**新疆大学本科毕业论文(设计)**



|  |  |
| --- | --- |
| **论文题目:** | **基于数字孪生的机房管理原型** |
|  | **系统设计与实现** |
| **学生姓名:** | **赵泽天** |
| **学 号:** | **20192501228** |
| **所属院系:** | **软件学院** |
| **专 业:** | **软件工程** |
| **班 级:** | **软件2019-3** |
| **指导老师:** | **冷洪勇** |
| **日 期:** | **2023年 4月 22日** |

声 明

本人郑重声明，本论文是在导师的指导下独立完成，除加注和致谢外，文中不包含他人所发表或撰写的成果。本人拥有自主知识产权，没有抄袭、剽窃他人成果，对于参考的文献已经加注并表示感谢。若有不实之处，本人愿意承担相关法律责任。

作者签名：

签字日期： 年 月 日

**新 疆 大 学**

**本科毕业论文（设计）任务书（2023届）**

学院：软件学院 班级： 软件19-3

姓名：赵泽天

毕业论文（设计）题目：基于数字孪生的机房管理原型系统设计与实现

毕业设计(论文)工作自2022 年 12 月 23 日起至 2023 年 5 月 23 日止

毕业设计(论文)题目的目的及意义

数据中心可视化管理平台作为新一代数据中心可视化管理平台，采用数字孪生技术，实现对机房的虚拟仿真。让管理人员可以清晰直观地掌握IT运营中的有效信息，实现透明化与可视化管理，进而有效提升资产管理与监控管理的效率，实现立体式、可视化的新一代机房运行管理。

毕业设计(论文)的主要工作任务

1、查阅、收集、整理毕业设计所需资料； 2、根据设计任务制定工作进度； 3、完成对数字孪生开发相关技术资料和交互可视化系统开发相关需求资料的查询和整理工作； 4、选择合适的软件开发框架，完成系统的需求分析； 5、完成系统的概要设计、详细设计，编码实现系统； 6、完成相关技术文档的整理及论文的撰写；

指 导教 师： Generated

教研室主任： 

学 生签 名： 

接受毕业论文(设计)任务日期： 2022 年 12 月 23 日

（注：本任务书由指导教师填写）

摘 要

基于数字孪生的机房管理原型系统，作为新一代数据中心可视化管理平台，采用数字孪生技术，实现对机房的虚拟仿真。让管理人员可以清晰直观地掌握IT运营中的有效信息，实现透明化与可视化管理，进而有效提升资产管理与监控管理的效率，实现立体式、可视化的新一代机房运行管理。

本系统采用前后端分离的设计模式，前端使用React框架搭配Ant Design组件库构建系统界面，以及WebGL第三方库three.js，后端使用SpringBoot开发框架搭配Mybatis-Plus开发符合restfull规范的数据接口，数据库使用Mysql，又加入了Nginx作为负载均衡服务，最后部署到AWS云平台上提供持久的可访问服务。本系统实现了登录注册模块、机房模拟及监控模块、机房状态管理模块、报修消息管理模块、用户管理模块以及个人资料模块，根据用户角色不同而展示不同的界面和权限，实现了管理员可以清晰直观监控机房状态、普通用户可以及时提交报修信息、管理员与普通用户权限分离的目标。

毕业设计系统实现的效果，如，实际测试和运行的情况，系统的性能如何，带来了哪些优势，应用价值如何，应用前景如何等（约70-170字）。

**关 键 词：**数字孪生；管理系统；SpringBoot；React；Three.js

ABSTRACT

The digital twin-based server room management prototype system, as a new generation data center visualization management platform, adopts digital twin technology to realize virtual simulation of the server room. It allows managers to clearly and intuitively grasp effective information in IT operations, realize transparent and visualized management, and then effectively improve the efficiency of asset management and monitoring management, and realize a three-dimensional and visualized new generation of server room operation management.

The system adopts the design pattern of front-end and back-end separation, the front-end uses React framework with Ant Design component library to build the system interface and WebGL third-party library three.js, the back-end uses SpringBoot development framework to develop the data interface in line with the restfull specification, the database uses Mysql, and adds Nginx as a load The backend uses SpringBoot development framework to develop data interfaces that conform to the restfull specification, the database uses Mysql, and Nginx is added as a load balancing service, and finally deployed to the AWS cloud platform to provide persistent accessible services. The system implements the login and registration module, the server room simulation and monitoring module, the server room status management module, the initiation and processing of repair messages module, the user management module, and the personal information module, displaying different interfaces and permissions according to the user roles, achieving the goal that the administrator can clearly and intuitively monitor the server room status, regular users can submit repair information in a timely manner, and the administrator and regular user permissions are separated.

**KEY WORDS:** Digital twins; Management System; SpringBoot; React; Three.js

目 录

[1 绪论 1](#_Toc133242246)

[1.1 研究背景及意义 1](#_Toc133242247)

[1.2 研究手段 1](#_Toc133242248)

[1.3 论文组织结构 2](#_Toc133242249)

[2 相关技术与工具概论 3](#_Toc133242250)

[2.1 相关技术介绍 3](#_Toc133242251)

[2.1.1 SpringBoot 3](#_Toc133242252)

[2.1.2 React 3](#_Toc133242253)

[2.1.3 Three.js 3](#_Toc133242254)

[2.1.4 MySql 4](#_Toc133242255)

[2.1.5 RESTful API 4](#_Toc133242256)

[2.1.6 Git 4](#_Toc133242257)

[2.2 相关工具介绍 4](#_Toc133242258)

[2.2.1 IntelliJ IDEA 4](#_Toc133242259)

[2.2.2 Visual Studio Code 4](#_Toc133242260)

[2.2.3 Postman API Platform 5](#_Toc133242261)

[2.2.4 GitHub 5](#_Toc133242262)

[2.3 本章小结 5](#_Toc133242263)

[3 系统需求分析 6](#_Toc133242264)

[3.1 系统业务流程分析 6](#_Toc133242265)

[3.2 系统功能需求分析 6](#_Toc133242266)

[3.2.1 登录注册模块 6](#_Toc133242267)

[3.2.2 机房模拟及监控模块需求分析 6](#_Toc133242268)

[3.2.3 机房状态管理模块 6](#_Toc133242269)

[3.2.4 报修消息管理模块 6](#_Toc133242270)

[3.2.5 用户管理模块 6](#_Toc133242271)

[3.2.6 个人资料模块 6](#_Toc133242272)

[3.3 系统非功能需求分析 6](#_Toc133242273)

[3.4 本章小结 6](#_Toc133242274)

[4 系统设计 7](#_Toc133242275)

[4.1 系统功能设计 7](#_Toc133242276)

[4.2 系统数据库设计 7](#_Toc133242277)

[4.2.1 E-R图设计 7](#_Toc133242278)

[4.2.2 数据库表设计 7](#_Toc133242279)

[4.3 本章小结 7](#_Toc133242280)

[5 系统开发与实现 8](#_Toc133242281)

[5.1 登录注册模块 8](#_Toc133242282)

[5.2 机房模拟及监控模块 8](#_Toc133242283)

[5.3 机房状态管理模块 8](#_Toc133242284)

[5.4 报修消息管理模块 8](#_Toc133242285)

[5.5 用户管理模块 8](#_Toc133242286)

[5.6 个人资料模块 8](#_Toc133242287)

[5.7 本章小结 8](#_Toc133242288)

[6 系统测试 9](#_Toc133242289)

[6.1 测试目标和原则 9](#_Toc133242290)

[6.2 系统测试环境 9](#_Toc133242291)

[6.3 系统功能测试 9](#_Toc133242292)

[6.4 测试结论 9](#_Toc133242293)

[6.4.1 本章小结 9](#_Toc133242294)

[7 总结与展望 10](#_Toc133242295)

[7.1 论文结论与总结 10](#_Toc133242296)

[7.2 未来研究展望 10](#_Toc133242297)

[参考文献 11](#_Toc133242298)

[致 谢 12](#_Toc133242299)

# 绪论

本章针对基于数字孪生的现代化的机房管理系统，研究了其背景和意义，分析了目前及以前的机房管理系统的发展状况，以服务器机房管理员的具体业务流程和工作内容为基础，分析了该系统设计时的研究手段，详细描述了论文的组织架构。

## 研究背景及意义

目前很多机房管理模式不能满足服务器机房的管理需要而暴露弊端，如机房服务器数量较大，管理接口不规范，造成了较大的管理和遇到某台服务器突发异常时的宕机服务器排查难度，以及服务器工作状态不正常并且消息更新不及时而造成主要业务受到影响的现象，机房管理人员的操作不能满足财务制度规范化的要求[1] 。

因此建立可视化、易观察、易管理，基于数字孪生技术模拟真实环境的机房管理系统，实现机房管理的信息化，透明化，可视化，自动化，让管理人员可以清晰直观地掌握服务器运行管理的有效信息，实现透明化与可视化管理，进而有效提升监控管理与资产管理的效率，对解决上述所暴露出来的问题是具有重要意义的。

## 研究手段

（1）查阅、收集、整理所需资料；

（2）根据设计任务制定工作进度；

（3）完成对数字孪生开发相关技术资料和交互可视化系统开发相关需求资料的查询和整理工作；

（4）选择合适的软件开发框架，完成系统的需求分析；

（5）完成系统的概要设计、详细设计，编码实现系统；

（6）完成相关技术文档的整理及论文的撰写；

## 论文组织结构

论文的结构为七个章节，其中各个章节的主要涉及的内容分别是：

第一章：绪论。提出论文的选题意义和相关背景，其中包括就机房管理系统的必要性研究以及投入使用后带来的优势，并阐述了研究过程和研究手段。

第二章：相关技术概述，详细说明基于多层架构，运行在浏览器中的机房管理系统的平台框架。列举并且简要介绍项目开发过程中涉及到的机房管理系统相关理论和技术，其中包括基于Java语言的SpringBoot开发框架技术，React 前端框架技术以及实现数字孪生3D模拟实际环境的WebGL第三方库three.js等。

第三章：系统需求分析，通过机房管理员对应的具体业务和对应的工作需求，来定义系统的具体功能和流程，完成了系统对应的过程建模，数据建模和需求建模。对于目标系统整体需求有一个完善的了解。

第四章：系统设计，设计系统的架构以及按照系统的功能需求分析划分功能模块，设计各个模块对应的功能业务。然后对应的数据库对象模型，根据对应的对象模型，在数据库中构造出对应的数据表。同时使用类图，序列图，流程图等配合文字，对本系统的主要模块进行详细的分析，最终集成各个模块构成系统。

第五章：系统开发与实现，根据已完成的系统架构，分模块逐个实现每个功能。通过系统各模块页面，搭配关键实现代码阐述功能或页面的实现思路。

第六章：系统测试，阐述了系统的运行环境，并构建单独的测试环境，对重点模块的类模型进行单独的单元测试，同时对系统主要功能点设计对应的测试用例，执行测试用例的同时梳理业务逻辑。最后就测试结果进行分析，并得出结论。

第七章：总结与展望，总结本论文的全部工作，分析工作内容的价值和必要性，总结目前系统的优点与长处，同时分析本系统的不足之处和需要改进的地方，并对本系统未来可能需要的功能、页面和性能方面的优化进行展望。

# 相关技术与工具概论

本章主要就机房管理系统开发过程中使用到的关键技术和主要工具进行介绍，相关技术介绍了系统后端数据接口实现技术：SpringBoot框架，前端页面实现技术React框架，实现在浏览器中3D模拟机房环境所用到的Three.js。数据存储方案使用了MySql数据库，数据接口遵循RESTful规范。开发过程的代码版本控制和同步使用到了Git。其中主要工具方案为，后端使用IntelliJ IDEA，前端使用Visual Studio Code，前后端接口调试工具使用Postman API，代码版本控制和同步平台GitHub。

## 相关技术介绍

### SpringBoot

SpringBoot概述和作用：即基于Spring框架的快速开发框架，用于简化Spring应用程序的配置和部署。

SpringBoot的优点和应用场景：具有快速启动、内嵌服务器、自动配置等，应用场景有Web应用、微服务等。

### React

React概述和作用：即用于构建用户界面的JavaScript库，遵循Flux单向数据流的应用架构。

React的优点和应用场景：可组合性、高效性、灵活性等，可以应用于单页应用、大型应用等。

### Three.js

Three.js概述和作用：即用于在Web上创建和呈现3D图形的JavaScript库，提供大量特性与API以绘制3D场景于浏览器。

Three.js的优点和应用场景：如易用性、跨平台、高性能等，应用场景有游戏、虚拟现实、实现基于数字孪生的数据可视化等。

### MySql

MySql概述和作用：即一个开源的关系型数据库管理系统，用于持久化存储和管理应用数据。

MySql的优点和应用场景：如可靠性、安全性、可扩展性等，另外也是最流行的关系型数据库之一，应用场景有Web应用、企业应用等。

### RESTful API

RESTful API概述和作用：REST即Representational State Transfer，用于设计和开发Web API的一种软件架构风格，REST原则是客户端和服务器之间的交互在请求之间是无状态的。

RESTful API的优点和应用场景：优点有行为和资源分离，接口开放性高，应用场景广泛，可应用于各式各样的的前后端服务请求。

### Git

Git概述和作用：即一个开源的分布式版本控制系统，用以有效、高速的处理从小到大的项目版本管理。

Git的优点和应用场景：是目前世界上最先进的分布式版本控制系统，没有之一，Git不但可以应用于应用代码的版本控制，并且可以用于文档的管理等。

## 相关工具介绍

### IntelliJ IDEA

IntelliJ IDEA 是一款由 JetBrains 公司开发的 Java 集成开发环境，提供了丰富的功能和插件，支持多种编程语言和框架，如 Java、Kotlin、Scala、Android、Spring 等等。它的主要特点是智能提示、代码重构、版本控制、调试等等。IntelliJ IDEA 适用于中大型项目的开发，具有良好的性能和稳定性。

### Visual Studio Code

Visual Studio Code（VSCode） 是由微软开发的一款跨平台开发工具，支持多种编程语言和框架，如 JavaScript、Python、C++、Vue、React 等等。VSCode 的特点是轻量、快速、可扩展，支持代码提示、调试、版本控制等等。它是一款适用于小型项目和日常编程的工具。

### Postman API Platform

Postman 是一款常用的 API 测试工具，支持多种 HTTP 请求方式，如 GET、POST、PUT、DELETE 等。它的特点是可视化的接口测试、请求的实时查看和响应的断言，可以帮助开发人员和测试人员更快速地调试和测试 API。

### GitHub

GitHub 是一个基于 Git 的代码托管平台，它为开发者提供了一个在线的代码托管服务，支持代码版本管理、代码审核、代码分享等等。在 GitHub 上可以找到很多开源项目和社区，方便开发者学习和交流。

## 本章小结

本章主要介绍了机房管理系统开发过程中所用到的关键技术和主要工具。在技术方面，介绍了系统后端数据接口实现技术SpringBoot框架、前端页面实现技术React框架和在浏览器中3D模拟机房环境所用到的Three.js等技术；数据存储方案使用了MySql数据库，数据接口遵循RESTful规范；在开发过程的代码版本控制和同步使用了Git。在工具方面，介绍了后端使用IntelliJ IDEA，前端使用Visual Studio Code，前后端接口调试工具使用Postman API，代码版本控制和同步平台GitHub等工具。同时，本章还对各个技术和工具的优点、应用场景进行了介绍。

# 系统需求分析

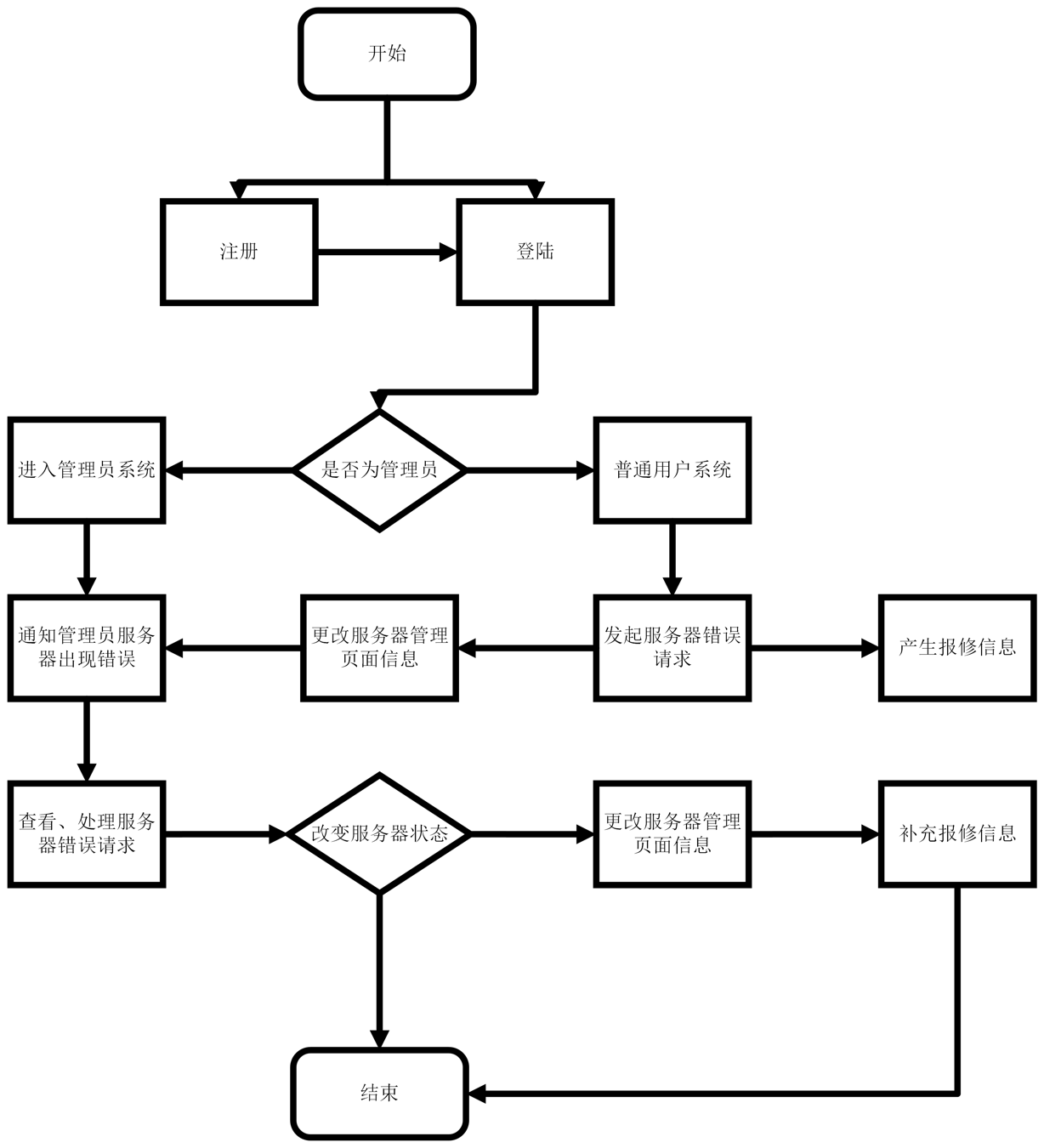
## 系统业务流程分析

首先确定业务目标和场景：业务目标是对机房服务器进行监测和管理维护，主要业务开始于普通用户在工作过程中发现服务器运行状态不正常，然后根据所用服务器的编号发起报错请求，并填写错误类型和具体错误内容。普通用户发起请求后会在系统上更新服务器管理界面中的相应编号的Fix Info状态为Show Fix Info，并展示错误类型和具体的错误内容，然后会管理员系统中在报修信息管理模块中生成一条报修记录，记录下本次报修相关信息如报修人，报修服务器编号，错误类型，错误具体内容等。

管理员在自己的系统上查看到某个状态不正常时，可以查看用户提交的错误类型和错误信息，然后去机房找到该编号的服务器进行检查，如果没有问题或者立即修复了错误则把状态修改为正常工作，如果检查到了错误并且没能当场修复，则在系统中将服务器状态修改为不工作。

以上流程完成之后会在报修信息管理模块中之前由于用户报修操作创建的报修记录中补全修复日期，修复人的用户名等。就此，本系统的主要业务流程结束。

根据上述对系统业务流程分析，可以设计出系统业务流程图，如图3-1所示：



**图3-1 系统业务流程图**

## 系统功能需求分析

### 登录注册模块

### 机房模拟及监控模块需求分析

### 机房状态管理模块

### 报修消息管理模块

### 用户管理模块

### 个人资料模块

## 系统非功能需求分析

## 本章小结

# 系统设计

## 系统功能设计

## 系统数据库设计

### E-R图设计

### 数据库表设计

## 本章小结

# 系统开发与实现

系统采用前后端分离开发模式，前端采用了在Node环境下部署的Vue项目，后端则采用Spring Boot框架，控制器在controller包下，系统发送的请求都会被对应的前端控制器锁捕获，在service包和mapper包下统一进行业务的实现。登录注册模块、机房模拟及监控模块、机房状态管理模块、报修消息管理模块、用户管理模块以及个人资料模块

## 登录注册模块

## 机房模拟及监控模块

## 机房状态管理模块

## 报修消息管理模块

## 用户管理模块

## 个人资料模块

## 本章小结

# 系统测试

## 测试目标和原则

## 系统测试环境

## 系统功能测试

## 测试结论

### 本章小结

# 总结与展望

## 论文结论与总结

## 未来研究展望

# 参考文献

# 致 谢