



03

CHAPTER

設備維修自動化管理系統： Google 表單 與 LineBot 無縫整合

在辦公場域或校園環境中，當燈管損壞、冷氣異常或其他設備故障時，若無一套即時回報並自動通知的機制，往往會因為溝通延遲或資料遺漏而導致修繕進度拖延，甚至影響日常運作。

本章所示範的「設備維修自動化管理系統」，就是針對這樣的需求設計的實務應用案例。透過 Google 表單收集報修資訊，並結合 Google Apps Script 與 LINE Bot，自動將報修內容（包括文字與圖片）即時推送給相關修繕人員。整個系統還能同步更新處理狀態、記錄歷史報修紀錄，以及自動設定照片的瀏覽權限，讓報修流程更透明、更有效率。

在此實作案例中，您將學到：

- 設計具備上傳功能的 Google 表單
- 將資料自動寫入試算表並觸發 LINE 通知
- 利用 GAS 程式控制 LINE Bot 發送訊息
- 自動處理圖片共享權限
- 建立錯誤處理與狀態追蹤機制

本範例非常適合應用於學校總務處、企業設備管理部門、IT 維運單位等場景，是提升報修效率、減少人力負擔的實用範本。

3.1 讓 ChatGPT 協助將想法具體化

在職場中，當對於系統的需求尚未明確時，可以先向 ChatGPT 描述該系統預期的運作情境、可能遇到的狀況、基本需求或概念，將其整理成一個基礎的脈絡。為了讓 ChatGPT 鎖定重點並精準回應，可以在文案中加入如「透過 Google Apps Script 技術」或「自動化」這類關鍵字，明確指出您希望採用的技術或解決方法。這樣能幫助 ChatGPT 快速理解您的需求並轉化為系統所需的成果，進一步將想法初步具體化。以下為提問的過程及 ChatGPT 的回應：



起初，我面臨的困境是，報修流程過於繁瑣且不夠即時。當同事遇到設備故障時，通常會透過紙本、口頭通知或填寫 Google 表單，而我再將這些報修資訊手動輸入到維修的 LINE 群組中使維修人員可以在獲得訊息後進行修繕動作。這樣的做法不僅效率低下，還容易出現遺漏或延遲的情況。此外，當報修者提供給我損壞照片時，其他相關人員無法即時檢視，必須我手動轉發，這讓整個流程變得更加複雜。

因此，我希望能建置一個更自動化的系統，將所有報修資料統一透過 Google 表單收集，並能自動推送到 LINE 群組，讓相關人員能夠第一時間獲得報修詳情和照片，以利進行處理。此外，我還希望系統能記錄發送到 LINE 的狀態，確保每次報修訊息都確實傳達到位，並且不再需要手動管理文件分享權限，使照片能夠被相關人員直接檢視。

有鑑於此，我想透過 Google Apps Script 技術來自動化這個過程，不僅能解決手動處理的效率問題，還能確保報修訊息的即時性和完整性。這樣的系統能大幅減少人工操作中的錯誤，提高整個報修流程的效率。

針對上述描述，幫我具體化我的需求。



根據您的描述，我將您的需求具體化如下：

自動化報修流程系統需求：

1. Google 表單報修資料收集：

- 所有報修資料透過 Google 表單統一收集，包括故障描述、損壞照片上傳等。
- 表單設計要簡潔、直觀，便於同事快速填寫報修資訊。

2. 自動推送報修訊息至 LINE 群組：

- 報修表單一旦提交，系統自動將報修內容（如設備名稱、問題描述、上傳的照片等）推送到指定的 LINE 群組，確保維修人員能即時取得報修資訊。
- 支援多圖傳輸，維修人員可直接查看故障照片。

3. 報修訊息狀態追蹤：

- 系統需記錄每次報修訊息推送到 LINE 群組的狀態，確保訊息成功發送。
- 若推送失敗，系統應有通知機制以便即時調整處理。

4. 無須手動文件分享權限：

- 報修表單中的照片和文件可自動設定適當的查看權限，讓 LINE 群組內的成員可直接檢視，無須手動管理檔案權限。

5. 報修紀錄保存與查詢：

- 系統將所有報修資料（包括圖片和相關訊息）儲存在 Google 試算表或 Google 雲端硬碟中，方便後續查詢和追蹤。
- 提供查詢功能，方便管理者查看歷史報修紀錄及維修狀況。

6. 減少人工操作錯誤，提升效率：

- 透過自動化流程，減少手動轉發和手動記錄的錯誤風險，確保報修訊息即時且完整地傳達到維修人員手中。
- 系統應能及時推送訊息，避免延遲。

這樣的需求定義將為您的自動化報修系統提供清晰的方向和功能範圍。若有其他細節或功能需要補充，請告訴我！

3.2 Google 表單與 LineBot 建置

當對「設備維修自動化管理系統」已有較完善的構思與系統運作邏輯後，在向 ChatGPT 提出進階問題並取得 Google Apps Script (GAS) 程式碼之前，需先建置系統所需的各類檔案，便於後續程式的連結與使用。

該系統的主要運作方式是透過 Google 表單收集報修資訊，並將其存儲於 Google 試算表中，作為推播訊息的來源。一旦試算表中新增了資料，系統將觸發推播功能，透過 LINE Bot 向相關人員發送通知。因此，需要完成的前置作業如下：

〔 3.2.1 Google 表單建置與調整 Google 試算表 〕

在報修過程中筆者設計了一個 Google 表單，讓報修者能透過表單直接填寫報修內容，並加入檔案上傳功能，使報修者能夠附上損壞情形的照片。這不僅有助於減少溝通誤解，還能有效統計與管理維修紀錄。以下為 Google 表單建置與 Google 試算表調整的具體步驟：

STEP01 在雲端硬碟中，點擊「新增 > 資料夾」。



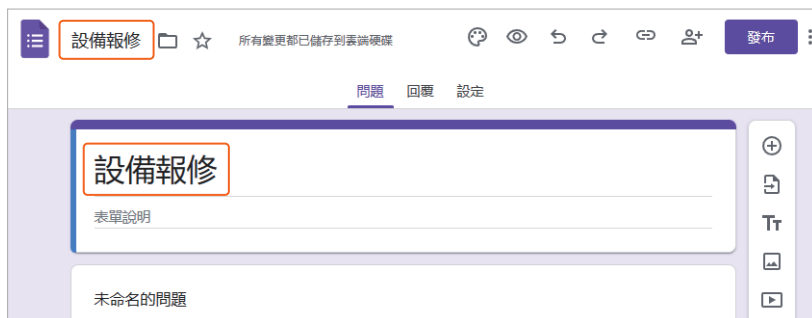
STEP02 將資料夾名稱命名為「設備維修自動化管理系統」。



STEP03 進入該資料夾後，點擊「滑鼠右鍵 > Google 表單」。



STEP04 將 Google 表單名稱與主旨改為「設備報修」。



STEP05 將預設的問題進行修改，設定如下：

- ◆ 問答類型：下拉式選單。
- ◆ 問題主旨：單位。
- ◆ 回答選項：(1) 教務處、(2) 總務處、(3) 學務處。
- ◆ 必 填：是。

This screenshot shows the configuration for a Google Form question titled '單位' (Unit). The question type is set to '下拉式選單' (Dropdown menu). The options listed are: 1. 教務處 (Academic Affairs Office), 2. 總務處 (General Affairs Office), 3. 學務處 (Student Affairs Office), and 4. 新增選項 (Add new option). The '必填' (Required) toggle is turned on. Numbered callouts indicate: 1. Question type dropdown, 2. Question title, 3. Option list, and 4. Required toggle.

STEP06 點擊「新增問題」按鈕後，設定如下：

- ◆ 問答類型：簡答。
- ◆ 問題主旨：姓名。
- ◆ 必 填：是。

This screenshot shows the configuration for a Google Form question titled '姓名' (Name). The question type is set to '簡答' (Short answer). The '簡答文字' (Short answer text) field is visible. The '必填' (Required) toggle is turned on. Numbered callouts indicate: 1. Question type dropdown, 2. Question title, 3. Required toggle, and 4. Question type dropdown.

STEP07 點擊「新增問題」按鈕後，設定如下：

- ◆ 問答類型：下拉式選單。
- ◆ 問題主旨：損壞地點。
- ◆ 回答選項：(1)A101、(2)A102、(3)A103。
- ◆ 必 填：是。

This screenshot shows the configuration for a Google Form question titled '損壞地點' (Damage location). The question type is set to '下拉式選單' (Dropdown menu). The options listed are: 1. A101, 2. A102, and 3. A103. The '必填' (Required) toggle is turned on. Numbered callouts indicate: 1. Question type dropdown, 2. Question title, 3. Option list, and 4. Required toggle.

3.3 使用 ChatGPT 完成系統規劃

首先，根據前述章節的內容，手動將 (1) 背景描述、(2) 任務目標、(3) 輸入來源這三項內容整理完畢。接續，將其作為系統的參考資料而提供給 ChatGPT，使其協作提供 (4) 任務拆解、(5) 邏輯與條件兩點結果。最終自動化設定的部分會由 GAS 所提供的觸發條件來手動設定。

以下為提問的過程及 ChatGPT 的回應：



扮演一位資深的系統工程師，善於在系統開發前將各功能與運作邏輯描述清晰以利於後續的開發工作。因此，根據下列三點對於系統的描述與用途之內容，提供給我有關「任務拆解」與「邏輯與條件」這兩項的說明。

1. 背景描述

該系統是一個基於 Google Apps Script (GAS) 的設備報修通知自動化系統，旨在處理由 Google 表單收集的報修資訊，並透過 LINE Bot 將報修通知推送到特定的使用者或群組。同時，它會處理報修照片的共享權限，確保相關人員可以查看照片。

2. 任務目標

- (1) 自動接收 Google 表單中的報修資訊。
- (2) 確保損壞照片的共享權限設定為“公開可檢視”。
- (3) 組織報修資訊為標準化訊息格式，並透過 LINE Bot 推送通知。
- (4) 更新試算表的“發送狀態”欄位，標記通知已發送。

3. 輸入來源

來源為 Google 表單提交的回應結果並存儲於試算表中，欄位包含：

- (1) 時間戳記：表單提交時間。
- (2) 單位：報修者所屬單位。
- (3) 姓名：報修者姓名。
- (4) 損壞地點：設備損壞的地點。
- (5) 損壞設備：報修的設備名稱。
- (6) 損壞狀況說明：設備損壞的具體描述。

- (7) 損壞照片：設備損壞的 Google 雲端硬碟照片 URL。
 - (8) 發送狀態：記錄通知是否已發送（“已發送”或空白）。
-

根據上述內容，ChatGPT 所回覆的 4 至 5 兩點內容，以及加入自動化需求的說明如下：

4. 處理過程（任務拆解）

- (1) 接收表單回應：程式觸發時，讀取試算表的最新一行數據。
- (2) 檢查發送狀態：確認是否需要處理（只處理“發送狀態”欄位為空白的行）。
- (3) 取得並處理損壞照片：
 - ◆ 從照片 URL 中取得 Google 雲端硬碟文件 ID。
 - ◆ 使用 GAS API 將文件設定為“任何人可檢視”。
 - ◆ 取得公開的檢視連結。
- (4) 統整訊息：將表單回應格式化為結構化訊息，包含時間戳記、單位、報修者、損壞地點、設備名稱、損壞描述和照片連結。
- (5) 發送訊息：將訊息推播至指定的 LINE 使用者或群組。
- (6) 更新試算表：將“發送狀態”設為“已發送”。

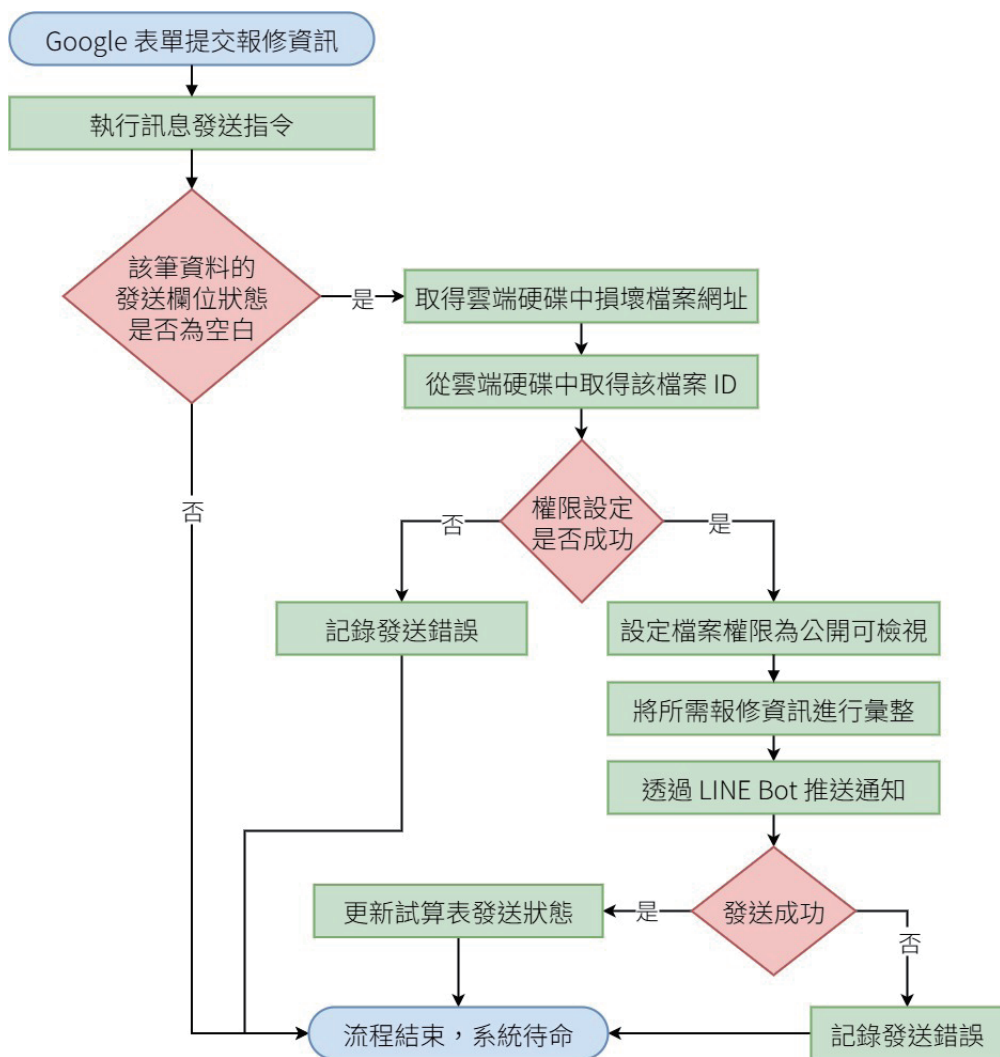
5. 邏輯與條件

- (1) 條件檢查：若“發送狀態”為空，則進行處理；否則跳過。
- (2) 若損壞照片 URL 無法取得文件 ID，返回錯誤提示。
- (3) 權限處理：確保損壞照片的共享權限設定正確，避免通知接收者無法檢視。
- (4) 錯誤處理：若 LINE Bot 發送訊息失敗，記錄錯誤訊息以便調整並測試。

6. 自動化設定

- (1) 觸發器設定：將表單提交事件設為觸發器（Triggers）。
- (2) 持續運行：每次提交表單後，系統會自動執行程式，無須人工干預。

筆者根據上述的結果，繪製一個簡單易懂的設備維修自動化管理系統架構圖，將系統的運作流程圖像化，幫助您快速理解其邏輯與功能，以下為系統架構圖。



3.5 完整程式碼與解說

此程式碼的功能是當報修者提交報修資料後，會將該筆報修資料自動推播到指定的 LINE 中，同時若報修資料中有圖片或影片時，則會自動將權限設為公開以利維修者正常查看。其完整程式碼與程式解說如下：

📁 檔案路徑：ch03 > code.docx

```
01 function onFormSubmit(e) {
02     const sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
03     const lastRow = sheet.getLastRow();
04     const rowData = sheet.getRange(lastRow, 1, 1, sheet.getLastColumn()).
getValues()[0];
05     const timeZone = "Asia/Taipei";
06     const [
07         timestampValue,
08         unit,
09         reporter,
10         location,
11         equipment,
12         description,
13         photoUrl,
14         status
15     ] = rowData;
16     const timestamp = new Date(timestampValue);
17     const formattedTimestamp = Utilities.formatDate(timestamp, timeZone,
"MM/dd HH:mm");
18     if (!status) {
19         const fileId = extractFileId(photoUrl);
20         const sharedPhotoUrl = makeFilePublic(fileId);
21         const message = `報修通知：
22 時間戳記：${formattedTimestamp}
23 單位：${unit}
24 報修者：${reporter}
25 損壞地點：${location}
26 損壞設備：${equipment}
27 損壞狀況說明：${description}
28 損壞照片：${sharedPhotoUrl}`;
29         sendLineNotify(message);
30         sheet.getRange(lastRow, 8).setValue("已發送");
31     }
```

```
75     }  
76   };
```

✓ 當 Google 表單提交時觸發

- 01 定義函數 onFormSubmit，當 Google 表單提交時自動觸發執行，用於處理新提交的報修資訊。
- 02 定義變數 sheet，取得當前工作表中的主要工作表物件（表單回應 1），作為讀取表單提交數據的來源。
- 03 定義變數 lastRow，取得試算表中最後一行的行數，表示最近提交的數據所在。
- 04 定義變數 rowData，取得最後一行的所有欄位數據，並將其存儲為一維陣列。
- 05 定義變數 timeZone，時區為「亞洲 / 台北」，確保日期與時間處理符合當地時間。
- 06 ~ 15 使用解構賦值將最後一行數據分別賦值給多個變數：
 - (1) timestampValue：提交表單的時間戳記。
 - (2) unit：報修單位。
 - (3) reporter：報修者姓名。
 - (4) location：損壞發生的地點。
 - (5) equipment：損壞的設備名稱。
 - (6) description：損壞狀況的描述。
 - (7) photoUrl：照片的 URL（上傳到 Google 雲端硬碟）。
 - (8) status：當前報修的狀態。
- 16 定義變數 timestamp，將 timestampValue 轉換為日期物件，便於後續格式化處理。
- 17 定義變數 formattedTimestamp，並使用 Utilities.formatDate 將日期格式化為「MM/dd HH:mm」，方便在訊息中使用。
- 18 檢查 status 是否為空（尚未處理）。若為空，表示該報修尚未發送 LINE 通知。
- 19 定義變數 fileId，調用 extractFileId 函數，從照片 URL 中取得 Google 雲端硬碟檔案的唯一 ID。
- 20 定義變數 sharedPhotoUrl，調用 makeFilePublic 函數，將檔案設為公開並取得可共享的照片 URL。
- 21 ~ 28 定義報修通知的內容，將所有相關資訊（例如時間、單位、損壞設備等）組合成一段文字訊息。
- 29 調用 sendLineNotify 函數，將報修通知內容發送到 LINE。
- 30 在試算表的第 8 欄（狀態欄）更新為「已發送」，表示該報修已處理。

✓ 從 Google 雲端硬碟 URL 提取文件 ID

- 33 定義函數 extractFileId，並從 url 中取得檔案 ID。
- 34 定義變數 match，使用正規表達式查找字串中符合模式的部分。

3.6 常見問題

設備維修自動化管理系統透過整合 Google 表單與 LINE Bot，實現了維修報修流程的即時性與自動化。然而，該系統涉及數據收集、文件處理、權限管理和通知推送等多個環節，在開發和運行中可能出現數據格式不符、權限設定不當或 LINE Bot 推送失敗等問題。以下列出常見錯誤及其解決建議，以提高系統穩定性和使用者體驗。

1. **表單數據格式錯誤：**表單中提交的數據格式（如日期或文字描述）與程式預期格式不符，導致系統處理失敗。
2. **照片權限設定錯誤：**系統未正確設定照片的共享權限，導致維修人員無法查看。
3. **通知推送失敗：**LINE Bot Token 配置錯誤或網絡連接不穩定，導致推送失敗。
4. **試算表欄位變更：**試算表欄位順序調整後，導致程式無法正確解析數據。因此在系統開發前固定欄位順序，並添加欄位檢查機制。
5. **觸發器未正確配置：**表單提交事件觸發器未正確設定，導致系統無法自動執行。