

医院挂号管理系统

- 产品需求分析文档

为进一步明确产品定位与实施路径，本 PRD 以最新运营策略为基准，重点刻画医院挂号后台系统的项目背景、目标、业务流程、功能优先级、交互与数据设计，确保研发团队在迭代中保持一致方向。

1. 项目背景与愿景

1.1 背景

- 基层医院数字化程度有限，挂号依赖纸质流程，信息滞后且重复录入严重。
- 医生排班、患者信息、挂号记录分散在不同表格中，缺乏统一管理平台。
- 政策鼓励“互联网+医疗”，需要具备线上登记和后续自助服务的能力。

1.2 愿景

- 打造一个轻量、易部署的挂号管理中枢，为未来扩展（移动端、分院协同）打下基础。
- 数据驱动运营：实时掌握医生供给、患者需求、挂号量趋势。

1.3 成功衡量指标

- 平均挂号录入耗时 ≤ 60 秒。
- 重复预约冲突率 ≤ 0.5%。
- 仪表盘实时数据误差率为 0。

2. 用户、场景与痛点

2.1 用户角色

角色	职责/目标	痛点
医院管理员	维护医生、患者、挂号记录；统计报表	手工记录效率低、易出错；缺少统一入口

导诊/前台人员	快速帮患者登记、查询预约信息	排队高峰时操作慢；信息无法快速检索
院方管理层 (延展)	掌握医疗资源利用率，制定策略	缺乏数据仪表盘支持决策

2.2 典型场景

- 场景 A 新医生入职 → 管理员在系统内创建档案 → 即刻可用于挂号。
- 场景 B 患者首次来院 → 前台录入基本信息 → 直接完成挂号。
- 场景 C 患者想调整预约 → 前台在挂号管理页搜索记录 → 更新时间与状态。
- 场景 D 院长早会 → 打开仪表盘查看医生/患者/挂号趋势。

3. 业务目标与流程梳理

3.1 目标拆解

- 数据标准化：医生、患者、挂号统一存储，避免多版本。
- 流程可视：挂号从创建、就诊到取消全程留痕。
- 操作高效：界面操作步骤 ≤ 3 步完成主要任务。

3.2 核心流程

1. 登录 → 鉴权通过后进入仪表盘。
2. 点击仪表盘卡片可直接跳转到医生/患者/挂号管理页面。
3. 在管理页面中，通过搜索 + 表格浏览 + 弹窗表单完成增删改。
4. 挂号流程夹带冲突校验，保障资源不重叠。
5. 重要操作配合弹窗确认和消息提示。

4. 功能需求（按优先级）

模块	功能点	优先级	说明
登录与权限	管理员登录	Must	账号密码 + 基于 token 的会话，后续支持 JWT/角色。

	退出登录	Must	清理本地缓存，回到登录页。
仪表盘	关键指标展示	Must	显示医生数、患者数、挂号数。
	数据刷新	Must	进入页面自动拉取；后续支持定时刷新。
	卡片跳转	Should	点击指标卡跳转至对应列表页（当前版本已实现）。
医生管理	医生列表	Must	搜索条件（姓名/科室）、数据表格。
	新增/编辑医生	Must	弹窗表单，包含电话、简介等。
	删除医生	Should	需二次确认；后续可考虑软删除。
患者管理	患者列表	Must	搜索（姓名/电话），展示性别、年龄、证件号。
	新增/编辑患者	Must	录入基本信息，表单校验。
	删除患者	Should	需提示是否存在挂号记录。
挂号管理	挂号列表	Must	按患者/医生/状态查询。
	新增挂号	Must	选择患者+医生+预约时间；校验冲突。
	编辑挂号	Should	修改时间、状态、备注。

	取消挂号	Should	删除或设为“已取消”。
--	------	--------	-------------

5. 交互与体验

- 采用 Element Plus，界面风格统一；左侧导航 + 右侧内容。
- 列表页：顶部搜索区 + 右上角“新增”按钮；表格支持列宽自适应。
- 弹窗：宽度 520px 左右，输入控件保持一致性。
- 仪表盘卡片支持 hover 反馈与点击跳转，增强可达性。
- 全局消息：操作成功/失败使用 `ElMessage` 提示。

6. 数据与接口

数据库结构详见 `docs/schema.sql`，API 列表参见 `docs/README.md`。主要字段摘要如下：

- doctor**: name, department, title, phone, description。
- patient**: name, gender, age, phone, id_number, address。
- registration**: patient_id, doctor_id, appointment_time, status, notes。
- admin**: username, password, real_name。

7. 风险、限制与对策

风险/限制	表现	对策
数据准确性	手动录入易产生错误，影响后续诊疗	表单校验、必填项控制、操作日志
扩展需求	未来需支持多角色、移动端	API 遵循 REST，预留 token/角色字段
性能瓶颈	高并发提交导致数据库压力	分页查询、限制一次加载数量，必要时读写分离
安全合规	患者隐私数据需防泄露	HTTPS、权限控制、脱敏与审计

8. 里程碑规划

- 第 1 周：需求确认 & 原型设计 & 数据库建模。
- 第 2-3 周：后端接口与前端主要页面开发。
- 第 4 周：联调/测试/性能优化。
- 第 5 周：试运行 & 培训 & 上线验收。

9. 附录

- API 详见 `docs/README.md`。
- ER 图：`docs/er-diagram.html`。
- 建表 SQL: `docs/schema.sql`。