

System Specification: Material Database

1. Introduction

The **Material Database** is a mission-critical web application designed for aerospace engineering teams. It serves as the "Single Source of Truth" for all material data, managing the lifecycle from raw material qualification to final assembly integration. It replaces disparate spreadsheets and file systems with a structured, relational, and traceable database.

1.1 Core Objectives

- Traceability:** Ensure every layup and assembly can be traced back to its constituent raw materials and their specific batch/test data.
- Integrity:** Prevent the deletion or modification of data that is currently in use (referential integrity).
- Analysis:** Empower engineers to perform side-by-side comparisons and statistical analysis (A-Basis/B-Basis) to optimize design.
- Standardization:** Enforce compliance with engineering standards through Requirement Profiles.

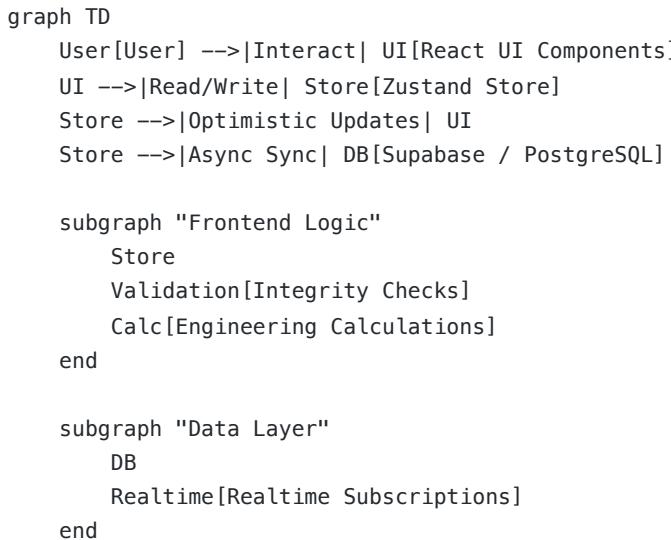
2. Technical Architecture

2.1 Technology Stack

- Frontend:** React 18 (Vite), TypeScript.
- UI Framework:** Tailwind CSS, Shadcn UI (Radix Primitives).
- Charts/Vis:** Recharts.
- State Management:** zustand (Global Store).
- Backend:** Supabase (PostgreSQL).
- Auth:** Supabase Auth (JWT).

2.2 Application Architecture

The application follows a **Client-Heavy** architecture where the majority of business logic resides in the frontend, synchronized with a backend API.



2.3 State Management (`store.ts`)

The `useAppStore` hook acts as the central controller.

- **Fetching:** Data is fetched on demand (e.g., `fetchMaterials`, `fetchLayups`) and cached in memory.
 - **Updates:** Writes are performed optimistically or with immediate feedback. Errors are caught and exposed via the `error` state.
 - **Logic:** Complex operations (like constructing an Assembly with sub-components) are handled here before being sent to the DB.
-

3. Data Model & Logic

3.1 Materials

Represents raw substances (Prepreg, Resin, Core, etc.).

- **Data Structure:**
 - **Core:** `id`, `name`, `manufacturer`, `type`, `status`.
 - **Traceability:** `variants` (Sub-types like "Grade A"), `measurements` (Lab data).
- **Logic:**
 - **Status Flow:** `Draft` → `Standard` → `Obsolete`.
 - **Deletion:** Protected. Before deletion, `checkTraceability(id)` runs to ensure no Layup or Assembly usage.

3.2 Layups

Composite stacks defined by a sequence of layers.

- **Data Structure:**
 - **Layers:** Array of `{ materialVariantId, orientation, sequence }`.
 - **Process:** Linked `ManufacturingProcess` (e.g., Curing Cycle).
- **Calculated Logic:**
 - **Total Thickness:** Sum of (Layer Thickness × Count).
 - **Total Weight:** Sum of (Layer Areal Weight × Count) or calculated via density.
 - Currently, these are often snapshots stored in DB, but recalculated on edit.

3.3 Assemblies

Hierarchical structures combining Layups and Materials.

- **Structure:** Recursive tree. An Assembly contains `AssemblyComponents`.
- **Components:** Can be a `Layup` or a `Material`.
- **Logic:**
 - **Bill of Materials (BOM):** Can be flattened to show total raw material usage.

3.4 Quality & Measurements

- **Measurement:** A single data point or set of points from a lab test.
 - Linked to: `Material`, `Layup`, or `Assembly`.
 - Contains: `resultValue`, `statistics` (Mean, StdDev), `testMethod`.

- **Inheritance:** Layups can "inherit" properties from their constituent materials for theoretical calculations, but also have their own direct measurements.
-

4. Workflows & Page Specifications

4.1 Dashboard (/)

- **Role:** High-level status overview.
- **Key Elements:**
 - **Stats Cards:** Count of active Materials, Layups, Test Runs.
 - **Recent Activity:** Log of changes (Who modified what).
 - **Pending Actions:** "Materials requiring review" (Future).

4.2 Material Management

- **List Page (/materials):**
 - **Table:** Sortable/Filterable by Type, Manufacturer, Status.
 - **Actions:** "View Details", "Add New".
 - **Logic:** Delete button removed to prevent accidental loss; only available in Details.
- **Detail Page (/materials/:id):**
 - **Tabs:**
 - **Overview:** Editable metadata.
 - **Properties:** Read-only view of aggregated measurements.
 - **Variants:** Add/Edit variants (e.g. thickness variations of same prepreg).
 - **Measurements:** Table of linked lab tests.
 - **Usage:** Reverse lookup showing which Layups use this material.
 - **Edit Mode:** Toggles input fields. "Save" triggers `updateMaterial`. "Delete" triggers integrity check.

4.3 Layup Engineering (/layups)

- **List Page:** Similar to Materials.
- **Detail Page / Builder:**
 - **Stack Editor:** Visual drag-and-drop or list interface to reorder layers.
 - **Calculations:** Real-time update of Thickness/Weight as layers are added.
 - **Tabs:**
 - **Standards:** Assign `RequirementProfile` (Spec) to the Layup.
 - **Allowables:** View design values (A-Basis) derived from measurements.

4.4 Assembly Integration (/assemblies)

- **Builder:**
 - **Component Selection:** Picker to add existing Layups/Materials.
 - **Tree View:** Visualize the hierarchy.
 - **Process Definition:** Assign manufacturing steps (Bonding, Painting) to the assembly.

4.5 Quality Control (/quality)

- **Measurement Entry (/quality/test-run):**
 - **Workflow:** Select Entity -> Select Property -> Input Values -> Save.
 - **Logic:** Auto-calculates Mean, Min, Max from a set of values (e.g. 5 coupon tests).
- **Analysis Dashboard (/quality/analysis):**

- **Comparison View:** Select up to 4 entities. Displays properties side-by-side in a table. Highlights differences.
- **Analytics Charts:** Scatter plots / Bar charts visualizing property distributions.
- **Substitution:** Tool to find materials with similar properties (e.g. "Find Resin with Tg > 120°C").

4.6 Quality Assurance & Standardization Logic

This module handles the complex relationships between physical Tests, authorized Laboratories, and engineering Requirements.

4.6.1 Master Data Relationships

1. Properties (**PropertyDefinition**):

- The atomic unit of data (e.g., "Tensile Strength", "Glass Transition Temp").
- Defines data type, potential unit, and category.

2. Test Methods (**TestMethod**):

- Standardized procedures (e.g., "ISO 527-4").
- **Mapping:** Explicitly maps to a subset of **Properties** that this method is capable of measuring.
- *Example:* "ISO 527-4" is configured to measure "Tensile Modulus" and "Tensile Strength".

3. Laboratories (**Laboratory**):

- Testing facilities (Internal or External).
- **Authorization:** Each Lab is explicitly authorized for specific **Test Methods**.
- **Logic:** You cannot select a Lab for a Test Run if it is not authorized for the selected Test Method.

4.6.2 Requirement Profiles (Standards)

Defines the "Acceptance Criteria" for Materials or Layups.

- **Structure:** A collection of **Rules**.

- **Rule Definition:**

- **Target Property:** (e.g., "Tensile Strength")
- **Limits:** Min, Max, Target Value.
- **Method Constraint:** Optionally mandates a specific *Test Method* (e.g., must use "ASTM D3039", not "ISO 527").

- **Usage:** Checked against actual Measurement data to determine "Pass/Fail" status.

4.6.3 Measurement Entry Workflow

The system strictly enforces data integrity during data entry (`/quality/test-run`):

1. **Context Selection:** User selects the Entity (Material/Layup) and the **Test Method** they performed.
2. **Dynamic Interface:** The UI **dynamically generates** the input matrix based *only* on the Properties defined in the selected Test Method.
 - **Effect:** Users cannot accidentally enter "Compression Strength" if the selected method is "Tensile Test".
3. **Statistical Calculation:**
 - **Inputs:** Raw values from multiple pairs/coupons (\$n\$ specimens).
 - **Outputs:** Auto-calculates Mean, Min, Max, StdDev, and CV.

- **Design Values:** If \$n >= 3\$, auto-calculates **A-Basis** (T99) and **B-Basis** (T90) estimates using statistical K-factors (e.g., from CMH-17).
-

5. Settings & Admin (/settings)

- **Access:** Restricted to `Admin` role.
 - **Features:**
 - **User Roles:** Manage permissions.
 - **Drop-down Editor:** Configure options for "Material Type", "Process Type", "Test Methods".
 - **Data Import:** CSV/Excel parsers to bulk loader legacy data.
-

5. Security & Permissions

The system uses a Role-Based Access Control (RBAC) model managed in `auth.ts` and enforced via the `<Protect>` component.

Role	Dashboard	Read Data	Edit Data	Delete Data	Admin/Settings
Viewer	✓	✓	✗	✗	✗
Lab	✓	✓	⚠ (Meas Only)	✗	✗
Engineer	✓	✓	✓	✓	✗
Admin	✓	✓	✓	✓	✓

- **Logic:** `Protect` component hides UI elements (buttons) if the user lacks the required permission scope (e.g. `delete:material`).

6. Future Roadmap (Identified Gaps)

- **Inventory:** Link Database to physical stock/inventory levels.
- **Report Gen:** PDF export of "Material Datasheets" generated from DB values.
- **Approval Workflows:** Formal digital signatures for status changes (Draft -> Standard).