

实验 04：医疗信息系统设计

一、实验目的

- 1、了解需求到部署的完整软件生命周期中的关键阶段与产出物。
- 2、掌握功能设计的一般方法。
- 3、理解功能层次图如何指导数据库表设计，以及清晰的功能描述为何是后端 API 开发的基础。
- 4、掌握业务系统的顶层功能设计与核心模块的详细功能描述。

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

实验基于信息技术学院教学容器化云计算平台开展。

2、软件

MySQL

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

Navicat Premium 12

五、实验任务

- 1、完成功能设计，撰写功能层次图、功能描述文档。
- 2、完成数据存储设计，设计数据库、制定文件存储标准。
- 3、完成软件结构与部署方案，确定软件结构、部署方案。
- 4、完成技术栈定义，按照开发、存储、类库、框架等分类。

六、实验内容及步骤

系统功能范围请参阅 [附件1](#)

1、功能设计

(1) 功能调研，软件所具有的功能。

调研类型	调研维度	具体问题	优先级	输出文档
功能性调研	患者管理	目前患者信息如何录入？主要来源是什么？ 需要记录哪些患者基本信息字段？哪些是必填项？ 查询患者时，最常用的搜索条件有哪些？	高	功能需求 调研表
功能性调研	健康档案	健康档案包含哪些类型的数据？（体检、检验、影像等） 医生如何查看患者历史健康数据？需要图表展示吗？	中	功能需求 调研表
业务流程调研	患者建档流程	患者建档的主要流程是什么？ 需要收集患者哪些信息？	中	业务流程 列表
用户角色	用户角色	有些用户会使用系统？ 不同角色的用户最关心的是什么？	中	用户角色 与权限矩阵表

表格 6-1-1 功能调研示例表

(2) 基于功能调研的结果制定功能层次图

父模块	子模块	英文标识	功能点	用户角色
患者管理	新增患者	patient_add	表单录入、身份证读取	医生、管理员
	患者查询	patient_query	多条件搜索、列表展示	所有用户
	信息维护	patient_maintain	信息修改、删除、日志	医生、管理员
	患者档案	patient_profile	全景视图、时间轴	所有用户
健康档案	档案查看	record_view	卡片视图、趋势图表	所有用户
	数据录入	record_input	手动录入、批量导入	医生、护士

表格 6-1-2 功能模块分解(示例)

(3) 细化功能及功能描述

要素	详细描述
功能名称	新增患者信息
功能标识	patient_add
功能描述	为系统录入新的患者基本信息，支持手动录入和身份证读卡器自动填充两种方式。
前置条件	用户已登录且具有"医生"或"管理员"角色权限。
后置条件	系统生成唯一的患者 ID，患者信息持久化存储，并记录操作日志。
输入数据	<p>1. 手动录入模式：</p> <p>患者姓名（字符串，必填）</p> <p>性别（枚举：0-男, 1-女, 2-未知，必填）</p> <p>出生日期（日期，必填）</p> <p>身份证号（字符串 18 位，必填，需校验格式）</p> <p>联系方式（字符串，必填，手机号格式校验）</p> <p>住址（字符串，可选）</p> <p>紧急联系人及电话（字符串，可选）</p> <p>2. 读卡器模式：</p> <p>身份证芯片数据（自动解析填充表单）</p>
处理逻辑	<p>1. 用户选择录入模式（手动 / 读卡器）</p> <p>2. 系统展示患者信息录入表单</p> <p>3. 用户填写 / 系统自动填充表单数据</p> <p>4. 系统进行数据校验：</p> <p>必填项完整性检查</p> <p>身份证号格式和合法性校验</p> <p>手机号格式校验</p> <p>数据唯一性检查（防止重复建档）</p> <p>5. 校验通过后保存至数据库</p> <p>6. 生成患者唯一 ID</p> <p>7. 记录"新增患者"操作日志</p>

输出结果	<p>成功情况：</p> <p> 返回成功消息：“患者信息新增成功”</p> <p> 返回生成的患者 ID</p> <p> 页面跳转至患者详情页</p> <p>失败情况：</p> <p> 返回具体的错误信息（如：“身份证号格式不正确”、“该患者已存在”等）</p>
业务规则	<ol style="list-style-type: none"> 1. 同一身份证号只能注册一次 2. 患者 ID 由系统按规则自动生成，不可修改 3. 通过读卡器录入时，自动计算出生日期和性别 4. 所有操作需记录操作人、操作时间、IP 地址

表格 6-1-3 细化功能及功能描述(示例)

2、数据存储设计

- (1) 实体设计
- (2) 属性设计
- (3) 类型设计
- (4) 表结构设计
- (5) 表结构关联关系设计
- (6) 制定文件存储标准(存储方式、根目录、命名规范等)

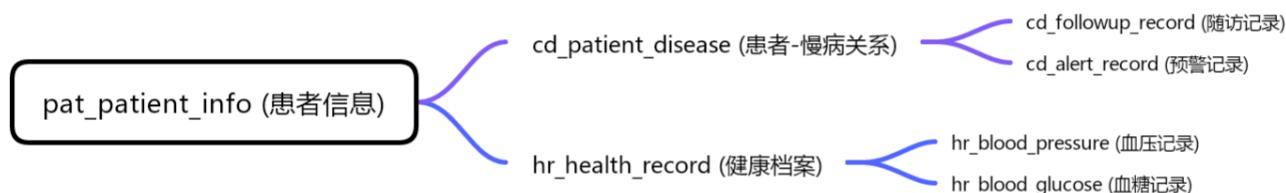


图 6-2-6-1 患者核心表关系图(表关系设计示例)

3、软件结构与部署方案

- (1) 制定软件功能层次图

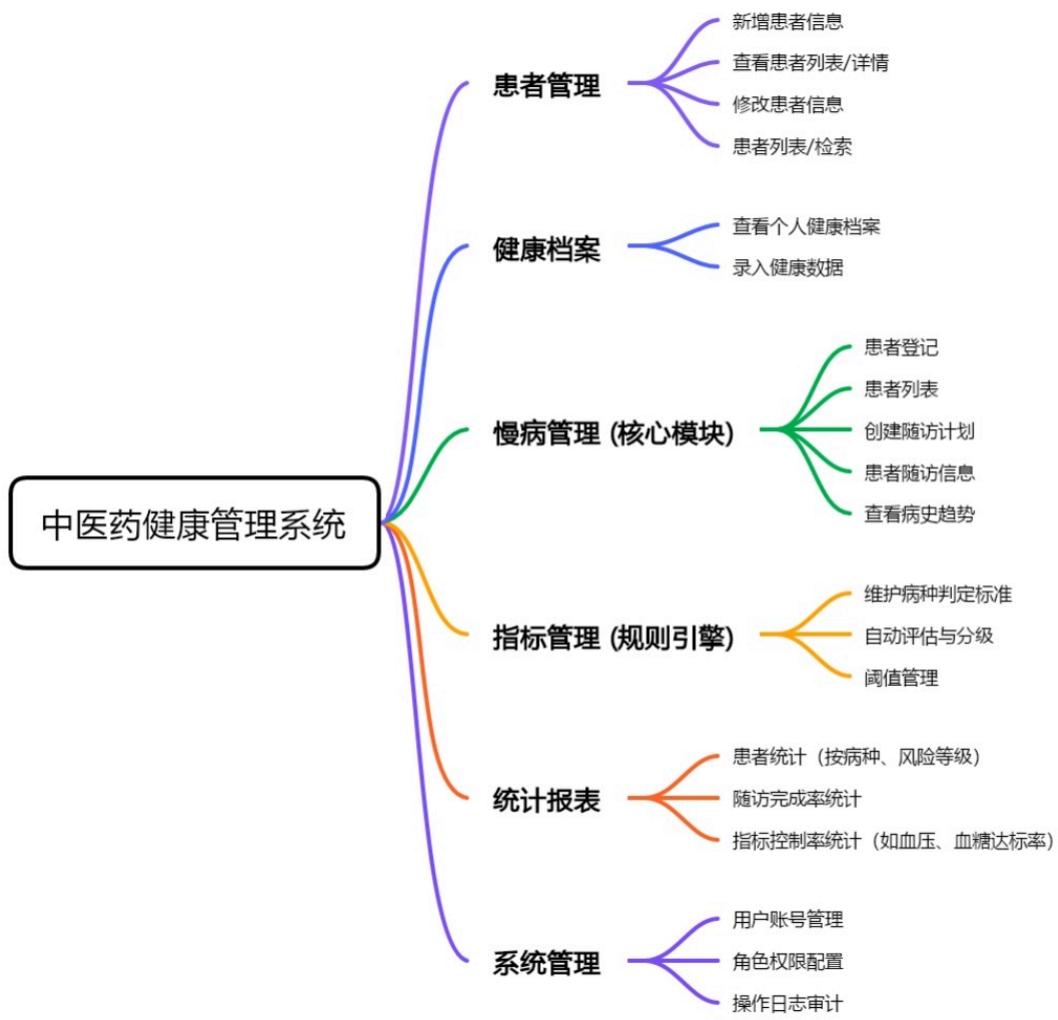


图 3-1 中医药健康管理系统功能层次图

(2) 制定软件方案(示例)

单服务器部署

使用代理服务器将来自外部的请求转发给后端 Spring Boot 应用。

将打包好的 Jar 包上传至服务器，使用 Docker 运行在 8080 端口。

在服务器上安装 MySQL，创建数据库和表结构，导入初始数据（如慢病类型、指标规则）。

在服务器上创建指定目录用于文件上传，并配置 Nginx 提供该目录的静态访问。

4、技术栈

明确系统所需的具体技术、工具和框架。

(1) 前端

框架：Vue3 + Element Plus UI

构建工具： Vite

HTTP 客户端： Axios

(2) 后端

框架： Spring Boot (Java)

安全框架： Spring Security (用于权限和登录)

数据访问： MyBatis

API 文档： Swagger

依赖管理： Maven

(3) 数据库

关系型数据库： MySQL

连接池： Druid

(4) 存储技术

主数据库： MySQL (存储所有结构化数据)

文件存储： 本地磁盘

(5) 类库

后端工具库

Lombok： 简化 JavaBean 编写。

Hutool： Java 工具类库。

Jackson： 处理 JSON 序列化 / 反序列化。

前端工具库

ECharts： 用于在统计报表模块绘制数据图表。

Day.js： 处理日期和时间。

七、实验考核

1、本课程实验考核方案

本课程实验考核采用【实验智能评】【实验随堂查】方式开展，根据不同的实验内容选择不同的考核方式。

【实验智能评】: 实验完成后提交 GitLab，通过自动化代码评审工具进行评分。

【实验随堂查】: 在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

2、本实验考核要求

本实验考核方式：实验智能评

实验 4-9 作为本课程第 2 次实验考核。

考核要求：

- (1) 学生通过 GitLab 提交实验成果：{此部分说明需要提交的内容}。
- (2) 由 GitLab 根据成果和交流情况综合评分。