

# 测试结果报告

## 1. 测试用例设计

本系统的测试用例设计遵循分层测试策略，结合多种黑盒测试技术，确保覆盖系统的功能逻辑、边界条件及异常处理。

### 1.1 设计方法

- 等价类划分:** 将输入数据划分为有效等价类（如合法的邮箱格式、正数的卡路里数值）和无效等价类（如错误的邮箱格式、负数或非数字输入），以减少测试用例数量同时保证覆盖率。
- 边界值分析:** 针对输入域的边界进行测试，例如密码长度的最小值、日期的最大值（如未来日期）、数值的零值等，因为错误往往发生在边界处。
- 错误推测:** 基于经验推测可能出现的错误，例如重复注册已存在的邮箱、在未登录状态下直接访问受保护页面、网络中断时的提交操作等。
- 场景分析:** 模拟用户真实的操作流程（User Journey），例如“用户注册 -> 登录 -> 记录饮食 -> 查看仪表盘 -> 退出登录”的完整闭环。
- 判定表驱动:** 针对逻辑组合复杂的场景（如登录认证），列出所有条件的组合（账号存在/不存在、密码正确/错误）及其对应的预期动作。

### 1.2 测试用例表

下表列出了基于上述方法设计的核心测试用例：

用例ID	模块	测试场景	前置条件	测试步骤	预期结果	实际结果
TC-AUTH-001	认证	正常登录	用户已注册	1. 进入登录页 2. 输入正确邮箱和密码 3. 点击登录	跳转至首页，显示用户昵称	<input checked="" type="checkbox"/> Pass
TC-AUTH-002	认证	登录失败-密码错误	用户已注册	1. 输入正确邮箱 2. 输入错误密码 3. 点击登录	提示“用户名或密码错误”，停留在登录页	<input checked="" type="checkbox"/> Pass
TC-AUTH-003	认证	登录失败-账号不存在	无	1. 输入未注册邮箱 2. 输入任意密码 3. 点击登录	提示“用户名或密码错误”	<input checked="" type="checkbox"/> Pass
TC-AUTH-004	认证	注册-邮箱格式校验	无	1. 邮箱输入 "invalid-email" 2. 填写其他必填项 3. 点击注册	提示邮箱格式不正确，注册按钮不可用或提交失败	<input checked="" type="checkbox"/> Pass
TC-AUTH-005	认证	注册-重复邮箱	邮箱已存在	1. 输入已注册的邮箱 2. 填写有效信息 3. 点击注册	提示“该邮箱已被注册”	<input checked="" type="checkbox"/> Pass
TC-DIET-001	饮食	添加饮食记录-正常	已登录	1. 进入饮食页面 2. 选择早餐，输入"燕麦", 卡路里"300" 3. 提交	列表显示该条记录，提示保存成功	<input checked="" type="checkbox"/> Pass

用例ID	模块	测试场景	前置条件	测试步骤	预期结果	实际结果
TC-DIET-002	饮食	添加饮食记录-数值边界	已登录	1. 卡路里输入 "0" 或 "-100" 2. 提交	提示卡路里必须为正数	<input checked="" type="checkbox"/> Pass
TC-BODY-001	身体	录入体重-未来日期	已登录	1. 选择明天的日期 2. 输入体重 3. 提交	系统应允许或提示确认 (视业务规则而定, 当前允许)	<input checked="" type="checkbox"/> Pass
TC-CHAT-001	AI助手	智能建议获取	已登录	1. 进入聊天页面 2. 发送 "我今天吃了汉堡, 该怎么运动?"	AI 能够读取饮食记录并给出针对性运动建议	<input checked="" type="checkbox"/> Pass
TC-SEC-001	安全	未授权访问	未登录	1. 直接访问 /dashboard URL	自动重定向至 /login 页面	<input checked="" type="checkbox"/> Pass

## 2. 测试计划制定

### 2.1 测试目标

验证 "健康生活管理系统" (Health Life Management System) v1.0 版本的功能完整性、稳定性和用户体验，确保满足发布标准。

### 2.2 测试范围

- 前端: Vue 3 + Nuxt 应用, 重点测试路由导航、组件渲染、表单交互。
- 后端: Spring Boot 应用, 重点测试 RESTful API 接口的正确性和异常处理。

### 2.3 测试环境与工具

- 操作系统: Windows
- 前端测试框架:
  - Vitest 4.0.15: 单元测试框架
  - @vue/test-utils 2.4.6: Vue 组件测试工具
  - @vitest/coverage-v8 4.0.15: 代码覆盖率统计
  - Playwright 1.57.0: E2E 端到端测试
  - happy-dom 20.0.11: DOM 模拟环境
- API 测试工具:
  - Newman 6.2.1: Postman 命令行运行器
  - Postman Collection: API 自动化测试集合
- 后端测试: JUnit (Gradle 集成)
- 浏览器: Chrome

### 2.4 测试命令

测试类型	命令	说明
单元测试	pnpm test	运行 Vitest 监听模式
单元测试(单次)	pnpm test:unit	运行所有单元测试
覆盖率测试	pnpm test:coverage	生成代码覆盖率报告
E2E 测试	pnpm test:e2e	运行 Playwright E2E 测试

测试类型	命令	说明
E2E 测试(UI)	<code>pnpm test:e2e:ui</code>	使用 Playwright UI 模式
API 测试	<code>pnpm test:api</code>	运行 Newman API 自动化测试
后端测试	<code>./gradlew test</code>	运行后端 JUnit 测试

## 2.5 测试策略

### 2.5.1 分层测试策略

本项目采用经典的测试金字塔模型，从底层到顶层依次为：



测试层级	工具	覆盖目标	执行频率
单元测试	Vitest, JUnit	工具函数、业务逻辑、组件渲染	每次代码提交
API 测试	Newman/Postman	RESTful 接口、数据校验、认证授权	每次构建
E2E 测试	Playwright	用户完整流程、跨页面交互	每日/发布前

### 2.5.2 测试类型分布

- **功能测试：**验证系统各模块功能是否符合需求规格
    - 认证模块：注册、登录、密码重置、Token 管理
    - 用户模块：资料管理、头像上传
    - 健康数据模块：身体指标、饮食、运动、睡眠的 CRUD 操作
    - AI 聊天模块：SSE 流式响应、记忆管理
  - **边界测试：**验证输入边界条件的处理
    - 无效体重值（负数、零值）
    - 无效睡眠时间顺序（起床时间早于入睡时间）
    - 空值和超长字符串处理
  - **异常测试：**验证错误处理机制
    - 错误密码登录
    - 重复邮箱注册
    - 未授权访问受保护资源
  - **集成测试：**验证模块间协作
    - 登录后自动获取用户资料
    - 创建记录后立即验证持久化
    - 删除记录后验证不可访问

### 2.5.3 数据驱动测试

API 测试采用 Postman 环境变量实现数据驱动：

- 测试数据通过 local.postman\_environment.json 配置

- 支持动态变量传递（如 {{userId}}、{{token}}）
- 测试间状态自动同步（登录获取 Token → 后续请求自动携带）

## 2.5.4 持续集成策略

- Pre-commit:** 运行单元测试和代码检查
- PR 构建:** 运行完整测试套件
- 主分支合并:** 触发 E2E 测试和 API 测试
- 发布前:** 执行全量回归测试

## 2.6 执行策略

- 自动化执行:**
  - 前端 E2E 测试配置为并行运行 (fullyParallel: true)，以提高执行效率。
  - 后端测试在构建阶段 (gradle build) 自动触发。
  - API 测试通过 Newman 执行 Postman 集合，支持 CI/CD 集成。
- 失败重试:** CI 环境下配置了重试机制 (retries: 2)，以应对网络波动导致的 flaky tests。
- 报告生成:** 生成 HTML 格式的测试报告，便于问题追踪。

# 3. 测试用例执行

## 执行记录摘要

### 前端 E2E 测试 (Playwright)

执行命令: pnpm exec playwright test

测试套件	测试用例	状态	备注
首页功能测试 (home.spec.ts)	页面应该有正确的标题	✓ Pass	验证 Title 包含 "健康生活管理系统"
	应该显示主标题	✓ Pass	H1 标签可见性
	应该显示描述文字	✓ Pass	验证 Slogan 文本
	应该显示登录和注册按钮	✓ Pass	验证导航链接
	应该显示核心功能列表	✓ Pass	验证 7 个核心功能模块标题
登录页面测试 (login.spec.ts)	应该显示登录表单	✓ Pass	验证输入框和按钮存在
	应该能够输入邮箱和密码	✓ Pass	验证表单填充功能
	应该能够切换到注册表单	✓ Pass	验证 URL 参数 ?mode=register 逻辑
	点击注册按钮应该切换到注册表单	✓ Pass	验证页面内交互跳转

### 后端 API 测试 (JUnit)

执行命令: ./gradlew test

测试类	测试方法	状态	备注
HealthLifeApplicationTests	contextLoads	✓ Pass	Spring 上下文加载正常
UserControllerTest	loginShouldReturnToken...	✓ Pass	登录成功返回 Token
	registerShouldReturnError...	✓ Pass	重复注册返回错误码

**API 自动化测试 (Newman/Postman)**

执行命令: pnpm test:api

测试模块	测试用例	状态	备注
<b>Auth</b>	Register	<input checked="" type="checkbox"/> Pass	用户注册成功或已存在
	Login	<input checked="" type="checkbox"/> Pass	登录成功返回 Token
	Login (Wrong Password)	<input checked="" type="checkbox"/> Pass	错误密码被拒绝
	Password Reset (New Password)	<input checked="" type="checkbox"/> Pass	密码重置成功
	Login (Using Reset Password)	<input checked="" type="checkbox"/> Pass	使用重置后密码登录
	Password Reset (Restore Password)	<input checked="" type="checkbox"/> Pass	恢复原密码
	Login (After Restore)	<input checked="" type="checkbox"/> Pass	恢复后登录成功
<b>User</b>	Get Profile	<input checked="" type="checkbox"/> Pass	获取用户资料
	Update Profile	<input checked="" type="checkbox"/> Pass	更新用户资料
	Get User Detail	<input checked="" type="checkbox"/> Pass	获取用户详情
	Update User Detail	<input checked="" type="checkbox"/> Pass	管理员更新用户
	Upload Avatar	<input checked="" type="checkbox"/> Pass	头像上传 (跳过无文件)
	Get Avatar	<input checked="" type="checkbox"/> Pass	获取头像
	Head Avatar	<input checked="" type="checkbox"/> Pass	HEAD 请求检查
<b>Body Metrics</b>	List Body Metrics	<input checked="" type="checkbox"/> Pass	列表查询
	Create Body Metric	<input checked="" type="checkbox"/> Pass	新增身体数据
	Get Body Metric Detail	<input checked="" type="checkbox"/> Pass	获取详情
	Create Body Metric (Invalid Weight)	<input checked="" type="checkbox"/> Pass	无效体重被拒绝
	Update Body Metric	<input checked="" type="checkbox"/> Pass	更新身体数据
	Delete Body Metric	<input checked="" type="checkbox"/> Pass	删除并验证
	Export Body Metrics to CSV	<input checked="" type="checkbox"/> Pass	导出 CSV
<b>Diet Records</b>	List Diet Records	<input checked="" type="checkbox"/> Pass	列表查询
	Create Diet Record	<input checked="" type="checkbox"/> Pass	新增饮食记录
	Get Diet Record Detail	<input checked="" type="checkbox"/> Pass	获取详情
	Update Diet Record	<input checked="" type="checkbox"/> Pass	更新饮食记录
	Delete Diet Record	<input checked="" type="checkbox"/> Pass	删除并验证
	Export Diet Records to CSV	<input checked="" type="checkbox"/> Pass	导出 CSV
<b>Exercise Records</b>	List Exercise Records	<input checked="" type="checkbox"/> Pass	列表查询
	Create Exercise Record	<input checked="" type="checkbox"/> Pass	新增运动记录
	Get Exercise Record Detail	<input checked="" type="checkbox"/> Pass	获取详情
	Update Exercise Record	<input checked="" type="checkbox"/> Pass	更新运动记录
	Delete Exercise Record	<input checked="" type="checkbox"/> Pass	删除并验证
	Export Exercise Records to CSV	<input checked="" type="checkbox"/> Pass	导出 CSV

测试模块	测试用例	状态	备注
<b>Sleep Records</b>	List Sleep Records	<input checked="" type="checkbox"/> Pass	列表查询
	Create Sleep Record	<input checked="" type="checkbox"/> Pass	新增睡眠记录
	Get Sleep Record Detail	<input checked="" type="checkbox"/> Pass	获取详情
	Create Sleep Record (Invalid Order)	<input checked="" type="checkbox"/> Pass	无效时间顺序被拒绝
	Update Sleep Record	<input checked="" type="checkbox"/> Pass	更新睡眠记录
	Delete Sleep Record	<input checked="" type="checkbox"/> Pass	删除并验证
	Export Sleep Records to CSV	<input checked="" type="checkbox"/> Pass	导出 CSV
<b>Chat</b>	Delete Chat Memory	<input checked="" type="checkbox"/> Pass	清除聊天记忆
	Chat Stream (SSE)	<input checked="" type="checkbox"/> Pass	SSE 流式响应

## 4. 测试结论

经过全面的自动化测试覆盖，系统核心功能（首页展示、用户认证流程）表现稳定。

- **前端：**所有关键 UI 组件渲染正常，交互逻辑符合预期，页面跳转流畅。
- **后端：**API 接口响应正确，能够正确处理业务逻辑和异常情况。
- **API 测试：**通过 Newman 执行的 Postman 自动化测试覆盖了 7 大模块，共计 42 个测试用例，全部通过。包括：
  - 认证模块（注册、登录、密码重置）
  - 用户管理（资料查看/更新、头像管理）
  - 身体数据 CRUD 及导出
  - 饮食记录 CRUD 及导出
  - 运动记录 CRUD 及导出
  - 睡眠记录 CRUD 及导出
  - AI 聊天（记忆清除、SSE 流式响应）

**结论：系统符合发布要求，准予发布。**