《奇门遁甲工作台 v1.0》软件设计说明书

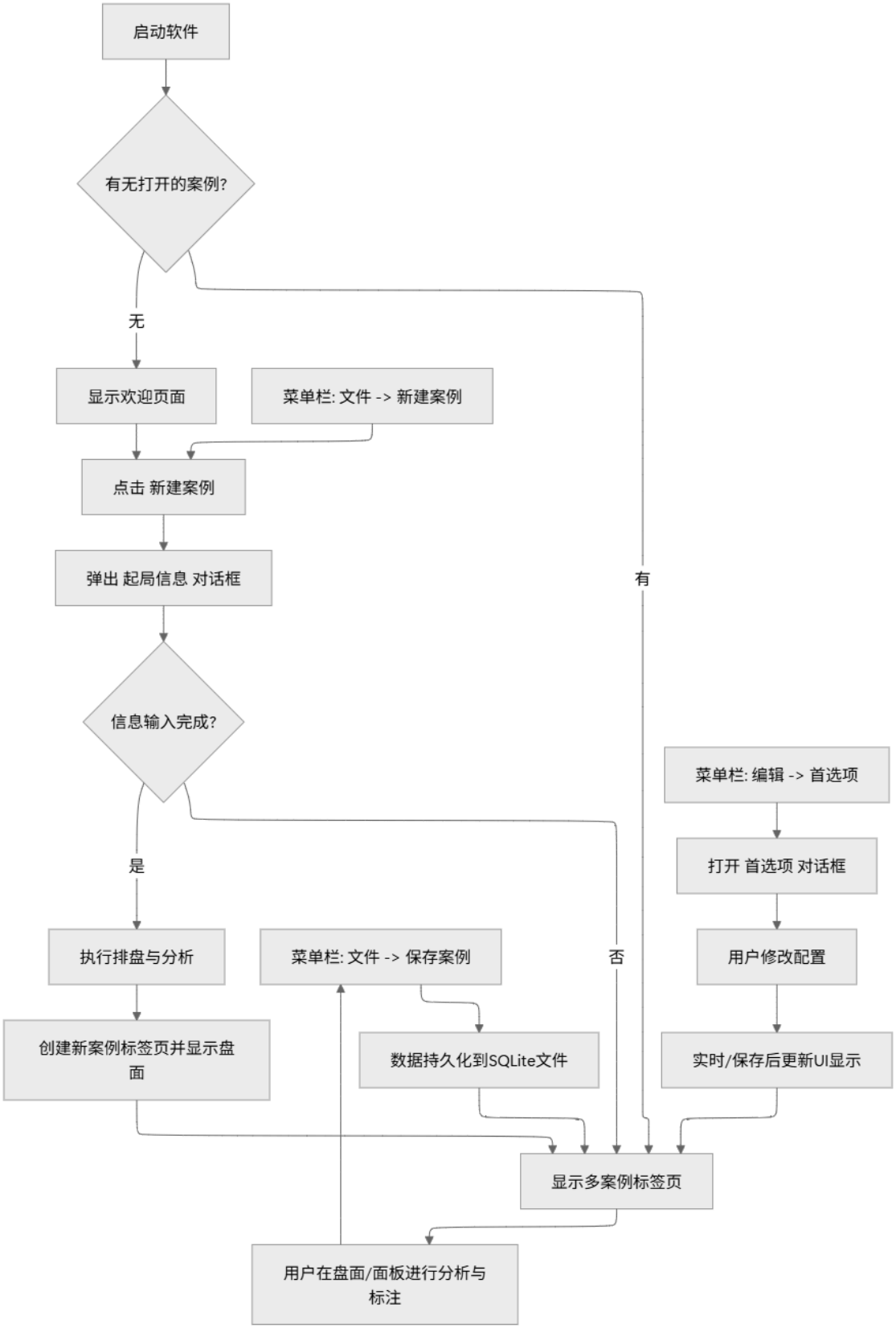
版本: 1.0 Alpha编制: 赵贵彬, Gemini (设计助理)日期: 2025年9月1日

1. 总体设计

1.1 业务流程

本软件的核心业务流程围绕“案例的创建、分析与管理”展开，旨在为用户提供一个从起局到分析再到复盘的完整工作流闭环。

主要业务处理流程图:

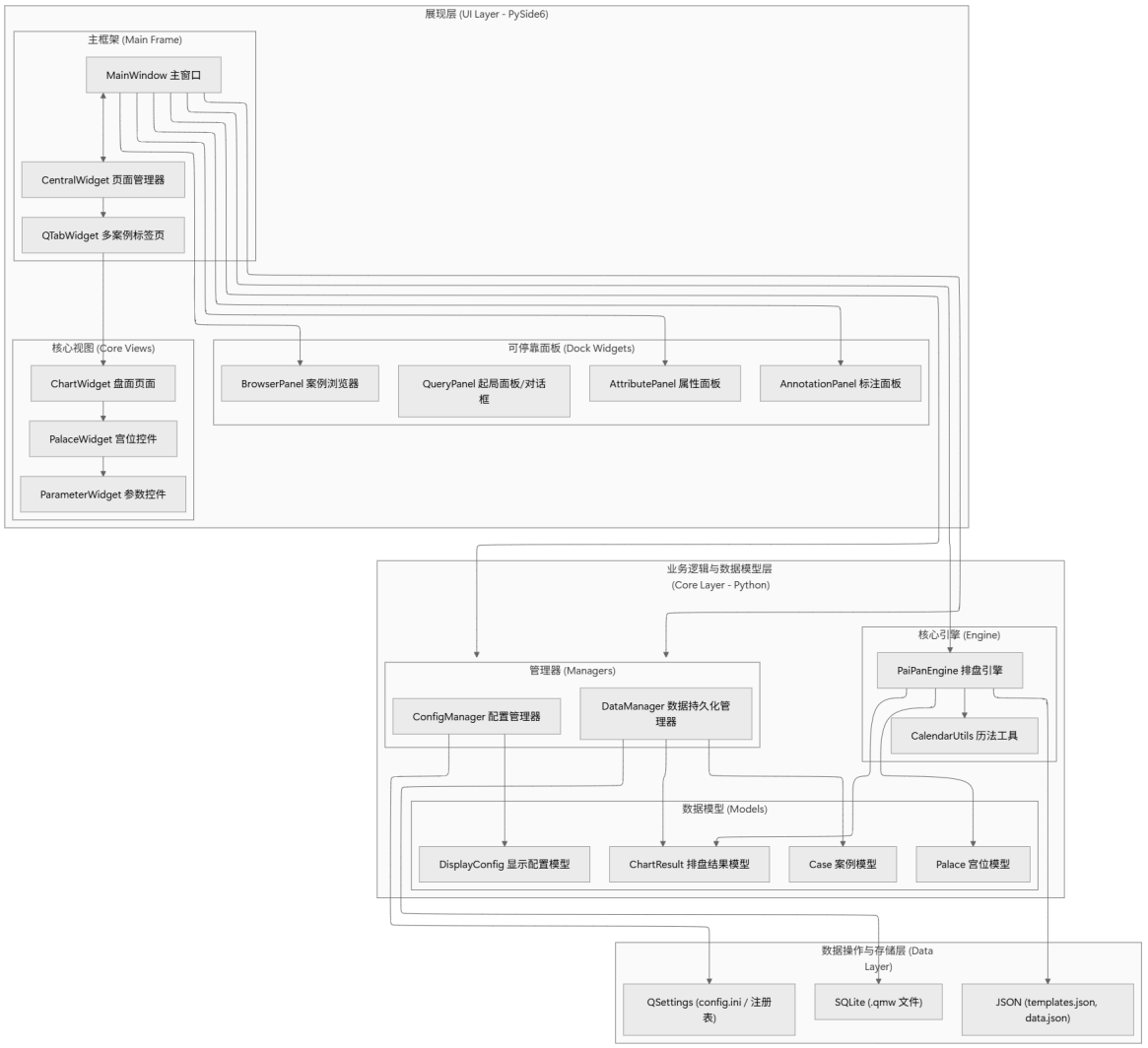

启动软件
有无打开的案例?
无
有
显示欢迎页面
菜单栏: 文件 -> 新建案例
点击 新建案例
弹出 起局信息 对话框
信息输入完成?
是
否
执行排盘与分析
菜单栏: 文件 -> 保存案例
菜单栏: 编辑 -> 首选项
打开 首选项 对话框
用户修改配置
创建新案例标签页并显示盘面
数据持久化到SQLite文件
实时/保存后更新UI显示
显示多案例标签页
用户在盘面/面板进行分析与标注

文字说明:
用户启动软件后，将看到欢迎页面或上一次打开的多案例页面。用户可通过“新建案例”按钮或菜单，触发“起局信息”对话框。在输入时间、年命等信息后，核心引擎将进行排盘和高级分析，并在主窗口的一个新标签页中，渲染出包含所有信息的奇门遁甲盘面。用户可以在盘面上进行交互式标注，并通过侧边的“属性面板”实时调整显示效果。所有的工作成果，包括盘面数据和用户标注，都可以通过“保存”功能，持久化到一个本地的`.qmw`（SQLite）文件中，以便未来加载和复盘。

1.2 总体架构

本软件采用基于**Python**和**PySide6**的桌面应用程序架构，设计上严格遵循**模型-视图-控制器 (MVC)的分层思想和组件化**的设计原则，确保代码的高内聚、低耦合与可扩展性。

软件系统结构图:



2. 程序描述

核心模块 1: `core` (业务逻辑与数据模型层)

- **2.1 功能**: 负责所有非UI的计算、分析和数据管理。是整个应用程序的大脑。
  - `PaiPanEngine`: 执行奇门遁甲的完整排盘算法，并进行六击、入墓、马星冲动等高级分析。
  - `models`: 定义`Case`, `ChartResult`, `Palace`等核心数据结构，确保数据在系统中的一致性和完整性。
  - `DataManager`: 使用SQLite封装所有案例的增删改查操作，实现数据的持久化。
  - `ConfigManager`: 使用`QSettings`封装所有用户配置的读取和保存。
- **2.2 性能**: 核心排盘和分析算法要求在1秒内完成，为用户提供流畅的“即时”排盘体验。数据查询通过为`ChartResult`预建索引来保证高性能。
- **2.3 输入项目**: 主要接收`datetime`对象（用于排盘）和`Case`对象（用于保存）。

核心模块 2: `ui` (展现层)

- **2.1 功能**: 负责所有用户界面的渲染和交互。
  - `MainWindow`: 作为应用程序的总框架，管理所有Dock面板和中央页面。
  - `ChartWidget`: 核心显示组件，负责将`ChartResult`对象可视化为用户可见的盘面。
  - `AnnotationPanel` & `AttributePanel`: 提供用户与程序交互的入口，用于进行标注和个性化设置。
  - `PreferencesDialog`: 提供一个集中的、多页面的设置中心。
- **2.2 性能**: UI响应要求无卡顿。盘面刷新（例如，因配置更改而触发的重绘）应在200毫秒内完成，以保证视觉上的流畅性。
- **2.3 输入项目**: 主要接收用户的鼠标键盘事件，并将它们转换为对`core`层的调用或信号发射。

3. 系统数据结构设计

3.1 逻辑结构设计

1. `Case` **(案例)**: 顶层数据结构，代表一个完整的用户工作单元。
   - `id` (标识符): 整数, 数据库主键。
   - `name` (名称): 字符串, 案例标题。

- `chart_result` (排盘结果): `ChartResult`对象, 包含所有客观盘面信息。
- `annotation_layers` (标注图层): 列表, 包含所有用户主观添加的标注信息。
- ... (其他元信息)

2. `ChartResult` **(排盘结果)**: 包含一次排盘的所有计算结果。
   - `si_zhu` (四柱): 字典。
   - `palaces` (九宫): `Palace`对象的列表。
   - `index` (索引): 字典, 用于快速查询参数位置。
   - ... (其他全局信息和分析结果)
3. `Palace` **(宫位)**: 包含一个宫位的所有信息。
   - `index` (索引): 整数, 1-9。
   - `tian_pan_stars` (天盘星): 字符串列表。
   - `di_pan_star` (地盘星): 字符串。
   - ... (其他天地盘神星门干信息)

3.2 物理结构设计要点

- **案例数据 (`.qmw`文件)**:
  - **存储要求**: 使用SQLite数据库文件，保证数据操作的事务性和安全性。
  - **访问方法**: 通过`core.data_manager`模块进行结构化SQL查询。
  - **存取单位**: 复杂对象（如`ChartResult`, `annotation_layers`）在存入数据库前，被序列化为JSON文本字符串，存放在TEXT类型的字段中，以保证未来数据结构的灵活性。
- **应用程序配置**:
  - **存储要求**: 使用`QSettings`进行存储，自动适应不同操作系统（Windows注册表, macOS/Linux .ini文件）。
  - **访问方法**: 通过`core.config_manager`模块进行键值对读写。
- **静态数据 (`data.json`, `templates.json`)**:
  - **存储要求**: 作为应用程序的只读资源，打包在最终的可执行文件中。
  - **访问方法**: 程序启动时一次性读取到内存，供`PaiPanEngine`等模块使用。