**系統架設流程**

**階段一：需求確認與準備**

1. **環境需求確認**
   * 確認您擁有一組 **Google 帳號** (用於 GCP 和 Google Sheets) 及 **LINE Developers 帳號** (用於建立 LINE Bot)。
   * 確認您電腦上已安裝 **Python 3.10** 或更高版本，以及必要的套件管理工具如 pip。
2. **文件查閱**
   * 仔細閱讀專案結案報告，特別是「系統需求分析」、「部署及維護計畫」、「系統分析與設計」等章節，了解專案的全貌、功能、非功能需求、系統架構、資料庫結構等。
   * 瀏覽 GitHub 儲存庫 https://github.com/cynthiaiii4/TSCBot，熟悉程式碼結構和檔案內容。

**階段二：GCP 環境設定**

1. **建立 GCP 專案**
   * 登入 Google Cloud Console。
   * 建立一個新的 GCP 專案，例如命名為 tpo-pos-chatbot。
2. **啟用所需 API**
   * 在 GCP Console 中，導航至「API 和服務」>「已啟用的 API 和服務」。
   * 啟用以下 API：
     + Google Sheets API
     + Cloud Firestore API
     + Cloud Run Admin API
     + Cloud Logging API
     + Cloud Build API
3. **建立服務帳戶並生成金鑰**
   * 導航至「IAM 與管理」>「服務帳戶」。
   * 建立一個新的服務帳戶，例如命名為 tpo-pos-chatbot-sa。
   * 給予此服務帳戶必要的角色，例如：
     + Cloud Run 呼叫端 (Cloud Run Invoker)
     + Cloud Run 管理員 (Cloud Run Admin)
     + Firestore 使用者 (Firestore User)
     + Google Sheets API 編輯者 (Google Sheets API Editor)
     + 日誌寫入者 (Logs Writer)
   * 建立此服務帳戶的

**JSON 金鑰** ，並將其下載保存為

service\_account\_key.json。**請務必妥善保管此檔案，不可外洩，也勿上傳至公開儲存庫。**

**階段三：LINE Bot 設定**

1. **建立 LINE 專屬 Channel (通道)**
   * 登入 LINE Developers Console。
   * 建立一個新的 Provider，然後在其下建立一個新的 Messaging API Channel。
2. **獲取 Channel Access Token 和 Channel Secret**
   * 在您建立的 Channel 設定頁面中，找到並記下 **Channel Access Token** 和 **Channel Secret**，這些將用於程式碼的環境變數。
3. **設定 LINE 官方帳號的圖文選單**
   * 登入 LINE Official Account Manager (LINE 官方帳號管理頁面)。
   * 選擇對應的官方帳號。
   * 設定圖文選單：
     + 選擇「大型四格版型」 。
     + 上傳專案提供的單張四格圖 (banner.jpg 或自製一張) 。
     + 設定每個區塊 (A, B, C, D) 的「動作」：
       - A (知識寶典): 類型「文字」，文字內容「知識寶典」 。
       - B (查中油點數): 類型「文字」，文字內容「查中油點數」 。
       - C (通訊錄查詢): 類型「連結」，連結 URL 需指向您的通訊錄文件連結，例如

https://docs.google.com/spreadsheets/d/1fPV2FmIC6Sps8n\_...，動作標籤「通訊錄查詢」 。

* + - * D (熱門查詢): 類型「文字」，文字內容「熱門查詢」 。
    - 設定選單列顯示文字為「常用功能」 。

**階段四：Google Sheets 資料庫準備**

1. **複製專案範本 Google Sheet**
   * 打開專案文件中提供的 Google Sheet 結構範本連結 (例如：https://docs.google.com/spreadsheets/d/13ohhljNfKJQUdXEIUyENcwl8\_YH\_vvGqSXtIYsz5XNs/edit?usp)。
   * 建立一份副本到您自己的 Google Drive。
2. **授予服務帳戶編輯權限**
   * 打開您剛剛複製的 Google Sheet。
   * 點擊「分享」按鈕。
   * 將在 GCP 建立的服務帳戶電子郵件地址（例如：

tpo-pos-chatbot-sa@your-gcp-project-id.iam.gserviceaccount.com）新增為**編輯者** 。

1. **記錄 Google Sheet URL**
   * 複製您新建的 Google Sheet 的完整 URL，這將用於程式碼的環境變數。

**階段五：程式碼部署與設定**

1. **下載程式碼**
   * 從 GitHub 儲存庫 https://github.com/cynthiaiii4/TSCBot 下載完整的專案程式碼（通常是下載 ZIP 檔並解壓縮）。
2. **準備程式碼檔案**
   * 確認您有 tscbot.py (可能是 main.py 的變更版) 和 requirements.txt。
   * 將在 GCP 步驟中下載的 service\_account\_key.json 檔案放在與 tscbot.py 相同的目錄下。
3. **部署到 GCP Cloud Run**
   * 在 GCP Console 中，導航至「Cloud Run」。
   * 建立一個新的服務：
     + 選擇從原始碼部署。
     + 選擇您的專案程式碼目錄。
     + 執行階段選

**Python 3.10** 。

* + - 設定資源：記憶體至少

**4 GiB**，CPU 至少 **1** 。

* + - 要求逾時設定為

**300 秒** 。

* + - 設定執行個體下限為 **1** (避免冷啟動延遲，但會增加費用) 或 **0** (預設，會冷啟動)。
    - 啟用「啟動時CPU效能強化」 。
    - 選擇允許「未經驗證的叫用」以供 LINE Webhook 訪問。

1. **設定環境變數**
   * 在 Cloud Run 服務的「修訂版本」設定中，新增以下環境變數：
     + GOOGLESHEET\_URL：您在階段四複製的 Google Sheet URL 。
     + LINE\_BOT\_CHANNEL\_SECRET：您在階段三獲取的 Channel Secret 。
     + LINE\_BOT\_CHANNEL\_ACCESS\_TOKEN：您在階段三獲取的 Channel Access Token 。
     + ALLOWED\_DESTINATION：您的 LINE 官方帳號 User ID (可以在 LINE Developers Console 的 Channel 設定頁面中找到，通常在 Basic settings 的 Channel ID 下方) 。
     + GEMINI\_API\_KEY：您的 Gemini API 金鑰 (從 Google AI Studio 獲取) 。
     + FIRESTORE：將您 service\_account\_key.json 檔案的**所有內容複製貼上**到這個環境變數中，作為 JSON 字串 。

**確保是有效的 JSON 格式，不含換行符。**

1. **設定 LINE Developers Webhook URL**
   * 部署完成後，Cloud Run 會提供一個服務 URL。複製這個 URL。
   * 回到 LINE Developers Console，在您 Channel 的「Messaging API」設定頁面中，將這個 Cloud Run 服務 URL 貼到

**Webhook settings 的 Webhook URL** 欄位 。

* + 點擊「Verify」進行測試，確保 LINE 平台能正確連接到您的服務 。

**階段六：功能測試與驗證**

1. **基本問答測試**
   * 在 LINE 應用程式中，加入您的 LINE 官方帳號。
   * 嘗試發送簡單的問題，例如「你好」、「測試」等，看機器人是否有回應。
2. **功能按鈕測試**
   * 測試圖文選單上的按鈕，例如「知識寶典」、「查中油點數」、「熱門查詢」，驗證是否能正確顯示對應內容。
   * 點擊知識寶典下的分類和問題，驗證是否能獲取解決方案。
3. **數據記錄驗證**
   * 檢查您的 Google Sheet 中的「統計紀錄」和「回答」工作表，確認使用者問題和機器人回答是否被正確記錄。
   * 檢查 Firestore 中的 conversations 和 feedback collections，確認對話紀錄和回饋數據是否被正確儲存。
4. **優化問題精準度測試** (針對問題清單)
   * 針對您需要優化的10個問題，逐一在 LINE 機器人中提問，評估回答的精準度。
   * 根據結果，回顧並優化 Google Sheet 知識庫中的內容、同義詞，甚至考慮調整 retrieve\_top\_n 函式中的權重或閾值。

**階段七：專案完成與持續優化**

1. **資料庫內容優化與新增**
   * 開始解決「資料庫舊的內容雜亂，敘述不一，多無用字符，待統整」的問題，按照統一格式清洗資料。
   * 依計畫新增 **100 題**問題至資料庫。
2. **監控與迭代**
   * 定期查看 GCP Cloud Logging，監控機器人運行狀況和潛在錯誤。
   * 分析 Firestore 中的對話紀錄和回饋，了解使用者需求，持續優化知識庫和機器人回答邏輯。
   * 根據需求，考慮是否需要調整 BM25 和語意比對的權重，或探索更先進的 AI 模型。