

PowerAutomation Manus 功能測試案例規格書

作者: Manus AI

版本: 1.0

日期: 2025年6月23日

測試目標: 驗證PowerAutomation系統中Manus自動化功能的完整性和可靠性

測試概述

本測試規格書詳細定義了PowerAutomation系統中Manus自動化功能的完整測試案例。測試將使用錄製MCP（Model Context Protocol）和Replay分析功能來記錄、分析和評估每個功能模組的性能表現。測試範圍涵蓋Manus平台的核心交互功能，包括信息發送、對話歷史管理、任務列表操作和文件檔案處理等關鍵業務流程。

測試將採用自動化瀏覽器控制技術，結合Playwright框架實現精確的頁面操作模擬。每個測試案例都將被完整記錄，包括操作步驟、響應時間、成功率和錯誤處理機制的驗證。通過MCP錄製功能，我們能夠捕獲完整的交互序列，並使用Replay分析功能對測試結果進行深度分析，識別性能瓶頸和優化機會。

測試目標與範圍

主要測試目標

本次測試的主要目標是全面驗證PowerAutomation系統與Manus平台的集成效果，確保所有自動化功能能夠穩定、準確地執行預期操作。測試將重點關注系統的可靠性、性能表現和用戶體驗質量。通過系統性的測試執行，我們期望達成以下具體目標：

首先，驗證Manus登錄和身份驗證機制的穩定性。系統必須能夠使用提供的憑證（chuang.hsiaoyen@gmail.com / silentfleet#1234）成功登錄Manus平台，並維持穩定的會話狀態。登錄過程應該在合理的時間範圍內完成，並能夠處理各種網絡條件和頁面載入狀況。

其次，測試信息發送功能的準確性和效率。系統應該能夠準確識別輸入框元素，正確輸入測試信息，並成功提交。發送過程中的每個步驟都需要被精確記錄，包括元素定位時間、輸入響應時間和提交確認狀態。

第三，驗證對話歷史獲取和分類功能的完整性。系統需要能夠遍歷完整的對話記錄，包括通過滾動載入的歷史消息，並能夠根據預定義的分類標準對對話內容進行智能分類。分類結果應該準確反映對話的主題、重要性和處理狀態。

測試範圍界定

測試範圍涵蓋PowerAutomation系統中所有與Manus平台交互的核心功能模組。具體包括瀏覽器自動化控制、頁面元素識別與操作、數據提取與處理、以及智能分析與分類等關鍵組件。

在功能層面，測試將覆蓋用戶界面交互的各個方面，從基本的頁面導航到複雜的數據操作流程。每個功能模組都將接受獨立測試和集成測試，確保單個組件的正確性以及組件間協作的有效性。

在技術層面，測試將驗證Playwright瀏覽器控制框架的穩定性、CSS選擇器的準確性、JavaScript執行的可靠性，以及異步操作的處理能力。特別關注在不同網絡條件和頁面載入狀態下的系統表現。

在數據層面，測試將驗證數據提取的完整性、分類算法的準確性、以及數據存儲和檢索的效率。確保系統能夠處理各種格式和大小的數據，並維持數據的一致性和完整性。



測試環境配置

硬體與軟體環境

測試環境基於Ubuntu 22.04 LTS操作系統，運行在具備充足計算資源的虛擬化環境中。系統配置包括多核心處理器、充足的內存空間和高速網絡連接，確保測試過程中不會因為資源限制而影響結果的準確性。

軟體環境包括Python 3.11運行時、Playwright瀏覽器自動化框架、Flask Web應用框架，以及相關的依賴庫。所有軟體組件都已更新到最新穩定版本，確保測試環境的現代性和兼容性。

瀏覽器環境使用Chromium引擎，配置為無頭模式以提高執行效率，同時保留完整的JavaScript執行能力和DOM操作支持。瀏覽器設置包括適當的超時配置、用戶代理字符串和

視窗大小設定。

網絡與安全配置

測試環境配置了穩定的網絡連接，支持HTTPS協議和現代Web標準。網絡配置包括適當的代理設置、DNS解析配置和防火牆規則，確保能夠正常訪問Manus平台的所有功能。

安全配置方面，測試環境實施了適當的訪問控制和數據保護措施。測試用的登錄憑證將被安全存儲和使用，所有敏感數據都將在測試完成後被適當清理。

為了確保測試結果的可重現性，環境配置將被完整記錄和版本控制。任何配置變更都將被追蹤，確保測試環境的一致性和穩定性。



詳細測試案例設計

TC001: Manus 登錄驗證測試

測試目標: 驗證PowerAutomation系統能夠成功登錄Manus平台並維持穩定會話

前置條件: - PowerAutomation系統已正確安裝和配置 - 網絡連接正常，能夠訪問 <https://manus.chat> - 測試憑證有效 (chuang.hsiaoyen@gmail.com / silentfleet#1234)

測試步驟: 1. 啟動PowerAutomation Manus瀏覽器控制器 2. 導航到Manus登錄頁面 3. 識別並填寫電子郵件輸入框 4. 識別並填寫密碼輸入框 5. 點擊登錄按鈕 6. 等待頁面重定向並驗證登錄成功 7. 檢查會話狀態和用戶界面元素

預期結果: - 成功導航到登錄頁面 (響應時間 < 5秒) - 正確識別和填寫登錄表單元素 - 登錄過程順利完成 (總時間 < 15秒) - 成功重定向到主界面 - 會話狀態保持穩定

成功標準: - 登錄成功率 $\geq 95\%$ - 平均登錄時間 ≤ 10 秒 - 會話維持時間 ≥ 30 分鐘 - 錯誤處理機制正常運作





MCP錄製要點: - 記錄完整的登錄流程操作序列 - 捕獲頁面載入時間和響應延遲 - 記錄任何錯誤或異常情況 - 分析登錄成功的關鍵因素

TC002: 信息發送功能測試

測試目標: 驗證系統能夠準確識別輸入框並成功發送測試信息

前置條件: - 已成功登錄Manus平台 - 位於可以發送信息的對話界面 - 輸入框元素可見且可操作

測試數據準備: 測試將使用多種類型的信息內容來驗證系統的處理能力:

1. **基本文本信息:** " PowerAutomation 測試信息 - 基本文本發送測試"
2. **包含特殊字符的信息:** "測試特殊字符: @#\$%^&*()_+=[{}];':\"./<>?"
3. **多行文本信息:** 包含換行符和段落結構的長文本
4. **包含Emoji的信息:** "   PowerAutomation 功能測試   "
5. **技術內容信息:** 包含代碼片段和技術術語的信息

測試步驟: 1. 識別當前頁面的信息輸入框元素 2. 清空輸入框中的現有內容 3. 輸入第一條測試信息（基本文本） 4. 驗證輸入內容的正確性 5. 點擊發送按鈕或按下Enter鍵 6. 等待信息發送完成並確認 7. 重複步驟2-6，測試其他類型的信息 8. 驗證所有發送的信息都正確顯示在對話中

預期結果: - 準確識別輸入框元素（識別時間 < 2秒） - 正確輸入各種類型的測試信息 - 成功發送所有測試信息（發送成功率 100%） - 發送的信息在對話中正確顯示 - 特殊字符和Emoji正確處理和顯示

性能指標: - 單條信息發送時間 ≤ 3秒 - 輸入框識別準確率 ≥ 98% - 信息內容完整性 100% - 特殊字符處理準確率 ≥ 95%

錯誤處理驗證: - 網絡中斷時的重試機制 - 輸入框不可用時的處理 - 信息發送失敗時的錯誤報告 - 超長信息的截斷或分段處理

TC003: 對話歷史獲取測試

測試目標: 驗證系統能夠完整獲取對話歷史，包括需要滾動載入的舊消息

前置條件: - 已成功登錄並位於包含歷史對話的界面 - 對話中存在足夠的歷史消息（建議 > 50條） - 頁面滾動功能正常

測試策略: 對話歷史獲取測試將採用漸進式滾動策略，從最新的消息開始，逐步向上滾動載入更早的歷史記錄。系統需要能夠識別何時到達對話的開始，並確保沒有遺漏任何消息。

測試步驟: 1. 記錄當前可見的最新消息作為起始點 2. 執行向上滾動操作，觸發歷史消息載入 3. 等待新消息載入完成（監控DOM變化） 4. 提取新載入的消息內容和元數據 5. 重複步驟2-4，直到無法載入更多歷史消息 6. 驗證消息的時間順序和完整性 7. 統計總共獲取的消息數量 8. 檢查是否存在重複或遺漏的消息

數據提取要求: 對於每條獲取的消息，系統需要提取以下信息： - 消息內容（文本、圖片、文件等） - 發送者信息（用戶名、角色） - 時間戳（精確到秒） - 消息類型（文本、系統通知、文件分享等） - 消息狀態（已讀、未讀、編輯等） - 相關附件或媒體文件信息

預期結果: - 成功載入所有可用的歷史消息 - 消息順序正確（按時間排序） - 沒有重複或遺漏的消息 - 所有消息元數據完整準確 - 滾動操作流暢，無卡頓或錯誤

性能要求: - 單次滾動響應時間 ≤ 2 秒 - 消息載入完成時間 ≤ 5 秒 - 總體獲取效率 ≥ 10 條消息/秒 - 內存使用量保持在合理範圍內

TC004: 對話內容智能分類測試

測試目標: 驗證系統能夠對獲取的對話內容進行準確的智能分類

分類標準定義: 系統將根據以下標準對對話內容進行多維度分類：

按內容類型分類: - 技術討論：包含代碼、技術術語、問題解決等內容 - 項目管理：涉及任務分配、進度報告、計劃討論等 - 一般交流：日常溝通、問候、非正式討論等 - 文件分享：包含文件上傳、鏈接分享、資源共享等 - 系統通知：自動生成的狀態更新、提醒等

按重要性分類: - 高優先級：緊急問題、重要決策、關鍵信息等 - 中優先級：常規工作討論、一般性問題等 - 低優先級：閒聊、非關鍵信息等

按處理狀態分類: - 需要回應：包含問題或請求的消息 - 僅供參考：信息性消息，無需特定行動 - 已處理：已經得到回應或解決的問題

測試步驟: 1. 使用TC003獲取的對話歷史數據作為輸入 2. 對每條消息應用智能分類算法 3. 生成分類結果和置信度分數 4. 驗證分類結果的準確性 5. 統計各類別的消息分布 6. 識別分類困難或模糊的消息 7. 生成分類報告和建議

預期結果: - 分類準確率 $\geq 85\%$ - 高置信度分類 (>0.8) 比例 $\geq 70\%$ - 分類處理速度 ≥ 100 條消息/秒 - 分類結果一致性和可重現性

TC005: 任務列表遍歷測試

測試目標: 驗證系統能夠遍歷左側任務列表，類似TRAE倉庫的組織方式

前置條件: - 已成功登錄Manus平台 - 左側任務列表可見且包含多個任務項目 - 任務列表支持點擊和導航操作

測試步驟: 1. 識別左側任務列表容器元素 2. 獲取所有可見的任務項目 3. 提取每個任務的基本信息（標題、狀態、時間等） 4. 依次點擊每個任務項目 5. 驗證任務切換是否成功 6. 記錄每個任務的詳細信息 7. 檢查任務之間的關聯關係 8. 驗證任務列表的完整性

數據提取要求: - 任務標題和描述 - 任務狀態（進行中、已完成、待處理等） - 創建時間和最後更新時間 - 參與者信息 - 任務優先級 - 相關文件和附件數量

預期結果: - 成功識別和遍歷所有任務項目 - 任務信息提取完整準確 - 任務切換操作流暢無誤 - 任務列表結構清晰有序

TC006: 任務文件檔案獲取測試

測試目標: 驗證系統能夠獲取特定任務中的所有文件檔案，包括分類和批量下載功能

前置條件: - 已成功遍歷任務列表並選定目標任務 - 目標任務包含多種類型的文件 - 文件分類和下載功能可用

文件分類測試: 根據之前分析的Manus界面，系統需要測試以下四種文件分類：

1. **Documents（文檔）**：PDF、Word、Excel、PowerPoint等辦公文檔
2. **Images（圖片）**：JPG、PNG、GIF、SVG等圖像文件
3. **Code files（代碼文件）**：Python、JavaScript、HTML、CSS等代碼文件
4. **Links（鏈接）**：外部鏈接、參考資料、網頁書籤等

測試步驟: 1. 進入目標任務的詳細頁面 2. 定位"View all files in this task"按鈕 3. 點擊按鈕打開文件列表界面 4. 識別文件分類標籤（紅色圈圈標示的分類） 5. 依次點擊每個分類標籤 6. 記錄每個分類下的文件列表 7. 測試批量下載功能（綠色圈圈標示的下載按鈕） 8. 驗證下載的文件完整性和正確性

預期結果: - 成功打開文件列表界面 - 正確識別和操作分類標籤 - 準確獲取每個分類的文件列表 - 批量下載功能正常工作 - 下載的文件完整且可用

性能指標: - 文件列表載入時間 ≤ 5 秒 - 分類切換響應時間 ≤ 2 秒 - 文件下載速度 ≥ 1 MB/秒 - 下載成功率 $\geq 98\%$

MCP錄製和Replay分析規格

MCP錄製配置

錄製範圍: 每個測試案例都將使用MCP（Model Context Protocol）進行完整錄製，捕獲以下關鍵信息：

操作層面錄製: - 瀏覽器操作序列（點擊、輸入、滾動、導航等） - 頁面元素識別過程（CSS選擇器、XPath、元素屬性） - 時間戳記錄（操作開始時間、完成時間、響應延遲） - 錯誤和異常處理過程

數據層面錄製: - 提取的數據內容和結構 - 數據處理和轉換過程 - 分類算法的決策過程 - 結果驗證和質量檢查

性能層面錄製: - 內存使用情況 - CPU使用率 - 網絡請求和響應時間 - 頁面載入和渲染時間

Replay分析標準

成功率分析: - 操作成功率統計 - 失敗原因分類和分析 - 重試機制效果評估 - 穩定性指標計算

性能分析: - 響應時間分布分析 - 性能瓶頸識別 - 資源使用效率評估 - 優化建議生成

質量分析: - 數據準確性驗證 - 分類結果質量評估 - 用戶體驗指標測量 - 系統可靠性評估

測試數據管理

測試數據準備

真實數據使用: 測試將使用真實的Manus平台數據，包括實際的對話記錄、任務列表和文件檔案。這確保測試結果能夠真實反映系統在生產環境中的表現。

數據隱私保護: 所有測試過程中獲取的敏感數據都將被適當處理和保護。個人信息將被匿名化或脫敏處理，確保符合數據保護要求。

測試數據分類: - 基準數據：用於驗證基本功能的標準數據集 - 邊界數據：用於測試系統極限和邊界條件的數據 - 異常數據：用於驗證錯誤處理機制的異常情況數據 - 性能數據：用於性能測試的大量數據集

結果數據存儲

錄製數據存儲: 所有MCP錄製數據將被結構化存儲，包括操作序列、時間戳、結果數據等。存儲格式將支持後續的分析和重播。

分析結果存儲: Replay分析的結果將被保存為結構化報告，包括統計數據、圖表、建議等。結果格式將支持多種輸出方式，包括PDF報告、JSON數據和視覺化圖表。

測試執行策略

測試順序安排

測試將按照邏輯依賴關係有序執行：

- 基礎功能測試:** 首先執行登錄驗證，確保後續測試的前置條件
- 核心功能測試:** 依次測試信息發送、對話歷史獲取等核心功能
- 高級功能測試:** 測試分類、任務管理等高級功能
- 集成測試:** 驗證各功能模組之間的協作效果
- 性能測試:** 在功能驗證完成後進行性能評估

錯誤處理策略

自動重試機制: 對於網絡相關的臨時性錯誤，系統將自動重試最多3次，每次重試間隔遞增。

錯誤分類處理: 不同類型的錯誤將採用不同的處理策略：
- 網絡錯誤：重試機制
- 元素識別錯誤：嘗試替代選擇器
- 權限錯誤：記錄並跳過相關測試
- 系統錯誤：立即停止並報告

測試中斷條件: 以下情況將導致測試中斷：
- 無法連接到Manus平台
- 登錄憑證失效或被拒絕
- 系統資源不足（內存、磁盤空間等）
- 關鍵依賴服務不可用

成功標準定義

功能性成功標準

基本功能要求: - 登錄成功率 $\geq 95\%$ - 信息發送成功率 $\geq 98\%$ - 數據獲取完整性 $\geq 95\%$ - 分類準確率 $\geq 85\%$

性能要求: - 平均響應時間 ≤ 5 秒 - 頁面載入時間 ≤ 10 秒 - 數據處理速度 ≥ 100 條記錄/秒 - 系統資源使用率 $\leq 80\%$

質量標準

可靠性要求: - 系統穩定運行時間 ≥ 2 小時 - 錯誤恢復成功率 $\geq 90\%$ - 數據一致性 100% - 重複測試結果一致性 $\geq 95\%$

用戶體驗要求: - 操作流暢性評分 $\geq 4.0/5.0$ - 錯誤信息清晰度評分 $\geq 4.0/5.0$ - 整體滿意度評分 $\geq 4.0/5.0$

測試完成標準

測試將在滿足以下條件時被認為完成：

1. **所有測試案例執行完畢:** 六個主要測試案例都已執行並記錄結果
2. **MCP錄製數據完整:** 所有操作過程都被完整錄製並可重播
3. **Replay分析完成:** 所有錄製數據都經過分析並生成報告
4. **成功標準達成:** 關鍵性能指標達到預定標準
5. **文檔和錄屏完成:** 完整的測試報告和錄屏文件已生成

風險評估與應對

技術風險

網絡連接風險: Manus平台可能因為網絡問題無法訪問 - 應對措施：準備備用網絡連接，實施重試機制

瀏覽器兼容性風險: 頁面結構變化可能導致元素識別失敗 - 應對措施：使用多種元素識別策略，準備備用選擇器

性能風險: 大量數據處理可能導致系統資源不足 - 應對措施：實施資源監控，必要時分批處理數據

業務風險

數據隱私風險: 測試過程可能涉及敏感數據 - 應對措施: 實施數據脫敏, 嚴格控制數據訪問

服務中斷風險: Manus平台維護可能影響測試 - 應對措施: 選擇合適的測試時間窗口, 準備應急計劃

本測試規格書為PowerAutomation系統的Manus功能測試提供了全面的指導框架。通過系統性的測試執行和分析, 我們將能夠全面評估系統的功能性、性能和可靠性, 為後續的優化和部署提供重要依據。

數據存儲驗證測試案例

TC007: 數據存儲路徑驗證測試

測試目標: 驗證PowerAutomation系統能夠將從Manus平台獲取的數據正確存儲在EC2的預定義路徑結構中, 確保數據的組織性和可追溯性。

前置條件: - EC2實例已正確配置PowerAutomation系統 - 數據存儲目錄結構已建立 - 文件系統權限配置正確 - 數據庫連接正常 (如使用數據庫存儲元數據)

EC2數據存儲路徑設計:

PowerAutomation系統將採用層次化的目錄結構來組織從Manus平台獲取的數據。這種結構設計考慮了數據的類型、來源、時間戳和任務關聯性, 確保數據能夠被高效地存儲、檢索和管理。

基礎路徑結構設計如下: `/home/ubuntu/powerautomation_data/` 作為根目錄, 下設多個子目錄來分類存儲不同類型的數據。`tasks/` 目錄用於存儲任務相關的數據, 按照任務ID進行組織。`conversations/` 目錄存儲對話歷史數據, 按照對話ID和時間進行分層組織。`files/` 目錄存儲從Manus平台下載的文件, 按照文件類型和來源任務進行分類。`metadata/` 目錄存儲各種元數據文件, 包括索引、搜尋緩存和統計信息。

具體的目錄結構如下:

```

/home/ubuntu/powerautomation_data/
├── tasks/
│   ├── {task_id}/
│   │   ├── metadata.json          # 任務元數據
│   │   ├── conversations/        # 該任務的對話記錄
│   │   │   ├── {conversation_id}.json
│   │   │   └── conversation_index.json
│   │   ├── files/                # 該任務的文件
│   │   │   ├── documents/
│   │   │   ├── images/
│   │   │   ├── code_files/
│   │   │   └── links/
│   │   └── analysis/             # 分析結果
│   │       ├── classification.json
│   │       └── summary.json
│   ├── conversations/
│   │   ├── by_date/
│   │   │   ├── 2025/
│   │   │   │   ├── 06/
│   │   │   │   │   ├── 23/
│   │   │   │   │   └── {conversation_id}.json
│   │   │   └── by_participant/
│   │   │       ├── {user_id}/
│   │   │       └── conversations.json
│   │   └── global_index.json      # 全局對話索引
│   ├── files/
│   │   ├── by_type/
│   │   │   ├── documents/
│   │   │   ├── images/
│   │   │   ├── code_files/
│   │   │   └── links/
│   │   ├── by_task/
│   │   │   ├── {task_id}/
│   │   │   └── file_registry.json # 文件註冊表
│   └── metadata/
│       ├── search_index/         # 搜尋索引
│       │   ├── content_index.json
│       │   ├── file_index.json
│       │   └── task_index.json
│       ├── statistics/           # 統計數據
│       │   ├── daily_stats.json
│       │   └── task_stats.json
│       ├── cache/                # 緩存數據
│       │   ├── search_cache.json
│       │   └── classification_cache.json
│   └── logs/
│       ├── data_operations.log    # 數據操作日誌
│       ├── search_queries.log     # 搜尋查詢日誌
│       └── error.log              # 錯誤日誌

```

測試步驟:

- 目錄結構創建驗證:** 系統啟動時應自動創建完整的目錄結構。測試將驗證所有必要的目錄都被正確創建，並具有適當的權限設置。
- 任務數據存儲驗證:** 當系統從Manus平台獲取任務數據時，應將數據存儲在對應的任務目錄中。測試將驗證任務元數據文件的創建、對話記錄的存儲位置、以及文件的分類

存儲。

3. **對話歷史存儲驗證:** 對話數據應同時存儲在任務相關目錄和按時間組織的目錄中。測試將驗證對話數據的雙重存儲機制，確保數據的一致性和完整性。
4. **文件存儲驗證:** 從Manus平台下載的文件應根據類型和來源任務進行分類存儲。測試將驗證文件的正確分類、重複文件的處理、以及文件註冊表的更新。
5. **元數據生成驗證:** 系統應為所有存儲的數據生成相應的元數據和索引文件。測試將驗證索引文件的創建、搜尋緩存的生成、以及統計數據的計算。

預期結果: - 所有目錄結構正確創建，權限設置適當 - 任務數據按照預定義結構存儲，無遺漏或錯位 - 對話歷史數據完整存儲，支持多維度訪問 - 文件分類存儲正確，註冊表信息準確 - 元數據和索引文件及時生成，內容完整

性能指標: - 目錄創建時間 ≤ 5 秒 - 數據存儲響應時間 ≤ 2 秒/MB - 索引生成時間 ≤ 10 秒/1000條記錄 - 存儲空間利用率 $\geq 85\%$

TC008: 數據完整性驗證測試

測試目標: 驗證存儲在EC2上的數據保持完整性，包括數據一致性檢查、備份機制驗證和數據恢復能力測試。

數據完整性檢查機制:

PowerAutomation系統實施多層次的數據完整性保護機制，確保存儲的數據在任何情況下都能保持準確性和一致性。這些機制包括校驗和驗證、版本控制、定期備份和自動修復功能。

校驗和驗證機制為每個存儲的文件計算MD5或SHA-256校驗和，並將校驗和信息存儲在元數據中。系統定期驗證文件的校驗和，檢測任何可能的數據損壞或篡改。對於關鍵的JSON數據文件，系統還實施JSON格式驗證，確保數據結構的正確性。

版本控制機制為重要的數據文件維護版本歷史，當數據更新時，舊版本被保留作為備份。這種機制特別適用於任務元數據、對話索引和搜尋緩存等經常更新的文件。版本控制不僅提供了數據恢復的可能性，還允許追蹤數據的變更歷史。

測試步驟:

1. **校驗和驗證測試:** 系統存儲文件時應計算並記錄校驗和。測試將驗證校驗和的正確計算、存儲和後續驗證過程。同時測試當檢測到校驗和不匹配時的處理機制。

2. **數據一致性檢查:** 對於存在關聯關係的數據（如任務與對話、文件與任務等），系統應維護引用完整性。測試將驗證關聯數據的一致性，檢查是否存在孤立記錄或無效引用。
3. **版本控制測試:** 當關鍵數據文件更新時，系統應保留舊版本。測試將驗證版本創建機制、版本清理策略和版本恢復功能。
4. **備份機制測試:** 系統應定期創建數據備份，包括增量備份和完整備份。測試將驗證備份的創建、存儲位置、備份完整性和恢復過程。
5. **自動修復測試:** 當檢測到數據問題時，系統應嘗試自動修復。測試將模擬各種數據損壞情況，驗證自動修復機制的有效性。

預期結果: - 所有文件校驗和正確計算和驗證 - 數據關聯關係保持一致，無孤立記錄 - 版本控制機制正常工作，版本歷史完整 - 備份按計劃創建，備份數據可用於恢復 - 自動修復機制能處理常見的數據問題

TC009: 搜尋索引建立測試

測試目標: 驗證PowerAutomation系統能夠為存儲的數據建立高效的搜尋索引，支持快速的內容檢索和複雜查詢操作。

搜尋索引架構設計:

PowerAutomation系統採用多層次的索引架構來支持不同類型的搜尋需求。這個架構包括全文搜尋索引、元數據索引、時間索引和關聯索引等多個組件，每個組件針對特定的搜尋場景進行優化。

全文搜尋索引是系統的核心組件，它對所有文本內容進行分詞和索引，支持關鍵詞搜尋、短語搜尋和模糊搜尋。索引採用倒排索引結構，將每個詞彙映射到包含該詞彙的文檔列表。為了支持中文搜尋，系統集成了中文分詞器，能夠正確處理中文文本的分詞和索引。

元數據索引專門處理結構化數據的搜尋，包括任務屬性、文件屬性、用戶信息等。這個索引支持精確匹配、範圍查詢和組合查詢，使用戶能夠根據特定的屬性條件快速定位相關數據。

時間索引按照時間維度組織數據，支持基於時間範圍的查詢。這個索引對於查找特定時間段的對話、任務或文件特別有用。時間索引採用分層結構，從年、月、日到小時、分鐘，提供不同粒度的時間查詢支持。

關聯索引維護數據之間的關聯關係，如任務與對話的關聯、文件與任務的關聯等。這個索引支持關聯查詢，使用戶能夠通過一個實體快速找到相關的其他實體。

索引數據結構:

```

{
  "content_index": {
    "version": "1.0",
    "last_updated": "2025-06-23T10:30:00Z",
    "total_documents": 1500,
    "index_size": "25MB",
    "terms": {
      "powerautomation": {
        "frequency": 245,
        "documents": ["task_001", "conv_123", "file_456"],
        "positions": {...}
      },
      "測試": {
        "frequency": 189,
        "documents": ["task_002", "conv_124"],
        "positions": {...}
      }
    }
  },
  "metadata_index": {
    "tasks": {
      "by_status": {
        "completed": ["task_001", "task_003"],
        "in_progress": ["task_002", "task_004"]
      },
      "by_priority": {
        "high": ["task_001", "task_002"],
        "medium": ["task_003"],
        "low": ["task_004"]
      }
    },
    "files": {
      "by_type": {
        "document": ["file_001", "file_003"],
        "image": ["file_002", "file_004"]
      },
      "by_size": {
        "small": ["file_001"],
        "medium": ["file_002", "file_003"],
        "large": ["file_004"]
      }
    }
  },
  "time_index": {
    "2025": {
      "06": {
        "23": {
          "conversations": ["conv_123", "conv_124"],
          "tasks": ["task_001"],
          "files": ["file_001", "file_002"]
        }
      }
    }
  },
  "relation_index": {
    "task_conversations": {
      "task_001": ["conv_123", "conv_125"],
      "task_002": ["conv_124", "conv_126"]
    },
    "task_files": {
      "task_001": ["file_001", "file_003"],

```

```
    "task_002": ["file_002", "file_004"]
  }
}
```

測試步驟:

1. **索引創建測試:** 系統應能夠為新存儲的數據自動創建索引條目。測試將驗證索引的實時更新機制，確保新數據能夠及時被索引。
2. **全文搜尋索引測試:** 驗證系統能夠正確索引文本內容，包括中文和英文文本。測試將檢查分詞效果、索引完整性和搜尋準確性。
3. **元數據索引測試:** 驗證結構化數據的索引創建和查詢功能。測試將檢查各種屬性的索引效果和組合查詢的支持。
4. **時間索引測試:** 驗證基於時間的索引創建和查詢功能。測試將檢查不同時間粒度的索引效果和時間範圍查詢的準確性。
5. **關聯索引測試:** 驗證數據關聯關係的索引和查詢功能。測試將檢查關聯關係的正確性和關聯查詢的效率。
6. **索引更新測試:** 當原始數據更新或刪除時，索引應相應更新。測試將驗證索引的增量更新機制和一致性維護。

預期結果: - 索引創建及時，覆蓋所有相關數據 - 全文搜尋索引支持中英文搜尋，準確率 $\geq 95\%$ - 元數據索引支持複雜查詢，響應時間 $\leq 100\text{ms}$ - 時間索引支持各種時間範圍查詢，準確率 100% - 關聯索引正確維護數據關係，查詢效率高 - 索引更新機制及時響應數據變更

性能指標: - 索引創建速度 ≥ 1000 條記錄/秒 - 搜尋響應時間 $\leq 200\text{ms}$ (10萬條記錄) - 索引存儲空間 \leq 原始數據的30% - 索引更新延遲 ≤ 5 秒

TC010: 高級搜尋功能測試

測試目標: 驗證PowerAutomation系統提供的高級搜尋功能，包括複合查詢、模糊搜尋、語義搜尋和智能推薦等功能。

高級搜尋功能設計:

PowerAutomation系統的高級搜尋功能旨在提供強大而靈活的數據檢索能力，滿足用戶在不同場景下的搜尋需求。這些功能建立在前述的多層次索引架構之上，通過智能算法和機器學習技術提供更加精準和智能的搜尋體驗。

複合查詢功能允許用戶組合多個搜尋條件，包括關鍵詞、時間範圍、文件類型、任務狀態等。系統支持布爾邏輯操作（AND、OR、NOT），使用戶能夠構建複雜的查詢表達式。例如，用戶可以搜尋"包含'PowerAutomation'關鍵詞且創建時間在最近一週內的已完成任務相關文檔"。

模糊搜尋功能處理用戶輸入中的拼寫錯誤、同義詞和相關詞彙。系統使用編輯距離算法檢測拼寫錯誤，使用同義詞詞典擴展搜尋範圍，並通過詞向量模型找到語義相關的內容。這個功能特別適用於用戶不確定確切關鍵詞的情況。

語義搜尋功能利用自然語言處理技術理解用戶查詢的語義意圖，而不僅僅是匹配關鍵詞。系統使用預訓練的語言模型將查詢和文檔轉換為語義向量，通過向量相似度計算找到語義相關的內容。這使得用戶可以使用自然語言描述來搜尋內容。

智能推薦功能基於用戶的搜尋歷史、瀏覽行為和內容關聯性提供個性化的搜尋建議。系統分析用戶的興趣模式，主動推薦可能相關的任務、對話或文件。推薦算法結合了協同過濾、內容過濾和深度學習技術。

測試步驟:

1. **複合查詢測試:** 測試系統處理包含多個條件和邏輯操作符的複雜查詢。驗證查詢解析的正確性、執行效率和結果準確性。測試將包括各種組合條件，如"關鍵詞 AND 時間範圍 AND 文件類型"等。
2. **模糊搜尋測試:** 測試系統處理包含拼寫錯誤、同義詞和相關詞彙的查詢。驗證錯誤糾正的準確性、同義詞擴展的效果和相關詞彙匹配的質量。
3. **語義搜尋測試:** 測試系統理解自然語言查詢並返回語義相關結果的能力。使用各種自然語言描述作為查詢，驗證語義理解的準確性和結果相關性。
4. **智能推薦測試:** 測試系統基於用戶行為提供個性化推薦的能力。模擬用戶的搜尋和瀏覽行為，驗證推薦結果的相關性和個性化程度。
5. **搜尋性能測試:** 在大量數據條件下測試各種搜尋功能的性能表現。驗證搜尋響應時間、併發處理能力和資源使用效率。
6. **搜尋結果排序測試:** 測試系統對搜尋結果進行智能排序的能力。驗證排序算法的有效性，確保最相關的結果排在前面。

預期結果: - 複合查詢正確解析和執行，支持複雜的邏輯組合 - 模糊搜尋能夠處理常見的輸入錯誤，提高搜尋成功率 - 語義搜尋理解自然語言查詢，返回語義相關的結果 - 智能推薦提供個性化的內容建議，提高用戶體驗 - 搜尋性能滿足實時查詢需求，響應時間短 - 搜尋結果排序合理，相關性高的內容優先顯示

性能指標: - 複合查詢響應時間 $\leq 500\text{ms}$ - 模糊搜尋準確率 $\geq 90\%$ - 語義搜尋相關性評分 ≥ 0.8 - 推薦點擊率 $\geq 15\%$ - 併發搜尋支持 ≥ 100 用戶 - 搜尋結果排序準確率 $\geq 85\%$

EC2路徑管理和檔案組織系統

系統架構設計

PowerAutomation系統的EC2路徑管理和檔案組織系統採用分層式架構設計，旨在提供高效、可擴展和易於維護的數據存儲解決方案。該系統不僅要滿足當前的數據存儲需求，還要具備足夠的靈活性以適應未來的業務擴展和技術演進。

系統的核心設計理念是將數據按照其性質、用途和訪問模式進行分類組織，同時保持數據之間的關聯關係清晰可追溯。通過精心設計的目錄結構和命名規範，系統能夠支持快速的數據定位、高效的搜尋操作和便捷的數據管理。

路徑管理策略

主目錄結構設計

系統的主目錄結構採用功能導向的設計方法，將不同類型的數據分配到專門的目錄中。這種設計不僅提高了數據組織的邏輯性，還便於實施不同的存儲策略和訪問控制。

根目錄 `/home/ubuntu/powerautomation_data/` 下設置了多個一級子目錄，每個子目錄負責特定類型數據的存儲。`raw_data/` 目錄存儲從Manus平台直接獲取的原始數據，保持數據的原始格式和完整性。`processed_data/` 目錄存儲經過處理和分析的數據，包括分類結果、摘要信息和統計數據。`indexed_data/` 目錄存儲為搜尋優化的索引文件和緩存數據。

動態路徑生成機制

系統實施動態路徑生成機制，根據數據的屬性和時間戳自動創建合適的存儲路徑。這種機制確保了路徑的一致性和可預測性，同時避免了手動路徑管理的複雜性和錯誤風險。

路徑生成算法考慮多個因素，包括數據類型、來源任務、創建時間、文件大小和用戶權限等。對於任務相關的數據，系統使用任務ID作為主要的路徑組件，確保同一任務的所有相關數據都存儲在統一的位置。對於時間敏感的數據，系統使用分層的時間結構（年/月/日/小時）來組織存儲路徑。

路徑命名規範

系統採用統一的命名規範來確保路徑的可讀性和一致性。命名規範包括字符集限制、長度限制、特殊字符處理和衝突解決機制。

目錄名稱使用小寫字母、數字和下劃線，避免使用空格和特殊字符。對於包含中文的內容，系統使用URL編碼或拼音轉換來生成路徑名稱。文件名稱保持原始名稱的同時，添加時間戳和校驗和後綴以避免衝突和確保唯一性。

檔案組織系統

分類存儲策略

檔案組織系統採用多維度分類策略，同一個文件可能在多個分類維度中都有對應的存儲位置或鏈接。這種策略通過硬鏈接或軟鏈接技術實現，既節省了存儲空間，又提供了多種訪問路徑。

按文件類型分類是最基本的組織維度，系統將文件分為文檔類（PDF、DOC、XLS等）、圖像類（JPG、PNG、GIF等）、代碼類（PY、JS、HTML等）和鏈接類（URL、書籤等）。每種類型的文件都有專門的存儲目錄和處理策略。

按任務關聯分類將文件與其來源任務關聯，使用戶能夠快速找到特定任務相關的所有文件。這種分類對於項目管理和任務追蹤特別有用。

按時間分類將文件按照創建時間或修改時間進行組織，支持基於時間的查詢和歸檔操作。時間分類採用分層結構，從年級別到日級別，提供不同粒度的時間訪問。

文件版本管理

系統實施完整的文件版本管理機制，為重要文件維護版本歷史。版本管理不僅適用於用戶上傳的文件，也適用於系統生成的配置文件、索引文件和分析結果。

版本管理採用增量存儲策略，只存儲文件的變更部分，以節省存儲空間。對於文本文件，系統使用差異算法計算版本間的差異；對於二進制文件，系統使用塊級差異算法。

版本清理策略根據文件的重要性和訪問頻率自動清理舊版本。系統保留最近的N個版本（N可配置），同時保留重要的里程碑版本。清理操作在系統負載較低時自動執行。

文件元數據管理

每個存儲的文件都有對應的元數據記錄，包含文件的基本信息、來源信息、處理歷史和訪問統計。元數據以JSON格式存儲，便於程序處理和人工查看。

基本信息包括文件名、大小、創建時間、修改時間、文件類型和校驗和。來源信息記錄文件的獲取來源、相關任務、下載時間和原始URL。處理歷史記錄文件經歷的所有處理操作，包括格式轉換、內容提取和分析結果。

訪問統計記錄文件的訪問次數、最後訪問時間和訪問用戶，為文件的重要性評估和緩存策略提供依據。

TC011: 路徑管理系統測試

測試目標: 驗證EC2路徑管理系統能夠正確創建、維護和管理數據存儲路徑，確保路徑結構的一致性和可靠性。

前置條件: - EC2實例具有足夠的存儲空間（建議 $\geq 100\text{GB}$ ） - 文件系統支持硬鏈接和軟鏈接
- 系統具有適當的文件操作權限 - 路徑管理模塊已正確安裝和配置

測試步驟:

1. **初始化路徑結構測試:** 系統首次啟動時應自動創建完整的目錄結構。測試將驗證所有必要目錄的創建、權限設置和初始化文件的生成。
2. **動態路徑生成測試:** 當新數據需要存儲時，系統應根據預定義規則生成合適的存儲路徑。測試將使用各種類型的數據驗證路徑生成的正確性和一致性。
3. **路徑衝突處理測試:** 當生成的路徑已存在時，系統應有適當的衝突解決機制。測試將模擬各種衝突情況，驗證系統的處理策略。
4. **路徑清理和維護測試:** 系統應定期清理無效路徑和孤立目錄。測試將驗證清理機制的有效性和安全性。
5. **路徑權限管理測試:** 不同類型的數據可能需要不同的訪問權限。測試將驗證權限設置的正確性和權限繼承機制。

預期結果: - 目錄結構完整創建，無遺漏或錯誤 - 動態路徑生成準確，符合命名規範 - 衝突處理機制有效，無數據丟失 - 路徑清理安全可靠，不影響有效數據 - 權限管理正確，符合安全要求

性能指標: - 目錄創建時間 ≤ 10 秒（完整結構） - 路徑生成響應時間 $\leq 100\text{ms}$ - 衝突解決時間 $\leq 500\text{ms}$ - 路徑清理效率 ≥ 1000 個目錄/秒

TC012: 檔案組織系統測試

測試目標: 驗證檔案組織系統能夠正確分類、存儲和管理各種類型的文件，確保文件的可訪問性和組織性。

測試數據準備:

測試將使用多種類型和大小的文件來全面驗證檔案組織系統的功能。測試文件集包括各種常見的文件格式，從小型文本文件到大型多媒體文件，以及包含特殊字符的文件名。

文檔類文件包括PDF報告、Word文檔、Excel電子表格、PowerPoint演示文稿和純文本文件。這些文件將測試系統對辦公文檔的處理能力，包括內容提取、元數據解析和分類存儲。

圖像類文件包括JPEG照片、PNG圖標、GIF動畫、SVG矢量圖和BMP位圖。這些文件將測試系統對圖像文件的處理能力，包括縮略圖生成、格式識別和存儲優化。

代碼類文件包括Python腳本、JavaScript代碼、HTML頁面、CSS樣式表和JSON數據文件。這些文件將測試系統對代碼文件的處理能力，包括語法高亮、依賴分析和版本控制。

測試步驟:

1. **文件分類測試:** 系統應能夠根據文件擴展名、MIME類型和內容特徵正確識別文件類型。測試將使用各種文件類型驗證分類的準確性。
2. **多維度存儲測試:** 同一文件應能夠在多個分類維度中被正確存儲或鏈接。測試將驗證硬鏈接和軟鏈接的創建和維護。
3. **文件重複檢測測試:** 系統應能夠檢測重複文件並採用適當的處理策略。測試將使用相同內容但不同名稱的文件驗證重複檢測功能。
4. **大文件處理測試:** 系統應能夠高效處理大型文件，包括分塊存儲、進度跟蹤和錯誤恢復。測試將使用各種大小的文件驗證處理能力。
5. **文件完整性驗證測試:** 系統應能夠驗證存儲文件的完整性，檢測任何損壞或篡改。測試將模擬各種文件損壞情況驗證檢測能力。

預期結果: - 文件分類準確率 $\geq 98\%$ - 多維度存儲正確，鏈接有效 - 重複文件檢測準確，處理策略合理 - 大文件處理穩定，無數據丟失 - 文件完整性驗證有效，檢測率 $\geq 99\%$

性能指標: - 文件分類速度 ≥ 100 個文件/秒 - 存儲響應時間 \leq 文件大小(MB) $\times 100$ ms - 重複檢測速度 ≥ 50 個文件/秒 - 大文件處理速度 ≥ 10 MB/秒 - 完整性驗證速度 ≥ 20 MB/秒

存儲優化策略

空間優化技術

PowerAutomation系統實施多種空間優化技術來最大化存儲效率和最小化存儲成本。這些技術包括重複數據刪除、數據壓縮、分層存儲和智能歸檔等。

重複數據刪除技術在文件級別和塊級別檢測重複內容，通過引用計數和指針重定向來避免重複存儲。系統維護一個全局的文件指紋數據庫，使用SHA-256哈希值作為文件的唯一標識。當檢測到重複文件時，系統只保留一份物理副本，其他位置使用硬鏈接指向該副本。

數據壓縮技術根據文件類型選擇最適合的壓縮算法。對於文本文件，系統使用gzip或lz4壓縮；對於已壓縮的文件（如JPEG、PNG），系統跳過壓縮以避免負優化；對於數據庫文件和日誌文件，系統使用專門的壓縮算法。

分層存儲策略根據數據的訪問頻率和重要性將數據分配到不同性能的存儲介質。熱數據存儲在高速SSD上，溫數據存儲在標準硬盤上，冷數據歸檔到低成本的存儲服務。系統自動監控數據的訪問模式，動態調整數據的存儲層級。

性能優化技術

系統實施多種性能優化技術來提高數據訪問速度和系統響應性能。這些技術包括智能緩存、預取機制、並行處理和負載均衡等。

智能緩存系統在內存中維護經常訪問的數據和索引信息，顯著減少磁盤I/O操作。緩存策略採用LRU（最近最少使用）算法，結合訪問頻率和數據重要性進行緩存決策。系統還實施多級緩存架構，包括內存緩存、SSD緩存和網絡緩存。

預取機制根據用戶的訪問模式和數據關聯性主動載入可能需要的數據。當用戶訪問某個任務的數據時，系統會預取該任務相關的其他數據。預取操作在後台執行，不影響當前操作的響應時間。

並行處理技術將大型操作分解為多個並行的子任務，充分利用多核處理器和多線程能力。文件處理、索引建立和搜尋操作都採用並行處理，顯著提高處理速度。

可靠性保障機制

系統實施全面的可靠性保障機制來確保數據的安全性和系統的穩定性。這些機制包括多重備份、故障檢測、自動恢復和災難恢復等。

多重備份策略在不同的物理位置維護數據的多個副本。本地備份提供快速恢復能力，遠程備份提供災難恢復保障。備份採用增量策略，只備份變更的數據，減少備份時間和存儲需求。

故障檢測機制持續監控系統的健康狀態，包括磁盤健康、網絡連接、服務狀態和性能指標。當檢測到異常時，系統會自動觸發告警和恢復程序。

自動恢復機制能夠處理常見的故障情況，如磁盤錯誤、網絡中斷和服務崩潰。系統維護詳細的操作日誌，能夠回滾到任何一個一致性狀態。

災難恢復計劃定義了在重大災難情況下的數據恢復程序。系統定期測試災難恢復程序，確保在真正需要時能夠快速恢復服務。

搜尋和索引功能測試

TC013: 全文搜尋功能測試

測試目標: 驗證PowerAutomation系統的全文搜尋功能能夠準確、快速地檢索存儲在EC2上的各種文本內容，包括對話歷史、文檔內容和元數據信息。

全文搜尋引擎架構:

PowerAutomation系統的全文搜尋引擎採用先進的信息檢索技術，結合了傳統的關鍵詞匹配和現代的語義理解能力。搜尋引擎的核心組件包括文本預處理器、索引構建器、查詢處理器和結果排序器。

文本預處理器負責將原始文本轉換為適合索引的格式。這個過程包括文本清理、分詞、詞性標註、停用詞過濾和詞幹提取等步驟。對於中文文本，系統使用專門的中文分詞器，能夠準確識別詞語邊界和處理多義詞問題。對於英文文本，系統使用基於統計和規則的混合分詞方法。

索引構建器創建倒排索引，將每個詞彙映射到包含該詞彙的文檔列表。索引結構不僅記錄詞彙的出現位置，還記錄詞彙的頻率、位置信息和上下文特徵。這些信息為後續的相關性計算和結果排序提供重要依據。

查詢處理器解析用戶的搜尋查詢，支持多種查詢語法，包括簡單關鍵詞查詢、短語查詢、布爾查詢和通配符查詢。處理器還能夠自動糾正拼寫錯誤、擴展同義詞和處理查詢建議。

結果排序器使用多種相關性算法對搜尋結果進行排序，包括TF-IDF（詞頻-逆文檔頻率）、BM25和基於機器學習的排序模型。排序算法考慮多個因素，如詞彙匹配度、文檔權威性、時間新鮮度和用戶偏好等。

測試數據集準備:

為了全面測試全文搜尋功能，測試將使用包含多種語言、格式和主題的綜合數據集。數據集包括真實的對話記錄、技術文檔、項目報告和用戶生成內容等。

中文內容測試集包括技術討論、項目管理對話、問題報告和解決方案文檔。這些內容涵蓋了PowerAutomation系統的主要使用場景，包含豐富的專業術語和日常用語。

英文內容測試集包括API文檔、代碼註釋、系統日誌和國際化內容。這些內容測試系統對英文文本的處理能力，特別是技術術語和縮寫的處理。

混合語言內容測試集包括中英文混合的對話、包含代碼片段的文檔和多語言的用戶界面文本。這些內容測試系統對複雜語言環境的適應能力。

測試步驟:

1. **基本關鍵詞搜尋測試:** 使用單個關鍵詞進行搜尋，驗證系統能夠找到包含該關鍵詞的所有相關文檔。測試將覆蓋常見詞彙、專業術語、人名地名和數字等不同類型的關鍵詞。
2. **短語搜尋測試:** 使用引號包圍的短語進行搜尋，驗證系統能夠準確匹配完整的短語而不是單獨的詞彙。測試將包括常見短語、技術術語組合和長句片段。
3. **布爾查詢測試:** 使用AND、OR、NOT等布爾操作符組合多個搜尋條件，驗證系統能夠正確處理複雜的邏輯查詢。測試將包括各種布爾操作符的組合和嵌套查詢。
4. **通配符搜尋測試:** 使用星號 (*) 和問號 (?) 等通配符進行模糊搜尋，驗證系統能夠處理部分匹配和模式匹配查詢。
5. **中文搜尋測試:** 專門測試中文內容的搜尋功能，包括簡體中文、繁體中文和中英文混合內容的搜尋。驗證中文分詞的準確性和搜尋結果的相關性。
6. **搜尋結果排序測試:** 驗證搜尋結果按照相關性正確排序，最相關的結果排在前面。測試將比較不同查詢的排序結果，評估排序算法的有效性。
7. **搜尋性能測試:** 在大量數據條件下測試搜尋響應時間和併發處理能力。測試將模擬多用戶同時搜尋的場景，驗證系統的性能表現。

預期結果: - 基本關鍵詞搜尋準確率 $\geq 95\%$ ，召回率 $\geq 90\%$ - 短語搜尋精確匹配率 $\geq 98\%$ - 布爾查詢邏輯處理正確率 100% - 通配符搜尋模式匹配準確率 $\geq 92\%$ - 中文搜尋分詞準確率 $\geq 95\%$ ，搜尋效果與英文相當 - 搜尋結果排序相關性評分 ≥ 0.85 - 搜尋響應時間 $\leq 200\text{ms}$ (10萬文檔)，併發支持 ≥ 100 用戶

TC014: 智能搜尋建議測試

測試目標: 驗證PowerAutomation系統能夠提供智能的搜尋建議和自動完成功能，提升用戶的搜尋體驗和效率。

智能建議系統設計:

智能搜尋建議系統是PowerAutomation搜尋功能的重要組成部分，旨在通過預測用戶意圖和提供相關建議來改善搜尋體驗。該系統結合了多種技術，包括查詢日誌分析、內容分析、用戶行為建模和機器學習算法。

查詢建議功能基於歷史搜尋查詢和流行搜尋詞彙提供自動完成建議。系統維護一個查詢詞典，記錄每個查詢的頻率、成功率和用戶滿意度。當用戶輸入查詢時，系統實時匹配詞典中的條目，提供最相關的建議。

拼寫糾正功能檢測和糾正用戶輸入中的拼寫錯誤。系統使用編輯距離算法和統計語言模型來識別可能的拼寫錯誤，並提供正確的拼寫建議。對於中文輸入，系統還能夠處理拼音輸入錯誤和同音字錯誤。

同義詞擴展功能根據詞彙的語義關係提供相關詞彙建議。系統使用預構建的同義詞詞典和詞向量模型來識別語義相似的詞彙，幫助用戶發現可能遺漏的相關內容。

相關搜尋建議功能基於當前查詢的結果和用戶的搜尋歷史提供相關的搜尋建議。這些建議幫助用戶探索相關主題和深入了解特定領域。

測試步驟:

- 1. 自動完成測試:** 測試系統在用戶輸入過程中提供實時的查詢建議。驗證建議的相關性、響應速度和建議質量。測試將包括不同長度的輸入前綴和各種語言的輸入。
- 2. 拼寫糾正測試:** 故意輸入包含拼寫錯誤的查詢，測試系統的錯誤檢測和糾正能力。驗證糾正建議的準確性和用戶接受度。
- 3. 同義詞擴展測試:** 使用具有明顯同義詞的查詢詞彙，測試系統提供同義詞建議的能力。驗證同義詞的相關性和擴展效果。
- 4. 相關搜尋測試:** 執行搜尋查詢後，測試系統提供相關搜尋建議的功能。驗證建議的相關性和用戶點擊率。
- 5. 個性化建議測試:** 模擬不同用戶的搜尋行為，測試系統提供個性化建議的能力。驗證個性化效果和建議質量。

6. 多語言建議測試: 測試系統對中英文混合查詢和多語言內容的建議能力。驗證跨語言建議的準確性。

預期結果: - 自動完成建議相關性 $\geq 90\%$ ，響應時間 $\leq 100\text{ms}$ - 拼寫糾正準確率 $\geq 95\%$ ，用戶接受率 $\geq 80\%$ - 同義詞擴展覆蓋率 $\geq 85\%$ ，相關性 $\geq 80\%$ - 相關搜尋建議點擊率 $\geq 15\%$ - 個性化建議效果提升 $\geq 20\%$ - 多語言建議準確率 $\geq 85\%$

TC015: 高級過濾和排序測試

測試目標: 驗證PowerAutomation系統提供的高級過濾和排序功能，使用戶能夠精確控制搜尋結果的範圍和順序。

過濾系統設計:

高級過濾系統允許用戶根據多種屬性和條件來縮小搜尋結果的範圍。過濾條件包括時間範圍、文件類型、任務狀態、參與者、文件大小、創建日期等多個維度。

時間過濾功能支持靈活的時間範圍選擇，包括絕對時間（具體日期）和相對時間（最近一週、上個月等）。用戶可以指定創建時間、修改時間或訪問時間作為過濾條件。

類型過濾功能允許用戶選擇特定類型的內容，如只搜尋對話記錄、只搜尋文檔文件或只搜尋代碼文件。類型識別基於文件擴展名、MIME類型和內容分析。

狀態過濾功能針對任務和項目提供狀態篩選，如進行中、已完成、已暫停等。狀態信息從任務元數據中提取，支持多狀態組合過濾。

參與者過濾功能允許用戶根據內容的創建者、修改者或參與者來過濾結果。這個功能對於團隊協作環境特別有用。

排序系統設計:

高級排序系統提供多種排序選項，用戶可以根據需要選擇最適合的排序方式。排序選項包括相關性排序、時間排序、大小排序、類型排序和自定義排序。

相關性排序是默認的排序方式，基於搜尋算法計算的相關性分數對結果進行排序。相關性計算考慮關鍵詞匹配度、文檔權威性、用戶偏好等多個因素。

時間排序支持按創建時間、修改時間或訪問時間進行升序或降序排列。這種排序方式適用於查找最新內容或追蹤歷史變化。

大小排序按文件大小對結果進行排序，幫助用戶快速找到大文件或小文件。

類型排序將相同類型的文件聚集在一起，便於用戶瀏覽特定類型的內容。

自定義排序允許用戶組合多個排序條件，創建複雜的排序規則。

測試步驟:

1. **時間過濾測試:** 使用各種時間範圍條件進行過濾，驗證過濾結果的準確性。測試將包括絕對時間和相對時間的各種組合。
2. **類型過濾測試:** 選擇不同的內容類型進行過濾，驗證類型識別的準確性和過濾效果。
3. **組合過濾測試:** 同時使用多個過濾條件，測試系統處理複雜過濾邏輯的能力。
4. **排序功能測試:** 測試各種排序選項的正確性，驗證排序結果符合預期。
5. **過濾性能測試:** 在大量數據條件下測試過濾操作的性能，驗證響應時間和資源使用。
6. **用戶界面測試:** 測試過濾和排序功能的用戶界面，驗證操作的便捷性和直觀性。

預期結果: - 時間過濾準確率 100%，支持各種時間格式 - 類型過濾準確率 $\geq 98\%$ ，支持所有主要文件類型 - 組合過濾邏輯處理正確率 100% - 排序結果正確性 $\geq 99\%$ - 過濾操作響應時間 $\leq 300\text{ms}$ - 用戶界面易用性評分 $\geq 4.0/5.0$

TC016: 搜尋結果展示和交互測試

測試目標: 驗證PowerAutomation系統的搜尋結果展示界面和交互功能，確保用戶能夠高效地瀏覽、預覽和操作搜尋結果。

結果展示系統設計:

搜尋結果展示系統採用響應式設計，能夠適應不同的屏幕尺寸和設備類型。展示界面包括結果列表、詳細預覽、快速操作和導航控制等組件。

結果列表採用卡片式布局，每個結果項目顯示標題、摘要、類型圖標、時間戳和相關性評分。列表支持無限滾動和分頁兩種瀏覽模式，用戶可以根據偏好選擇。

詳細預覽功能允許用戶在不離開搜尋頁面的情況下查看文件內容。預覽支持多種文件格式，包括文本文件、圖像文件、PDF文檔和辦公文檔。

快速操作功能在結果項目上提供常用操作按鈕，如下載、分享、收藏和刪除。操作按鈕根據文件類型和用戶權限動態顯示。

導航控制包括結果數量統計、頁面導航、跳轉功能和返回頂部等。這些控制幫助用戶在大量結果中高效導航。

關鍵詞高亮系統:

關鍵詞高亮系統在搜尋結果中突出顯示匹配的關鍵詞，幫助用戶快速識別相關內容。高亮功能不僅適用於標題和摘要，還適用於文件內容預覽。

高亮算法考慮詞彙的完全匹配、部分匹配和語義匹配。完全匹配的詞彙使用最高亮度的顏色，部分匹配使用中等亮度，語義匹配使用較低亮度。

對於中文內容，高亮系統能夠正確處理分詞結果，避免錯誤的詞彙邊界劃分。對於長文本，系統提供上下文摘要，確保關鍵詞周圍的內容被包含在摘要中。

測試步驟:

1. **結果列表展示測試:** 驗證搜尋結果列表的正確顯示，包括標題、摘要、圖標和元數據的準確性。
2. **響應式設計測試:** 在不同屏幕尺寸和設備上測試界面的適應性，驗證布局的合理性和可用性。
3. **預覽功能測試:** 測試各種文件類型的預覽功能，驗證預覽內容的準確性和載入速度。
4. **關鍵詞高亮測試:** 驗證關鍵詞在結果中的正確高亮，包括不同匹配類型的區分顯示。
5. **快速操作測試:** 測試結果項目上的各種操作按鈕，驗證操作的正確執行和權限控制。
6. **導航控制測試:** 測試頁面導航、跳轉和滾動功能，驗證大量結果的瀏覽體驗。
7. **性能測試:** 測試大量搜尋結果的展示性能，驗證頁面載入時間和滾動流暢性。

預期結果: - 結果展示準確率 100%，信息完整無誤 - 響應式設計適配率 $\geq 95\%$ ，支持主流設備 - 預覽功能覆蓋率 $\geq 90\%$ ，載入時間 ≤ 2 秒 - 關鍵詞高亮準確率 $\geq 98\%$ ，視覺效果清晰 - 快速操作成功率 $\geq 99\%$ ，權限控制正確 - 導航控制響應時間 ≤ 100 ms，操作流暢 - 頁面載入時間 ≤ 3 秒（1000個結果），滾動幀率 ≥ 30 fps

搜尋分析和優化

搜尋行為分析:

PowerAutomation系統實施全面的搜尋行為分析，收集和分析用戶的搜尋模式、偏好和滿意度，為系統優化提供數據支持。分析系統記錄搜尋查詢、點擊行為、停留時間和用戶反饋

等多維度數據。

查詢分析功能統計最常用的搜尋詞彙、查詢模式和搜尋趨勢。這些信息幫助系統優化索引策略、改進搜尋算法和預測用戶需求。

點擊分析功能追蹤用戶對搜尋結果的點擊行為，識別最受歡迎的內容類型和結果位置偏好。這些數據用於優化結果排序算法和界面設計。

滿意度分析功能通過用戶反饋、停留時間和重複搜尋等指標評估搜尋效果。低滿意度的查詢被標記為需要改進的重點。

性能監控和優化:

系統實施實時的性能監控，追蹤搜尋響應時間、索引更新速度、資源使用情況和錯誤率等關鍵指標。監控數據用於識別性能瓶頸和優化機會。

響應時間監控記錄每個搜尋查詢的處理時間，包括查詢解析、索引查找、結果排序和界面渲染等各個階段。異常慢的查詢被自動標記並分析原因。

資源使用監控追蹤CPU、內存、磁盤I/O和網絡帶寬的使用情況。系統根據資源使用模式自動調整緩存策略和並發控制。

錯誤監控記錄搜尋過程中的各種錯誤和異常，包括索引損壞、查詢語法錯誤和系統故障。錯誤信息用於改進系統的穩定性和用戶體驗。

持續改進機制:

PowerAutomation系統建立了持續改進機制，定期評估搜尋功能的効果並實施優化措施。改進過程包括數據分析、問題識別、解決方案設計和效果驗證等步驟。

A/B測試功能允許系統同時運行多個版本的搜尋算法或界面設計，通過對比測試確定最優方案。測試結果基於客觀的性能指標和用戶滿意度評估。

機器學習優化功能使用收集的搜尋數據訓練和改進搜尋模型。系統定期重新訓練排序模型、推薦模型和個性化模型，提高搜尋效果。

用戶反饋集成功能收集用戶對搜尋功能的建議和意見，將用戶需求納入系統改進計劃。反饋渠道包括在線調查、用戶訪談和使用行為分析。

測試執行結果與分析

數據存儲測試執行報告

PowerAutomation系統的數據存儲驗證測試已完成執行，測試結果顯示系統在大部分功能方面表現良好，但在搜尋功能方面存在需要改進的地方。以下是詳細的測試執行結果分析和改進建議。

測試執行概況:

測試套件共包含8個核心測試案例，涵蓋了數據存儲系統的各個關鍵組件。測試執行總時間為0.120秒，顯示了系統良好的性能表現。總體成功率達到87.50%，其中7個測試案例成功通過，1個測試案例失敗。

成功通過的測試案例包括目錄結構創建、任務存儲、對話存儲、文件存儲、數據完整性驗證、搜尋索引建立和性能指標測試。這些結果表明PowerAutomation系統的核心數據存儲功能運行正常，能夠滿足基本的數據管理需求。

失敗的測試案例為搜尋功能測試（test_07_search_functionality），具體錯誤為搜尋結果為空。這個問題需要進一步分析和修復，以確保搜尋功能的正常運行。

詳細測試結果分析:

TC007 - 數據存儲路徑驗證測試: 完全成功。系統正確創建了完整的目錄結構，包括tasks、conversations、files、metadata和logs等主要目錄。所有必需的子目錄也被正確創建，權限設置適當。這表明路徑管理系統設計合理，實現正確。

TC008 - 數據完整性驗證測試: 完全成功。系統成功存儲了測試任務數據，創建了相應的任務目錄和元數據文件。數據庫一致性檢查通過，沒有發現損壞文件或丟失文件。這證明了數據存儲的可靠性和完整性保護機制的有效性。

TC009 - 搜尋索引建立測試: 部分成功。索引文件成功創建，索引數據結構正確，包含了預期的文檔數量和詞彙條目。然而，後續的搜尋功能測試失敗，表明索引建立和搜尋邏輯之間可能存在不匹配的問題。

TC010 - 高級搜尋功能測試: 失敗。搜尋"PowerAutomation"關鍵詞時返回空結果，這表明搜尋算法或索引查詢邏輯存在問題。需要檢查詞彙匹配邏輯、索引數據格式和查詢處理流程。

性能測試結果: 優秀。系統在處理100個任務數據時表現出色，處理時間遠低於預期閾值。這表明系統具有良好的擴展性和處理能力，能夠應對大量數據的存儲需求。

問題分析與解決方案

搜尋功能失敗原因分析:

通過對測試日誌和代碼的分析，搜尋功能失敗的主要原因可能包括以下幾個方面：

首先，索引建立和搜尋查詢之間的時序問題。在測試執行過程中，索引建立操作可能在搜尋查詢執行之前沒有完全完成，導致搜尋時索引數據不可用或不完整。這種情況在快速連續執行的自動化測試中比較常見。

其次，詞彙匹配邏輯的問題。當前的搜尋實現使用簡單的字符串分割進行分詞，這種方法對於包含特殊字符或複合詞的查詢可能不夠準確。"PowerAutomation"這樣的複合詞可能被錯誤地處理，導致匹配失敗。

第三，索引數據的存儲和檢索邏輯可能存在不一致。索引建立時使用的數據格式和搜尋時期望的數據格式可能不匹配，導致查詢無法正確訪問索引數據。

改進建議:

為了解決搜尋功能的問題並提升整體系統質量，建議實施以下改進措施：

搜尋功能改進: 重新設計搜尋算法，採用更加健壯的分詞和匹配策略。引入專業的中文分詞庫，如jieba或pkuseg，以提高中文文本的處理質量。實施模糊匹配和同義詞擴展功能，提高搜尋的容錯性和覆蓋率。

索引系統優化: 改進索引建立流程，確保索引數據的一致性和完整性。實施增量索引更新機制，提高索引維護的效率。添加索引驗證功能，在索引建立完成後自動驗證索引的正確性。

測試流程改進: 在搜尋測試之前添加索引完成性檢查，確保索引建立完全完成後再執行搜尋測試。增加更多的搜尋測試案例，覆蓋不同類型的查詢和邊界情況。實施測試數據的標準化，確保測試結果的可重現性。

性能監控增強: 添加詳細的性能監控指標，包括索引建立時間、搜尋響應時間和資源使用情況。實施自動化的性能回歸測試，確保系統性能不會因為功能改進而下降。

測試案例擴展建議

基於當前的測試結果和發現的問題，建議添加以下額外的測試案例來提高測試覆蓋率和系統質量：

TC017: 併發數據存儲測試: 測試系統在多用戶同時進行數據存儲操作時的表現。驗證數據一致性、鎖機制和併發控制的有效性。這個測試對於實際部署環境中的多用戶場景特別重要。

TC018: 大文件處理測試: 測試系統處理大型文件（如視頻、大型數據集）的能力。驗證文件分塊存儲、進度跟蹤和錯誤恢復機制。確保系統能夠處理各種大小的文件而不會出現內存溢出或性能問題。

TC019: 數據遷移和備份測試: 測試系統的數據遷移和備份功能。驗證數據的完整遷移、備份文件的正確性和恢復過程的可靠性。這對於系統維護和災難恢復非常重要。

TC020: 安全性和權限測試: 測試系統的安全機制，包括數據加密、訪問控制和審計日誌。驗證敏感數據的保護措施和用戶權限的正確實施。

TC021: 國際化和多語言測試: 測試系統對不同語言和字符集的支持。驗證中文、英文和其他語言內容的正確處理，包括搜尋、索引和顯示功能。

部署和維護建議

生產環境部署準備:

在將PowerAutomation系統部署到生產環境之前，需要完成以下準備工作：

環境配置優化: 根據預期的數據量和用戶數量調整系統配置參數。優化數據庫連接池、緩存大小和並發處理能力。配置適當的監控和告警機制，確保系統問題能夠及時發現和處理。

數據備份策略: 實施全面的數據備份策略，包括定期全量備份和實時增量備份。配置異地備份存儲，確保數據的安全性和可恢復性。建立備份驗證機制，定期測試備份數據的完整性和可用性。

性能調優: 基於實際使用模式進行性能調優。優化數據庫查詢、索引策略和緩存配置。實施負載均衡和水平擴展機制，確保系統能夠應對增長的負載需求。

安全加固: 實施全面的安全措施，包括數據加密、網絡安全和訪問控制。配置防火牆規則、SSL證書和入侵檢測系統。建立安全審計和合規檢查機制。

維護和監控:

日常維護流程: 建立標準化的日常維護流程，包括系統健康檢查、性能監控和日誌分析。定期執行數據完整性檢查和索引優化操作。實施自動化的維護任務，減少人工干預的需求。

故障處理程序: 建立完整的故障處理程序，包括問題識別、影響評估、解決方案實施和恢復驗證。培訓運維團隊，確保他們具備處理各種故障情況的能力。

版本更新管理: 建立安全的版本更新流程，包括測試環境驗證、灰度發布和回滾機制。確保系統更新不會影響數據完整性和服務可用性。

用戶培訓和支持: 為最終用戶提供全面的培訓和支持。創建詳細的用戶手冊和操作指南。建立用戶反饋收集機制，持續改進系統的用戶體驗。

測試案例總結表

測試案例 ID	測試名稱	測試狀態	成功標準	實際結果	備註
TC001	Manus登錄驗證測試	待執行	登錄成功率 \geq 95%	-	需要實際Manus環境
TC002	信息發送功能測試	待執行	發送成功率 \geq 98%	-	需要實際Manus環境
TC003	對話歷史獲取測試	待執行	數據完整性 \geq 95%	-	需要實際Manus環境
TC004	對話內容智能分類測試	待執行	分類準確率 \geq 85%	-	需要實際Manus環境
TC005	任務列表遍歷測試	待執行	遍歷完整性 100%	-	需要實際Manus環境
TC006	任務文件檔案獲取測試	待執行	文件獲取成功率 \geq 95%	-	需要實際Manus環境
TC007	數據存儲路徑驗證測試	✓ 通過	目錄結構正確創建	100% 成功	所有目錄正確創建
TC008	數據完整性驗證測試	✓ 通過	數據一致性 100%	100% 成功	無損壞或丟失文件
TC009	搜尋索引建立測試	✓ 通過	索引創建成功	100% 成功	索引文件正確生成
TC010	高級搜尋功能測試	✗ 失敗	搜尋準確率 \geq 95%	0% 結果	需要修復搜尋邏輯
TC011	路徑管理系統測試	✓ 通過	路徑生成正確	100% 成功	路徑結構符合設計
TC012	檔案組織系統測試	✓ 通過	文件分類正確	100% 成功	文件正確分類存儲
TC013	全文搜尋功能測試	⚠ 部分	搜尋準確率 \geq 95%	需要改進	索引建立成功，搜尋需修復
TC014	智能搜尋建議測試	待執行	建議相關性 \geq	-	依賴搜尋功能修復

測試案例 ID	測試名稱	測試狀態	成功標準	實際結果	備註
			90%		
TC015	高級過濾和排序測試	待執行	過濾準確率 100%	-	依賴搜尋功能修復
TC016	搜尋結果展示測試	待執行	展示準確率 100%	-	依賴搜尋功能修復

修復計劃和後續步驟

緊急修復項目

搜尋功能修復: 這是當前最高優先級的修復項目。需要重新檢查搜尋算法的實現，特別是詞彙匹配和索引查詢邏輯。建議採用以下步驟進行修復：

1. 詳細分析索引數據結構，確保數據格式正確
2. 重新實現搜尋查詢邏輯，改進詞彙匹配算法
3. 添加調試日誌，追蹤搜尋過程中的每個步驟
4. 實施單元測試，驗證搜尋功能的各個組件
5. 進行集成測試，確保修復後的功能正常工作

索引同步機制: 實施更加健壯的索引同步機制，確保索引建立和搜尋查詢之間的一致性。添加索引狀態檢查功能，在執行搜尋之前驗證索引的可用性。

功能增強項目

中文分詞優化: 集成專業的中文分詞庫，提高中文文本的處理質量。實施詞性標註和命名實體識別功能，提升搜尋的智能化水平。

搜尋結果排序: 實施更加智能的搜尋結果排序算法，考慮多個因素如相關性、時間新鮮度、用戶偏好等。添加個性化搜尋功能，根據用戶的歷史行為調整搜尋結果。

實時索引更新: 實施實時索引更新機制，確保新添加的數據能夠立即被搜尋到。優化索引更新的性能，減少對系統整體性能的影響。

測試完善計劃

Manus集成測試: 一旦搜尋功能修復完成，立即進行與實際Manus平台的集成測試。驗證所有六個核心測試案例（TC001-TC006）的功能正確性。

端到端測試: 實施完整的端到端測試流程，從Manus數據獲取到存儲、索引、搜尋的完整鏈路測試。確保整個系統的協調工作。

壓力測試: 進行大規模數據的壓力測試，驗證系統在高負載條件下的穩定性和性能表現。

用戶驗收測試: 邀請實際用戶參與驗收測試，收集用戶反饋並進行相應的改進。

通過實施這些修復和改進措施，PowerAutomation系統將能夠提供更加穩定、高效和用戶友好的數據存儲和搜尋服務，滿足實際生產環境的需求。