

武汉纺织大学

计算机与人工智能学院

Web应用开发课程设计报告

2020 ~2021学年第2学期

|  |  |
| --- | --- |
| 大作业题目 | 健康管家管理平台系统 |
| 学生姓名 | 张 洁 |
| 指导教师 | 黄俊杰 |
| 评定成绩 |  |
| 填写时间 | 2020年7月8日 |
|  | |

摘 要

近年以来，越来越快的生活节奏，不可避免的生活压力和不健康规律的生活作息使得当代人忙于工作，忽略了自身的健康管理，因为去医院体检一次得花费一定得时间和精力。那么怎么使管理健康状况变得简单及时可视化呢？通过手机APP监测管理便是一个很好的解决办法。

健康管家APP是一款健康类安卓客户端应用，包含运动计步、睡眠管理、饮食贴士、寻医生、找医院、查疾病、社区咨询、健康测评、自我激励、个人中心等多个模块，而健康管家管理平台是健康管家APP数据的Web管理端，其实际上是一款Web应用。通过健康管家管理平台管理用户和平台的一系列信息，使用简单，并且达到了基本功能覆盖全的标准。

本文基于当代人对健康管理的实际情况，结合整个健康管理监测的需求分析，采用J2EE的体系化设计模式，使用Spring MVC+Mybatis+Maven框架，以PostgreSQL为关系型数据库平台，设计了健康管理平台的Web管理端。系统主要采用B/S系统架构,并重点考虑了对多种业务需求进行设计。该系统具有分多种管理员角色登录，进行不同权限的管理等功能。

本文分析了当代人对实时健康管理的需求，重点对系统的超级管理员功能和其他角色的模块管理功能进行了分析，并给出了系统主要功能的实现方法，展示了最终的实现效果和测试效果。

关键词：Java； 健康管理； Spring MVC

**ABSTRACT**

In recent years, the increasingly fast pace of life, the inevitable stress of life and unhealthy routine make the contemporary people busy with work and neglect their own health management, because it takes time and effort to go to the hospital for a medical checkup. So how to make the management of health status become simple and timely visualization? A good solution is to monitor and manage through a mobile app.

Health Manager APP is a health Android client application, including exercise pedometer, sleep management, diet tips, find a doctor, find a hospital, check the disease, community consultation, health assessment, self-motivation, personal center and many other modules, and Health Manager Management Platform is the Web management side of Health Manager APP data, which is actually a Web application. It is easy to use and meets the standard of full coverage of basic functions by managing a series of information of users and the platform through HealthManager Management Platform.

In this paper, based on the actual situation of contemporary health management and the analysis of the whole health management monitoring requirements, we designed the Web management side of the health management platform using the J2EE systematic design pattern, Spring MVC + Mybatis + Maven framework, and PostgreSQL as the relational database platform. The system mainly adopts B/S system architecture, and focuses on the design of various business requirements. The system has multiple administrator roles to log in and manage different rights.

This paper analyzes the contemporary demand for real-time health management, focuses on the super administrator function and the module management function of other roles, gives the implementation method of the main functions of the system, and shows the final implementation and testing results.

**Keywords：Web; Health Management; Spring MVC**

目 录

[1 引言 1](#_Toc77189248)

[1.1 健康管家Web管理端开发背景 1](#_Toc77189249)

[1.2 健康管理行业发展状况 1](#_Toc77189250)

[1.3 系统开发的目的与意义 2](#_Toc77189251)

[2 系统技术概括 3](#_Toc77189252)

[2.1 系统前端技术概括 3](#_Toc77189253)

[2.1.1 jQuery 3](#_Toc77189254)

[2.1.2 jQuery EasyUI 4](#_Toc77189255)

[2.2 系统后端技术概括 4](#_Toc77189256)

[2.2.1 PostgreSQL数据库 4](#_Toc77189257)

[2.2.2 Tomcat 服务器 5](#_Toc77189258)

[2.2.3 MVC 设计模式 6](#_Toc77189259)

[2.2.4 Maven工具 7](#_Toc77189260)

[2.2.5 MyBatis框架 7](#_Toc77189261)

[2.2.6 Spring框架 8](#_Toc77189262)

[2.2.7 Spring MVC框架 8](#_Toc77189263)

[3 系统分析与设计 9](#_Toc77189264)

[3.1 健康管理系统概述 9](#_Toc77189265)

[3.2 健康管理系统需求概述 10](#_Toc77189266)

[3.2.1 其他管理员需求分析 10](#_Toc77189267)

[3.2.2 超级管理员需求分析 10](#_Toc77189268)

[3.3 系统设计 10](#_Toc77189269)

[3.3.1 数据库设计 10](#_Toc77189270)

[3.3.2 其他管理员功能设计 12](#_Toc77189271)

[3.3.3 超级管理员功能设计 13](#_Toc77189272)

[4 系统实现 14](#_Toc77189273)

[4.1 数据库实现 14](#_Toc77189274)

[4.2 其他管理员功能实现 16](#_Toc77189275)

[4.2.1 登录功能 17](#_Toc77189276)

[4.2.2 修改密码 17](#_Toc77189277)

[4.2.3 模块管理 18](#_Toc77189278)

[4.3 超级管理员功能实现 20](#_Toc77189279)

[4.3.1 用户管理 21](#_Toc77189280)

[4.3.2 菜单管理 22](#_Toc77189281)

[4.3.3 角色管理 23](#_Toc77189282)

[4.3.4 资源管理 24](#_Toc77189283)

[5 总结与展望 25](#_Toc77189284)

[5.1 总结 25](#_Toc77189285)

[5.2 展望 26](#_Toc77189286)

[参考文献 27](#_Toc77189287)

1. 引言
   1. 健康管家Web管理端开发背景

紧张的工作节奏和生活的压力、个人不良的工作生活习惯、以及伴随工作压力而来的家庭关系、人际关系紧张等因素使得当代人成为亚健康的群体。入职体检，定期体检工作，甚至个人自愿定期进行身体健康检查等都离不开医院的体检。人们对体检工作越来越重视，目前，传统的体检一直延用手工管理的方式，缺乏一种实时简单的，对健康信息可视化的管理方式，对健康管理平台需求迫切。

为了对用户进行规范化管理和实时管理用户每个健康项的数据，实现对健康信息分析的实时可视化，根据日常行为，建立此方面的数据模型，根据该数据模型进行分析预判，对用户的健康状况做出合理正确的提示与干预。健康管理平台不仅包含了对个体健康曲线的实时可视化，同时包含用户寻求健康帮助的渠道管理和个人测评激励管理的部分，并以此为基础对个体日常行为的干预显示，以此为背景设计出了健康管理平台的Web管理端，能够初步的解决这些需求。

论文所做的工作具有一定的社会意义，能够解决实际问题。

* 1. 健康管理行业发展状况

健康管理的概念最早见于美国，早在1929年美国洛杉矶水利局就成立了最早的健康维护组织。早在20世纪70年代的美国，由于健康保险的发展以及医疗服务模式的改变，要求医生能更好地指导病人进行疾病预防以及更加节省医疗费用。健康服务组织通过对人群的健康状况进行评价和疾病的风险评估，科学预测疾病的危险因素，制定具体的、有针对性的预防和保健措施。通过健康管理美国的医疗费比以前节省了大约10%，同时避免了疾病的发生，提高了生活质量。资料显示，通过健康管理计划，在1978-1983年，美国人的胆固醇水平下降了2%；高血压水平下降了4%；冠心病发病率下降了16%。美国密执安大学健康管理研究中心主任第·艾鼎敦博士（Dee.W.Edington）曾经提出美国20多年的研究得出了9:1定律结论，即90%的人通过健康管理后，医疗费用降到了原来的10%；10%的人未做健康管理，医疗费用比原来上升了90%。通过经济学分析发现，在健康管理方面投入1元钱，相当于减少3-6元医疗费用的开销。如果加上由此产生的劳动生产率提高的回报，实际效益达到投入的8倍[1]。

健康管理在中国刚刚起步，是一个朝阳的产业。目前在中国大陆地区仅有少数专业的健康管理机构，大部分为医院及体检中心的附属部门。健康管理的从业人数没有准确的数据，估计全国在10万人以上，也就是说我国享受科学、专业的健康管理服务的人数只占人口总数的万分之二，这一比例与美国70%居民能够在健康管理公司或企业接受完善的服务相去甚远。而且目前健康管理仍然主要依赖于简单的单次体检或就诊数据由医生提供一些生活习惯的建议和意见，并没有把个人的历史病历以及日常行为对个人健康的影响考虑进去。此外，目前的健康管理也主要是针对已感觉不适甚至是患病的患者，还未做到“早筛选、早预防、早发现、早诊断”的动态追踪的健康管理模式。

* 1. 系统开发的目的与意义

健康管理既是一个概念，也是一种方法，更是一套完善、周密的服务程序，其目的在于使病人以及健康人更好地拥有健康、恢复健康、促进健康并尽量节约经费开支，有效降低医疗支出。健康管理有如下几大作用：一是可以根据日常行为预测身体健康状况，以此为基础提供健康保障方案；二是可以长期跟踪用户的健康，建立用户个人健康档案；三是可以节省维护健康的时间和金钱，提高保健效率。

建立健康管理Web管理端的目的，实时管理用户健康信息，管理资源渠道和相关渠道服务。而健康信息信息管理系统的管理和利用，不仅需要具备基础医学知识、临床医学知识、流行病学知识、计算机技术、数学统计知识等综合素质的专业人员，同时需要有庞大的保健医学、预防医学、临床医学、康复医学等专业的资深专家团队的支持，才能够为个人提供一系列的健康管理服务[2]。

当今世界，信息管理数字化非计算机莫属，只有利用计算机技术，使用统一标准的健康体检系统表格，开发健康管理系统的系统软件，对用户的健康状况实现实时的动态管理。健康信息管理系统以计算机为工具，通过对用户体检所得的数据进行信息管理，把管理人员从繁琐的数据计算处理中解脱出来，为更好的帮组用户身体健康监督检查，从而全面提高质量。具体通过本系统可以对用户的基本健康习惯及状况进行各种必要的统计和分析。

本项目将针对当代亚健康群体研发多功能的健康数据采集系统，对睡眠习惯、运动习惯、生活习惯等健康相关信息建立自我健康管理系统，对当代个人广泛存在的亚健康状态进行综合客观评价和风险预测，最终实现针对当代亚健康群体健康实时的管理。

1. 系统技术概括
   1. 系统前端技术概括
      1. jQuery

jQuery是一个快速的简洁的框架，它封装了JavaScript常用的功能代码，提供了一种简便的JavaScript设计模式，优化了HTML文档操作、事件处理、动画设计和AJAX交互，它是目前最受欢迎的JavaScript框架。

一般来说，jQuery具有以下特点：

1. 选择器使用范围广

jQuery允许开发者使用从CSS1到CSS3几乎所有的选择器，以及JQuery独创的高级而且复杂的选择器，另外还可以加入插件使其支持XPath选择器，甚至开发者可以编写属于自己的选择器。由于JQuery支持选择器这一特性，因此有一定CSS经验的开发人员可以很容易的切入到JQuery的学习中来。

1. DOM操作方便

jQuery封装了大量常用的DOM操作，使开发者在编写DOM操作相关程序的时候能够得心应手。JQuery轻松地完成各种原本非常复杂的操作，让JavaScript新手也能写出出色的程序。

1. AJAX异步处理

jQuery提供多个与AJAX有关的方法，在不重载整个网页的情况下，AJAX通过后台加载数据，并在网页上显示。

1. 动态修改页面样式

使用jQuery我们可以动态的修改页面的CSS即使在页面呈现以后。jQuery仍然能够改变文档中某个部分的类或者个别的样式属性。此外，jQuery可以为页面添加动态效果，jQuery中内置的一批淡入、擦除之类的效果，以及制作新效果的工具包，为此提供了便利。

1. 响应用户的交互操作

jQuery提供了截获形形色色的页面事件（比如用户单击某个链接）的适当方式，而不需要使用事件处理程序拆散HTML代码。此外，它的事件处理API也消除了经常困扰Web开发人员浏览器的不一致性。

* + 1. jQuery EasyUI

jQuery EasyUI是一种基于jQuery的用户界面插件集合，它使web开发人员能快速地在流行的jQuery核心和HTML5上建立程序页面。其功能相对没extjs强大，但页面也是相当好看的，同时页面支持各种themes以满足使用者对于页面不同风格的喜好。一些功能也足够开发者使用，相对于extjs更轻量[4]。

一般来说，jQuery EasyUI具有以下特点：

1. 基于jquery用户界面插件的集合

jQuery EasyUI 提供了用于创建跨浏览器网页的完整的组件集合，包括功能强大的 datagrid（数据网格）、treegrid（树形表格）、 panel（面板）、combo（下拉组合）等等。 用户可以组合使用这些组件，也可以单独使用其中一个。

1. 为一些当前用于交互的js应用提供必要的功能

jQuery Easy UI优化了 Ajax方式提交表单，前后台交互方面 jQuery 最常用的就是 Ajax 的包装的方法，实现异步提交信息，并且是局部刷新网页。

1. 渲染方式

jQuery EasyUI支持两种渲染方式分别为javascript方式和html标记方式。

1. 代码量少

jQuery EasyUI用于写后台管理系统的可以很清晰显示数据的，便于后台人员对信息进行管理，用很少的代码实现一个网页的架构。

* 1. 系统后端技术概括
     1. PostgreSQL数据库

PostgreSQL是一个功能非常强大的、源代码开放的客户/服务器关系型数据库管理系统，在灵活的BSD许可证下发行，支持大部分 SQL标准并且提供了许多其他现代特性：复杂查询、外键、触发器、视图、事务完整性、多版本并发控制。同样，PostgreSQL 可以用许多方法扩展，比如， 通过增加新的数据类型、函数、操作符、聚集函数、索引。免费使用、修改、和分发 PostgreSQL[3]。它具有以下优点：

1. PostgreSQL 是一个开源的，免费的，同时非常强大的关系型数据管理系统。
2. PostgreSQL 背后有热忱而经验丰富的社区，可以通过知识库和问答网站获取支持，全天候免费。
3. 即使其本身功能十分强大，PostgreSQL 仍附带有许多强大的开源第三方工具来辅助系统的设计、管理和使用。
4. 可以用预先存储的流程来程序性扩展 PostgreSQL。
5. PostgreSQL 不只是一个关系型数据库，还是一个面向对象数据库——支持嵌套，及一些其他功能。
   * 1. Tomcat 服务器

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。其主要组件有服务器Server，服务Service，连接器Connector、容器Container。连接器Connector和容器Container是Tomcat的核心。

Tomcat Server处理一个HTTP请求的过程如下：

1. 用户点击网页内容，请求被发送到本机端口8080，被在那里监听的Coyote HTTP/1.1 Connector获得。
2. Connector把该请求交给它所在的Service的Engine来处理，并等待Engine的回应。
3. Engine获得请求localhost/test/index.jsp，匹配所有的虚拟主机Host。
4. Engine匹配到名为localhost的Host（即使匹配不到也把请求交给该Host处理，因为该Host被定义为该Engine的默认主机），名为localhost的Host获得请求/test/index.jsp，匹配它所拥有的所有的Context。Host匹配到路径为/test的Context（如果匹配不到就把该请求交给路径名为“”的Context去处理）。
5. path=“/test”的Context获得请求/index.jsp，在它的mapping table中寻找出对应的Servlet。Context匹配到URL PATTERN为\*.jsp的Servlet,对应于JspServlet类。
6. 构造HttpServletRequest对象和HttpServletResponse对象，作为参数调用JspServlet的doGet（）或doPost（）.执行业务逻辑、数据存储等程序。
7. Context把执行完之后的HttpServletResponse对象返回给Host。
8. Host把HttpServletResponse对象返回给Engine。
9. Engine把HttpServletResponse对象返回Connector。
10. Connector把HttpServletResponse对象返回给客户Browser。
    * 1. MVC 设计模式

MVC即Model-View-Controller（模型-视图-控制器）是一种软件设计模式，这3个模块又可被称为业务层-视图层-控制层。顾名思义，它们三者在应用程序中的主要作用如下：

　　业务层：负责实现应用程序的业务逻辑，封装有各种对数据的处理方法。它不关心它会如何被视图层显示或被控制器调用，它只接受数据并处理，然后返回一个结果。

　　视图层：负责应用程序对用户的显示，它从用户那里获取输入数据并通过控制层传给业务层处理，然后再通过控制层获取业务层返回的结果并显示给用户。

　　控制层：负责控制应用程序的流程，它接收从视图层传过来的数据，然后选择业务层中的某个业务来处理，接收业务层返回的结果并选择视图层中的某个视图来显示结果。

MVC是一种设计思想，它并没有统一的标准，其中一种MVC思想的典型实现为JavaBean（模式）+jsp（视图）+servlet（控制器），也可以将JavaBean中的业务逻辑再单独列出来，形成service（模式）+JavaBean（数据集）+jsp（视图）+servlet（控制器）的结构。

在项目中使用MVC框架具有以下三点优势：

1. 有利于代码的复用。
2. 有利于开发人员分工。
3. 有利于降低程序模块间的耦合，便于程序的维护与扩展。
   * 1. Maven工具

Maven是一个项目管理工具，它包含了一个项目对象模型（Project Object Model）、一组标准集合、一个项目生命周期（Project Lifecycle）、一个依赖管理系统（Dependency Management System）和用来运行定义在生命周期阶段（Phase）中插件（Plugin）目标（Goal）的逻辑。Maven能够方便地用于管理项目报告、生成站点、管理JAR文件等。对于Maven，用户需要做的只是创建一个pom.xml，它跨平台，最大限度地消除了构建的重复。Maven具有以下功能：

Maven对所依赖的包有明确的定义，使用哪个包，版本是多少，一目了然。

Maven是基于中央仓库的编译，当编译的时候，Maven会自动在仓库中找到相应的包，如果本地仓库没有，则从设定好的远程仓库下载到本地，这个过程都是自动的。

* + 1. MyBatis框架

MyBatis是支持定制化SQL、存储过程以及高级映射的优秀的持久层框架，它避免了几乎所有的JDBC代码和手动设置参数及获取结果集。MyBatis可以对配置和原生Map使用简单的XML或注解，将接口和Java的POJO映射成数据库中的记录。

MyBatis框架的优点在于：

1. 与JDBC相比，减少了50%以上的代码量。
2. MyBatis是最简单的持久化框架，小巧且简单易学。
3. MyBatis相当灵活，不会对应用程序或者数据库的现有设计强加任何影响，SQL写在XML里，从程序代码中彻底分离，降低了耦合度，便于统一管理和优化，并可重用。
4. 提供了XML标签，支持编写动态SQL语句。
5. 提供了映射标签，支持对象与数据库的ORM映射。
   * 1. Spring框架

Spring是一个开放源代码的设计层面框架，解决的是业务逻辑层和其他各层的松耦合问题。Spring框架使用依赖注入，不仅可以为Bean注入普通的属性值，还可以注入其他Bean的引用，相当于在spring这个容器中，替你管理着一系列的类，前提是你需要将这些类交给spring容器进行管理，然后在你需要的时候，不是自己去定义，而是直接向spring容器索取，当spring容器知道你的需求之后，就会去它所管理的组件中进行查找，然后直接给你所需要的组件。Spring提供了一个一致的事务管理界面，该界面可以缩小成一个本地事务或拓展成一个全局事务。

Spring框架最大的的特点是降低组件之间的耦合度，实现软件各层之间的解耦。除此之外，还有一些其他特点，可以使容器提供众多服务如事务管理消息服务处理等等。当我们使用容器管理事务时，开发人员就不需要手工控制事务，也不需要处理复杂的事务传播；提供单例模式支持，开发人员不需要自己编写实现代码；提供了AOP技术，利用它很容易实现如权限拦截，运行期监控等功能；提供众多的辅佐类，使这些类可以加快应用的开发.如jdbcTemplate、HibernateTemplate。

* + 1. Spring MVC框架

Spring MVC是基于Spring框架开发的，在Spring框架中加入了MVC框架，以请求为驱动，围绕Servlet设计，将请求发给控制器，然后通过模型对象，分派器来展示请求结果视图。其中核心类是DispatcherServlet，它是一个Servlet，顶层是实现的Servlet接口。目的是帮助开发者简化开发。

借助于注解，Spring MVC提供了几乎是POJO的开发模式，使得控制器的开发和测试更加简单。这些控制器一般不直接处理请求，而是将其委托给Spring上下文中的其他Bean，通过Spring的依赖注入功能，这些Bean将被注入控制器中。

Spring MVC的整个流程简化如下：

1. 用户发送请求至 前端控制器DispatcherServlet。
2. 前端控制器DispatcherServlet收到请求后调用处理器映射器HandlerMapping。
3. 处理器映射器HandlerMapping根据请求的Url找到具体的处理器，生成处理器对象Handler及处理器拦截器HandlerIntercepter（如果有则生成）一并返回给前端控制器DispatcherServlet。
4. 前端控制器DispatcherServlet通过处理器适配器HandlerAdapter调用处理器Controller。
5. 执行处理器（Controller，也叫后端控制器）
6. 处理器Controller执行完后返回ModelAnView。
7. 处理器映射器HandlerAdapter将处理器Controller执行返回的结果ModelAndView返回给前端控制器DispatcherServlet。
8. 前端控制器DispatcherServlet将ModelAnView传给视图解析器ViewResolver。
9. 视图解析器ViewResolver解析后返回具体的视图View。
10. 前端控制器DispatcherServlet对视图View进行渲染视图（即：将模型数据填充至视图中）。
11. 前端控制器DispatcherServlet响应用户。
12. 系统分析与设计
    1. 健康管理系统概述

健康管理后台Web管理端采用J2EE的体系化设计模式，主要采用B/S系统架构，使用SSM框架，以PostgreSQL为关系型数据库平台，集合jQuery和jQuery EasyUI等前端框架而成的网站应用。

该系统希望为广大群体提供一个健康信息管理的平台，易操作，功能齐全。分不同角色进行不同权限功能的管理，超级管理员有最大的权限，可以管理所有事务，其他管理员（用户管理员、角色管理员，资源管理员，菜单管理员）分别根据管理员身份进行不同模块的管理，如用户管理员进行用户管理，角色管理员进行用户角色管理，资源管理员进行资源管理，菜单管理员进行菜单管理。所有管理模块都涉及一定的增删改查。

* 1. 健康管理系统需求概述
     1. 其他管理员需求分析

对于除超级管理员之外的管理员，书上的项目为了简化，只给出了模块管理这一功能，即对模块进行增删改查。

* + 1. 超级管理员需求分析

超级管理员管理所有模块，即用户管理模块，角色管理模块，菜单管理模块，资源管理模块。

管理员都有登录的需求，具体如下：

(1) 登录：根据用户名、密码和验证码进行进行登录，进入管理页面必须经过登录这一步骤，登录成功方可进入管理页面进行管理操作。

1. 修改密码：在登录页面有修改密码入口，进入修改密码页面，输入用户名，原密码，新密码，确认新密码进行修改。

此外，超级管理员的功能管理需求如下：

1. 用户管理：增删改用户，重置用户登录密码。
2. 角色管理：增删改用户角色，增加上下文菜单。
3. 菜单管理：增删改各个菜单项。
4. 资源管理：增删改各个资源项。
   1. 系统设计
      1. 数据库设计

数据库设计是整个健康管理系统中的重要设计部分，需要根据系统中所需的功能来设定数据库中具体的数据表数量及其属性，整个系统中的数据交互都需要

借助数据库实现。健康管理平台后台管理系统数据库总共设定了九个数据表，这些数据表的名称、作用及具体属性如下：

1. account 表（用户账号信息表）：

account 表记录了所有用户账号的信息，其具体的属性有account\_id（用户编号，主键，非空）、account\_name（用户名，非空）、account\_pwd（用户密码，非空）、true\_name（用户真名）、sex（性别）、birthday（出生日期）、dept（部门）、office\_phone（办公室电话）、mobile（手机号码）、email（用户邮箱）、is\_del（用户是否已被删除，1是0否）、entry\_number（用户登录次数）、create\_time（账号创立时间）、pwdb（密码加密）、city\_id（城市编号）。

1. account\_group 表（用户组信息表）：

account\_group 表记录了所有用户的分组信息，其具体的属性有account\_group\_id（用户组序号，非空，主键）、account\_id（用户编号，非空）、account\_name（用户名，非空）、group\_id（组号、非空）、create\_time（用户组创建时间）。

1. mobile\_support\_citys 表（手机号码城市对应表）：

mobile\_support\_citys 表记录了所有用户手机号码信息与其对应的城市信息，其具体的属性有city\_id（城市编号，非空，主键）、index（城市索引）、parent\_id（父城市编号）、name（城市名称）、level（城市级别）、tag（标签）、product\_id（产品编号）、cell\_station\_id（基站编号）。

1. module\_info 表（功能模块信息表）：

module\_info 表记录了所有功能模块的信息，其具体的属性有module\_id（模块编号，非空，主键）、module\_name（模块名称、非空）、module\_key（模块关键字）、options（子模块编号）、icon\_url（图标地址）、module\_type（模块分类）、module\_level（模块级别）、module\_url（模块地址）、display\_client\_version（安卓显示版本）、iphone\_display\_client\_version（iphone显示版本），run\_client\_version（安卓运行版本）、iphone\_run\_client\_version（iphone运行版本）、description（描述）、tag（标签）。

1. sys\_menu 表（菜单项信息表）：

sys\_menu 表记录了所有系统菜单项的信息，其具体的属性有id（菜单项编号，非空，主键）、pid（父菜单编号）、text（菜单项名称）、iconcls（图标）、src（菜单项路径）、seq（菜单项序号）。

1. sys\_resources 表（资源项信息表）：

sys\_resources表记录了所有系统资源项的信息，其具体的属性有id（资源项编号，非空，主键）、pid（上级编号）、text（菜资源项名称）、src（资源项路径）、seq（资源项序号）、descript（资源项描述）、onoff（是否验证）。

1. sys\_role 表（角色信息表）：

sys\_role表记录了所有角色的信息，其具体的属性有rid（角色序号，非空，主键）、id（角色编号）、pid（上级角色编号）、text（角色名称）、src（资源项路径）、seq（角色排序序号）、descript（角色描述）。

1. sys\_role\_menu 表（角色菜单项信息对应表）：

sys\_role\_menu表记录了所有角色菜单项对应的信息，其具体的属性有id（表序号，非空，主键）、role\_id（角色编号）、menu\_id（菜单项编号）。

1. sys\_role\_resources 表（角色资源项信息对应表）：

sys\_role\_resources表记录了所有角色资源项对应的信息，其具体的属性有sys\_role\_resources\_id（表序号，非空，主键）、role\_id（角色编号）、resources\_id（资源项编号）。

* + 1. 其他管理员功能设计

基于其他管理员可能的需求，进行功能设计，在书上的项目中进行了简化处理，将其他所有管理员的功能都集中于模块管理这一部分，进行增删改查。

其他管理员在使用该后台管理系统的流程图如下：

|  |
| --- |
|  |
| 图3.1 其他管理员流程图 |

首先进入系统的第一步是进入登录页面进行登录，如果其他管理员选择不登录，而选择修改密码的话，则进入修改密码页面，输出相关信息进行修改密码，修改密码成功则重新回到登录页面进行登录；如果其他管理员选择输入相关信息进行登录，则进入功能首页，即模块首页，此页面包含各个菜单项，书中只给出了模块菜单项下面模块管理部分，通过在模块管理部分进行增删改查操作来进行管理。

* + 1. 超级管理员功能设计

超级管理员和其其他管理员一样，需要先进行登录，只是他们进入的管理页面不一样，超级管理员根据需求进行所有权限功能的管理。

超级管理员在使用该后台管理系统的流程图如下：

|  |
| --- |
|  |
| 图3.2 超级管理员流程图 |

超级管理员进入系统，第一步是进入登录页面，如果超级管理员选择不登录，而选择修改密码的话，则进入修改密码页面，输出相关信息进行修改密码，修改密码成功则重新回到登录页面进行登录；如果超级管理员选择输入相关信息进行登录，则进入系统管理首页，此页面包含各个管理功能，包括用户管理、角色管理、菜单管理、资源管理，通过在各个管理部分进行增删改操作来进行管理。

1. 系统实现
   1. 数据库实现

在确定了数据库的设计方案后，可以实际开始创建数据库。数据库的创建有两种方式，一种是不借助数据库工具直接调用控制台输入 sql 语句创建数据库，一种是借助数据库工具使用可视化页面创建数据库。这里为了后续方便操作使用的是pgAdmin客户端。

我这里是直接从书上资源给出的数据库备份文件health.bak恢复得到的数据库。

具体的操作如下所示：

1. 首先创建名为health的数据库。

|  |
| --- |
|  |
| 图4.1 创建数据库 |

1. 在已创建的health数据库上点击鼠标右键，选择恢复，选择所要恢复的备份文件。

|  |
| --- |
|  |
| 图4.2 恢复数据库 |

1. 所有数据表恢复后效果如下：

|  |  |
| --- | --- |
|  | |
| 图4.3 数据表结构目录 | |
|  |  |
| 图4.4 用户账号信息表结构 | 图4.5 功能模块信息表结构 |
|  |  |
| 图4.6 用户组信息表结构 | 图4.7 菜单项信息表结构 |
|  |  |
| 图4.8 手机号码城市对应表结构 | 图4.9 角色菜单项信息对应表结构 |
|  |  |
| 图4.10 资源项信息表结构 | 图4.11 角色信息表结构 |
|  |  |
| 图4.12 角色资源项信息对应表结构 |  |

通过对这些表的数据的增删查改，整个健康管理平台的后台系统得以实现。

* 1. 其他管理员功能实现

所有的功能实现采用了 MVC 设计模式，将业务规则、数据访问及合法性校验等工作放在中间层处理。客户端不直接与数据库交互，而是通过组件与中间层建立连接，再由中间层与数据库交互。都是在 SSM 框架的帮助下完成这些功能。大体的功能实现思路按照以下步骤：

1. 完成 Model 层，由预先设计的数据库创建对应的模型。
2. 编写MyBatis的配置文件及其调用的接口，编写之后要使用的sql语句。
3. 完成 View 层，编写前端页面，为对应的按钮和鼠标点击事件设置向后端的请求操作。
4. 完成 Controller 层，处理前端传过来的请求，获取前端传过来的数据并根据需求调用实现写好的 sql 语句。

系统根据不同管理员的需求来一一实现管理功能，具体的实现图如下：

* + 1. 登录功能

|  |
| --- |
|  |
| 图4.13 登录页面实现图 |

进入登录页面，输入已经添加的用户信息，即用户名，登录密码，随机生成的验证码。进行登录；如要修改密码，则点击下方的修改密码链接，跳转到修改密码页面。

* + 1. 修改密码

|  |
| --- |
|  |
| 图4.14 修改密码页面实现 |

进入修改密码页面，输入用户名，原密码，新密码，确认密码，则可以修改密码。修改成功，则回到登录页面并显示修改密码成功，如图4.15；若修改不成功，则显示相应的错误信息。

|  |
| --- |
|  |
| 图4.14 修改密码成功 |

* + 1. 模块管理

|  |
| --- |
|  |
| 图4.15 模块首页 |

进入模块首页，首页左边是各个模块菜单项显示区域，呈树结构，右边是功能显示区域，点击左边模块菜单项下方的模块管理，则在右方出现模块管理页面，如图4，16。

|  |
| --- |
|  |
| 图4.16 模块管理 |

在右方区域对模块进行增删改查等管理操作。上方可以输入模块名称和模块key进行模糊查询；点击增加按钮出现增加模块信息的弹框，如图4.17，在谈框中输入所要增加的模块信息，点击确定按钮确定添加；选中一条记录，点击编辑按钮，出现该条信息的弹框，如图4.18，在该弹框中进行修改信息，修改完点击确定按钮；选中一条记录，点击删除按钮进行删除。

|  |
| --- |
|  |
| 图4.17 增加模块信息 |
|  |
| 图4.18 编辑模块信息 |

* 1. 超级管理员功能实现

超级管理员登录和修改密码与其他管理员一样，此处不再详细说明。超级管理员登录成功后进入系统管理首页，如图4.19。系统管理首页左边是系统菜单区域，子菜单有用户管理，菜单管理，角色管理，资源管理。

|  |
| --- |
|  |
| 图4.19 系统管理首页 |

* + 1. 用户管理

点击左边系统菜单下的子菜单——用户管理，右边区域出现用户管理页面，如图4.20，通过在用户管理页面进行增删改等管理操作对用户进行管理。

点击增加按钮，弹出增加用户弹框，如图4.21，在弹框中输入合法的用户相关信息，点击确定按钮，则添加成功。

选择一条记录，点击编辑按钮，出现该条信息的弹框，如图4.22，在该弹框中进行修改用户信息，修改完点击确定按钮。

选中一条或多条记录，点击删除按钮进行删除。

选中一条或多条记录，点击重置密码按钮，将密码重置为123456。

|  |
| --- |
|  |
| 图4.20用户管理 |
|  |
| 图4.21 增加用户 |
|  |
| 图4.22 编辑用户 |

* + 1. 菜单管理

点击左边系统菜单下的子菜单——菜单管理，右边区域出现菜单管理页面，如图4.23，通过在菜单管理页面进行增删改等管理操作对菜单进行管理。

点击增加按钮，在表格中出现新的一列，在新出现的一列中输入合法的菜单相关信息，点击保存按钮，则添加成功。

双击一条记录，即可进行编辑，编辑完点击保存按钮进行保存。

选中一条记录，点击删除按钮进行删除。

除此之外，还可对树形菜单项进行展开和折叠，默认是展开的。

|  |
| --- |
|  |
| 图4.23 菜单管理 |

* + 1. 角色管理

点击左边系统菜单下的子菜单——角色管理，右边区域出现角色管理页面，如图4.24，通过在角色管理页面进行增删改等管理操作对用户角色进行管理。

选中一个父角色记录，点击增加按钮，在该父角色下的表格中出现新的一列，在新出现的一列中输入合法的用户角色相关信息，点击保存按钮，则添加成功。

双击一条记录，即可进行编辑，编辑完点击保存按钮进行保存。

选中一条记录，点击删除按钮进行删除。

除此之外，还可对树形用户角色进行展开和折叠，默认是折叠的。

|  |
| --- |
|  |
| 图4.24 角色管理 |

* + 1. 资源管理

点击左边系统菜单下的子菜单——资源管理，右边区域出现资源管理页面，如图4.25，通过在资源管理页面进行增删改等管理操作对资源项进行管理。

点击增加按钮，在表格中出现新的一列，在新出现的一列中输入合法的资源项相关信息，点击保存按钮，则添加成功。

双击一条记录，即可进行编辑，编辑完点击保存按钮进行保存。

选中一条记录，点击删除按钮进行删除。

除此之外，还可对树形资源项进行展开和折叠，默认是折叠的。

|  |
| --- |
|  |
| 图4.25 资源管理 |

1. 总结与展望
   1. 总结

我选择的是书上最后一章的健康管理系统，因为觉得这个项目将这学期学的所有Java Web的知识基本上都有涉及到，是一个很好的回顾跟提升的机会。虽然书上给出的源码简化了某些功能，但丝毫不影响对这个项目及实现方法的理解。

我先去网上下载了该项目的源码，发现这个项目搭配的是PostgreSQL数据库，我在之前没有接触过这个数据库，所以第一步应该是安装数据库环境。我首先在PostgreSQL官网下载了最新版的PostgreSQL安装包进行安装，安装完成后，通过启动与其搭配的pgAdmin4连接数据库，进行数据库的还原时报错，发现是版本的问题，应该使用书上的版本，最新版无法对老版本的数据库备份进行还原，这才重新安装书上的对应版本的PostgreSQL，然后还原数据库成功。

接着便在eclipse上搭建项目。最开始将项目添加到eclipse中时报错，有三个依赖未导入，所以在Maven中央仓库中找到对应的包的依赖，将其添加到pom.xml文件中，添加依赖成功。接着又出现了Logger的错误，pom.xml没有添加其依赖，我去Maven中央仓库也没有找到对应版本的相关依赖，所以在网上下载了对应的JAR包，通过Build Path将包导入了项目中，至此，整个项目没有再报错。

跟着就是配置Tomcat，并启动Tomcat，运行项目。然而，在启动Tomcat这一步又出现了问题，通过在网上查阅资料，发现我没有在Tomcat中配置PostgreSQL数据库的连接池，导致整个项目没有数据源。所以成功在Tomcat的healthbutler-manager.xml文件中配置好连接池后，启动Tomcat成功，启动项目成功。

至此，我的准备工作终于完成，进入了编写文档的阶段。编写文档也不是一个简单的事情，需要细心与耐心，需要自己头脑清晰，对整个文章架构清楚。在编写文档之前，我首先根据老师给出的格式模板进行设置每个标题的样式，和整个文章的字体设置等，做好了准备工作，正式开始写的时候就可以专心写内容了。整个文档写下来还是挺流畅的，前期准备工作已经做得足够充分，整个项目已经大致在自己脑子里了。

通过这次这一个不一样的作业经验，虽然没有写任何代码，但收获的东西绝不比写代码收获的东西少，因为要完成这篇文章，前提是得对项目了解，要对项目了解就必须得去让项目跑起来，得去读代码，搞清楚实现原理，这个过程是一个吸收的过程，巩固跟提升了自身对Java Web各个知识点的理解，并将其形成了自己的一套体系。此外，这次让项目跑起来的这个过程还提高了自己解决问题的能力，主动去看问题是什么，思考这个问题出现的原因，然后试着去解决问题。每当自己成功解决一个问题的时候，成就感倍增。

* 1. 展望

通过这次作业，希望自己在将来的学习中也能像这次一样，主动的去学习，主动的去解决问题，将主动贯穿我整个学习过程中。

这次没能获得完整的这个健康管理系统是有点遗憾的，总是觉得自己有什么事尚未完成的，缺少了一部分。但是，这个系统没简化前的功能是很复杂的，数据量也是很庞大，所以我想，如果以我现在的水平去读这个项目，还是有一定的难度的。

每次的项目都让我深感自己的不足，深感自己知识的缺乏，在将来的学习中一定要努力，要有目标，要将主动执行到底。

通过这个项目，我了解了健康管理这个行业，在此之前我也没有过多关注过自己的健康管理的，现在觉得很有必要。健康管理这个行业是一个很适应现如今社会人群的一个行业，它将实时健康管理提上日程，让人们能够简便实时的管理自己的健康状况，现在正处于一个发展期，还有很大的上升空间。

这个项目对现如今的社会来说是很有价值的，996，007，各种身体亚健康的因素导致了人们现在越来越关注自身健康管理，越来越重视健康管理，而随着社会的不断发展，生活水平的不断提高，也使得健康管理成为了一个受欢迎的行业。所以这个健康管理项目虽然是多年前完成的，但仍有一定的价值导向。

此外，随着时代的发展，人们需求的变化，这个项目也有一定的改进的空间，比如可以分用户进行实时数据监测分析，追踪带病用户的病情，并给出实时提示。

希望大家都能关注自身的健康管理，毕竟只有身体好才能学习好、工作好,才能均衡地发展。

参考文献

1. 唐钧,李军. 健康社会学视角下的整体健康观和健康管理 [J]. 中国社会科学, 2019,No.284 08 131-149+208.
2. 王鹏. 基于WEB的健康管理系统的优化和设计. (Doctoral dissertation, 吉林大学).
3. 百度百科. PostgreSQL, 2021-01-25.

https://baike.baidu.com/item/PostgreSQL/530240?fr=aladdin

1. 韩冬. Web应用开发—基于Spring MVC+MyBatis+Maven. 电子工业出版社. 2018年9月.