Implementation Plan:Read AI

Overview

本实施计划将Read AI 游戏化阅读工具分解为可执行的开发任务。**实施策略：先开发前端界面确认视觉效果符合预期，然后再实现后端功能。**这种方法可以快速验证用户体验和交互设计，确保最终产品符合预期。

技术栈：TypeScript + Node.js (后端), React + Next.js (前端), PostgreSQL (数据库), Redis (缓存)

Phase 1: 前端界面开发（使用 Mock 数据）

任务说明

第一阶段专注于前端界面开发，使用 Mock 数据模拟后端响应。这样可以快速迭代 UI/UX 设计，确认视觉效果和交互流程符合预期。

Tasks

- [ ] 1. 项目初始化和前端基础架构
  - 创建 Next.js 项目结构
  - 配置 TypeScript、ESLint、Prettier
  - 配置 TailwindCSS 和 Framer Motion（动画库）
  - 设置基础路由和布局
  - 创建设计系统（颜色、字体、间距、组件库）
  - *Requirements: 所有需求的基础*
- [ ] 2. 创建 Mock 数据和类型定义
  - 定义所有 TypeScript 类型（Article, Quiz, Card, UserStats 等）
  - 创建 Mock 数据生成器
  - 生成示例文章、Quiz、卡片数据
  - *Requirements: 所有需求*
- [ ] 3. 实现文章收藏列表界面
  - [ ] 3.1 创建文章列表页面
    - 显示收藏的文章列表
    - 显示文章标题、来源、收藏时间
    - 显示文章状态标记（待生成 Quiz、已完成等）
    - 实现添加文章的输入框
    - *Requirements: 9.1, 9.3, 9.5*
  - [ ]* 3.2 编写文章列表的单元测试
    - 测试列表渲染
    - 测试状态标记显示
    - *Requirements: 9.3, 9.5*

- [ ] 4. 实现 Quiz 交互界面
  - [ ] 4.1 创建 Quiz 卡片组件
    - 设计吸引人的卡片样式
    - 实现问题和选项展示
    - 实现选项选择交互
    - 添加选择后的视觉反馈
    - *Requirements: 1.3, 1.4*
  - [ ] 4.2 实现答案提交和解锁动画
    - 创建答案提交流程
    - 设计解锁内容的动画效果
    - 显示正确/错误答案的反馈
    - *Requirements: 1.5*
  - [ ]* 4.3 编写 Quiz 卡片的单元测试
    - 测试卡片渲染
    - 测试选项交互
    - 测试动画触发
    - *Requirements: 1.3, 1.4, 1.5*
- [ ] 5. 实现文章阅读页面
  - [ ] 5.1 创建文章详情页面布局
    - 设计页面整体布局
    - 实现首次访问隐藏正文的逻辑
    - 显示 Quiz 卡片列表
    - 显示阅读进度指示器
    - *Requirements: 1.3, 2.3*
  - [ ] 5.2 实现内容片段展示
    - 创建内容片段组件
    - 实现解锁后的片段展示
    - 显示片段在原文中的位置
    - 添加片段展开/收起动画
    - *Requirements: 2.2*
  - [ ] 5.3 实现完整文章查看
    - 完成所有 Quiz 后显示完整文章按钮
    - 实现完整文章阅读视图
    - *Requirements: 2.3*
  - [ ]* 5.4 编写阅读页面的单元测试
    - 测试页面渲染
    - 测试进度显示

        - 测试内容解锁逻辑
        - *Requirements: 1.3, 2.2, 2.3*
- [ ] 6. Checkpoint - 验证核心阅读体验 UI
    - 确认 Quiz 交互流程符合预期
    - 确认视觉设计和动画效果
    - 如有调整需求请询问用户
- [ ] 7. 实现 Keep/Throw Away 决策界面
    - [ ] 7.1 创建决策按钮组件
        - 设计 Keep 和 Throw Away 按钮样式
        - 实现按钮点击交互
        - 添加决策后的反馈动画
        - *Requirements: 3.1*
    - [ ] 7.2 实现个人思考输入界面
        - 创建思考输入框（可选显示）
        - 支持 Markdown 格式
        - 实现快速输入体验
        - *Requirements: 3.4, 3.5*
    - [ ]* 7.3 编写决策界面的单元测试
        - 测试按钮渲染和交互
        - 测试输入框显示逻辑
        - *Requirements: 3.1, 3.4*
- [ ] 8. 实现知识卡片管理界面
    - [ ] 8.1 创建卡片列表页面
        - 实现类似 Flomo 的卡片布局
        - 实现瀑布流或网格布局
        - 显示原文片段和个人思考
        - 显示卡片元数据（来源、时间）
        - *Requirements: 4.5, 4.6*
    - [ ] 8.2 实现卡片编辑功能
        - 创建卡片编辑模态框
        - 支持编辑个人思考
        - 支持添加标签
        - 实现保存和取消操作
        - *Requirements: 4.4*
    - [ ] 8.3 实现卡片搜索和过滤
        - 创建搜索输入框
        - 实现标签过滤

        - 实现时间范围过滤
        - *Requirements: 4.5*
    - [ ]* 8.4 编写卡片界面的单元测试
        - 测试卡片列表渲染
        - 测试编辑功能
        - 测试搜索和过滤
        - *Requirements: 4.4, 4.5, 4.6*
- [ ] 9. Checkpoint - 验证知识管理 UI
    - 确认卡片界面符合 Flomo 风格
    - 确认编辑和管理体验流畅
    - 如有调整需求请询问用户
- [ ] 10. 实现分享卡片生成界面
    - [ ] 10.1 创建分享卡片预览组件
        - 设计多个高颜值卡片模板
        - 实现卡片预览渲染
        - 显示原文片段、思考、二维码位置
        - *Requirements: 7.1, 7.2, 7.3*
    - [ ] 10.2 实现模板选择界面
        - 创建模板选择器
        - 显示模板缩略图
        - 实现模板切换
        - *Requirements: 7.3*
    - [ ] 10.3 实现样式自定义界面
        - 创建颜色选择器
        - 创建字体选择器
        - 实现实时预览
        - *Requirements: 7.5*
    - [ ] 10.4 实现下载和分享功能
        - 添加下载按钮
        - 添加分享到社交媒体按钮
        - 实现复制链接功能
        - *Requirements: 7.1*
    - [ ]* 10.5 编写分享界面的单元测试
        - 测试模板选择
        - 测试样式自定义
        - 测试预览渲染
        - *Requirements: 7.1, 7.3, 7.5*

- [ ] 11. 实现播客播放界面
  - [ ] 11.1 创建播客播放器组件
    - 设计播放器 UI
    - 实现播放/暂停按钮
    - 实现进度条
    - 显示播放时间
    - *Requirements: 6.3*
  - [ ] 11.2 实现播客生成状态显示
    - 显示生成进度（生成脚本、合成音频）
    - 显示生成失败的错误信息
    - 提供重新生成按钮
    - *Requirements: 6.1*
  - [ ] 11.3 实现后台播放支持
    - 实现最小化播放器
    - 支持页面切换时继续播放
    - *Requirements: 6.4*
  - [ ]* 11.4 编写播放器的单元测试
    - 测试播放控制
    - 测试进度显示
    - 测试后台播放
    - *Requirements: 6.3, 6.4*
- [ ] 12. 实现游戏化界面
  - [ ] 12.1 创建用户统计仪表板
    - 显示阅读连续天数（大字体突出显示）
    - 显示已读文章数、Quiz 完成数、卡片数
    - 使用图表可视化统计数据
    - *Requirements: 8.3, 8.5*
  - [ ] 12.2 创建成就徽章展示
    - 设计徽章图标
    - 显示已解锁和未解锁徽章
    - 实现徽章解锁动画
    - *Requirements: 8.4*
  - [ ] 12.3 创建每日推荐界面
    - 显示每日推荐文章卡片
    - 显示完成任务的正向反馈
    - 实现庆祝动画
    - *Requirements: 8.1, 8.2*

- [ ]* 12.4 编写游戏化界面的单元测试
  - 测试统计数据显示
  - 测试徽章显示
  - 测试推荐界面
  - *Requirements: 8.1, 8.2, 8.3, 8.4, 8.5*

- [ ] 13. 实现反常规解读展示
  - [ ] 13.1 创建解读展示组件
    - 设计解读卡片样式
    - 实现解读内容展示
    - 添加"添加到卡片"按钮
    - *Requirements: 5.3, 5.4*
  - [ ]* 13.2 编写解读展示的单元测试
    - 测试解读渲染
    - 测试添加到卡片功能
    - *Requirements: 5.3, 5.4*

- [ ] 14. 前端界面完善和优化
  - [ ] 14.1 实现响应式设计
    - 适配移动端布局
    - 适配平板布局
    - 测试各种屏幕尺寸
    - *Requirements: 所有需求*
  - [ ] 14.2 优化动画和交互
    - 调整动画时长和缓动函数
    - 优化页面切换过渡
    - 添加加载状态动画
    - *Requirements: 所有需求*
  - [ ] 14.3 实现暗色模式
    - 设计暗色主题配色
    - 实现主题切换功能
    - 保存用户主题偏好
    - *Requirements: 所有需求*

- [ ] 15. Final Checkpoint - 前端界面验收
  - 完整演示所有界面和交互流程
  - 确认视觉设计符合预期
  - 确认用户体验流畅
  - 收集用户反馈并调整
  - **前端验收通过后，开始后端开发**

Phase 2: 后端功能开发

任务说明

第二阶段在前端界面验收通过后，开始实现后端功能，将 Mock 数据替换为真实的 API 调用。

- [ ] 16. 后端项目初始化
  - 创建 Node.js + Express 项目
  - 配置 TypeScript
  - 设置 PostgreSQL 和 Redis 连接
  - 配置测试框架（Jest + fast-check）
  - 创建基础的 API 网关和认证服务
  - *Requirements: 所有需求的基础*
- [ ] 17. 实现内容解析服务
  - [ ] 17.1 创建 Content Parser Service
    - 定义服务接口和数据模型
    - 集成 Mozilla Readability
    - 实现文章 URL 提取和解析
    - 添加微信公众号、知乎、Medium 特殊处理
    - *Requirements: 10.1, 10.2, 10.4, 10.5*
  - [ ]* 17.2 编写内容解析的属性测试
    - **Property 25: 文章解析过滤广告内容**
    - **Property 27: 解析保留文章结构**
    - **Property 28: 解析提取元数据**
    - **Validates: Requirements 10.1, 10.4, 10.5**
  - [ ] 17.3 实现解析错误处理
    - 处理 404、网络错误、超时
    - 实现回退到手动输入
    - *Requirements: 10.3*
  - [ ]* 17.4 编写错误处理的属性测试
    - **Property 26: 解析失败提供错误信息**
    - **Validates: Requirements 10.3**
- [ ] 18. 实现文章收藏管理 API
  - [ ] 18.1 创建 Article 数据模型和 API
    - 定义数据库 schema
    - 创建迁移脚本
    - 实现 CRUD API 端点
    - *Requirements: 9.1, 9.4*

- [ ]* 18.2 编写收藏管理的属性测试
  - **Property 20: 添加文章到收藏列表**
  - **Property 21: 多种方式收藏文章**
  - **Property 22: 收藏列表显示必需信息**
  - **Property 23: 删除文章从列表移除**
  - **Property 24: 文章状态正确标记**
  - **Validates: Requirements 9.1-9.5**

- [ ] 19. 实现 Quiz 生成服务
  - [ ] 19.1 创建 Quiz Generator Service
    - 配置 LLM 服务（OpenAI/Claude）
    - 实现文章分析和 Quiz 生成
    - 实现 Quiz 质量验证
    - *Requirements: 1.1, 1.2*
  - [ ]* 19.2 编写 Quiz 生成的属性测试
    - **Property 1: Quiz 生成数量约束**
    - **Property 2: Quiz 包含有效选项**
    - **Validates: Requirements 1.1, 1.4**
  - [ ] 19.3 实现反常规解读生成
    - 设计解读生成 prompt
    - 实现生成逻辑
    - *Requirements: 5.1*
  - [ ]* 19.4 编写解读生成的属性测试
    - **Property 13: 完成阅读生成反常规解读**
    - **Validates: Requirements 5.1**
  - [ ] 19.5 实现错误处理和异步队列
    - 实现重试机制
    - 使用 Bull 队列异步处理
    - *Requirements: 1.1*

- [ ] 20. 实现内容解锁服务
  - [ ] 20.1 创建 Content Unlock Service
    - 定义数据模型
    - 配置 Redis 缓存
    - 实现解锁逻辑
    - *Requirements: 2.1, 2.5*
  - [ ]* 20.2 编写内容解锁的属性测试
    - **Property 3: 回答 Quiz 解锁对应片段**
    - **Property 4: 解锁片段包含位置信息**

        - **Property 5: 完成所有 Quiz 解锁完整文章**
        - **Property 6: 解锁进度持久化（Round-trip）**
        - **Validates: Requirements 1.5, 2.1-2.5**
- [ ] 21. Checkpoint - 验证核心后端功能
    - 确保所有测试通过
    - 集成前端和后端
    - 测试完整的阅读流程
    - 如有问题请询问用户
- [ ] 22. 实现知识卡片管理 API
    - [ ] 22.1 创建 Card Management Service
        - 定义数据模型和 API
        - 实现 Keep/Throw Away 逻辑
        - 实现卡片 CRUD 操作
        - *Requirements: 3.2, 3.3, 4.1, 4.4*
    - [ ]* 22.2 编写卡片管理的属性测试
        - **Property 7: Keep 操作创建知识卡片**
        - **Property 8: Throw Away 操作隐藏内容**
        - **Property 9: 创建卡片时思考字段可选**
        - **Property 10: 卡片思考内容关联存储**
        - **Property 11: 卡片思考可编辑**
        - **Property 12: 卡片按时间顺序排列**
        - **Validates: Requirements 3.2, 3.3, 3.5, 4.1-4.4, 4.6**
- [ ] 23. 实现 AI 播客生成服务
    - [ ] 23.1 创建 Podcast Generator Service
        - 配置 TTS 服务
        - 实现播客脚本生成
        - 实现音频合成
        - *Requirements: 6.1*
    - [ ]* 23.2 编写播客生成的属性测试
        - **Property 14: 播客生成返回音频文件**
        - **Validates: Requirements 6.1**
    - [ ] 23.3 实现异步处理和错误处理
        - 使用队列处理播客生成
        - 实现状态追踪
        - *Requirements: 6.1*
- [ ] 24. 实现分享卡片生成服务
    - [ ] 24.1 创建 Share Card Service

        - 集成二维码生成 API
        - 实现卡片图片生成（Canvas/Puppeteer）
        - 实现模板系统
        - *Requirements: 7.1, 7.2, 7.3*
    - [ ]* 24.2 编写分享卡片的属性测试
        - **Property 15: 分享卡片包含二维码**
        - **Property 16: 自定义样式应用到分享卡片**
        - **Validates: Requirements 7.2, 7.5, 7.6**
- [ ] 25. 实现用户管理和游戏化系统
    - [ ] 25.1 创建 User Management Service
        - 定义数据模型
        - 实现统计数据追踪
        - 实现阅读连续天数追踪
        - *Requirements: 8.3, 8.5*
    - [ ]* 25.2 编写统计追踪的属性测试
        - **Property 18: 完成任务更新统计数据**
        - **Validates: Requirements 8.3**
    - [ ] 25.3 实现成就和徽章系统
        - 定义徽章类型和解锁条件
        - 实现徽章授予逻辑
        - *Requirements: 8.4*
    - [ ]* 25.4 编写成就系统的属性测试
        - **Property 19: 连续阅读授予徽章**
        - **Validates: Requirements 8.4**
    - [ ] 25.5 实现每日推荐系统
        - 实现推荐算法
        - 实现正向反馈逻辑
        - *Requirements: 8.1, 8.2*
    - [ ]* 25.6 编写推荐系统的属性测试
        - **Property 17: 每日推荐生成**
        - **Validates: Requirements 8.1**
- [ ] 26. 前后端集成
    - [ ] 26.1 替换前端 Mock 数据为真实 API
        - 更新所有 API 调用
        - 实现错误处理
        - 实现加载状态
        - *Requirements: 所有需求*

- [ ] 26.2 实现 API 认证和授权
  - 配置 JWT 认证
  - 实现用户登录/注册
  - 保护 API 端点
  - *Requirements: 所有需求*
- [ ] 27. 实现集成测试
  - [ ]* 27.1 编写 Quiz 流程集成测试
    - 测试完整的 Quiz 流程
    - 验证数据一致性
    - *Requirements: 1.1-2.5*
  - [ ]* 27.2 编写卡片流程集成测试
    - 测试完整的卡片流程
    - 验证数据完整性
    - *Requirements: 3.1-4.6*
  - [ ]* 27.3 编写分享流程集成测试
    - 测试完整的分享流程
    - 验证所有元素
    - *Requirements: 7.1-7.6*
- [ ] 28. 性能优化和部署准备
  - [ ] 28.1 实现缓存策略
    - 配置 Redis 缓存
    - 实现 API 响应缓存
    - 配置 CDN
    - *Requirements: 所有需求*
  - [ ] 28.2 实现监控和日志
    - 配置结构化日志
    - 实现关键指标追踪
    - 配置错误告警
    - *Requirements: 所有需求*
  - [ ] 28.3 准备部署配置
    - 创建 Docker 配置
    - 配置环境变量
    - 准备数据库迁移
    - *Requirements: 所有需求*
- [ ] 29. Final Checkpoint - 完整系统验证
  - 运行所有测试（单元、属性、集成）
  - 验证所有 28 个正确性属性

    - 确保测试覆盖率达标（> 80%）
    - 进行端到端测试
    - 如有问题请询问用户

Notes

- **Phase 1（任务 1-15）**：专注于前端界面开发，使用 Mock 数据
- **Phase 2（任务 16-29）**：在前端验收通过后，开发后端功能
- 标记为 `*` 的任务是可选的测试任务，可以跳过以加快开发
- 每个任务都引用了相应的需求编号，确保可追溯性
- Checkpoint 任务用于在关键阶段验证系统状态
- 属性测试使用 fast-check 库，每个测试至少运行 100 次迭代
- 所有属性测试都标注了对应的设计文档属性编号