

# 交接文件-5

■ 建立時間	@2025年8月22日 上午2:34
■ 狀態	未完成

## 生產管理系統 V5.0 - 專案交接文件

文件版本: 3.0

更新時間: 2025年8月22日

**文件目的:** 本文件旨在詳細記錄「生產管理系統 V5.0」專案截至當前的開發狀態、已完成的功能、遇到的核心瓶頸與解決思路，以及下一步的明確計畫，以便順利交接給下一個 AI 開發對話。

### 1. 已安裝的套件與元件

專案的核心技術棧為 Next.js 14 (App Router), React 18, TypeScript 5, Tailwind CSS 3, 以及 Firebase (Authentication, Firestore, Cloud Functions v2)。

核心依賴套件:

- `firebase`: 用於前端與 Firebase 服務的互動。
- `react-hook-form` & `zod`: 用於處理表單邏輯與驗證。
- `lucide-react`: 提供圖示。
- `sonner`: 用於顯示現代化的提示訊息 (Toast)。
- `@tanstack/react-table`: **(近期新增)** 用於建構功能強大的資料表格，是目前卡關的核心。

UI 元件 (shadcn/ui):

- 基礎: `button`, `input`, `label`, `card`, `avatar`, `badge`, `checkbox`
- 互動: `dialog`, `alert-dialog`, `dropdown-menu`, `select`, `tooltip`
- 佈局: `table`
- 提示: `sonner`

### 2. 完整的資料夾與檔案結構

前端 ( `src/` )

- `app/dashboard/`: 儀表板的核心佈局 ( `layout.tsx` ) 與各功能模組頁面。

- `personnel/` , `suppliers/` , `materials/` , `fragrances/` , `product-series/` , `products/` : 各基礎資料的管理頁面，包含列表與新增/編輯對話框。
- `purchase-orders/` : 採購單管理模組，包含列表 `page.tsx` 與詳情頁 `[id]/page.tsx` 。
- `production-calculator/` : 生產計算機頁面 ( `page.tsx` ) 。
- `work-orders/` : **(當前工作目錄)** 工單管理模組。
  - `page.tsx` : 工單列表頁，目前存在編譯錯誤。
  - `columns.tsx` : 定義工單列表的欄位，存在類型錯誤。
  - `data-table.tsx` : 可重用的表格元件，存在類型錯誤。
- `components/ui/` : `shadcn/ui` 元件。
- `context/AuthContext.tsx` : 全域的用戶身份驗證狀態管理。
- `lib/firebase.ts` : Firebase 前端 SDK 的初始化設定。

## 後端 ( `functions/` )

- `src/api/` : 存放所有按功能模組拆分的 Cloud Functions 。
  - `purchaseOrders.ts` , `inventory.ts` , `workOrders.ts` 等，每個檔案對應一個核心功能。
- `src/utils/auth.ts` : 存放共用的權限檢查函式 (如 `ensureIsAdmin` ) 。
- `src/index.ts` : 主入口檔案，僅負責初始化與匯出 `api/` 中的所有函式。

## 3. 目前已完成跟未完功能，並準備要做什麼

- **已完功能:**
  - **F2: 權限與人員管理:** 基礎 CRUD 完成。
  - **F3: 基礎資料管理:** 所有子模組 (供應商、物料、產品等) 的 CRUD 功能已全部開發完成。
  - **F4: 採購管理:** 從建立採購需求、下單、到收貨入庫的完整流程已開發並通過端到端測試。
  - **F5: 庫存與盤點管理:** 已在物料管理頁面成功整合盤點模式，可更新庫存並生成異動日誌。
  - **F6.1: 配方計算與工單生成:** 後端 `createWorkOrder` 函式與前端「生產計算機」頁面均已完成。
- **未完成/進行中:**

- **F6.2: 工單管理:** 正在開發工單列表頁 ( `/dashboard/work-orders` )，但目前被嚴重的 TypeScript 類型錯誤卡住，導致頁面無法編譯。
- **下一步計畫:**
  1. **首要任務:** 解決 `work-orders` 資料夾內所有檔案的 TypeScript 類型錯誤，讓工單列表頁能成功渲染。
  2. **接著開發:** 建立工單詳情頁 ( `/dashboard/work-orders/[id]` )，這是下一個核心開發任務。

## 4. 部署狀態

- **已完成部署:**
  - **Firebase Cloud Functions:** 所有後端 API (包含 `users` , `purchaseOrders` , `inventory` , `workOrders` 等) 均已成功部署。
- **未完成部署:**
  - Firebase Hosting (前端 Next.js 應用)。
  - GitHub Repository 與 CI/CD 流程。

## 5. 目前遇到什麼未解決的問題，準備解決思路是什麼

- **核心問題:** 在開發工單列表頁時，`page.tsx`、`columns.tsx`、`data-table.tsx` 三個檔案出現了頑固的 TypeScript 類型錯誤，導致編譯失敗。`page.tsx` 最終顯示「找不到模組 `./columns` 或其對應的型別宣告」，這是因為 `columns.tsx` 因內部類型錯誤而無法被編譯。
- **問題根源分析:**
  1. **套件缺失 (已解決):** 最初，專案缺少 `@tanstack/react-table` 這個核心依賴，導致 TypeScript 無法找到任何表格相關的類型定義。**此套件已被要求安裝 ( `npm install @tanstack/react-table` )。**
  2. **嚴格類型檢查 (根本原因):** 專案的 `tsconfig.json` 啟用了 `"strict": true` 模式。這要求 `@tanstack/react-table` 的所有泛型參數 (如 `column` , `row` , `headerGroup` , `cell` 的上下文) 都必須有極其明確的類型定義，否則會被推斷為 `any` 並導致編譯錯誤。
  3. **`AuthContext` 類型不完整 (已解決):** `layout.tsx` 中的 `logout` 函式報錯，是因為 `AuthContext.tsx` 中 `useAuth` hook 的返回類型定義不完整。
- **解決思路:**
  1. **確認套件安裝:** 下一個對話需要先確認 `@tanstack/react-table` 已成功安裝在 `node_modules` 中。

2. **提供 `tsconfig.json`**：必須將 `tsconfig.json` 提供給下一個對話，讓他完全理解專案的編譯上下文和路徑別名 (`@/*`) 設定。
3. **嚴格定義類型**：需要對 `work-orders` 資料夾下的 `columns.tsx` 和 `data-table.tsx` 進行細緻的類型修正，確保所有來自 `@tanstack/react-table` 的回呼函式參數都有明確的類型註釋。
4. **修正 `AuthContext`**：需要修正 `AuthContext.tsx`，在 `AuthContextType` 和 `useAuth` 的返回 `value` 中明確包含 `logout` 函式。

## 6. 需要提供下一個對話什麼檔案

為了讓下一個對話快速掌握問題，請務必提供以下檔案的**完整內容**：

1. **`tsconfig.json`：(最重要)** 讓 AI 了解編譯器的嚴格規則和路徑設定。
2. **`src/context/AuthContext.tsx`**：用於解決 `layout.tsx` 中的 `logout` 函式類型錯誤。
3. **`src/app/dashboard/layout.tsx`**：包含 `logout` 錯誤的檔案。
4. **`src/app/dashboard/work-orders/page.tsx`**：發生「找不到模組」錯誤的檔案。
5. **`src/app/dashboard/work-orders/columns.tsx`**：存在大量 `any` 類型錯誤的根源之一。
6. **`src/app/dashboard/work-orders/data-table.tsx`**：存在大量 `any` 類型錯誤的根源之二。

## 7. 版本提醒

- `shadcn/ui` 的 CLI 指令已更新為 `npx shadcn-ui@latest add ...`。
- 專案使用 `sonner` 元件庫來顯示提示訊息 (Toasts)。
- 所有 Firebase Cloud Functions 均使用 v2 語法。
- `@tanstack/react-table` 是一個解決當前問題所必需的核心套件。

## 8. 重要過程描述

我們成功地完成了從採購到庫存盤點的完整流程，證明了核心架構的穩定性。在進入「工單管理」模組開發時，我們決定採用 `@tanstack/react-table` 來建構功能豐富的列表。然而，在整合過程中，由於未能充分考慮到專案 `tsconfig.json` 的嚴格模式，導致了持續的、難以解決的類型錯誤。

經過多次嘗試，從修正程式碼邏輯到最後定位問題根源為「**缺少套件依賴**」和「**不夠嚴格的類型定義**」，這個過程是解決當前瓶頸的關鍵。下一個對話必須從這個根本原因入手，而不是重複之前無效的程式碼邏輯修改。