

Dash RMS 配方管理系統 - 工作說明書 (Statement of Work)

文件版本: 1.0 編制日期: 2025-10-11 系統版本: v0.5.4 編制單位: Dash Project Team

目錄

- 專案概述
- 系統架構
- 功能規格
- 技術規格
- 資料模型
- API 介面規格
- 品質要求
- 交付標準
- 測試要求
- 部署要求
- 維護與支援
- 專案時程
- 驗收標準

1. 專案概述

1.1 專案背景

Dash RMS (Recipe Management System, 配方管理系統) 是一套模組化的配方管理與簽核流程系統，旨在協助製造業（特別是化學瀘網、食品、製藥等產業）進行配方的標準化管理、版本控制、品質追蹤與審核流程自動化。

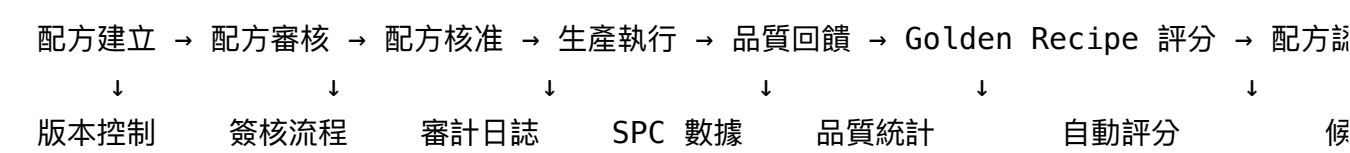
1.2 專案目標

- 配方標準化管理**：建立統一的配方格式與管理流程
- 簽核流程自動化**：支援多層級、多模式的審核流程（單簽、並簽、串簽、條件分支）
- 版本控制**：完整的配方版本追蹤與回滾功能
- 品質追蹤**：整合 SPC（統計製程管制）數據，自動評分與 Golden Recipe 認證
- 產業模組化**：核心系統可快速適配不同產業需求
- 系統整合**：與 EAP（Engineering Automation Platform）及 ECU（Equipment Control Unit）無縫整合

1.3 使用者角色

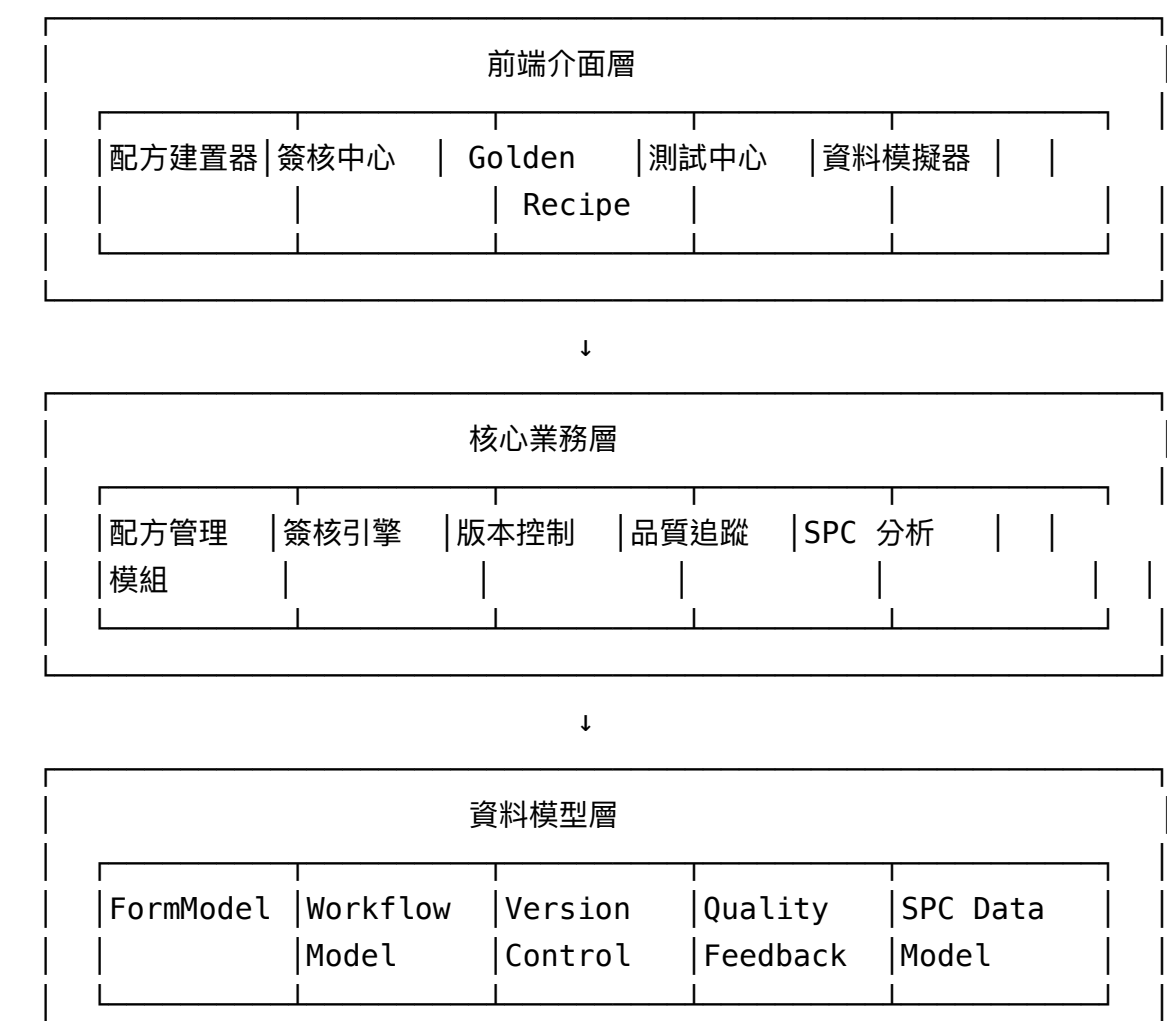
角色	職責	系統權限
一般員工	建立配方、提交審核申請	配方建立、申請提交
主管	審核配方、核准/退回申請	簽核審批、配方查看
系統管理員	系統設定、使用者管理、資料維護	所有權限
品質工程師	品質數據分析、Golden Recipe 認證	品質數據管理、認證審核

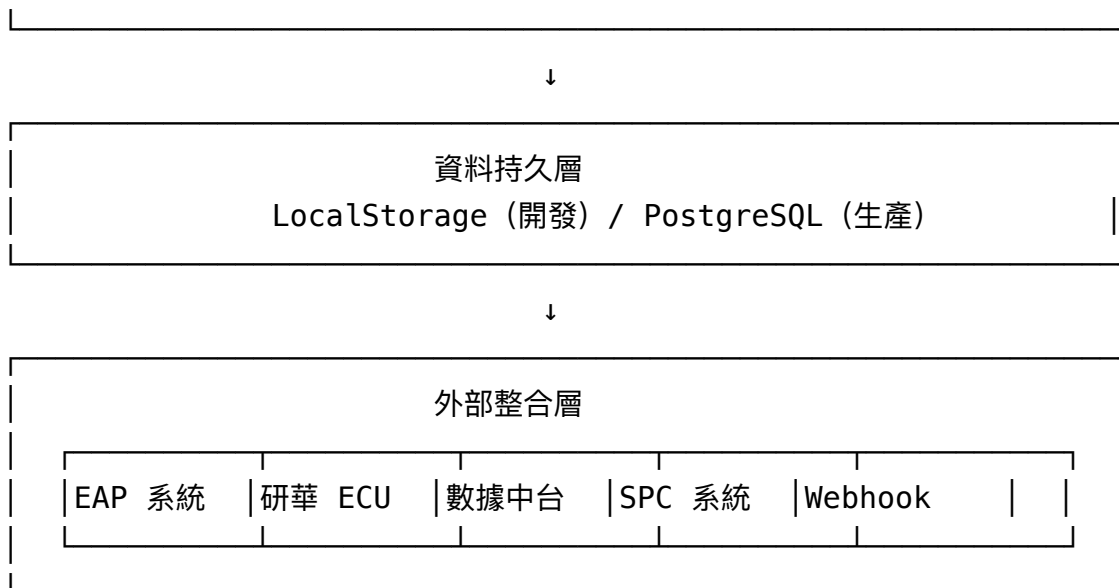
1.4 業務流程



2. 系統架構

2.1 整體架構圖





2.2 模組化架構

系統採用「核心 + 產業模組」架構：

```
src/
├── core/                # 核心功能（通用，可跨產業重用）
│   ├── auth/           # 認證系統
│   ├── workflow/       # 簽核流程引擎
│   ├── audit/          # 審計日誌
│   └── recipe-base/    # 配方基礎架構
├── industries/         # 產業模組（客製化）
│   ├── amc-filter/     # AMC 化學濾網產業
│   │   ├── fields.config.js # 欄位定義
│   │   ├── validations.js  # 驗證規則
│   │   └── templates.js    # 範本定義
│   ├── food/           # 食品製造（未來）
│   └── pharma/         # 製藥產業（未來）
├── api/                # API 介面層
│   ├── endpoints.js    # API 端點定義
│   └── adapter.js      # API 適配器
└── config/             # 系統配置
    └── industry.config.js # 產業配置檔
```

架構優勢：

- 80% 程式碼可跨產業重用
- 新產業部署快速（僅需新增產業模組）
- 各產業獨立演進，互不影響
- 維護成本低

2.3 技術堆疊

層級	技術選型	版本要求	說明
前端框架	Vanilla JavaScript (ES6+)	ES2020+	無框架依賴，輕量化
建置工具	Vite	7.1.9+	快速開發與建置
樣式系統	CSS Variables + Utility Classes	-	自訂 Design System
拖拽功能	@dnd-kit	latest	流程設計器
圖表繪製	Chart.js	4.x	SPC 管制圖
PDF 生成	jsPDF	latest	報表匯出
畫布處理	html2canvas	latest	截圖功能
開發資料庫	LocalStorage	-	開發與 POC 階段
生產資料庫	PostgreSQL	14+	生產環境建議
後端框架	Node.js + Express	18+	API 服務（未來）
容器化	Docker	latest	部署封裝（未來）

3. 功能規格

3.1 配方管理模組

3.1.1 配方建置器 (RecipeBuilderPage)

功能描述：

- 動態載入產業模組欄位定義
- 支援 4 大欄位群組：基本資訊、材料配方、製程參數、品質標準
- 自動驗證配方資料（必填欄位、數值範圍、格式檢查）
- 配方範本系統（快速套用預設值）
- 配方編號自動生成（格式：AMC-YYYY-XXX）
- 版本號自動管理（新建 1.0，更新 +0.1）

欄位範例（AMC 化學濾網）：

群組	欄位名稱	類型	必填	驗證規則
基本資訊	產品名稱	text	✓	長度 1-100
基本資訊	濾網類型	select	✓	活性碳濾網/化學濾網/複合濾網
材料配方	化學藥劑	text	✓	-

材料配方	濃度 (%)	number	✓	0-100
製程參數	反應溫度 (°C)	number	✓	20-200
製程參數	壓力 (bar)	number	✓	1-10
品質標準	過濾效率 (%)	number	✓	90-100
品質標準	使用壽命 (月)	number	✓	12-36

技術實作：

- `src/pages/RecipeBuilderPage.js` - 主頁面元件
- `src/industries/amc-filter/fields.config.js` - 欄位定義
- `src/industries/amc-filter/validations.js` - 驗證規則
- `src/industries/amc-filter/templates.js` - 範本定義

3.1.2 配方列表頁面 (FormsPage)

功能描述：

- 顯示所有配方清單 (支援分頁)
- 搜尋與篩選功能 (依配方編號、名稱、狀態)
- 顯示配方編號、版本、Golden Score
- 配方詳情查看、編輯、刪除
- 版本歷史查看與回滾

資料欄位：

- 配方編號 (font-mono 顯示)
- 配方名稱
- 版本號 (badge 樣式)
- Golden Score (顏色標記：綠色 ≥ 92 ，黃色 85-91，灰色 < 85)
- 建立時間
- 最後更新時間
- 操作按鈕 (查看、編輯、刪除、版本歷史)

3.1.3 範本管理系統 (TemplatesPage)

功能描述：

- 範本 CRUD 功能 (建立、讀取、更新、刪除)
- 範本版本控制
- 範本匯入/匯出 (JSON 格式)
- 範本使用統計追蹤
- 範本預覽功能

範本資料結構：

```

{
  id: "TMPL-001",
  name: "標準 AMC 濾網配方",
  description: "適用於一般室內空氣淨化",
  category: "amc-filter",
  tags: ["標準", "室內"],
  version: "1.0",
  defaultValues: {
    productName: "標準化學濾網",
    filterType: "化學濾網",
    chemicalAgent: "活性氧化鋁",
    concentration: 15,
    // ... 更多欄位
  },
  usageCount: 42,
  createdAt: "2025-10-01T00:00:00.000Z",
  updatedAt: "2025-10-11T00:00:00.000Z"
}

```

3.2 簽核流程模組

3.2.1 流程設計器 (WorkflowBuilderPage)

功能描述：

- 拖拽式流程節點設計
- 支援節點類型：
 - **開始節點**：流程起點
 - **單簽節點**：單一簽核人
 - **並簽節點**：多人同時簽核（全部同意才通過）
 - **串簽節點**：多人依序簽核（任一同意即可進行下一關）
 - **條件分支節點**：根據條件分流（例如：金額 > 100萬 → 總經理核准）
 - **結束節點**：流程終點
- 節點連線設定
- 簽核人員選擇（支援選擇使用者或角色）
- 流程驗證（檢查是否有未連接節點、循環路徑等）

節點資料結構：

```

{
  id: "node-001",
  type: "approval", // start, approval, parallel, sequential, condition,
  label: "部門主管審核",
  position: { x: 100, y: 100 },

```

```
    config: {  
      approvers: ["user-002", "user-003"], // 簽核人員 ID  
      mode: "all", // all: 並簽, any: 串簽  
      condition: null // 條件分支邏輯  
    }  
  }  
}
```

3.2.2 簽核中心 (ApprovalCenterPage)

功能描述：

- 我的申請：查看自己提交的申請單
- 待我簽核：需要我審核的申請單
- 已簽核：我已處理的申請單
- 所有申請：系統所有申請單（管理員）
- 申請單詳情查看
- 核准/退回操作（支援意見填寫）
- 申請單狀態追蹤
- 簽核歷史記錄

申請單狀態：

- draft：草稿
- pending：審核中
- approved：已核准
- rejected：已退回
- cancelled：已取消

簽核操作：

```
// 核准  
approveApplication(instanceId, {  
  comment: "審核通過",  
  approver: currentUser,  
  timestamp: Date.now()  
})
```

```
// 退回  
rejectApplication(instanceId, {  
  comment: "配方濃度過高，請調整",  
  approver: currentUser,  
  timestamp: Date.now()  
})
```

3.3 版本控制模組

3.3.1 版本控制引擎 (versionControl.js)

功能描述：

- 自動版本追蹤（每次儲存配方時自動建立版本）
- 版本歷史查看
- 版本差異比對（找出兩個版本間的差異）
- 回滾到指定版本
- 版本清理（保留最近 N 個版本，清除舊版本）
- 版本匯出/匯入（JSON 格式）

版本資料結構：

```
{
  id: "form-001_v1728456789000",
  entityId: "form-001",
  version: 5,
  data: { /* 配方完整資料快照 */ },
  comment: "調整濃度參數",
  createdBy: "張三",
  createdById: "user-001",
  createdAt: "2025-10-11T10:30:00.000Z",
  size: 2048 // JSON 字串長度 (bytes)
}
```

版本儲存策略：

- 新建配方時：建立版本 1
- 更新配方時：自動建立新版本（版本號 +1）
- 回滾操作時：將回滾作為新版本儲存（不刪除歷史）

版本比對演算法：

```
// 遞迴比較兩個物件，找出新增、修改、刪除的欄位
findDifferences(obj1, obj2) {
  // 回傳差異列表：
  // [
  //   { path: "concentration", type: "modified", oldValue: 15, newValue:
  //   { path: "testMethod", type: "added", oldValue: undefined, newValue
  // ]
}
```


3.3.2 版本歷史 UI

功能描述：

- 時間軸式版本列表
- 顯示版本號、建立時間、建立人、註解
- 點選版本查看完整資料
- 比對兩個版本的差異（視覺化標記）
- 回滾到指定版本（需確認操作）

3.4 Golden Recipe 模組

3.4.1 品質回饋數據模型（QualityFeedbackModel）

功能描述：

- 接收生產執行後的品質數據
- 支援多種數據來源：SPC 系統、手動輸入、檔案匯入
- 儲存品質指標：良率、過濾效率、使用壽命、缺陷率、CPK、穩定性
- 自動觸發配方評分更新

品質數據結構：

```
{
  id: "QF-1728456789000",
  recipeId: "form-001",
  recipeVersion: "1.0",
  executionId: "EXEC-001",
  batchNo: "BATCH-20251011-001",
  qualityMetrics: {
    yieldRate: 98.5,           // 良率 (%)
    filterEfficiency: 99.2,    // 過濾效率 (%)
    lifespan: 28,              // 使用壽命 (月)
    defectRate: 1.5,           // 缺陷率 (%)
    cpk: 1.67,                 // 製程能力指標
    stabilityScore: 95         // 穩定性評分 (0-100)
  },
  testResults: {
    passed: true,
    testDate: "2025-10-11T10:00:00.000Z",
    inspector: "王品管"
  },
  productionInfo: {
    line: "產線 A",
    shift: "早班",
  }
}
```

```
    operator: "李操作員"
  },
  issues: [], // 品質異常記錄
  notes: "正常生產",
  createdAt: "2025-10-11T10:00:00.000Z",
  source: "SPC-SYSTEM" // 數據來源
}
```

3.4.2 Golden Recipe 評分引擎

評分公式：

Golden Score =
良率 × 30% +
過濾效率 × 25% +
壽命達標率 × 20% +
CPK × 10 × 15% +
穩定性 × 10%

其中：

- 良率、效率、穩定性：0-100 分
- 壽命達標率 = (平均壽命 / 24個月) × 100%，允許超過 100%
- CPK × 10：將 CPK 值（通常 1.0-2.0）轉換為百分比

自動認證條件（系統自動認證為 Golden Recipe）：

1. 最低樣本數 ≥ 10 批次
2. 時間跨度 ≥ 30 天
3. Golden Score ≥ 92 分
4. 最近 20 批次平均良率 ≥ 97%
5. 平均 CPK ≥ 1.33
6. 無重大品質異常

候選配方條件（推薦為候選，需人工審核）：

- Golden Score ≥ 85 分
- 尚未符合自動認證條件

3.4.3 Golden Recipe 管理頁面（GoldenRecipePage）

功能描述：

- 顯示所有 Golden Recipe 清單
- 顯示 Golden Recipe 候選清單
- 配方品質統計圖表

- 手動認證 Golden Recipe（需審核主管核准）
- 降級 Golden Recipe（取消認證）
- 品質趨勢分析

手動認證流程：

品質工程師提出認證申請 → 指定審核主管 → 主管審核 → 核准/退回 → 正式認證

3.5 SPC 統計製程管制模組

3.5.1 SPC 數據模型

功能描述：

- 接收製程量測數據
- 支援多種數據來源：研華 ECU、數據中台、API、Webhook、手動輸入、檔案匯入
- 儲存關鍵參數：溫度、壓力、濃度、流量等
- 計算統計指標：平均值、標準差、上下限

SPC 數據結構：

```
{  
  id: "SPC-001",  
  recipeId: "form-001",  
  batchNo: "BATCH-20251011-001",  
  parameter: "temperature",    // 量測參數  
  value: 85.2,                 // 量測值  
  timestamp: "2025-10-11T10:00:00.000Z",  
  operator: "李操作員",  
  equipment: "反應槽 A",  
  notes: ""  
}
```

3.5.2 製程能力分析

分析指標：

- **Cp**（製程能力指數）： $Cp = (USL - LSL) / (6\sigma)$
- **Cpk**（製程能力指標）： $Cpk = \min[(USL - \mu) / 3\sigma, (\mu - LSL) / 3\sigma]$
- **Pp**（製程績效指數）：長期製程能力
- **Ppk**（製程績效指標）：長期製程能力（考慮偏移）

其中：

- USL: Upper Specification Limit（規格上限）

- LSL: Lower Specification Limit (規格下限)
- μ : 平均值
- σ : 標準差

判定標準：

Cpk 值	製程能力	說明
$Cpk < 1.0$	不足	需改善
$1.0 \leq Cpk < 1.33$	尚可	可接受
$1.33 \leq Cpk < 1.67$	良好	符合標準
$Cpk \geq 1.67$	優秀	六標準差水準

3.5.3 SPC 管制圖

圖表類型：

- **X-bar & R Chart** (平均值-全距管制圖)
- **X-bar & S Chart** (平均值-標準差管制圖)
- **I-MR Chart** (個別值-移動全距管制圖)

異常檢測規則 (Western Electric Rules)：

1. 任一點超出管制界限 ($\pm 3\sigma$)
2. 連續 9 點在中心線同側
3. 連續 6 點遞增或遞減
4. 連續 14 點交替上下
5. 連續 2/3 點落在 2σ 區域
6. 連續 4/5 點落在 1σ 區域

技術實作：

- 使用 Chart.js 繪製管制圖
- 即時計算管制界限
- 標記異常點 (紅色)
- 顯示趨勢線

3.6 權限控制模組

系統採用「**權限表 + 角色對應**」的 RBAC (Role-Based Access Control) 架構，實現靈活且易於擴展的權限管理。

架構組成：

1. 權限定義檔 (src/config/permissions.config.js)
 - 定義所有頁面權限 (17 個) - 格式：page:xxx

- 定義所有功能權限（22 個） - 格式：action:xxx
- 按群組分類（工作區、配方管理、流程管理、組織管理、系統管理、開發工具）
- 提供權限查詢方法

2. 角色配置檔 (src/config/roles.config.js)

- 定義角色與權限的對應關係
- 支援系統內建角色與自訂角色
- 儲存自訂角色至 localStorage
- 向後兼容舊版角色名稱

3. 權限管理器 (src/utils/permissionManager.js)

- 集中處理權限檢查邏輯
- 權限快取機制（提升效能）
- 動態選單生成（根據角色）
- 資源擁有權檢查

4. 權限管理頁面 (src/pages/PermissionsPage.js)

- 視覺化角色與權限管理
- 僅系統管理員可訪問
- 支援查看角色權限列表
- 未來擴展：建立/編輯/刪除自訂角色

3.6.1 權限管理器 (permissionManager.js)

功能描述：

- 基於權限配置檔實現靈活的權限控制
- 頁面訪問權限檢查
- 動作執行權限檢查
- 動態選單生成（根據角色）
- 資源擁有權檢查
- 向後兼容舊版 API

角色定義：

角色	英文	說明	徽章顏色
系統管理員	Admin	完整系統管理權限	紅色
主管	Manager	管理與審核權限	藍色
一般員工	Employee	基本操作權限	綠色

權限矩陣：

功能模組	系統管理員	主管	一般員工
------	-------	----	------

工作區

總覽	✓	✓	✓
我的申請	✓	✓	✓
待我簽核	✓	✓	×

管理功能

配方管理	✓	✓	✓ (僅檢視)
配方範本	✓	✓	✓ (僅檢視)
產業模組配置	✓	×	×
Golden Recipe	✓	✓	✓ (僅檢視)
SPC 品質管制	✓	✓	✓ (僅檢視)
審核流程	✓	✓	×
人員管理	✓	×	×
部門管理	✓	×	×

系統功能

報表統計	✓	✓	×
操作日誌	✓	×	×
權限管理	✓	×	×
測試中心	✓	×	×
模擬中心	✓	×	×

動作權限：

系統使用新的權限代碼格式（action:xxx），向後兼容舊版格式。

// 系統管理員可執行的動作（共 22 個）

```
permissions: [  
  'action:create-form',           // 建立配方  
  'action:edit-form',            // 編輯所有配方  
  'action:delete-form',          // 刪除配方  
  'action:create-template',       // 建立範本  
  'action:edit-template',        // 編輯範本  
  'action:delete-template',       // 刪除範本  
  'action:create-workflow',       // 建立流程  
  'action:edit-workflow',         // 編輯流程  
  'action:delete-workflow',       // 刪除流程
```

```

'action:approve-application', // 審核申請
'action:reject-application', // 退回申請
'action:certify-golden', // 認證 Golden Recipe
'action:revoke-golden', // 撤銷 Golden Recipe
'action:view-spc', // 查看 SPC 數據
'action:edit-spc', // 編輯 SPC 數據
'action:delete-spc', // 刪除 SPC 數據
'action:manage-users', // 管理使用者
'action:manage-departments', // 管理部門
'action:view-reports', // 查看報表
'action:view-logs', // 查看日誌
'action:manage-industry-config', // 管理產業配置
'action:manage-permissions', // 管理權限
'action:run-tests', // 執行測試
'action:use-simulator' // 使用模擬器
]

```

// 主管可執行的動作 (共 12 個)

```

permissions: [
  'action:create-form',
  'action:edit-form', // 可編輯所有配方
  'action:view-forms',
  'action:export-form',
  'action:create-template',
  'action:edit-template',
  'action:approve-application',
  'action:reject-application',
  'action:certify-golden',
  'action:view-spc',
  'action:edit-spc',
  'action:view-reports'
]

```

// 一般員工可執行的動作 (共 4 個)

```

permissions: [
  'action:create-form',
  'action:edit-own-form', // 僅能編輯自己的配方
  'action:view-forms',
  'action:view-spc'
]

```

向後兼容： 系統保留舊版 action 名稱支援（使用底線格式），自動轉換為新格式：

- create_form → action:create-form

- edit_own_form → action:edit-own-form
- certify_golden_recipe → action:certify-golden

3.6.2 權限檢查機制

頁面訪問控制：

系統在 Router 層級實施權限檢查，當使用者嘗試訪問頁面時：

1. 檢查使用者是否已登入
2. 取得使用者角色
3. 查詢該角色是否有權限訪問該頁面
4. 若無權限，顯示「權限不足」頁面並記錄日誌

```
// Router 權限檢查
async handleRoute() {
  const path = window.location.hash.slice(1) || '/';

  // 權限檢查
  if (!permissionManager.canAccessPage(path)) {
    this.showAccessDenied(path);
    return;
  }

  // 載入頁面
  const content = await route();
  this.layout.setContent(content);
}
```

動態選單生成：

Sidebar 根據使用者角色動態生成選單，只顯示有權限的項目：

```
// 取得可訪問的選單
const accessibleMenu = permissionManager.getAccessibleMenu();

// 動態生成 HTML
const menuHTML = accessibleMenu.map(section => `
<div class="sidebar-section">
  <h3>${section.title}</h3>
  <ul>
    ${section.items.map(item => `
      <li><a href="#${item.path}>${item.label}</a></li>
    `).join('')}
  </ul>
</div>
`).join('');
```



```
    </ul>
  </div>
  `).join('');
```

資源擁有權檢查：

對於配方等資源，系統檢查是否為使用者自己建立：

```
// 檢查是否可編輯配方
canEditForm(form) {
  const user = authService.getCurrentUser();

  // 管理員和主管可編輯所有配方
  if (user.role === '系統管理員' || user.role === '主管') {
    return true;
  }

  // 一般員工只能編輯自己的配方
  return user.id === form.createdBy;
}

// 檢查是否可刪除配方
canDeleteForm(form) {
  // 只有系統管理員可刪除
  return user.role === '系統管理員';
}
```

3.6.3 使用者介面權限呈現

Header 角色徽章：

Header 顯示當前使用者資訊與角色徽章：

- 系統管理員：紅色徽章，顯示「管理員」
- 主管：藍色徽章，顯示「主管」
- 一般員工：綠色徽章，顯示「員工」

權限不足頁面：

當使用者嘗試訪問無權限頁面時，顯示友善的錯誤訊息：



權限不足

您的帳號（張三，角色：一般員工）無法訪問此頁面。

請求路徑：/simulator

[\[返回首頁\]](#)

切換使用者：

系統提供使用者切換功能（開發/測試用），切換後自動：

1. 更新 Header 顯示
2. 重新生成 Sidebar 選單
3. 重新載入當前頁面

3.7 測試與模擬模組

3.7.1 自動化測試系統（TestRunner）

功能描述：

- 完整的測試執行引擎
- 支援測試劇本（Scenario-based Testing）
- 豐富的斷言函數
- 可視化測試報告

支援的斷言函數：

<code>assertEqual(actual, expected)</code>	// 相等斷言
<code>assertDeepEqual(actual, expected)</code>	// 深度相等斷言
<code>assertNotEqual(actual, expected)</code>	// 不相等斷言
<code>assertTrue(value)</code>	// 真值斷言
<code>assertFalse(value)</code>	// 假值斷言
<code>assertThrows(fn, errorMessage)</code>	// 例外斷言
<code>assertGreaterThan(value, threshold)</code>	// 大於斷言
<code>assertLessThan(value, threshold)</code>	// 小於斷言
<code>assertArrayLength(arr, length)</code>	// 陣列長度斷言
<code>assertObjectHasKey(obj, key)</code>	// 物件屬性斷言

測試劇本範例：

```
{
  name: "配方建立與版本控制測試",
  description: "測試配方建立、更新、版本追蹤功能",
  steps: [
    {
```

```

    name: "建立新配方",
    action: () => {
      const form = new FormModel({
        name: "測試配方",
        description: "測試用配方",
        fields: [...]
      });
      form.save("初始版本");
      return form;
    },
    assertions: [
      { type: "assertEqual", params: ["{{result.version}}", 1] },
      { type: "assertGreaterThan", params: ["{{result.id.length}}", 0]
    ]
  },
  {
    name: "更新配方",
    action: () => {
      form.name = "測試配方（已更新）";
      form.save("更新名稱");
      return form;
    },
    assertions: [
      { type: "assertEqual", params: ["{{result.version}}", 2] }
    ]
  }
]
}

```

測試報告格式：

```

{
  scenarioName: "配方建立與版本控制測試",
  totalSteps: 5,
  passedSteps: 5,
  failedSteps: 0,
  passRate: 100,
  duration: 145, // ms
  results: [
    {
      stepName: "建立新配方",
      passed: true,
      assertions: [

```

```

        { type: "assertEqual", passed: true, message: "版本號正確" },
        { type: "assertGreaterThan", passed: true, message: "ID 已生成" }
    ]
}
]
}

```

3.6.2 資料模擬器 (DataSimulator)

功能描述：

- 快速生成測試資料
- 支援生成：部門、使用者、配方、工作流程、申請單、SPC 數據、品質回饋
- 支援壓力測試（生成大量資料）
- 支援清除所有模擬資料

生成策略：

```

// 部門生成：隨機中文部門名稱
generateDepartments(count) // 研發部、生產部、品保部等

// 使用者生成：隨機中文姓名 + 職位
generateUsers(count) // 張三（研發工程師）、李四（品保主管）等

// 配方生成：符合產業模組定義的完整配方
generateForms(count) // 包含所有必填欄位，數值在合理範圍內

// 品質數據生成：支援指定品質分佈
generateProductionQualityData({
    superhighQualityCount: 1, // 超高品質 (≥95分，自動認證)
    highQualityCount: 2,     // 高品質 (85-91分，候選)
    mediumQualityCount: 0,    // 中等品質 (80-84分)
    lowQualityCount: 1,       // 低品質 (<80分)
    minBatches: 25,           // 最小批次數 (建議 ≥10 符合認證)
    maxBatches: 25            // 最大批次數
})

```

Golden Recipe 測試工具：

```

// 快速測試（一鍵完成所有流程）
runGoldenQuickTest() {
    // 1. 清除舊資料

```

```
// 2. 生成 4 個配方
// 3. 生成混合品質數據 (1超高 + 2高 + 1低)
// 4. 自動評分
// 5. 自動認證
// 6. 回報結果
}
```

```
// 分析評分 (顯示所有配方的評分詳情)
```

```
debugGoldenScores()
```

```
// 檢查候選 (顯示所有候選配方)
```

```
checkGoldenCandidates()
```

```
// 檢查已認證 (顯示所有 Golden Recipe)
```

```
checkCertifiedGolden()
```

4. 技術規格

4.1 前端技術規格

4.1.1 專案結構

```
RMS/
├─ index.html          # HTML 入口
├─ vite.config.js      # Vite 建置配置
├─ package.json        # 專案配置
├─ .gitignore          # Git 忽略檔案
├─ README.md           # 專案說明
├─ CHANGELOG.md        # 版本變更記錄
├─ docs/               # 文件目錄
│   └─ GOLDEN_RECIPE_DESIGN.md
│   └─ TEMPLATE_MANAGEMENT_PLAN.md
│   └─ 工作說明書_RMS配方管理系統.md
├─ public/             # 靜態資源
└─ src/
    ├─ main.js          # 程式入口
    ├─ router.js        # 路由管理
    ├─ components/      # 元件目錄
    │   └─ common/      # 通用元件
    │       └─ Button.js
    │       └─ Modal.js
    │       └─ Table.js
```


JavaScript 規範：

- 使用 ES6+ 語法（箭頭函數、解構賦值、模組化等）
- 使用 `const / let`，避免 `var`
- 函數命名：camelCase（例如：`getUserById`）
- 類別命名：PascalCase（例如：`FormModel`）
- 常數命名：UPPER_SNAKE_CASE（例如：`STORAGE_KEY`）
- 檔案命名：PascalCase for components（例如：`RecipeBuilderPage.js`）
- 檔案命名：camelCase for utilities（例如：`dataModel.js`）

註解規範：

```
/**
 * 儲存新版本
 * @param {string} entityId - 實體 ID
 * @param {Object} data - 資料內容
 * @param {string} comment - 版本註解
 * @returns {Object} - 版本資訊
 */
saveVersion(entityId, data, comment = '') {
  // 實作邏輯
}
```

錯誤處理：

```
try {
  const result = doSomething();
  return { success: true, data: result };
} catch (error) {
  console.error('操作失敗:', error);
  return { success: false, error: error.message };
}
```

4.1.3 樣式規範

CSS Variables（全域變數）：

```
:root {
  /* 主色 */
  --color-primary: #3b82f6;
  --color-primary-dark: #2563eb;

  /* 灰階 */
}
```

```

--color-gray-50: #f9fafb;
--color-gray-100: #f3f4f6;
--color-gray-200: #e5e7eb;
/* ... */

/* 語意色 */
--color-success: #10b981;
--color-warning: #f59e0b;
--color-danger: #ef4444;

/* 字體 */
--font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, s
--font-mono: "SF Mono", Monaco, Consolas, monospace;

/* 間距 */
--spacing-xs: 4px;
--spacing-sm: 8px;
--spacing-md: 16px;
--spacing-lg: 24px;
--spacing-xl: 32px;
}

```

Utility Classes :

```

/* 文字 */
.text-xs { font-size: 12px; }
.text-sm { font-size: 14px; }
.text-base { font-size: 16px; }
.text-center { text-align: center; }
.font-mono { font-family: var(--font-mono); }

/* Badge 樣式 */
.badge { display: inline-block; padding: 2px 8px; border-radius: 4px; }
.badge-success { background: #10b981; color: white; }
.badge-warning { background: #f59e0b; color: white; }

```

4.2 後端技術規格 (未來擴展)

4.2.1 API 服務架構

Node.js + Express

— /api/v1/recipes	# 配方管理 API
— /api/v1/workflows	# 工作流程 API
— /api/v1/approvals	# 簽核 API
— /api/v1/quality	# 品質數據 API
— /api/v1/spc	# SPC 數據 API
— /api/v1/users	# 使用者管理 API
— /api/v1/webhooks	# Webhook 端點

4.2.2 資料庫設計 (PostgreSQL)

主要資料表：

- forms：配方定義
- form_versions：配方版本
- form_instances：申請單
- workflows：工作流程定義
- approval_history：簽核歷史
- quality_feedbacks：品質回饋數據
- spc_data：SPC 量測數據
- users：使用者
- departments：部門
- audit_logs：操作日誌

5. 資料模型

5.1 配方模型 (FormModel)

```
{
  id: "form-001",           // 配方 ID (自動生成)
  name: "高效化學濾網配方", // 配方名稱
  description: "適用於高污染環境", // 配方描述
  fields: [                // 配方欄位陣列
    {
      name: "recipeNo",      // 欄位名稱
      label: "配方編號",    // 欄位標籤
      value: "AMC-2025-001", // 欄位值
      type: "text",          // 欄位類型
      required: true         // 是否必填
    },
    // ... 更多欄位
  ],
  industry: "amc-filter",   // 產業模組
}
```

```

    createdAt: 1728456789000,           // 建立時間 (timestamp)
    updatedAt: 1728456789000,          // 更新時間 (timestamp)

    // Golden Recipe 相關
    goldenScore: 95.3,                  // Golden 評分 (0-100)
    qualityStats: {                    // 品質統計
      totalExecutions: 25,             // 總執行次數
      avgYield: 98.5,                  // 平均良率 (%)
      avgEfficiency: 99.2,             // 平均效率 (%)
      avgLifespan: 28,                 // 平均壽命 (月)
      avgCpk: 1.67,                    // 平均 CPK
      avgDefectRate: 1.5,              // 平均缺陷率 (%)
      lastExecutionDate: "2025-10-11T10:00:00.000Z",
      qualityTrend: "improving"        // improving, stable, declining
    },
    isGolden: true,                     // 是否為 Golden Recipe
    goldenCertifiedAt: "2025-10-11T00:00:00.000Z",
    goldenCertifiedBy: "SYSTEM-AUTO",   // SYSTEM-AUTO or 使用者 ID
    goldenCertificationReason: "自動認證：符合所有 Golden Recipe 標準",
    certificationStatus: "approved",    // pending, approved, rejected
    reviewers: [                        // 審核者清單 (手動認證時)
      {
        id: "user-002",
        name: "李主管",
        status: "approved",             // pending, approved, rejected
        approvedAt: "2025-10-11T00:00:00.000Z",
        comment: "符合標準"
      }
    ]
  }
}

```

5.2 工作流程模型 (WorkflowModel)

```

{
  id: "workflow-001",
  name: "標準配方審核流程",
  formId: "form-001",                 // 關聯的配方 ID
  nodes: [
    {
      id: "node-001",
      type: "start",                   // 節點類型
      label: "開始",
      position: { x: 100, y: 100 }
    }
  ]
}

```

```

},
{
  id: "node-002",
  type: "approval",           // 單簽節點
  label: "部門主管審核",
  position: { x: 100, y: 200 },
  config: {
    approvers: ["user-002"], // 簽核人 ID
    mode: "single"           // single, all, any
  }
},
{
  id: "node-003",
  type: "parallel",          // 並簽節點
  label: "品保與研發並簽",
  position: { x: 100, y: 300 },
  config: {
    approvers: ["user-003", "user-004"],
    mode: "all"              // 所有人都要同意
  }
},
{
  id: "node-004",
  type: "condition",         // 條件分支節點
  label: "金額判斷",
  position: { x: 100, y: 400 },
  config: {
    condition: {
      field: "totalCost",
      operator: ">",
      value: 1000000
    },
    truePath: "node-005",    // 條件為真時的下一節點
    falsePath: "node-006"   // 條件為假時的下一節點
  }
},
{
  id: "node-005",
  type: "end",
  label: "結束",
  position: { x: 100, y: 500 }
}
],
connections: [              // 節點連線

```

```

    { from: "node-001", to: "node-002" },
    { from: "node-002", to: "node-003" },
    { from: "node-003", to: "node-004" },
    { from: "node-004", to: "node-005" }
  ],
  createdAt: 1728456789000,
  updatedAt: 1728456789000
}

```

5.3 申請單模型 (FormInstanceModel)

```

{
  id: "instance-001",
  applicationNo: "20251011-001",           // 申請編號
  formId: "form-001",                     // 配方 ID
  formName: "高效化學濾網配方",          // 配方名稱
  workflowId: "workflow-001",             // 工作流程 ID
  workflowName: "標準配方審核流程",      // 流程名稱
  applicant: "張三",                     // 申請人
  department: "研發部",                  // 部門
  data: {                                 // 申請資料 (配方完整資料)
    recipeNo: "AMC-2025-001",
    productName: "高效化學濾網",
    // ... 所有配方欄位
  },
  status: "pending",                      // draft, pending, approved, reject
  currentNodeId: "node-002",              // 當前節點 ID
  history: [                              // 簽核歷史
    {
      nodeId: "node-001",
      nodeName: "開始",
      userId: "user-001",
      userName: "張三",
      action: "submit",
      comment: "提交配方審核",
      timestamp: 1728456789000
    },
    {
      nodeId: "node-002",
      nodeName: "部門主管審核",
      userId: "user-002",
      userName: "李主管",
      action: "approve",

```

```

        comment: "核准通過",
        timestamp: 1728456889000
    }
],
parallelState: {},           // 並簽狀態追蹤
sequentialState: {},         // 串簽狀態追蹤
createdAt: 1728456789000,
updatedAt: 1728456889000
}

```

5.4 版本控制模型 (Version)

```

{
    id: "form-001_v1728456789000",
    entityType: "form",           // form, workflow, template
    entityId: "form-001",
    version: 3,                   // 版本號
    data: {                       // 完整資料快照
        id: "form-001",
        name: "高效化學濾網配方",
        fields: [/* ... */],
        // ... 所有欄位
    },
    comment: "調整濃度參數從 15% 到 18%",
    createdBy: "張三",
    createdById: "user-001",
    createdAt: "2025-10-11T10:30:00.000Z",
    size: 2048                    // JSON 字串大小 (bytes)
}

```

5.5 品質回饋模型 (QualityFeedback)

參見 [3.4.1 品質回饋數據模型](#)

5.6 SPC 數據模型 (SPCData)

```

{
    id: "spc-001",
    recipeId: "form-001",
    batchNo: "BATCH-20251011-001",
    parameter: "temperature",    // 量測參數名稱
}

```

```

parameterName: "反應溫度",
value: 85.2, // 量測值
unit: "°C", // 單位
target: 85, // 目標值
usl: 90, // 規格上限
lsl: 80, // 規格下限
ucl: 88, // 管制上限 (+3σ)
lcl: 82, // 管制下限 (-3σ)
timestamp: "2025-10-11T10:00:00.000Z",
operator: "李操作員",
equipment: "反應槽 A",
shift: "早班",
line: "產線 A",
notes: "",
source: "ECU" // ECU, API, MANUAL, FILE
}

```

5.7 使用者模型 (UserModel)

```

{
  id: "user-001",
  account: "user001", // 帳號
  name: "張三", // 姓名
  department: "研發部", // 部門
  role: "一般員工", // 一般員工, 主管, 系統管理員
  email: "user001@example.com",
  phone: "0912-345-678",
  status: "在職", // 在職, 離職
  password: "hashed_password", // 加密後的密碼 (生產環境需使用 bcrypt)
  createdAt: 1728456789000,
  updatedAt: 1728456789000
}

```

5.8 操作日誌模型 (AuditLog)

```

{
  id: "log-001",
  timestamp: 1728456789000,
  userId: "user-001",
  userName: "張三",
  userAccount: "user001",

```

```
    action: "create",                // login, logout, create, update, d
    module: "form",                 // form, workflow, application, use
    targetId: "form-001",
    targetName: "高效化學濾網配方",
    result: "success",              // success, failure
    ipAddress: "192.168.1.100",
    details: "建立配方：高效化學濾網配方",
    changes: {                      // 變更內容 (update 時)
      oldValue: { concentration: 15 },
      newValue: { concentration: 18 }
    }
  }
}
```

6. API 介面規格

6.1 配方管理 API

6.1.1 查詢配方清單

端點：GET /api/v1/recipes

查詢參數：

- status (string, optional): 配方狀態 (all, approved, pending, draft)
- industry (string, optional): 產業模組 (amc-filter, food, pharma)
- isGolden (boolean, optional): 是否為 Golden Recipe
- page (number, optional): 頁碼，預設 1
- limit (number, optional): 每頁筆數，預設 20

回應範例：

```
{
  "success": true,
  "data": {
    "recipes": [
      {
        "id": "form-001",
        "recipeNo": "AMC-2025-001",
        "name": "高效化學濾網配方",
        "version": "1.2",
        "status": "approved",
        "goldenScore": 95.3,

```

```
        "isGolden": true,
        "createdAt": "2025-10-01T00:00:00.000Z",
        "updatedAt": "2025-10-11T00:00:00.000Z"
    }
],
"total": 42,
"page": 1,
"limit": 20
}
}
```

6.1.2 查詢單一配方

端點：GET /api/v1/recipes/:id

回應範例：

```
{
  "success": true,
  "data": {
    "recipe": {
      "id": "form-001",
      "recipeNo": "AMC-2025-001",
      "name": "高效化學濾網配方",
      "description": "適用於高污染環境",
      "fields": [
        {
          "name": "productName",
          "label": "產品名稱",
          "value": "高效化學濾網",
          "type": "text"
        }
      ],
      "goldenScore": 95.3,
      "qualityStats": {
        "totalExecutions": 25,
        "avgYield": 98.5,
        "avgEfficiency": 99.2
      },
      "isGolden": true,
      "createdAt": "2025-10-01T00:00:00.000Z",
      "updatedAt": "2025-10-11T00:00:00.000Z"
    }
  }
}
```



```
}  
}
```

6.1.3 建立配方

端點：POST /api/v1/recipes

請求本文：

```
{  
  "name": "高效化學濾網配方",  
  "description": "適用於高污染環境",  
  "industry": "amc-filter",  
  "fields": [  
    {  
      "name": "productName",  
      "label": "產品名稱",  
      "value": "高效化學濾網",  
      "type": "text",  
      "required": true  
    }  
  ]  
}
```

回應範例：

```
{  
  "success": true,  
  "data": {  
    "recipe": {  
      "id": "form-042",  
      "recipeNo": "AMC-2025-042",  
      "name": "高效化學濾網配方",  
      "version": "1.0",  
      "createdAt": "2025-10-11T10:00:00.000Z"  
    }  
  }  
}
```

6.1.4 更新配方

端點：PUT /api/v1/recipes/:id

請求本文：

```
{
  "name": "高效化學濾網配方（已優化）",
  "fields": [/* 更新後的欄位 */],
  "versionComment": "優化濃度參數"
}
```

回應範例：

```
{
  "success": true,
  "data": {
    "recipe": {
      "id": "form-042",
      "recipeNo": "AMC-2025-042",
      "name": "高效化學濾網配方（已優化）",
      "version": "1.1",
      "updatedAt": "2025-10-11T11:00:00.000Z"
    }
  }
}
```

6.1.5 匯出配方（供 EAP 使用）

端點：GET /api/v1/recipes/:id/export

回應範例：

```
{
  "success": true,
  "data": {
    "recipeId": "form-001",
    "recipeNo": "AMC-2025-001",
    "version": "1.2",
    "status": "approved",
    "parameters": {
      "productName": "高效化學濾網",
      "filterType": "化學濾網",
      "chemicalAgent": "活性氧化鋁",
      "concentration": 18,
      "carbonType": "椰殼活性碳",
    }
  }
}
```

```

        "carbonWeight": 500,
        "temperature": 85,
        "pressure": 5,
        "mixingTime": 30,
        "curingTime": 24,
        "filterEfficiency": 99.5,
        "lifespan": 30,
        "testMethod": "ASHRAE 145.2"
    },
    "timestamp": 1728456789000,
    "exportedBy": "EAP-SYSTEM",
    "exportedAt": "2025-10-11T10:00:00.000Z"
}
}

```

6.1.6 批次匯出配方

端點：POST /api/v1/recipes/export

請求本文：

```

{
  "recipeIds": ["form-001", "form-002", "form-003"]
}

```

回應範例：

```

{
  "success": true,
  "data": {
    "recipes": [
      {
        "recipeId": "form-001",
        "recipeNo": "AMC-2025-001",
        "parameters": { /* ... */ }
      },
      {
        "recipeId": "form-002",
        "recipeNo": "AMC-2025-002",
        "parameters": { /* ... */ }
      }
    ],
    "total": 3,

```

```
    "exportedAt": "2025-10-11T10:00:00.000Z"
  }
}
```

6.2 簽核流程 API

6.2.1 查詢待我簽核的申請單

端點：GET /api/v1/approvals/pending

查詢參數：

- userId (string, required): 使用者 ID

回應範例：

```
{
  "success": true,
  "data": {
    "applications": [
      {
        "id": "instance-001",
        "applicationNo": "20251011-001",
        "formName": "高效化學濾網配方",
        "applicant": "張三",
        "department": "研發部",
        "currentNode": "部門主管審核",
        "createdAt": "2025-10-11T09:00:00.000Z"
      }
    ],
    "total": 5
  }
}
```

6.2.2 核准申請

端點：POST /api/v1/approvals/:instanceId/approve

請求本文：

```
{
  "userId": "user-002",
  "comment": "審核通過"
}
```

回應範例：

```
{
  "success": true,
  "data": {
    "instance": {
      "id": "instance-001",
      "status": "approved",
      "currentNode": "結束"
    },
    "message": "配方已核准"
  }
}
```

6.2.3 退回申請

端點：POST /api/v1/approvals/:instanceId/reject

請求本文：

```
{
  "userId": "user-002",
  "comment": "配方濃度過高，請調整後重新提交"
}
```

回應範例：

```
{
  "success": true,
  "data": {
    "instance": {
      "id": "instance-001",
      "status": "rejected"
    },
    "message": "申請已退回"
  }
}
```

6.3 品質數據 API

6.3.1 提交品質回饋

端點：POST /api/v1/quality/feedbacks

請求本文：

```
{
  "recipeId": "form-001",
  "batchNo": "BATCH-20251011-001",
  "qualityMetrics": {
    "yieldRate": 98.5,
    "filterEfficiency": 99.2,
    "lifespan": 28,
    "defectRate": 1.5,
    "cpk": 1.67,
    "stabilityScore": 95
  },
  "testResults": {
    "passed": true,
    "testDate": "2025-10-11T10:00:00.000Z",
    "inspector": "王品管"
  },
  "productionInfo": {
    "line": "產線 A",
    "shift": "早班",
    "operator": "李操作員"
  },
  "source": "SPC-SYSTEM"
}
```

回應範例：

```
{
  "success": true,
  "data": {
    "feedback": {
      "id": "QF-1728456789000",
      "recipeId": "form-001",
      "createdAt": "2025-10-11T10:00:00.000Z"
    },
    "updatedRecipe": {
      "goldenScore": 95.3,
      "qualityStats": {
        "totalExecutions": 26,
        "avgYield": 98.4,

```

```
        "avgEfficiency": 99.1
      }
    }
  }
}
```

6.3.2 批次同步品質數據

端點：POST /api/v1/quality/sync

請求本文：

```
{
  "feedbacks": [
    {
      "recipeId": "form-001",
      "batchNo": "BATCH-001",
      "qualityMetrics": { /* ... */ }
    },
    {
      "recipeId": "form-002",
      "batchNo": "BATCH-002",
      "qualityMetrics": { /* ... */ }
    }
  ]
}
```

回應範例：

```
{
  "success": true,
  "data": {
    "total": 50,
    "success": 48,
    "failed": 2,
    "results": [ /* ... */ ]
  }
}
```

6.4 Webhook 通知

6.4.1 配方核准通知

當配方核准完成時，系統自動推送通知到 EAP 系統。

端點：POST http://eap-system/api/recipes/import (EAP 系統提供)

請求本文：

```
{
  "event": "recipe.approved",
  "recipeId": "form-001",
  "recipeNo": "AMC-2025-001",
  "version": "1.2",
  "approvedAt": "2025-10-11T10:00:00.000Z",
  "approvedBy": "李主管",
  "parameters": { /* 完整配方參數 */ },
  "timestamp": 1728456789000
}
```

6.4.2 Golden Recipe 認證通知

當配方自動認證為 Golden Recipe 時，系統推送通知。

端點：POST http://notification-system/api/notify (通知系統提供)

請求本文：

```
{
  "event": "golden_recipe.certified",
  "recipeId": "form-001",
  "recipeNo": "AMC-2025-001",
  "recipeName": "高效化學濾網配方",
  "goldenScore": 95.3,
  "certifiedAt": "2025-10-11T10:00:00.000Z",
  "certifiedBy": "SYSTEM-AUTO",
  "timestamp": 1728456789000
}
```

7. 品質要求

7.1 功能品質

品質指標	要求標準	驗證方法
功能完整性	符合功能規格 100%	功能測試、驗收測試

資料正確性	計算結果誤差 < 0.01%	單元測試、整合測試
流程正確性	簽核流程邏輯正確	流程測試、邊界測試
版本控制	版本追蹤無遺漏	版本測試、回滾測試

7.2 效能品質

效能指標	要求標準	驗證方法
頁面載入時間	< 2 秒（首次載入） < 0.5 秒（快取後）	效能測試工具
API 回應時間	一般查詢 < 500ms 複雜查詢 < 2s	API 壓力測試
資料處理能力	支援 1000+ 配方 支援 10000+ 申請單	壓力測試
並發使用者	支援 100 人同時線上	並發測試

7.3 安全品質

安全指標	要求標準	驗證方法
認證機制	強制登入、Session 管理	安全測試
權限控制	角色權限分離（RBAC）	權限測試
資料加密	密碼 bcrypt 加密	安全審查
SQL 注入防護	使用參數化查詢	滲透測試
XSS 防護	輸入輸出過濾	安全掃描
CSRF 防護	Token 驗證	安全測試
操作日誌	所有敏感操作記錄	審計測試

7.4 可用性品質

可用性指標	要求標準	驗證方法
UI 一致性	遵循 Design System	UI 審查
回饋機制	所有操作提供即時回饋	使用者測試
錯誤處理	友善的錯誤訊息	錯誤測試
響應式設計	支援桌面/平板瀏覽器	相容性測試

7.5 維護性品質

維護性指標	要求標準	驗證方法
程式碼註解	關鍵函數需有 JSDoc	Code Review
模組化程度	單一職責、低耦合	架構審查
測試覆蓋率	核心模組 > 80%	測試報告
文件完整性	README、API 文件、使用手冊	文件審查

8. 交付標準

8.1 交付項目清單

交付項目	說明	格式
原始碼	完整專案原始碼（含註解）	Git Repository
建置檔案	生產環境建置檔案	dist/ 目錄
資料庫腳本	資料庫建立、遷移腳本	SQL 檔案
部署指南	環境設定、部署步驟	Markdown / PDF
API 文件	完整 API 規格說明	Swagger / Postman
使用手冊	使用者操作手冊	PDF
測試報告	測試結果與覆蓋率報告	HTML / PDF
原始碼授權	MIT License	LICENSE 檔案

8.2 文件交付要求

8.2.1 技術文件

- **README.md**：專案說明、快速開始、安裝指南
- **CHANGELOG.md**：版本變更記錄
- **API 文件**：完整 API 端點說明（使用 Swagger 或 Postman Collection）
- **資料庫設計文件**：ER Diagram、資料表說明
- **系統架構文件**：架構圖、模組說明、技術堆疊
- **部署文件**：環境需求、部署步驟、設定說明

8.2.2 使用者文件

- **使用者手冊**：操作流程、功能說明、常見問題
- **管理員手冊**：系統設定、使用者管理、資料維護
- **教學影片（選配）**：主要功能操作示範

8.3 驗收程序

階段 1：功能驗收（UAT）

- 1. 廠商提交功能完成通知
- 2. 依照驗收測試計畫進行測試
- 3. 記錄缺陷並要求修正
- 4. 重新測試直到通過

階段 2：效能驗收

- 1. 執行效能測試腳本
- 2. 驗證效能指標符合要求
- 3. 記錄效能瓶頸並優化

階段 3：安全驗收

- 1. 執行安全掃描工具
- 2. 檢查安全機制實作
- 3. 驗證權限控制正確性

階段 4：文件驗收

- 1. 檢查文件完整性
- 2. 驗證文件與實作一致性
- 3. 確認文件可讀性

階段 5：最終驗收

- 1. 所有測試通過
- 2. 文件齊全
- 3. 簽署驗收報告

9. 測試要求

9.1 測試範圍

測試類型	測試範圍	覆蓋率要求
單元測試	所有工具函數、資料模型	> 80%
整合測試	模組間整合、API 整合	> 70%
功能測試	所有使用者功能	100%
流程測試	簽核流程各種情境	100%

效能測試	關鍵功能效能指標	100%
安全測試	認證、權限、注入攻擊	100%
相容性測試	Chrome, Firefox, Safari, Edge	100%

9.2 測試案例範例

9.2.1 配方建立測試

測試案例 ID: TC-FORM-001

測試名稱: 建立新配方 (正常流程)

前置條件: 使用者已登入

測試步驟:

1. 進入「配方建置器」頁面
2. 選擇範本「標準 AMC 濾網配方」
3. 填寫所有必填欄位
4. 點選「儲存」按鈕

預期結果:

- 配方儲存成功
- 自動生成配方編號 (格式: AMC-2025-XXX)
- 版本號為 1.0
- 顯示成功訊息
- 建立版本 1 記錄

9.2.2 簽核流程測試

測試案例 ID: TC-WORKFLOW-001

測試名稱: 單簽流程 (核准)

前置條件:

- 已建立配方
- 已設定單簽流程 (部門主管審核)

測試步驟:

1. 使用者 A 提交申請
2. 部門主管 (使用者 B) 登入
3. 進入「待我簽核」
4. 查看申請詳情
5. 點選「核准」並填寫意見
6. 確認核准

預期結果:

- 申請狀態變更為「已核准」
- 簽核歷史記錄正確
- 申請人收到通知

- 操作日誌記錄正確

9.2.3 版本控制測試

測試案例 ID: TC-VERSION-001

測試名稱: 版本回滾功能

前置條件:

- 配方已有 3 個版本

測試步驟:

1. 進入配方詳情頁面
2. 點選「版本歷史」
3. 選擇版本 2
4. 點選「回滾到此版本」
5. 確認回滾操作

預期結果:

- 配方資料還原為版本 2 的內容
- 建立新版本 4 (內容與版本 2 相同)
- 版本註解為「回滾到版本 2」
- 版本 3 保留不刪除

9.2.4 Golden Recipe 評分測試

測試案例 ID: TC-GOLDEN-001

測試名稱: 自動評分與認證

前置條件:

- 配方已建立
- 已有 25 批次品質數據

測試步驟:

1. 使用資料模擬器生成超高品質數據 (良率 98.5%、CPK 1.67)
2. 系統自動計算 Golden Score
3. 檢查是否符合自動認證條件

預期結果:

- Golden Score = 95.3 (誤差 < 0.1)
- 自動認證為 Golden Recipe (isGolden = true)
- goldenCertifiedBy = "SYSTEM-AUTO"
- 配方顯示 🏆 Golden Recipe 標記

9.3 測試環境

環境類型	說明	用途
------	----	----

開發環境	本機開發環境	開發與單元測試
測試環境	獨立測試伺服器	整合測試、功能測試
預生產環境	與生產環境相同配置	UAT、效能測試
生產環境	正式上線環境	正式服務

9.4 測試工具

工具類型	工具名稱	用途
單元測試	Jest / Vitest	JavaScript 單元測試
整合測試	Cypress / Playwright	E2E 測試
API 測試	Postman / Newman	API 自動化測試
效能測試	Apache JMeter / k6	壓力測試
安全掃描	OWASP ZAP / Burp Suite	安全漏洞掃描
程式碼品質	ESLint / SonarQube	程式碼靜態分析

10. 部署要求

10.1 系統需求

10.1.1 伺服器需求（生產環境）

項目	最低需求	建議配置
CPU	2 Core	4 Core
記憶體	4 GB	8 GB
硬碟	50 GB SSD	100 GB SSD
網路	100 Mbps	1 Gbps
作業系統	Ubuntu 20.04+ / CentOS 8+	Ubuntu 22.04 LTS

10.1.2 軟體需求

軟體	版本要求	說明
Node.js	18.x LTS	執行環境
PostgreSQL	14+	資料庫
Nginx	1.20+	反向代理

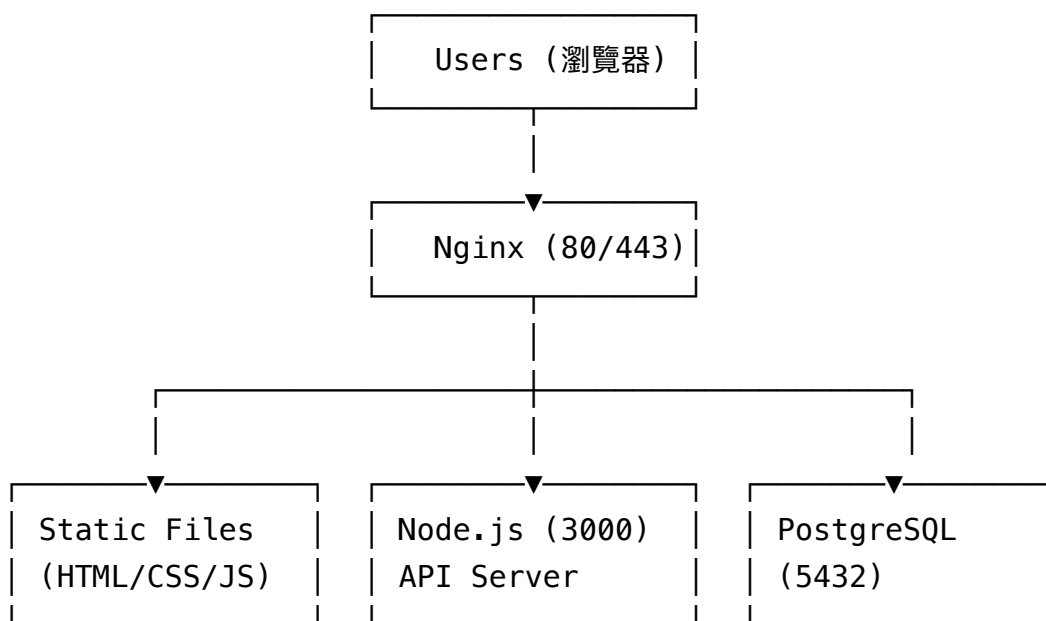
Redis	6.x	Session 儲存（選配）
Docker	20.x	容器化部署（選配）

10.1.3 瀏覽器支援

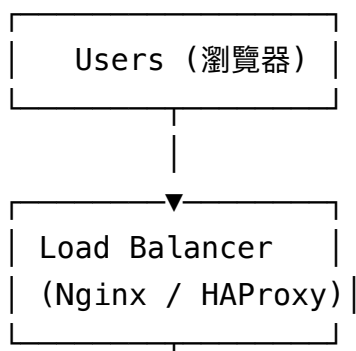
瀏覽器	最低版本
Chrome	100+
Firefox	100+
Safari	15+
Edge	100+

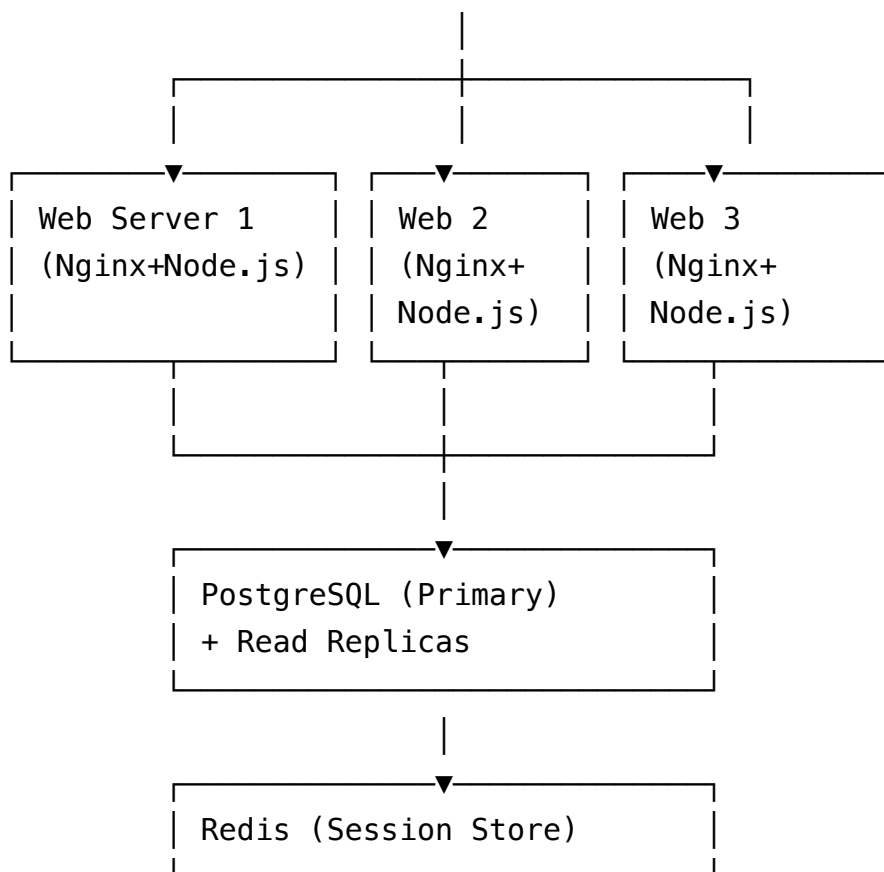
10.2 部署架構

10.2.1 單機部署（小型環境）



10.2.2 負載平衡部署（大型環境）





10.3 部署步驟

10.3.1 準備階段

1. 環境檢查

```
# 檢查 Node.js 版本
node -v # 應為 v18.x

# 檢查 PostgreSQL 版本
psql --version # 應為 14+

# 檢查 Nginx 版本
nginx -v # 應為 1.20+
```

2. 建立資料庫

```
# 連線到 PostgreSQL
psql -U postgres

# 建立資料庫與使用者
CREATE DATABASE rms_db;
```



```
CREATE USER rms_user WITH PASSWORD 'secure_password';  
GRANT ALL PRIVILEGES ON DATABASE rms_db TO rms_user;
```

3. 執行資料庫遷移

執行建立資料表腳本

```
psql -U rms_user -d rms_db -f database/schema.sql
```

執行初始資料腳本

```
psql -U rms_user -d rms_db -f database/seed.sql
```

10.3.2 應用部署

1. 下載原始碼

```
git clone https://github.com/your-org/RMS.git  
cd RMS
```

2. 安裝依賴

```
npm install --production
```

3. 環境變數設定

建立 .env 檔案

```
cp .env.example .env
```

編輯 .env

```
nano .env
```

.env 內容範例：

```
NODE_ENV=production  
PORT=3000
```

資料庫設定

```
DB_HOST=localhost  
DB_PORT=5432  
DB_NAME=rms_db  
DB_USER=rms_user
```

```
DB_PASSWORD=secure_password
```

```
# Session 設定
```

```
SESSION_SECRET=your_random_secret_key
```

```
# API 設定
```

```
API_BASE_URL=https://api.yourdomain.com
```

```
# EAP 系統整合
```

```
EAP_WEBHOOK_URL=http://eap-system/api/recipes/import
```

```
# Logging
```

```
LOG_LEVEL=info
```

4. 建置前端

```
npm run build
```

5. 啟動應用

```
# 使用 PM2 管理 Node.js 程序
```

```
npm install -g pm2
```

```
pm2 start npm --name "rms-api" -- start
```

```
pm2 save
```

```
pm2 startup
```

10.3.3 Nginx 設定

1. 建立 Nginx 設定檔

```
sudo nano /etc/nginx/sites-available/rms
```

2. 設定檔內容

```
server {  
    listen 80;  
    server_name yourdomain.com;  
  
    # 重導向到 HTTPS  
    return 301 https://$server_name$request_uri;  
}
```

```

server {
    listen 443 ssl http2;
    server_name yourdomain.com;

    # SSL 憑證
    ssl_certificate /etc/ssl/certs/yourdomain.com.crt;
    ssl_certificate_key /etc/ssl/private/yourdomain.com.key;

    # 靜態檔案
    location / {
        root /var/www/rms/dist;
        try_files $uri $uri/ /index.html;
    }

    # API 代理
    location /api/ {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    # 快取靜態資源
    location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg|woff|woff2)$ {
        expires 1y;
        add_header Cache-Control "public, immutable";
    }
}

```

3. 啟用設定並重啟 Nginx

```

sudo ln -s /etc/nginx/sites-available/rms /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx

```

10.3.4 Docker 部署 (選配)

1. Dockerfile

```
FROM node:18-alpine
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install --production
```

```
COPY . .
```

```
RUN npm run build
```

```
EXPOSE 3000
```

```
CMD ["npm", "start"]
```

2. docker-compose.yml

```
version: '3.8'
```

```
services:
```

```
  db:
```

```
    image: postgres:14-alpine
```

```
    environment:
```

```
      POSTGRES_DB: rms_db
```

```
      POSTGRES_USER: rms_user
```

```
      POSTGRES_PASSWORD: secure_password
```

```
    volumes:
```

```
      - postgres_data:/var/lib/postgresql/data
```

```
    ports:
```

```
      - "5432:5432"
```

```
  api:
```

```
    build: .
```

```
    ports:
```

```
      - "3000:3000"
```

```
    environment:
```

```
      NODE_ENV: production
```

```
      DB_HOST: db
```

```
      DB_PORT: 5432
```

```
      DB_NAME: rms_db
```

```
      DB_USER: rms_user
```

```
      DB_PASSWORD: secure_password
```

```
    depends_on:
```

- db

nginx:

image: nginx:alpine

ports:

- "80:80"

- "443:443"

volumes:

- ./nginx.conf:/etc/nginx/nginx.conf

- ./dist:/usr/share/nginx/html

depends_on:

- api

volumes:

postgres_data:

3. 啟動容器

```
docker-compose up -d
```

10.4 監控與日誌

10.4.1 應用監控

使用 PM2 監控

```
pm2 monit
```

查看日誌

```
pm2 logs rms-api
```

重啟應用

```
pm2 restart rms-api
```

10.4.2 日誌管理

應用日誌位置

```
/var/log/rms/application.log
```

```
/var/log/rms/error.log
```

Nginx 日誌位置

```
/var/log/nginx/access.log
```

/var/log/nginx/error.log

10.4.3 資料庫備份

```
# 每日自動備份腳本
#!/bin/bash
BACKUP_DIR="/backup/rms"
DATE=$(date +%Y%m%d_%H%M%S)

pg_dump -U rms_user rms_db > $BACKUP_DIR/rms_db_$DATE.sql

# 保留最近 30 天的備份
find $BACKUP_DIR -name "*.sql" -mtime +30 -delete
```

11. 維護與支援

11.1 維護範圍

維護類型	說明	回應時間
錯誤修正	修正系統錯誤與 Bug	關鍵錯誤 4 小時 一般錯誤 24 小時
功能調整	小幅度功能調整	依排程協調
效能優化	系統效能調校	依排程協調
安全更新	安全性修補	24 小時內
版本升級	系統版本升級	依排程協調

11.2 支援服務

11.2.1 技術支援

- 支援時間：週一至週五 09:00-18:00
- 支援管道：Email、電話、線上客服
- 支援內容：
 - 使用問題諮詢
 - 錯誤排除協助
 - 功能操作指導
 - 系統設定協助

11.2.2 教育訓練

- 系統管理員訓練（4 小時）
 - 系統架構說明
 - 使用者管理
 - 系統設定
 - 資料備份與還原
 - 常見問題排除
- 一般使用者訓練（2 小時）
 - 配方建立與管理
 - 簽核流程操作
 - 品質數據查詢
 - 報表產生

11.3 版本更新計畫

更新類型	頻率	內容
補丁更新	每月	Bug 修正、安全更新
小版本更新	每季	功能調整、效能優化
大版本更新	每年	重大功能新增、架構升級

12. 專案時程

12.1 里程碑規劃

階段	工作項目	預計工期	交付成果
需求確認	需求訪談、規格確認	2 週	需求規格書（簽核）
系統設計	架構設計、UI 設計、資料庫設計	3 週	設計文件、原型圖
開發階段 1	配方管理、簽核流程	6 週	Demo 1（核心功能）
開發階段 2	Golden Recipe、SPC	4 週	Demo 2（進階功能）
開發階段 3	測試系統、API 整合	3 週	Demo 3（完整功能）
測試階段	單元測試、整合測試、UAT	4 週	測試報告
部署階段	環境建置、系統部署	2 週	上線系統
驗收階段	功能驗收、文件驗收	2 週	驗收報告
教育訓練	使用者訓練、管理員訓練	1 週	訓練記錄
保固維護	系統維護、問題修正	12 個月	維護報告

總工期：約 6 個月（不含保固期）

12.2 人力配置

角色	人數	主要職責
專案經理	1	專案管理、進度控制、客戶溝通
系統分析師	1	需求分析、系統設計、文件撰寫
前端工程師	2	UI 開發、前端功能實作
後端工程師	2	API 開發、資料庫設計
測試工程師	1	測試計畫、測試執行、缺陷管理
UI/UX 設計師	1	介面設計、原型設計（部分時間）

13. 驗收標準

13.1 功能驗收標準

所有功能項目需符合以下標準：

驗收項目	驗收標準	驗收方法
配方管理	能建立、編輯、刪除、查詢配方 配方編號自動生成 版本自動管理	功能測試、實際操作
簽核流程	支援單簽、並簽、串簽、條件分支 流程正確推進 簽核歷史完整記錄	流程測試、各種情境
版本控制	自動版本追蹤 版本歷史查看 版本比對 版本回滾	版本測試、回滾測試
Golden Recipe	自動評分計算正確 自動認證邏輯正確 候選推薦正確	評分測試、認證測試
SPC 模組	管制圖正確顯示 Cp/Cpk 計算正確 異常偵測正確	數據測試、計算驗證
測試系統	所有測試劇本通過 測試報告正確顯示	執行測試、檢查報告

13.2 效能驗收標準

效能指標	驗收標準	驗收方法
頁面載入	首次載入 < 2 秒	Chrome DevTools
API 回應	一般查詢 < 500ms	Postman / JMeter
資料處理	支援 1000+ 配方	壓力測試
並發使用者	支援 100 人同時線上	JMeter 並發測試

13.3 安全驗收標準

安全項目	驗收標準	驗收方法
認證機制	強制登入、Session 管理正常	安全測試
權限控制	角色權限正確分離	權限測試
資料加密	密碼已加密儲存	資料庫檢查
注入防護	無 SQL 注入、XSS 漏洞	OWASP ZAP 掃描

13.4 文件驗收標準

文件項目	驗收標準
技術文件	README、API 文件、架構文件齊全
使用者手冊	操作步驟清晰、截圖完整
測試報告	測試結果記錄完整、覆蓋率達標
原始碼註解	關鍵函數有 JSDoc 註解

13.5 驗收簽核

驗收通過後，雙方簽署「系統驗收報告」，確認以下事項：

-  所有功能符合規格要求
-  效能、安全指標達標
-  文件齊全且正確
-  教育訓練完成
-  原始碼與建置檔案交付
-  進入保固維護期

附錄

附錄 A：術語表

術語	英文	說明
配方	Recipe	產品製程的參數與步驟定義
簽核流程	Approval Workflow	多層級審核流程
Golden Recipe	Golden Recipe	經過驗證的最佳化配方
SPC	Statistical Process Control	統計製程管制
CPK	Process Capability Index	製程能力指標
EAP	Engineering Automation Platform	工程自動化平台
ECU	Equipment Control Unit	設備控制單元

附錄 B：參考文件

- README.md - 系統說明文件
- CHANGELOG.md - 版本變更記錄
- docs/GOLDEN_RECIPE_DESIGN.md - Golden Recipe 設計文件
- docs/TEMPLATE_MANAGEMENT_PLAN.md - 範本管理計畫

附錄 C：聯絡資訊

專案團隊：

- 專案經理：（待指定）
- 技術負責人：（待指定）
- Email：（待提供）
- 專案網站：<https://github.com/seikaikyo/Dash-RMS>

文件結束

版本歷史：

- v1.0 (2025-10-11) - 初版發布