AI 旅游助手 Agent 产品需求文档（PRD）

版本：V1.0
产品类型：任务型 AI Agent
用途：需求评审 / Demo 验证 / 开发实现 / 测试验收

1. 项目概述

1.1 项目背景

用户规划旅行时通常需要在地图、天气、攻略、交通、酒店等多个产品之间切换，信息分散且需要自行整合目的地、时间、预算和偏好。传统搜索或问答产品能够分别提供信息，但难以连续完成“理解需求—补充信息—查询实时信息—规划方案—持续调整”的完整任务。

因此设计 AI 旅游助手 Agent：以自然语言对话为主要交互方式，根据用户需求判断下一步需要的信息和工具，在必要时追问，并调用天气、交通、酒店、景点等外部能力，最终生成可执行的旅行方案。

1.2 目标用户

用户类型	核心需求
普通旅行用户	降低规划成本，快速获得可执行行程
有明确需求用户	根据日期、预算、偏好生成个性化方案
需要持续调整用户	记住有效条件并根据新要求调整方案

1.3 核心问题

- 旅行信息分散，需要跨多个产品查询。
- 用户常只提供部分条件，系统需要主动补全关键约束。
- 天气、价格、营业状态等信息具有时效性，不能仅依赖模型内部知识。
- 生成攻略后需要继续根据用户反馈调整。

1.4 产品定位

一个以“帮助用户完成旅行规划任务”为目标的任务型 AI Agent，而非单纯回答旅游问题的聊天机器人。

2. 项目目标与范围

2.1 产品目标

目标类型	目标
用户目标	通过自然语言描述需求即可获得旅行方案
业务目标	减少用户在多个产品间切换和手动整理信息的成本
AI 目标	正确理解需求、合理追问、正确调用工具，并基于真实结果生成方案

2.2 MVP 范围

- 支持目的地、日期、人数、预算、旅行偏好的多轮收集。
- 支持天气、景点、酒店、交通/路线信息查询。
- 支持基于已获取信息生成 1—7 天旅行计划。
- 支持用户继续修改条件并重新规划。
- 支持会话级 Memory 保存当前旅行任务的关键条件。

2.3 非本期范围

- 直接支付、购买机票、购买酒店。
- 自动办理签证等高风险事务。
- 长期保存所有用户历史旅行偏好。
- 复杂多 Agent 协作，MVP 优先验证单 Agent 效果。

3. 核心用户场景

3.1 信息完整，直接规划

用户：“我 10 月 1 日到 5 日去东京，两个人，预算 2 万元，喜欢动漫和美食，帮我规划一下。”

1. Agent 提取目的地、日期、人数、预算和偏好。
2. 判断关键信息完整。
3. 根据需要调用天气、景点、酒店等工具。
4. 综合工具返回结果生成每日行程。
5. 输出后允许用户继续调整。

3.2 信息缺失，需要追问

用户：“帮我规划一次旅行。” Agent 不得直接生成具体方案，应按“最少必要追问”原则获取目的地、日期/天数等关键条件。示例：“你想去哪里、计划玩几天？如果暂时没有确定，也可以告诉我大概预算和旅行偏好。”

3.3 方案中途修改

用户先确定“东京 5 天、预算 2 万元”，后续说：“预算改成 1 万元，酒店不用太贵。”Agent 应识别条件变化，更新 Memory，并重新规划受影响部分，而不是继续沿用旧条件。

4. 产品整体流程

用户输入 → 理解需求 → 读取 Memory → 判断信息是否完整 → 追问或 Planning → 选择 Tool → Function Calling → Tool/API 执行 → 获取结果 → 判断是否继续调用 Tool → 生成方案 → 用户继续调整 → 更新 Memory。

4.1 Agent 循环

6. Observe：读取用户输入、当前对话上下文及 Memory。
7. Plan：判断用户目标、缺失信息及下一步行动。
8. Act：选择是否调用 Tool，并通过 Function Calling 传递参数。
9. Observe Result：读取 Tool 返回结果。

10. Re-plan：判断是否完成任务；未完成则继续调用工具或追问。
11. Answer：完成任务后生成最终结果。

产品只需定义可观察的任务状态、工具调用结果和最终输出，不要求展示模型内部隐藏推理过程。

5. Agent 能力设计

5.1 LLM

职责：理解自然语言、提取旅行条件、判断是否追问、选择工具、组织工具结果并生成最终方案。模型选型重点评测中文理解、多轮对话、Function Calling 稳定性、结构化输出能力和成本，不在 PRD 中强绑定具体厂商。

5.2 Skill 设计

模块	规则
角色	你是 AI 旅行规划助手，帮助用户完成旅行规划任务。
目标	理解需求，补充必要信息，获取需要的实时信息，生成可执行方案。
信息收集	优先提取目的地、日期/天数、人数、预算、偏好；避免一次性问卷式追问。
工具使用	涉及实时天气、价格、营业状态时优先调用对应工具，不得编造。
规划	复杂任务先拆分，再根据工具结果完成规划。
输出	方案包含每日安排、推荐理由、交通/时间建议和预算提示。
边界	信息不足时追问；工具失败时如实说明；不把不确定信息说成确定事实。

5.3 Planning 设计

MVP 采用“LLM 自主决策 + 产品规则约束”。Planning 定义 Agent 面对复杂任务时的决策原则，而不要求产品经理固定每一步。

阶段	判断
需求理解	用户最终想完成什么任务？
信息检查	完成任务还缺哪些必要信息？
任务拆分	是否需要天气、景点、酒店、交通等子任务？
工具选择	哪个 Tool 最适合获取当前缺失信息？
结果判断	Tool 结果是否足够完成任务？
重新规划	条件变化或结果不足时是否需要再次查询？
结束判断	任务是否完成，可以输出最终方案？

约束：设置最大工具调用次数和超时限制，避免无限循环；达到阈值后进行降级或提示用户。

5.4 Tool 设计

Tool	能力	主要参数	调用条件	返回

get_weather	查询天气	城市、日期	规划涉及天气或用户明确询问	天气、温度、降雨等
search_attractions	查询景点	城市、偏好	需要推荐景点/安排路线	景点名称、类型、位置、开放信息
search_hotels	查询酒店	城市、日期、人数、预算	需要住宿推荐	酒店列表、价格、位置
search_resturants	查询饭店	位置、特色菜	需要饮食推荐	大众点评评价
search_transport	查询交通/路线	出发地、目的地、日期	需要比较交通方式	交通方案、时间、价格
search_redbook	查询小红书	攻略	需要景点攻略	点赞最多的帖子

Function Calling 流程：LLM 根据 Tool 名称、描述和参数 Schema 生成调用请求 → Agent 运行时/后端执行实际 API 或函数 → 将结果返回 LLM → LLM 决定继续行动或回答。

- Tool 描述必须说明“做什么”和“什么时候使用”。
- 参数定义明确类型、必填项和格式。
- 参数缺失时优先追问，不允许无依据猜测。
- Tool 失败必须返回可识别状态，Agent 不得把失败当作成功。
- 未来涉及下单/支付等写操作必须设计用户确认。

5.5 Memory 设计

MVP 采用“会话级结构化 Memory + 对话 Context”。

信息	是否记忆	原因
目的地	是	持续影响规划
日期/天数	是	影响工具查询与行程
人数	是	影响酒店、预算
预算	是	影响筛选与方案
旅行偏好	是	影响景点/餐饮推荐
酒店偏好	是	影响住宿推荐
临时闲聊	低优先级	通常不影响任务
工具报错原文	否	不作为用户偏好

用户明确修改条件时覆盖旧值；存在冲突时以用户最新明确表达为准或主动确认。Context 是当前模型可见的上下文；Memory 是产品选择保存、在后续需要时重新注入 Context 的信息。

6. 功能需求

6.1 需求信息识别

需求 ID	需求	验收
FR-01	识别目的地、日期/天数、人数、预算、偏好等实体，能识别中文数字，还有自己这种表示数量的词语	测试集达到预设实体识别目标
FR-02	识别新建规划、补充信息、修改	分类错误进入 Bad Case 分析

	条件或普通咨询	
FR-03	识别条件修改并更新 Memory	后续方案不得继续使用已废弃条件

6.2 多轮追问

- 仅追问当前任务真正必要的信息。
- 已经获得的信息不得重复询问。
- 用户无法提供精确条件时允许使用范围或推荐模式。
- 用户拒绝回答非必要问题时，可基于已知条件继续，并说明假设。

6.3 旅行方案生成

- 行程概览：目的地、日期/天数、人数、预算和主要偏好。
- 每日安排：上午/下午/晚上或时间段。
- 三餐推荐：早/午/晚餐，餐馆推荐不超过推荐景点 3km
- 推荐理由：说明与用户偏好的关联。
- 交通与时间建议。
- 小红书攻略贴链接
- 预算提示：区分实时查询价格与估算信息。
- 提示用户可继续修改日期、预算、酒店偏好等。

7. AI 行为边界与异常兜底

场景	Agent 行为
信息不足	追问关键缺失信息，不无依据假设
工具无结果	明确说明未获取到结果，可建议修改条件
Tool 调用失败	说明实时查询暂不可用，不编造结果
Tool 结果冲突	优先再次查询或提示信息可能存在差异
高风险操作	要求明确确认；MVP 不实际执行支付/购买
模型不确定	使用不确定性表达，不输出伪确定事实
工具循环	达到最大调用次数后停止，说明当前限制

8. 交互与页面要求

- 用户可自然语言输入需求。
- 展示“正在查询/规划”等用户可理解状态，不暴露内部 Prompt 或隐藏推理。
- 工具失败时展示可理解的错误提示。
- 最终方案支持继续对话修改，例如“第三天换成轻松一点”。
- 可选信息确认卡：展示目的地、日期、人数、预算、偏好，允许用户修改后开始规划。
- 如果用户对当前的输出结果不满意，有其他想要知道的景点或者飞机/火车班次，可以选择追问

9. 数据埋点与指标

9.1 关键埋点

事件	说明
agent_session_start	开始一次旅行任务
user_message	用户输入
memory_update	Memory 新增/修改
clarification_asked	Agent 发起追问
tool_call	发起工具调用
tool_success/tool_failure	工具成功/失败
plan_generated	成功生成方案
plan_regenerated	用户要求重新规划
user_feedback	点赞/点踩/问题反馈
session_complete	任务完成或退出

9.2 核心指标

维度	指标
使用	DAU、任务发起量、会话完成率
任务效果	方案生成成功率、任务完成率、用户采纳率
Agent 能力	工具调用成功率、参数正确率、无效调用率、追问有效率
AI 质量	需求理解准确率、事实错误率、Bad Case 率
体验	平均轮次、平均响应时间、重新提问率
成本	单次任务 Token 成本、单次工具调用成本

10. 评测与 Bad Case 方案

10.1 测试集分类

Case 类型	目的
正常 Case	验证完整需求能否生成方案
信息缺失	验证是否正确追问
条件修改	验证 Memory 更新
模糊表达	验证意图理解
工具失败	验证兜底
工具误调用	验证是否调用不必要工具
事实风险	验证是否编造实时信息
边界 Case	验证超预算、超长行程、相互矛盾条件

10.2 Bad Case 定位流程

发现最终答案错误 → 还原运行链路 → 查看原始 Query → 查看 Memory → 查看意图理解/Planning → 查看 Tool 选择 → 查看 Function Calling 参数 → 查看 Tool 返回结果 → 查看最终 Context/Prompt → 判断 LLM 生成 → 分类根因 → 修复 → 回归测试。

现象	优先排查
重复追问已有信息	Context/Memory 读取或更新
调用错误工具	Tool 描述、Skill 规则、意图判断
正确工具但参数错误	实体提取、参数 Schema、Function Calling
工具结果正确但方案错误	Prompt/Skill、Context 组织、LLM 生成
未查实时信息直接编造	Tool 调用规则、Prompt 约束
修改条件后仍用旧方案	Memory 更新与覆盖逻辑

11. 权限与安全

- MVP 仅开放查询和规划类 Tool。
- 未来新增写操作必须定义风险等级。
- 中高风险操作必须展示关键参数并获得用户确认。
- 敏感信息只在完成任务所需范围内使用。
- 日志用于定位问题时遵循脱敏和权限控制原则。

12. 验收标准

类别	验收要求
基础功能	可完成需求理解、追问、工具调用、方案生成和继续调整
Memory	用户修改条件后，后续规划正确使用新条件
Tool	参数完整时正确调用；参数缺失时不盲目调用
异常	失败、无结果、超时等场景有明确兜底
AI 质量	达到预设评测集指标；关键事实错误不超过阈值
性能	达到项目预设响应时间和稳定性要求
安全	未经确认不得执行未来定义的高风险操作

13. 开发与协作说明

13.1 产品经理职责

- 定义用户问题、目标用户和产品边界。
- 设计 Agent 整体任务流程。
- 设计 Skill：角色、目标、规则和行为边界。
- 定义 Tool 清单、调用条件、参数、返回值和异常处理。
- 定义 Memory：记什么、何时更新、何时读取、如何覆盖。
- 定义 Planning 原则和任务结束条件。
- 建立测试集，分析 Bad Case，推动 Prompt、Tool、Memory 等迭代。

13.2 技术实现可能涉及

- Agent 运行时/编排框架。
- LLM 接入与 Function Calling 协议。
- API/数据库/搜索服务封装为 Tool。

- Memory 存储与检索。
- 日志、Tracing 和监控。
- 权限、安全、限流、超时与重试机制。

产品经理不一定亲自完成代码开发，但应通过 PRD 明确 Agent 要做什么、每个组件承担什么职责、何时调用，以及如何判断成功或失败。

14. MVP 上线流程

12. 完成 PRD 和核心流程评审。
13. 使用 workbuddy 平台搭建 Demo，验证 Skill、Tool、Memory 和多轮逻辑。
14. 建立测试集并进行正常 Case、异常 Case 和 Bad Case 测试。
15. 根据测试结果优化 Prompt、Tool 描述、参数 Schema、Memory 规则。
16. 技术实现生产版本并接入日志与监控。
17. 联调与验收。
18. 小流量灰度上线。
19. 监控任务完成率、工具失败率、Bad Case 率、响应时间和成本。
20. 根据真实用户反馈持续迭代。

15. 产品经理视角总结

本项目的核心不是“让模型回答旅游问题”，而是设计一个能够围绕旅行任务持续行动的 Agent。完整链路为：用户任务 → Skill 定义行为 → Memory 保存关键条件 → Planning 决定下一步 → Tool 提供外部能力 → Function Calling 完成调用 → 结果反馈给 LLM → 继续判断或生成最终方案 → 通过评测和 Bad Case 持续迭代。