编号：



**毕业设计说明书**

题 目：基于二维码的生产

信息管理系统

学 院： 信息与通信学院

专 业： 导航工程

学生姓名： 张鹏帆

学 号： 1700200138

指导教师： 张红梅

职 称： （三号宋体居中）

**题目类型：**  理论研究 实验研究 工程设计 工程技术研究 软件开发

2020年 5 月 10 日

摘 要

本设计使用vue和node实现生产信息管理系统，采用前后端分离架构，使用接口进行前后端通讯，实现了账号管理系统，日志系统，生产任务管理系统。账号管理系统实现了账号的创建、删除、编辑、登录、退出登录以及权限等；日志系统实现了各个阶段埋点打日志，并写入硬盘，实时读取；生产任务管理系统实现了生产任务的创建，任务详情的查看，生产流程的变更，负责人的变更，关于此任务的相关动态，任务的嵌套，以及查看任务所需要的权限。任务还实现了生成二维码，扫描二维码可查看任务详情。

服务器采用linux系统，前后端独立部署，前端部署于nginx服务器，后端使用pm2持久化部署，使用nginx的反向代理进行接口的转发。

代码和版本管理使用github和git，采用github的actions和shell脚本进行持续集成与自动化部署。

关键字：生产信息管理；二维码；接口；前端；后端

**Abstract**

This design uses vue and node to implement a production information management system, adopts a front-end and back-end separation architecture, uses interfaces for front-end and back-end communication, and realizes an account management system, a log system, and a production task management system. The account management system realizes the creation, deletion, editing, login, logout and permissions of accounts; the log system realizes the logging of various stages, and writes it to the hard disk, and reads it in real time; the production task management system realizes the production task Creation, view of task details, change of production process, change of person in charge, related dynamics about this task, nesting of tasks, and permissions required for viewing tasks. The task also realizes the generation of a QR code. Scan the QR code to view the details of the task.

The server uses the Linux system, the front and back ends are independently deployed, the front end is deployed on the nginx server, the back end uses pm2 persistent deployment, and the nginx reverse proxy is used for interface forwarding.

Code and version management uses github and git, and uses github's actions and shell scripts for continuous integration and automated deployment.

**Key words：**Production information management; QR code; interface; front end; back end

引言

传统的生产信息都采用人工手动记录，繁琐而复杂，且不能分配权限，在任务多了之后很容易造成混乱。而随着互联网的发展，生产信息管理也逐渐互联网化，所有信息云端存储，不担心丢失损毁；可配置权限，不担心信息安全；流程信息可视化，不担心混乱。

随着生产任务越来越复杂，一件物品可能拆分为多个元器件多个生产流程。所以此系统支持任务拆分，单任务自定义生产流程等。

实际生产问题是非常复杂的，不同场景可能非常多，所以系统要做一定的抽象才能满足大部分场景，所以此系统将一条生产信息称为“任务”，由用户自定义去输入此任务的信息，比如任务名，描述，起止与截至时间，元器件数量，材料成本，备注，负责人，相关资料文件等。

一条生产任务肯定是由不同流程状态组成，称之为工作流，顺着流程做完此任务，则算完成此任务，实际情况中可能有异常流程，所以此系统也实现了自定义流程以及自定义跳转流程。大的流程上分为待办，进行中，已完成。在进行中有由用户自定义的单独的子流程，走完这些子流程方会进入已完成。

实际生产任务可能会不断变动，比如负责人的变动，或者更新了某些信息，或者添加了某些备注，此系统采用“动态”来标注这些信息的变动，做到了变动永久记录，实时追踪。

生产信息最终要的就是权限管理，此系统将账户管理与任务管理隔离开来，确定好以权限为标尺，为账户赋予不同的权限，将任务分配给不同的负责人，只有此任务的负责人或者高权限的管理人员才可查看此任务。具体实现上，由于不同人员拥有不同权限，所以将系统菜单也权限化。

在系统层面上，日志也是重要的组成部分，可以在软件运行的不同生命周期写入信息或标识至日志，系统管理员可查看日志，方便分析解决问题。

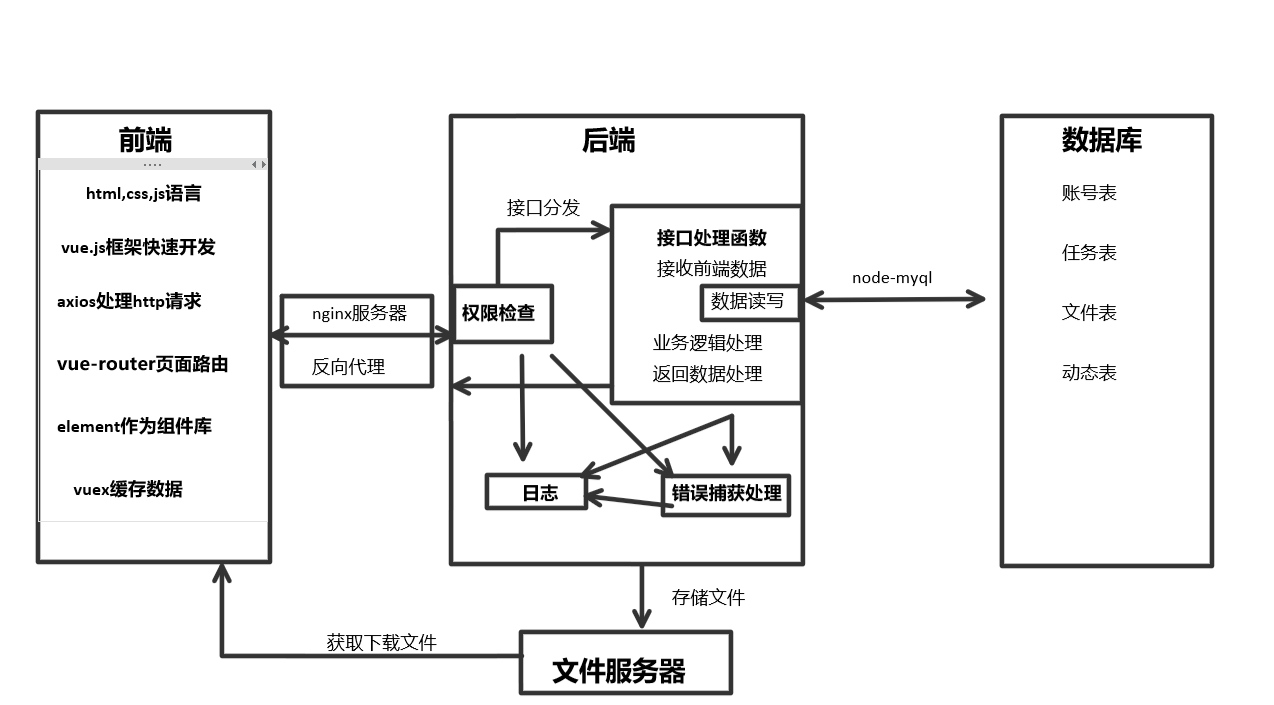
系统是逐步开发的，每次手动去打包部署都会显得很麻烦，所以此系统也通过Github action实现了持续集成与自动化部署。

# 系统的架构与设计

此系统采用前后端分离的方式同步开发，前后端通过接口进行通讯。前后端分离可以单独部署，单独调试开发，大大增加开发速度与系统稳定性。

前端采用html，less，js，vue，vue-router，axios，vuex，element-ui等技术栈。后端采用node,mysql原生开发，pm2持久化运行。前后端都部署于阿里云CentOS，使用nginx作为前端服务器，反向代理服务器以及文件服务器。使用mysql作为数据库存储数据。

系统架构图如下：



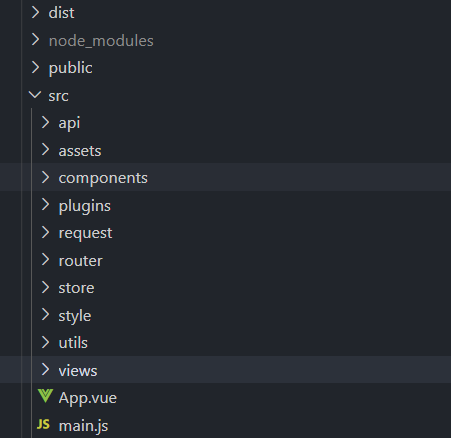
图表 1系统架构

## 前端系统设计

前端采用vue作为框架以达到快速开发，使用axios作为接口管理以及和后端通信，vue-router作为路由器控制页面跳转以及跳转传参，vuex来保存页面状态，缓存全局数据。Element-ui作为组件库，避免写繁琐的css和重复开发。

前端项目结构如下：dist目录为打包目录，编译出来的代码会在此目录；node\_modules为存放第三方插件以及安装的插件框架等目录；public存放index.html以及网站图标；

src为开发目录，大部分的开发都在此目录进行。api目录用来统一管理接口；assets存放字体，图片等静态文件；components存放全局的公共组件；plugins存放插件；request把axios进一步封装达到请求拦截，相应处理的目的；router用来管理项目页面之间的路由，以及组件与url之间的绑定和页面路由传参等；store用来缓存全局数据，避免重复请求；style为全局的样式，存放less等文件。Utils里面存放了封装好的工具函数等；views就是存放页面组件的目录了，所有显示在页面上的组件都存放在此。App.vue就是项目的主组件了，所有页面都会通过此组件加载；main.js是此项目的入口文件，打包，运行都是通过此文件开始。



图表 2前端目录

## 后端系统设计