20250617郵務流程

想問你關於台灣公司的離職申請，會放在內部網頁給使用者本人填寫嗎，還是會隱藏起來不讓使用者直接看呢

**ChatGPT 說：**

這個問題牽涉到企業文化、人資制度與資訊安全等層面，以下是我的分析與說明方式：

**✅ 問題確認**

你想了解的是：「**台灣企業在處理員工離職申請時，這項功能（離職申請填寫）會不會直接開放在內部網頁讓員工自己填？還是通常不會讓員工直接看到或操作？**」

我確認你的核心問題是針對：

* **離職申請操作權限**（是否讓員工自己發起）
* **是否會藏起來避免員工主動點選／申請**
* **是否有通用的慣例或流程設計**

**🔍 背後邏輯分析**

從我作為**系統分析師＋人資流程顧問**的角度來說，離職申請是否開放給使用者本人填寫，會受到下列幾個因素影響：

**1. 公司文化與人資策略**

* **開放透明文化**（如新創、中大型外商）：多半會開放員工在內部HR系統中主動點選「離職申請」或「申請異動」等功能，表明意願。
* **管控導向文化**（如傳產、中小企業）：會將離職申請流程設計為需要「主管面談」或「人資介入」後才能進行，並不直接讓員工操作。

**2. 系統設計與權限規劃**

* 若使用像 **PeopleSoft、Workday、SAP SuccessFactors** 等大型HR系統，功能設計通常支援「員工主動申請」並啟動工作流程。
* 若是自建或簡易ERP（如MF2000或傳產HR系統），可能不具備前台主動申請的功能，或預設關閉權限避免員工操作。

**3. 資訊安全與預警機制**

* 有些公司擔心員工點選離職申請後產生心理落差或離職流言，會選擇將該功能**隱藏起來，僅於特定條件顯示**（例如主管提報、面談結束後才開通填單權限）。

**🧩 實際運作範例**

| **公司類型** | **離職申請可見？** | **操作方式說明** |
| --- | --- | --- |
| 新創公司 / 外商 | ✅ 可見 | 內網系統中可點選「提出離職申請」，會啟動流程（如主管核准、交接流程） |
| 傳產公司 | ❌ 不可見 | 員工無法看到「離職」表單，必須透過主管或人資主動代為啟動 |
| 部分ERP系統（如MF2000） | ❌ 預設隱藏 | 系統流程設計上通常為內部操作，由人資或主管進行離職資料填寫與異動作業 |

**📌 我的建議**

若你是要設計或管理內部系統，可依以下方式處理：

1. **預設不開放離職申請給員工點選，但保留人資權限手動開通**
   * 可依照面談後或特定流程節點自動開啟填單權限。
2. **設置提示或說明機制**
   * 若員工主動點進「異動申請」，可顯示說明：「如需辦理離職，請聯繫您的主管或人資窗口」。
3. **留存離職申請紀錄與流程軌跡**
   * 無論誰填寫，離職流程務必具備流程簽核、紀錄保留與通知機制。

**📚 延伸資源與學習建議**

* 可參考 Workday、SAP SuccessFactors 或 BambooHR 的「自助服務流程設計」
* 若你們公司使用 MF2000 或自行開發系統，我可以協助設計一套「離職流程設計建議」，包含：
  + 權限控制（誰可填）
  + 流程觸發邏輯（何時顯示）
  + 流程節點設計（交接、離職面談、資產歸還）

如你願意提供你們目前用的是哪套系統（或ERP、HR系統），我可以更精準地給你具體的流程設計建議。是否需要？或者你目前已在規劃內網流程？我可以協助你思考整體設計。

**你說：**

那如果走人資介入跟主管面談的流程，內部網站我們會需要放，但同時我們的人資也有MF2000要維護，怎麼樣規劃比較適合，請以一個PM的腳色去思考這件事

**ChatGPT 說：**

從\*\*專案經理（PM）\*\*的角度，我會將這件事情拆解成幾個構面，協助你同時滿足「內部網站可查詢或通知操作」與「MF2000 系統正式維護與作業」的雙重需求，以下是完整的規劃邏輯與建議：

**🔍 一、任務確認**

**目標：**

* 建立一套 **離職流程規劃與系統整合機制**，讓員工可以在內部網站得知流程與啟動管道，但**不直接操作 MF2000 離職維護**。
* 人資仍在 MF2000 執行正式維護、異動、離職後續（如計薪、健保勞保、財務作業等）。

**現況資源：**

* 內部網站（自建或公司入口網站）可用來提供操作指引、收單與通知。
* MF2000 是核心人事資料維護系統，離職需由人資人工操作。

**🧭 二、流程設計藍圖（PM思維分層）**

**🔹 A. 對員工端：內部網站為申請入口與流程指引平台**

| **元件** | **說明** |
| --- | --- |
| 操作說明 | 放置「我要申請離職」說明頁，說明需經主管與人資面談，並連結申請入口。 |
| 申請表單 | Google Form、內建表單或簡易收單系統（如PHP Form）均可，重點是紀錄員工意願與基本說明（如預計離職日、原因）。 |
| 通知機制 | 表單送出後，自動寄信或以 webhook 通知主管與人資窗口。 |
| 可選功能 | 進階可整合 SSO，記錄申請人資訊與申請歷程。 |

**🔹 B. 對主管端：收到通知後進行初步溝通與面談確認**

| **任務** | **作法** |
| --- | --- |
| 溝通記錄 | 可將面談紀錄表或同意離職確認單於 Teams/內網中填寫。 |
| 確認交接責任 | 確認是否需建立交接清單（由部門自行維護或交由人資後續督導）。 |

**🔹 C. 對人資端：後端作業執行者（MF2000）**

| **項目** | **說明** |
| --- | --- |
| 離職維護 | 由人資登入 MF2000，於「人員異動」功能中維護「離職日期」、「離職原因」、「交接人員」等。 |
| 資料稽核 | 確認是否同步關閉各系統帳號、完成請款、薪資結算、特休清除等作業。 |
| 流程記錄 | 可在 MF2000 備註區欄位加註：申請日、面談日、主管確認日，以便後續查考。 |

**🧩 三、系統串接建議（無需直接整合 MF2000 的實用做法）**

| **面向** | **建議作法** |
| --- | --- |
| **不進行直接API串接** | MF2000 為封閉式PB系統，通常無API供資料寫入，故不建議讓員工操作資料。 |
| **使用中介表單** | 由網站產生「申請通知」與「審核紀錄」，作為人資進入 MF2000 維護的依據。 |
| **追蹤機制** | 可建立一個「離職進度追蹤表」供人資更新進度（如：已面談、已維護、已交接完成、已註銷帳號等）。 |
| **人資流程表單電子化** | 將主管簽核表、交接清單轉為 Google 表單或 SharePoint 文件，避免紙本流轉。 |

**🔧 四、角色與權責分工表**

| **角色** | **工作項目** | **系統操作範圍** |
| --- | --- | --- |
| 員工 | 填寫離職申請表 | 內部網站（前台表單） |
| 主管 | 面談、確認意願、交接安排 | 表單審核 / Teams 或 Email 通知 |
| 人資 | 離職維護、薪資處理、帳號關閉 | MF2000 系統（後台人工維護） |

**✅ 五、PM落地行動建議**

1. **先確認公司是否允許員工主動申請離職、是否保留決定權給主管／人資？**
2. **若允許申請，可先行導入簡易表單＋通知流程，不動到 MF2000 架構。**
3. **人資持續使用 MF2000 為唯一資料源，由流程單據做為憑據。**
4. **導入「離職追蹤看板」（如Excel或Notion），明確區分階段與負責人。**
5. **可製作 SOP 文件與視覺化流程圖，供網站與新人手冊使用。**

如需，我也可以幫你整理這套流程成：

* ✅【流程圖】供網站展示或主管溝通
* ✅【員工端網頁說明文字】
* ✅【表單欄位建議】
* ✅【人資後端操作對照表】  
  你需要嗎？或者想先從哪一部分開始製作？

**你說：**

這樣會有重工的問題嗎，因為我們面談的話通常不會由主管本人進行記錄，因為我們是出版單位屬於傳統產業

**ChatGPT 說：**

這個問題問得非常好，的確「**重工**（重複作業）在離職流程中非常常見\*\*」，尤其在傳統產業或出版業這類對流程控管與手動處理比例高的產業更為明顯。以下我將以專案經理＋系統分析師角度幫你**精準分析你們的情境**，並提出具體改善建議：

**🎯 一、你的情境確認**

根據你的補充，我整理出下列關鍵事實：

1. **主管不會自行記錄面談內容** → 通常僅口頭了解，或由人資主導面談。
2. **流程主導在於人資**，主管角色偏向知情與支持，不一定參與詳細記錄。
3. **屬於出版業，偏向傳統產業流程設計** → 權限集中、人員分工明確，系統自助化程度較低。
4. **你們有 MF2000 系統**，實際離職維護須在人資端操作，無自動串接。

**🧩 二、目前流程下會產生的「重工」問題點**

| **流程步驟** | **潛在重工點** |
| --- | --- |
| 員工填寫網站離職申請 | 人資收到後仍需**手抄或重複錄入到 MF2000** 進行正式離職作業 |
| 面談紀錄產出 | 主管未記錄，人資需「重述」「手寫」面談過程，甚至口頭記錄 |
| 簽核或交接流程 | 每位離職者需「重複跑流程單」、「列印簽名」、「掃描留存」等 |
| 離職清單處理 | 各部門（如 IT/財務/行政）需手動重複追蹤清單項目完成情況 |

**✅ 三、PM改善建議：避免重工的三層精簡策略**

**▶ 1.【保留人工流程，但「集中收單 + 整合追蹤」】**

**重點：** 不要讓網站自動導向太多步驟，而是**將所有員工申請與後續事項統一記錄於一個系統化追蹤表單**中，由人資統籌處理。

* ✅ 使用 Google Sheet / Notion / Excel 離職追蹤清單：
  + 員工填寫申請後，自動登記一筆資料。
  + 同一筆資料由人資後續填入：面談完成日、主管確認、交接清單狀態、MF2000完成日。
  + 建立欄位如：

複製編輯

員工姓名｜部門｜申請日期｜預計離職日｜面談狀態｜主管確認｜MF2000處理日｜備註

**▶ 2.【主管不記錄，但「給主管一個按鈕」完成確認】**

**重點：** 不要要求主管輸入紀錄，**只給「確認同意」或「已知情」的按鈕或欄位**，由人資統一維護紀錄。

* ✅ 表單或追蹤表設計中，只需要：
  + 「主管是否已同意」→ 由人資詢問後手動填寫「是」
  + 若主管想備註，可讓人資統一收集，再加在欄位中

**▶ 3.【MF2000僅為最終輸入點，其他前置流程不要硬對接】**

**重點：** MF2000 是正式記錄平台，但不是流程推動者，因此：

* 不應該嘗試自動串接網站填單 → MF2000
* 但可以使用前置表單當作「MF2000維護依據」
* ✅ 比方說人資收到網站申請後，在追蹤表上註明 MF2000 的登錄時間與備註

**🧭 四、建議規劃架構（簡化且無重工）**

markdown

複製編輯

員工端：內部網站表單送出

↓

人資端：收表單 → 問主管 → 面談 → 填追蹤表

↓

人資端：依照追蹤表內容登入 MF2000 維護

↓

人資端：統一通知各部門（清帳、關帳號等）

* 所有「追蹤」、「狀態更新」都維護在同一個表格
* 員工不接觸 MF2000、人資操作一次，不需來回查問

**🛠 補充建議：可協助你製作下列項目**

若你需要，我可以幫你做出：

1. ✅ **離職申請表範本**（Google 表單或網頁格式）
2. ✅ **離職流程追蹤表範例**（Excel 或 Notion）
3. ✅ **簡化流程圖（A4尺寸、適合傳統產業）**
4. ✅ **主管與人資的操作說明稿**（簡明步驟與提醒）

你希望從哪個部份先開始呢？或是否希望我幫你整理流程圖與表格設計？

**你說：**

目前要規劃內部網站填新人到職有多個欄位然後填寫人員可能是該單位的人幫填寫的，內容的個欄位會需要提供給三個部門人資、財務、資訊，分別是不太相同的資訊也可能不同時間點會知道，我現在有個問題是不知道該資料庫要設計一個表或三個鰾然後有一種想法是不知道能不能是一個表來設計然後填寫人田內容可以存檔可以修改資料然後真正案送出來把資料存到另一支表那張表就能查詢輸出三張表單給三個部門，你可以給我建議怎麼設計跟建議PK或什麼之類的 目前已知大概如圖這些欄位跟想法

**ChatGPT 說：**

從你提供的圖與需求說明，我以系統分析師及資料庫規劃角度，給你具體且完整的建議，包含思考邏輯、資料表設計、PK建議與實務建議：

**一、需求確認與設計原則**

**🚩 核心需求確認：**

* 填寫人可能是各單位（非本人）幫新人填寫到職資料。
* 填寫資料內容最終需提供給**人資、財務、資訊三部門**，各部門欄位需求不盡相同且可能不同時點才會完成填寫。
* 希望先暫存或保存後仍可修改，最終提交後資料要能清楚、易於產出給三個部門。

**📌 設計原則：**

* **避免重複建表**，一張主要儲存表作為統一資料庫設計。
* 提供方便維護、修改的中間狀態（草稿模式）。
* 最終資料提交後清楚明確，方便以報表或查詢方式分別產出三個部門所需資訊。

**二、建議的資料庫架構設計（兩張表設計法）**

**🌳 資料庫表格概觀：**

1. **新人到職資料主表**（員工資料暫存表，欄位齊全）
2. **正式提交後記錄表**（正式表，資料完整不可任意修改）

**✅ 表格設計說明**

**🔸 表1：新人到職資料主表（暫存編輯用，含所有欄位）**

| **欄位名稱** | **型態與建議** | **備註** |
| --- | --- | --- |
| ID（PK） | 字串或整數流水號 | EM+日期流水（如EM20250617001） |
| 員工姓名 | VARCHAR |  |
| 填寫單位 | VARCHAR |  |
| 填寫人員 | VARCHAR |  |
| 到職日期 | DATE |  |
| 身分類型 | VARCHAR (正職、計時、約聘) |  |
| 薪資資訊 | JSON 或 VARCHAR | 複合資訊可JSON或多欄位 |
| 軟硬體設備 | JSON | 電腦規格、軟體需求 |
| 通訊需求 | VARCHAR | Email、帳號相關 |
| 狀態 | VARCHAR（草稿、已送出） | 控制填寫流程 |
| 修改日期 | TIMESTAMP | 記錄最後異動時間 |
| 建立日期 | TIMESTAMP | 初次建立時間 |

* **PK建議：** 以規則編碼，如 **EM+YYYYMMDD+流水號**，易於識別與追蹤。

**🔸 表2：正式提交後記錄表（唯讀，最終產出用）**

| **欄位名稱** | **型態與建議** | **備註** |
| --- | --- | --- |
| ID（PK） | VARCHAR | 同表1 PK |
| 資料內容 | JSON | 完整提交後所有資料 |
| 提交日期 | TIMESTAMP | 正式提交的時間戳記 |
| 提交人員 | VARCHAR | 正式提交的人員資訊 |

* **此表建議設定僅可新增不可修改（唯讀存檔），保持正式資料不可任意變動。**

**三、操作流程建議（從填寫到正式提交）**

流程說明：

javascript

複製編輯

填寫人登入網站→填寫或修改資料（主表，狀態=草稿）

↓

資料確認完成後→點擊「提交」（主表狀態更新為「已送出」）

↓

同時將完整資料JSON複製到正式記錄表（提交表）

↓

正式表依部門需求，以報表或查詢輸出三個部門需求資料

**📌 優點：**

* 主表允許草稿狀態隨意編輯、不會產生重複記錄。
* 正式表明確保存最終狀態，供三部門精準讀取。
* 可用單一PK貫穿整個流程，方便日後追蹤與關聯查詢。

**四、部門資料產出建議**

你提到需要輸出給**財務、人資、資訊**三個部門：

* 採用SQL View或自動化報表工具（如Crystal Reports、BI工具、Excel VBA）自動將正式提交表內資料以部門需求自動篩選、輸出Excel或PDF表單：
  + 財務部View
  + 人資部View
  + 資訊部View

**範例SQL View：**

sql

複製編輯

CREATE VIEW 人資部新人到職資訊 AS

SELECT

JSON\_VALUE(資料內容, '$.ID') AS 員工編號,

JSON\_VALUE(資料內容, '$.員工姓名') AS 員工姓名,

JSON\_VALUE(資料內容, '$.到職日期') AS 到職日期,

JSON\_VALUE(資料內容, '$.身分類型') AS 身分類型,

JSON\_VALUE(資料內容, '$.薪資資訊') AS 薪資資訊

FROM 正式提交後記錄表;

此方式靈活、高效且後續維護容易。

**五、實務建議與注意事項**

1. **PK設計**：
   * PK採用**EM+日期流水號**，保證獨特性與易記性。
   * 若使用整數流水號，也須加上日期欄位作為輔助識別。
2. **資料儲存方式**：
   * 建議使用 **JSON欄位** 處理複雜、多欄位資訊（如設備與薪資），後續擴充容易。
3. **流程追蹤欄位（狀態控制）**：
   * 狀態欄位控制流程節點（草稿→已送出），防止重複送出。
4. **資料庫權限設定**：
   * 主表可增刪改（草稿階段），正式提交後表僅新增（唯讀），資料安全性佳。

**🖥 實務架構圖示意（推薦）：**

sql

複製編輯

填寫頁面 →【主表（暫存資料）】↔ 可反覆修改

↓（點擊送出後）

【正式提交記錄表（正式儲存）】

↓

(SQL View/報表工具產出)

↙ ↓ ↘

財務部 人資部 資訊部

**🚩 我的下一步建議：**

若你同意這樣的資料庫設計與規劃方式，建議：

* ✅ 我可進一步協助你產出更明確的資料表建立SQL。
* ✅ 提供你流程圖與視覺化示意圖（A4清晰呈現）。
* ✅ 若有其他需求（如SQL View語法設計、JSON資料存取範例），我亦可協助提供。

你希望從哪個步驟繼續深入協助你？

**你說：**

給我程式語法該怎麼寫唯獨草稿的部分該怎麼設計

**ChatGPT 說：**

以下為你提供 **資料庫語法設計與實務程式邏輯範例**，包含兩個主要資料表的建立、狀態控管機制及唯讀正式提交的範例語法，搭配簡明且清楚的解說：

**🔹 一、資料表設計 SQL 語法範例**

**🌱 1. 主表（允許修改的草稿表）**

sql

複製編輯

CREATE TABLE NewEmployeeDraft (

ID VARCHAR(20) PRIMARY KEY,

EmployeeName VARCHAR(50) NOT NULL,

Department VARCHAR(50),

FilledBy VARCHAR(50),

StartDate DATE,

EmployeeType VARCHAR(20),

SalaryInfo JSON,

EquipmentInfo JSON,

CommunicationInfo VARCHAR(255),

Status VARCHAR(10) DEFAULT 'Draft', -- 草稿(Draft)、已送出(Submitted)

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

**📌 說明：**

* **狀態欄位** Status 控制紀錄目前狀態。
* **UpdatedAt** 自動記錄最後更新時間。

**🌳 2. 正式唯讀提交表**

sql

複製編輯

CREATE TABLE NewEmployeeOfficial (

ID VARCHAR(20) PRIMARY KEY,

DataContent JSON NOT NULL,

SubmittedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

SubmittedBy VARCHAR(50)

);

**📌 說明：**

* **DataContent** 儲存正式提交時的完整JSON資料。
* 此表格**不提供UPDATE操作**，僅允許新增、查詢、匯出。

**🔹 二、唯讀機制程式邏輯範例（以 PHP 為例）**

**🖥 範例情境：**

* **草稿時期允許新增修改。**
* **送出後變更狀態為 Submitted，同時寫入正式表格**。

php

複製編輯

// 假設有資料庫連線 $db (mysqli 或 PDO)

// 1. 更新草稿表的狀態為「已送出」

function submitDraft($id, $submittedBy, $db) {

// 先撈取草稿內容

$query = $db->prepare("SELECT \* FROM NewEmployeeDraft WHERE ID = ?");

$query->execute([$id]);

$draftData = $query->fetch(PDO::FETCH\_ASSOC);

if (!$draftData) {

return "草稿不存在！";

}

if ($draftData['Status'] === 'Submitted') {

return "此草稿已送出，不能重複送出！";

}

// 更新草稿狀態為 Submitted

$update = $db->prepare("UPDATE NewEmployeeDraft SET Status='Submitted' WHERE ID=?");

$update->execute([$id]);

// 寫入正式表（唯讀表）

$insert = $db->prepare("INSERT INTO NewEmployeeOfficial (ID, DataContent, SubmittedBy) VALUES (?, ?, ?)");

$jsonData = json\_encode($draftData); // 將整筆資料轉成 JSON

$insert->execute([$id, $jsonData, $submittedBy]);

return "提交成功！";

}

**🔹 三、前端草稿唯讀狀態控管範例**

**邏輯說明：**

* 草稿狀態可任意修改，送出後鎖定畫面，顯示唯讀。

php

複製編輯

// 撈取狀態，控制畫面元素是否可編輯

$id = $\_GET['id'];

$query = $db->prepare("SELECT Status FROM NewEmployeeDraft WHERE ID=?");

$query->execute([$id]);

$status = $query->fetchColumn();

$isEditable = ($status !== 'Submitted');

?>

<!-- HTML 範例 -->

<form method="POST">

<input type="text" name="EmployeeName" <?= $isEditable ? '' : 'readonly' ?> >

<input type="date" name="StartDate" <?= $isEditable ? '' : 'readonly' ?> >

<!--其他欄位相同方式處理-->

<?php if ($isEditable): ?>

<button type="submit">儲存草稿</button>

<button type="submit" formaction="submit.php">正式送出</button>

<?php else: ?>

<span>資料已送出，不可再修改。</span>

<?php endif; ?>

</form>

**🔹 四、權限設計補充建議（資料庫層面）**

**🛡 確保正式表僅新增唯讀的資料庫權限設定範例：**

sql

複製編輯

GRANT SELECT, INSERT ON NewEmployeeOfficial TO your\_user;

REVOKE UPDATE, DELETE ON NewEmployeeOfficial FROM your\_user;

* 如此可確保正式表格不被任意修改或刪除，保障資料完整性。

**🗂 五、實務操作流程摘要（整體整理）**

1. 使用者建立草稿→ 存於 NewEmployeeDraft，狀態為Draft。
2. 使用者可多次修改儲存，無限次修改草稿。
3. 正式提交時，系統：
   * 將 NewEmployeeDraft 狀態改為 Submitted。
   * 同步複製該筆資料到 NewEmployeeOfficial，正式記錄。
4. 正式提交後，前端立即變成唯讀狀態，防止使用者修改。
5. 各部門透過唯讀表格產出的View或報表，分別讀取各自需要的欄位資料。

**🚩 完整範例操作流程圖：**

scss

複製編輯

新增資料（草稿）

↓

儲存暫存 Draft

(NewEmployeeDraft)

↓

多次修改（可反覆編輯）

↓

正式提交

↓

更新 Draft 狀態 Submitted

同步寫入正式唯讀表(NewEmployeeOfficial)

↓

正式唯讀表 (各部門讀取輸出)

以上內容已全面涵蓋：

* **資料表設計**
* **唯讀機制的程式邏輯**
* **範例 PHP 實作**
* **權限控管實務建議**

你可以依照這個架構與範例程式碼快速落地與實作，有任何進一步的需求或問題隨時告訴我！