

本科毕业设计（论文）

题 目 基于区块链技术的老年人医疗

服务管理系统设计与实现

英文题目 Design and Implementation of Teaching State Management System

Based on Blockchain Technology

院（部） 智慧产业学院

姓 名 ✕✕✕

专 业 软件工程

年 级 2021级

指导教师 林亮 万为清

江西工程学院 教务处制

二〇二五年五月

**独创性声明**

本人郑重声明：

所呈交的毕业设计（论文）是本人在指导教师指导下进行的研究工作及取得的研究成果。其中除加以标注和致谢的地方，以及法律规定允许的之外，不包含其他人已经发表或撰写完成并以某种方式公开过的研究成果，也不包含为获得其他教育机构的学位或证书而作的材料。其他同志对本研究所做的任何贡献均已在文中作了明确的说明并表示谢意。

本毕业设计（论文）成果是本人在江西工程学院期间在指导教师指导下取得的，成果归江西工程学院所有。

特此声明。

作者签字： 日期： 年 月 日

**版权使用授权书**

本毕业设计（论文）作者及指导教师完全了解江西工程学院有关保留、使用毕业设计（论文）的规定，有权保留并向国家有关部门或机构送交毕业设计（论文）的复印件和磁盘，允许毕业设计（论文）被查阅和借阅。

作者签字： 指导教师签字：

年 月 日 年 月 日

基于区块链技术的老年人医疗服务管理系统设计与实现

摘 要

随着人口老龄化的不断加剧，老年人群体的医疗服务需求也日益增加。然而，老年人在传统的医疗服务就医过程中往往面临对医疗信息的不理解，以及在填写病历和获取处方方面的困难。此外，家属往往需要频繁介入，增加了照护负担。因此，如何优化老年人的就医体验，提升医疗服务的效率，成为当前医疗服务系统中需要解决的问题。

本文基于区块链技术设计一套老年人医疗服务系统，为老年人医疗服务系统提供创新解决方案。系统结合Gin与Grom框架构建Web应用，并通过运用Vue3编写页面。系统利用Hyperledger Fabric构建区块链网络环境，使其能够在区块链网络中实现病患的个人档案管理、就诊管理、取药管理、亲属账号管理、个人数据管理，并为管理者提供用户管理与数据加解密与访问控制。

本文分为四部分、共六章。从实际出发，对系统涉及的相关技术进行介绍，并对系统需求和系统设计进行详细描述，展示系统核心部分功能实现，最后对整个研究过程进行总结，对未来系统的发展提出展望。

关键词：老年人医疗服务；区块链；HyperLedger Fabric；Vue；Gin

**Design and Implementation of Teaching State Management System Based on**

**Blockchain Technology**

**ABSTRACT**

The college needs to participate in status data reporting, professional evaluation, and teaching evaluation. To generate a "Professional Teaching Status Data Analysis Report" that reflects the quality of professional talent cultivation, it requires multiple dimensions of status, such as faculty data, student data, teaching condition data, talent cultivation plan data, and teaching quality monitoring data. To ensure the quality of teaching, the college must monitor and obtain real-time status data from the above aspects. A set of software that matches the evaluation indicators is needed to collect data, obtain status data, and provide decision-making support for the college's teaching situation.

At present, the existing educational administration systems are all based on daily teaching business processes, without organizing data collection and monitoring from multiple dimensions mentioned above, which has a certain gap with the requirements of teaching status data analysis reports. Based on the operation and management of the college, a teaching status management system has been designed under the guidance of teachers, which is in line with the actual teaching situation of the college. Using Springboot+Vue+blockchain technology, collect and organize relevant teaching management data, generate teaching management status data, and compare the basic teaching status data with evaluation indicators as the basis for self-evaluation of the college. At the same time, this system can make teaching management work easier.

This article is divided into four parts and consists of six chapters. Taking the practical topic of "Teaching Status Management System Based on Blockchain Technology" in the teaching work of the college as a clue, this paper introduces the relevant technologies involved in the system, elaborates on system requirement analysis, system design and implementation, and explores the application of cross domain access and blockchain technology in this system. Finally, a summary of the entire text and the development work of this system was made, and prospects for future project development work and system improvement were proposed.

**KEYWORDS:**Teaching state; Blockchain;Springboot;Vue;Hash

目 录

[1 绪论 1](#_Toc15502)

[1.1 研究背景和意义 1](#_Toc17982)

[1.2 国内外发展现状及发展趋势 2](#_Toc19385)

[1.2.1 国外发展现状 2](#_Toc29011)

[1.2.2 国内发展现状 2](#_Toc17743)

[1.3 论文的组织、结构 2](#_Toc1584)

[2 相关技术和框架 3](#_Toc11135)

[2.1 Spring Boot 3](#_Toc26732)

[2.2 MybatisPlus 3](#_Toc15619)

[2.3 Vue 3](#_Toc10073)

[2.4 ElementPlus 3](#_Toc20859)

[2.5 跨域访问技术 4](#_Toc29531)

[2.6 Hash算法 4](#_Toc22998)

[2.7 加密解密技术 4](#_Toc31551)

[3 系统需求 5](#_Toc11495)

[3.1 需求概述 5](#_Toc22780)

[3.2 功能需求分析 5](#_Toc18821)

[3.3 系统用例模型 7](#_Toc12737)

[3.4 系统的性能需求 9](#_Toc17143)

[4 系统设计 10](#_Toc2885)

[4.1 系统设计原则 10](#_Toc29050)

[4.2 系统架构和开发环境 11](#_Toc10411)

[4.3 系统功能模块设计 13](#_Toc4122)

[4.4 数据库设计 18](#_Toc4988)

[5 系统实现 31](#_Toc31289)

[5.1 后端功能实现 31](#_Toc22920)

[5.1.1 登录接口以及加密解密技术应用 31](#_Toc3709)

[5.1.2 添加数据接口 33](#_Toc3751)

[5.1.3 删除数据接口 33](#_Toc10746)

[5.1.4 修改数据接口 34](#_Toc22967)

[5.1.5 查询数据接口 34](#_Toc1488)

[5.1.6 分页查询接口 35](#_Toc20037)

[5.1.7 利用Hash算法获取用户token接口 36](#_Toc21143)

[5.1.8 跨域访问 37](#_Toc8385)

[5.1.9 上传文件和图片的接口 38](#_Toc2126)

[5.2 前端功能实现 41](#_Toc28987)

[5.2.1 登录模块 41](#_Toc22185)

[5.2.2 工作平台首页 42](#_Toc2282)

[5.2.3 学生管理模块 43](#_Toc16631)

[5.2.4 教师管理模块 47](#_Toc2321)

[5.2.5 教学管理模块 49](#_Toc2542)

[5.2.6 科研管理模块 53](#_Toc15878)

[5.2.7 竞赛管理模块 54](#_Toc30496)

[5.2.8 资产管理模块 54](#_Toc8030)

[5.2.9 各类考证模块 55](#_Toc28911)

[5.2.10 调查问卷模块 56](#_Toc3726)

[6 系统测试 58](#_Toc27438)

[6.1 系统测试环境 58](#_Toc12161)

[6.1.1 服务器端 58](#_Toc9183)

[6.1.2 客户端 58](#_Toc21235)

[6.2 系统功能测试 59](#_Toc28831)

[6.3 系统性能测试 60](#_Toc26770)

[7 展望 61](#_Toc22725)

[7.1 总结 61](#_Toc24370)

[7.2 展望 61](#_Toc26954)

[参考文献 63](#_Toc19353)

[在校期间取得的成果 65](#_Toc22750)

[致谢 67](#_Toc12969)

[附录 69](#_Toc1511)

# 绪论

## 研究背景和意义

随着人口老龄化的不断加剧，老年人群体的医疗服务需求也日益增加。然而，老年人在传统的医疗服务就医过程中往往面临对医疗信息的不理解，以及在填写病历和获取处方方面的困难。此外，家属往往需要频繁介入，增加了照护负担。因此，如何优化老年人的就医体验，提升医疗服务的效率，成为当前医疗服务系统中需要解决的问题。

当前的医疗服务系统中都是针对大众群体所设计实现的，没有针对关爱人群如老年人展开针对性的服务。为给老年患者提供安全、便捷的就医体验，在多方面考虑与研究之下，本文结合对老年患者以及家属在就医过程中的需求，提出设计一套基于区块链技术实现的老年人医疗服务管理系统。

基于区块链技术的老年人医疗服务管理系统能够让老年患者方便地与医生沟通，并获取准确，可信的病历和处方。系统将数据所有权归属于病患本身所有，病患能够对自身档案、就诊记录、取药进行管理，支持病患对自身上链的数据设置访问策略，只有符合策略的用户才能访问数据，确保数据的所有权与隐私。家属能够代病患实施挂号、取药、数据查询等操作，使其可以更轻松地协助管理老年人的医疗事务，提高照护效率。此外，系统的实施将有效减少医院和药房的资源浪费，优化整体医疗资源配置。本系统的自主取药功能，使患者和家属可以灵活选择取药点，安排取药时间，降低就医和取药的难度。

系统结合区块链技术，实现老年人医疗服务的集中管理，提高医疗服务的效率和效果。系统的成功实施将为医疗行业的创新注入新动能，为区块链技术中医疗领域的其他应用探索提供宝贵的经验和实践数据，推动行业向信息化方向发展。

## 国内外发展现状及发展趋势

### 国外发展现状

在国外，区块链技术的应用在老年人医疗服务领域已经初见成效。例如，Manocha Ankush等人通过结合物联网、区块链等技术，开发了一种智能情境感知体力活动监测框架，该框架能够实时分析老年人的身体数据，并通过区块链技术保障个人数据的安全性和隐私性。Marco Rosa等人基于社区分散式护理服务的趋势，构建了一个社区服务平台，该平台利用区块链日志服务，保证社区平台活动记录的完整性，从而支持老年人的综合护理与援助服务。此外，该平台还利用快速医疗互操作性资源（FHIR）来促进健康数据的互操作性。Emilien等人设计并实现了一款老年人协作护理奖励系统，通过智能合约连接老年人和志愿者，鼓励社区成员参与老年人护理工作，增强社会对老年人的关注和支持。

### 国内发展现状

在国内，区块链技术在老年医疗服务领域的应用同样引发了广泛关注。温璐亚、赵袁杰基于当前养老医疗产业体系现状以及养老就医面临的痛点难点，构建了基于区块链技术的助老服务平台，其从信息机制、运行机制、维护机制三方面建设助老服务平台，用以完善现有养老医疗体系，提升养老医疗服务水平。刘一鸣、杨进娴通过分析老年人健康信息服务中存在的资源采集渠道单一、服务单元彼此割裂、服务效率存在延宕、评价结构松散僵化等问题，提出使用区块链技术实现老年人健康信息服务，并针对其精准定位、精准推送、精准反馈、精准评估的可行性进行研究分析。王婉娟基于区块链技术的老龄人口医养信息服务平台的技术基础，以“公有链+联盟链”的区块链组网形式，构建了老龄人口医养信息服务平台，以支撑医疗健康档案共享、医疗和养老信息管理以及医保账目记录与医养数据传递等业务应用。陈婕、侯进通过对当前社区老年听力健康管理现状分析，基于区块链技术构建了社区老年听力健康管理平台，以期为形成连续性、动态化的老年听力障碍预防、治疗和康复医疗服务提供参考。

综上所述，国内外的研究均表明，区块链技术在老年人医疗服务中的应用前景广阔，不仅能够保障老年人健康数据的隐私性，还能通过智能合约、互操作性等技术手段优化服务流程，提升服务效率。

## 论文的组织、结构

本文分为四部分，共六章。以学院教学工作中的实际课题--“基于区块链技术的✕✕✕✕管理系统”为线索，介绍了系统涉及的相关技术和框架，阐述了系统分析、设计和实现，作了较为详细的系统需求分析、系统设计、系统实现描述，深入探讨了跨域访问及区块链技术在本系统中的应用，并给出了解决方案。

第一部分即本论文的第一章：叙述了本课题的来源和背景，研究目的和意义以及相关项目在国内外研究和发展现状。

第二部分即本论文的第二章，主要阐述了本系统用到的相关技术。分析了Springboot、MybatisPlus、Vue、ElementPlus、跨域访问技术、Hash算法、加密解密技术的基本情况和在本系统中的应用情况。

第三部分即本论文的第3至5章：阐述了本课题的系统需求分析、系统设计和实现以及部分核心代码分析。

第四部分即本论文的第6章：对✕✕✕✕管理系统开发工作和本论文作总结，并对后期的项目开发工作及系统改进方面提出展望。

# 相关技术和框架

上一章对本系统的研究背景与意义、国内外同类系统的研究总结、本论文结构分别进行了阐述，本章将对系统中涉及的相关技术与框架进行说明。

## Gin

Gin是使用Golang语言编写的Web框架，其为开发者提供了简洁灵活API功能的同时提供了高性能，使开发者能够高效、快速的开发Web项目。Gin能够快速解析请求的Json以及

## Grom

Mybatis是一套连接数据库的框架，集成自Mybatis，是阿里工程师自行研发的，已经被国内工程师大量采用。将以往jdbc连接数据库的方法封装，极度简化了操作，通过注解的方式注入查询语句，实现数据查询和增删改等操作。

## Vue3

Vue是一套构建用户界面的渐进式框架。采用数据双向绑定的方式是它最大的优势，方便用于构建单文档页面应用程序，目前流行的有Vue2.0和3.0。它是一套双向绑定数据的java script库，使程序员更多的是专注数据，无需花大量时间和精力去关注view层渲染。

本系统中采用了Vue3.0。利用Vue和ElementPlus来实现前端页面构建。

## SoftHSM

ElementPlus是一套基于vue的UI组件库，本系统中主要用ElementPlus来做前端开发，用于表格、按钮、分页组件等。

## CP-ABE

目前主流的浏览器为了自身的数据访问安全，都设置了同源策略。同源策略限制是检查请求的URL协议、域名和端口，只要有其中一项与当前页面的URL不同，就会出现跨域访问问题。同源策略会阻止一个域和另一个域的数据和内容交互传递[4]。

当前主流开发模式都是把前端和后端分离，基本都存在跨域访问的问题。本系统中采用全局跨域，[重写 WebMvcConfigurer](https://mp.weixin.qq.com/s?__biz=MzUzMTA2NTU2Ng==&mid=2247487551&idx=1&sn=18f64ba49f3f0f9d8be9d1fdef8857d9&scene=21" \l "wechat_redirect" \o "2. 重写 WebMvcConfigurer(全局跨域))来实现跨域访问。

## HyperLedger Fabric

Hash算法的核心就是设计一个hash函数 ，每次传给它一个给定值。Hash算法可以将一个数据转换为一个标志，这个标志和源数据的每一个字节都有十分紧密的关系。Hash算法还具有一个特点，就是很难找到逆向规律。Hash算法是一个广义的算法，也可以认为是一种思想[5]。

本系统中主要利用Hash算法来生成链码，保存上一数据库的哈希值，防止篡改。

# 系统需求

针对专业✕✕✕✕管理的实际情况，根据用户需求调研，结合大量的文献检索得出了本系统的总体需求，以下将从功能性需求和非功能性需求两方面展开论述。

## [需求概述](" \t "_blank)

学院需要参加状态数据填报、专业评价、教学评估，要生成《专业✕✕✕✕数据分析报告》，反映专业人才培养质量，需要师资队伍数据、学生数据、教学条件数据、人才培养方案数据、教学质量监控数据等多个维度的状态。学院需要实时监控并得到以上多个方面的状态数据，就需要一套与评价指标相吻合的软件来收集数据，得出状态分析，对学院教学情况提供决策方面的支持。对相关教学管理数据进行收集、整理，生成教学管理状态数据，将教学基本状态数据与评估指标对照，作为学院自我评价的依据。同时✕✕✕✕管理系统能够使管理教学工作变得更为轻松。

目前现有的教务系统都是从日常教学业务来开展的业务流程，没有从以上多个维度来组织数据收集并开展监控，与✕✕✕✕数据分析报告的要求有一定的差距。

## 功能需求分析

评价指标中，专业基本情况主要反映了专业认证的基本情况和专业带头人的情况。培养目标主要包括学校目标与定位，发展目标，人才培养定位。专业毕业要求主要是毕业理论和实践课的学分要求。课程体系包括人才培养方案、教学大纲、实验课程、学时学分、课程性质、课程属性、实习实训等。师资队伍包括专人、兼任老师数量和结构，学缘关系、科研等情况。教学条件包括生均教学经费、图书资源、实验设施设备、校企合作基地等情况。质量保障包括专业质量保障体系，质量主要控制制度和手段，专业教学质量调查，满意度评价。学生发展包括学生数据、各类考证情况、比赛获奖情况、参与教师研究项目情况、论文和专利以及软著等取得的情况。

从以上功能需求出发，将系统分为八个模块并赋予相应的功能。

1.学生管理

班级数据：新建班级信息，修改班级数据和删除班级数据，根据班级编号查询班级详细信息[7]。

学生数据：新建学生信息，修改学生数据和删除学生数据，能够根据学生姓名，班级名称，或学生所在地区作为关键字，查询到学生的详细信息。

考勤数据：添加学生考勤信息，导出考勤数据，根据班级查询考勤数据，或者根据日期查询学生考勤数据。

导入导出：将已经按模板数据表做好的Excel的学生信息批量导入数据库中；将查询得到的学生信息表导出为Excel格式文件(.xls)。

分析数据：用echart图表将历年来的所有学生数据按照学生所在省份统计学生来源，形成直方图和圆饼图。

2.教师管理

教师数据：能够对教师数据做增删改操作，也能根据教师姓名或者工号查询教师详细信息。

操作员管理：能够实现增删改操作，也能根据工号查询操作员详细信息。在操作员信息表中能够展示编号，用户姓名，头像，备注。能够编辑操作员信息。

3.教学管理

人才方案：能够将各专业的人才培养方案录入到系统中，也可以根据课程名查询课程信息，或者根据学期查询这个学期所开设的课程，也可以编辑人才培养方案的信息[8]。

学院课表：课表中能够展示上课时间，课程名，任课老师，上课地点，展示形式和各班级张贴的课表形式一致。能够根据学年学期查询所开设的课程，可以导出查询结果

专业教材：能够根据课程名查询所使用的教材，根据学期查询本学期所开设课程用到的教材，能够将查询结果导出到电子表格。

学院公文：收录学院历年来的各类公文，能够根据公文名中所包含的关键字模糊查询到对应的文件，将查询到的文件导出，同时展示PDF格式的学院公文。

实习基地：这个部分主要完成校外实训实习基地数据的增删改查工作。能够根据公司名查询基地详细信息。

4.科研管理

主要收录教师和学生历年来所完成的各类科研成果信息，科研成果主要包括出版的教材，论文，专利，软件著作权，教改课题，科研项目等成果[9]。

5.竞赛管理

主要完成教师和学生所参加的比赛获奖情况的增加、删除、修改，可以根据教师或者学生的姓名来查询所获得的比赛奖励。

6.资产管理

资产管理主要完成教学设备的增删改操作，可以根据设备名称查询到设备的详细信息。查询结果能够导出为Excel格式文件(.xls)。

7.考证管理

考证管理主要记录教师和学生在各类考证中获得的证书情况，同时对考证信息做增删改操作，可以根据姓名查询到所获得的证书详细信息。查询结果能够导出为Excel格式文件(.xls)。

8.质量监控

质量监控主要完成评教评学的调查问卷，主要包括学生评教，教师评学的调查问卷。能够生成问卷分析报告[10]。

## 系统用例模型

根据✕✕✕✕管理系统的业务需求和系统使用目标，只涉及✕✕✕✕的数据生成以及质量监控等相关内容。

根据系统使用范围及需求，✕✕✕✕管理系统应该有三类参与者：系统管理员、学生、教师。系统管理员是本系统的组织者，依照业务流程组织完成各项操作。教师和学生分别完成身份对应的职责和操作。

系统顶层用例如图3-1。

考勤查询

学生

教师

管理员员

科研成果

各类比赛

教学设备

各类考证

人才培养方案

学院课表

专业教材

学院公文

实习基地

学生评教

教师评学

数据维护

**图3-1 系统顶层用例**

## 系统的性能需求

为了确保系统使用不受地点和时间限制，使用环境是基于互联网的网络环境。系统应该与教育部《专业✕✕✕✕数据分析报告》有较高程度的吻合。为了满足以上功能需求，系统在性能方面应该满足以下要求：

1.准确性

数据准确性是一个系统的起码要求，本系统的准确性要求在学生信息、教师信息、人才培养方案、课程体系信息等多方面做到准确无误，否则系统就没有存在的必要性。

2.可靠性

可靠性是指系统在意外情况的处理上的可靠程度。本系统依赖云端服务器和互联网网络环境。实际使用中可能会出现各种突发情况，要求系统具备防止突发的措施，以免出现突发状况时丢失数据。

3.易用性

系统应该做到UI符合主流商业软件UI规范，方便管理人员和师生使用。系统还应操作简单方便，兼顾常规操作习惯。系统部署运维置应该简单方便，方便数据和系统恢复。

4.安全性

系统中存储大量的师生信息和教学运行信息，要求数据加密存储，加密传递，防止敏感信息外泄，利用区块链技术防止运行数据被任意篡改。

5.容错性

系统实际过程中难免遇到非法操作，系统必须根据操作情况给予警告提示，防止异常操作影响数据准确性和及时性。

# 系统设计

根据上一章所做系统需求分析，明确了功能性和非功能性需求。本章将根据需求分析，阐述系统设计原则、系统架构和系统功能模块设计、数据库设计。

## 系统设计原则

为了满足系统功能和性能方面的需求，根据✕✕✕✕管理业务逻辑，设计本系统时将遵循如下几个原则：

1. 系统应符合专业✕✕✕✕数据分析业务要求

根据专业✕✕✕✕数据分析报告要求，系统应涵盖人才培养方案、学生基本数据、教师基本数据、资产数据、教材数据、各类考证数据、教师科研数据、质量保障等教学业务数据。其中包括专业相关详细信息，如学生管理、班级数据、学生数据、考勤数据、分析数据；教师管理、教师数据、操作员管理、科研管理、成果查询、竞赛管理、比赛查询、资产管理、教学设备、各类证书、考证查询、教学管理、人才方案、学院课表、专业教材、学院公文、实习基地的情况。

1. 设计出符合数据库原理理论要求和✕✕✕✕管理实际需求的数据结构

在✕✕✕✕管理系统中共有十五种数据表：待办日程、学生数据、班级数据、考勤数据、教师数据、操作员管理、科研成果、竞赛管理、教学设备、各类考证、人才方案、学院课表、专业教材、学院公文、实习基地。涵盖了✕✕✕✕管理的主要业务数据。设计系统中各类数据表的时候，其结构必须遵循数据库基本理论，表中各字段宽度要达到实际业务需要，也要前瞻性的留有一定的余地以应对系统面临特殊情况或者系统升级带来的变动，同时各数据表中字段各项属性必须完整且准确。

1. 设计出符合区块链技术要求的链式数据存储方法

充分利用对称加密和非对称加密方法，用于数据区块数据的数字签名和前后数据区块哈希值的数字验证，数据传输中采用经过数字签名后的哈希值，重要的基础信息都应该在每个数据块上存储前一个数据块的哈希值，相互关联链式存储，避免数据篡改。后一个数据块的哈希值是由前一个数据块的所有字段求出的哈希值，如果要篡改后面的数据，必须向前修改每一个数据块的哈希值，递归方式向前递推修改，这样就增加了篡改的难度[11]。

1. 系统设计应该功能完善合理

系统应涵盖专业✕✕✕✕数据分析所必需的业务，系统应该包含学生管理，教师管理，科研管理，竞赛管理，资产管理，各类考证管理，教学管理，调查问卷等功能。具备学生、教师、人才培养方案等基础信息的建立和维护，能够实现多角度精确查询和模糊查询，生成分析图表等功能，涵盖专业✕✕✕✕管理流程中的各项业务流程。

1. 设计的系统要有安全性的保障

系统的安全性在整个✕✕✕✕管理系统中具有非常重要的意义，✕✕✕✕管理虽然是属于日常教学管理，但是系统中涉及大量的学生、教师信息和教学运行状态数据的安全问题也是一项相当重要事情，事关师生和学院数据敏感问题，因此系统和数据库应该具有强大的安全性，不允许出现敏感数据泄露的情况发生。因而在服务器上保存基础信息和运行状态数据的数据库除了设置MySql数据库本身的安全机制之外，还有必要额外增加通过私钥加密等防范措施。根据权限设置多级身份验证、按角色权限管理等措施，防止非法用户和非法程序窃取数据信息，同时也要防止撞库等恶意攻击数据库行为的发生。

1. 设计的系统要稳定性好、实用性强

✕✕✕✕管理系统应该具有较高的稳定性，要求系统能满足和适应不同角色的师生使用，适应广大师生具备的多种软硬件环境[12]。简化对使用者的操作要求，实现系统简单易用。系统运行在B/S模式下，各种角色的师生只要能上网开网页就会使用本系统，客户端不用作任何配置，不用安装特定的软件和插件，也不需要做数据库环境设置等任何操作。

根据以上分析，✕✕✕✕管理系统应该满足安全可靠、简单易用、功能与管理需求吻合等多方面要求。

## 系统架构和开发环境

**4.2.1** 系统架构设计

根据以上系统设计原则和需求分析的情况，为实现系统基于区块链技术的✕✕✕✕管理系统，需要将系统分为前端和后端分离开发，前端主要解决数据展示以及与用户交互功能，后端主要解决连接数据库，向前端提供数据，实现数据持久化，以及数据加密解密等功能。

✕✕✕✕管理系统采用阿里云服务器作为运营服务器，以Internet网络环境为基础，采用Java EE+Vue3.0+Element Plus为技术平台，以B/S模式为结构，采用前后端分离的设计方法，把整个✕✕✕✕管理系统划分为八大功能模块，各模块根据✕✕✕✕业务流程通过后端数据接口实现数据处理，实现基础数据集中管理、运行状态数据分散操作，使日常✕✕✕✕管理得到系统化、规律化、自动化、实时化，最终实现随时随地都可以得到实时的专业✕✕✕✕数据，达到提升管理水平的目的。✕✕✕✕管理系统框架如图4-1。



MySql数据库云端服务器



Web云端服务器

B/S客户端

互联网

bk

图4-1 ✕✕✕✕管理系统框架

**4.2.2** 系统开发环境

✕✕✕✕管理系统在实际使用中，客户端可能是在PC端使用，可能在手机上使用，也可能在平板电脑上使用，为了适应各种设备，本系统开发采用Java EE的Springboot框架作为后端开发框架，前端采用Vue3.0+ElementPlus，数据库则采用MySql8.0。前端开发工具采用WebStorm 2022.3，后端开发工具采用InteLLiJ IDEA 2022.3，这是目前各大软件公司普遍采用的主流开发环境。

✕✕✕✕管理系统开发的具体软件细节、服务器配置和环境，因篇幅所限不作详细描述。

## 系统功能模块设计

**4.3.1** 系统功能模块结构

满足前述各种需求的训练系统包括以下几个模块：学生管理，教师管理，科研管理，竞赛管理，资产管理，考证管理，教学管理，质量监控调查问卷。系统功能模块结构如图4-2所示。

✕✕✕✕管理系统

学生管理

科研管理

考证管理

资产管理

质量监控

竞赛管理

教师管理

教学管理

教学设备

考证查询

成果查询

学生评教

比赛查询

人才方案

教师数据

班级管理

学院课表

操作员管理

教师评学

专业教材

学院公文

考勤数据

学生数据

分析数据

实习基地

图4-2系统功能模块结构

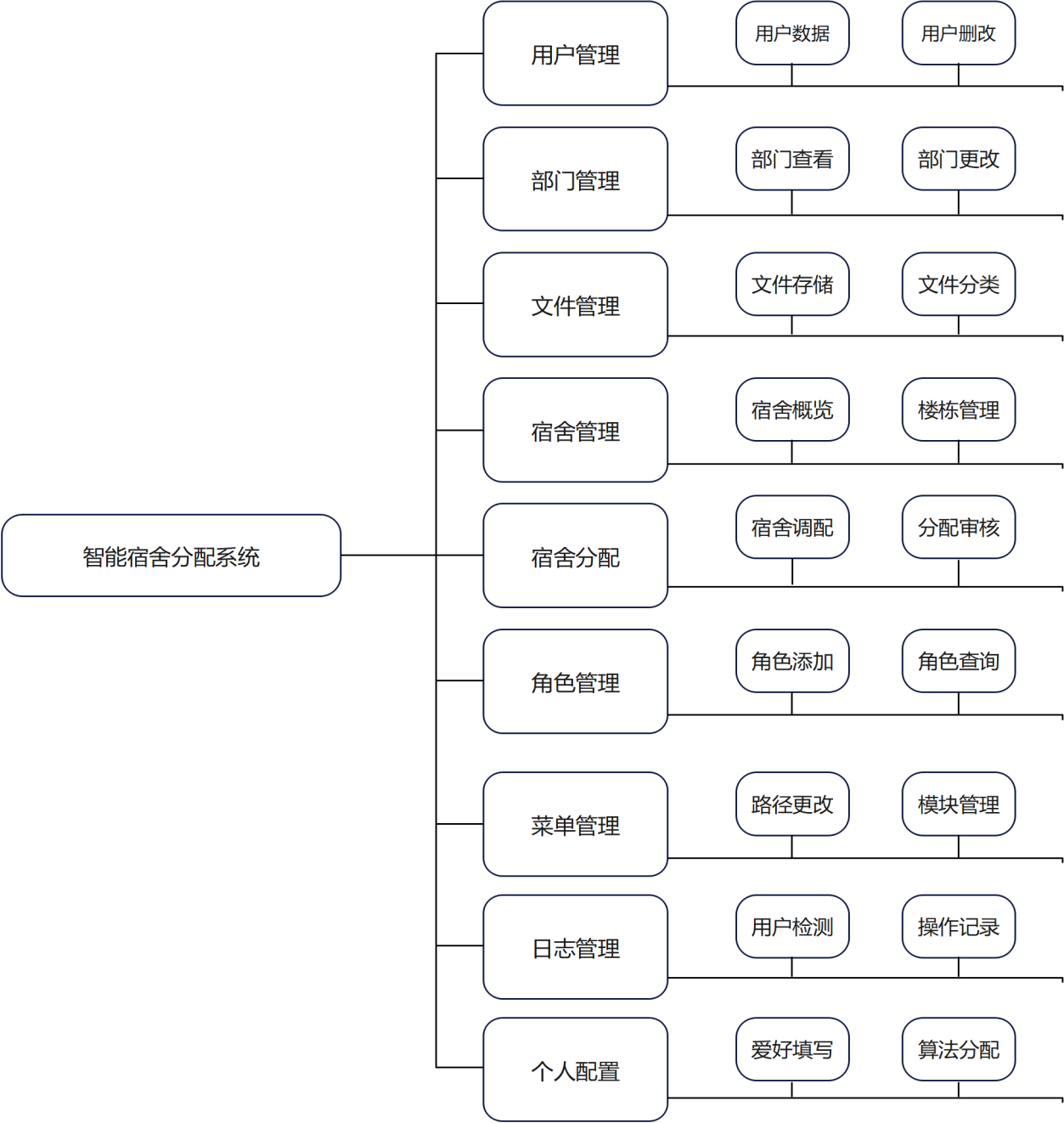


图4-2另外一种方式的模块结构

**4.3.2** 系统模块功能描述

1.学生管理

* 班级数据：新建班级信息，修改班级数据和删除班级数据，根据班级编号查询班级详细信息。班级数据主要包括编号，班级名称，班级位置，班主任班级人数等基本信息[13]。
* 学生数据：新建学生信息，修改学生数据和删除学生数据，能够根据学生姓名，班级名称，或学生所在地区作为关键字，查询到学生的详细信息。学生信息主要包括学号，姓名，性别，班级，电话，出生年月，家庭住址，宿舍地址，身份证，民族籍贯，政治面貌，照片，上链时间，前一数据块哈希值[14]。
* 考勤数据：添加学生考勤信息，导出考勤数据，根据班级查询考勤数据，或者根据日期查询学生考勤数据。考勤数据主要包括班级编号，班级名称，班级人数，出勤人数，出勤率，考缺勤人数，考勤时间，考勤员，前一数据块哈希值和备注。
* 导入导出：将已经按模板数据表做好的Excel的学生信息批量导入数据库中；将查询得到的学生信息表导出为Excel格式文件(.xls)。
* 分析数据：用echart图表将历年来的所有学生数据按照学生所在省份统计学生来源，形成直方图和圆饼图。

2.教师管理

* 教师数据：能够对教师数据做增加、删除、修改，也能根据教师姓名或者工号查询教师详细信息。教师信息主要包括工号，姓名，性别，出生年月，电话，民族职称，教师资格证，编号所在，部门政治面貌，学历学位，上链时间，前一区块哈希值，照片，备注。
* 操作员管理：能够增加、删除、修改操作员信息，也能根据工号查询操作员详细信息。在操作员信息表中能够展示编号，用户姓名，头像，备注。能够编辑操作员信息。操作员信息主要包括编号，姓名，密码，token值，照片，备注。这个部分能够上传或修改操作员的照片。
* 导入导出：将已经按模板数据表做好的Excel的教师或操作员信息批量导入数据库中；将查询得到的教师或操作员信息表导出为Excel格式文件(.xls)。

3.教学管理

* 人才方案：能够将各专业的人才培养方案录入到系统中，也可以根据课程名查询课程信息，或者根据学期查询这个学期所开设的课程，也可以编辑人才培养方案的信息。人才培养方案信息主要包括课程编号，课程名，学分，理论学时，实践学时，考核方式，课程类别，专业开设学期[15]。
* 学院课表：课表中能够展示上课时间，课程名，任课老师，上课地点，展示形式和各班级张贴的课表形式一致。能够根据学年学期查询所开设的课程，可以导出查询结果。
* 专业教材：能够根据课程名查询所使用的教材，根据学期查询本学期所开设课程用到的教材，能够将查询结果导出到电子表格。教材信息主要包括课程编号，课程名，教材名，教材编号，作者，出版社，数量，单价，使用范围。
* 学院公文：收录学院历年来的各类公文，能够根据公文名中所包含的关键字模糊查询到对应的文件，将查询到的文件导出，同时展示PDF格式的学院公文。公文信息主要包括公文编号，公文名，发文时间，公文内容。
* 实习基地：这个部分主要完成校外实训实习基地数据的增删改查工作。能够根据公司名查询基地详细信息。基地信息包括基地编号，基地名称，基地地址，联系人，联系电话。

4.科研管理

* 主要收录教师和学生历年来所完成的各类科研成果信息，科研成果主要包括出版的教材，论文，专利，软件著作权，教改课题，科研项目等成果。成果信息主要包括编号，名称，类别，详情，时间，作者备注。

5.竞赛管理

* 这部分主要完成教师和学生所参加的比赛获奖情况的增加、删除、修改，可以根据教师或者学生的姓名来查询所获得的比赛奖励。竞赛信息主要包括编号，姓名，比赛项目，获奖情况，参赛时间。查询结果能够导出为Excel格式文件(.xls)。

6.资产管理

* 资产管理主要完成教学设备的增删改操作，可以根据设备名称查询到设备的详细信息，设备信息包括编号，名称，型号，单位，单价，数量，备注。查询结果能够导出为Excel格式文件(.xls)。

7.考证管理

* 考证管理主要记录教师和学生在各类考证中获得的证书情况，同时对考证信息做增删改操作，可以根据姓名查询到所获得的证书详细信息，证书信息包括编号，姓名，证书名称，证书编号，获证时间，发证单位。查询结果能够导出为Excel格式文件(.xls)。

8.质量监控

* 质量监控主要完成评教评学的调查问卷，主要包括学生评教，教师评学的调查问卷。能够生成问卷分析报告。

**4.3.2** 各模块的业务流程

登录流程：登录流程是每一个系统所必须具备的，它是用户进入系统的入口，登录流程分为注册和登录两个部分，其中对于注册部分，用户可以通过注册功能创建新的账号。注册功能需要验证用户输入的信息，如用户名、密码、邮箱等，确保其合法性和唯一性。而对于以及注册的用户，可以使用登录功能进入系统。登录功能模块需要先验证用户登录账户时所输入的用户名以及该用户名所对应的密码，确保输入的准确性，随后再提供验证码从而进一步保证登录的可靠性。在登录过后，系统需要在一段时间内保持用户的登录状态，以便用户可以持续访问系统而不需要重复登录，避免频繁的登入登出。与此同时，也要对用户的登录信息进行记录以及实时地监控。登录业务流程如图4-3所示。

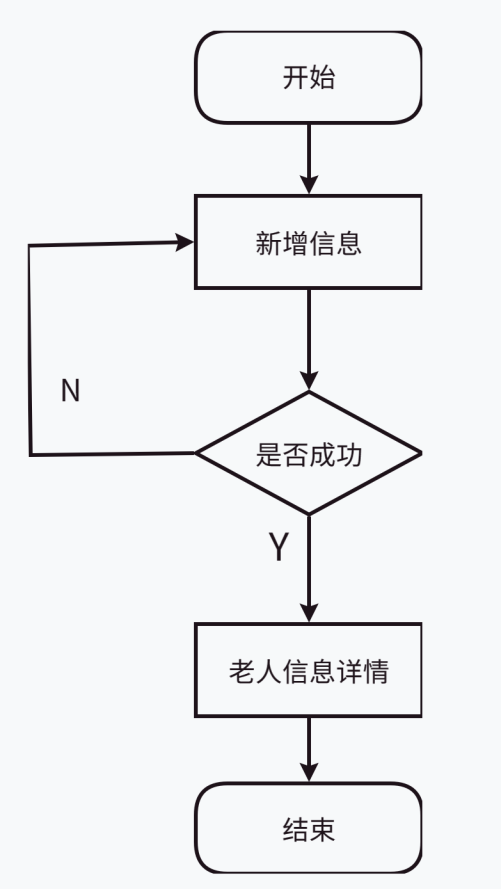


图4-3xxxx流程图

其他模块意义列举................

## 数据库设计

**4.5.1** 数据库设计概述

数据库设计是在对业务逻辑做深入分析之后，结合数据库设计原理来构造符合业务需求的逻辑模型。数据库设计中应考虑数据冗余性，业务之间的关联性、系统运行效率、数据库实施问题等[16]。

经过对需求分析综合考量，本系统选择使用关系数据库。

结合对✕✕✕✕管理系统所做的需求分析，本系统后台数据库采用MySQL 8.0。MySQL 8.0具有稳定性好、成熟度高、免费开源的优势，被业界大量使用。能够完全满足✕✕✕✕管理系统在数据安全性、完整性、可靠性等方面的需求。

**4.5.2** 数据库关系模型设计

根据✕✕✕✕管理系统的需求，系统采用MySQL 8.0的数据库，库名为BlockChain。实体属性和E-R图如下。

先做E-R图，再做实体属性图。

1.xxxx系统E-R图如图4-14所示

学生

1

教学设备

属于

各类考证

n

班级

人才方案

CRUD

n

管理

m

专业教材

教师

CRUD

科研成果

CRUD

课表

CRUD

学院公文

CRUD

操作员

CRUD

CRUD

竞赛

CRUD

CRUD

实习基地

图4-4 系统整体E-R图

1. 用户信息实体属性图如图4-5所示。

图4-5待办日程实体属性图

1. 用户信息实体属性图如图4-6所示。

图4-6待办日程实体属性图

1. 用户信息实体属性图如图4-7所示。

图4-7待办日程实体属性图

系统包含十个数据表，各数据表的名称与功能如表4-1所示。

**表4-1老年人医疗服务管理系统数据库的主要数据表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 数据表名称 | 数据表描述 | 数据表功能 |
| 1 | t\_authority | 授权机构 | 存储授权机构信息 |
| 2 | t\_encrypted\_data | 加密数据记录 | 存储指定数据加密时生成的数据 |
| 3 | t\_user\_attributes | 用户属性 | 存储用户属性信息 |
| 4 | t\_users | 病患信息 | 存储病患证书信息 |
| 5 | t\_user\_registration | 用户注册申请 | 存储用户注册信息 |
| 6 | t\_medical\_facility | 医疗机构信息 | 存储医疗机构基本信息 |
| 7 | t\_doctor | 医生信息 | 存储医生基本信息 |
| 8 | t\_departments | 科室信息 | 存储科室信息 |
| 9 | t\_log | 日志信息 | 存储用户操作日志记录 |
| 10 | t\_medical\_record | 就诊记录 | 存储病患就诊记录信息 |
| 11 | t\_medical\_process | 就诊过程 | 存储病患在就诊过程中的产生的信息 |
| 12 | t\_ctypto\_exp | 加密策略 | 存储参考加密策略信息 |
| 13 | t\_dispensing | 取药单据记录 | 存储取药单据记录 |

各数据表结构描述如下：

1. 授权机构表(t\_authority)

授权机构表是用于存储生成用户属性密钥时涉及的授权机构的信息，其中包括授权机构ID、授权机构名称、授权机构说明、授权机构内部包含的属性值、创建时间、修改时间、数据版本。系统将单个属性均作为一个授权机构，授权机构属性值以name:value的形式存储，多个值通过逗号对数据进行分割。表结构如表4-2所示。

**表4-2待办日程表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| ta\_key | varchar | 64 | Not Null | PK | 授权机构ID |
| auth\_name | varchar | 64 | Null |  | 授权机构名称 |
| auth\_description | varchar | 255 | Null |  | 授权机构说明 |
| attribute | varchar | 255 | Null |  | 授权机构属性值 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 加密数据记录表(t\_encrypted\_data)

加密数据记录表主要对存储智能合约中的每条加密数据涉及的辅助加密数据进行存储。其中包括数据ID、加密者用户ID、加密后生成的相关文件内容 、创建时间、修改时间与数据版本。系统使用cp-abe算法加密后生成的相关内容将分别存入c0、c1x、c2x、c3x、msp中。表结构如表4-3所示。

**表4-3加密数据记录表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| ted\_key | varchar | 64 | Not Null | PK | 数据ID |
| user\_id | varchar | 65 | Not Null |  | 加密者的用户ID |
| c0 | blob |  | Not Null |  | 加密生成的C0文件 |
| c1x | blob |  | Not Null |  | 加密生成的C1X文件 |
| c2x | blob |  | Not Null |  | 加密生成的C2X文件 |
| c3x | blob |  | Not Null |  | 加密生成的C3X文件 |
| msp | blob |  | Not Null |  | 加密生成的MSP文件 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 用户属性表(t\_user\_attributes)

用户属性表存放用户身份属性信息，包括属性记录ID、用户ID、授权机构ID、属性值、数据创建时间、数据修改时间、数据版本。属性值以name:value的形式存储。表结构如表4-4所示。

**表4-4用户属性表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| tua\_key | char | 64 | Not Null | PK | 属性记录ID |
| user\_id | varchar | 64 | Not Null |  | 用户ID |
| ta\_key | varchar | 64 | Not Null |  | 授权机构ID |
| attribute | varchar | 255 | Not Null |  | 属性值 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 用户信息表(t\_users)

用户信息表用于存储用户身份信息，包括用户ID、用户姓名、用户身份证号码、用户密码哈希值、所属MspId、身份证书、链上身份ID、数据创建日期、数据修改日期、数据版本。表结构如表4-5所示。

**表4-5用户信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| user\_id | varchar | 64 | Not Null | PK | 用户ID |
| user\_name | varchar | 64 | Not Null |  | 用户名 |
| id\_number | varchar | 18 | Not Null |  | 用户身份证号码 |
| password\_hash | varchar | 100 | Not Null |  | 用户密码哈希 |
| msp\_id | varchar | 20 | Not Null |  | Msp id |
| enrollment\_cert | blob |  | Not Null |  | 身份证书 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 用户注册申请表(t\_user\_registration)

用户注册申请表用于存储用户发起的注册申请信息，包括用户ID、姓名、身份证号码、性别、出生年月、血型、身高、体重、居住地址、联系方式、过敏史、历史病史、数据创建时间、数据修改时间、数据版本。其中result包括通过和拒绝两种结果，当result为空时，则说明申请未被审核，表结构如表4-6所示。

**表4-6用户注册申请表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| tur\_id | varchar | 64 | Not Null | PK | 申请记录ID |
| name | varchar | 20 | Not Null |  | 姓名 |
| id\_number | varchar | 20 | Not Null |  | 身份证号码 |
| gender | varchar | 2 | Not Null |  | 性别 |
| date\_of\_birth | varchar | 26 | Not Null |  | 出生年月 |
| blood\_type | varchar | 10 | Null |  | 血型 |
| height\_cm | varchar | 10 | Null |  | 身高 |
| weight\_kg | varchar | 10 | Null |  | 体重 |
| address | varchar | 200 | Null |  | 居住地址 |
| phone\_number | varchar | 18 | Not Null |  | 联系方式 |
| allergy\_history | varchar | 200 | Null |  | 过敏史 |
| medical\_history | varchar | 200 | Null |  | 历史病史 |
| password | varchar | 64 | Not Null |  | 密码 |
| result | varchar | 10 | Null |  | 审核结果 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 医疗机构信息表(t\_medical\_facility)

药房信息表用于存储医疗结构的基础信息，包括机构ID、机构名称、许可编号、联系人姓名、联系人电话、邮箱、地址、状态、介绍、营业时间、数据创建时间、数据修改时间、数据版本。医疗机构类型包括药房、医院两种类型，状态值包含营业中、已关闭、暂停营业，另外，医院级别仅适用于类型为医院的记录。表结构如表4-7所示。

**表4-7医疗机构信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| tmf\_id | varchar | 64 | Not Null | PK | 机构ID |
| name | varchar | 20 | Not Null |  | 机构名称 |
| facility\_type | varchar | 10 | Not Null |  | 医疗机构类型 |
| license\_number | varchar | 30 | Not Null |  | 许可编号 |
| contact\_person | varchar | 20 | Null |  | 联系人姓名 |
| phone\_number | varchar | 18 | Null |  | 联系人电话 |
| email | varchar | 30 | Null |  | 联系邮箱 |
| address | varchar | 200 | Null |  | 地址 |
| status | varchar | 10 | Not Null |  | 状态 |
| desc | varchar | 200 | Null |  | 介绍 |
| operating\_hours | varchar | 30 | Null |  | 营业时间 |
| hospital\_level | varchar | 10 | Null |  | 级别 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 医生信息表(t\_doctor)

医生信息表用于存储医生的基本信息，包括医生ID、所属医院ID、医生姓名、医生性别、所属科室、状态、医生级别、医生介绍、数据创建时间、数据修改时间、数据版本。状态包括接诊中，已接满和已休息两种状态，医生级别包含实习医生、住院医师、主治医师、副主任医师、主任医师、教授。其表结构如表4-8所示。

**表4-8医生信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| td\_id | varchar | 64 | Not Null | PK | 医生ID |
| hospital\_id | varchar | 64 | Not Null |  | 所属医院ID |
| name | varchar | 10 | Not Null |  | 医生名称 |
| gender | varchar | 120 | Not Null |  | 医生性别 |
| specialty | varchar | 10 | Not Null |  | 所属科室 |
| status | varchar | 10 | Not Null |  | 状态 |
| level | varchar | 10 | Not Null |  | 医生级别 |
| desc | varchar | 200 | Null |  | 医生介绍 |
| max\_number | int |  | Not Null |  | 最大就诊数 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 科室信息表(t\_departments)

科室信息表用于存储医院相关科室信息，包括科室ID、所属医院ID、科室名称、科室代码、科室类别、科室描述、科室负责人、科室电话、科室位置、科室服务内容、科室工作时间、科室状态、数据创建时间、数据更新时间、数据版本。其中科室状态包含Active和Inactive两种类型，科室类别包含临床科室、医技科室、辅助科室、特色科室。其表结构如表4-9所示。

**表4-9科室信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| td\_id | varchar | 64 | Not Null | PK | 科室ID |
| hospital\_id | varchar | 64 | Not Null |  | 所属医院ID |
| name | varchar | 10 | Not Null |  | 科室名称 |
| code | varchar | 120 | Not Null |  | 科室代码 |
| category | varchar | 10 | Not Null |  | 科室类别 |
| description | varchar | 10 | Not Null |  | 科室描述 |
| department\_head | varchar | 10 | Not Null |  | 科室负责人 |
| contact\_number | varchar | 200 | Null |  | 科室电话 |
| location | varchar | 100 | Not Null |  | 科室位置 |
| specialties | varchar | 20 | Not Null |  | 科室服务内容 |
| working\_hours | varchar | 50 | Not Null |  | 科室工作时间 |
| status | varchar | 10 | Not Null |  | 科室状态 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 就诊记录表(t\_medical\_record)

就诊记录表用于存储病患的就诊记录基础信息，包含就诊记录ID、患者ID、科室ID、医生ID、就诊时间、就诊状态、数据创建时间、数据修改时间、数据版本。数据状态包括待就诊、就诊中、就诊结束、就诊取消，其表结构如表4-10所示。

**表4-10就诊记录表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| tmr\_id | varchar | 64 | Not Null | PK | 记录ID |
| patient\_id | varchar | 10 | Not Null |  | 患者ID |
| department\_id | varchar | 64 | Not Null |  | 科室ID |
| doctor\_id | varchar | 64 | Not Null |  | 医生ID |
| visit\_date | varchar | 10 | Not Null |  | 就诊时间 |
| status | varchar | 10 | Not Null |  | 就诊状态 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 就诊过程表(t\_medical\_process)

就诊过程表用于存储病患在就诊过程中产生的相关记录，包括就诊过程的记录ID、所属就诊记录ID、记录类型、具体内容、记录时间、记录创建时间、记录修改时间、数据版本。每一条就诊记录可以存储多条不同类型的就诊过程记录，记录类型包括病症描述、诊断结果、检查结果、处方、治疗方案，记录状态包括已上链和未上链。其表结构如表4-11所示。

**表4-11就诊过程表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| tmp\_id | varchar | 64 | Not Null | PK | 记录ID |
| tmr\_id | varchar | 10 | Not Null |  | 就诊记录ID |
| status | varchar | 10 | Not Null |  | 记录状态 |
| record\_type | varchar | 64 | Not Null |  | 就诊类型 |
| record\_value | varchar | 64 | Null |  | 记录内容 |
| pharmacy | varchar | 64 | Not Null |  | 药房ID |
| sign | varchar | 200 | Null |  | 签名 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 加密策略表(t\_ctypto\_exp)

加密策略表用于存储系统参考加密策略信息，包括策略ID、策略名、策略说明、策略表达式、授权机构列表、策略状态、创建时间、修改时间、数据版本。策略状态包括关闭、启用。其表结构如表4-12所示。

**表4-12加密策略表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| tce\_id | varchar | 64 | Not Null | PK | 策略ID |
| name | varchar | 10 | Not Null |  | 策略名称 |
| description | varchar | 10 | Not Null |  | 策略说明 |
| exp | varchar | 64 | Not Null |  | 策略表达式 |
| auth | varchar | 64 | Null |  | 授权机构列表 |
| status | varchar | 10 | Not Null |  | 策略状态 |
| create\_at | varchar | 26 | Not Null |  | 创建时间 |
| update\_at | varchar | 26 | Not Null |  | 修改时间 |
| version | int |  | Not Null |  | 数据版本 |

1. 取药单据记录表(t\_dispensing)

取药单据记录表用于存储病患与药房之间的取药单据记录，包括记录ID、处方ID、药房ID、规定时间、取药状态。取药状态包括待取药、已取药。其表结构如表4-13所示。

**表4-13取药单据记录表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许为空 | 是否为主键 | 字段描述 |
| td\_id | varchar | 64 | Not Null | PK | 记录ID |
| prescriptionID | varchar | 10 | Not Null |  | 处方ID |
| pharmacyID | varchar | 64 | Not Null |  | 策略表达式 |
| time | varchar | 64 | Null |  | 预约时间 |
| status | varchar | 10 | Not Null |  | 取药状态 |

# 系统实现

前几章对✕✕✕✕管理系统的需求分析和系统设计作了详细分析，本章将阐述✕✕✕✕管理系统中主要模块的具体实现，主要是实现代码和实现过程分析。采用代码结合注解的方式展开分析。

## 后端功能实现

本系统中后端功能主要是为前端页面提供数据，后端的实现实际是各种接口的编码实现。系统中大量用到了jar包中的函数，Springboot框架通常是通过依赖来引用各类jar包，在pom.xml中标明jar包的坐标即可得到相应的jar包。比如为token生成和验证提供函数需要用到jwt包，它的坐标是io.jsonwebtoken，因篇幅所限，在pom.xml中添加lombok、mybatis-plus、mysql-connector-j等的依赖就不在此赘述。

本系统主要包括以下八个接口，以下是各接口详细介绍。

### 登录接口以及加密解密技术应用

根据输入的用户名和密码，通过UserDao，依靠BaseMapper，通过mybatisplus链接数据库，查询得到数据。

由于UserDao集成自 BaseMapper类。而BaseMapper类来自mybatisplus，所以UserDao可以通过mybatisplus连接数据库，操作制定的数据表。

通过注解@Select("SELECT \* FROM user where username=#

{username\_param} and password=#{password\_param}")来实现自定义SQL查询，从user 表中查询用户名和密码吻合的记录。

部分核心代码如下：

List<User> users = userDao.findUsersByNamePassword(username, password);

@Mapper

public interface UserDao extends BaseMapper<User>{

@Select("SELECT \* FROM user where username=#{username\_param} and password=#{password\_param}")//自定义SQL

public List<User> findUsersByNamePassword(@Param("username\_param") String username, @Param("password\_param") String password);

}

如果用户名和密码合法，生成token。

token = JwtUtil.genToken("AppId18910755", secret, "wwq");

返回token数据给前端，今后前端需要想后端获取数据都要携带这个token，后端得到这个token先要验证，如果合法就返回所要的携带时间戳的数据，不合法或者过期，就拒绝提供数据。

根据用户名和密码获取token的详细实现代码如下：

if (!(username==null || password==null))

{if (!(username.isEmpty()||password.isEmpty()))

{List<User> users = userDao.findUsersByNamePassword(username, password);

if (users.isEmpty()) {

token = "非法用户";

status = 500;

message = "fail";

} else {

token = JwtUtil.genToken("AppId18910755", secret, "wwq");

status = 200;

message = "success";

}

}else {

token = "用户名或密码为空";

status = 500;message = "fail";

}

}else {

token = "用户名或密码没有提供";

status = 500;message = "fail";

}

hashMap.put("token", token);

int timestamp = (int) (System.currentTimeMillis() / 1000);//时间戳

hashMap.put("timestamp", timestamp);

hashMap.put("message", message);

hashMap.put("status", status);

return hashMap;

}

## 前端功能实现

本系统中前端页面主要是完成UI展示，获取用户数据传递给后端接口，从接口获取数据展示给用户。

### 登录模块

这个模块主要是用来获取用户登录账号和密码，通过验证码拦截部分非法暴力登录请求。通过axios向后端请求验证。如果验证通过，把得到的token保存在session中，以便后期向后端接口请求数据时携带token作权鉴。

这部分主要是写在template中的vue+ElementPlus代码，实现以上UI。其中对账号和密码要做数据合法性验证和数据格式验证。界面如图5-2所示。

**图5-2 用户登录**

采用异步方式绘制验证码，同时做验证码校验。

login: async () => {

if (verifyCode.value != verifyCodeInput.value) {

ElMessage.error("验证码不正确！");

return ;

}

从后端获取Token采用axios方法，本系统中已经将axios做了封装，只需要提供请求名和用户名及密码参数即可。具体实现方法如下：

await myaxios.get('getToken', {params: data.loginForm}).then(r => {}

保存token采用sessionStorage来实现。

window.sessionStorage.setItem("token", r.data.token);

验证码的绘制采用图片形式，通过生成随机数，随机颜色绘制图片，随机确定填充颜色和填充的位置，设置干扰线和干扰点，随机产生字符串，并且随机旋转，随机设置字体大小，随机设置字体旋转角度，实现过程解决了字符间距过大,导致显示不全的问题。用户读取系统生成的验证码，输入到编辑框，系统将用户输入的验证码与之前生成的验证码做比对，不一致则拒绝向服务器发起验证请求。用户点击图片可以重新绘制验证码。

# 系统测试

本章将介绍本系统测试环境、各模块功能测试、系统性能测试情况。

## 系统测试环境

### 服务器端

1.硬件配置

服务器端的硬件配置，如表6-1 所示。

**表6-1 服务器端的硬件配置表**

|  |  |
| --- | --- |
| 硬件 | 参数 |
| CPU | Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz  1.19 GHz |
| 内存 | 8GB |
| 显卡 | Inter(R) UHD Graphics |
| 硬盘 | 512GB |

2.软件配置

服务器端的软件配置，如表6-2 所示。

**表6-2服务器端的软件配置表**

|  |  |
| --- | --- |
| 软件 | 参数 |
| 系统 | Windows Server 2023 |
| MySQL 数据库 | 8.0.30 |
| Navicat | 11.0.17 |
| Spring Tool Suite4 | 4.3.1 |
| JDK 版本 | 1.8.0 |

### 客户端

1.硬件配置

客户端的硬件配置，如表6-3所示。

**表6-3 客户端的硬件配置表**

|  |  |
| --- | --- |
| 软件 | 参数 |
| CPU | Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz  1.19 GHz |
| 内存 | 8GB |
| 显卡 | Intel(R) UHD Graphics |
| 硬盘 | 512GB |

2.软件配置

客户端的软件配置，如表6-4所示。

**表6-4客户端的软件配置表**

|  |  |
| --- | --- |
| 软件 | 参数 |
| 系统 | Windows11 |
| 浏览器 | Edge 浏览器、谷歌浏览器、火狐浏览器 |
| 防病毒软件 | 360 杀毒软件 |

## 系统功能测试

1.注册登录测试

测试用户注册登录功能的正确性。测试结果见表6-5。

表6-5 用户注册登录页面测试表。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 操作 | 预期结果 | 测试结果 |
| 1 | 点击注册，输入新的用户名、密码、昵称 | 注册成功 | 注册成功 |
| 2 | 点击登录，输入注册成功的用户名、密码 | 登录成功 | 登录成功 |
| 3 | 使用未注册的用户名登录或者不填写用户名 | 提示用户不存 在，登录失败 | 登录失败 |
| 4 | 使用错误的密码或者不填写密码 | 提示密码错 误，登录失败 | 登录失败 |

按照以上形式，依次介绍各个模块的测试情况..........

## 系统性能测试

通过性能测试需求分析，确定了系统需要做性能测试的业务功能，分别是学生管理、教师管理、✕✕、✕✕、✕✕、✕✕✕✕、✕✕、✕✕、✕✕、✕✕、✕✕。使用Jmeter测✕✕✕✕✕✕✕✕系统性能。接下来主要介绍用户从登录到✕✕✕✕✕✕✕✕✕✕的业务流程的性能测试，✕✕✕✕✕✕✕✕✕✕✕✕系统的测试范围如表6-12 所示。

表6-12✕✕系统的测试范围

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能模块 | 业务功能 | TPS | 响应时间 |
| 1 | 登录模块 | 12 | 50 | 106ms |
| 2 | 学生管理 | 10 | 50 | 520ms |
| 3 | 教师管理 | 20 | 40 | 304ms |
| 4 | ✕✕ | 20 | 100 | 336ms |
| 5 | ✕✕ | 18 | 100 | 309ms |
| 6 | ✕✕ | 50 | 50 | 198ms |

# 展望

## 总结

本系统用于学院专业✕✕✕✕管理，系统的主要功能是辅助学院实现日常✕✕✕✕管理，帮助学院在各类专业评估认证过程中得到状态数据，用于自我评价，逐步减小与评估要求的差距，提高人才培养质量。从师资队伍、教学条件、课程与教学、合作与实践、学生发展等多方面，从学院日常运行中收集一手数据，为专业认证和评估等工作提供有力支撑。

学生管理，教师管理，科研管理，竞赛管理，资产管理，考证管理，教学管理，质量监控调查问卷等模块以web形式运行，减少了数据收集的工作量，提高了学院教学管理效率。从实际运行情况来看，以上各个功能模块都基本上达到了设计要求。实现了全方位对✕✕✕✕数据收集和辅助学院教学管理要求。

从用户使用、系统性能测试和可靠性分析等方面来看，本系统都达到了设计要求。

本系统的架构设计、编码、调试、部署运行等方面的工作均由本人单独完成，在整个工作过程中得到了各方面的支持，前端使用Vue过程中，从2.0到3.0的学习和编码中得到了很大的提升，对elementPlus也有了更深刻的认识。后端使用Java EE和springboot框架，参考了阿里的大量技术资料，接口编写技术得到了进一步提高。

## 展望

根据管理要求和专业技术发展情况来看，后期可在如下几个方面做进一步的工作：

1.在学生管理和教师管理模块加入人脸识别技术，更进一步提高工作效率。

2.在系统中进一步加入区块链技术的应用，使系统朝分布式、弱中心化方向发展。终极目标是建成一个联盟链模式的系统。

3. 本系统是为帮助学院在✕✕✕✕数据评估和认证中及时了解基本状态，反映✕✕✕✕数据信息，后期有必要分析模块中加入类似“教育部✕✕✕✕数据分析报告”之类的分析报告。

本人将与时俱进进一步完善✕✕✕✕管理系统。

参考文献

[1]覃凤彩. 应用型高校教学基本状态数据库的构建——以南宁学院为例[J]. [广西教育](https://navi.cnki.net/knavi/journals/GXJY/detail?uniplatform=NZKPT" \t "https://kns.cnki.net/kcms2/article/_blank).

[2020,1(27](https://navi.cnki.net/knavi/journals/GXJY/issues/pbvOL2cnFbIjH9DqLYMriU1j6R1tJ2qVVg3OHgIPai_e3UTG_nqimkVb8On8l6d7?uniplatform=NZKPT" \t "https://kns.cnki.net/kcms2/article/_blank)):12-14

[2]沈镱武. 教务管理系统在高校的现状分析研究[J]. [电脑知识与技术](https://navi.cnki.net/knavi/journals/DNZS/detail?uniplatform=NZKPT" \t "https://kns.cnki.net/kcms2/article/_blank), 2022,18(14):125-126

[3]万为清. 基于遗传算法的省电脑知识赛训练系统的设计与实现[D]. 南昌:南昌大学, 2012.

[4]林智鑫. 跨域访问控制研究[D]. 武汉:武汉理工大学, 2008.

[5]Dai, Jun,[Vasarhelyi, Miklos A.](https://s.wanfangdata.com.cn/paper?q=%E4%BD%9C%E8%80%85:"Vasarhelyi, Miklos A."" \t "https://d.wanfangdata.com.cn/periodical/_blank).[Toward blockchain-based accounting and assurance](https://d.wanfangdata.com.cn/periodical/258ba0f0c858bdc3ee50621d51aa50e0" \t "https://d.wanfangdata.com.cn/periodical/_blank)[J].Journal of information systems.2017,31(3):5-21.

[6]张中祥. 区块链技术在会计信息系统中的应用[J]. 经济师, 2022,1(03):76-78

[7]袁月. 职业技能培训管理系统的设计与实现[D]. ,贵阳:贵州大学,2021

[8]郭浩. 体现CDIO理念的计科专业实践教学方式探讨[J]. [教育教学论坛](https://kns.cnki.net/kns8/Navi?DBCode=CJFD&BaseID=JYJU" \t "https://kns.cnki.net/kns8/defaultresult/_blank),2017,1(10):169-171

[9]吴新华. 高职院校教师科研业绩管理系统设计[J]. 价值工程, 2017,36(36):165-166

[10]蔺跟荣. 高等教育质量检测平台的数据分析与挖掘研究[D]. 西安:西北大学, 2017.

[11]Liu A, Du X, Wang N, Li S. Blockchain technology and its research progress in the field of information security[J]. Journal of Software, 2018, 29(07): 2092-2115.

[12]严竞雄. 基于ASP.NET的高校辅导员工作管理系统的设计与实现[J]. [电脑知识与技术](https://navi.cnki.net/knavi/journals/DNZS/detail?uniplatform=NZKPT" \t "https://kns.cnki.net/kcms2/article/_blank). [2017,13](https://navi.cnki.net/knavi/journals/DNZS/issues/pbvOL2cnFbKhAJ4SW1ikpbfvtobGv1jS57EyMiUpEdcyZ-ih5370EDFRY5vh4nn3?uniplatform=NZKPT" \t "https://kns.cnki.net/kcms2/article/_blank)(36):55-57

[13]冯川. 基于J2EE的高校学生基本信息分布式系统的设计[D]. 青岛:中国海洋大学, 2012.

[14]盛卓. 泽达职业技术学院学籍管理系统设计与实现[D]. 沈阳:东北大学, 2013.

[15]向容. 校园信息管理系统的研究与开发实践[D]. 杭州:浙江工业大学, 2012.

[16]李桂花. 中职学生成绩管理系统的设计与实现[D].长沙:湖南大学, 2015.

[17]赵亮,刘建国,陈志奎,,基于JavaWeb的实验室管理系统设计与实现[J],实验室研究与探索. 2022,41(08):283-287.

[18]吴志锋,柯崇榕,吴敏霞,俞如旺. 基于Web的实验教学中心管理系统设计[J]. 实验室研究与探索, 2010,29(11):366-368

[19]朱玉艳. 学生评语系统的设计与实现[J]. 湖南工程学院学报(自然科学版), 2013,23(03):38-40

[20]程玉健. 山东信校教师教学综合评价系统的设计与实现[D]. 成都:电子科技大学, 2010.

在校期间取得的成果

2022年11月获得2022一带一路暨金砖国家技能发展与技术创新大赛区块链智能合约开发与应用赛项全国总决赛一等奖。

致谢

在我的导师XXXX和XXXX耐心指导下，本论文和系统项目开发的全部工作已经顺利完成。从论文研究、技术架构选型、确定论文结构、结合软件项目完成论文并定稿，老师都及时跟进并关注我的论文进展。他们学识渊博、工作方法科学、治学作风严谨、师德崇高，方方面面都给了我很大的影响，使我受益良多，令我终生难忘。在我的毕业论文工作即将画上句号之际，衷心感谢两位老师对我的指导和帮助。

感谢我的家人和每一位关心和帮助过我的人，使我能顺利完成学业和论文。

感谢参与评审我论文的各位专家，希望大家不吝赐教。

附录

附录1：系统运行界面

登录模块