**中北大学软件学院毕业论文**



|  |  |
| --- | --- |
| **论文题目:** | **基于SpringBoot+HarmonyOS的餐饮管理与服务系统的**  **设计与实现** |
| **学生姓名:** | **冯佳和** |
| **学 号：** | **2113042621** |
| **所属院系：** | **软件学院** |
| **专 业：** | **软件工程** |
| **班 级：** | **21130416** |
| **指导老师:** | **卢鑫培** |
| **日 期：** | **2025年 05 月20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： 2025年05月20日

摘 要

随着国内外卖行业的蓬勃发展，外卖平台间的竞争影响了许多餐饮企业。许多餐饮企业面临着第三方外卖平台抽佣高、营销策略受限等问题，导致利润降低甚至亏本。针对这一问题，本论文提出了开发一个开放的外卖系统的想法，旨在使餐饮企业能够直接面向客户，摆脱对第三方平台的依赖。

通过调研和分析，本论文将重点关注中小型餐厅，以城区为服务对象，结合市场特点和消费者行为，开发一个适合这一需求的开放型外卖系统。通过该系统，中小型餐饮企业能够减少对第三方外卖平台的依赖，降低成本，提高利润。同时，消费者也能够享受到更多优质、多样化的餐饮选择和个性化的服务。

本项目选择Java语言进行开发，系统角色包括管理员和用户，管理员可以管理员工信息、分类管理、菜品管理、营业数据统计以及运营数据导出等。用户可以查看菜品信息、管理购物车、管理地址、修改个人信息等。项目经过需求分析、功能模块设计和数据库设计后，详细描述了各个功能模块的实现方式。本毕业设计前端使用HTML 5、AJAX、Vue、Element-UI框架实现，后端使用Spring、Spring MVC、Spring Boot、MyBatis实现，数据库采用MySQL，缓存使用Redis实现，项目部署使用Linux系统和Tomcat。

**关键词**：外卖订餐系统；HTML5；AJAX；Vue；Spring Boot

ABSTRACT

With the booming development of the domestic food delivery industry, many catering companies are faced with problems such as high commissions from third-party food delivery platforms and limited marketing strategies, resulting in reduced profits or even losses. To solve this problem, this paper proposes the idea of ​​developing an open takeout system suitable for small and medium-sized restaurants, aiming to enable catering companies to face customers directly, get rid of dependence on third-party platforms, reduce costs and increase profits. Through research and analysis of market characteristics and consumer behavior,

this project will use Java language to develop the system. The front end will use HTML 5, AJAX, Vue, and Element-UI framework, and the back end will use Spring, Spring MVC, Spring Boot, and MyBatis. The database uses MySQL and the cache uses Redis. The roles of this system include administrators and users. Administrators can perform operations such as employee management, category management, dish information management, business data statistics, and operational data export. Users can view dish information, manage shopping carts, addresses, and personal information. After the project went through requirements analysis, functional module design and database design, the implementation of each functional module was described in detail. Finally, the system will be deployed on Linux systems and Tomcat. Through this system, small and medium-sized catering companies can better control their own operations and provide consumers with more high-quality, diversified catering options and personalized services.

**KEY WORDS:** Takeway ordering system; HTML 5; AJAX;Vue;Spring Boot

目 录

[1 绪论 1](#_Toc6009)

[1.1 研究背景和意义 1](#_Toc18760)

[1.2 国内外研究现状 2](#_Toc28868)

[1.2.1 国内研究现状 3](#_Toc31499)

[1.2.2 国外研究现状 3](#_Toc29620)

[1.3 研究目标和内容 4](#_Toc13864)

[1.4 论文组织框架 5](#_Toc20653)

[2 开发技术及开发环境 7](#_Toc26096)

[2.1 开发技术 7](#_Toc6244)

[2.1.1 Vue简介 7](#_Toc2109)

[2.1.2 Axios简介 7](#_Toc8665)

[2.1.3 Java简介 7](#_Toc7359)

[2.1.4 Spring Boot简介 8](#_Toc22319)

[2.1.5 MyBatis简介 9](#_Toc15268)

[2.1.6 MySQL简介 9](#_Toc21349)

[2.1.7 Maven简介 10](#_Toc31603)

[2.1.8 JDK简介 10](#_Toc14664)

[2.1.9 MD5简介 11](#_Toc4602)

[2.1.10 ECharts简介 11](#_Toc29886)

[2.2 开发环境 11](#_Toc23447)

[3 系统分析 13](#_Toc14481)

[3.1设计思想 13](#_Toc28735)

[3.2 设计原则 13](#_Toc5078)

[3.3 可行性分析 14](#_Toc26082)

[3.3.1 技术可行性 14](#_Toc9016)

[3.3.2 经济可行性 14](#_Toc21237)

[3.3.3 市场可行性 15](#_Toc17904)

[3.3.4 操作可行性 15](#_Toc31259)

[3.4 需求分析 16](#_Toc15157)

[3.4.1 功能性需求 16](#_Toc4166)

[3.4.2 非功能性需求 20](#_Toc29795)

[4 系统概要设计 23](#_Toc22736)

[4.1 系统架构设计 23](#_Toc2349)

[4.2 系统功能设计 24](#_Toc28499)

[4.2.1 个人信息模块 24](#_Toc26266)

[4.2.2 健康档案管理 24](#_Toc5451)

[4.2.3 主界面模块 26](#_Toc22167)

[4.2.4 体检数据分析模块 26](#_Toc18890)

[4.3 系统业务流程设计 27](#_Toc11616)

[4.3.1 个人信息模块 27](#_Toc25538)

[4.3.2 健康档案管理模块 28](#_Toc4893)

[4.3.3 主页面 32](#_Toc27574)

[4.3.4 体检数据分析 33](#_Toc21338)

[4.4 数据库ER模型 34](#_Toc6094)

[4.4.1 整体ER图 34](#_Toc10565)

[4.4.2 局部ER图 35](#_Toc9628)

[5 系统详细设计 37](#_Toc30421)

[5.1 系统功能模块设计 37](#_Toc26398)

[5.1.1 个人信息模块 37](#_Toc1616)

[5.1.2 健康档案管理模块 37](#_Toc17603)

[5.1.3 主界面模块 38](#_Toc12483)

[5.1.4 体检数据分析模块 38](#_Toc17517)

[5.2 数据库表设计 38](#_Toc22883)

[6 系统实现 44](#_Toc26067)

[6.1 系统主界面 44](#_Toc27903)

[6.1.1 系统主界面效果图 44](#_Toc26289)

[6.1.2 系统主界面实现 46](#_Toc19940)

[6.2 个人信息界面 47](#_Toc21922)

[6.2.1个人信息界面效果图 47](#_Toc6105)

[6.2.2 个人信息界面实现 47](#_Toc20630)

[6.3 健康档案管理界面 48](#_Toc17703)

[6.3.1 健康档案管理界面效果图 48](#_Toc29770)

[6.3.2 健康档案管理界面实现 50](#_Toc17614)

[6.4 体检数据分析界面 52](#_Toc27690)

[6.4.1 体检数据分析界面效果 52](#_Toc21322)

[6.4.2 体检数据分析界面实现 53](#_Toc20095)

[7 系统测试 55](#_Toc12951)

[7.1测试方法 55](#_Toc3802)

[7.2 系统功能模块测试 55](#_Toc861)

[7.2.1个人信息模块测试 55](#_Toc24863)

[7.2.2健康档案管理模块测试 56](#_Toc20582)

[7.2.3 主界面模块测试 57](#_Toc31133)

[7.2.4 体检数据分析模块测试 58](#_Toc9926)

[7.3 系统性能测试 59](#_Toc15115)

[7.4 测试分析和总结 60](#_Toc10348)

[8 结论与展望 62](#_Toc10607)

[参考文献 64](#_Toc13284)

[致 谢 65](#_Toc28982)

# 绪论

随着国民经济的快速发展，人民消费水平不断提高，促使了餐饮行业的繁荣。尤其是学生和上班族这些群体，他们忙碌的学习和工作使得他们更加注重节约时间。现在人们对食物的品种和味道的要求越来越高，同时也对如何方便快捷地订餐提出了更高的要求。因此，外卖行业应运而生，旨在节约用餐时间，提供高效的服务。市场上出现了很多第三方外卖平台，比如美团外卖、饿了么外卖、百度外卖等。许多餐饮店也开始依赖于这些外卖平台，但同时也带来了一些新问题。例如，平台之间的竞争会影响到商家的收益，平台收取的佣金较高，要求商家降低利润做活动，同时还存在营销策略受限的问题等。因此，许多大型和连锁餐饮企业选择不依赖第三方外卖平台，而是独立开发一套自己的餐饮外卖系统直接面向用户。然而，中小型餐厅由于资金和技术问题，仍然采用传统的订餐方式。通过调查发现，中小型餐厅将餐饮与网络应用结合的程度并不普遍，因此提出了区域性餐饮网络营销的想法，即以城区中小型餐厅为服务对象，结合市场特点和消费者购物行为，开发一个开放型餐饮服务平台，对于解决餐饮企业问题和为市民提供服务具有重要意义。

通过分析中小型餐饮痛点，决定以城区中小型餐厅为主体服务对象，结合城区周边住宅和学校园区的特点，开发出一套满足中小型餐饮需求的餐饮外卖系统。减轻餐饮店对第三方外卖平台的依赖和支出，实现更自由和更低成本的营销策略。减轻中小型餐厅的负担，提高收入。

本系统是面向所有人的餐饮外卖系统。系统用户角色可以分为管理员和用户。管理员可以登录后台管理端进行后台工作的管理，用户可以登录前台餐厅进行操作管理员可以管理员工信息、分类管理、导出分类信息数据表格、菜品信息管理、套餐信息管理、订单明细查询、订单管理、导出订单明细数据表格、修改个人信息等。用户可以查看菜品信息、查看菜品详情、选择菜品规格、管理购物车、收货地址管理、修改个人信息、查看订单信息、下单支付等。

本项目选择使用Java语言进行开发，Java语言具有可跨平台和可移植性的特点，使项目能在不同的系统上运行，解决兼容性问题。此次开发的餐饮外卖系统是面向所有人的，所以在前台采取页面简洁的方法，前台使用Vue和Element-UI组件，使用户能快速、简便的浏览网页，使得用户体验更加良好。后台全部使用轻量级的Vue前端框架，操作简便，功能齐全，界面简单而优美使得用户体验以及功能都能得到保障。在后端上选择使用较为火热的Spring Boot框架，并整合MyBatis-Plus框架，在前后端交互上选择使用JQuery+AJAX技术，AJAX 是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。总体来说，此次餐饮外卖系统的开发环境和运行环境需要的软件和工具比较复杂，但是它可以实现跨平台的应用程序开发和运行，具有很好的移植性和扩展性。

## 1.1 研究背景和意义(未完成)

在网络不发达的年代，大多数高校采用体检表的方式来保存学生的体检档案。这种方法存在许多弊端，如档案易损坏、检索不便、占用大量存储空间等。随着信息技术的发展，特别是B/S架构的普及，电子化管理逐渐成为可能。1968年，美国相关机构提出了以问题为导向的健康档案管理方式，这种方法已被大多数国家和地区所采用。历史早期的档案管理，如商代的甲骨文、周代的石刻板、汉朝的竹简等，都体现了人们对信息记录的需求。21世纪是信息产业发展迅速的时代，信息技术的发展带动了企业的发展，也对健康管理提出了新的要求。随着互联网技术的飞速发展，社会对电子化管理的需求日益增长。特别是在教育领域，大学生的健康管理成为了一个备受关注的议题。传统的纸质体检档案管理方式已经无法满足现代社会对效率和安全性的要求。纸质档案不仅容易损坏、丢失，而且在存储和检索上也存在诸多不便。此外，随着学生数量的增加，纸质档案的管理成本也在不断上升。因此，开发一个基于JavaEE的大学生健康档案管理系统显得尤为必要。

本系统对于大学生健康管理有着非凡的意义，如以下几点：

（1）提升管理效率：通过电子化管理，可以大幅度提高档案管理的效率，减少人工操作的时间和出错率。系统能够实现体检数据的快速录入、存储和检索，极大提升档案管理的工作效率。

（2）保障信息安全：与传统纸质档案相比，电子档案更易于进行加密和备份，从而增强了档案数据的安全性。通过权限管理，确保只有授权用户才能访问敏感信息，有效防止数据泄露和篡改。

（3）优化资源利用：电子化管理减少了对物理存储空间的需求，节约了大量的纸张和存储空间。同时，电子档案的云端存储和访问，使得资源利用更加灵活和高效。

（4）辅助决策制定：系统能够对体检数据进行快速分析和统计，为高校管理层提供科学、准确的决策支持。通过分析学生的健康状况，学校能够制定更为有效的健康促进计划和疾病预防策略。

（5）增强用户体验：对于学生和医务人员而言，基于JavaEE的管理系统提供了更为便捷和个性化的服务。学生可以随时随地访问自己的健康档案，医务人员也能够快速获取学生的体检信息，提供更为精准的健康建议。

（6）促进健康教育：系统可以集成健康知识阅读模块，发布相关的健康教育文章，方便学生获取健康相关信息，提高他们的健康意识和自我保健能力。这种教育方式不仅有助于提升学生的生活质量，也为培养健康的生活方式打下了基础。

通过本系统的研究与开发，不仅可以推动高校健康管理工作的信息化进程，也为其他领域的档案电子化管理提供了参考和借鉴。这一创新举措对于提升高校管理水平、保障学生健康以及推动健康教育具有重要的现实意义和深远的社会影响。

## 1.2 国内外研究现状(未完成)

总体来看，无论是国内还是国外，大学生健康管理系统的研究都在不断发展和进步。国内外研究者们都在积极探索如何利用现代信息技术，提高健康管理的效率和质量。国内的研究更多地集中在系统的开发和应用上，而国外的研究则更加注重服务的个性化和综合性。随着技术的不断进步和人们对健康重视程度的提高，未来的大学生健康管理系统将更加智能化、个性化和便捷化，为大学生提供更加全面和高效的健康管理服务。

### 1.2.1 国内研究现状(未完成)

随着国内经济的快速发展和人们生活水平的不断提高，健康问题逐渐成为社会关注的焦点，特别是在高校中，大学生的健康管理更是受到重视。高校通过组织定期体检，全面检测学生的身体状况，确保他们的健康状况良好，预防疾病的发生。对于学校来说，对学生体检的重视程度直接关系到学生的健康服务和身体健康。近年来，随着信息技术的发展，国内许多医院和学校开始采用电子文档来存储和管理体检档案，体检档案的电子化管理逐渐受到关注。传统的人工处理体检档案方式已经无法满足日益增长的信息量需求，因此，开发一个体检档案的管理系统显得尤为重要。体检档案管理系统不仅可以存储体检者的信息，而且能够快速规范地保存这些信息，保证信息的长时间可用性。国内在体检档案管理方面的研究主要集中在电子化管理、信息安全、系统集成、用户体验、功能扩展和技术应用等方面。随着互联网的发展，越来越多的机构开始采用电子化的方式来管理体检档案，这种方式不仅提高了档案管理的效率，也降低了档案丢失和损坏的风险。在电子化管理的过程中，信息安全成为一个重要的研究课题，如何保证体检档案的安全性和隐私性，防止数据泄露和被未授权访问，是当前研究的重点。此外，体检档案管理系统需要与医院或学校的其他信息系统进行集成，实现数据的共享和交换，这涉及到不同系统之间的接口设计和数据格式的统一。系统的易用性和用户体验也是研究的重点之一，如何设计一个直观、易操作的界面，让用户能够方便地管理和查询自己的体检档案，是提升系统使用率和满意度的关键。一些系统还开始尝试增加更多的功能，如健康建议、健康知识普及等，以提供更全面的健康管理服务。在技术应用方面，国内的研究者们也在不断探索新技术在体检档案管理中的应用，如云计算、大数据、人工智能等，以提高系统的智能化水平和处理能力[2]。总的来说，国内在大学生体检档案管理系统的研究和应用方面已经取得了一定的进展，但仍有很大的发展空间和提升潜力。随着技术的不断进步和人们对健康重视程度的提高，未来的体检档案管理系统将更加智能化、个性化和便捷化。

### 1.2.2 国外研究现状(未完成)

随着全球范围内对健康问题关注度的提升，国外在大学生健康管理系统方面的研究和实践已经取得了显著进展。20世纪中叶以来，互联网的发展极大地推动了健康管理服务的变革。在国外，健康管理服务已经从最初的单一体检和生活指导，发展到现在的群体综合体检、健康风险评估和国家健康促进计划。这些服务不仅覆盖了个体的健康检查，还扩展到了健康教育、疾病预防和健康促进等多个方面。国外许多国家和地区已经建立了完善的居民健康档案系统，这些系统通常采用以问题为导向的管理方式，要求在健康服务过程中，医护人员要采用以个体健康问题为导向的方法。这种方法已被广泛采用，并在实践中显示出其有效性。此外，国外在体检档案信息化管理方面也取得了重要进展，顺应了时代潮流。国外研究者们在体检档案管理系统的开发上，注重系统的多功能性和用户体验。他们通常采用先进的信息技术，如云计算、大数据和人工智能，来提高系统的智能化水平和处理能力。这些系统不仅能够存储和管理体检者的信息，还能够提供快速规范的信息保存，保证信息的长时间可用性。在系统架构方面，国外研究者们倾向于使用B/S（浏览器/服务器）架构，这种架构使得体检信息可以在不同系统的设备上访问，如安卓、iOS、Windows等。通过建立体检档案数据库并与医院或学校的计算机实现联网，可以实现对体检档案的快速更新，从而提高健康管理的效率和准确性。此外，国外在健康管理服务的提供上，也更加注重个性化和综合性。他们通过分析体检者的体检信息，向体检者提出合理的健康建议，进行健康观察分析，并快速诊断出现的身体状况。这种服务模式不仅提高了健康管理的科学性，也增强了服务的针对性和有效性。总体来看，国外在大学生健康管理系统的研究和应用方面已经形成了一套成熟、系统的服务体系，这些研究成果和实践经验为其他国家提供了宝贵的参考和借鉴。随着技术的不断进步和人们对健康重视程度的提高，未来的健康管理服务将更加智能化、个性化和便捷化。

## 1.3 研究目标和内容

在数字化时代背景下，大学生体检档案管理系统的开发已成为提升高校管理效率和保障学生健康的重要手段。随着互联网技术的普及，传统的纸质档案管理方式已无法满足现代教育的需求，因此，本研究旨在开发一个基于JavaEE的大学生体检档案管理系统，以解决现有管理方式中存在的问题。该系统将采用JavaEE技术，结合IDEA开发平台和Spring Boot框架，实现体检档案的电子化管理，提高档案管理的效率和安全性。

本研究的内容涵盖了从系统需求分析到最终实现的全过程。首先，通过深入分析大学生体检档案管理的现有问题和需求，确定了系统应具备的功能，包括用户管理、健康档案管理、健康建议管理、健康知识阅读管理、角色管理和权限管理等。系统设计阶段，将采用面向对象的设计方法，构建系统的总体架构，并设计数据库模型，确保系统的数据存储和检索效率。在技术选型方面，本系统将使用Java语言作为开发语言，利用IDEA作为开发工具，前端采用Vue框架，后端采用Spring Boot框架，通过MyBatis与MySQL数据库进行交互，实现系统的业务逻辑。

系统开发过程中，将注重用户体验，设计一个用户友好的操作界面，满足不同用户（如管理员、医生、学生等）的操作需求。同时，系统将实现体检档案的增删改查功能，支持个性化健康建议，以及灵活的权限分配，确保系统的安全性和可操作性。在系统测试阶段，将对各功能模块进行全面测试，确保系统功能的正确性和稳定性，及时发现并修复潜在的错误。最终，系统将部署到服务器上，并进行必要的维护和优化，确保系统在实际运行中的稳定性和可靠性。

此外，本研究还将提供用户培训和编写相关文档，帮助用户快速掌握系统使用方法，并为系统的长期运行提供支持。通过这些研究内容的实施，本项目将为高校提供一个高效、安全、便捷的大学生体检档案管理系统，推动高校健康管理工作的信息化进程，提升高校管理水平和服务质量。这不仅能够减轻管理人员的工作负担，也能为大学生提供便捷的健康信息查询和个性化健康建议服务，具有重要的应用价值和广阔的应用前景。

## 1.4 论文组织框架

论文首先介绍了大学生健康档案电子化管理系统的研究背景和意义，强调了在数字化时代，电子化管理对于提升高校管理效率和保护学生健康的重要性。接着，论文详细阐述了开发技术和开发环境，包括所采用的Vue框架、Java语言、Spring Boot框架、MyBatis以及MySQL数据库等关键技术，并解释了这些技术在项目中的应用。

在系统分析阶段，论文进行了技术可行性、经济可行性和市场可行性的全面分析，确保了项目的实施基础。需求分析部分深入探讨了系统所需满足的功能和非功能需求，为系统设计提供了明确的指导。系统概要设计和详细设计部分则分别描述了系统的架构、功能、业务流程以及数据库设计，为实现阶段打下了坚实的基础。

实现部分通过界面截图和代码示例，具体展示了系统的主要功能模块，包括个人信息管理、健康档案管理、体检数据分析等。系统测试章节通过一系列测试用例，验证了系统的功能和性能，此外，论文还包括了参考文献的规范列举和对在研究过程中提供帮助的个人和集体的致谢。

最后，论文在结论部分总结了研究成果，并对未来的发展方向提出了建议和期望。

# 开发技术及开发环境

## 2.1 开发技术

### 2.1.1 Vue简介

Vue是一个用于构建用户界面的渐进式JavaScript框架。它以其易用性、灵活性和高效性而广受欢迎。Vue能够极大地简化前端开发的复杂性，使得开发者能够更专注于构建富有交互性和动态效果的用户界面[3]。Vue通过其独特的响应式数据绑定系统，让数据与视图之间保持自动同步，这一特性极大地减少了开发者在DOM操作上的负担，同时也提升了应用的性能和用户体验。随着Web技术的日益成熟和前端应用复杂度的增加，Vue也在不断演进，以适应更广泛的需求和挑战。除了持续的性能优化和新特性的引入外，Vue还注重与其他现代前端技术的集成，以满足不同场景下的开发需求。

### 2.1.2 Axios简介

Axios 是一种 JavaScript 的 HTTP 客户端，它遵循 Promise API，适用于浏览器和 node.js。Axios 不是一种新技术，而是对原生 XMLHttpRequest（XHR）对象的封装，并基于 Promise 实现，使其更符合现代 JavaScript 的异步编程模式。

### 2.1.3 Java简介

Java 是一种广泛使用的高级编程语言，由 Sun Microsystems（现为 Oracle 公司的一部分）于 1995 年首次发布。它被设计为具有尽可能少的实现依赖性，这意味着 Java 编写的应用程序可以在任何支持 Java 虚拟机（JVM）的平台上运行，这被称为“一次编写，到处运行”（Write Once, Run Anywhere，WORA）的特性。核其心特性如下：

（1）面向对象:Java 是一种面向对象的语言，支持封装、继承和多态等基本概念。这使得代码更加模块化、易于理解和维护。

（2）平台无关性:Java 代码被编译成字节码，而不是直接编译成机器码。字节码可以在任何安装了 JVM 的设备上运行，从而实现了跨平台兼容性。

（3）多线程:Java内置了对多线程编程的支持，使得编写并发程序更加容易。它提供了线程管理、同步和通信等机制。

（4）安全性:Java提供了一个安全管理器，允许应用程序定义其安全策略。它还通过沙箱（Sandbox）机制限制了代码对系统资源的访问，增加了程序的安全性。

（5）可移植性：Java代码可以在任何支持JVM的平台上运行，无需修改。这使得 Java 成为开发跨平台应用程序的理想选择。

（6）分布式：Java 支持网络通信和远程调用，使得开发分布式应用程序变得容易。它提供了RMI（远程方法调用）和CORBA（公共对象请求代理体系结构）等技术。

### 2.1.4 Spring Boot简介

Spring Boot是一个开源的Java基础框架，用于创建独立、生产级的基于Spring框架的应用程序。它简化了基于Spring的应用程序的初始搭建和开发过程，使得开发者可以快速启动和运行Spring应用程序[4]。以下是Spring Boot的详细介绍：其核心特性如下：

（1）自动配置:Spring Boot的核心特性之一是自动配置。它通过类路径中的类来自动配置 Spring 应用程序。例如，如果类路径中存在 H2DataSource，Spring Boot 会自动配置内存数据库。

（2）独立运行:Spring Boot 应用程序包含内嵌的 HTTP 服务器（如 Tomcat、Jetty 或 Undertow），使得应用程序可以打包成一个独立的 JAR 文件，通过简单的 java -jar 命令运行。

（3）微服务支持:Spring Boot 与 Spring Cloud 等微服务框架无缝集成，支持微服务的快速开发和部署。

（4）社区支持:Spring Boot 拥有一个庞大的社区和生态系统，提供了大量的插件和扩展，使得开发者可以轻松地扩展应用程序的功能。

（5）测试支持:Spring Boot 提供了对 Spring Test 和 JUnit 的支持，使得编写和运行单元测试和集成测试变得容易。

（6）依赖管理:Spring Boot 使用 Maven 或 Gradle 作为构建工具，并提供了spring-boot-starter依赖，简化了依赖管理。

### 2.1.5 MyBatis简介

MyBatis 是一个开源的持久层框架，用于简化数据库操作和简化数据库事务管理。它是一个半自动的持久层框架，介于全自动化的 Hibernate 和完全手动编写 SQL 语句之间。MyBatis 允许开发者编写 SQL 语句，并将其与 Java 对象映射，从而实现数据的持久化[5]。其核心特性如下：

（1）SQL 映射:MyBatis 允许开发者编写 SQL 语句，并将其映射到 Java 对象。这种方式使得 SQL 语句的编写和维护更加灵活和直观。

（2）性能:MyBatis 的性能通常优于全自动化的 ORM 框架，因为它允许开发者编写优化的 SQL 语句，并且减少了不必要的数据库交互。

（3）可扩展性:MyBatis 支持插件机制，允许开发者扩展其功能。例如，可以通过插件实现分页、缓存等高级功能。

（4）事务管理:MyBatis 支持声明式事务管理，可以通过配置文件或注解来管理事务的边界和行为。

（5）动态 SQL:MyBatis 支持动态 SQL，允许开发者根据条件动态生成 SQL 语句。这使得处理复杂的查询条件变得更加简单。

（6）集成 Spring:MyBatis 可以与 Spring 框架无缝集成，利用 Spring 的依赖注入和事务管理功能，简化了 MyBatis 的配置和使用。

（7）支持多种数据库:MyBatis 支持多种关系型数据库，如 MySQL、Oracle、SQL Server 等，使得它可以在不同的数据库环境中使用。

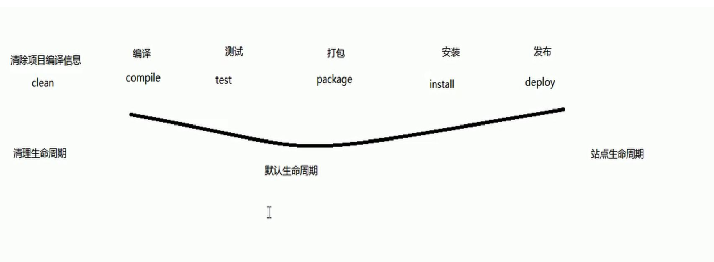
### 2.1.6 MySQL简介

MySQL是一个开源的关系型数据库管理系统，广泛应用于各种应用程序中。它以高性能、可靠性和易用性著称。MySQL使用表格形式存储数据，并支持SQL进行数据查询和管理。它适用于需要结构化数据存储和管理的场景。MySQL提供了多种存储引擎，如InnoDB和MyISAM，其中InnoDB支持事务处理和行级锁，适用于高并发读写操作的场景。MySQL支持主从复制、分区、集群等技术，可以扩展到大规模数据存储和处理需求。它可以通过水平和垂直扩展满足不断增长的业务需求。MySQL提供事务支持，包括原子性、一致性、隔离性和持久性，保证数据操作的可靠性和一致性。MySQL提供了用户权限管理、数据加密、审计日志等安全特性，保护数据库系统和数据的安全性。

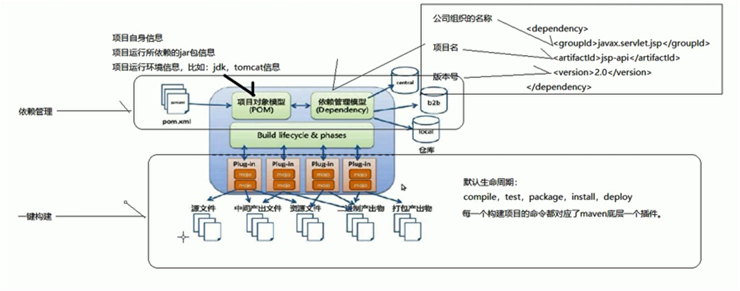
### 2.1.7 Maven简介

Maven 是一个由 Apache 软件基金会维护的项目管理和构建自动化工具，它主要用于 Java 项目。Maven 的设计目标是使得构建、测试和部署软件项目的过程自动化，并且简化管理项目依赖的复杂性。

Maven 使用一个名为 POM（Project Object Model）的 XML 文件来描述项目的构建过程、依赖关系和其他配置信息。POM 文件中定义了项目的基本信息，如项目名称、版本、开发者信息等，同时也列出了项目所需的依赖库和插件。这些依赖库可以是其他 Java 库，也可以是 Maven 插件，它们被存储在 Maven 的中央仓库中，Maven 会根据 POM 文件自动下载这些依赖。如图2-1、2-2。



**图2-1 Maven生命周期图**



**图2-2 Maven原理图**

### 2.1.8 JDK简介

JDK，全称为Java Development Kit（Java开发工具包），是由Oracle公司（之前是Sun Microsystems）发布的用于开发Java应用程序的一套软件开发环境。它提供了Java语言的核心API、编译工具、调试工具以及其他一些工具和库，使得开发者能够编写、编译和运行Java程序。

### 2.1.9 MD5简介

MD5（Message-Digest Algorithm 5）是一种广泛使用的加密散列函数，它可以产生一个128位（16字节）的散列值（hash value），通常用32位的十六进制字符串表示。MD5由 Ronald Rivest 在1991年设计，最初被用来作为一种安全的密码散列算法。MD5 通过一系列复杂的操作，包括位运算、位移、条件逻辑等，将任意长度的输入数据转换成固定长度的输出。

### 2.1.10 ECharts简介

数据可视化技术ECharts是一个基于JavaScript的开源可视化库，最初由百度团队开发并开源，后捐赠给Apache基金会成为ASF顶级项目。ECharts以其强大的兼容性和优异的性能，在前端数据可视化领域占据了重要地位。它提供了丰富的图表类型，如折线图、柱状图、散点图、饼图、地图、热力图、关系图等，满足了各种复杂的数据可视化需求。

ECharts的优势在于其灵活的配置项和强大的数据驱动能力。开发者可以通过简单的配置就能实现复杂的数据可视化效果，而且ECharts的图表具有高度可定制性，无论是图表的样式、颜色还是交互方式，都可以根据实际需求进行个性化设置。

此外，ECharts还具有良好的跨平台性，它可以在PC和移动设备上流畅运行，兼容当前绝大部分主流浏览器。这使得ECharts在Web开发中得到了广泛的应用，无论是数据统计分析、地理数据可视化还是关系数据可视化，ECharts都能提供有力的支持[6]。

## 2.2 开发环境

（1）网络环境：局域网。

（2）硬件环境：笔记本电脑、运存16G、硬盘512G。

（3）操作系统：Windows10。

（5）开发工具和集成开发环境：大学生健康档案管理系统的开发环境构建在一个稳定而高效的技术栈之上。系统采用 Windows操作系统作为基础运行平台，确保了开发过程的兼容性和稳定性。后端开发选择了 Java 语言，并以 JDK 1.8 版本为核心，利用其跨平台的特性和强大的社区支持。开发工具选择了 IntelliJ IDEA，这款集成开发环境以其智能代码补全、代码分析和版本控制功能著称，极大地提升了开发效率。前端技术栈则采用了 Vue.js，这是一个轻量级且易于上手的 JavaScript 框架，非常适合动态用户界面的构建。为了支持前端的运行，系统配置了 node.js 环境，这是一个基于 Chrome V8 引擎的 JavaScript 运行环境，能够提供高效的服务器端脚本执行能力。数据库方面，系统选择了 MySQL 作为数据存储解决方案，这是一个广泛使用的开源关系型数据库管理系统，以其高性能、可靠性和易用性著称。在后端框架的选择上，系统采用了 Spring Boot，这是一个简化 Spring 应用开发的工具，提供了自动配置、内嵌服务器等便利功能。此外，系统的开发过程中还涉及到了 Maven 的使用，这是一个项目管理和构建自动化工具，负责处理项目的依赖管理和生命周期管理。为了进一步简化 Java 代码的编写，系统还集成了 Lombok 插件，它通过注解的方式减少了样板代码。整个开发环境的配置旨在提供一个高效、稳定且易于维护的平台，以便开发团队能够专注于系统功能的开发和优化。见表2-1

**表2-1 开发环境表**

|  |  |
| --- | --- |
| 开发环境组件 | 描述 |
| 操作系统 | Windows |
| JDK版本 | 17 |
| 开发工具 | IntelliJ IDEA |
| 前端运行环境 | node.js |
| 数据库系统 | MySQL |
| 前端框架 | Vue.js |
| 后端框架 | Spring Boot |
| 版本控制系统 | Git |
| 项目管理工具 | Maven |
| 数据库操作框架 | MyBatis |
| 代码简化插件 | Lombok |

# 系统分析

## 3.1设计思想

采用分层架构：本系统采用经典的三层架构模式，明确划分为表示层、业务逻辑层和数据访问层。

系统按照功能或业务需求进行模块化划分，每个模块具有明确的功能和职责。这种设计使得系统结构更加直观，便于开发、测试和维护。每个模块可以独立开发和测试，提高了代码的复用性和系统的可测试性。模块化设计还允许团队成员分工合作，各自负责特定模块的开发，从而提升整体开发效率。

后端框架：选择Spring Boot作为后端框架，它以其快速开发、易于部署和对微服务架构的支持而受到青睐。Spring Boot简化了基于Spring的应用开发，加速了从原型到生产的过程。数据库：MySQL 被选为主要的关系型数据库，用于存储用户信息、心理健康数据等。前端框架：选择Vue.js等现代化的前端框架，实现用户友好的界面和良好的交互体验。

## 3.2 设计原则

（1）单一职责原则（Single Responsibility Principle，SRP）：

对于单一职责原则最基本的理解就是:一个类,应该只有一个可以导致其发生变化的原因每个模块、类或方法应该只有一个单一的功能或责任[12]。通过采用明确的职责划分和模块化设计，代码结构将变得更加清晰，各个组件之间的依赖关系得以简化。

（2）开放封闭原则（Open/Closed Principle，OCP）：

开放-封闭原则就是:软件实体 (类、模块、函数等) 应该是可以扩展的, 但是不可修改。这种做法有助于减少对现有代码的影响，提升系统的稳定性和可扩展性[7]。

（3）依赖倒置原则（Dependency Inversion Principle，DIP）：

在软件设计中，高层模块不应直接依赖于低层模块，而应共同依赖于抽象。抽象应该独立于具体实现细节，而具体实现细节则应依赖于抽象。

## 3.3 可行性分析

### 3.3.1 技术可行性

大学生健康管理系统的技术可行性分析显示，该项目在技术层面具备实现的潜力和条件。系统采用Windows操作系统，利用其稳定性为开发和运行提供基础。开发工具选用了功能强大的IntelliJ IDEA，支持高效的Java语言开发，同时前端技术采用Vue.js框架，这使得用户界面构建变得简洁且易于维护。后端则采用Spring Boot框架，简化了基于Spring应用的搭建和开发，同时集成了MyBatis ORM框架，便于数据库操作。系统还引入了Apache Shiro框架，增强了权限管理和安全性。

数据库方面，选择MySQL作为数据存储解决方案，其开源性和广泛的社区支持为数据的稳定性和安全性提供了保障。系统设计了详细的E-R模型，并构建了相应的数据库表，如用户表、建议表、角色权限控制表等，确保了数据的组织和存储满足需求。安全性方面，系统通过HTTPS协议和Apache Shiro框架确保数据传输和访问的安全性。此外，系统的代码结构清晰，易于维护和扩展，技术选型均考虑了当前流行趋势和社区支持，有利于系统的长期发展。

系统的功能模块经过了详细的测试，包括用户管理、健康档案管理、健康建议、权限管理等，测试结果表明各模块功能正确且稳定。系统测试进一步验证了整个系统的集成和运行效果，确保了系统的可靠性。因此，从技术角度来看，大学生体检档案管理系统的开发不仅是可行的，而且具备了高效、安全、稳定运行的潜力。

### 3.3.2 经济可行性

在当前信息技术飞速发展的背景下，大学生健康管理逐渐成为社会关注的焦点。为了提升大学生体检档案的管理效率和服务质量，开发一个基于Web的大学生健康管理管理系统显得尤为必要。这样的系统不仅可以减少传统纸质档案管理所带来的资源浪费，还能提高信息的安全性和可访问性。

系统的主要成本包括初期的技术开发和硬件投入，以及后期的维护和运营费用。然而，这些成本可以通过提高管理效率、减少档案丢失风险以及提升服务质量等方式得到补偿。例如，通过电子化管理，可以显著减少因纸质档案损坏或丢失所带来的损失，同时也降低了人工管理的成本。此外，系统带来的便捷性和及时性也将提升用户满意度，从而可能吸引更多的用户使用，间接增加收益。尽管存在一定的风险，但通过敏感性分析和风险管理策略，可以降低这些风险对项目的影响。

总之，基于Web的大学生健康管理管理系统在经济上是可行的。它不仅能够提高档案管理的效率和安全性，还能为学生提供更好的健康服务。随着技术的进步和市场需求的增长，该系统有望在未来发挥更大的作用，带来更广泛的社会效益和经济效益。

### 3.3.3 市场可行性

在进行大学生健康管理系统的市场可行性分析时，我们首先关注市场需求，当前社会对健康问题的关注日益增加，特别是大学生群体，作为国家未来发展的关键，他们的健康状况尤其重要。然而，传统的纸质健康档案管理方式已经无法满足现代信息化管理的需求，这就为高效、便捷的健康管理平台的出现提供了市场空间。目标用户群体包括大学生、校医院工作人员、辅导员及学校管理层，他们需要一个能够方便查询和管理健康信息的平台。

在竞争对手分析方面，虽然市场上已有一些健康管理系统，但它们大多数功能单一，并没有专门针对大学生群体的需求进行定制化服务。因此，我们的系统通过提供个性化健康建议和健康知识普及等功能，能够在现有市场中脱颖而出，形成差异化竞争优势。探讨盈利模式时，该系统可以采取多种策略，包括向高校提供定制化服务并收取服务费，或通过平台广告、数据分析服务等方式进行商业化运营，从而实现盈利。当然，市场风险也是必须考虑的因素，包括技术迭代速度、用户习惯变化以及数据安全和隐私保护问题，这些都可能对系统的推广和使用造成影响。

为了推广系统，我们可以采取与高校合作、线上线下结合的宣传策略，并通过提供试用期和定制化服务来吸引用户。同时，建立有效的用户反馈机制，根据用户需求不断优化系统功能，提高用户满意度。综合考虑市场需求、用户需求、竞争环境和盈利潜力，大学生健康管理系统在市场上具有明显的可行性。通过精心设计的营销策略和风险管理措施，该系统有潜力在健康管理领域占据一席之地，并为大学生群体提供更加精准和高效的健康管理服务。

### 3.3.4 操作可行性

大学生健康管理系统的操作可行性分析强调了用户界面的直观性、操作流程的简洁性以及个性化设置的可用性。系统设计考虑了用户易用性，提供了清晰的导航和逻辑流程，确保用户能够迅速掌握并有效使用各项功能。无论是学生查看健康建议，还是医务人员管理健康档案，都设计为仅需简单几步即可完成。

系统的响应式设计确保了跨设备的一致体验，而强大的后端支持则保证了操作的快速响应。此外，系统还提供了帮助和用户支持功能，以降低用户的学习难度。

安全性方面，系统通过细致的权限管理保护用户数据，使用户可以安全地进行操作。通过全面的用户测试，包括功能测试和可用性测试，系统显示出高度的用户操作可行性，能够满足不同用户群体的需求，并确保了广泛的接受度和实用性。

## 3.4 需求分析

### 3.4.1 功能性需求

大学生健康管理系统的功能性需求集中在提升健康管理效率、保障数据安全、增强用户体验和支持决策制定。系统需满足不同角色用户的特定需求，提供用户友好的操作界面和高效的档案管理功能。同时，系统应保证数据的安全性和隐私性，提供灵活的权限控制以适应不同角色的需求，并提供必要的健康教育内容。通过实现这些功能模块，系统将提高体检档案管理的效率，加强健康管理服务，最终提升大学生的整体健康水平。大学生健康管理系统的设计旨在通过信息技术提升健康管理服务的效率和质量。以下是对该系统的功能性需求分析，覆盖四个用户角色和九大功能模块：

用户角色及权限如下：

（1）学生

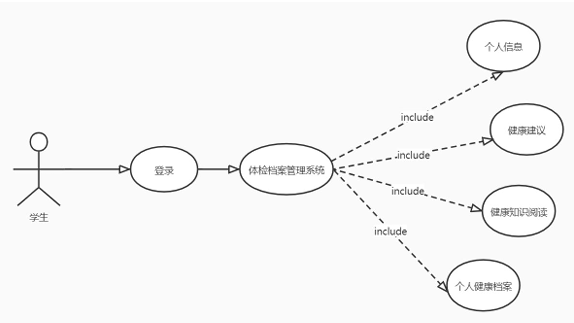
A. 访问和更新个人信息。

B. 管理个人健康档案，包括查看和更新体检数据。

C. 阅读健康知识文章。

D. 查看针对自身健康状况的医生建议。

如图3-1：



**图3-1 学生用例图**

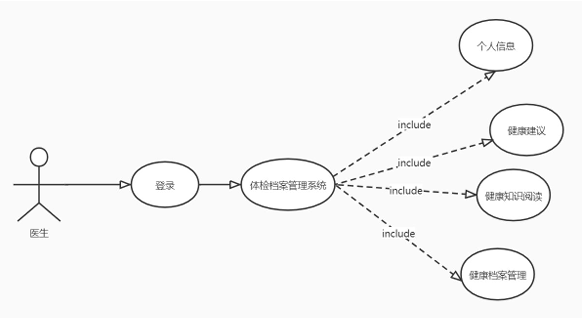
（2）医生

A. 访问特定学生的个人健康档案，以便进行分析和提供健康建议。

B. 发布健康建议和指导。

C. 管理健康知识文章，包括发布和更新。

如图3-2：



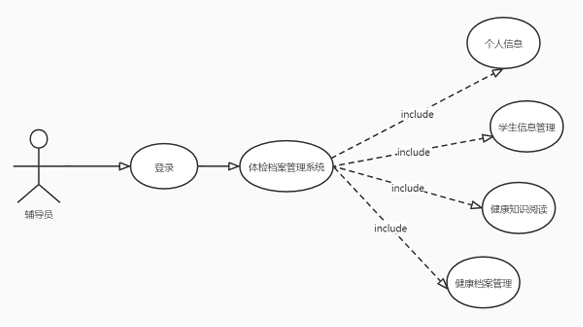
**图3-2 医生用例图**

（3）辅导员

A. 管理学生用户信息，包括查询和更新学生档案。

B. 访问健康档案，以提供必要的辅导和支持。

如图3-3：



**图3-3 辅导员用例图**

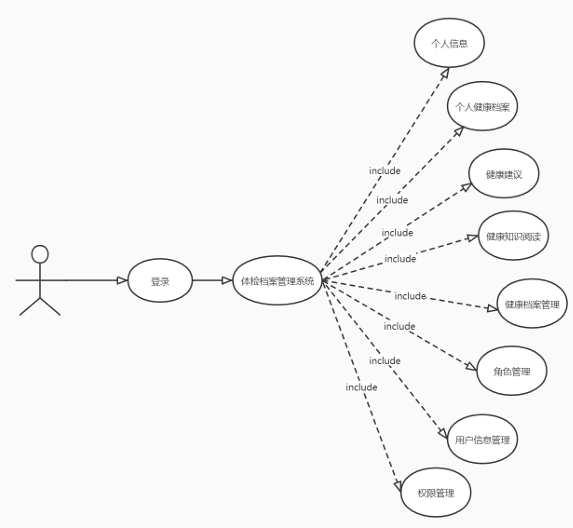
（4）管理员

A. 管理所有用户信息，包括添加、编辑和删除用户账户。

B. 管理角色和权限，确保系统安全和数据的合理访问。

C. 监控系统运行状态，执行数据备份和恢复。

如图3-4：



**图3-4 管理员用例图**

九大功能模块需求如下：

（1）个人信息管理

用户能够更新自己的基本信息，如联系方式、教育背景等。

（2）个人健康档案

学生可以上传和维护自己的健康档案，包括体检报告和健康数据。医生可以访问和分析健康档案，以提供专业建议。

（3）健康建议

医生可以发布健康建议，学生可以查看并反馈。

（4）健康知识阅读

所有用户都可以阅读和学习健康知识，提升健康意识。

（5）健康档案管理

医生和辅导员可以管理健康档案，包括档案的增删改查。

（6）角色管理

管理员可以创建新角色，分配权限，以适应不同管理需求。

（7）用户信息管理

管理员和辅导员可以管理用户信息，确保信息的准确性和完整性。

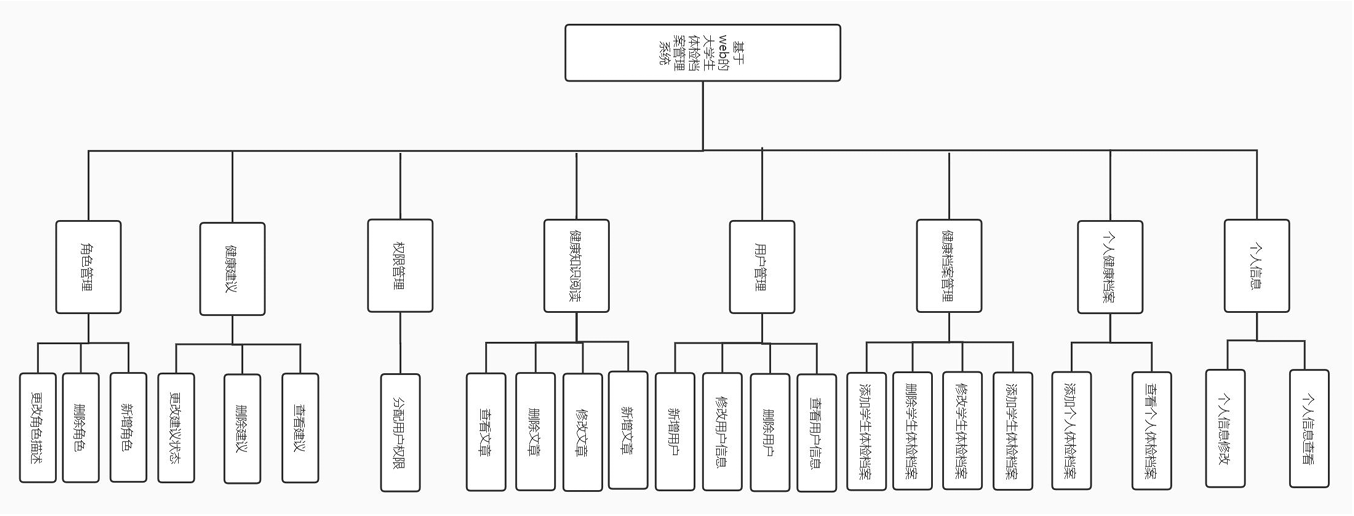
（8）权限管理

管理员可以分配和调整用户权限，控制对系统功能的访问。

（9）体检数据分析

系统应提供数据分析工具，帮助医生和管理员分析健康趋势和统计信息。

如图3-5：



**图3-5 系统功能模块图**

### 3.4.2 非功能性需求

大学生健康管理系统的非功能性需求确保了系统的性能、可靠性、可用性、安全性等关键质量属性。通过满足这些需求，系统将为用户提供高效、稳定、安全的服务，同时保证系统的长期可维护性和可持续发展。这些需求是系统设计和实现过程中的重要考量，确保系统能够满足不断变化的技术和业务需求。以下是具体的非功能性需求分析：

（1）性能需求

响应时间：系统应保证在正常使用条件下，页面加载时间不超过2秒，数据操作响应时间不超过1秒。

并发用户：系统应能够支持至少1000个用户同时在线，保证在高并发场景下系统稳定性[8]。

（2）可靠性需求

系统稳定性：系统应保证99.9%的在线时间，计划外的系统宕机时间不超过1小时/年。

数据备份：系统应实现每天自动备份，备份数据应能在5分钟内恢复。

（3）可用性需求

用户界面：界面应满足至少95%的用户满意度，确保用户能够快速上手。

帮助和文档：提供完整的在线帮助文档和用户手册，覆盖所有功能模块。

（4）安全性需求

数据加密：所有敏感数据，如用户个人信息和健康数据，应使用AES-256加密标准进行加密。

用户认证：系统应实现基于OAuth 2.0的认证机制，提供安全的访问控制。

权限控制：系统应实现基于角色的访问控制，确保用户只能访问对应的数据和功能。

（5）兼容性需求

浏览器兼容性：系统应在Chrome、Firefox、Edge等主流浏览器上正常运行，支持浏览器版本至少覆盖90%的用户使用场景。

设备兼容性：系统应支持iOS、Android和Windows操作系统的移动设备和桌面设备。

（6）可维护性和可扩展性需求

代码可读性：代码应遵循编码标准，代码复杂度不超过20，以便于维护和扩展。

（7）法律和规范需求

数据保护：系统应遵守GDPR等数据保护法规，确保用户数据的安全和隐私。

（8）环境需求

硬件环境：系统运行所需的最低硬件配置应为4核CPU、8GB内存、100GB硬盘空间。

软件环境：系统应能在Windows 10以上操作系统上运行。

# 系统概要设计

## 4.1 系统架构设计

大学生健康管理系统采用B/S框架和MVC框架：具体介绍如下：

（1）B/S框架

B/S 架构是一种客户端与服务器端分离的架构模式。客户端主要通过浏览器访问服务器端提供的服务。在这种架构中，服务器端通常负责处理业务逻辑、数据存储和网络通信，而客户端则负责展示界面和用户交互。

前端（浏览器端）： 使用 Vue.js 框架进行开发，负责展示用户界面和处理用户交互。Vue.js 作为前端框架，可以高效地构建用户界面，并通过 Ajax 与后端进行数据交互。

后端（服务器端）： 使用 Spring Boot 框架，负责处理业务逻辑、数据访问和与前端的数据交互。Spring Boot 框架简化了传统 Spring 框架的配置，使得开发和部署更加便捷。

数据库： 使用 MySQL 数据库进行数据存储。数据库设计包括用户表、建议表、角色权限控制表、角色表、权限表和文章表等。

（2）MVC框架

在大学生健康管理系统中，采用了MVC结构，MVC（Model-View-Controller，模型-视图-控制器）结构设计是一种常见的软件架构模式，用于分离系统中的不同组件，提高代码的可维护性和可扩展性[9]。以下是对系统MVC结构的具体介绍：

模型（Model）

模型层是系统的数据和业务逻辑核心，负责处理数据存储、检索和业务规则的实现。在本系统中：

(1)数据持久化：通过MyBatis等ORM（对象关系映射）工具与MySQL数据库进行交互，实现数据的增删改查操作。

(2)业务逻辑：封装了体检档案管理的所有业务逻辑，如用户信息验证、档案的添加、修改、删除和查询等。

(3)数据封装：将从数据库中检索到的数据封装成对象，供控制器层使用。

视图（View）

视图层负责展示数据（模型）以及接收用户输入，是用户与系统交互的界面。在本系统中：

(1)用户界面：使用Vue.js框架构建，提供了丰富的用户界面组件和动态效果。

(2)数据展示：将模型层的数据以表格、列表或图表等形式展示给用户。

(3)用户输入：接收用户的输入，如搜索条件、新增档案信息等，并将这些输入传递给控制器层。

控制器（Controller）

控制器层是模型和视图之间的中介，负责接收用户的输入，调用模型层的业务逻辑，并选择视图以展示结果。在本系统中：

(1)请求处理：接收来自视图层的用户请求，如点击按钮、提交表单等。

(2)业务调用：根据用户请求调用模型层的方法，执行相应的业务操作。

(3)视图渲染：将模型层返回的数据传递给视图层，由视图进行渲染展示。

## 4.2 系统功能设计

### 4.2.1 个人信息模块

(1)个人信息展示：允许用户查看自己的详细信息，包括但不限于姓名、学号、性别、出生日期、学院、专业、班级、家庭住址、籍贯、职业、工作地点、照片等。

(2)个人信息编辑：用户可以修改自己的部分个人信息，如联系方式、家庭住址、职业等。

(3)查看个人体检信息：用户可以看到以图形化形式展示的个人身高、体重在各个学期的变化，非常直观形象；用户还可以看到自己当前的BMI指数。

(4)反馈机制：在用户进行信息编辑或更新操作后，系统应提供明确的成功或错误反馈。

### 4.2.2 健康档案管理

(1)档案浏览:

用户能够浏览学生的体检档案列表，展示每个档案的基础信息，如学生姓名、学号、体检日期等。

支持分页、排序，以便于在大量档案中快速找到目标记录。

(2)档案查询:

实现综合查询功能，用户可以通过单一或组合关键字（如姓名、学号、体检日期等）进行档案检索。

提供高级查询选项，允许用户根据特定条件过滤结果，如特定时间段内的体检记录。

(3)档案添加:

提供一个表单，医生和授权用户可以在其中输入学生的体检数据，包括但不限于身高、体重、血压、心率等。

支持上传体检报告的扫描件或图片作为档案附件。

(4)档案编辑

用户可以打开现有档案，并对其内容进行修改，包括更新体检数据和更正信息错误。

编辑操作后，系统记录变更历史，保留原始数据的完整性。

(5)档案删除

用户可以通过选择一个或多个档案，并执行删除操作以从系统中移除这些档案。

系统提供确认步骤，以防止误删除，并记录删除操作的日志。

(6)档案默认数据填充

在创建新档案时，系统根据预设模板自动填充学生的基本信息，减少重复性数据输入。

(7)健康建议关联

允许医生在档案中添加针对学生体检结果的健康建议，并将建议与档案关联，便于后续跟踪和评估。

(8)档案信息展示

档案详细页面展示学生的全面体检信息，包括体检数据、历史体检记录、健康建议等。

### 4.2.3 主界面模块

(1)主菜单导航:

显示主菜单和用户账户名，罗列系统提供的所有功能模块，如个人信息、健康档案管理等。

(2)角色特定功能访问:

根据用户角色（如管理员、医生、辅导员、学生）展示相应的功能模块。

(3)退出系统:

允许用户安全退出系统，保护用户信息不被未授权访问。

(4)修改密码:

用户可以修改已经登录的账号的密码

### 4.2.4 体检数据分析模块

(1)数据追踪与历史对比:

功能描述：页面应提供学生BMI指数随时间变化的详细追踪记录，允许用户对比不同学期间的BMI最高值和最低值，以识别体重变化趋势。

(2)时间序列分析:

功能描述：用户应能够选择不同的时间点（如不同的学期），系统将展示该时间点的BMI数据，支持从大一到大四整个学习周期的数据分析。

(3)数据对比视图

功能描述：通过图表直观展示同期内最高和最低BMI值，使用户能够一眼看出体重分布的极端情况。

(4)图表展示与定制

功能描述：系统应提供多种图表展示方式，如折线图、柱状图或面积图，同时允许用户根据个人喜好或需求定制图表样式。

(5)肥胖占比统计分析

功能描述：页面应展示不同学期学生体重分类的占比，包括偏瘦、正常、偏胖和极胖，使用户能够了解学生体重分布的变化。

(6)健康体重参考范围

功能描述：提供健康体重的参考范围或标准，帮助用户评估和理解BMI指数的健康含义。

(7)交互式图表操作

功能描述：图表应支持交互操作，如点击或悬停显示更多详细信息，选择特定数据点进行深入分析。

(8)角色定制化视图

功能描述：根据用户角色展示不同的数据视图，如学生查看个人趋势，医生查看群体统计。

(9)响应式和可访问性设计

功能描述：页面应具有响应式设计，兼容各种设备和屏幕尺寸，同时考虑视觉障碍用户的需求。

(10)数据导出与分享

功能描述：允许用户将图表和数据导出为常见格式，如Excel、PDF等，便于进一步分析或报告。

## 4.3 系统业务流程设计

### 4.3.1 个人信息模块

(1)用户登录:

用户通过系统的登录界面进入系统。输入注册时所用的用户名和密码。用户点击登录按钮后，系统进行身份验证。如果凭证有效，系统允许用户进入。

(2)访问个人信息页面:

用户在成功登录后，通过点击导航栏中的“个人信息”选项卡或链接进入个人信息页面。用户可以通过此页面访问更详细的个人信息，包括个人资料、健康档案、历史记录等。

(3)个人信息展示:

系统展示用户的基本信息，如全名、学号、性别、出生日期、联系方式等。显示用户的学院、专业、班级等教育背景信息。包括用户的居住地址、籍贯等个人信息。

(4)信息编辑权限判断:

系统检查用户是否具有编辑个人信息的权限。根据用户的角色（如学生、教师、管理员），系统展示不同的编辑选项。

(5)编辑个人信息:

用户在个人信息页面选择“编辑”选项，进入编辑模式。用户可以更新联系方式、家庭住址、职业等信息。系统提供实时反馈，确保用户输入的数据符合要求。

(6)信息更新:

用户完成编辑后，提交更新请求。系统对提交的数据进行验证，包括格式、长度和数据的唯一性。如果数据验证通过，系统将更新请求发送到服务器。

(7)更新结果反馈:

系统处理更新请求，并在操作完成后提供反馈。如果更新成功，系统显示成功消息，并在个人信息页面展示更新后的信息。如果更新失败，系统显示错误消息，说明失败的原因，并允许用户重新提交。

(8)异常处理:

系统具备异常捕获机制，能够在数据验证失败或更新失败时通知用户。显示友好的错误信息，指导用户如何纠正问题。

(9)用户登出:

用户在完成所有操作后，可以选择点击“退出系统”按钮安全退出系统。登出后，用户回到登录界面，需要重新登录才能再次访问系统。

### 4.3.2 健康档案管理模块

(1)用户登录:

用户通过系统的登录界面进入系统。输入注册时所用的用户名和密码。用户点击登录按钮后，系统进行身份验证。如果凭证有效，系统允许用户进入。

(2)访问健康档案管理页面:

用户在成功登录后，通过点击导航栏中的“健康档案管理”选项卡或链接进入个人信息页面。用户可以通过此页面访问更详细的健康档案。

(3)档案浏览:

用户登录系统后，具有浏览权限的用户可以访问体检档案管理页面。系统展示一个包含所有学生体检档案的列表，显示关键信息，如学生姓名、学号、体检日期等。用户可以通过翻页或滚动查看完整的档案列表。列表中每条记录都可以展开以显示更详细的体检数据和个人信息。

(4)档案查询:

用户可以使用查询功能，输入学生的姓名或其他相关信息进行搜索。系统提供模糊搜索和高级搜索选项，允许用户根据特定条件过滤结果。查询结果以列表形式展示，用户可以选择查看具体档案的详细信息。

(5)档案添加:

医生或管理员在体检完成后，通过档案管理页面的“添加档案”功能创建新的体检档案。用户输入学生的个人信息和体检数据，包括但不限于身高、体重、血压等。系统提供数据校验，确保输入的数据准确无误。成功添加后，系统自动为档案生成唯一标识符，并保存档案。

(6)档案编辑:

授权用户可以通过档案列表或档案详细信息页面的“编辑”选项修改现有档案。用户可以更新体检数据、更正错误信息或添加额外的备注。

(7)档案删除:

授权用户可以对不再需要的档案执行删除操作。删除操作后，档案从系统中彻底移除。

(8)档案查看:

学生和医生可以查看学生的体检档案，包括最新状态等。

系统提供档案的详细视图，展示完整的体检数据和医生的评语。

(9)健康建议:

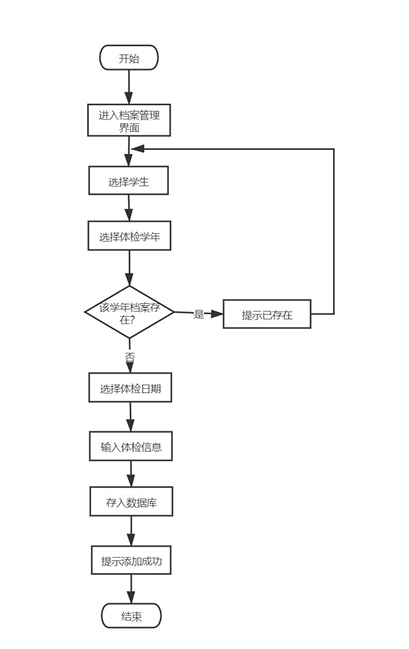
医生在查看体检档案后，可以填写健康建议，包括生活方式的改善、饮食习惯的调整等。系统允许医生根据体检结果定制个性化的健康建议。健康建议与体检档案关联，便于学生和医生查阅。

(10)建议查看和管理:

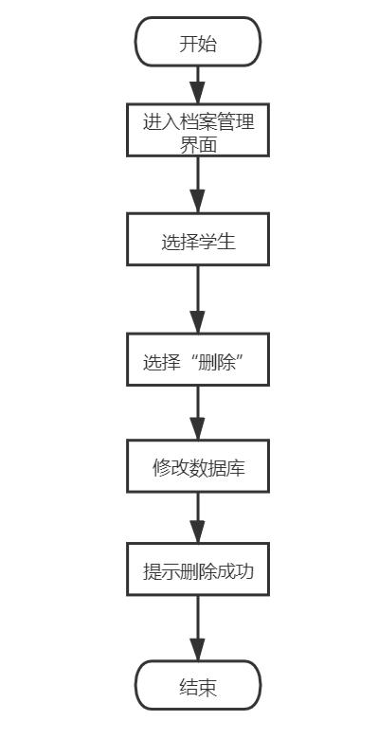
学生登录系统后，可以在健康建议部分查看医生提供的所有建议。学生可以对每条建议进行阅读状态的更新，如标记为“已读”。学生可以对建议进行反馈，如提出问题或报告建议的执行情况。

(11)用户登出:

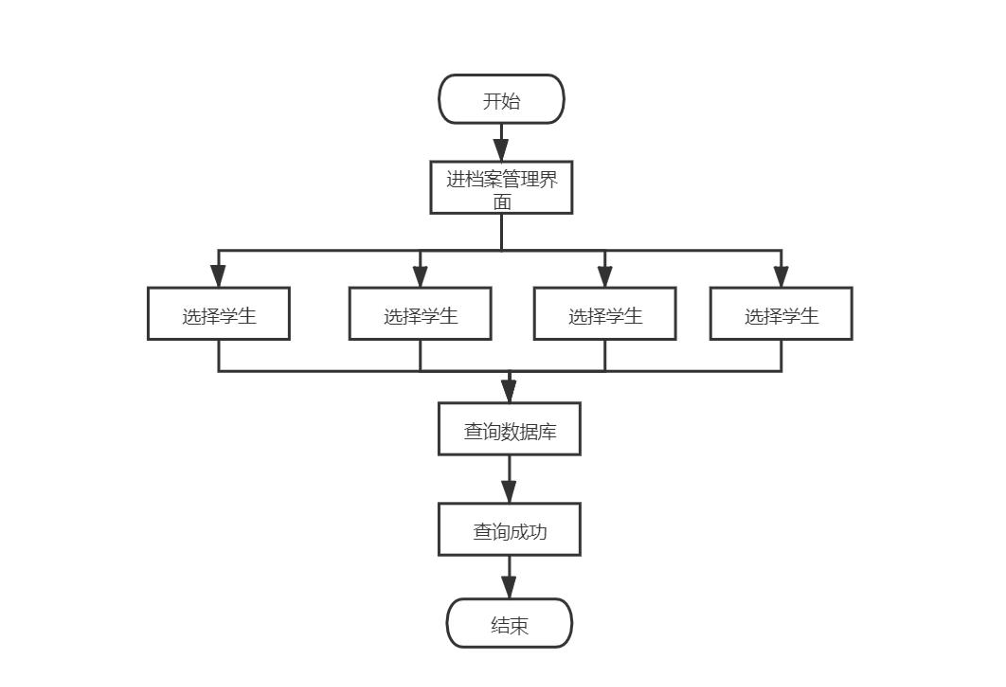
用户在完成所有操作后，可以选择点击“退出系统”按钮安全退出系统。登出后，用户回到登录界面，需要重新登录才能再次访问系统。如图4-1、4-2、4-3：



**图4-1 添加档案流程图**



**图4-2 删除档案流程图**



**图4-3 查询档案流程图**

### 4.3.3 主页面

(1)用户登录:

用户访问系统登录页面，输入注册时设置的用户名和密码。系统通过后端验证用户名和密码的正确性。如果凭证正确，系统允许用户登录；如果不正确，系统拒绝登录并提示用户重新输入或重置密码。

(2)角色识别:

成功登录后，系统根据用户的账户信息识别用户的角色。角色可能包括管理员、医生、学生、辅导员等，每种角色具有不同的权限和功能。系统根据识别到的角色动态加载用户权限，确保用户只能访问其权限范围内的功能。

(3)主界面展示:

用户登录后进入主界面，主界面的设计和内容根据用户的角色进行个性化展示。主界面通常清晰、直观，提供用户最常用的功能和信息。

(4)导航栏:

导航栏位于主界面的顶部或侧边，提供系统主要功能模块的快速访问。导航栏包含如账号名、修改密码、退出系统等链接。

用户可以通过点击导航栏中的链接快速跳转到相应的功能模块。

(5)功能区:

功能区是主界面的核心部分，展示用户可以执行的具体操作。功能区的内容和布局根据用户的角色动态调整，确保用户能够看到与其角色相关的功能。例如，医生可能会看到“添加健康建议”、“查看学生档案”等功能，而学生可能只能看到“查看健康档案”和“查看健康建议”。

(6)数据加载与显示:

当用户点击导航栏或功能区的链接时，系统加载并显示相关数据。系统确保数据加载的效率，提供加载中的提示信息，如旋转的加载图标或进度条。数据加载完成后，系统展示详细内容，并确保显示的数据准确无误。

(7)安全退出:

用户在完成操作后，可以通过点击界面右上角的“退出”按钮安全退出系统。系统提供退出确认提示，防止用户误操作导致未保存的数据丢失。退出系统后，用户的会话结束，系统不再保留用户的状态和数据。

(8)异常处理:

如果在数据加载或用户操作过程中出现异常，系统捕获异常并提供错误提示。系统记录异常信息，供管理员进行问题排查和系统维护。用户可以通过联系技术支持或查看帮助文档来解决遇到的问题。

### 4.3.4 体检数据分析

(1)用户登录:

用户访问系统登录页面，输入注册时设置的用户名和密码。系统通过后端验证用户名和密码的正确性。如果凭证正确，系统允许用户登录；如果不正确，系统拒绝登录并提示用户重新输入或使用“忘记密码”功能。

(2)角色识别:

登录成功后，系统根据用户的账户信息识别用户的角色（如管理员、医生、学生、辅导员）。系统根据用户角色提供相应的访问权限和功能，确保用户只能访问其权限范围内的模块和数据。

(3)访问体检数据分析模块:

用户登录后，通过导航栏或功能区选择“体检数据分析”模块。加载体检数据分析模块的界面，并展示相关的数据和分析工具。

(4)数据展示:

系统展示体检数据的统计和分析结果，包括学生的最高和最低BMI（体重指数）值的变化分析。用户可以查看不同学期或学年的BMI值变化，了解学生体重变化趋势。

(5)时间序列分析:

用户可以选择不同的学期或学年来查看数据。系统展示不同时间点的BMI值变化，如大一上学期到大四，帮助用户了解学生体重变化的长期趋势。

(6)数据对比:

系统提供最高BMI和最低BMI的对比视图。用户可以直观地看到不同学期或学年的BMI值变化，识别体重变化的模式和异常。

(7)肥胖占比分析:

系统展示不同体重分类（如偏瘦、健康体重、偏胖、极瘦、极胖）的占比。用户可以了解学生体重分布的总体情况，识别需要特别关注的体重问题。

(8)图表展示:

系统使用图表（如柱状图、折线图等）展示数据，增强可视化效果。用户可以通过图表直观地理解数据，识别趋势和模式。

(9)数据导出:

用户可以导出体检分析数据，如保存为Excel、CSV或PDF文件。导出的数据可以用于进一步的分析或报告生成。

(10)异常处理:

如果在数据加载或分析过程中出现异常，系统捕获异常并提供错误提示。系统记录异常信息，供管理员进行问题排查和系统维护。

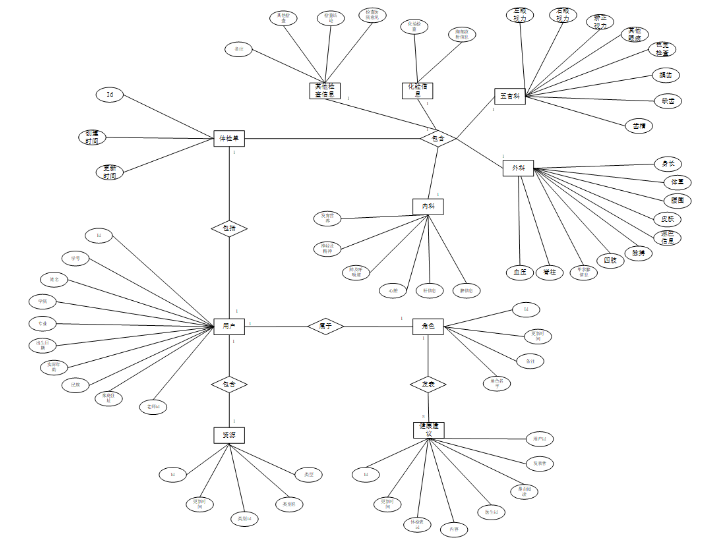
(11)安全退出:

用户完成操作后，可以通过点击界面右上角的“退出系统7”按钮安全退出系统。

## 4.4 数据库ER模型

### 4.4.1 整体ER图

大学生体检档案管理系统的总体ER图提供了系统数据结构的全面视图，详细展示了系统中所有关键实体及其相互关系。用户实体作为系统的核心，记录了包括基本信息和动态信息在内的全部用户数据。角色实体定义了不同用户的类别，如管理员、医生等，并通过角色权限绑定实体与权限实体相结合，实现了角色与权限的灵活分配。建议实体记录了医生对学生提出的健康建议，与用户实体形成一对多关系。健康知识文章则由文章实体管理，支持学生对健康知识的学习和了解。虽然体检档案实体在文档中未明确列出，但可推断其与用户实体的关联性，记录学生体检信息。ER图中的连线清晰表达了实体间的关联，如角色与权限的多对多关系，用户与建议的一对多关系。每个实体的属性设计确保了数据的完整性和准确性，主键和必填字段的使用保障了记录的唯一性和数据的完整性。系统设计考虑了扩展性，允许基于现有结构添加新的实体和关系，适应系统功能的未来发展。ER图为数据库设计、系统开发和维护提供了直观的参考，帮助开发者深入理解数据结构，优化设计，并为系统迭代提供指导，是系统开发过程中不可或缺的关键文档。如图4-4：

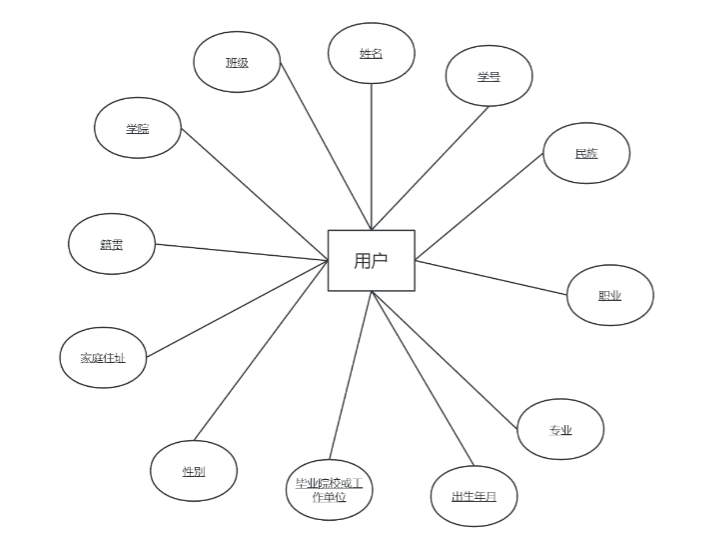


**图4-4 总体ER图**

### 4.4.2 局部ER图

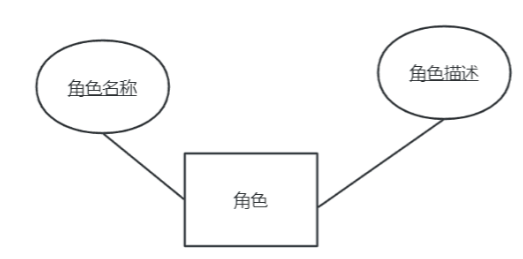
用户实体属性图提供了大学生体检档案管理系统中用户实体的基本信息概览。它包括了学生的学号、姓名、性别、学院、专业等教育背景信息，同时记录了家庭住址、民族等个人身份数据，以及毕业院校或工作单位等职业信息，涵盖了出生年份等基础个人信息。这些数据为系统提供了一个全面的学生档案视角，便于进行有效的健康管理和信息查询。

属性图见图4-5：



**图4-5 用户实体属性图**

该实体属性图列出了角色管理模块的核心元素，包括角色名称和对应的角色描述，为系统用户分配不同的权限和职责提供了基础。如图4-6：



**图4-6 角色实体属性图**

# 系统详细设计

## 5.1 系统功能模块设计

### 5.1.1 个人信息模块

(1)查看个人基本信息：例如姓名、民族等基本信息；

(2)查看体检数据信息：可以以图形的形式，形象的查看各个学年的身高和体重变化，并查看相应体质指数。

(3)可以修改个人基本信息：如果个人信息需要更新，可以点击修改按钮，输入新的信息并保存。

### 5.1.2 健康档案管理模块

管理员：

(1)查看用户的健康档案信息：包括各种个人信息和一些基本体检数据。

(2)编辑用户的健康档案信息：如果用户信息有误，用户管理员可以进行修改，或补充用户信息。

(3)删除用户的健康档案信息：如果档案有误，可以进行删除。

(4)查询用户健康档案：可以根据学年、学院、专业、姓名等对用户的健康档案进行筛选查询，

(5)添加用户健康档案：也以为用户添加新的健康档案。

医生：

(1)查看用户的健康档案信息：包括各种个人信息和一些基本体检数据。

(2)编辑用户的健康档案信息：如果用户信息有误，用户管理员可以进行修改，或补充用户信息。

(3)删除用户的健康档案信息：如果档案有误，可以进行删除。

(4)查询用户健康档案：可以根据学年、学院、专业、姓名等对用户的健康档案进行筛选查询，

(5)添加用户健康档案：也以为用户添加新的健康档案。

(6)提出健康建议：查看完用户的体检表后，可以根据实际情况给出健康建议。

### 5.1.3 主界面模块

(1)显示账号信息：显示登录账号的名称。

(2)修改账号密码：可以修改登录账号的密码。

(3)登出系统：登录的账号可以登出系统，返回到登录界面。

(4)显示功能模块：根据登录的账号的角色，显示出相应的功能模块，如学生角色要显示个人信息，管理员要显示权限管理。

### 5.1.4 体检数据分析模块

(1)显示学生的PIM值的变化：显示学生最近几年的最低和最高的PIM值变化，以便于分析学生的健康状况。

(2)显示肥胖占比：在不同学期中，以条形图的形式展示胖瘦和正常学生的占比情况。

(3)其他：以圆形图等其他形式展示学生群里的体检情况。

## 5.2 数据库表设计

在大学生健康管理系统中，我负责的模块有个人信息模块、主界面模块、健康管理模块和体检数据分析模块，要涉及到数个数据库表，具体的模块涉及到的数据库表如下：

1. 个人信息表：个人信息数据库表是一个关键的数据存储组件，用于在大学生健康管理系统中保存学生的基本信息。这张表以`id`作为主键，包含了创建和更新时间戳，确保记录的时效性。表中记录了学生的出生日期、学院、班级、专业等教育背景信息，以及姓名、密码、角色ID、性别、学号等身份识别信息。此外，还包括了用户名、学历、民族、籍贯、职业、现住址、工作地点、照片和年龄等个人资料，以及辅导员ID，为学生提供指导和支持。每条记录都通过字段长度和非空约束确保数据的准确性和完整性。通过这张表，系统能够高效地管理和检索学生的个人信息，支持健康管理的个性化服务和决策制定[10]。

**表5-1 个人信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段 | 类型 | 长度 | 非空 |
| id | Id | int | - | 是 |
| create\_datetime | 创建时间 | datetime | - | 是 |
| update\_datetime | 更新时间 | datetime | - | 是 |

**表5-1 个人信息表（续）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段 | 类型 | 长度 | 非空 |
| birth | 出生日期 | date | - | 否 |
| college | 学院 | varchar | 255 | 否 |
| grade | 班级 | varchar | 255 | 否 |
| major | 专业 | varchar | 255 | 否 |
| name | 姓名 | varchar | 255 | 是 |
| password | 密码 | varchar | 255 | 是 |
| role\_id | 角色ID | int | - | 是 |
| sex | 性别 | varchar | 10 | 否 |
| stu\_no | 学号 | varchar | 255 | 否 |
| username | 用户名 | varchar | 255 | 是 |
| culture\_level | 学历 | varchar | 255 | 否 |
| nation | 民族 | varchar | 255 | 否 |
| native\_place | 籍贯 | varchar | 255 | 否 |
| occupation | 职业 | varchar | 255 | 否 |
| present\_address | 现住址 | varchar | 255 | 否 |
| work\_place | 工作地点 | varchar | 255 | 否 |
| photo | 照片 | varchar | 255 | 否 |
| real\_age | 年龄 | int | - | 否 |
| teacher\_id | 辅导员ID | int | - | 否 |

1. 角色权限表：角色权限表是大学生健康管理系统的核心数据库组件，专注于存储和管理不同角色的权限设置。该表使用`Id`作为唯一的主键，确保每条记录的可识别性。表中记录了与角色相关联的`Resource\_id`，标识不同资源或操作，而`role\_id`字段则明确了角色的身份。`Create\_datetime`和`Update\_datetime`字段以时间戳的形式记录了记录的创建和最后更新时间，确保了数据的时效性和可追踪性。此表的设计满足了系统对权限控制的需求，允许系统管理员根据角色分配或修改对应的资源访问权限。通过维护角色权限表，系统能够灵活地处理不同用户的角色差异，实现对功能访问的精确控制，从而保障操作的安全性和合规性。

**表5-2 角色权限表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段 | 类型 | 长度 | 非空 |
| Id | id | int | - | 是 |
| Create\_datetime | create\_datetime | timestamp | - | 是 |
| Update\_datetime | update\_datetime | timestamp | - | 是 |
| Resource\_id | resource\_id | int | - | 否 |
| Role\_id | role\_id | int | - | 是 |

1. 健康档案表：健康档案数据库表是大学生健康管理系统中用于存储学生体检详细信息的表。它通过唯一编号`id`来标识每条记录，并记录创建和更新时间。表中包含学生的基本信息，如姓名、国籍、籍贯、职业等，以及体检数据，包括血压、身高、体重等关键指标。此外，还记录了特定健康检查的结果，如心脏、肝脏、肺脏等器官的状况，以及教育背景信息，如学院、专业和年级。通过`user\_id`与学生个人资料关联，`check\_year`记录体检年份，确保信息的连续性和追踪性。这个设计使得健康档案表成为健康管理中不可或缺的数据支撑，便于进行健康分析和提供个性化建议[11]。

**表5-3 健康档案表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段 | 类型 | 长度 | 非空 |
| id | 编号 | int | - | 是 |
| create\_datetime | 创建时间 | datetime | - | 是 |
| update\_datetime | 更新时间 | datetime | - | 是 |
| resource\_id | 资源编号 | int | - | 否 |
| role\_id | 角色ID | int | - | 是 |
| arthrosis | 关节炎 | varchar | 255 | 否 |
| assay\_check | 体检检查 | varchar | 255 | 否 |
| birth | 出生日期 | date | - | 否 |
| blood\_pressure | 血压 | varchar | 50 | 否 |
| check\_result | 检查结果 | text | - | 否 |
| check\_time | 检查时间 | datetime | - | 否 |
| chest\_check | 胸部检查 | varchar | 255 | 否 |
| college | 学院 | varchar | 255 | 否 |
| color\_or\_code | 颜色或代码 | varchar | 50 | 否 |

**表5-3 健康档案表（续）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段 | 类型 | 长度 | 非空 |
| correct\_left\_view | 左侧视图纠正 | varchar | 255 | 否 |
| correct\_right\_view | 右侧视图纠正 | varchar | 255 | 否 |
| culture\_level | 教育水平 | varchar | 50 | 否 |
| ear\_sick | 耳朵疾病 | varchar | 255 | 否 |
| foot | 脚部 | varchar | 255 | 否 |
| glandula\_thyroidea | 甲状腺 | varchar | 255 | 否 |
| grade | 年级 | varchar | 50 | 否 |
| growth | 成长 | varchar | 50 | 否 |
| heart | 心脏 | varchar | 255 | 否 |
| height | 身高 | varchar | 50 | 否 |
| history\_check | 历史检查 | text | - | 否 |
| hypodontia | 先天性缺牙 | varchar | 255 | 否 |
| left\_audition | 左耳听力 | varchar | 50 | 否 |
| left\_view | 左眼视力 | varchar | 50 | 否 |
| liver | 肝脏 | varchar | 255 | 否 |
| lung | 肺脏 | varchar | 255 | 否 |
| lymph | 淋巴系统 | varchar | 255 | 否 |
| major | 专业 | varchar | 255 | 否 |
| mind | 心智状态 | varchar | 255 | 否 |
| name | 姓名 | varchar | 255 | 是 |
| nation | 国籍 | varchar | 50 | 否 |
| native\_place | 籍贯 | varchar | 255 | 否 |
| occupation | 职业 | varchar | 255 | 否 |
| other\_check | 其他检查 | varchar | 255 | 否 |
| other\_eyes\_sick | 其他眼病 | varchar | 255 | 否 |
| other\_internal\_sick | 其他内科疾病 | varchar | 255 | 否 |
| other\_sense\_sick | 其他感官疾病 | varchar | 255 | 否 |
| other\_surgery\_sick | 其他手术疾病 | varchar | 255 | 否 |
| paranasal\_sinus | 鼻窦 | varchar | 255 | 否 |
| photo | 照片 | varchar | 255 | 否 |
| present\_address | 现居住地址 | varchar | 255 | 否 |

**表5-3 健康档案表（续）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段 | 类型 | 长度 | 非空 |
| pulse | 脉搏 | varchar | 50 | 否 |
| real\_age | 实际年龄 | int | - | 否 |
| remark | 备注 | text | - | 否 |
| right\_audition | 右耳听力 | varchar | 50 | 否 |
| right\_view | 右眼视力 | varchar | 50 | 否 |
| sealer | 密封剂 | varchar | 255 | 否 |
| sex | 性别 | varchar | 10 | 否 |
| sign\_for\_assay | 化验标志 | varchar | 255 | 否 |
| sign\_for\_chest | 胸部检查标志 | varchar | 255 | 否 |
| sign\_for\_ear | 耳部检查标志 | varchar | 255 | 否 |
| sign\_for\_eyes | 视力检查标志 | varchar | 255 | 否 |
| sign\_for\_internal | 内科检查标志 | varchar | 255 | 否 |
| sign\_for\_mouth | 口腔检查标志 | varchar | 255 | 否 |
| sign\_for\_result | 结果标志 | varchar | 255 | 否 |
| sign\_for\_surgery | 手术标志 | varchar | 255 | 否 |
| single\_color\_judge | 单色判断 | varchar | 50 | 否 |
| skin | 皮肤 | varchar | 255 | 否 |
| smell | 嗅觉 | varchar | 255 | 否 |
| spine\_backbone | 脊柱 | varchar | 255 | 否 |
| spleen | 脾脏 | varchar | 255 | 否 |
| stammer | 口吃 | varchar | 255 | 否 |
| stu\_no | 学号 | varchar | 50 | 否 |
| suggest\_for\_check | 体检建议 | text | - | 否 |
| throat | 咽喉 | varchar | 255 | 否 |
| tooth\_place | 牙齿位置 | varchar | 255 | 否 |
| tooth\_sick | 牙齿疾病 | varchar | 255 | 否 |
| user\_id | 用户ID | int | - | 否 |
| waistline | 腰围 | varchar | 50 | 否 |
| weight | 体重 | varchar | 50 | 否 |
| work\_place | 工作地点 | varchar | 255 | 否 |
| check\_year | 体检年份 | year | - | 否 |

1. 医生建议表：医生建议表是大学生健康管理系统中用于记录医生对学生健康体检结果所提出建议的数据库表。它以`id`作为主键，确保每条建议的唯一性，并通过`create\_datetime`和`update\_datetime`时间戳记录建议的创建和更新时间。表中通过`check\_info\_id`与学生的体检信息相联系，并通过`is\_read`标记来追踪学生是否已查看这些建议。医生名称、编号以及学生编号分别通过`office`、`doctor\_id`和`user\_id`字段记录。建议的具体内容则存储在`content`字段中，允许详细描述医生的反馈和指导。这个设计不仅方便了医生对学生健康状况的持续跟踪，也使得学生能够方便地访问和回应医生的专业建议，从而提升了健康管理的个性化和互动性。

**表5-4 医生建议表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段名 | 类型 | 长度 | 非空 |
| id | 编号 | int | - | 是 |
| create\_datetime | 创建时间 | datetime | - | 是 |
| update\_datetime | 更新时间 | datetime | - | 是 |
| check\_info\_id | 体检信息编号 | int | - | 否 |
| is\_read | 已读标记 | int | - | 否 |
| office | 医生名称 | varchar | 255 | 否 |
| user\_id | 用户编号 | int | - | 否 |
| content | 内容 | text | - | 否 |
| doctor\_id | 医生编号 | int | - | 否 |

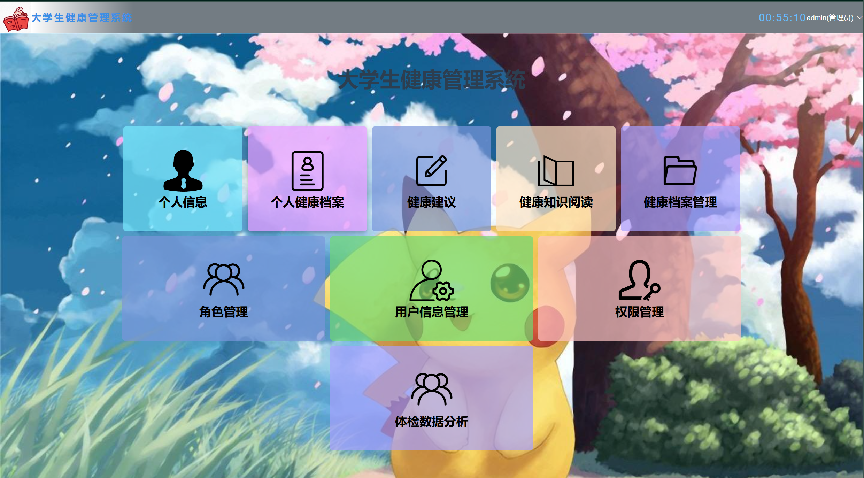
# 系统实现

## 6.1 组员1（子目录是组员1的实现功能）

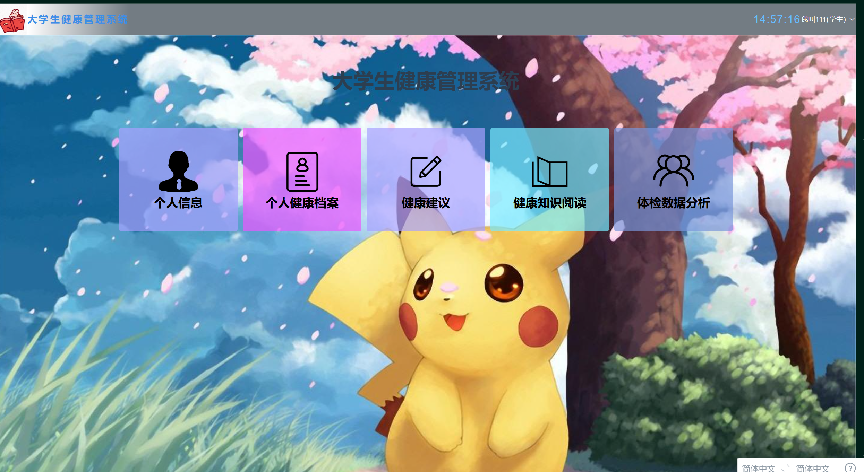
### 6.1.1 系统主界面效果图

大学生健康管理系统提供了一个基于角色定制化的用户界面，使不同角色的用户能够高效地访问特定功能。管理员拥有系统管理的全面权限，能够管理用户信息、健康档案、角色权限等，而学生则可以查看个人信息、健康档案和接受健康建议。医生界面专注于健康建议和档案管理，便于提供专业医疗指导。辅导员则侧重于学生信息管理和健康知识普及。整个系统的设计确保了用户界面的个性化和功能访问的便捷性，提升了用户体验和管理效率。

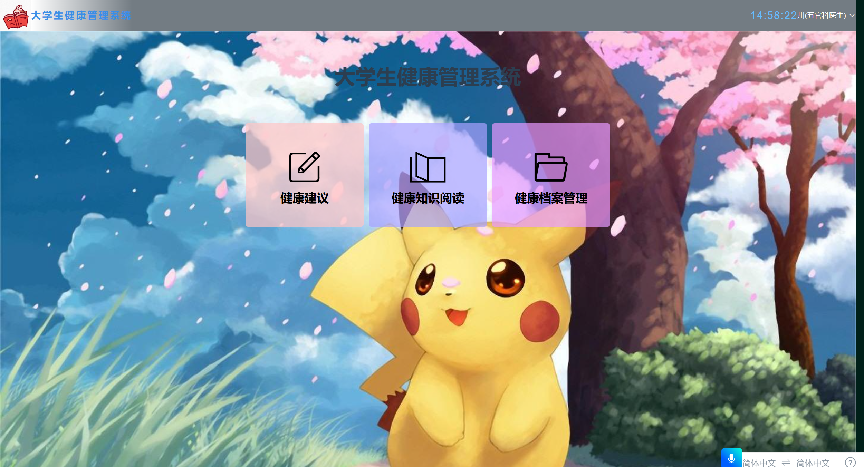
具体效果图如图6-1、6-2、6-3、6-4：



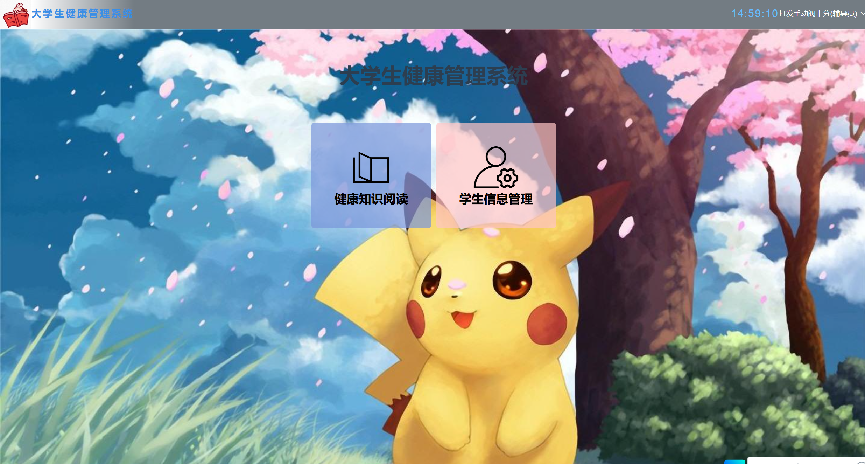
**图6-1 管理员主界面**



**图6-2 学生主界面**



**图6-3 医生主界面**



**图6-4 辅导员主界面**

### 6.1.2 系统主界面实现

主界面是通过Vue.js构建的，利用`<div>`标签创建了一个侧边栏，其中`.menu`类定义了菜单区域。系统名称作为标题显示在菜单顶部。菜单项通过`v-for`循环动态生成，每个项包含一个图标和名称，点击时可以触发页面跳转。后端使用Spring Boot，通过`@GetMapping`注解处理请求，提供菜单数据。具体来说，`/get/{id}`用于根据ID获取单个资源，而`/getMenuByUserId`根据用户ID获取并返回菜单数据列表。如果数据请求失败或返回为空，后端将抛出异常。这种设计实现了前后端分离，前端负责展示和用户交互，后端处理数据和业务逻辑。

具体的重要代码如下：

|  |
| --- |
| <div class="aside">  <div class="menu">  <h1 style="padding: 60px 0;">大学生健康管理系统</h1>  <div class="menu-body">  <div class="item1" v-for="(item,index) in generateMenu" :style="{'background-color': item.color}" :key="index">  <div class="item-div" @click="toDetail(item.url)">  <!--<img :src="item.pic" style="width: 100px;height: 100px;"/>-->  <i :class="item.icon"></i>  <h3>{{item.name}}</h3>  </div>  </div>  </div>  </div>  </div>  @GetMapping(value = "get/{id}")  public ResponseEntity<T> get(@PathVariable("id")ID id){  T t = null;  try {  t = service.selectByKey(id);  }catch (Exception e){  log.error(e.getMessage());  throw new MyException(ExceptionEnums.GET\_ITEM);  }  return ResponseEntity.ok(t);  }  @GetMapping("/getMenuByUserId")  ResponseEntity< List<ResourceVO> > getMenuByUserId(@RequestParam("userId") Integer userId,@RequestParam("typeId") Integer typeId){  List<ResourceVO> resourceVOS;  try {  resourceVOS = resourceService.getResourceTreeByUserId(userId,typeId);  }catch (Exception e){  log.error(e.getMessage());  throw new MyException(ExceptionEnums.RESOURCE\_FOUNT\_FAIL);  }  if(resourceVOS == null){  throw new MyException(ExceptionEnums.RESOURCE\_FOUNT\_FAIL);  }  return ResponseEntity.ok(resourceVOS);  } |

## 6.2 组员2（子目录是组员2的功能）

### 6.2.1个人信息界面效果图

这张个人健康效果图为学生提供了一个直观的身高体重分析。图中清晰地展示了学生的当前身高（185cm）和体重（70kg），并计算出其体质指数（BMI）为20.46，标明了属于健康体重范围[12]。此外，图表以条形图的形式呈现了体重数据，使得学生可以快速把握自己的体重变化情况。整体而言，该效果图通过简洁明了的视觉呈现，帮助学生了解自己的健康状况，并鼓励他们维持在健康的BMI指数范围内。

具体效果如图6-5：



**图6-5 个人信息模块**

### 6.2.2 个人信息界面实现

个人页面的实现依赖于前端JavaScript代码与后端API的交互。页面初始化时，`initData`函数通过解析本地存储中的用户信息，并发送HTTP GET请求到`/user/get/`接口，根据用户ID获取用户数据，然后渲染到界面上。用户对健康信息做出更改后，点击保存会触发`saveData`函数，该函数通过HTTP POST请求将更新后的数据发送到`/user/update`接口，实现数据的持久化存储。后端Spring Boot控制器使用`@GetMapping`和`@PostMapping`注解来处理这些请求，提供数据检索和更新服务。成功更新后，前端会收到响应，更新本地存储，并可能刷新页面以显示最新数据。这个过程确保了用户界面的响应性和数据的实时同步。

具体实现代码如下：

|  |
| --- |
| initData(){  let user = JSON.parse(localStorage.getItem("user"));  this.$http.get("/user/get/"+user.id).then(res=>{  if (res.status===200){  this.userData = res.data  }  })  },  saveData(){  this.$http.post("/user/update",this.userData).then(res=>{  if (res.status===200){  Message.success("操作成功");  localStorage.setItem("user",JSON.stringify(this.userData));  this.disable = !this.disable;  this.initData();  }  })  },  @ApiOperation(value = "基础接口: 修改数据")  @PostMapping(value = "update")  public ResponseEntity<T> update(@RequestBody T entity){  try {  if (service.update(entity)<1){  throw new MyException(ExceptionEnums.UPDATE\_ERROR);  }  }catch (Exception e){  log.error(e.getMessage());  throw new MyException(ExceptionEnums.UPDATE\_ERROR);  }  return ResponseEntity.ok(entity);  }  @ApiOperation(value = "基础接口: 返回指定ID的数据")  @GetMapping(value = "get/{id}")  public ResponseEntity<T> get(@PathVariable("id")ID id){  T t = null;  try {  t = service.selectByKey(id);  }catch (Exception e){  log.error(e.getMessage());  throw new MyException(ExceptionEnums.GET\_ITEM);  }  return ResponseEntity.ok(t);  } |

## 6.3 健康档案管理界面

### 6.3.1 健康档案管理界面效果图

健康档案管理功能在大学生健康管理系统中提供了一套全面的记录和编辑学生健康信息的工具。从提供的图片来看，系统允许用户查看和编辑学生的基本信息，包括姓名、性别、学号、学院、专业、体检日期等。用户可以输入或更新学生的文化程度、现居住址、工作单位等详细资料。此外，系统还特别关注学生的体检信息，提供了记录和编辑体检结果的功能，如视力检查等。用户能够添加新的健康档案，选择学生、记录体检日期，并输入学生的基本健康信息。这些档案不仅帮助学生和医务人员跟踪健康状况，而且为学校管理层提供了重要的数据支持，以制定健康促进计划和疾病预防策略。整体而言，健康档案管理功能通过提供一个集中的平台，使得健康信息的管理更加高效、有序，同时确保了信息的安全性和可访问性。

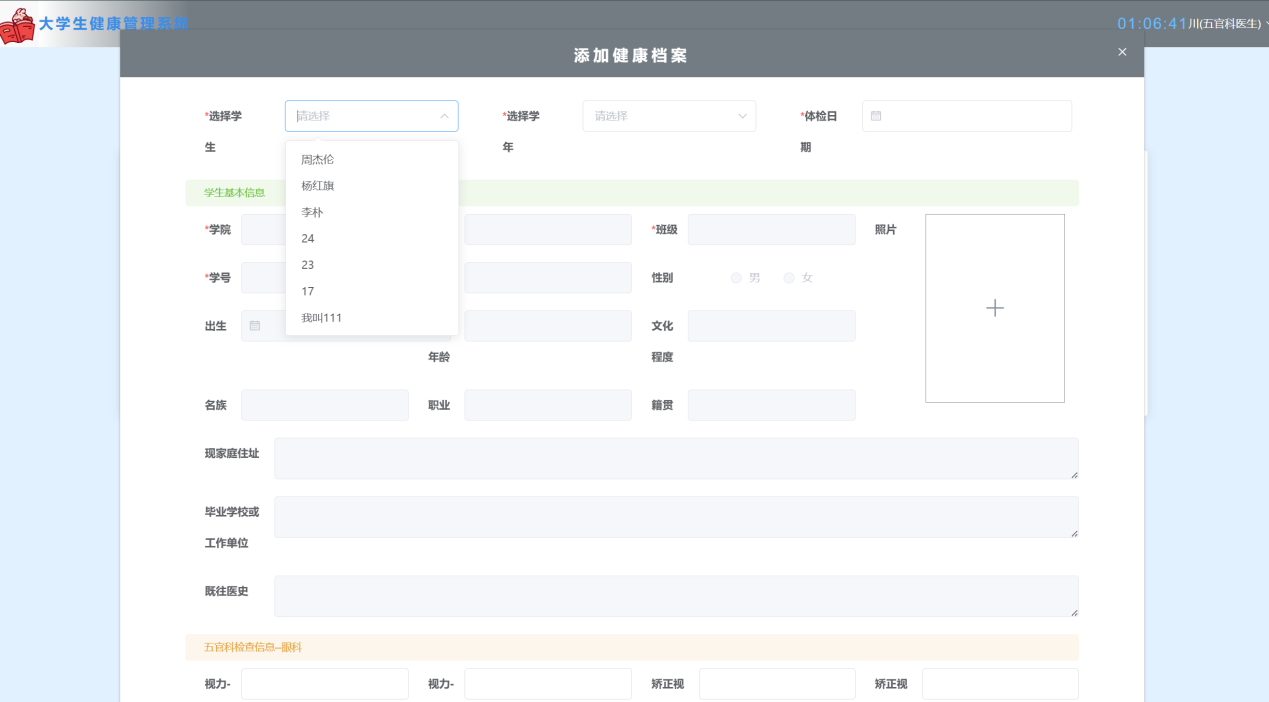
具体效果图如图6-6、6-7、6-8：



**图6-6 健康档案管理**



**图6-7 编辑健康档案**



**图6-8 添加健康档案**

### 6.3.2 健康档案管理界面实现

健康档案管理功能通过前端的`initData`方法实现界面的初始化和数据加载。该方法根据用户角色动态调整界面显示，隐藏或显示相应的操作选项，并通过POST请求从后端获取健康档案列表，更新到前端表格中。后端提供了`save`、`update`和`delete`方法来处理健康档案的增加、修改和删除操作，这些方法通过权限注解确保只有授权用户才能执行。`getCurrentCheckInfo`方法用于检索特定用户的最新体检信息，而`judgeCheckIsExist`方法检查特定用户和年份的体检记录是否存在，以避免重复。整个实现流程确保了健康档案管理功能既灵活又安全，允许不同角色的用户根据权限执行相关操作，同时后端API保证了数据的一致性和完整性。

具体的重要代码如下：

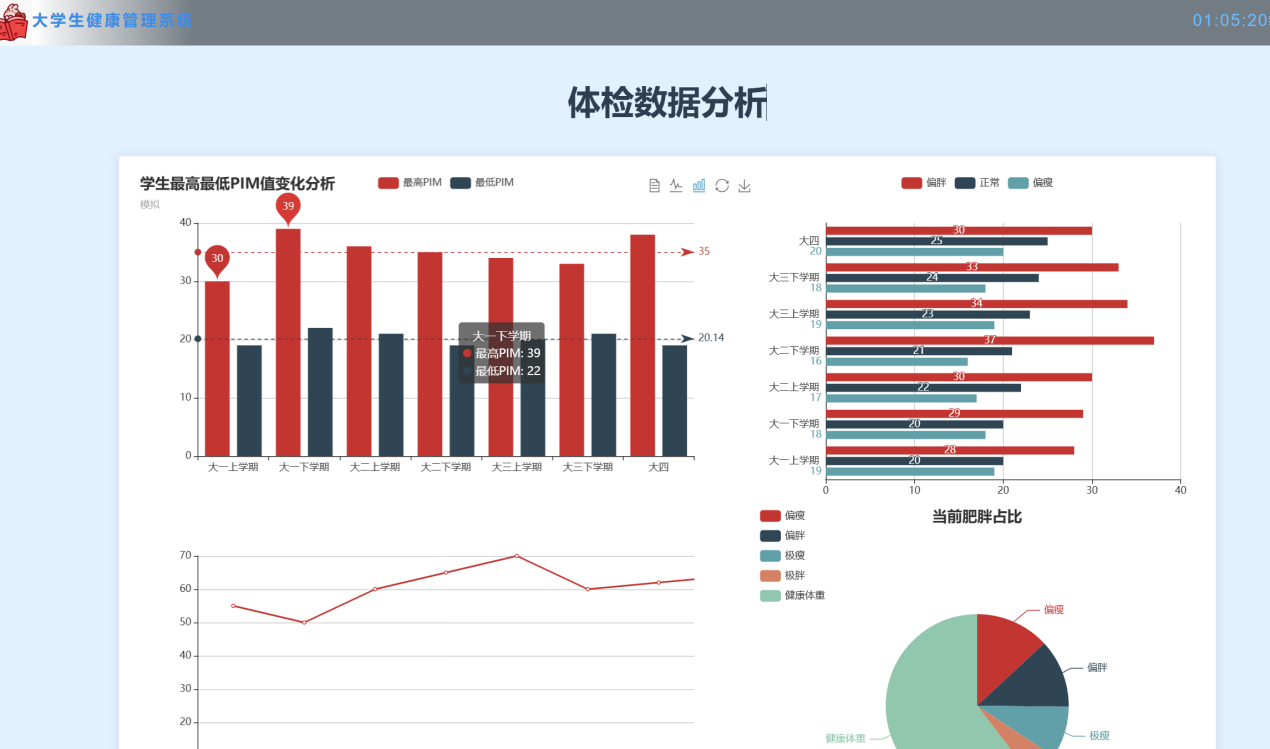
|  |
| --- |
| initData(){  this.$http.post("/checkInfo/page",this.searchData).then(res=>{  if (res.status===200){  this.tableData = res.data.list;  this.searchData.total = res.data.total;  }  });  let roleData = JSON.parse(localStorage.getItem("role"));  if (roleData.roleName === '五官科医生'){  this.showFlag1 = true;  this.showFlag2 = false;  this.showFlag3 = false;  this.showFlag4 = false;  }else if (roleData.roleName === '外科医生'){  this.showFlag1 = false;  this.showFlag2 = true;  this.showFlag3 = false;  this.showFlag4 = false;  }else if (roleData.roleName === '内科医生') {  this.showFlag1 = false;  this.showFlag2 = false;  this.showFlag3 = true;  this.showFlag4 = false;  }else if (roleData.roleName === '负责医师') {  this.showFlag1 = false;  this.showFlag2 = false;  this.showFlag3 = false;  this.showFlag4 = true;  }else if(roleData.roleName === '其他科医生'){  this.showFlag4 = true;  } else {  this.showFlag1 = true;  this.showFlag2 = true;  this.showFlag3 = true;  this.showFlag4 = true;  }  },  @Override  @RequiresPermissions("checkInfo:add")  public ResponseEntity<CheckInfo> save(@RequestBody CheckInfo entity) {  return super.save(entity);  }  @Override  @RequiresPermissions("checkInfo:update")  public ResponseEntity<CheckInfo> update(@RequestBody CheckInfo entity) {  return super.update(entity);  }  @Override  @RequiresPermissions("checkInfo:delete")  @GetMapping(value = "delete/{id}")  public ResponseEntity<String> delete(@PathVariable("id") Integer integer) {  return super.delete(integer);  }  @GetMapping("/getCurrentCheckInfo/{userId}")  ResponseEntity<CheckInfo> getCurrentCheckInfo(@PathVariable("userId") Integer userId){  CheckInfo checkInfo = checkInfoDao.getCurrentCheckInfo(userId);  if (checkInfo == null){  throw new MyException(ExceptionEnums.NO\_CHECK\_INFO);  }else {  return ResponseEntity.ok(checkInfo);  }  }  @GetMapping("/judgeCheckIsExist")  ResponseEntity judgeCheckIsExist(Integer userId,String checkYear){  Example example = new Example(CheckInfo.class);  example.createCriteria().andEqualTo("userId",userId).andEqualTo("checkYear",checkYear);  List<CheckInfo> checkInfos = this.service.selectByExample(example);  if (checkInfos==null || checkInfos.size()==0){  return ResponseEntity.ok(true);  }else {  return ResponseEntity.ok(false);  }  } |

## 6.4 体检数据分析界面

### 6.4.1 体检数据分析界面效果

体检数据分析模块在学生健康管理系统中扮演着至关重要的角色，它通过展示学生的最高和最低体重指数（PIM）值，帮助用户识别体重变化趋势和潜在健康问题。该模块能够分析不同学期的PIM值变化，提供肥胖占比的统计数据，从而快速了解学生体重分布状况。此外，它还支持时间序列分析，使用户能够追踪长期体重变化模式。通过直观的图表展示，这一模块使得复杂的数据易于理解，同时保证了信息的实时更新，为学校提供了一个监测学生健康状况的有效工具。

具体效果如图6-9、6-10：



**图6-9 体检数据分析**



**图6-10 体检数据下载**

### 6.4.2 体检数据分析界面实现

体检数据分析模块通过前端的`getAnalysisData`函数和ECharts图表库实现了数据的动态可视化展示。该函数初始化了四个图表实例，分别对应页面上的不同容器元素，并通过调用`setOption`方法应用了预先定义好的配置选项，这些选项详细规定了图表的类型、数据系列、轴标签等关键信息。图表类型可能包括展示BMI值变化趋势的折线图、不同学期BMI值对比的柱状图、体重分类占比的饼图以及时间序列分析图。这些图表直观地呈现了学生的最高和最低BMI值变化、不同学期间的BMI对比、体重分布的占比以及从大一到大四的BMI值趋势。后端API如`/getStudentHealthData`或`/getBMITrends`提供了数据支持，通过处理前端请求并返回所需的体检数据，包括BMI值和体重分类信息。后端还可能执行数据聚合和计算，确保前端接收到的数据可以直接用于图表配置[13]。整个模块的设计提供了一个全面、直观的体检数据分析工具，帮助用户深入理解学生健康状况。

具体重要代码如下：

|  |
| --- |
| getAnalysisData(){  let firstEcharts = echarts.init(document.getElementById("first"));  let firstOptions = {  //...  };  let secondEcharts = echarts.init(document.getElementById("second"));  let secondOptions = {  //...  };  let thirdEcharts = echarts.init(document.getElementById("third"));  let thirdOption = {  //...  };  let fourEcharts = echarts.init(document.getElementById("four"));  let fourOption = {  };  firstEcharts.setOption(firstOptions);  secondEcharts.setOption(secondOptions);  thirdEcharts.setOption(thirdOption);  fourEcharts.setOption(fourOption);  } |

# 结论与展望

在本次研究中，我们成功开发了一个基于JavaEE技术的大学生体检档案管理系统，该系统采用Spring Boot框架和Vue.js前端框架，实现了体检档案的电子化管理。系统的主要功能包括用户管理、健康档案管理、权限管理、健康知识阅读、角色管理和健康建议管理等，全面覆盖了体检档案管理的各个环节。实际测试表明，系统运行稳定，功能完善，显著提升了档案管理的效率和用户体验。系统的实施，不仅减轻了管理人员的工作负担，也为大学生提供了便捷的健康信息查询和个性化健康建议服务，具有重要的应用价值和广阔的应用前景。

系统的技术实现与创新体现在采用了面向对象的设计方法，结合了MyBatis与MySQL数据库的交互，实现了数据的高效管理和操作。开发过程中遵循了模块化和组件化的原则，提高了代码的可维护性和可扩展性。在性能方面，系统在关键性能指标上均达到了预期目标，如系统启动迅速，首页加载和登录响应时间均在用户可接受的范围内，档案管理操作的响应时间迅速，保证了用户在进行档案操作时的流畅体验。用户体验方面，系统界面友好，操作简便，用户可以轻松地进行个人信息的查看和编辑，以及健康档案的管理和查询。系统还提供了个性化的健康建议和健康知识阅读，增强了用户的参与感和满意度。

尽管本系统在功能和性能上均表现良好，但仍有进一步提升和完善的空间。未来可以考虑增加更多的个性化服务，如健康风险评估、疾病预防建议等，以提供更全面的健康管理服务。随着用户数量的增加和数据量的增长，系统的性能可能会面临新的挑战，因此，持续的性能优化和资源管理将是未来工作的重点。在数据安全和隐私保护方面，系统需要进一步加强，可以考虑引入更先进的加密技术和访问控制机制，确保用户数据的安全。此外，建立一个有效的用户反馈机制，及时收集用户的意见和建议，不断优化系统功能和用户体验，也是未来工作的重要方向。

随着新技术的不断涌现，如人工智能和大数据分析，未来的研究可以考虑将这些技术应用于健康管理系统中，提高系统的智能化水平和处理能力[16]。系统未来可以考虑在其他高校或医疗机构中推广应用，以实现更广泛的社会价值。通过不断的努力和创新，大学生健康管理系统有望成为健康管理领域的佼佼者，为用户提供更高质量的服务。同时，本研究的成果也为其他领域的档案电子化管理提供了参考和借鉴。

# 参考文献

1. 庄彦,未培.基于云数据库的大学生健康档案管理系统的设计与实现[J].辽宁科技学院学报,2016,18(01):17-19.
2. 陈淦涛.AI时代人工智能在健康教育领域的应用探索与思考[C]//广州市卫生健康宣传教育中心.广州市第十四届健康教育与健康促进学术交流活动稿集.广州市黄埔区南岗街社区卫生服务中心;,2024:2.DOI:10.26914/c.cnkihy.2024.014211.
3. Maya S ,Raymond C .Frontend Development Projects with Vue.js 3:Learn the fundamentals of building scalable web applications and dynamic user interfaces with Vue.js[M].Packt Publishing Limited:2023-03-17.DOI:10.0000/9781803236315.
4. Liu Y .Campus Second-Hand Textbook Trading Platform based on Vue 3 and Spring Boot[J].International Core Journal of Engineering,2024,10(6):
5. 罗辉琼,贺超波,李姣姣,等.基于SpringMVC及MyBatis的高校学术成果库设计与实现[J].现代信息科技,2024,8(04):25-29+35.DOI:10.19850/j.cnki.2096-4706.2024.04.006.
6. 滕明星,张恒,李超.ECharts多端图表引擎在变形监测数据可视化中的应用[J].北京测绘,2024,38(03):318-324.DOI:10.19580/j.cnki.1007-3000.2024.03.008.
7. 李大伟.基于人工智能的计算机应用软件开发技术分析[J].软件,2023,44(09):124-126.
8. 马晓光,封桂荣,孙睿,等.自主可控软件性能需求与关键技术研究[J].工业控制计算机,2022,35(03):1-3.
9. 孔文秀,慈航,李京,等.基于MVC架构的枣庄市重点企业防雷减灾平台[J].山西电子技术,2024,(02):91-93.
10. 柯妍,孙佳留,朱士飞,等.基于MySQL的煤质信息数据库设计[J].资源信息与工程,2024,39(03):117-121.DOI:10.19534/j.cnki.zyxxygc.2024.03.021.
11. 马瑀浓,刘滢,孟丹.大学生健康管理系统的设计与实现[J].数码世界,2019,(06):96.
12. 邵媛.大学生体质健康促进模型的构建研究[J].成都体育学院学报,2024,50(04):124-131.DOI:10.15942/j.jcsu.2024.04.015.
13. 刘慧玲,谭定英,陈平平.基于SpringBoot和Vue.js的大学生团队管理系统的设计[J].电脑编程技巧与维护,2024,(03):120-122.DOI:10.16184/j.cnki.comprg.2024.03.039.
14. ]张耀文.软件测试对于信息技术发展的影响[J].大众标准化,2024,(01):164-166.
15. ]Mehmood N M ,Jürgen B ,Kai P .Checklists to support decision-making in regression testing[J].The Journal of Systems & Software,2023,202
16. 俞成功.基于大数据分析的医院病案质量自动核查系统[J].自动化技术与应用,2024,43(07):184-188.DOI:10.20033/j.1003-7241.(2024)07-0184-05.

# 致 谢

在完成这份实习实训论文的过程中，我深刻感受到了来自不同方向的支持与帮助，对此我怀有深深的感激之情。

首先，我要向我的实习公司表达最诚挚的感谢。公司不仅提供了一个充满活力和挑战的工作环境，更让我有机会将所学知识应用于实践中。特别要感谢公司里的导师们，像卢老师这样认真且耐心的导师们，你们不仅在工作中以耐心和专业的态度指导我，而且在学术探索上也给予了我巨大的帮助。你们的经验分享和专业知识的教授，让我在专业领域的认识更加深入，对工作的理解更加透彻。

其次，我要感谢我的母校，感谢学校提供的实习机会和全方位的支持。软件工程专业的老师们，你们严谨的学术态度和深入浅出的教学方式，为我打下了坚实的理论基础。正是这些知识，让我在实习岗位上能够迅速适应，有效解决实际问题。

此外，我还要特别感谢我的组员们。在共同的项目开发过程中，你们的智慧、协作和不懈努力，是我们团队能够成功的关键。每一个讨论、每一次头脑风暴、每一个共同工作到深夜的时刻，都是我宝贵的记忆。我们之间的相互支持和团队精神，是完成这项工作不可或缺的部分。

我也要感谢所有在这段旅程中给予我帮助的人，包括给予我灵感和启发的同学、提供资源和帮助的人员，以及所有在背后默默支持我的朋友和同事。请接受我最诚挚的谢意。这份项目和论文的完成，是我们共同努力的成果，我将继续携带着这份感激，走向未来的学术和职业生涯。