**HƯỚNG DẪN CẤU TRÚC HỒ SƠ THIẾT KẾ SẢN PHẨM (PDD/SRS)**

# I. Cấu Trúc Cây Thư Mục Đề Xuất

Việc sử dụng đánh số thứ tự (1.0, 2.0, 3.0,...) không chỉ áp dụng cho tài liệu giới thiệu chung mà còn áp dụng cho các thư mục con, giúp việc tham chiếu chéo (cross-reference) trong quá trình phát triển trở nên dễ dàng và chính xác.

/HỒ SƠ THIẾT KẾ DỰ ÁN [TÊN DỰ ÁN]  
├── 1.0. TỔNG QUAN DỰ ÁN (Project Overview)  
│ ├── 1.1. Introduction\_&\_Scope.md (Tài liệu giới thiệu, mục đích, phạm vi tổng thể)  
│ └── 1.2. Definitions\_&\_Glossary.md (Danh sách các thuật ngữ chung của toàn bộ ứng dụng)  
│  
├── 2.0. YÊU CẦU CHỨC NĂNG (Functional Specifications)  
│ ├── 2.1. SPEC\_GLOBAL\_RULES.md (Các quy tắc nghiệp vụ chung áp dụng cho toàn app)  
│ ├── 2.2. SCR\_DASHBOARD.md (Spec màn hình 1)  
│ ├── 2.3. SCR\_EXPERIENCE\_REPORT.md (Spec màn hình 2 - Tài liệu bạn vừa tạo)  
│ └── ... (Tất cả các tài liệu spec màn hình còn lại)  
│  
├── 3.0. LUỒNG NGHIỆP VỤ & NGƯỜI DÙNG (Business & User Flow)  
│ ├── 3.1. USER\_PERSONAS.md  
│ ├── 3.2. USER\_STORIES\_LIST.md (Tổng hợp tất cả US của dự án)  
│ └── 3.3. APP\_FLOW\_DIAGRAMS/  
│ ├── Main\_Navigation\_Flow.png (Sơ đồ luồng điều hướng chính)  
│ └── Task\_Management\_Flow.png (Sơ đồ luồng quản lý công việc)  
│  
├── 4.0. THIẾT KẾ KỸ THUẬT & KIẾN TRÚC (Technical Design & Architecture)  
│ ├── 4.1. DB\_SCHEMA/  
│ │ ├── 4.1.1. ERD\_Diagram.png (Sơ đồ quan hệ thực thể)  
│ │ └── 4.1.2. Table\_Definitions.md (Đặc tả chi tiết các bảng/collection)  
│ ├── 4.2. API\_SPEC/  
│ │ ├── 4.2.1. API\_Contract\_Tasks.md (Đặc tả API liên quan đến công việc)  
│ │ └── 4.2.2. API\_Contract\_Users.md (Đặc tả API liên quan đến người dùng)  
│ └── 4.3. SYSTEM\_ARCHITECTURE.png (Sơ đồ kiến trúc tổng thể)  
│  
└── 5.0. TIÊU CHUẨN & CHẤT LƯỢNG (Standards & Quality)  
 ├── 5.1. PERMISSION\_MATRIX.md (Ma trận phân quyền)  
 ├── 5.2. NON\_FUNCTIONAL\_REQUIREMENTS.md (Tổng hợp các yêu cầu PNT như Performance, Security)  
 ├── 5.3. CODING\_STYLE\_GUIDE.md (Hướng dẫn code style)  
 └── 5.4. QA\_TEST\_PLAN.md (Kế hoạch kiểm thử)

# II. Yêu Cầu và Nội dung Chi tiết cho Từng Nhóm Tài Liệu

## 1. Yêu cầu Chức năng (Functional Specifications - Nhóm 2.0)

|  |  |  |
| --- | --- | --- |
| **Tài liệu** | **Nội dung chính** | **Yêu cầu quan trọng** |
| **SPEC MÀN HÌNH** (VD: SCR\_EXPERIENCE\_REPORT.md) | Chi tiết về Luồng hoạt động, Quy tắc nghiệp vụ, Yêu cầu hiển thị cho **một màn hình cụ thể**. | Phải có số thứ tự/ID (VD: SCR\_EXPERIENCE\_REPORT). Phải có thông tin **Nơi lưu trữ dữ liệu** (Database/API) để Dev biết lấy dữ liệu từ đâu. |
| **SPEC\_GLOBAL\_RULES.md** | Định nghĩa các quy tắc chung áp dụng toàn hệ thống (VD: Quy tắc tính Điểm Cơ Bản, Quy tắc xác thực phiên làm việc). | Đảm bảo tính nhất quán (Consistency). Mọi màn hình đều phải tuân thủ các quy tắc này. |

## 2. Luồng Nghiệp vụ & Người Dùng (Business & User Flow - Nhóm 3.0)

|  |  |  |
| --- | --- | --- |
| **Tài liệu** | **Nội dung chính** | **Yêu cầu quan trọng** |
| **USER\_PERSONAS.md** | Mô tả các chân dung người dùng chính của ứng dụng. | Giúp đội ngũ Dev hiểu được **bối cảnh (Context)** người dùng sử dụng ứng dụng. |
| **USER\_STORIES\_LIST.md** | Danh sách tổng hợp tất cả các User Story (US-1, US-2,...) đã được duyệt. | Mỗi US phải có **Tiêu chí Chấp nhận (Acceptance Criteria)** rõ ràng để QA dựa vào đó mà kiểm thử. |
| **APP\_FLOW\_DIAGRAMS** | **Sơ đồ Luồng (Flowchart)** kết nối các màn hình lại với nhau. | Sử dụng các ký hiệu chuẩn (hình chữ nhật cho màn hình, hình thoi cho quyết định) và có mũi tên chỉ dẫn rõ ràng. |

## 3. Thiết kế Kỹ thuật & Kiến trúc (Technical Design & Architecture - Nhóm 4.0)

|  |  |  |
| --- | --- | --- |
| **Tài liệu** | **Nội dung chính** | **Yêu cầu quan trọng** |
| **DB\_SCHEMA** | **ERD** và đặc tả chi tiết các **Bảng/Collection** (Tên, Kiểu dữ liệu, Mô tả). | Phải đảm bảo tính toàn vẹn (Integrity) và hiệu suất (Performance) khi truy vấn. Dev cần tài liệu này để bắt đầu xây dựng Backend. |
| **API\_SPEC** | Đặc tả chi tiết từng Endpoint (URL, Method, Request Body, Response Body). | Phải rõ ràng về **Status Code** (200, 400, 401, 500) và cấu trúc dữ liệu trả về cho cả trường hợp thành công và lỗi. |
| **SYSTEM\_ARCHITECTURE** | **Sơ đồ kiến trúc** (client, server, database, dịch vụ bên thứ ba). | Phải nêu rõ công nghệ sử dụng (VD: Frontend: React, Backend: NodeJS/Express, Database: PostgreSQL). |

## 4. Tiêu chuẩn & Chất lượng (Standards & Quality - Nhóm 5.0)

|  |  |  |
| --- | --- | --- |
| **Tài liệu** | **Nội dung chính** | **Yêu cầu quan trọng** |
| **PERMISSION\_MATRIX.md** | Bảng phân quyền chi tiết theo từng vai trò (Role). | Phải xác định rõ ràng quyền **Xem/Sửa/Thêm/Xóa** trên từng đối tượng dữ liệu. |
| **NON\_FUNCTIONAL\_REQUIREMENTS.md** | Tổng hợp các yêu cầu phi chức năng (Performance: Thời gian phản hồi < 2s; Security: Mọi API đều dùng HTTPS; Độ ổn định/Tính sẵn sàng). | Đặt các con số cụ thể có thể đo lường được để QA có thể kiểm thử. |
| **CODING\_STYLE\_GUIDE.md** | Quy ước chung về code style, đặt tên, và cách tổ chức thư mục code. | Đảm bảo mã nguồn nhất quán, dễ đọc và dễ bảo trì. |
| **QA\_TEST\_PLAN.md** | Mô tả chiến lược kiểm thử, môi trường kiểm thử (Staging, Production), và quy trình báo cáo lỗi (Bug Report). | Là tài liệu chính cho đội QA bắt đầu công việc của họ. |

# III. Yêu cầu cụ thể cho từng nhóm tài liệu:

1. **Functional Specifications - Nhóm 2.0**

### 🚀 Các Yêu Cầu Chung

* **Tính Rõ Ràng và Chi tiết (Clarity and Detail):** Mọi tính năng, luồng, và quy tắc nghiệp vụ phải được mô tả một cách rõ ràng, không mơ hồ. Tránh sử dụng ngôn ngữ chung chung.
* **Tính Nhất Quán (Consistency):** Sử dụng cùng một thuật ngữ cho các khái niệm và thành phần trong toàn bộ tài liệu.
* **Tính Khả Thi (Feasibility):** Các yêu cầu phải thực tế và có thể xây dựng được trong khuôn khổ công nghệ và thời gian cho phép.
* **Có Thể Kiểm Thử (Testable):** Mỗi yêu cầu nên được viết sao cho có thể tạo ra các trường hợp kiểm thử (test cases) cụ thể để xác minh tính năng hoạt động đúng.
* **Tài Liệu Sống (Living Document):** Spec nên được cập nhật liên tục mỗi khi có thay đổi hoặc bổ sung yêu cầu.

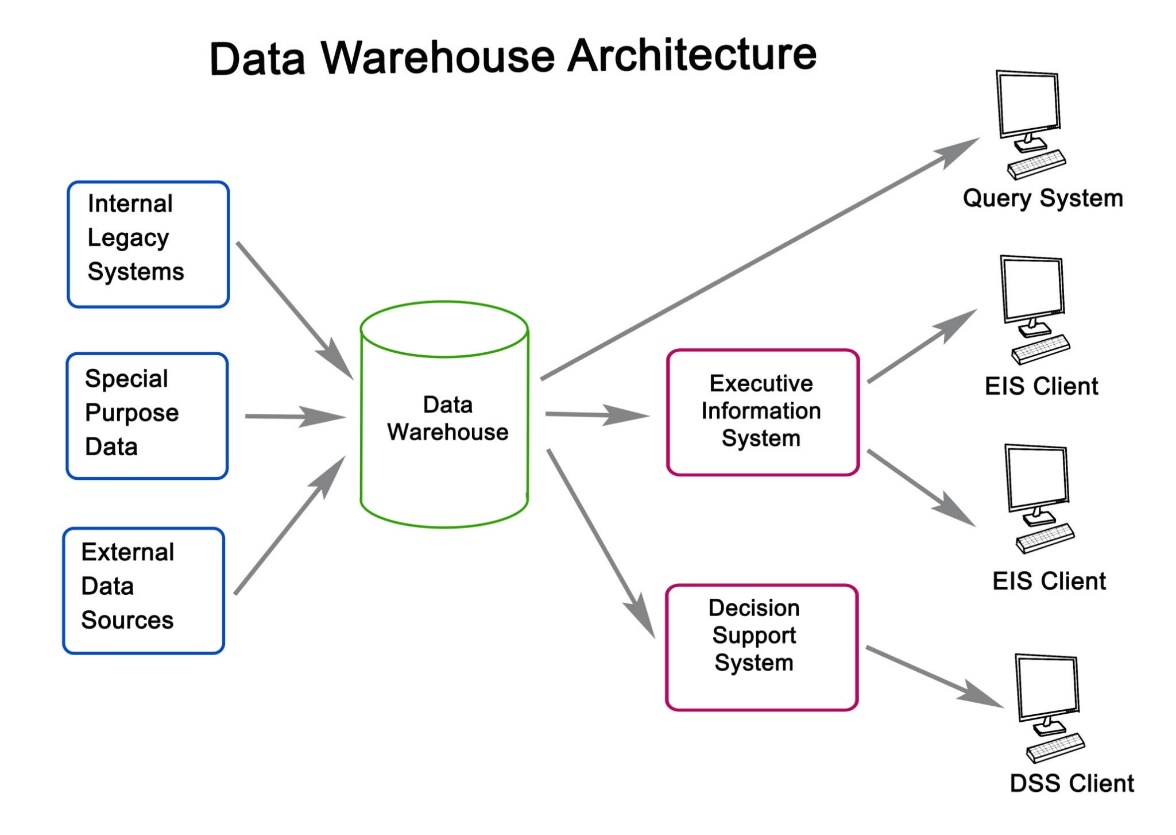
### 📋 Cấu Trúc và Nội Dung Chính của Tài Liệu Spec

**1. Giới Thiệu (Introduction)**

* **Mục đích (Purpose):** Xác định lý do tồn tại của tài liệu này (VD: Dùng làm cơ sở để phát triển và kiểm thử phiên bản X của app).
* **Đối tượng (Audience):** Ai sẽ sử dụng tài liệu này (VD: Team Phát triển, QA, Product Owner).
* **Phạm vi (Scope):** Ứng dụng này là gì, giải quyết vấn đề gì, và những tính năng **không** thuộc phạm vi của phiên bản này (out-of-scope).
* **Định nghĩa/Thuật ngữ (Definitions/Glossary):** Liệt kê và giải thích các thuật ngữ chuyên môn hoặc viết tắt được sử dụng trong tài liệu.

**2. Mô Tả Tổng Quan (Overall Description)**

* **Chân dung người dùng (User Personas):** Mô tả người dùng mục tiêu.
* **Các kịch bản sử dụng (Use Cases/User Stories):** Mô tả cách người dùng tương tác với app để hoàn thành mục tiêu. (Thường viết dưới dạng: **Là một [Vai trò], tôi muốn [Mục tiêu] để [Lợi ích].**)
* **Kiến trúc hệ thống tổng quan (High-Level Architecture):** Nếu cần, mô tả sơ đồ tổng quát về các thành phần chính của hệ thống (Frontend/Backend, Database, API bên ngoài).



Getty Images

Khám phá

**3. Yêu Cầu Chức Năng (Functional Requirements - Điều quan trọng nhất)**

Đây là phần mô tả **các tính năng cụ thể** của ứng dụng. Mỗi tính năng cần được mô tả chi tiết:

* **Tên tính năng:** (VD: Đăng nhập bằng email và mật khẩu)
* **Mô tả:** Chi tiết cách tính năng hoạt động.
  + **Luồng hoạt động (User Flow):** Bước 1, Bước 2, ...
  + **Giao diện người dùng (UI/UX - Thường kèm theo wireframes/mockups):** Mô tả các thành phần trên màn hình.
  + **Quy tắc nghiệp vụ (Business Rules):** Các điều kiện và ràng buộc (VD: Mật khẩu phải có tối thiểu 8 ký tự, bao gồm chữ hoa, chữ thường và số).
  + **Trạng thái hệ thống/Thông báo:** Các thông báo hiển thị khi thành công, thất bại, hoặc lỗi (VD: "Đăng nhập thành công," "Email không tồn tại").
  + **Tương tác với API/Dữ liệu:** Dữ liệu cần thiết để thực hiện tính năng, cách thức lưu trữ hoặc truy xuất.

**4. Yêu Cầu Phi Chức Năng (Non-Functional Requirements)**

Đây là những yêu cầu về chất lượng và giới hạn của hệ thống:

* **Hiệu suất (Performance):** Tốc độ tải trang, thời gian phản hồi API (VD: Màn hình chính phải tải trong vòng dưới 2 giây).
* **Bảo mật (Security):** Yêu cầu mã hóa dữ liệu, xác thực người dùng (VD: Sử dụng HTTPS, lưu trữ mật khẩu đã mã hóa).
* **Khả năng mở rộng (Scalability):** Hệ thống có thể hỗ trợ bao nhiêu người dùng đồng thời.
* **Khả năng tương thích (Compatibility):** Ứng dụng hoạt động trên các nền tảng, hệ điều hành và phiên bản nào (VD: iOS 15+, Android 12+).
* **Khả năng sử dụng (Usability):** Yêu cầu về trải nghiệm người dùng, tuân thủ các hướng dẫn thiết kế (VD: Tuân thủ Human Interface Guidelines của Apple).

**5. Yêu Cầu Thiết Kế và Giao Diện (Design and Interface Requirements)**

**5.1. Tài liệu Thiết kế (Design Assets):** Liên kết đến Figma/Sketch/Zeplin chứa các màn hình và Design System.

**5.2. Thành phần UI (UI Components):** Cách thức hoạt động của các nút, form, và các thành phần tương tác khác.

**5.3. Thông số kỹ thuật:** Kích thước, màu sắc (dùng mã HEX), font chữ.

**Lưu ý:**

1. Loại bỏ mọi đề cập trực tiếp đến Firestore hoặc các tên collection cụ thể (như monthly\_plans, users, employee). Không khoá cứng các yêu cầu kỹ thuật như Framework Tailwind CSS.

2. Thay thế bằng các thuật ngữ chung như "Cơ sở Dữ liệu" (Database), "Bảng" (Table), hoặc "Nơi Lưu Trữ" (Storage).

3. Thêm một lưu ý rõ ràng trong phần Giới Thiệu (Mục 1.3) rằng việc xây dựng cấu trúc DB chi tiết sẽ được thực hiện ở giai đoạn sau khi tất cả các tài liệu đặc tả logic đã được chốt.