**TÀI LIỆU PRD + SRS  
HỆ THỐNG QUẢN TRỊ NHÂN SỰ (HRM) NỘI BỘ**

Phiên bản: 1.0  
Ngày: 17/09/2025  
Đơn vị: FOTU / Healing Center (nếu áp dụng)

# BẢNG LỊCH SỬ PHIÊN BẢN

|  |  |  |  |
| --- | --- | --- | --- |
| Phiên bản | Ngày | Người soạn | Nội dung thay đổi |
| 1.0 | 17/09/2025 | ChatGPT | Khởi tạo tài liệu PRD + SRS tiêu chuẩn HRM |

Ghi chú: Tài liệu gồm các phần PRD (yêu cầu sản phẩm) và SRS (đặc tả yêu cầu phần mềm), chi tiết đủ để đội BA/DEV/TEST triển khai; kèm ‘Copilot Prompt Pack’ ở cuối để dùng trực tiếp với GitHub Copilot.

# 1. PRD – Product Requirements Document

## 1.1 Mục tiêu & KPI

- Xây dựng hệ thống HRM nội bộ quản lý người dùng, phòng ban, sơ đồ tổ chức, và hồ sơ nhân sự.  
- Tăng hiệu quả vận hành: giảm ≥50% thời gian tổng hợp thông tin nhân sự.  
- Bảo mật và phân quyền rõ ràng: không rò rỉ dữ liệu cá nhân; 100% truy cập phải qua xác thực.  
- Tính sẵn sàng ≥ 99.5% cho môi trường production.

## 1.2 Phạm vi (In-Scope)

• Web app nội bộ (desktop-first) cho các vai trò: ADMIN, HR, MANAGER, EMPLOYEE.  
• Chức năng: Đăng nhập & RBAC; Quản lý phòng ban; Quản lý nhân sự; Sơ đồ tổ chức; Tìm kiếm/lọc; Audit log; Nhập/Xuất dữ liệu.  
• API REST tiêu chuẩn; CSDL quan hệ (PostgreSQL).

## 1.3 Out-of-Scope (v1)

• Tính lương/payroll; Quản lý chấm công; Tuyển dụng; Đánh giá OKR/360; Cổng tự phục vụ phức tạp.  
• Tích hợp bên thứ ba (AD/SSO, CRM, ERP) – có thể cân nhắc v2.

## 1.4 Personas & Vai trò

- ADMIN: Toàn quyền cấu hình hệ thống, quản lý người dùng & phân quyền.  
- HR: Xem & chỉnh sửa toàn bộ dữ liệu nhân sự; import/export; báo cáo.  
- MANAGER: Xem dữ liệu phòng ban mình quản lý và các cấp dưới trong org tree; xác nhận/mô tả công việc.  
- EMPLOYEE: Xem hồ sơ cá nhân; cập nhật các trường được phép (ví dụ: liên hệ, ảnh).

## 1.5 Giả định & Ràng buộc chung

- Mỗi nhân sự thuộc tối đa 1 phòng ban chính (có thể mở rộng đa phòng ban ở v2).  
- Quan hệ quản lý theo cây (manager\_id) và/hoặc theo head phòng ban.  
- Tất cả yêu cầu đọc/ghi dữ liệu phải qua API có kiểm soát quyền.  
- HTTPS bắt buộc; nhật ký (audit log) cho thao tác CRUD.

# 2. SRS – Software Requirements Specification

Phần này đặc tả chi tiết từng tính năng, điều kiện, luồng, ràng buộc, dữ liệu, API và tiêu chí chấp nhận.

## 2.1 Ma trận phân quyền (RBAC)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tài nguyên | Hành động | ADMIN | HR | MANAGER | EMPLOYEE |
| Auth | Login/Logout/Refresh | ✔ | ✔ | ✔ | ✔ |
| User | Create | ✔ | ✔ | ✖ | ✖ |
| User | Read (self) | ✔ | ✔ | ✔ | ✔ |
| User | Read (all) | ✔ | ✔ | Giới hạn theo cấp | ✖ |
| User | Update (self profile fields) | ✔ | ✔ | Giới hạn self | ✔ (trường được phép) |
| User | Update (any) | ✔ | ✔ | Giới hạn cấp dưới | ✖ |
| User | Delete | ✔ | ✖ | ✖ | ✖ |
| Department | CRUD | ✔ | ✔ | Read | Read tối thiểu |
| OrgChart | View | ✔ | ✔ | ✔ (cấp dưới) | Self/Phòng ban (tối thiểu) |
| Audit Log | View | ✔ | ✔ | ✖ | ✖ |
| Config | Update | ✔ | ✖ | ✖ | ✖ |

## 2.2 Yêu cầu chức năng chi tiết

### F1. Xác thực & Phân quyền

Mô tả: Người dùng đăng nhập bằng username/password; hệ thống cấp JWT; kiểm soát quyền theo vai trò và/hoặc theo quan hệ tổ chức.

Luồng chính:

1) Người dùng nhập username/password → POST /auth/login.

2) Nếu hợp lệ → trả về {access\_token, refresh\_token, user:{id, role, department\_id}}.

3) Frontend lưu token (HttpOnly cookie hoặc secure storage).

4) Với mỗi request API, gửi access token; nếu hết hạn dùng refresh token để lấy token mới.

Ràng buộc & Quy tắc:

- Mật khẩu hash bằng bcrypt; độ dài ≥ 8 ký tự; có chữ hoa, chữ thường, số, ký tự đặc biệt.

- Khoá tài khoản sau 5 lần đăng nhập sai trong 15 phút (unlock tự động sau 30 phút hoặc do ADMIN/HR).

- Token hết hạn: access 15–30 phút; refresh 7–30 ngày; có cơ chế revoke.

- Mọi endpoint phải kiểm tra quyền truy cập dựa trên role và cây tổ chức.

Tiêu chí chấp nhận (AC):

- Không người dùng nào truy cập được khi không có token hợp lệ.

- Audit ghi nhận sự kiện đăng nhập thành công/thất bại.

### F2. Quản lý Phòng ban

Mô tả: Tạo/Sửa/Xoá phòng ban; gán Trưởng phòng (head); (tuỳ chọn) hỗ trợ phòng ban cha-con để tạo cây tổ chức.

Luồng chính:

1) HR/ADMIN tạo phòng ban: tên duy nhất trong công ty, mô tả (tuỳ chọn).

2) Gán head là 1 user thuộc đúng phòng ban đó.

3) Cập nhật/thay đổi head; chuyển nhân sự sang phòng ban khác cần ghi log.

4) Xoá phòng ban chỉ khi không còn nhân sự; nếu còn → yêu cầu chuyển/cập nhật trước.

Ràng buộc & Quy tắc:

- Tên phòng ban unique; không rỗng; độ dài ≤ 100.

- head\_id phải trỏ tới user có department\_id khớp.

- Nếu có parent\_department: cấm vòng lặp.

AC:

- Chỉ ADMIN/HR có thể CRUD; MANAGER chỉ xem; EMPLOYEE xem tối thiểu.

### F3. Quản lý Nhân sự

Mô tả: CRUD hồ sơ nhân sự (thông tin cá nhân, liên hệ, hợp đồng, vị trí, cấp bậc, ngày vào làm, tình trạng).

Trường dữ liệu cốt lõi:

- Mã NV (employee\_code), họ tên, email công ty (unique), số điện thoại, ngày sinh (tuổi ≥ 18), giới tính (tuỳ chọn),  
- Phòng ban, chức danh, manager\_id, ngày vào làm, trạng thái (Active/Inactive/OnLeave),  
- Địa chỉ, người liên hệ khẩn cấp (tên+điện thoại), ảnh đại diện (URL), ghi chú.

Luồng chính:

1) HR tạo nhân sự mới → gửi email kích hoạt (tuỳ chọn).

2) HR/ADMIN cập nhật hồ sơ; MANAGER cập nhật trường liên quan đến công việc (tuỳ chính sách).

3) EMPLOYEE cập nhật một số trường cá nhân cho phép (địa chỉ, ảnh, liên hệ).

Ràng buộc & Quy tắc:

- email unique, định dạng chuẩn; phone theo chuẩn E.164 hoặc nội địa.

- department\_id bắt buộc; manager\_id không bắt buộc nhưng khuyến nghị.

- Không xoá cứng nhân sự; chuyển trạng thái Inactive (soft delete) để giữ lịch sử.

AC:

- HR nhìn thấy toàn bộ; MANAGER xem được tất cả nhân sự dưới cấp theo cây; EMPLOYEE chỉ xem self.

### F4. Sơ đồ tổ chức (Org Chart)

Mô tả: Hiển thị cấu trúc tổ chức dạng cây từ CEO → cấp quản lý → nhân viên; filter theo phòng ban; tìm kiếm theo tên.

Ràng buộc & Quy tắc:

- Nút (node) hiển thị: tên, chức danh, phòng ban, head/manager, số lượng cấp dưới.

- Quyền xem: HR xem toàn bộ; MANAGER xem subtree của mình; EMPLOYEE xem bản thân và tuyến trên.

AC:

- Tải < 1s với cây ≤ 1.000 nhân sự (cache phân trang/mức).

### F5. Audit Log & Báo cáo

Mô tả: Ghi nhận hành động quan trọng (login, create/update/delete user/department, thay đổi head, thay đổi manager).

Ràng buộc & Quy tắc:

- Lưu: actor\_id, hành động, tài nguyên, giá trị trước/sau (diff rút gọn), thời gian, IP.

- Lưu tối thiểu 365 ngày; hỗ trợ export CSV/JSON.

AC:

- ADMIN/HR có thể tra cứu theo bộ lọc thời gian/người dùng/tài nguyên.

## 2.3 Yêu cầu phi chức năng (NFR)

- Bảo mật: HTTPS, JWT, bcrypt(12+), CSRF cho form nhạy cảm, rate limit login 100 req/15m/IP.  
- Hiệu năng: P95 API < 300ms; trang bảng < 2s với 10k bản ghi (phân trang, index).  
- Khả dụng: 99.5%; backup hằng ngày; RPO ≤ 24h, RTO ≤ 4h.  
- Khả mở rộng: tách frontend/backend; DB có index; cache đọc org chart.  
- Khả quan sát: log tập trung (request/response code, latency), cảnh báo 5xx, tỷ lệ lỗi login.

# 3. Mô hình dữ liệu & CSDL

3.1 Lược đồ bảng (DDL rút gọn – PostgreSQL):

CREATE TABLE departments (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(100) UNIQUE NOT NULL,  
 description TEXT,  
 parent\_id INT NULL,  
 head\_id INT NULL,  
 created\_at TIMESTAMP DEFAULT NOW(),  
 updated\_at TIMESTAMP DEFAULT NOW()  
);  
  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 employee\_code VARCHAR(30) UNIQUE,  
 username VARCHAR(50) UNIQUE NOT NULL,  
 password\_hash VARCHAR(255) NOT NULL,  
 full\_name VARCHAR(120) NOT NULL,  
 email VARCHAR(120) UNIQUE NOT NULL,  
 phone VARCHAR(20),  
 gender VARCHAR(10),  
 dob DATE,  
 role VARCHAR(20) NOT NULL CHECK (role IN ('ADMIN','HR','MANAGER','EMPLOYEE')),  
 title VARCHAR(100),  
 status VARCHAR(20) NOT NULL DEFAULT 'ACTIVE' CHECK (status IN ('ACTIVE','INACTIVE','ONLEAVE')),  
 department\_id INT NOT NULL,  
 manager\_id INT NULL,  
 address TEXT,  
 emergency\_contact\_name VARCHAR(120),  
 emergency\_contact\_phone VARCHAR(20),  
 avatar\_url TEXT,  
 created\_at TIMESTAMP DEFAULT NOW(),  
 updated\_at TIMESTAMP DEFAULT NOW()  
);  
  
ALTER TABLE departments ADD CONSTRAINT fk\_dept\_parent FOREIGN KEY (parent\_id) REFERENCES departments(id);  
ALTER TABLE users ADD CONSTRAINT fk\_user\_dept FOREIGN KEY (department\_id) REFERENCES departments(id);  
ALTER TABLE users ADD CONSTRAINT fk\_user\_manager FOREIGN KEY (manager\_id) REFERENCES users(id);  
ALTER TABLE departments ADD CONSTRAINT fk\_dept\_head FOREIGN KEY (head\_id) REFERENCES users(id);  
  
-- Indexes  
CREATE INDEX idx\_users\_dept ON users(department\_id);  
CREATE INDEX idx\_users\_manager ON users(manager\_id);  
CREATE INDEX idx\_dept\_parent ON departments(parent\_id);

## 3.2 Từ điển dữ liệu (Data Dictionary)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng | Trường | Kiểu | Ràng buộc | Mô tả |
| users | email | VARCHAR(120) | UNIQUE, NOT NULL | Email công ty |
| users | role | VARCHAR(20) | CHECK IN (ADMIN,HR,MANAGER,EMPLOYEE) | Vai trò |
| users | department\_id | INT | FK → departments.id | Phòng ban |
| departments | name | VARCHAR(100) | UNIQUE, NOT NULL | Tên phòng ban |
| departments | head\_id | INT | FK → users.id | Trưởng phòng |

# 4. API Contract (REST)

Mọi endpoint yêu cầu JWT trừ /auth/login; áp dụng RBAC theo ma trận ở mục 2.1.

## 4.1 Auth

POST /api/auth/login  
Req: {"username":"string","password":"string"}  
Res: {"access\_token":"jwt","refresh\_token":"jwt","user":{"id":1,"role":"HR","department\_id":2}}  
  
POST /api/auth/refresh  
Req: {"refresh\_token":"jwt"}  
Res: {"access\_token":"jwt"}  
  
POST /api/auth/change-password  
Req: {"old":"string","new":"string"}  
Res: 204 No Content

## 4.2 Users

GET /api/users/me → 200 {id, full\_name, email, role, department\_id}  
GET /api/users/:id → quyền: HR/ADMIN; MANAGER nếu :id thuộc cấp dưới; EMPLOYEE chỉ nếu :id=self  
POST /api/users (ADMIN/HR)  
Req: {employee\_code?, username, password, full\_name, email, phone?, role, department\_id, manager\_id?, ...}  
Res: 201 {id}  
  
PUT /api/users/:id (ADMIN/HR; MANAGER nếu cấp dưới; EMPLOYEE nếu self và chỉ các trường cho phép)  
Req: {full\_name?, phone?, address?, avatar\_url?, status?, department\_id?, manager\_id?}  
Res: 200 {updated:true}  
  
DELETE /api/users/:id (ADMIN) → soft delete: status=INACTIVE

## 4.3 Departments

GET /api/departments?page=&page\_size=&q=  
Res: 200 [{id, name, head\_id, parent\_id, counts:{employees}}]  
  
POST /api/departments (ADMIN/HR)  
Req: {name, description?, parent\_id?, head\_id?}  
Res: 201 {id}  
  
PUT /api/departments/:id (ADMIN/HR)  
Req: {name?, description?, parent\_id?, head\_id?}  
Res: 200 {updated:true}  
  
DELETE /api/departments/:id → chỉ khi counts.employees=0; nếu khác 0 trả 409 Conflict

## 4.4 Org Chart

GET /api/orgchart?root\_id=:userId|dept\_id=:deptId  
Res: 200 {  
 "id": 1,  
 "name": "CEO",  
 "children": [{"id":2,"name":"Head of HR","children":[...]}]  
}

## 4.5 Quy tắc lỗi & mã trạng thái

- 400 Bad Request (validate sai); 401 Unauthorized; 403 Forbidden (không đủ quyền);  
- 404 Not Found; 409 Conflict (xung đột ràng buộc nghiệp vụ); 422 Unprocessable Entity; 500 Internal Server Error.  
- Mã lỗi dạng: AUTH\_INVALID\_CREDENTIALS, USER\_NOT\_FOUND, DEPT\_NOT\_EMPTY, VALIDATION\_FAILED.

# 5. Yêu cầu giao diện (Frontend)

Công nghệ: Laravel Blade (hoặc SPA tách biệt nếu cần) (App Router), TypeScript, Ant Design, Axios, Zustand/Redux (tuỳ chọn).

Trang & thành phần:

- /login: Form đăng nhập, nhớ tài khoản (tuỳ chọn), xử lý lỗi, lockout.

- /dashboard: Thống kê nhanh: tổng nhân sự, số phòng ban, biểu đồ phân bổ (tuỳ chọn).

- /employees: Bảng danh sách (phân trang, lọc theo phòng ban, trạng thái), nút tạo mới; form chi tiết.

- /departments: Danh sách cây; CRUD; gán trưởng phòng; chỉ số số lượng nhân sự.

- /orgchart: Tree-view; tìm kiếm; click node để xem chi tiết; điều hướng theo quyền.

- Header: avatar user, menu đổi mật khẩu, logout; Footer: version, bản quyền.

Field-level validation trên client đồng bộ với server; hiển thị thông báo lỗi thân thiện.

# 6. Kiến trúc & ADR

6.1 Lựa chọn công nghệ:

- Frontend: Laravel Blade (hoặc SPA tách biệt nếu cần) + Ant Design → năng suất thi công, SSR cho trang đăng nhập/SEO nội bộ.

- Backend: PHP (Laravel) (Laravel hoặc Express + Zod/Joi validate) → nhanh, nhiều thư viện.

- DB: PostgreSQL → quan hệ mạnh, hỗ trợ ràng buộc & index; Eloquent ORM/Eloquent ORM (tuỳ).

- Auth: JWT (access+refresh); HttpOnly cookie cho access (khuyến nghị); CORS cấu hình whitelist.

6.2 ADR chính:

ADR-001: Dùng RBAC thay vì ABAC ở v1 để đơn giản, đủ nhu cầu; mở rộng ABAC v2 nếu cần.

ADR-002: Soft delete nhân sự (status=INACTIVE) để giữ lịch sử; hard delete chỉ cho dữ liệu test.

ADR-003: Org chart tính theo manager\_id và head phòng ban; cache subtree theo root.

# 7. Kiểm thử & Tiêu chí chấp nhận

7.1 Test phân quyền:

- ADMIN có thể CRUD mọi tài nguyên.

- HR có thể CRUD users/departments (trừ delete user), xem tất cả, xem audit.

- MANAGER chỉ xem/đổi thông tin cấp dưới theo chính sách; không truy cập dữ liệu ngoài cấp.

- EMPLOYEE chỉ xem self; đổi các trường cho phép.

7.2 Test chức năng:

- Auth: đăng nhập đúng/sai; lockout sau 5 lần; refresh token; đổi mật khẩu; CSRF (nếu áp dụng).

- Departments: tạo tên trùng → 409; gán head không thuộc phòng → 422; xoá khi còn nhân sự → 409.

- Employees: tạo email trùng → 409; ngày sinh < 18 tuổi → 422; chuyển phòng cập nhật audit.

- Org chart: MANAGER thấy đúng subtree; HR thấy toàn bộ; hiệu năng với 1.000+ node.

- Audit: ghi nhận trước/sau; lọc theo thời gian/tài nguyên; export CSV.

# 8. Bảo mật, Vận hành & Tuân thủ

- Mã hoá: HTTPS, TLS 1.2+; password hash bcrypt(12+); token ký RS256.  
- Secret: lưu trong vault/.env (không commit); rotate khóa định kỳ.  
- Logging/Monitoring: tập trung; cảnh báo tỉ lệ 5xx, login fail spike, độ trễ API.  
- Backup: hằng ngày; kiểm thử khôi phục hàng quý.  
- Tuân thủ: bảo vệ dữ liệu cá nhân (ẩn/che bớt khi cần), phân quyền tối thiểu; nhật ký truy cập.

# 9. Triển khai & Môi trường

- Môi trường: dev, staging, prod; CI/CD (GitHub Actions) build+test+lint; Docker (tuỳ chọn).  
- Biến môi trường backend: DB\_URL, JWT\_PRIVATE\_KEY, JWT\_PUBLIC\_KEY, RATE\_LIMIT, PASSWORD\_POLICY,...  
- Biến môi trường frontend: API\_BASE\_URL, SSO/LOGIN\_PAGE (nếu có).

# 10. Giả định, Rủi ro & Giảm thiểu

Giả định: user dùng email công ty; mạng nội bộ/ VPN bảo mật; cấp dữ liệu đầu vào CSV hợp lệ.

Rủi ro: lạm quyền truy cập; dữ liệu cá nhân rò rỉ; hiệu năng kém với cây lớn.

Giảm thiểu: RBAC chặt; masking/limited fields; index + cache; rate limit + audit; review định kỳ.

# Phụ lục A – Copilot Prompt Pack (dùng trực tiếp)

## Backend (Laravel + PostgreSQL)  
Hãy tạo dự án backend Laravel với các module: auth, users, departments, orgchart, audit.  
- Áp dụng RBAC: ADMIN, HR, MANAGER, EMPLOYEE theo ma trận mục 2.1.  
- Triển khai API theo mục 4; validate bằng Form Request; lỗi theo mục 4.5.  
- Eloquent ORM schema dựa trên DDL mục 3; migration + seed 1-2 phòng ban và 5 user.  
- Bảo mật: bcrypt(12+), JWT access (15m) + refresh (7d), refresh rotation, revoke list.  
- Test Feature/Unit cho login, tạo user, gán head, orgchart subtree.

## Frontend (Laravel Blade (hoặc SPA tách biệt nếu cần) + Ant Design)  
Hãy tạo frontend Laravel Blade (hoặc SPA tách biệt nếu cần) (App Router) với các route: /login, /dashboard, /employees, /departments, /orgchart.  
- State auth + guard theo role; fetch API bằng Axios; hiển thị lỗi thân thiện.  
- Trang employees/departments có bảng, lọc, phân trang; form CRUD; xác thực client-side trùng schema server.  
- Org chart tree-view; chỉ hiển thị dữ liệu theo quyền.