 表repository\_image | 主要存储仓储镜像信息 |
| 表sys\_image | 主要存储本地镜像信息 |
| 表sys\_network | 主要存储用户网络管理信息。 |
| 表conatiner\_network | 主要存储用户网络和容器关联信息。 |
| 表sys\_program | 主要存储系统服务信息 |
| 表sys\_volume | 主要存储数据卷信息 |
| 表user\_container | 主要存储用户容器信息 |
| 表user\_notice | 主要存储用户消息信息 |
| 表user\_notice\_desc | 主要存储用户消息详细信息 |
| 表user\_service | 主要存储用户集群服务信息 |
| 表sys\_log | 主要存储日志信息 |

# 

# 4.通信协议及接口设计

前端：Axious

SendGet 接收数据

|  |  |  |
| --- | --- | --- |
| 属性名称 | 属性类型 | 属性说明 |
| url | String | 后台接收数据的文件链接 |
| data | json | 传输的数据 |
| async | True/false | 传输状态 同步/异步 |
| successCallback | 回调函数 | 成功返回信息操作 |
| errorCallback | 回调函数 | 错误返回信息 |

SendPost 发送数据

|  |  |  |
| --- | --- | --- |
| 属性名称 | 属性类型 | 属性说明 |
| url | String | 后台接收数据的文件链接 |
| data | json | 传输的数据 |
| async | True/false | 传输状态 同步/异步 |
| successCallback | 回调函数 | 成功返回信息操作 |
| errorCallback | 回调函数 | 错误返回信息 |

后端

message类

|  |  |  |
| --- | --- | --- |
| 属性名称 | 属性类型 | 属性说明 |
| status | Boolean | 函数状态值 |
| info | String | 信息 |
| name | String | 名称 |
| userId | String | 用户Id |
| containerId | String | 容器Id |
| imageId | String | 镜像Id |
| email | String | 用户邮件 |
| user | User | 用户对象 |
| container | Container | 容器对象 |
| image | Image | 镜像对象 |
| containers | List<Container> | 容器对象集合 |
| images | List<Image> | 镜像对象集合 |
| users | List<User> | 用户对象集合 |
| userFiles | List<UserPan> | 用户云盘文件集合 |
| list | List<String> | 其他字符集合 |
| notifies | List<Notify> | 消息集合 |

接口设计具体接口设计详见接口设计文档。

# 5.系统权限设计

本系统的用户角色主要分为普通用户和后台管理员。

普通用户有拉取上传镜像、申请运行容器、运行测试docker服务、查看容器运行日志、建立私有、上传数据卷等权限；

后台管理员有用户权限审核管理、查看平台所有容器的信息、实现对用户容器及集群服务进行管理等权限。

# 6.安全保密设计

（1）注册用户  
 用户注册经过邮件验证后方可浏览全部网页，查看自己的个人信息，密码采用加密方式。

（2）非注册用户  
 非注册用户不能访问全部网页，可以浏览部分网页，没有属于个人的一个模板。

（3）系统管理员

系统管理员具有最高管理系权限，可以对系统资源进行管理分配，但无权对用户信息进行随意更改。