

- ASD地板时光干预辅助系统 - MVP功能模块与技术架构设计
 - 目录
 - 1. 功能模块划分
 - 模块1: 评估与画像模块
 - 模块2: 智能对话助手模块
 - 模块3: 游戏推荐与实施模块
 - 模块4: AI分析与总结模块
 - 模块5: 反馈与再评估模块
 - 模块6: 可视化与报告模块
 - 2. 页面逻辑设计
 - 2.1 四页面架构
 - 2.2 页面1: AI对话 (首页)
 - 2.3 页面2: 日历
 - 2.4 页面3: 孩子档案
 - 2.5 页面4: 游戏 (三个状态)
 - 状态1: 游戏列表
 - 状态2: 游戏进行中
 - 状态3: 游戏总结
 - 2.6 页面间跳转
 - 2.7 核心流程
 - 3. 周计划推荐技术设计
 - 3.1 周计划数据结构
 - 3.2 周计划推荐Agent设计
 - 3.3 日历数据模型
 - 3.4 周计划生成流程
 - 3.5 日历页面技术实现
 - 3.6 数据库表设计
 - 3.7 API接口设计
 - 4. 系统基础与数据管理模块
 - 5. 完整技术栈清单
 - 前端技术栈
 - 后端技术栈
 - 6. 开发时间线 (6个月)
 - 第1个月
 - 第2个月
 - 第3个月
 - 第4个月

- 第5个月
- 第6个月
- 附录
 - A. 核心设计原则
 - B. 关键技术决策

ASD地板时光干预辅助系统 - MVP功能模块与技术架构设计

文档日期: 2026-01-26 版本: v1.0 (功能完整版, 6个月) 目标: 实现完整的7步闭环流程

目录

- 功能模块划分
- 页面逻辑设计
- 周计划推荐技术设计
- 完整技术栈清单
- 开发时间线

1. 功能模块划分

模块1: 评估与画像模块

功能点:

- 医院报告导入
 - 支持PDF/图片格式上传
 - OCR识别 + LLM解析报告内容
 - 提取关键指标: 诊断结果、能力评估、医生建议
- 在线量表填写
 - 内置3套权威量表: CARS-2、ADOS-2、ATEC (三选一或组合)

- 分步骤引导填写，降低家长填写门槛
- 自动保存草稿，支持随时继续

3. AI画像构建

- 融合报告数据 + 量表数据
- 建立6大维度指标：自我调节、亲密关系、双向沟通、复杂沟通、情绪想法、逻辑思考
- 动态创建自定义维度（根据孩子特点）
- 生成初始观察框架和重点关注点

核心价值: 从"零"开始认识孩子，建立个性化基础

模块2: 智能对话助手模块

功能点:

1. 通用对话界面（类似ChatGPT风格）

- 家长可随时提问："他最近有什么进步？""为什么他总喜欢转车轮？"
- AI基于孩子的完整数据回答，不是泛泛而谈
- 支持语音输入 + 文字输出

2. RAG知识库系统（2个核心知识库）

- 地板时光专业知识库：
 - 对话助手调用（回答家长专业问题）
 - 实时指引Agent调用（提供专业话术）
- 游戏知识库：
 - 对话助手调用（回答游戏相关问题）
 - 推荐Agent调用（RAG检索匹配游戏）
 - 指引Agent调用（提取具体玩法步骤）

3. Graphiti记忆交互（长期时序记忆）

- 主动查询场景（家长提问时触发）：
 - "最近眼神接触有进步吗？" → 查询Graphiti趋势分析数据
 - "这是他第一次主动微笑吗？" → 查询Graphiti里程碑记录
 - "孩子对什么最感兴趣？" → 查询Graphiti兴趣追踪数据

- **被动使用场景**（回答问题时引用）：
 - 利用State中已加载的Graphiti数据（如趋势、里程碑）
 - 不在对话过程中主动调用Graphiti写入操作
 - 所有写入操作在模块5（反馈与再评估）统一处理
- **数据获取方式**：
 - 从State中读取：优先使用State.currentContext和State.metrics中已有的Graphiti分析结果
 - 按需查询：仅当家长提出具体问题时，才调用Graphiti查询接口获取特定数据
 - 避免重复查询：State中的数据由模块5定期更新，对话时直接复用

4. 上下文记忆

- 记住对话历史
- 关联孩子实际数据
- 个性化回答

核心价值: 随时获取专业建议，降低学习成本

知识库调用关系:

地板时光专业知识库 → 对话助手 + 实时指引Agent
游戏知识库 → 对话助手 + 推荐Agent + 指引Agent
Graphiti → 对话助手（查询），模块5（写入）

模块3: 游戏推荐与实施模块

功能点:

3.1 周计划推荐部分

1. 智能周计划生成引擎

- **输入数据**：
 - 孩子画像（能力短板、兴趣偏好）
 - 历史干预数据（最近趋势、平台期检测）
 - 上周完成情况（如果有）
- **推荐策略**：

- 基于RAG检索游戏知识库（向量匹配）
- 渐进式目标设定（小步快跑，稳扎稳打）
- 游戏难度递进（周一简单 → 周日适度挑战）
- 考虑重复练习（同一游戏可在一周内出现2-3次）

○ 输出内容：

本周总目标："建立稳定的眼神接触习惯"

周一（上午10:00）：

游戏：积木传递游戏

今日目标：尝试3次眼神接触

预期时长：10-15分钟

周二（上午10:00）：

游戏：积木传递游戏（重复）

今日目标：达到5次眼神接触

预期时长：10-15分钟

周三（上午10:00）：

游戏：球类互动游戏

今日目标：在新场景中保持3次眼神接触

预期时长：15分钟

...（周四至周日）

2. 周计划详情展示

○ 周目标卡片：

- 本周要达成的核心目标（1-2个）
- 目标拆解说明（为什么这样安排）
- 预期成果（本周结束时孩子可能的变化）

○ 每日游戏卡片：

- 游戏名称、图标
- 今日小目标（具体、可衡量）
- 所需道具、预期时长
- 为什么今天玩这个游戏（透明化推荐理由）

○ 一键加入日历：

- 点击"加入日历"按钮
- 自动将7天游戏添加到日历页面
- 默认时间段：上午10:00（可在日历中调整）

3. 灵活调整机制

○ 家长可以在日历中：

- 调整游戏时间
- 跳过某天游戏
- 替换游戏（从备选列表中选择）
- 系统会根据实际完成情况动态调整下周计划

3.2 游戏实施部分

1. 分步游戏指引

- 步骤1：准备阶段（语音播报目标、注意事项）
- 步骤2：游戏进行中（分步骤语音 + 文字引导）
- 步骤3：实时话术推荐（"看看孩子会怎么做，跟随他的节奏"）
- 步骤4：禁忌话术提醒（避免命令式语气）

2. 快捷记录工具

- 底部常驻按钮：😊 微笑 👁️ 眼神 🗣️ 出声 🤝 互动 😭 情绪
- 点击即记录时间戳 + 上下文
- 语音记录功能（按住说话）：
 - 网页底部语音输入条（类似微信）
 - 按住录音，松开发送
 - 自动语音识别 + AI结构化提取

3. 视频录制（可选）

- 前置摄像头录制
- 实时显示录制时长
- 支持暂停/继续
- 不强制使用，家长可根据情况选择是否录制

核心价值: 让零基础家长也能像专业治疗师一样精准干预

模块4: AI分析与总结模块




完整流程设计:

```
步骤1: 游戏结束 → 提醒用户上传视频（可选）
├─ 家长选择: 上传视频 / 跳过视频分析
└─ State记录: hasVideo: true/false
```

步骤2：如果有视频 → 视频分析（与步骤3并行执行）

- └ 启动视频分析Agent（后台任务）
- └ AI分析内容：
 - └ 表情识别（微笑、哭泣、专注、困惑）
 - └ 眼神追踪（时长、频率、主动性）
 - └ 肢体动作分析（主动互动、刻板行为、模仿动作）
 - └ 声音分析（发声频率、语言尝试、情绪表达）
 - └ 关键片段提取（自动生成阶段name, summary, duration）
- └ 输出：videoAnalysisResult（存入State.currentSession）

步骤3：快捷记录AI验证（与步骤2并行）

- └ 交叉验证：家长快捷按钮记录 vs AI视频分析结果
- └ 一致性判断：
 - └  匹配 → 确认记录有效
 - └  冲突（家长点了但AI未检测到）→ 提示家长确认
 - └  遗漏（AI检测到但家长未点）→ 温和提示"视频里还发现了..."
- └ 输出：validatedObservations（已验证的观察记录，存入State）

步骤4：生成初步游戏总结

- └ 数据来源：已验证的快捷记录 + 语音记录 + 视频分析结果（如有）
- └ 分析内容：
 - └ 识别本次游戏的亮点（突破性进步）
 - └ 识别需要关注的地方
 - └ 与上次游戏对比
 - └ 与历史平均水平对比
- └ 输出：preliminarySummary（存入State.currentSession）

步骤5：生成智能反馈表

- └ 基于preliminarySummary动态生成3-5个问题
- └ 题型灵活：评分题、选择题、开放题
- └ 针对开放题提供引导选项
- └ HITL暂停点：等待家长填写

步骤6：家长填写反馈表

- └ 家长提交反馈
- └ State记录：parentFeedback数据
- └ 触发下一步流程

步骤7：生成最终总结（归档）

- └ 融合所有数据源：
 - └ 反馈表数据（State.currentSession.parentFeedback）
 - └ 视频分析总结（State.currentSession.videoAnalysisResult）
 - └ 整个流程的State（包含Graphiti的历史数据）
 - └ State.childProfile（孩子基础画像）
 - └ State.metrics（时序指标数据）
 - └ State.currentContext（当前上下文、趋势、关注点）
 - └ State.sessionHistory（历史会话数据）
 - └ 快捷记录 + 语音记录（已验证）
- └ 生成内容：
 - └ 完整的游戏总结（文字 + 可视化卡片）
 - └ 里程碑标注（如有）
 - └ 下一步建议
 - └ 归档数据包（保存至SQLite + 传递给模块5）
- └ 输出：finalSummary（存入State.currentSession）+ archivedData（传递给模块5）

- **并行处理**：视频分析与快捷记录验证同时进行，提高效率
 - **State驱动**：所有中间结果都存入State，避免重复调用Graphiti查询
 - **一次性融合**：在步骤7一次性融合所有数据（包括State中的Graphiti数据），而非分多次查询
-

功能点:

4.1 视频分析部分（可选功能，仅在家长上传视频时触发）

1. AI视频分析




- 表情识别（微笑、哭泣、专注、困惑）
- 眼神追踪（眼神接触时长、频率、主动性）
- 肢体动作分析（主动互动、刻板行为、模仿动作）
- 声音分析（发声频率、语言尝试、情绪表达）

2. 关键片段提取

- 分析不同阶段的时间，自动生成阶段name, summary, duration
- 自动标注重要时刻时间戳（如"首次主动互动"）
- 生成高光片段剪辑
- 置信度评分

4.2 快捷记录AI验证（仅在视频分析时触发）

1. 交叉验证机制

- 家长快捷按钮记录 vs AI视频分析结果对比
- 一致性验证：
 -  匹配 → 确认记录有效
 -  冲突（家长点了但AI未检测到）→ 提示家长确认
 -  遗漏（AI检测到但家长未点）→ 温和提示"视频里还发现了..."

2. 验证结果展示

- 不直接修改家长的记录
- 呈现证据让家长判断
- 家长可选择采纳或忽略

4.3 游戏总结部分

1. 多数据融合总结

- 融合来源：已验证的快捷记录 + 语音记录 + 视频分析
- 识别本次游戏的亮点（突破性进步）
- 识别需要关注的地方

2. 对比分析

- 与上次游戏对比
- 与历史平均水平对比
- 生成可视化总结卡片

4.4 智能反馈表

1. 个性化问卷生成

- 基于总结动态生成3-5个问题
- 题型灵活：评分题、选择题、开放题
- 针对开放题提供引导选项，降低填写门槛
- 支持语音输入回答

核心价值: AI自动分析 + 数据校验，确保记录准确性

模块5: 反馈与再评估模块

设计原则:

- 尽量少的function call：优先使用State中已有的数据，避免重复查询外部系统（包括Graphiti）
- 多用State数据：整个流程流转过程中积累的State包含了丰富的信息，包括：
 - State.childProfile：孩子基础画像
 - State.metrics：完整的时序指标数据（包含Graphiti的历史分析结果）
 - State.currentContext：当前上下文、趋势、关注点
 - State.currentSession：当前会话的所有数据（包括归档数据）
 - State.sessionHistory：历史会话数据
- Graphiti写入策略：仅在最后一步统一写入，而非多次调用

功能点:

5.1 家长反馈部分

1. 智能反馈表（已在模块4提及）

- 基于游戏总结动态生成3-5个问题
- 支持语音输入回答

2. 主观感受记录

- 家长对本次干预的主观评价
- 孩子的情绪状态、配合度
- 家长遇到的困惑或需要的帮助

5.2 数据融合与记忆更新

1. 利用State数据进行分析

- 直接读取State.metrics中的时序指标数据（包含Graphiti历史分析）
- 直接读取State.currentContext中的趋势和关注点
- 直接读取State.sessionHistory中的历史会话数据
- 避免在此时调用Graphiti查询接口

2. Graphiti记忆网络更新（一次性写入）

- 融合所有新数据：快捷记录（已验证）+ 语音记录 + 视频分析 + 家长反馈
- 为每个观察事件创建记忆节点（批量写入，减少调用次数）
- 建立事件间的关联关系（因果关系、时序关系）
- **关键优化**：使用Graphiti的批量API，一次性写入所有节点和边

3. 时序指标更新

- 更新6大维度指标的数据点
- 更新自定义维度数据
- 重新计算趋势分析（基于State.metrics，而非重新查询Graphiti）

5.3 再评估与画像更新

1. 里程碑检测

- 首次出现检测（如"第一次主动眼神接触"）
- 突破性进展识别
- 平台期预警（连续3周无变化）
- **数据来源**：State.metrics和State.sessionHistory，无需额外调用Graphiti

2. 孩子画像更新

- 对比基线数据，生成进展报告
- 更新能力评估（各维度当前水平）
- 更新兴趣偏好追踪
- 调整下一步干预重点
- 数据来源：综合State中的所有数据

3. 闭环触发

- 如果某维度需要调整 → 返回游戏推荐模块，推荐新游戏
- 如果整体进展良好 → 继续当前计划

核心价值: 每次干预都会更新孩子画像，系统越用越精准

模块6: 可视化与报告模块

说明: 这些是独立的功能模块，不直接参与主流程的State流转，而是通过页面上的按钮/菜单点击触发。用户可以随时进入可视化页面或生成报告，系统根据当前State数据展示内容。

功能点:

6.1 日历视图（核心页面）

页面布局:

17

2026年1月

[本周]

周一	周二	周三	周四	周五	周六	周日
20	21	22	23	24	25	26
✓	✓	🕒	📅	📅	📅	📅
积木	积木	球类	音乐	积木	自由	总结
+3👁️	+5👁️	10:00	10:00	10:00	玩耍	回顾

[查看本周目标]

[生成下周计划]

图例:

✓

已完成

🕒

今日待完成

📅

未来计划

+3👁️

关键指标变化（眼神接触+3次）

功能详情:

1. 日历格子状态

- 已完成 (✅):
 - 显示游戏名称
 - 显示关键成果 (如"+3👁️"表示眼神接触增加3次)
 - 点击查看详情 → 跳转到游戏记录详情页
- 今日待完成 (🕒):
 - 显示游戏名称和时间
 - 显示今日目标
 - 点击"开始游戏"按钮 → 进入游戏实施流程
- 未来计划 (📅):
 - 显示游戏名称和时间
 - 可点击编辑 (调整时间、替换游戏)

2. 渐变成长可视化

- 周视图热力图:
 - 每个格子的背景色深浅表示当天的进步程度
 - 颜色渐变: 灰色 (未完成) → 浅绿 (小进步) → 深绿 (大突破)
 - 鼠标悬停显示具体数据
- 月视图趋势线:
 - 在日历下方显示本月的核心指标趋势曲线
 - 可切换查看不同维度 (眼神接触、互动次数、情绪稳定性等)

3. 本周目标卡片

- 点击"查看本周目标"展开:

本周总目标: 建立稳定的眼神接触习惯

进度: 60% (3/5天已完成)

关键成果:

- ✅ 周一: 达成3次眼神接触 (目标3次)
- ✅ 周二: 达成5次眼神接触 (目标5次)
- 🕒 周三: 目标3次眼神接触 (进行中)

预计本周末: 孩子能在游戏中主动进行5-7次眼神接触

4. 游戏记录详情页 (点击已完成的日期进入)

- 游戏基本信息:

- 游戏名称、日期、时长
- 今日目标 vs 实际完成情况
- 关键数据：
 - 眼神接触：5次（目标5次）✅
 - 微笑：8次
 - 主动互动：3次
- AI总结：
 - 亮点："首次主动递积木"
 - 需要关注："后半段注意力下降"
- 视频片段（如果有）：
 - 关键时刻高光片段
 - 可播放完整视频
- 家长反馈：
 - 显示家长填写的反馈内容

5. 生成下周计划

- 点击"生成下周计划"按钮
- 系统基于本周完成情况生成新的周计划
- 预览后可一键加入日历

6.2 多维度成长可视化

页面入口: 日历页面顶部导航 → "成长报告"

1. 多维度雷达图

- 展示6大情绪发展里程碑的当前水平
- 支持时间轴播放（查看1个月/3个月/半年的变化）
- 可叠加多个时间点对比（如"本月 vs 上月"）

2. 里程碑时间线

- 标注重要突破性事件
- 每个里程碑关联证据（视频片段/记录文本）
- 时间轴缩放查看
- 与日历联动：点击里程碑 → 跳转到对应日期的游戏记录

3. 趋势图

- 各维度指标的变化曲线

- 平滑趋势线 + 实际数据点
- 标注关键干预节点（哪天开始新游戏）
- **与日历联动：** 点击数据点 → 跳转到对应日期

4. 周对比视图

- 按周聚合显示关键指标
- 柱状图对比：第1周 vs 第2周 vs 第3周...
- 显示每周的平均值和峰值

6.3 报告生成

1. 家长版报告

- 可视化图表 + 文字总结
- 突出孩子的进步和亮点
- 提供下一步建议
- 支持导出PDF
- **新增：** 包含日历视图截图，展示完成情况

2. 医生版报告（标准化医学报告）

- 包含评估数据、干预记录、进展分析
- 使用专业术语和标准格式
- 附带视频证据片段
- **新增：** 包含周计划执行情况统计
- 支持一键导出，便于医家沟通

6.4 随时记录功能（辅助）

1. 日常观察记录

- 不限于游戏时间，随时可记录
- 语音/文字/快捷方式
- AI自动分类并融入孩子画像
- **新增：** 可关联到日历中的某一天

2. 兴趣追踪

- 识别新出现的兴趣点
- 追踪兴趣演变趋势
- 提示如何利用兴趣进行干预

- **新增**：在日历中标记兴趣爆发日

核心价值: 让每个微小的进步都被看见、被记录、被理解，通过日历让成长轨迹一目了然

- 包含评估数据、干预记录、进展分析
- 使用专业术语和标准格式
- 附带视频证据片段
- 支持一键导出，便于医家沟通

6.3 随时记录功能（辅助）

1. 日常观察记录

- 不限于游戏时间，随时可记录
- 语音/文字/快捷方式
- AI自动分类并融入孩子画像

2. 兴趣追踪

- 识别新出现的兴趣点
- 追踪兴趣演变趋势
- 提示如何利用兴趣进行干预
- **新增**：在日历中标记兴趣爆发日

核心价值: 让每个微小的进步都被看见、被记录、被理解，通过日历让成长轨迹一目了然

2. 页面逻辑设计

2.1 四页面架构

页面1：AI对话（首页）
页面2：日历
页面3：孩子档案
页面4：游戏

2.2 页面1: AI对话（首页）

核心功能：




- 对话框：家长随时提问，AI回答
- 按钮：孩童评估、地板游戏
- 左滑菜单：快速跳转其他页面

使用场景：

- "他最近有什么进步？"
- "今天该玩什么游戏？"
- 点击"孩童评估" → 上传报告或填写量表

2.3 页面2: 日历

核心功能：


- 日历格子：显示每天的游戏
- 状态标识： 已完成  今日  未来
- 颜色深浅：灰色→浅绿→深绿（进步程度）
- 本周目标卡片：显示本周要达成什么
- 生成下周计划按钮

使用场景：

- 查看本周游戏安排
- 点击今日游戏 → 跳转到页面4开始玩
- 点击已完成日期 → 查看那天的记录
- 周末点击"生成下周计划" → 预览 → 确认


日历格子示例：

周一 20日

 积木游戏

+3👁️ 眼神接触

周二 21日

 球类游戏

10:00 开始

2.4 页面3: 孩子档案

核心功能：

- 孩子基本信息：姓名、年龄、诊断
- 雷达图：6大维度当前水平
- 时间线：重要突破事件
- 趋势图：各维度变化曲线
- 孩童评估入口（同页面1）
- 导出报告按钮

使用场景：

- 查看孩子整体情况
 - 点击时间线上的里程碑 → 跳转到日历对应日期
 - 点击趋势图数据点 → 跳转到日历对应日期
 - 导出报告给医生
-

2.5 页面4: 游戏（三个状态）

状态1: 游戏列表

显示内容：

- 本周推荐的游戏
- 每个游戏卡片：名称、目标、时长、推荐理由
- 搜索框：输入"积木游戏""提升眼神接触"

操作：

- 点击游戏卡片 → 查看详情
 - 点击"开始游戏" → 进入状态2
-

状态2: 游戏进行中

显示内容：

- 当前步骤指引（文字+语音播报）

- 快捷记录按钮：😊👁️🗣️👏😓
- 语音记录：按住说话
- 视频录制（可选）

操作：

- 跟随指引玩游戏
- 随时点击快捷按钮记录
- 随时按住说话记录观察
- 点击"下一步" → 继续
- 点击"结束游戏" → 进入状态3

状态3: 游戏总结

显示内容：

- 上传视频（可选，可跳过）
- AI总结：本次亮点、关键数据
- 反馈表：3-5个问题（评分/选择/开放）

操作：

- 选择上传视频或跳过
- 查看AI总结
- 填写反馈表（支持语音输入）
- 点击"提交" → 返回游戏列表

2.6 页面间跳转

页面1（AI对话）

- └ 左滑 → 页面2/3/4
- └ 按钮 → 页面4

页面2（日历）

- └ 点击今日游戏 → 页面4（状态2）
- └ 点击已完成日期 → 展开详情

页面3（档案）

- └ 点击里程碑/数据点 → 页面2

2.7 核心流程

首次使用：

1. 页面1 → 点击"孩童评估" → 上传报告
2. AI生成第一周计划
3. 跳转到页面2 → 查看本周游戏
4. 点击今日游戏 → 页面4开始玩
5. 完成游戏 → 填写反馈
6. 返回页面2 → 看到日历格子变绿✔

日常使用：

1. 打开页面2 (日历)
2. 查看今日游戏
3. 点击"开始" → 页面4玩游戏
4. 填写反馈 → 提交
5. 返回页面2 → 查看更新

周末生成计划：

1. 周日晚上收到提醒
2. 打开页面2
3. 点击"生成下周计划"
4. 预览 → 确认
5. 查看下周安排

3. 周计划推荐技术设计

3.1 周计划数据结构

```
interface WeeklyPlan {
  id: string;
  childId: string;
  weekStartDate: string; // "2026-01-20"
  weekEndDate: string;   // "2026-01-26"

  // 本周总目标
  weeklyGoal: {
    title: string; // "建立稳定的眼神接触习惯"
    description: string; // 目标拆解说明
    targetDimensions: string[]; // ["eye_contact", "two_way_communication"]
    expectedOutcome: string; // 预期成果
  };

  // 每日游戏计划
  dailyPlans: DailyPlan[];

  // 计划状态
  status: 'draft' | 'active' | 'completed';

  // 完成情况统计
  progress: {
    completedDays: number;
    totalDays: number;
    completionRate: number; // 0-1
  };

  // 生成依据
  reasoning: {
    basedOnMetrics: any; // 基于哪些指标数据
    basedOnTrends: any; // 基于哪些趋势
    lastWeekPerformance?: any; // 上周完成情况
  };

  createdAt: string;
  updatedAt: string;
}

interface DailyPlan {
  date: string; // "2026-01-20"
  dayOfWeek: string; // "周一"

  game: {
    id: string;
    name: string;
    category: string;
  };

  // 今日小目标
  dailyGoal: {
    title: string; // "尝试3次眼神接触"
    metrics: {
      [key: string]: number; // {"eye_contact": 3}
    };
  };
};
```

```
// 时间安排
scheduledTime: string; // "10:00"
estimatedDuration: number; // 15 (分钟)

// 推荐理由
reasoning: string;

// 完成状态
status: 'pending' | 'in_progress' | 'completed' | 'skipped';

// 实际完成情况 (完成后填充)
actualResult?: {
  sessionId: string;
  completedAt: string;
  achievedMetrics: any;
  summary: string;
};
}
```

3.2 周计划推荐Agent设计

name: "weekly_plan_agent"

输入:

- childProfile: 孩子画像
- metrics: 时序指标数据
- currentContext: 当前上下文 (趋势、关注点)
- lastWeekPlan?: 上周计划 (如果有)
- lastWeekPerformance?: 上周完成情况

输出:

- weeklyPlan: **WeeklyPlan**对象

核心逻辑:

1. 分析当前状态
 - 识别需要重点提升的维度 (最多2个)
 - 分析上周完成情况 (如果有)
 - 检测平台期或快速进步期
2. 设定本周总目标
 - 基于优先维度设定1-2个核心目标
 - 目标必须: 具体、可衡量、渐进式、可达成
 - 示例: "建立稳定的眼神接触习惯"而非"提升社交能力"
3. RAG检索候选游戏
 - 构建查询向量 (融合孩子画像+本周目标)
 - 从游戏知识库检索**Top 20**候选游戏
 - 过滤条件: 年龄适配、难度适中、兴趣匹配
4. 生成7天游戏序列
 - 策略1: 渐进式难度 (周一简单 → 周日适度挑战)
 - 策略2: 重复练习 (同一游戏可出现2-3次)

- 策略3：场景切换（避免连续3天同类游戏）
- 策略4：周末放松（周六自由玩耍，周日总结回顾）

5. 为每天设定小目标

- 基于本周总目标拆解
- 每日目标递进（周一3次 → 周二5次 → 周三保持5次）
- 目标数值化、可验证

6. 生成推荐理由

- 为整体计划生成reasoning
- 为每日游戏生成reasoning

tools:

- rag_search_games: RAG检索游戏知识库
- calculate_priority_dimensions: 计算维度优先级
- generate_progressive_goals: 生成渐进式目标序列
- validate_plan_feasibility: 验证计划可行性

3.3 日历数据模型

```
interface CalendarEvent {
  id: string;
  childId: string;
  date: string; // "2026-01-20"

  // 关联的每日计划
  dailyPlanId: string;
  weeklyPlanId: string;

  // 游戏信息
  game: {
    id: string;
    name: string;
  };

  // 时间安排
  scheduledTime: string; // "10:00"
  actualStartTime?: string;
  actualEndTime?: string;

  // 状态
  status: 'pending' | 'in_progress' | 'completed' | 'skipped';

  // 完成后的数据
  sessionId?: string; // 关联到干预会话
  keyMetrics?: {
    [key: string]: {
      value: number;
      change: number; // 相比上次的变化
    };
  };
};
```

```
// 可视化数据
visualScore?: number; // 0-100, 用于热力图颜色

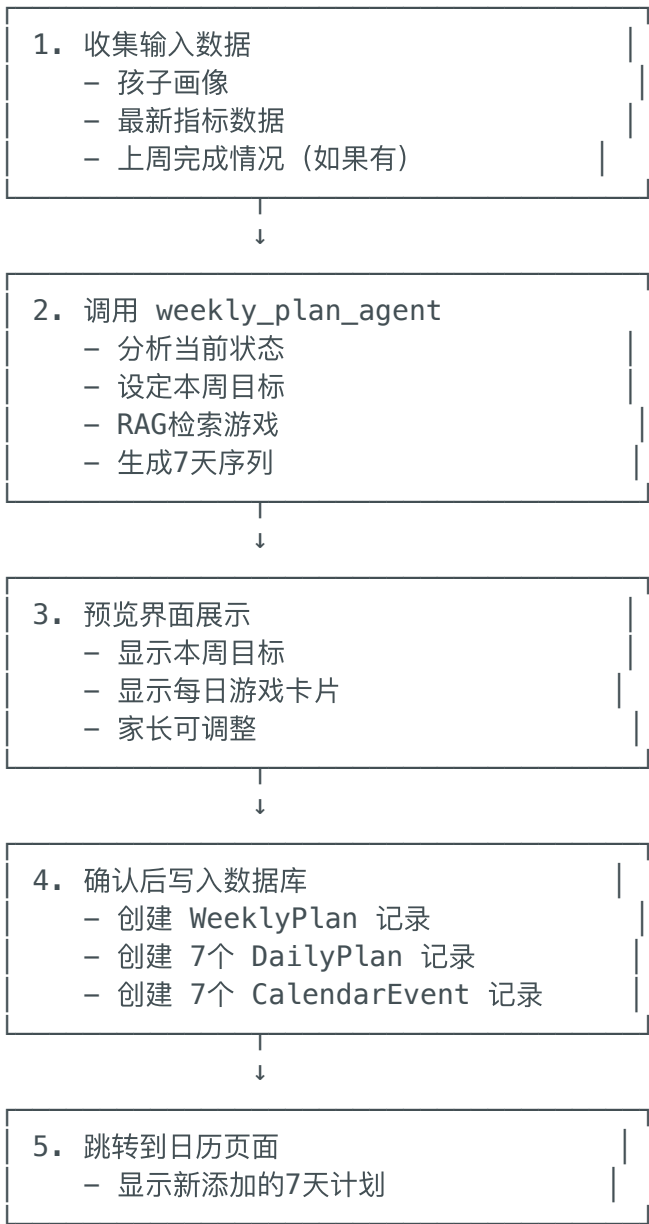
createdAt: string;
updatedAt: string;
}
```

3.4 周计划生成流程

触发时机：

1. 初次评估完成后 → 自动生成第一周计划
2. 家长点击"生成下周计划" → 基于本周完成情况生成
3. 周日晚上自动提醒 → 引导家长生成下周计划

流程：



3.5 日历页面技术实现

前端组件结构:



热力图颜色计算:

```
function calculateHeatmapColor(event: CalendarEvent): string {
  if (event.status !== 'completed') {
    return '#E5E5E5'; // 灰色 (未完成)
  }

  const score = event.visualScore || 0; // 0-100

  if (score < 30) return '#C6F6D5'; // 浅绿 (小进步)
  if (score < 60) return '#9AE6B4'; // 中绿
  if (score < 80) return '#68D391'; // 深绿
  return '#38A169'; // 最深绿 (大突破)
}

// visualScore 计算逻辑
function calculateVisualScore(session: Session): number {
  let score = 50; // 基础分

  // 目标达成度 (+0 to +30)
  const goalAchievement = session.goalAchievementRate; // 0-1
  score += goalAchievement * 30;

  // 里程碑突破 (+20)
  if (session.hasMilestone) {
    score += 20;
  }

  // 相比上次的进步 (+0 to +20)
  const improvement = session.improvementRate; // 0-1
  score += improvement * 20;
}
```



```
    return Math.min(100, score);  
}
```

3.6 数据库表设计

-- 周计划表

```
CREATE TABLE weekly_plans (  
  id VARCHAR(50) PRIMARY KEY,  
  child_id VARCHAR(50) NOT NULL,  
  week_start_date DATE NOT NULL,  
  week_end_date DATE NOT NULL,  
  weekly_goal_title VARCHAR(200),  
  weekly_goal_description TEXT,  
  target_dimensions TEXT[],  
  expected_outcome TEXT,  
  status VARCHAR(20),  
  completed_days INTEGER DEFAULT 0,  
  total_days INTEGER DEFAULT 7,  
  reasoning JSONB,  
  created_at TIMESTAMP DEFAULT NOW(),  
  updated_at TIMESTAMP DEFAULT NOW()  
);
```

-- 每日计划表

```
CREATE TABLE daily_plans (  
  id VARCHAR(50) PRIMARY KEY,  
  weekly_plan_id VARCHAR(50) REFERENCES weekly_plans(id),  
  child_id VARCHAR(50) NOT NULL,  
  date DATE NOT NULL,  
  day_of_week VARCHAR(10),  
  game_id VARCHAR(50),  
  game_name VARCHAR(200),  
  daily_goal_title VARCHAR(200),  
  daily_goal_metrics JSONB,  
  scheduled_time TIME,  
  estimated_duration INTEGER,  
  reasoning TEXT,  
  status VARCHAR(20),  
  actual_result JSONB,  
  created_at TIMESTAMP DEFAULT NOW(),  
  updated_at TIMESTAMP DEFAULT NOW()  
);
```

-- 日历事件表

```
CREATE TABLE calendar_events (  
  id VARCHAR(50) PRIMARY KEY,  
  child_id VARCHAR(50) NOT NULL,  
  date DATE NOT NULL,  
  daily_plan_id VARCHAR(50) REFERENCES daily_plans(id),  
  weekly_plan_id VARCHAR(50) REFERENCES weekly_plans(id),  
  game_id VARCHAR(50),  
  game_name VARCHAR(200),
```

```

scheduled_time TIME,
actual_start_time TIMESTAMP,
actual_end_time TIMESTAMP,
status VARCHAR(20),
session_id VARCHAR(50), -- 关联到干预会话
key_metrics JSONB,
visual_score INTEGER,
created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW()
);

-- 索引
CREATE INDEX idx_weekly_plans_child ON weekly_plans(child_id,
week_start_date);
CREATE INDEX idx_daily_plans_weekly ON daily_plans(weekly_plan_id);
CREATE INDEX idx_calendar_events_date ON calendar_events(child_id, date);

```

3.7 API接口设计

```

// 生成周计划
POST /api/weekly-plans/generate
Request: {
  childId: string;
  weekStartDate: string; // "2026-01-20"
}
Response: {
  weeklyPlan: WeeklyPlan;
}

// 获取周计划
GET /api/weekly-plans/:childId/current
Response: {
  weeklyPlan: WeeklyPlan;
  dailyPlans: DailyPlan[];
}

// 更新每日计划
PATCH /api/daily-plans/:id
Request: {
  scheduledTime?: string;
  gameId?: string; // 替换游戏
}

// 获取日历数据
GET /api/calendar/:childId?month=2026-01
Response: {
  events: CalendarEvent[];
  weeklyPlans: WeeklyPlan[];
}

// 标记游戏完成
POST /api/calendar/events/:id/complete

```

```
Request: {  
  sessionId: string;  
  keyMetrics: any;  
}  
Response: {  
  event: CalendarEvent;  
  visualScore: number;  
}
```

4. 系统基础与数据管理模块

功能点:

4.1 孩子档案管理

1. 多孩子支持

- 添加/编辑孩子档案
- 切换当前查看的孩子
- 每个孩子独立的数据隔离

4.2 会话管理

1. 游戏会话生命周期

- 创建会话 → 开始游戏 → 进行中 → 暂停 → 结束
- 支持中断恢复 (Checkpoint机制)
- 会话历史查询

2. 人机协作暂停点

- 等待家长填写反馈时暂停工作流
- 保存完整上下文
- 家长提交后自动恢复

4.3 数据存储

1. SQLite数据库 (用户数据)

- 孩子档案信息
- 干预会话记录
- 微观观察记录

- 时序指标数据
- 周计划和每日计划
- 日历事件

2. PostgreSQL + pgvector（游戏知识库）

- 游戏信息表
- 向量嵌入索引
- RAG检索支持

3. Graphiti记忆网络（长期时序记忆）

- 记忆节点（观察事件）
- 关系边（因果、时序）
- 自动趋势分析和里程碑检测

4.4 通知与提醒

1. 干预提醒

- 定时推送提醒（根据家长设定的计划）
- "今天该关注什么"智能提醒
- 周日晚上提醒生成下周计划

2. 重要发现通知

- 检测到里程碑时推送庆祝通知
- 平台期预警通知

4.5 API接口预留

- 保留用户认证相关API接口设计
- 便于后续快速接入用户系统

核心价值: 系统基础功能，保障数据管理和会话流转

5. 完整技术栈清单

前端技术栈

```
{
  "核心框架": {
    "framework": "React 18",
    "language": "TypeScript 5.0+",
    "build_tool": "Vite 5.0+"
  },
  "UI组件库": {
    "mobile": "Ant Design Mobile 5.x",
    "calendar": "react-big-calendar / 自定义日历组件",
    "charts": "recharts / echarts",
    "icons": "@ant-design/icons"
  },
  "状态管理": {
    "local_state": "Zustand 4.x",
    "server_state": "React Query 5.x"
  },
  "路由": {
    "router": "React Router 6.x"
  },
  "样式方案": {
    "css": "CSS Modules",
    "utility": "可选 TailwindCSS"
  },
  "多媒体处理": {
    "video_recording": "MediaRecorder API",
    "audio_recording": "Web Audio API",
    "speech_synthesis": "Web Speech API (TTS)"
  },
  "实时通信": {
    "websocket": "Socket.IO Client"
  },
  "工具库": {
    "date": "dayjs",
    "http": "axios",
    "form": "react-hook-form"
  }
}
```

后端技术栈

```
{
  "开发环境": {
    "python": "Python 3.11+",
    "node": "Node.js 18+"
  },
  "核心框架": {
    "runtime": "Node.js 18+",
    "framework": "Express.js 4.18+",
    "language": "TypeScript 5.0+"
  },
}
```

```

"AI框架": {
  "langchain": "@langchain/core",
  "langgraph": "@langchain/langgraph",
  "openai": "@langchain/openai (可选)"
},
"AI服务": {
  "llm": "DeepSeek-V3 (默认)",
  "embedding": "通义 Embedding",
  "vision": "DeepSeek-Vision (默认)",
  "备用模型": "支持OpenAI GPT-4o (通过配置切换)",
  "speech_stt": "阿里云语音识别",
  "speech_tts": "阿里云语音合成"
},
"数据库": {
  "关系型": "PostgreSQL 16 + pgvector (游戏知识库)",
  "嵌入式": "SQLite (用户数据、干预记录、周计划)",
  "图记忆": "Graphiti (长期时序记忆)"
},
"向量检索": {
  "library": "pgvector",
  "similarity": "cosine similarity"
},
"文件处理": {
  "pdf_parse": "pdf-parse",
  "ocr": "Tesseract.js / 百度OCR API",
  "video_process": "FFmpeg (可选)"
},
"实时通信": {
  "websocket": "Socket.IO"
},
"认证与安全": {
  "jwt": "jsonwebtoken",
  "encryption": "bcrypt",
  "validation": "joi"
},
"任务队列": {
  "queue": "Bull + Redis (可选, 用于视频分析)"
},
"开发工具": {
  "testing": "Jest + Supertest",
  "linting": "ESLint + Prettier",
  "api_doc": "Swagger/OpenAPI"
}
}

```

6. 开发时间线（6个月）

第1个月

- ☒ 环境搭建 + 技术验证
- ☒ 初评估模块
- ☒ 游戏知识库构建 (50个游戏)
- ☒ 周计划推荐Agent开发

第2个月

- ☒ 日历页面开发 (核心功能)
- ☒ 周计划生成与展示
- ☒ 游戏实施流程
- ☒ 实时指引Agent + 观察捕获Agent

第3个月

- ☒ 视频分析Agent + 总结Agent
- ☒ 反馈表Agent + HITL实现
- ☒ 日历事件完成状态更新
- ☒ 热力图可视化



第4个月

- ☒ 记忆更新Agent + 再评估Agent
- ☒ Graphiti集成
- ☒ 闭环工作流调试
- ☒ 下周计划自动生成

第5个月

- ☒ 成长报告页面
- ☒ 多维度可视化组件
- ☒ 对话助手Agent
- ☒ 完整流程测试

第6个月

-  性能优化
-  用户体验优化
-  部署上线

附录

A. 核心设计原则

1. 以周为单位的渐进式干预 - 每周一个小目标，稳扎稳打
2. 日历化管理 - 让干预计划可视化、可追踪
3. 四页面架构 - AI对话、日历、档案、游戏，各司其职
4. 家长是主角，AI是助手 - AI不会"自说自话"，呈现证据让家长判断
5. 记录门槛极低 - 快捷按钮、按住说话、引导式填写
6. 让看不见的变可见 - 里程碑检测、趋势分析、热力图展示
7. 从记录到行动的闭环 - 每个分析都导向下一步建议

B. 关键技术决策

决策点	选择	理由
计划单位	周计划	符合家长认知习惯，便于坚持
核心页面	日历视图	直观展示进度和成果
目标设定	渐进式小目标	降低挫败感，提升成就感
可视化	热力图+趋势线	让微小进步可见
页面架构	四页面+状态切换	清晰的功能分区，流畅的流程

文档结束