

Yeung.chan 2025/07/18

**📘 SS.AI Customer Services Platform - System Design Specification**

**1. Introduction**

SS.AI 是由 Starchy Solution 開發的 WhatsApp 客戶服務平台，旨在為企業提供一個靈活、可視化且自動化的訊息溝通解決方案。平台整合了 Meta WhatsApp API、Elsa Workflow 工作流引擎、React 前端設計器、ZXing QR 解碼、Google Sheets 等功能，適用於查詢、審批、交付簽收、ERP 整合等場景。

產品核心目標包括：

* 建立圖像化的工作流程設定
* 整合訊息傳送與等待回覆的自動化機制
* 整合 QR 碼掃描與 ERP 外接系統資料查詢
* 使用者易於理解的外部整合方式（如 Excel 或 Google Sheets）
* 可擴展的模組化平台（如 NLP、BI）

**2. Key Features（核心功能）**

**2.1 WhatsApp Messaging Automation**

* 整合 Meta WhatsApp Business API，不依賴第三方平台（如 WATI）
* 支援下列自動化流程：
  + 發送訊息（Send Message）
  + 等待回覆（Wait for Reply）
  + 自動提醒（例如每天 5 點提醒未回覆客戶）
  + 跳至人工客服（如有問題或超時）
  + 篩選特定關鍵字進行下一步操作

**2.2 Graphical Workflow Designer（圖形化流程設計器）**

* 使用 Elsa Workflows 作為後端工作流程引擎
* 前端使用 React + Elsa Dashboard UI
* 支援拖拉式設計：
  + 建立 WhatsApp 操作節點
  + 設定條件分支（如依客戶回覆內容判斷流程）
  + 可視化每個流程狀態
* 儲存為流程定義 JSON，並儲存在資料庫中

**2.3 QR Code 圖像處理與解碼**

* 支援送貨員拍照傳送的圖片中自動尋找 QR Code 區塊
* 採用 ZXing.Net + System.Drawing.Common 解碼
* 即使圖片略有歪斜或拍攝品質不佳，仍能解出
* 從 QR code 中讀出送貨單號碼（Delivery Note No.）

**2.4 ERP 資料查詢／更新模組**

* 送貨單號碼解出後，可透過 API 查詢第三方系統（如 ERP）：
  + 查出詳細送貨資料（品項、數量等）
  + 核對是否已送達與簽收
  + 更新 ERP 狀態（例如已完成、已回傳簽名等）
* 如不使用 API，可支援：
  + Google Sheets 作為資料中介
  + Excel 上傳至 Google Drive，系統讀取其中內容回應
  + 將查詢結果更新回 Excel，再由客戶匯入其 ERP

**2.5 模板訊息管理與回覆**

* 建立訊息模板庫：
  + 分類管理（如訂單回覆、提醒通知等）
  + 可插入動態欄位（如客戶姓名、單號）
* 回覆訊息時支援快速套用模板
* 將每次通訊儲存於會話歷史，支援查閱與分析

若你準備好，可回覆「下一段」或「繼續」，我將發送第三部分：**3. Workflow Activity Types（流程節點類型）**。

**3. Workflow Activity Types（流程節點類型）**

為了靈活定義 WhatsApp 溝通與外部操作流程，系統設計了下列節點類型（Activity Types）供使用者在流程圖中選用：

**3.1 WhatsApp 專用節點**

| **節點名稱** | **說明** |
| --- | --- |
| **Send Message** | 傳送文字、模板或圖像訊息給指定聯絡人 |
| **Wait for Message** | 停在此節點直到收到對應用戶的訊息（可設定超時） |
| **Branch by Keyword** | 根據訊息內容（關鍵字或正則）跳轉不同流程 |
| **Forward to Agent** | 將會話轉交人工客服並暫停機器人處理 |
| **Send File** | 傳送 PDF、圖片、Excel 等附件 |
| **Receive File** | 接收並儲存客戶傳送之檔案，可觸發解析（如 QR code 圖） |

**3.2 資料處理與查詢節點**

| **節點名稱** | **說明** |
| --- | --- |
| **Decode QR Image** | 接收圖片後進行 QR Code 解碼，取出內容（如單號） |
| **Lookup ERP Data (API)** | 呼叫第三方 API 根據單號查詢詳細資訊 |
| **Lookup Google Sheet** | 根據條件查詢 Google Sheet（作為 ERP 替代） |
| **Write to Google Sheet** | 更新資料回 Google Sheet 表格 |
| **Update ERP Status** | 透過 API 更新 ERP 的處理狀態（如已簽收） |

**3.3 控制與輔助節點**

| **節點名稱** | **說明** |
| --- | --- |
| **Start / End** | 流程起點與終點 |
| **Delay / Wait** | 延遲指定時間後執行下一步（如 5 分鐘後提醒） |
| **Condition Gateway** | 條件判斷（如表格值、訊息內容）進行分流 |
| **Manual Input** | 提示人工輸入或後台操作 |
| **Log & Notify** | 寫入日誌／Email / Telegram / Line 推播通知 |
| **Execute SQL / C# Code** | 執行後台邏輯或自定程式段落（如查詢資料庫） |

📌 **說明：**

* 上述節點有部分 Elsa Workflow 原生已有（如 Start, End, Delay, Condition）
* 大部分 WhatsApp 或 ERP 操作節點需自訂開發（可使用 Elsa 的 Custom Activity 機制）
* 所有節點均支援設定輸入輸出參數，供流程其他節點引用

**4. Technology Stack（技術架構）**

**4.1 系統整體架構圖（簡要）**

css

複製編輯

[Client Browser / Mobile]

│

▼

[React Flow Designer Frontend] ←→ [SSO 登入：Google / Email]

│

▼

[ASP.NET Core API (Elsa Workflow Host)]

│

├─ [Elsa Persistence DB（PostgreSQL or SQL Server）]

├─ [Workflow Execution Engine]

├─ [Message Template Repository（Database）]

├─ [Google Sheet / ERP Connector]

├─ [Meta WhatsApp API Connector]

└─ [Logging, Monitoring, Notification Modules]

**4.2 前端技術（React）**

| **元件** | **說明** |
| --- | --- |
| **Elsa** | 視覺化流程編輯器（流程圖 UI） |
| **Ant Design / TailwindCSS** | UI 元件與樣式庫 |
| **React Google Login** | Google OAuth 整合登入 |
| **LocalStorage / Redux** | 快取流程草稿、本地狀態管理 |
| **流程圖物件定義** | 支援各 Activity Type 拖放與參數設定 UI（彈窗或表單） |

**4.3 後端技術（.NET C#）**

| **元件** | **說明** |
| --- | --- |
| **ASP.NET Core Web API** | API 與後台邏輯 |
| **Elsa Workflows 3.0** | Workflow host 引擎（自訂 Activity 節點） |
| **Entity Framework Core** | 資料庫 ORM 映射 |
| **ZXing.Net** | QR Code 解碼（若需） |
| **Google Sheets API SDK** | 讀寫 Google Sheet |
| **Meta WhatsApp API SDK** | 發送與接收訊息 |
| **OAuth2 Google Auth** | 登入與權限驗證 |
| **SignalR / WebSocket (選用)** | 即時訊息、通知同步至前端 |
| **NLog / Serilog** | Log 管理 |
| **Hangfire** | 後台排程與重試任務（如每日自動提醒） |

**4.4 資料儲存與擴展**

| **模組** | **說明** |
| --- | --- |
| **SQL Server** | 儲存 Workflow Definitions、執行紀錄、帳號權限等資料 |
| **Google Drive / Excel / API** | 支援 ERP 互動模擬與資料交換 |
| **可擴充 OAuth Provider** | 日後支援更多登入方式（如 AzureAD、LINE Login） |

✅ **備註：**

* 若需部署於中國大陸，須使用國內主機及備案、改用本地 OAuth 登入方案
* WhatsApp API 本身需 VPN 或港/星 gateway server 轉接
* 整體設計支援 SaaS 擴展，可作多租戶（Multi-tenant）設計

**5. Database Schema（資料庫設計）**

**5.1 系統用戶與登入**

| **Table Name** | **說明** |
| --- | --- |
| users | 用戶基本資料（email, name, login method, etc.） |
| user\_roles | 用戶角色定義（admin, operator, viewer 等） |
| login\_sessions | 登入紀錄與 token 儲存（可擴充為 SSO） |

**5.2 WhatsApp 對話訊息**

| **Table Name** | **說明** |
| --- | --- |
| chat\_sessions | WhatsApp 對話會話（會包含每個 wa\_id） |
| chat\_messages | 每則訊息紀錄（傳入/傳出訊息、時間戳、訊息內容） |
| chat\_media | 儲存圖片、PDF 等媒體訊息路徑（可搭配 CDN / GDrive） |

**5.3 Workflow 與節點執行紀錄（Elsa）**

可使用 Elsa 內建的 Elsa.Persistence.EntityFramework.Core schema 或自定表結構。

| **Table Name** | **說明** |
| --- | --- |
| workflow\_definitions | 視覺流程圖的定義 JSON（包含節點、邏輯條件） |
| workflow\_instances | 每次流程執行實例 |
| activity\_instances | 各節點的執行紀錄 |
| workflow\_variables | 執行中的上下文變數值 |

**5.4 Message Template 庫**

| **Table Name** | **說明** |
| --- | --- |
| message\_templates | 預設訊息格式，如文字、圖片、清單、按鈕等（JSON） |
| template\_category | 模板分類（例：驗證、訂單查詢、售後等） |
| template\_usage\_logs | 模板使用次數與時間（作為統計用途） |

**5.5 QRCode 解碼結果記錄（選用）**

| **Table Name** | **說明** |
| --- | --- |
| qrcode\_logs | 儲存圖片處理結果，包含圖片來源、解析內容、對應 session |
| delivery\_links | 若成功解析送貨單號，紀錄對應 ERP 訂單資訊 |

**5.6 Google Sheets 連接**

| **Table Name** | **說明** |
| --- | --- |
| sheet\_connections | 記錄每個工作表的 URL、範圍、欄位對應方式（Excel → 系統） |
| sheet\_pull\_logs | 拉取紀錄、資料筆數、錯誤資訊 |

**5.7 ERP / API 整合設定（開發中）**

| **Table Name** | **說明** |
| --- | --- |
| external\_systems | 第三方 ERP 或 API 設定，如 URL、Token、方式 |
| external\_mapping | 定義欄位對應（從 workflow variables → 外部 API 欄位） |

✅ 所有關鍵資料表皆可設定租戶欄位（如 tenant\_id），日後擴展為多租戶架構。

**6. Workflow 核心流程與節點類型設計**

本平台使用 **Elsa Workflows** 作為後台工作流引擎，前端提供圖形化設計工具（React + Elsa Designer），使用者可自由拉動節點、配置條件、設定訊息內容，打造客製化客服流程。

**6.1 Workflow Activity 類型定義（核心類別）**

| **節點名稱** | **功能說明** | **是否需自行開發** |
| --- | --- | --- |
| **Start** | 流程起始點，可由 Webhook 或排程觸發 | ❌（Elsa 內建） |
| **Send WhatsApp Text** | 發送純文字訊息 | ✅ |
| **Send WhatsApp Media** | 發送圖片 / PDF 等媒體訊息（如送貨單圖、產品圖） | ✅ |
| **Send WhatsApp Template** | 使用已預存的 WhatsApp HSM template 內容傳送 | ✅ |
| **Wait For Reply** | 等待用戶回覆（文字或圖片），可設定 timeout 時間 | ✅ |
| **If Condition** | 根據變數條件分支流程（如判斷文字是否包含某字） | ❌（Elsa 內建） |
| **Switch** | 根據變數多選一執行分支（如流程分成驗證、查詢、投訴） | ❌（Elsa 內建） |
| **Extract QR From Image** | 接收圖片並從中解析 QR Code 內容 | ✅ |
| **Query Database** | 從系統資料庫查詢內容（如根據送貨單號查詢明細） | ✅ |
| **Update Database** | 寫入或更新系統資料庫（如寫入簽收紀錄） | ✅ |
| **Call External API** | 呼叫第三方系統 API，支援 REST / GraphQL | ✅ |
| **Update Google Sheet** | 將變數寫入 Google Sheets（如記錄聯絡紀錄） | ✅ |
| **Read Google Sheet** | 從 Google Sheet 中讀取對應資料 | ✅ |
| **Send Email Notification** | 發送 Email 給主管或使用者（如通知客服需跟進） | ✅ |
| **Send Webhook** | 發出 HTTP POST / GET 給其他系統 | ✅ |
| **End** | 流程結束 | ❌（Elsa 內建） |

**6.2 流程舉例（送貨查詢流程）**

text

複製編輯

Start

↓

Wait For Reply ("請上傳送貨單圖片")

↓

Extract QR From Image

↓

Query Database (根據 QR 內容查送貨明細)

↓

Send WhatsApp Text (顯示訂單狀態)

↓

End

**6.3 視覺流程設計（Elsa Designer）**

* 支援拖拉節點、雙向連線、條件設定
* React 介面中提供左側節點清單 + 右側屬性設定欄
* 設計完成可儲存為 JSON，傳送至後端啟動流程
* 可設定「誰能執行」此流程（範本）

**7. 前端功能模組（React UI）設計與整合說明**

**7.1 系統登入與帳戶管理**

| **模組名稱** | **功能描述** |
| --- | --- |
| 登入介面 | - Google SSO 或 Email 密碼登入 - 登入後取得 JWT Token |
| 用戶註冊與管理 | - 管理帳號、變更密碼、角色（Admin / Operator） |
| 帳戶綁定 | - 顯示已綁定的 WhatsApp 號碼，並提供 QR code 掃描登入驗證流程 |

**7.2 Workflow 流程編輯器（Elsa Designer）**

| **模組名稱** | **功能描述** |
| --- | --- |
| 視覺流程設計器 | - Elsa Designer for React - 拖拉式 UI 設計流程，設定訊息、條件、節點順序 |
| 變數管理 | - 支援流程變數與上下文資料定義 - 支援條件設計與資料輸入 |
| 流程部署與測試 | - 一鍵部署至後端 Elsa Engine - 可在模擬介面中測試流程互動 |

**7.3 WhatsApp 對話視窗（Chat Viewer）**

| **模組名稱** | **功能描述** |
| --- | --- |
| 對話紀錄查閱 | - 顯示某 WhatsApp 使用者過往訊息紀錄（按 wa\_id 或手機號） |
| 手動發送訊息 | - 輸入框傳送自訂訊息、上傳圖片 - 可選擇「使用預設訊息模板」 |
| 對話轉人工/記錄註解 | - 標註此對話需人工處理，並記錄處理人與處理進度 |

**7.4 Template Library（訊息模板庫）**

| **模組名稱** | **功能描述** |
| --- | --- |
| 模板列表 | - 按分類（歡迎訊息、查詢回覆等）列出預設訊息模板 |
| 模板建立 | - 新增文字、圖片、清單、按鈕等不同訊息類型 |
| 模板使用統計 | - 顯示每個模板的使用次數、成效（可供營運分析） |

**7.5 外部資料設定與整合（Excel / API）**

| **模組名稱** | **功能描述** |
| --- | --- |
| Google Sheets 設定 | - 設定 Google Sheet URL、工作表範圍、欄位對應 - 供流程查詢與寫入使用 |
| 外部 API 設定 | - 設定 API URL、Header、Token、輸入/輸出欄位 mapping |
| ERP 系統範本設定 | - 可選擇以 Excel 輸入/輸出或 API 兩種方式與第三方系統連接 |

**7.6 系統操作日誌與管理**

| **模組名稱** | **功能描述** |
| --- | --- |
| 日誌查詢 | - 顯示所有系統操作紀錄、訊息錯誤、流程執行紀錄 |
| 權限設定 | - 控制哪些使用者可設計流程、可發送訊息、可讀取報表等 |
| 設定中心 | - API Token 管理、Webhook 設定、排程同步設定 |

**UI 整合與開發提示**

* Elsa Designer 元件可嵌入 React 頁面，並使用 REST API 傳遞流程定義
* WhatsApp Webhook → 進入後端流程判斷 → 通知前端 Chat 視窗更新
* 建議使用 Tailwind CSS + ShadCN UI 套件設計介面（快速開發且現代風格）
* Google 登入可用 Firebase Auth / Google Identity Platform 實作 SSO

**8. 系統整合架構與模組間通訊設計**

**8.1 架構總覽圖（System Architecture Overview）**

┌───────────────────────────────────────────────────────┐

│ Frontend (React) │

│ - Elsa Workflow Designer UI │

│ - Chat Viewer & Template Management │

│ - Login / Google SSO │

│ - Admin Dashboard │

└────────────┬──────────────────────────────────────────┘

│ REST API / SignalR (for live updates)

▼

┌───────────────────────────────────────────────────────┐

│ Backend Server (.NET / ASP.NET Core) │

│ - Elsa Workflow Runtime │

│ - WhatsApp Webhook Handler │

│ - DB / API Activities Execution Layer │

│ - Message Template Processor │

│ - Google Sheets Integration │

└────────────┬──────────────────────────────────────────┘

│

▼

┌──────────────────────┐ ┌────────────────────────────┐

│ Meta WhatsApp API │◄──────►│ External ERP/3rd Party API │

└──────────────────────┘ └────────────────────────────┘

│

▼

┌──────────────────────┐

│ Database (SQL Server) │

│ - Users / Workflows / Messages │

│ - Logs / API Settings / Contacts │

└──────────────────────┘

**8.2 模組互動流程（以「拍照查詢送貨單」為例）**

1. **用戶拍照並發送給 WhatsApp Bot**
   * WhatsApp Webhook POST 到 .NET Core webhook controller
   * 判斷 sender wa\_id 並觸發對應 workflow
2. **Workflow 啟動 → Elsa 執行節點**
   * 節點順序：Wait for Image → QR Scan → Query Database → Send Message
3. **查詢流程**
   * QR Code 節點使用 Tesseract 或其他影像處理解碼（若已移除則直接跳轉）
   * 查詢資料庫或 Google Sheets，取得送貨單明細
   * 回傳訊息使用 WhatsApp Template 或純文字顯示結果
4. **流程中如需 API Call**
   * 使用 Elsa Custom Activity 呼叫對應 ERP 或 Google Apps Script API
   * Response 傳回後更新用戶資料或進一步通知客戶
5. **前端 UI**
   * 管理員可查看即時對話內容、處理紀錄、Workflow 設計

**8.3 資料庫設計關鍵表（簡要）**

| **資料表名稱** | **用途說明** |
| --- | --- |
| users | 使用者帳號資料、登入類型（Google/本地） |
| workflows | 儲存設計器儲存的 JSON 流程定義 |
| messages | 訊息收發紀錄，包括 wa\_id、時間、內容 |
| templates | 訊息模板定義，可供流程中快速使用 |
| api\_integrations | 外部 API 設定資訊（URL, headers, auth） |
| google\_sheets\_settings | 記錄 Google Sheet 設定檔 |
| execution\_logs | 每次流程執行歷程，含錯誤與變數快照 |

**8.4 架構特色與擴充彈性**

| **項目** | **說明** |
| --- | --- |
| **微服務支援** | 可獨立拆分 WhatsApp Handler、Workflow 執行器等模組 |
| **彈性流程邏輯** | 所有流程皆由使用者以視覺化方式設計，後端僅負責執行 |
| **訊息記錄完整** | 所有收發訊息與變數上下文皆可儲存，利於查核與後續分析 |
| **簡單擴充接口** | 只需新增 Elsa 活動（Custom Activity）即可增加功能 |
| **支援 CI/CD** | 前端 React + Vite，後端 .NET Core 皆可自動化建置部署 |

**9. 系統權限與角色設定（Permissions & Roles）**

**9.1 角色分類與使用場景**

| **角色名稱** | **描述與使用場景** |
| --- | --- |
| **系統管理員 (SysAdmin)** | 擁有整個平台的最高權限。可管理所有公司、帳號、流程、模組設定等。 |
| **公司擁有者 (Client Admin)** | 客戶方的負責人帳號，能新增本公司用戶、設計流程、管理 API、訊息模板等。 |
| **流程設計員 (Workflow Designer)** | 專責設計 WhatsApp 工作流程，能操作流程圖、設定變數、測試運行流程。 |
| **訊息管理員 (Message Manager)** | 專責建立及維護 WhatsApp 訊息模板、快速回覆語句庫。 |
| **客服操作員 (Agent)** | 只能處理自己的工作流程任務、查看客戶訊息記錄、填寫流程中所需資料等。 |
| **訪客/審核者 (Viewer/Auditor)** | 只可檢視流程記錄或訊息紀錄，無修改、無執行權限。 |

**9.2 權限結構與建議實作方式**

| **功能模組** | **權限粒度** | **建議設定方式** |
| --- | --- | --- |
| 流程設計器 | 建立、編輯、啟用、刪除流程 | Workflow Designer 以上可用 |
| 使用者管理 | 建立/停用使用者、角色設定 | 由 SysAdmin 或 Client Admin 處理 |
| 訊息模板管理 | 新增/刪除/發佈/測試 | Message Manager 或 Client Admin |
| ChatBot 訊息記錄 | 檢視指定 wa\_id、訊息過濾 | Agent 可見自己訊息，Admin 全域可見 |
| 外部 API 整合 | 設定 API URL、Key、測試 | 僅限 Client Admin 管理 |
| Google Sheets 整合 | 指定 Sheet 範本、欄位對應 | 由 SysAdmin 或 Workflow Designer |
| 登入機制 | Google SSO、帳號密碼登入設定 | 系統全域開啟/關閉，Client Admin 選擇 |
| 活動紀錄與稽核 | 查閱使用記錄、流程執行紀錄 | Viewer、Auditor、SysAdmin 可檢視 |

**9.3 使用者與角色關聯建議資料表**

| **資料表名稱** | **說明** |
| --- | --- |
| users | 含帳號、email、GoogleID、公司ID、角色ID |
| roles | 角色定義表，例如 Agent、Admin 等 |
| user\_roles | 使用者對應多角色（可多對多） |
| permissions | 模組功能與權限定義（CRUD 旗標） |
| role\_permissions | 將權限對應到角色上 |

備註：若使用 ASP.NET Identity 或 Elsa 官方 Identity 實作，已內建相關角色模組可擴充使用。

**9.4 多公司（Multi-Tenant）支援**

支援同一平台上註冊多個公司，每家公司使用者、流程、訊息紀錄、API 設定、訊息模板皆完全隔離，建議使用 TenantId 欄位識別。

**10. 系統運維與日誌（Monitoring & Logs）**

**10.1 系統運行監控指標**

為確保平台穩定運作，建議部署以下關鍵監控項目：

| **監控項目** | **說明** | **工具建議** |
| --- | --- | --- |
| **API 請求統計** | 所有 API 存取次數、延遲時間、錯誤率 | Azure App Insights / Prometheus |
| **Webhook 狀態** | WhatsApp webhook 接收成功率、平均延遲、訊息接收次數 | 自建 Dashboard 或 Kibana |
| **流程執行統計** | 每日流程啟動數量、執行結果統計、失敗流程占比 | Elsa ActivityLog + 自定報表 |
| **訊息傳送日誌** | WhatsApp API 發送記錄、錯誤訊息、template ID、response 狀態 | Database + 對接 WhatsApp 狀態回傳 |
| **工作佇列與排程** | 背景任務數量、佇列長度、排程狀態 | Hangfire / Quartz.NET |
| **登入與權限異常** | 可疑登入記錄、帳號連續登入失敗、權限不足錯誤 | ASP.NET Identity Logs |

**10.2 活動與流程執行日誌（Execution Logs）**

每一筆 WhatsApp 互動應產生完整的執行紀錄，便於追蹤與復盤：

| **欄位名稱** | **說明** |
| --- | --- |
| wa\_id | WhatsApp 帳號 ID |
| message\_type | text / image / document 等 |
| workflow\_id | 所屬流程 ID（若有） |
| activity\_path | 流程中的節點路徑（例如：Receive → DB → Send） |
| status | success / failed |
| error\_message | 若失敗，顯示錯誤訊息 |
| executed\_at | 執行時間 |

建議每日自動產出報表顯示流程成功/失敗趨勢圖，並發送給系統管理員審閱。

**10.3 系統除錯與開發模式支援**

為方便開發與測試，建議提供以下模式與工具：

* **模擬 WhatsApp 輸入訊息**：讓開發者能不透過真實手機測試 webhook 邏輯。
* **流程節點模擬器**：可單步執行流程節點，並顯示上下文變數與輸出結果。
* **完整 Request/Response 封包紀錄**：用於 WhatsApp API call 記錄與錯誤重現。
* **訊息追蹤 UI**：在前端顯示特定使用者的訊息來往與流程觸發記錄。

**10.4 日誌儲存與存取方式**

| **資料表或儲存項目** | **建議方式** |
| --- | --- |
| execution\_logs | 儲存流程每一步的執行結果 |
| whatsapp\_messages | 儲存 webhook 接收與 API 傳送記錄 |
| api\_audit\_logs | 外部 API 呼叫與回應紀錄 |
| login\_audit\_logs | 所有登入與登入錯誤記錄 |
| 文件儲存（如拍照上傳） | 儲存至雲端（如 GCS, S3），紀錄 metadata |

**11. 採用技術棧與部署策略（Tech Stack & Deployment）**

**11.1 前端（Frontend）**

| **項目** | **技術或框架** | **說明** |
| --- | --- | --- |
| UI 框架 | **React.js** | 搭配 TailwindCSS 或 ShadCN UI 做簡潔模組化設計 |
| Flow Builder UI | JointJS / React Flow | 作為流程設計器的基礎圖形元件，可視化流程節點、連線與屬性調整 |
| 登入與授權 | Firebase Auth / Google SSO | 整合 Google 登入並支援系統內帳號同步 |
| 前端打包工具 | Vite / Webpack | 更快的開發與建構速度 |

**11.2 後端（Backend）**

| **項目** | **技術或框架** | **說明** |
| --- | --- | --- |
| Web Framework | ASP.NET Core | 高效能後端框架，支援 RESTful API、SignalR 及中介層架構 |
| 工作流程引擎 | Elsa Workflows v3 | 開源 .NET 標準工作流程引擎，支援流程儲存、事件觸發、活動設計 |
| Webhook Controller | 自訂 API Controller | 負責接收 Meta WhatsApp webhook，並觸發對應流程 |
| 任務排程與延遲執行 | Hangfire / BackgroundService | 管理需要延遲處理的步驟，如等待回覆、定時訊息等 |
| 身分與權限管理 | ASP.NET Identity / OpenIddict | 提供登入、角色、權限等功能 |
| 影像處理（如未來使用） | ImageSharp / Tesseract (已取消) | 未啟用，但預留結構支援圖像處理或 OCR 分析 |

**11.3 數據儲存（Database & Storage）**

| **資料類型** | **技術建議** | **說明** |
| --- | --- | --- |
| 主系統資料庫 | PostgreSQL / SQL Server | 儲存帳號、訊息、流程定義、日誌、權限等 |
| 文件與圖片（如簽收單） | Google Cloud Storage (GCS) 或 S3 | 儲存拍照、PDF、Excel 等文件；與 metadata 關聯 |
| Excel 整合 | Google Sheets API | 提供讀取及寫入 Excel 能力，便於與客戶第三方系統整合 |
| 流程日誌 | 內建資料表 / ElasticSearch | 儲存每一筆流程活動記錄，便於查詢及統計報表 |

**11.4 外部整合元件（External Integrations）**

| **整合項目** | **API/技術** | **描述** |
| --- | --- | --- |
| WhatsApp | Meta WhatsApp Business Cloud API | 作為所有訊息發送與接收的主幹平台 |
| Google 登入 | Google OAuth2 | 使用者可透過 Google 帳號快速登入 |
| Google Sheets | Sheets API | 可作為輕量級第三方資料交換與查詢方式 |
| 第三方 API | 自訂定義的 API 連線 | 客戶系統（如 ERP、CRM）透過 API 查詢或更新資料 |

**11.5 部署與 CI/CD 建議**

| **項目** | **建議技術** | **備註** |
| --- | --- | --- |
| 容器化部署 | Docker + Docker Compose | 分層部署前後端與 Elsa 引擎服務 |
| 自動部署 | GitHub Actions / GitLab CI | 自動建構、測試與部署流程 |
| 伺服器選擇 | GCP / Azure / AWS | 選擇具彈性與儲存空間的雲平台部署 |
| Domain & HTTPS | Cloudflare / NGINX + LetsEncrypt | 免費 HTTPS 憑證 + 快速 DNS 設定 |

**12. 擴展規劃與商業模式建議**

（Scalability & Business Model）

**12.1 擴展方向（Scalability Strategy）**

| **項目** | **說明** |
| --- | --- |
| 多租戶 SaaS 架構 | 將平台升級為多租戶架構，每個企業可擁有獨立資料空間與帳號權限 |
| 模組化功能啟用機制 | 各模組（流程設計器、訊息管理、API 整合、報表）可按需開關，利於依客戶需求提供不同級別方案 |
| 插件擴展機制 | 建立 plugin SDK，讓開發者可編寫自訂活動（如特定 CRM 連接、Webhook 回調邏輯） |
| 語言與區域擴展 | UI 支援繁中、簡中、英文；訊息模版支援 i18n 格式；並可根據地區法律處理資料儲存與 GDPR/FIPA 等合規要求 |

**12.2 商業模式建議（Business Models）**

| **模式** | **描述** |
| --- | --- |
| SaaS 月費訂閱制 | 分為 Basic / Pro / Enterprise 三種等級，依功能與用戶數收費；建議提供 7 天試用期 |
| Chatbot流程模板商店 | 提供企業或顧問用戶販售流程模板（如送貨查詢、報修流程、回覆 FAQ 等），平台抽佣 |
| Plugin 市集 | 提供社群或第三方開發者上架自訂活動節點與工具模組，平台提供 Marketplace，抽佣與驗證 |
| API 限額超額收費機制 | 基本方案每月提供 N 次 Meta API 呼叫量，超出部分按次收費 |
| White-label 解決方案 | 提供品牌客戶可自訂 LOGO/網域的企業版（含部署至其私有雲），一次性專案費 + 維護年費 |

**12.3 潛在應用產業（Target Industry Applications）**

| **行業** | **應用情境示例** |
| --- | --- |
| 物流／快遞 | 查詢送貨單狀態、自動通知延誤、收件人身份確認、收據回傳拍照 |
| 銷售團隊 | 以 WhatsApp 傳送報價單、下單流程、產品推廣自動話術 |
| 金融／保險 | WhatsApp 諮詢、自動回覆條款問題、上載文件、流程化審批 |
| 建造／工程 | 地盤工人到場打卡通知、與 HR 系統整合、每日簡報通知 |
| 客服中心 | 用戶發訊後觸發流程、自動分類問題類型、導入真人客服或 SLA 緊急級別處理 |

**12.4 長遠目標建議（Long-term Goals）**

1. **AI 自動流程建議引擎**：根據歷史對話與操作，系統可主動建議最合適的節點串接方式，讓非技術用戶也能快速設計。
2. **整合 LLM（大語言模型）理解訊息意圖**：結合 GPT/Claude 等模型理解客戶對話，並選擇相應流程。
3. **全通道整合**：除 WhatsApp 外支援 LINE、Messenger、WebChat、Email等多通道整合。
4. **ISO 認證準備**：依據客戶要求準備 ISO27001 或 SaaS Security 認證標準。

**13. 測試計劃與品質保證**

（Testing & QA Strategy）

**13.1 測試階段規劃（Testing Phases）**

| **測試階段** | **說明** |
| --- | --- |
| 單元測試（Unit Test） | 對各個模組（例如：SendMessageActivity、DatabaseQueryActivity）進行功能邏輯測試 |
| API 測試 | 使用 Postman / Swagger 進行 REST API 呼叫驗證，涵蓋成功案例、異常輸入、授權測試等 |
| 整合測試（Integration Test） | 測試流程中節點與節點之間的資料流、Webhook 回調、Meta API 實際送出與回應 |
| 前端功能測試 | 測試流程設計器 Drag-and-Drop、節點屬性面板、訊息模版庫操作等是否正常，使用 Cypress/Selenium 可自動化處理 |
| 使用者驗收測試（UAT） | 由試用企業或內部業務人員進行端對端實務操作測試，評估實際商業流程中使用的易用性與正確性 |
| 回歸測試 | 每次更新後針對既有功能進行測試，避免舊功能在版本迭代中失效 |

**13.2 自動化測試工具建議**

| **類型** | **工具** |
| --- | --- |
| API 測試工具 | Postman, Swagger, RestSharp（C# 測試套件） |
| 單元測試 | xUnit（C#）、NUnit、Moq（模擬依賴注入） |
| 前端測試 | React Testing Library, Cypress（E2E UI 測試） |
| 工作流程模擬測試 | 可考慮為 Elsa Workflow 寫工作流程 YAML 測試集並用 Elsa.Testing 進行模擬運行 |

**13.3 錯誤追蹤與回報（Bug Tracking & Reporting）**

| **工具** | **說明** |
| --- | --- |
| GitHub Issue | 每個功能模組建立獨立 Issue 並指派，建立測試步驟與再現方式 |
| ClickUp / Trello | 視覺化顯示測試進度與回報狀況，用來支援跨部門測試協作 |
| Application Insights / Sentry | 錯誤報表整合，記錄後台 API/流程錯誤、反應時間、失敗率等，支援事後 Debug 與監控 |

**13.4 測試案例與覆蓋範圍建議（Test Coverage）**

| **模組/功能** | **測試項目** |
| --- | --- |
| 使用者管理與登入功能 | 帳號註冊、Google SSO 登入、密碼重設、權限限制 |
| 流程設計器 | 拖拉節點、儲存流程、重新載入流程、流程節點配置改變即時反映 |
| Meta WhatsApp API | Send Message、Template Message、Webhook、Media Upload/Receive 處理 |
| 節點邏輯測試 | Message Wait、條件分支、資料寫入寫出節點、Webhook 處理、第三方資料來源整合節點 |
| 資料查詢與 DB 操作 | 檢查 SQL 操作、Excel 模擬資料操作、錯誤資料處理 |
| UI 表單與模版庫 | 訊息模版建立、範本測試、變數帶入、儲存與讀取 |

**14. 上線準備與部署建議**

（Deployment & Go-live Checklist）

**14.1 部署環境建議**

| **環境類型** | **說明** |
| --- | --- |
| 開發環境 | 本機開發使用 Visual Studio 2022 + Docker / IIS Express；React 本地使用 npm run dev |
| 測試環境 | 建議部署至內部伺服器或雲端（如 Azure / AWS / Railway）進行端對端流程測試 |
| 生產環境 | 最終部署使用容器（Docker Compose 或 K8s），獨立資料庫（SQL Server / PostgreSQL）與 HTTPS 支援 |
| 靜態資源 | React 前端可部署於 Vercel / Netlify 或內部 NGINX；流程圖、模版資源儲存於雲端儲存（GDrive / S3） |

**14.2 Go-live 檢查清單（Checklist）**

| **項目** | **說明** | **完成狀態** |
| --- | --- | --- |
| ✅ 使用者帳號、權限設定 | 測試所有帳號層級（管理員、一般員工、審批員、客戶支援） | ✔️ |
| ✅ WhatsApp Business API 實名認證 | 確認 Facebook Meta API 的帳號和 webhook 設定完成 | ✔️ |
| ✅ 模板訊息審批與測試 | 所有預設訊息範本已於 Meta 後台審批並成功測試 | ✔️ |
| ✅ 流程圖流程儲存與版本管控 | Elsa 中建立流程後能正確儲存、導入、導出並支援版本切換 | ✔️ |
| ✅ DB 資料測試與清理 | 測試表單資料 CRUD 流程、確認預設 demo 資料與初始化腳本 | ✔️ |
| ✅ Error Log 與通知機制 | API Exception 有記錄，重大錯誤發 Slack / Email 通知 | ✔️ |
| ✅ 備份與復原策略 | 設有每日自動備份，提供還原指令或 UI 工具 | ⬜ |
| ✅ 文件與使用者手冊 | 提供內部測試文件、操作手冊、快速上手指南 | ⬜ |

**14.3 持續整合與自動部署（CI/CD）**

| **項目** | **建議工具** |
| --- | --- |
| 版本控制 | GitHub 或 GitLab |
| 自動化部署 CI/CD | GitHub Actions / Azure DevOps / Jenkins |
| 版本標記與回滾 | 每次部署標記版本（v1.0.0）並支援回滾部署 |

**14.4 Google 整合項目（選用）**

| **功能** | **描述** |
| --- | --- |
| Google Login (OAuth2) | SSO 登入，使用 Gmail 帳號即可存取系統 |
| Google Drive Excel 整合 | 使用者可上傳 Excel 自動讀取（如發票號碼、配送狀況）、寫入更新訊息 |
| Google Sheet API 使用 | 建立機器人帳號，用以讀寫客戶提供的 Sheet 作為 ERP 數據橋接 |
| GCP Cloud Storage 整合 | 用來儲存大型圖片附件、簽收單據，並於 WhatsApp chatbot 回傳連結 |

**15. 未來擴展方向**

（Future Roadmap）

**15.1 功能擴展**

| **模組** | **說明** |
| --- | --- |
| 多平台支援 | 支援 Facebook Messenger、LINE OA、Telegram Bot 作為訊息入口 |
| 多語系 / 翻譯支援 | 根據客戶來源自動識別語言、支援 Google Translate API 做自動翻譯 |
| Bot 自學習回應 | 使用 RAG（Retrieval Augmented Generation）方式導入知識文件與常見問答 |
| 聊天記錄 AI 摘要與標籤 | 將歷史對話進行自動摘要與分類，用於統計與回應優化 |
| 自助式流程表單建立工具 | 客戶支援人員可透過流程圖介面，自行拖拉建立流程與 API 銜接 |
| 表單／輸入欄位設定 UI 工具 | 支援每一節點定義輸入欄位、驗證規則與預設值，例如“請上傳送貨單”或“輸入電話號碼” |
| Flow 模版市場與分享平台 | 建立一個模版庫，上載常用流程與訊息模版並支援複製與二次修改 |

**15.2 技術發展重點**

| **項目** | **說明** |
| --- | --- |
| API Orchestration 平台 | 把多個外部 API 整合為一個流程節點，例如 ERP 查單、物流派送查詢、Email 寄出等 |
| AI 聊天增強引擎 | 使用 GPT API / Mistral 等大型語言模型，對自然語言指令進行理解與回應 |
| 資料分析與 KPI Dashboard | 為管理者提供實時客服互動量、完成率、流程平均耗時等視覺化報表 |
| 多租戶（Multi-Tenant）機制 | 為不同客戶公司提供獨立資料庫 Schema，確保安全隔離並支援自訂流程模版 |
| Audit Log + 訊息追蹤 | 每次流程呼叫、訊息觸發與 API 請求皆具備紀錄與追溯 ID，可便於法規遵循與錯誤調查 |

**15.3 業務應用延伸**

| **類型** | **範例** |
| --- | --- |
| 外送物流 | 拍照確認送貨 → 自動辨識 QR → 回寫 ERP，生成 Proof of Delivery (POD) |
| 客戶支援 | 按輸入的保單／訂單編號查詢詳情、自助修改資料、自助下載文件 |
| 課程預約與提醒 | 使用 WhatsApp 發出付款通知、課程通知、自動延期提醒等 |
| 招募與培訓流程 | 應徵者透過 WhatsApp 收取測驗連結、完成測試、通知安排面試或自動寄合約 |
| 醫療提醒與文件送達 | 醫療機構可透過 WhatsApp 傳送預約記錄、檢查報告（含加密 PDF），並記錄讀取情況與查詢歷史 |

**16. 總結與建議**

（Conclusion & Suggestions）

**16.1 系統目標回顧**

SS.AI 客服平台的核心設計理念是建構一個以 WhatsApp 為主要通訊渠道的智能客服流程平台，透過 **圖形化流程設計器**、**整合 Meta API**、**後台支援 C# 開發**、**可擴展的 Database 架構** 與 **自動化節點配置能力**，讓非技術人員也能操作並落地複雜的工作流程，達成以下三大目標：

1. **提升客戶服務效率**：以 WhatsApp 為入口，自動處理送貨通知、回單查閱、簽收確認、保單查詢等常見任務。
2. **減少系統整合成本**：無需重新開發現有 ERP 或 HR 系統，改以 API 或 Excel Gateway 整合數據讀寫。
3. **支援持續擴展與複用**：流程模組化與流程模板市場可協助未來針對新客戶、新業務快速佈署。

**16.2 推薦執行策略（Roadmap Strategy）**

| **階段** | **建議行動** |
| --- | --- |
| POC 開發 | 建立最小可用產品（MVP）：包括 WhatsApp Send / Wait for Reply + Database 查詢/寫入 |
| 初始部署 | 為 1–2 個典型客戶（如物流／保險）建立實際流程，測試平台穩定性與流程設計器可用性 |
| 團隊建立 | 招聘 1–2 位專職 C# / React 工程師，持續維護平台與發展流程模塊 |
| 市場試點 | 開始拓展至其他需要高互動性的行業，例如教育機構、售後支援、公共服務等 |
| 商業化 | 推出訂閱制定價模式（如月費／每流程／每用戶）+ 建立模板商城與顧問服務收益 |

**16.3 延伸合作與技術考量**

* **OAuth 2.0 / Google SSO 整合**：統一身份驗證機制，簡化登入體驗與安全管理。
* **低代碼平台插件方向**：可考慮未來將本平台模組作為 Make.com、n8n 等平台的 Plugin 發佈。
* **訊息追蹤與 AI 記錄摘要**：導入 GPT 模型自動產出回應紀錄摘要與情緒標籤以供審查與改善。
* **Chat Template 智能補全**：開發類似 Gmail Smart Compose 之建議回應模組。

**16.4 最後建議**

* 以模組方式構建平台各部分，可快速迭代與測試。
* 鼓勵團隊採用 Cursor / Augment 等 AI 開發工具，提高開發效率。
* 盡早導入流程設計器與模版管理系統，降低對程式人員依賴。
* 定期與使用者訪談，優化流程建立 UX 與前台互動介面。