

实验室测试台 Dashboard 规划方案

1. 项目概述

本项目旨在为实验室创建一个中心化的信息管理与协作平台，以提高测试设备的利用率、规范测试流程、增强跨部门沟通效率。系统将服务于测试工程师、研发人员及其他授权员工，实现从测试申请、计划排程、状态更新到结果跟踪的全流程线上管理。

2. 系统架构设计

我们将采用基于 .NET 的现代 Web 应用架构，以确保系统的稳定性、可扩展性和安全性。

• 前端技术: Blazor Server

- **优势:**所有 UI 逻辑在服务器端执行，客户端仅处理 UI 更新，开发速度快，与 .NET 后端无缝集成，安全性高（业务逻辑不暴露到浏览器），且对客户端设备要求低。实时性强，非常适合需要即时更新状态的 Dashboard 场景。

• 后端技术: ASP.NET Core Web API

- **优势:** 虽然 Blazor Server 可以直接访问服务，但将核心业务逻辑封装在独立的 API 层中，是一种更健壮的架构。这为未来可能的扩展（如移动端 App、与其他系统集成）提供了便利。我们将使用它来处理数据访问、业务逻辑和身份验证。

• 数据库: SQL Server (或 PostgreSQL/MySQL)

- **优势:** 与 .NET 生态结合紧密，性能稳定，工具支持完善。使用 Entity Framework Core (EF Core) 作为 ORM（对象关系映射）框架，简化数据库操作。

• 身份认证与授权: ASP.NET Core Identity

- **优势:** 提供完整的用户账户管理、角色管理、登录、注册和权限控制功能，能够精确地为不同用户群 (A, B, C) 分配不同权限。

3. 用户角色与权限设计

根据您的需求，我们定义以下核心角色：

角色名称(代码)	对应用户群	核心权限
测试工程师 (TestEngineer)	A	查看所有设备和规划； 管理（创建/更新/删除）自己负责的测试计划； 更新测试状态； 调整测试优先级和内容，并提交审批； 查看所有测试申请。
申请人 (Requester)	B (研发) + C (非研发)	查看所有设备和规划（特别是可用时间）； 提交、编辑（在审批前）、撤销自己的测试申请； 查看自己申请的测试进度和状态。
审批人 (Approver)	(建议增设)	审批由测试工程师提交的计划变更； 审批（或指派）新的测试申请。 (此角色可由资深测试工程师或测试经理担任)
管理员 (Admin)	系统管理员	管理用户账户和角色； 管理设备信息（增删改查）； 系统配置。

4. 功能模块规划

模块一：公共仪表盘 (Dashboard)

- 功能:

1. 设备概览: 以卡片或列表形式展示所有测试设备，包括设备名称、型号、当前状态（空闲、使用中、维护中）、负责人等。
 2. 测试规划日历/甘特图: 以日历视图或时间轴（甘特图）形式，直观展示每个测试台在未来一段时间（日/周/月）的排程计划。不同颜色的条块表示不同的测试项目或状态。
- 用户: 对所有登录用户可见 (A, B, C)。

模块二：测试申请模块 (For B+C)

- 功能:

1. 提交测试申请: 提供一个清晰的表单，用于录入测试需求。
 - 必填字段: 项目名称/代号、申请人信息（自动填充）、期望测试时间段、样品信息（名称、数量、型号）、测试内容描述、优先级（高/中/低）。
 - 可选字段: 附件上传（如测试标准、样品图片等）。
 2. 我的申请列表: 用户可以查看自己提交的所有申请记录、当前状态（待审批、进行中、已完成、已驳回）和处理人信息。
- 用户: 申请人 (B+C)。

模块三：测试管理模块 (For A)

- 功能:

1. 测试任务看板 (Kanban): 以看板视图（如：待处理、进行中、待审批、已完成）展示分配给自己的测试任务。
2. 状态更新: 测试工程师可以直接在任务卡片上拖拽或点击，快速更新测试状态。例如，从“进行中”更新为“已完成”。
3. 计划调整与审批:

- 工程师可以打开某个测试计划的详情页。
 - 在详情页中，可以修改测试的具体步骤、调整优先级。
 - 提交修改后，系统将该计划的状态置为“待审批”，并向指定的【审批人】发送通知。
 - 【审批人】可以看到修改前后的对比，并进行“批准”或“驳回”操作。
- **用户:** 测试工程师 (A)。

模块四：审批中心 (建议)

- **功能:**
 1. **待我审批列表:** 集中展示所有需要当前用户审批的请求，包括测试计划变更、新的测试申请等。
 2. **审批操作:** 提供批准、驳回（需填写理由）的操作界面。
- **用户:** 审批人。

模块五：后台管理模块

- **功能:**
 1. **用户管理:** 创建、编辑、禁用用户，分配角色。
 2. **设备管理:** 添加新的测试设备，维护设备信息。
- **用户:** 管理员。

5. 产能评估 (预估)

阶段	主要内容	预估工时 (人/天)
第一阶段：基础架构与核心功能 (MVP)	1. 项目搭建: .NET 项目结构、数据库设计、EF Core 配置。 2. 用户与权限: 实现用户登录、角色管理 (Admin, TestEngineer, Requester)。 3. 公共部分: 设备信息的展示、基础的规划日历视图。 4. 核心流程: 实现测试申请的提交与查看功能。	15 - 20.
第二阶段：测试管理与审批流	1. 测试管理: 开发测试工程师的任务看板、测试状态更新功能。 2. 审批流程: 开发计划调整功能，并实现提交、审批、驳回的完整工作流。 3. 通知系统: 实现简单的站内消息或邮件通知 (如: 新申请、审批结果)。	20 - 25
第三阶段：优化、测试与部署	1. 界面优化: 根据反馈优化 UI/UX，增强图表的可视化效果。 2. 完善功能: 如数据筛选、搜索、导出等。 3. 单元测试与集成测试。 4. 系统部署、用户培训和文档编写。	10 - 15
总计		45 - 60 人/天