# **功能設計文件 (Functional Design Document)**

**專案名稱**：Bombus 企業管理系統

**版本**：V6.0 (PRD Strict Alignment Ver.)

**對應文件**：Bombus V6.0 PRD Section 3

**狀態**：完整版

**撰寫人**：Gemini (資深系統設計規劃師)

## 1. 引言

本文件依據 PRD 的 L0-L6 結構，詳細定義每個子模組的資料流 (Data Flow)、資料庫關聯 (Schema) 與核心演算法 (Algorithm)。

## 2. 模組功能詳細設計 (Module Detailed Design)

### L0 系統核心與儀表層 (System Core & Dashboard)

#### L0.0 企業管理儀表板

* **資料聚合邏輯**：
  + **KR 摘要**: GET /api/v6/okr/summary -> 呼叫 LLM API，傳入當前季度 OKR 狀態文字，回傳一句話摘要。
  + **Working Flow**: GET /api/v6/org/flow -> 查詢 DepartmentRelations 表，遞迴構建節點與邊緣 (Nodes & Edges) JSON。
* **資料庫實體**: DashboardWidgets (user\_id, widget\_type, config\_json).

#### L0.1 系統管理與權限控制

* **權限邏輯 (RBAC)**:
  + Middleware 攔截請求 -> 解析 JWT Token -> 讀取 UserRoles -> 比對 Permissions 表 -> Allow/Deny。
  + **列級權限**: 在 SQL 查詢時自動注入 WHERE department\_id = :user\_dept\_id (若權限 Scope 為 Department)。
* **自動備份**: 每日 03:00 Cron Job -> pg\_dump -> 加密 -> 上傳至 NAS 指定路徑 /backup/YYYYMMDD/。

### L1 員工管理模組 (Employee Management)

#### L1.1 招募與候選人管理 (AI Enhanced)

* **AI 評分演算法**:
  + Input: 面試逐字稿 (Transcript), 關鍵字庫 (Keywords), JD 權重 (Weights).
  + Process:
    1. **關鍵字匹配**: Score\_KW = Σ(Keyword\_Found \* Weight)
    2. **語意分析**: LLM 分析 "抗壓性", "邏輯" -> Score\_Semantic (0-100)
    3. **適配度**: Vector Similarity (Candidate\_Vector, JD\_Vector) -> Score\_Match
  + Output: Final\_Score = (Score\_KW \* 0.4) + (Score\_Semantic \* 0.3) + (Score\_Match \* 0.3)
* **資料庫實體**: Candidates, InterviewRecords, RiteTestResults.

#### L1.2 員工檔案與歷程管理

* **ROI 計算邏輯**:
  + Cost = 薪資 + 福利 + 培訓費用 (L3) + 設備折舊.
  + Value = 專案貢獻金額 (L4) + 績效係數折算值 (L5).
  + ROI = (Value - Cost) / Cost \* 100%.
* **資料庫實體**: Employees, JobHistory, RoiMetrics.

#### L1.3 人才庫與再接觸管理

* **再行銷邏輯**:
  + 每日掃描 JobOpenings (新職缺) 與 TalentPool (舊候選人)。
  + 若 Similarity(Candidate\_Skills, Job\_Reqs) > 80% -> 寫入 Notifications 表 (通知 HR)。

#### L1.4 職涯晉升與接班規劃 (含週休三日)

* **週休三日監控邏輯 (Scheduled Job)**:
  + 每月 1 號執行：
  + Task\_Rate = 完成任務數 / 總指派任務數 (From L4).
  + Collab\_Score = 來自 L5.3 的 360 同儕評分.
  + **判定**: 若 Task\_Rate < 90% OR Collab\_Score < 3.5 -> 寫入 TrialWarnings -> 發送 Email。
  + **退場**: 查詢 TrialWarnings，若 count(last\_3\_months) >= 3 -> 更新 EmployeeStatus 為 'Standard\_Week'。

#### L1.5 會議管理

* **同步邏輯**: 使用 Google Calendar API Webhook，雙向同步 MeetingRecords 與 GCal Events。
* **資料庫實體**: MeetingRecords, ActionItems (linked to L4 Tasks).

### L2 職能管理模組 (Competency Management)

#### L2.1 職等職級管理

* **管理辦法生成**:
  + Trigger: 組織架構變更.
  + Process: 讀取 OrgStructure, JobLevels -> 填入 "Management\_Policy\_Template.docx" -> 轉 PDF -> 存入 L0 文件庫。
* **資料庫實體**: JobLevels, CareerPaths (定義垂直/橫向/跨部門規則).

#### L2.2 職務說明書管理 (JD)

* **AI 撰寫**: 調用 LLM，Prompt: "Generate JD for [Job\_Title] with requirements [Keywords] based on ISO standards."
* **版本控制**: JobDescriptions 表包含 version, created\_at, parent\_id 欄位。

#### L2.3 職能框架開發

* **KSA 結構化**: CompetencyModel (id, type='K/S/A', definition).
* **連動邏輯 (L2->L3)**:
  + Event: CompetencyEvaluation\_Completed.
  + Action: 若 Score < Required\_Level -> 查詢 Competency\_Course\_Mapping -> 寫入 RecommendedTraining (L3)。

#### L2.4 職能評估系統

* **評估流程**:
  + 支援 SelfAssessment (自評) 與 ManagerAssessment (主管評)。
  + 儲存 EvaluationRecords，包含 eval\_date, rater\_id, score\_details.

#### L2.5 職能落差分析

* **落差計算演算法**:
  + Gap\_Score = Required\_Score (from L2.2 JD) - Actual\_Score (from L2.4 Evaluation).
  + 若 Gap\_Score > 0 標記為紅字，並計算 Severity\_Index (嚴重度指數) = Gap\_Score \* Weight.

#### L2.6 AI 職能生成引擎

* **生成邏輯**:
  + 輸入: 非結構化 JD 文本。
  + 處理: NLP 實體識別 (NER) 提取技能名詞。
  + 輸出: 自動建立或更新 CompetencyModel 條目。

### L3 教育訓練管理模組 (Training & Development)

#### L3.1 培訓計畫管理

* **計畫階層**: 支援 TrainingPlans (L1-L4 階層) 的 CRUD 與 parent\_id 關聯。
* **預算控管**: Budgets 表追蹤 allocated\_amount vs used\_amount。

#### L3.2 課程與報名管理 (含人才地圖)

* **人才地圖演算法**:
  + **熱力圖**: SQL Aggregation AVG(score) GROUP BY department, competency.
  + **九宮格**:
    - X = Performance\_Score (L5).
    - Y = Potential\_Score (L2 Growth Rate + L3.4 Conversion Rate).
    - Mapping: (High, High) -> 'Star', (Low, Low) -> 'Risk'.

#### L3.3 線上測驗系統

* **防作弊**: 紀錄 blur 事件 (視窗切換) 次數 -> 存入 ExamSessions。
* **自動評分**: 選擇題直接比對 AnswerKey，簡答題可選用 LLM 關鍵字比對。

#### L3.4 培訓成效追蹤與回饋

* **三個月反饋邏輯**:
  + 排程: Training\_End\_Date + 90 days 觸發通知。
  + **轉化率計算**: Conversion\_Rate = (問卷中 "已應用" 的技能數 / 課程教導總技能數).
  + 邏輯判斷：若 AVG(Conversion\_Rate) < 50% -> 更新 CourseStatus 為 'Under\_Review'。

### L4 專案管理模組 (Project Management)

#### L4.1 專案與任務管理

* **隱形成本計算**: Hidden\_Cost = SUM(Task\_Hours \* Employee\_Hourly\_Rate).
* **WBS 結構**: Tasks 表支援 parent\_task\_id 實現無限層級分解。

#### L4.2 專案協作與進度追蹤

* **資料庫實體**: Tasks, Comments, TaskDependencies.
* **進度計算**: 依據子任務完成權重計算父任務 % Complete。

#### L4.3 專案績效與毛利

* **未來損益預測 (AI Future P&L)**:
  + Base\_EAC = Budget / CPI (Cost Performance Index).
  + Risk\_Adjustment: 分析專案留言 Sentiment (負面情緒佔比) -> 係數 k (e.g., 1.05).
  + AI\_Predicted\_Cost = Base\_EAC \* k.
  + Predicted\_Margin = Revenue - AI\_Predicted\_Cost.

#### L4.4 專案報表與分析

* **報表生成**: 聚合 ProjectFinance, Tasks, WorkLogs 數據，生成 JSON 供前端繪製圖表或匯出 PDF。

#### L4.5 專案自動化引擎

* **自動化規則**:
  + Trigger: Task Overdue, Budget Exceeded.
  + Action: Send Email, Create Alert Task.
  + 實作: Node.js Schedule Service 每日掃描規則庫。

### L5 績效管理模組 (Performance Management)

#### L5.1 過程管理系統與毛利計算

* **毛利計算引擎 (Core Logic)**:
  + 1. 取得 Revenue (財務 API).
    2. Direct\_Cost = 專案外包費 + SUM(Project\_Hours \* Rate) (From L4).
    3. Indirect\_Cost = (總租金+水電) \* Dept\_Allocation\_Ratio (From L5 Params).
    4. Gross\_Margin = Revenue - Direct\_Cost - Indirect\_Cost.
    5. Bonus\_Pool = Gross\_Margin \* Lookup(Bonus\_Rate\_Table, Margin\_Rate).

#### L5.2 考核週期管理

* **流程引擎**: ReviewCycles 定義 start\_date, end\_date, stages (Goal Setting, Mid-Review, Final Review).
* **自動通知**: 狀態變更時觸發通知服務。

#### L5.3 360 度回饋系統

* **匿名處理**:
  + 寫入 Feedback360 時，若設定為匿名，則在 View 層級隱藏 reviewer\_id，或在 DB 存入 Hash 值。
  + 整合邏輯: 回饋分數匯入 L5.5 績效分析。

#### L5.4 績效紀錄與日誌系統

* **CITA 整合**: PerformanceLogs (Critical Incidents) 包含 type (Positive/Negative), date, description.
* **關聯**: 連結至 L4 Project\_ID 或 L3 Training\_ID.

#### L5.5 績效分析與改善計畫

* **PIP 邏輯**: 若 Performance\_Score 連續 2 期 < C -> 自動建立 PipRecords 並通知主管。
* **分析**: 整合 L1.2 ROI 與 L3 ROI 數據，產出績效趨勢報告。

### L6 文化管理模組 (Culture Management)

#### L6.1 企業文化手冊管理 (含 EAP & 員工旅遊)

* **EAP 匿名機制**:
  + 寫入 EapAppointments 時，User ID 欄位存入 Hash(User\_ID + Salt)，確保無法反查但可統計次數。
* **員工旅遊**: TravelRecords 記錄參與度與滿意度，連結至 L6.6。

#### L6.2 獎項資料庫管理

* **資料庫實體**: Awards, AwardDeadlines。
* **推薦邏輯**: 根據公司屬性與專案成果匹配合適獎項。

#### L6.3 文件儲存庫

* **歸檔邏輯**: Documents 表包含 category, tags, nas\_path.
* **NAS 整合**: 負責大檔案的 I/O 操作。

#### L6.4 AI 申請助理

* **生成流程**:
  + 聚合 L4 專案成果 + L5 績效數據 -> 組合成 Prompt -> 呼叫 LLM 生成申請書草稿。

#### L6.5 智慧文件分析

* **檢核邏輯**: OCR 解析上傳文件 -> 比對 AwardRequirements -> 輸出缺漏清單。

#### L6.6 影響力評估引擎

* **相關性分析**:
  + 執行 Python SciPy: Correlation(EAP\_Usage\_Count, Retention\_Rate).
  + 輸出 Impact\_Factor 存入 CultureImpactMetrics.