**基于Docker的分布式应用控制系统**

**需求规格说明书**

**目 录**

[一 引言 1](#_Toc9448)

[1.1 文档目的   1](#_Toc21324)

[1.2 项目背景 2](#_Toc14967)

[1.3 术语与约定 2](#_Toc10911)

[二 总体描述 2](#_Toc13084)

[2.1 项目目标 2](#_Toc28924)

[2.2 涉众分析 3](#_Toc19142)

[2.3 系统总体用例图 3](#_Toc30427)

[2.4 约束 3](#_Toc11054)

[三 需求详细描述 4](#_Toc30125)

[3.1 需求概述 4](#_Toc27705)

[3.2系统业务需求分析 4](#_Toc8495)

[3.3系统用户需求分析 4](#_Toc10349)

[3.4系统功能需求分析 5](#_Toc4232)

[3.4.1 主机管理 5](#_Toc20006)

[3.4.2 容器管理 8](#_Toc9514)

[3.4.3 镜像管理 12](#_Toc23412)

[3.4.4 系统日志 13](#_Toc1475)

[3.4.5 用户管理 13](#_Toc25021)

[3.5 系统非功能性需求 14](#_Toc13772)

# 一 引言

## 1.1 文档目的

详细需求由客户提出，编写本文档主要用于记录所有业务需求，以此作为开发人员开发依据，可减少开发过程中变动过大，需求不清晰等问题。主要供系统设计开发人员和需求审核人员浏览，以确保需求正确确立，系统被正确设计开发，供测试人员浏览，确保开发出的最终系统满足需求。

## 1.2 项目背景

一个产品在实际开发过程中，第一步是环境搭建与部署，在操作过程中非常容易产生错误，例如：在若干不同版本操作系统、不同配置的机器上搭建无差异化开发、测试环境难度高;产品研发环境与发布环境往往不同,正式上线通常会出现各种问题,产品发布难度比较大。项目扩大的过程中，参与人员也越来越多,走的走，来的来，每个人都要部署自己的开发环境，着实浪费时间。而且人为操作存在不可避免的失误,改正这些失误需要消耗更多的时间,造成更多的时间浪费,代价很大。而Dcoker的出现，则解决了这些让人头疼的问题的。

Docker的所有操作都只能在Linux 64位系统下进行，环境部署发布需要记忆大量命令行，对于不明白Docker运行原理的开发和运维还要花大把的时间弄明白，使用起来非常吃力，但如果存在一个Docker可视化管理工具，那就事半功倍了，无需linux基础，无需记忆大量命令行，只需简单的鼠标和键盘即可完成一项操作。

目前国内外已有Docker 可视化管理工具有Kubernetes、Dcokersh、DcokerUI、Panamax、Kitematic、Shipyard等，这些工具存在的同时也让Dcoker变得越来越强大。既然已经有了这些扩展工具，为什么还要自己开发一款呢？ 工具总会有各种各样的，但不一定有适合自己公司项目的可视化工具，同时由于Docker本身的安全系数不高，为了保证公司内部机密不外泄，为自己公司量身定做一款Docker开发工具也是有必要的。本项目开发主要是针对实习公司内部需要进行Docker管理工具开发，以便提高公司后期开发、测试、运维的工作效率。

## 1.3 术语与约定

|  |  |
| --- | --- |
| **名词** | **解释** |
| Dcoker | 容器引擎 |
| Container | 容器 |
| Image | 镜像 |
| Registry | 存放镜像的仓库 |
| Remote API | 远程操作Docker服务的接口 |

# 二 总体描述

## 2.1 项目目标

项目能够对容器进行批量处理，加速处理，简化Docker部署相关操作，节约开发运维人员项目部署时间。预计在5月中旬完成所有开发和测试。

## 2.2 涉众分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **大类** | **用户** | **代表人物** | **待解决的问题/对系统的期望** |
| 1 | 运维人员 | 公司内部运维 | 公司运维刘同志 | 1. 希望能够对容器进行以下操作：创建容器、启动容器、删除容器、查询容器、停止容器、终止容器、暂停容器、恢复暂停容器、重启容器等。 2. 对于镜像，至少能够实现拉取镜像、删除镜像、查询镜像列表等操作。 3. 记录系统操作日志，以便出现问题后更加快速地排查。 |
| 2 | 开发人员 | 公司内部所有开发 | 公司技术组长鄢同志 | 1. 人员流动大，希望简化开发环境配置，节约开发时间。 2. 希望对所有主机进行管理，包括创建主机、删除主机、查询主机、创建主机分组、更改主机风阻等。 |

## 2.3 系统总体用例图



## 2.4 约束

**法律法规**：项目仅限于内部使用，不可外传。

**硬件**：项目系统运行于PC机浏览器中。

**接口**：各模块过程之间采用函数调用、参数传递、返回值的方式进行消息传递。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在模块之间传递。

**安全**：系统信息绝对保密，不可向外界泄露。

# 三 需求详细描述

## 3.1 需求概述

基于Dcoker的分布式应用控制系统提供Dcoker Remote API相关的可视化操作，简化操作的同时，能够更直观地展示容器和镜像信息列表，方便了用户对容器和镜像的管理。

系统主要展示最新的主机列表信息、容器列表信息、镜像列表信息，登录用户在查看这些信息的同时可以对其进行其它操作，如：查看系统日志、连接主机、主机分组、拉取镜像、删除镜像、创建容器、删除容器、暂停容器、移除容器、重启容器、终止容器、查看容器日志和进程，且可对任意数量的容器进行批量操作。

## 3.2系统业务需求分析

使用Docker进行开发、测试、产品上线部署等过程的环境配置时，需要在linux 64位环境中操作镜像和容器，每次都要输入长长的并不容易记忆命令行，对于linux环境基础比较薄弱的同学来说，并不是很友好，熟练掌握linux的开发和测试也并不多。基于Dcoker的分布式应用控制系统技工的可视化操作，不仅操作简便，信息显示工整，便于查询，对于Dcoker初学者减少了Linux这一道门槛。

## 3.3系统用户需求分析

从系统用户角色来看，开发主要针对项目成员和项目管理员两大类型用户，其中项目成员可以是开发人员、测试人员和运维人员。



图 3-4-1 系统用例图

## 3.4系统功能需求分析

基于Dcoker的分布式应用控制系统主要包括主机管理、Container管理、Image管理以及系统日志等模块。



图3-5-1 系统功能模块图

### 3.4.1 主机管理

系统会显示主机列表以及主机分组信息，用户可以查看到某一主机具体在哪个分组里，主机分组是可以进行动态创建和删除的。具体功能描述如下：

1. 创建主机

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 创建主机 | 参与者 | 管理员 |
| 功能描述 | 用户填写好主机相关信息后，提交“创建”请求即可创建一个新主机。需要填写的主机信息包括：主机名、主机IP、主机端口号、主机分组（默认为未分组）。其中，主机名只能由数字、字母和下划线组成，主机IP和端口号符合对应的数字要求，主机分组不能输入只能选择。 | | |
| 前置条件 | 用户已登录系统 | | |
| 基本事件流 | 1. 用户输入主机信息，并提交“创建”请求 2. 系统验证输入数据是否有效 3. 系统验证通过，创建新主机 4. 系统返回“创建成功”提示信息 | | |
| 异常事件流 | 3b. 系统验证不通过，给出具体字段的错误提示信息 | | |
| 后置条件 | 系统记录操作日志，并显示最新的主机列表 | | |

1. 删除主机

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 删除主机 | 参与者 | 管理员 |
| 功能描述 | 用户可以删除不再适用的主机，一旦删除便不可恢复，主机列表中将不再显示，所以需要提醒用户慎重操作。一次只能删除一个主机，不能进行批量操作。 | | |
| 前置条件 | 用户已登录系统 | | |
| 基本事件流 | 1. 用户选择待删除的主机，提交“删除”请求 2. 系统删除用户选择的主机 3. 系统返回“删除成功”提示信息 | | |
| 异常事件流 | 3b. 若系统删除过程中出现未知错误，返回“删除失败”提示信息 | | |
| 后置条件 | 系统记录操作日志，并显示最新主机列表信息 | | |

1. 查看主机列表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 查看主机列表 | 参与者 | 管理员 |
| 功能描述 | 系统会显示主机列表信息，信息内容包括：主机名（ host name ）、主机IP （IP）、主机端口（Port）、主机状态（Status）、主机所在分组（Group）、运行中的容器数量（Running container）、停止运行的容器数量（Stoppped container）、镜像数量(Image number)。其中，主机状态只有“运行中”和“停止”。 | | |
| 前置条件 | 用户已登录系统 | | |
| 基本事件流 | 1. 用户请求显示主机列表 2. 系统显示主机列表信息 | | |
| 异常事件流 | 2b. 由于某些未知原因，主机列表信息拉取失败，返回“主机列表信息获取失败，请刷新重试！”提示信息。 | | |
| 后置条件 | 系统记录操作日志 | | |

5添加主机分组

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 添加主机分组 | 参与者 | 管理员 |
| 功能描述 | 主机分组主要用于对主机进行分组管理，方便主机查询和使用。添加主机分组只需填写分组名称，分组名称只能由数字、字母和下划线组成，且不可重复。默认有一个分组，组名为“未分组”。分组删除后，该分组的所有主机归为未分组。 | | |
| 前置条件 | 用户已登录系统 | | |
| 基本事件流 | 1. 用户输入组名，提交“添加”请求 2. 系统验证组名是否符合要求 3. 系统验证通过 4. 系统返回“分组添加成功”提示信息 | | |
| 异常事件流 | 3b. 系统验证失败，返回失败提示信息，如“组名只能由数字、字母和下划线组成”、“该分组已存在”。 | | |
| 后置条件 | 系统记录操作日志，并显示主机分组列表 | | |

6. 删除主机分组

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 删除主机分组 | 参与者 | 管理员 |
| 功能描述 | 如果主机分组命名不合适或者不再需要该分组，用户可删除该分组，分组一经删除不可恢复，所以要提醒用户慎重选择。主机分组删除后，该分组的所有主机归为未分组。 | | |
| 前置条件 | 用户已登录系统 | | |
| 基本事件流 | 1. 用户选择待删除的主机分组，发送“删除”请求 2. 系统删除用户选择的主机分组，并将该分组的所有主机分组置为“未分组” 3. 系统返回“删除成功”提示信息 | | |
| 异常事件流 | 2b. 若系统删除分组过程中出现未知错误，则返回提示信息“删除失败，请再次尝试！” | | |
| 后置条件 | 系统记录操作日志，并显示主机分组列表 | | |

7. 查看主机分组列表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 查看主机分组列表 | 参与者 | 管理员 |
| 功能描述 | 系统会显示所有分组，信息只包含分组名称 | | |
| 前置条件 | 用户已登录系统 | | |
| 基本事件流 | 1. 用户请求显示所有分组 2. 系统显示所有分组名称 | | |
| 异常事件流 | 2b. 系统显示出现未知错误，返回提示信息“显示分组列表错误，请刷新重试”。 | | |
| 后置条件 | 系统记录操作日志 | | |

8. 更改主机所在分组

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 更改主机所在分组 | 参与者 | 管理员 |
| 功能描述 | 由于主机用途可以随时更换，主机所在分组也需要跟随需求做出相应更改，故给出该功能。更换分组时只能选择不能输入。 | | |
| 前置条件 | 用户已登录系统 | | |
| 基本事件流 | 1. 用户找到待更改分组的主机，并选择目标分组 2. 系统根据用户选择将主机所在分组设置为目标分组 | | |
| 异常事件流 | 2b.系统设置分组时出现未知错误，返回提示信息“更改分组出现错误，请重试！” | | |
| 后置条件 | 系统记录操作日志，并显示最新主机列表（包含最新主机分组） | | |

### 3.4.2 容器管理

容器（container）是Docker技术的核心，所以容器操作比较多。用户除了可以搜索查看容器列表信息外，还可以对容器进行启动、重启、暂停、恢复暂停、重命名、终止、移除等操作。不同的操作，容器也会对应不同状态：停止、终止、暂停运行中。容器操作一般会生成记录，所以可以通过查看容器日志排查操作异常。容器运行过程中，查看容器进程也是必不可少的。



图3-5-1 容器状态图

1. 创建容器

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 创建容器 | 参与者 | 管理员 |
| 功能描述 | 创建容器，也叫部署容器。在创建容器之前，对应的镜像文件必须已经存在。创建容器时会同时启动容器。代码打包只能在已创建好的容器中运行。容器信息较多，每次创建无需输入每个信息，只需输入几个必要的信息即可，信息列表参见Remote API官网：  <https://docs.docker.com/engine/reference/api/docker_remote_api_v1.22/> | | |
| 前置条件 | 用户已登录系统，且主机已连接，对应镜像文件已创建完成 | | |
| 基本事件流 | 1. 用户输入容器基本信息，并发送“创建”容器请求 2. 系统验证输入信息 3. 系统验证通过 4. 返回“容器创建成功”提示信息 | | |
| 异常事件流 | 3b. 系统验证不通过，返回创建失败的原因 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 启动容器

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 启动容器（start） | 参与者 | 管理员 |
| 功能描述 | “停止”、“终止”和“运行中”状态的容器都能被启动。启动容器后，容器状态变为“运行中”。“运行中”状态使用“启动”操作不会重新计算运行时间。  “暂停”状态的容器不能被启动，只能使用“恢复暂停”操作才能继续运行容器。  可选择多个容器批量执行该操作。 | | |
| 前置条件 | 用户已登录系统，且主机已连接。容器状态不是“暂停”。 | | |
| 基本事件流 | 1. 用户选择要启动的容器，并发送“启动”容器请求 2. 系统启动相应容器 | | |
| 异常事件流 | 2b. 启动过程中出现未知错误，系统返回“容器启动错误，请重试！”。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 暂停容器

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 暂停容器（pause） | 参与者 | 管理员 |
| 功能描述 | 只有“运行中”状态的容器可以被暂停，被暂停后，也只能通过恢复暂停操作，让容器继续运行。可选择多个容器批量执行该操作。 | | |
| 前置条件 | 用户已登录系统，且主机已连接。容器状态为“运行中” | | |
| 基本事件流 | 1. 用户选择要暂停的容器，并发送“暂停”容器请求 2. 系统暂停相应容器 | | |
| 异常事件流 | 2b. 暂停过程中出现未知错误，系统返回“容器暂停错误，请重试！”。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 恢复暂停容器

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 恢复暂停容器（unpause） | 参与者 | 管理员 |
| 功能描述 | 只有“暂停”状态的容器能够被恢复暂停，可选择多个容器批量执行该操作。 | | |
| 前置条件 | 用户已登录系统，且主机已连接。容器状态为“暂停” | | |
| 基本事件流 | 1. 用户选择要恢复暂停的容器，并发送“恢复暂停”容器请求 2. 系统恢复暂停相应容器 | | |
| 异常事件流 | 2b. 恢复暂停过程中出现未知错误，系统返回“容器恢复暂停错误，请重试！”。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 停止容器

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 停止容器（Stop） | 参与者 | 管理员 |
| 功能描述 | “暂停”状态的容器不能被停止；“运行中”状态的容器可以被停止，停止容器后，容器状态变为“停止”；“停止”和“终止”状态的容器也可以使用停止操作，但容器状态保持不变。可选择多个容器批量执行该操作。 | | |
| 前置条件 | 用户已登录系统，且主机已连接。容器不能为“暂停状态”。 | | |
| 基本事件流 | 1. 用户选择要停止的容器，并发送“停止”容器请求 2. 系统停止相应容器 | | |
| 异常事件流 | 2b. 停止过程中出现未知错误，系统返回“容器停止错误，请重试！”。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 终止容器

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 终止容器（Kill） | 参与者 | 管理员 |
| 功能描述 | 只能终止运行状态的容器；“暂停”、“停止”、“终止状态”均不能进行终止容器操作。可选择多个容器批量执行该操作。 | | |
| 前置条件 | 用户已登录系统，且主机已连接。容器状态为“运行”。 | | |
| 基本事件流 | 1. 用户选择要终止的容器，并发送“终止”容器请求 2. 系统终止相应容器 | | |
| 异常事件流 | 2b. 终止过程中出现未知错误，系统返回“容器终止错误，请重试！”。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 重启容器

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 重启容器(Restart) | 参与者 | 管理员 |
| 功能描述 | 只有“暂停”状态的容器不能被重启；“运行中”、“停止”、“终止”状态的容器都可以被重启。容器被重启之后，运行时间重新开始计时。可选择多个容器批量执行该操作。 | | |
| 前置条件 | 用户已登录系统，且主机已连接。容器状态不为“暂停”。 | | |
| 基本事件流 | 1. 用户选择要重启的容器，并发送“重启”容器请求 2. 系统重启相应容器 | | |
| 异常事件流 | 2b. 重启过程中出现未知错误，系统返回“容器重启错误，请重试！”。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 移除容器

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 移除容器（Remove） | 参与者 | 管理员 |
| 功能描述 | 任何状态的容器都可以被移除。可选择多个容器批量执行该操作。 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户选择要移除的容器，并发送“移除”容器请求 2. 系统移除相应容器 | | |
| 异常事件流 | 2b. 移除过程中出现未知错误，系统返回“移除容器错误，请重试！”。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 查看容器日志

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 查看容器日志 | 参与者 | 管理员 |
| 功能描述 | 用户可以查看某一容器的运行日志，进行错误排查 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户选择某一容器，发送请求“显示”容器日志 2. 系统显示对应容器的日志列表 | | |
| 异常事件流 | 2b. 容器日志显示过程中出现未知错误，返回提示信息“容器日志获取失败，请重试”。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 查看容器进程

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 查看容器进程 | 参与者 | 管理员 |
| 功能描述 | 用户可以查看某一容器的进程，进程信息包括：UID/ PID/ PPID/ C/ STIME/ TTY/ TIME/ CMD。 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户选择目标容器，发送“显示”容器进程请求 2. 系统显示容器进程列表 | | |
| 异常事件流 | 2b. 系统显示容器进程过程中出现未知错误，返回提示“获取容器进程失败，请重试！” | | |
| 后置条件 | 系统记录操作日志 | | |

1. 查看容器列表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 查看容器列表 | 参与者 | 管理员 |
| 功能描述 | 显示的容器列表信息包括：容器编号（Id）、名字(Name)、状态(Status)、镜像（Image）、命令（Command）、端口号（Ports）、”IP”、“创建时间”。 可根据容器名称、容器状态和容器所在主机对容器列表进行筛选，默认显示所有容器。容器名称只能输入，容器状态主机只能选择不能输入。 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户输入筛选条件，发送“显示”容器列表请求  2. 系统验证输入信息  3. 系统验证通过  4. 系统显示容器列表信息 | | |
| 异常事件流 | 3b. 系统验证不通过，系统返回错误提示信息。 | | |
| 后置条件 | 系统记录操作日志 | | |

1. 容器重命名

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 容器重命名（Rename） | 参与者 | 管理员 |
| 功能描述 | 可以对任何状态的容器进行重命名，容器名只能由数字、字母、-、\_等组成。 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户选择想要重命名的容器，并输入新的容器名 2. 系统验证输入数据 3. 系统验证通过 4. 系统显示新的容器名 | | |
| 异常事件流 | 3b. 系统验证失败，返回错误提示信息。 | | |
| 后置条件 | 系统记录操作日志 | | |

### 3.4.3 镜像管理

用户可以从阿里云（或DCloud）拉取镜像，查看所有镜像列表，也可以对想要查看的镜像进行筛选，如果镜像不再被需要，也可删除该镜像。

1. 创建镜像

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 创建镜像 | 参与者 | 管理员 |
| 功能描述 | 镜像不像容器那样在Docker服务中进行创建，而是从远程镜像库中拉取需要的镜像放在Docker服务中。拉取镜像时只需输入镜像名称即可。镜像拉取需要较长时间，需提醒用户耐心等待。 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户输入镜像名称 2. 系统过滤镜像名称 3. 系统拉取镜像成功 4. 系统显示镜像列表 | | |
| 异常事件流 | 3b 系统拉取镜像失败，返回拉取失败原因，如：“该镜像不存在”。 | | |
| 后置条件 | 系统记录操作日志，并显示最新镜像列表 | | |

1. 删除镜像

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 删除镜像 | 参与者 | 管理员 |
| 功能描述 | 对于不再需要的镜像可以执行删除操作，删除操作不能批量执行，只能单个执行。 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户选择待删除的镜像，发送“删除”请求 2. 系统删除已选择的镜像 | | |
| 异常事件流 | 2b. 系统删除系统过程中出现未知错误，系统返回提示“删除镜像失败，请重试！” | | |
| 后置条件 | 系统记录操作日志，并显示最新镜像列表 | | |

1. 查看镜像列表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 查看镜像列表 | 参与者 | 管理员 |
| 功能描述 | 可根据镜像名称筛选镜像列表，列表信息包括：镜像名（Name）、镜像ID（ID）、镜像创建时间（Create Time）、镜像大小（Size）、镜像虚拟大小（Virtual Size）、镜像版本。 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户输入镜像名称（也可不输入），发送“显示”镜像列表请求 2. 系统过滤镜像名称 3. 系统显示镜像列表 | | |
| 异常事件流 | 3b. 系统显示过程中出现未知错误，系统返回提示“镜像列表拉取失败，请重试！” | | |
| 后置条件 | 系统记录操作日志 | | |

### 3.4.4 系统日志

系统会记录登录用户的一切操作，生成系统日志，运行环境出现问题时，方便排查以及责任追究。对于一定日期后的日志，管理员也可进行删除。

1. 查看系统日志

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 查看系统日志 | 参与者 | 管理员 |
| 功能描述 | 如果容器服务运行过程中出现问题，通过查看系统日志进行排查和责任追究。日志列表较多，分页显示，每页35条。系统日志可根据用户和时间筛选目标日志。用户名只能输入，日期范围只能选择 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户输入查询条件（用户名和时间），并发送“显示”系统日志列表请求 2. 系统根据验证查询条件 3. 系统验证通过 4. 系统分页显示系统日志 | | |
| 异常事件流 | 3b. 查询条件验证不通过，返回查询失败错误提示信息。 | | |
| 后置条件 | 无 | | |

### 3.4.5 用户管理

该模块只包含用户登录功能，所需账号和密码由系统自行分配，不能通过注册实现。

1. 用户登录

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 用户登录 | 参与者 | 游客 |
| 功能描述 | 用户输入用户名和密码通过验证后，即可进入系统。一旦进入系统，便拥有所有请求操作权限。用户名和密码只能是字母、数字和下划线组成。长度限制50。 | | |
| 前置条件 | 用户已登录系统，且主机已连接 | | |
| 基本事件流 | 1. 用户输入登录信息，发送“登录”请求 2. 系统验证输入信息 3. 系统验证通过 4. 系统开启 | | |
| 异常事件流 | 3b. 系统验证失败，返回失败提示信息，如：“用户名不存在”，“密码错误”。 | | |
| 后置条件 | 系统记录操作日志 | | |

## 3.5 系统非功能性需求

**1. 系统可靠性**

系统能否正常运行取决于系统的可靠性，数据库系统使用MySql。MySql 数据库是一款支持 ACID 即：事务 的原子性、一致性、独立性、持久性的 InnoDB（数据库引擎）的事务性数据库，采用该数据库保证了对数据进行操作的正确安全等要求。

1. **系统可扩展性**

系统开发中，结束后，都可能遇到需求变更或添加，如果系统拥有较强的扩展性，这些都不会出现很大的问题，或许，还能够很友好的解决。系统采用前后端完全分离的开发方式，任何一方的改变都将尽可能少的影响另外一方。系统整体采用MVC模式进行开发，对Controller 和 Model层进行了严格分类，可以很轻松地增删相应的功能点。

1. **系统可维护性**

系统开发完成后，维护成了日常主要的工作内容，所以系统的可维护性在开发时成了必不可少的要求。系统前后端的MVC模式可以很快定位错误出现的位置，并给出相应的解决方案。开发时必要的注释也会让代码更加容易阅读。

1. **系统安全性**

系统使用password hash对用户密码进行加密，密码只可生成和验证，不可解密，增强用户密码安全性。

系统会记录每一个用户的每一项操作日志，可以很快追溯事故原由以及相应的负责人。

1. **系统交互性**

系统前端开发要求至少兼容IE8+，采用渐进增强的思想进行开发，即保证低浏览器能够正常显示运行的情况下，提高高端浏览器的优美视觉感。