SizeWise HVAC Canvas - Product Requirements Document (PRD)

Version: 1.1.0
Date: 2026-02-16
Status: Phase 1 MVP + Enhancement
Document Owner: Architecture Team

Table of Contents

1. Executive Summary

1.1 Product Vision

SizeWise HVAC Canvas is a professional-grade, desktop-first HVAC design and estimation application t enables HVAC professionals to design ventilation systems, perform engineering calculations, and gene accurate bills of materials.

1.2 Target Users

User Type	Description	Primary Use Cases
HVAC Estimator	Prepares quotes for commercial HVAC projects	Create estimates, generate BOMs,
HVAC Designer	Designs ventilation systems for buildings	Layout ductwork, size equipment,
Kitchen Ventilation Specialist	Focuses on commercial kitchen exhaust systems	Design hood systems, calculate m requirements

1.3 Phase 1 MVP Goals

The Phase 1 MVP focuses on air-side HVAC design with these primary objectives:

1. Canvas-based design workspace for drawing rooms, ducts, and equipment
2. Component Library System for managing catalog items and templates
3. Service System for context-based engineering and visual organization
4. Automation Features including auto-sizing, smart fittings, and bulk operations
5. Real-time HVAC calculations for airflow, velocity, and pressure drop
6. Local-first architecture with .sws file format for project persistence
7. Bill of Materials generation with CRM-compatible CSV export
8. Basic Quoting System with configurable pricing and export scope
9. PDF export for professional documentation

1.4 Success Metrics

Metric	Target	Measurement
Project creation to first entity	< 30 seconds	User testing
Canvas rendering at 500 entities	60 fps	Performance benchmark
File save/load reliability	100%	Automated testing
Calculation accuracy	±1% of manual calculation	Validation testing

2. Functional Requirements

2.1 Dashboard (Project Management)

FR-DASH-001: Project Listing

Display all projects from configured project folder

Show project metadata: name, last modified date, entity count
Support sorting by: name, date modified (default: most recent first)
Cache project metadata in .index.json for fast loading

FR-DASH-002: Create Project
Create new blank project with user-specified name
Save as .sws file in project folder
Initialize with default project metadata and empty canvas state
Navigate to Canvas view upon creation

FR-DASH-003: Open Project
Open existing .sws file from project folder
Validate file schema before loading
Display error if file is corrupted or incompatible version
Navigate to Canvas view upon successful load

FR-DASH-004: Project Actions
Archive: Move project to /archived subfolder
Delete: Move to OS trash/recycle bin (not permanent delete)
Duplicate: Create copy with ” (Copy)” suffix
Rename: Update project name and filename

FR-DASH-005: Recent Projects
Track last 10 opened projects
Display as quick-access list on dashboard
Remove entries if file no longer exists

2.2 Canvas Interface

FR-UI-001: Layout Structure
Left Sidebar: Sizable Drawer Layer containing Project Details, Scope, and Site Conditions.
Right Sidebar: Sizable Drawer Layer containing Bill of Quantities and Calculations.
Bottom Toolbar: Dynamic Sizing Bar containing file operations, process, settings, and notificati
FAB Tool: Floating Action Button triggered by Spacebar (or F key) for quick entity creation. The

FR-UI-002: Left Sidebar (Project Context)
Project Details (Accordion):

Name
Project ID / Case Number (CRM Link)
Location
Client

Project Scope (Accordion):
Scope (Multi-select): HVAC, For future updates
Material (Multi-select with Dropdowns):
Galvanized Steel (G-60, G-90)
Stainless Steel (304S.S., 316S.S., 409S.S., 430S.S., 444S.S.)
Aluminum
PVS
Black Iron / Carbon Steel (16ga, 18ga)
Aluminized Steel
Double-Wall Insulated
Project Type (Dropdown): Residential, Commercial, Industrial

Site Conditions (Accordion):
Elevation (Text/Number)
Outdoor Temperature (Text/Number)
Indoor Temperature (Text/Number)
Wind Speed (Text/Number)
Humidity (Text/Number)

- Local Codes (Text/Number)

FR-UI-003: Right Sidebar (Engineering)

- Bill of Quantities (Accordion):
  - Table with columns: Quantity, Name (Separated by Size, Description, Weight), Description, Weight.
  - Categories: Ducts, Fittings, Equipment, Accessories.
- Calculation (Accordion):
  - Color coding for issues: Inappropriate (Red?), Warning (Yellow), Normal (Green/None).
  - Equipment Hierarchy: Organized by Air System -> Duct Section.
  - Air Volume (Unit Dropdown):
    - English: CFM (Default), CFS, CFH
    - Metric: m^3/s (Default), m^3/h, m^3/m, L/s
  - Air Velocity (Unit Dropdown):
    - English: FPM (Default), FPS
    - Metric: m/s (Default), cm/s
  - Air Pressure (Static & Dynamic):
    - English: in. W.C., psi
    - Metric: Pa, kPa
  - Temperature:
    - English: °F, °R
    - Metric: °C, °K

FR-UI-004: Bottom Toolbar

- Buttons (Icon + Tooltip):
  - File Upload: Upload external files.
  - Export: Opens Export Modal.

  - Process: Triggers calculation engine.
  - Save: Persist current state.
  - Save and Exit: Save and return to dashboard.
  - Exit: Return to dashboard (Confirmation modal if unsaved).
  - Settings:
    - Scale: 1/4” = 1’-0” (English Default) / 1:50 (Metric Default).
    - Unit of Measure: English/Metric.
    - System Toggles: Show/Hide Warnings for Ducts, Equipment, Room.
  - Notification: Opens notification drawer (close button top-left).

FR-UI-005: FAB Tool (Quick Create)

- Trigger: Press Spacebar (or F key). The 'D' key is reserved for the Duct drawing tool.
- Umbrella Menu:
  - Rooms: Select room templates.
  - Ducts: Select duct types.
  - Equipments: Select equipment types.

FR-UI-006: Feedback & Notifications

- Loaders: Three pulsating dots for async operations.
- Toast Notifications:
  - Status: “File uploaded”, “Progress saved”.
  - Interactive Warnings (Click to highlight entity):
    - Ducts: Sizing Issue (Too small/large).
    - Equipment: Capacity Issue (Inadequate/Excessive).
    - Rooms: Size Issue (Too small/large).

FR-UI-007: Device Compatibility

- Detection: Automatically detect device type (PC, Laptop, Tablet, Mobile).
- Behavior: On devices with screen width < 768px (phones/small tablets), enter read■only mode. Users
- Criteria: Screen width < 768px.

FR-UI-008: Responsive Elements

Requirement: Elements (Sidebars, Toolbars) must adjust to screen availability.

Documentation: See docs/guides/RESPONSIVE_DESIGN.md for specific behavior.

2.3 Component Library Management

FR-LIB-001: Unified Component Browser

Hierarchical Navigation: Browse components by Category -> Subcategory -> Type.

Search & Filter: Search by name, description, tags, or manufacturer.

Visual Preview: Display thumbnails for components.

Drag & Drop: (Future) Drag components onto canvas (currently click-to-activate).

FR-LIB-002: Component Management

Create Custom: Create new components from templates.

Edit/Delete: Modify or remove user-created components.

Import/Export: detailed in FR-LIB-003.

FR-LIB-003: Import/Export Components

CSV Import: Bulk import components from CSV with mapping.

JSON Import/Export: Full fidelity transfer of component data.

Validation: Validate imported data against schema.

2.4 Service System

FR-SERV-001: Service Management

Service Definition: Define services (Supply, Return, Exhaust, etc.) with color coding.

Templates: Standard templates (Ashrae colors/standards).

CRUD Operations: Create, Read, Update, Delete services.

FR-SERV-002: Service Assignment

Contextual Assignment: Active service determines properties of new objects.

Inheritance: Ducts inherit service from connected equipment.

Visual Indication: Color-coding of entities on canvas based on service.

2.5 Automation Features

FR-AUTO-001: Auto-Sizing

Reactive Calculation: Triggers on airflow/connection changes.

Velocity Constraints: Sizes based on max/min velocity rules.

User Confirmation: Non-destructive suggestions.

FR-AUTO-002: Fitting Insertion

Auto-Connect: Automatically insert elbows/tees when drawing ducts.

Smart Selection: Choose correct fitting type based on geometry.

FR-AUTO-003: Bulk Operations

Batch Edit: Modify properties of multiple selected entities.

Match Properties: Copy properties from one entity to others.

2.6 Calculations Engine

FR-CALC-001: Room Ventilation (ASHRAE 62.1)

Formula:

$$Vbz = Rp \times Pz + Ra \times Az$$

Where:

Vbz = Breathing zone outdoor airflow (CFM)

Rp = People outdoor air rate (CFM/person)

Pz = Zone population

Ra = Area outdoor air rate (CFM/sq ft)

Az = Zone floor area (sq ft)

Lookup Tables Required:

- Occupancy type → Rp value
- Occupancy type → Ra value
- Occupancy type → default occupant density

FR-CALC-002: ACH to CFM Conversion

Formula:

```
  CFM = (ACH × Volume) / 60
  Where:
    ACH = Air changes per hour
    Volume = Room volume (cu ft)
```

FR-CALC-003: Duct Velocity

Formula:

```
  Velocity (FPM) = CFM × 144 / Area (sq in)
```

Velocity Limits (warnings):

- Residential: 600-900 FPM
- Commercial: 1000-1500 FPM
- Industrial: 1500-2500 FPM
- Kitchen Exhaust: 500 FPM (Min) – 2,500 FPM (Recommended) – 4,000 FPM (Max)
- Generator Exhaust: 2,000-5,000 FPM

FR-CALC-004: Velocity Pressure

Formula:

```
  VP (in.w.g.) = (V / 4005)²
  Where:
    V = Velocity (FPM)
```

FR-CALC-005: Round Duct Sizing

Formula:

```
  Diameter (in) = 13.54 × √(CFM / Velocity)
```

FR-CALC-006: Rectangular Duct Sizing

Formula (equivalent diameter):

```
  De = 1.30 × ((a × b)^0.625) / ((a + b)^0.25)
  Where:
    De = Equivalent diameter
    a, b = Rectangular dimensions
```

FR-CALC-007: Friction Loss (Colebrook-White / Swamee-Jain)

Formula:

```
  1. Reynolds Number:
     Re = (D × V) / (12 × ν)
     (ν = Kinematic viscosity of air ≈ 1.6 × 10^-4 ft²/s)
```

2. Friction Factor (Swamee-Jain):

$f = 0.25 / [\log10((\varepsilon/(3.7D)) + (5.74 / Re^{0.9}))]^2$

3. Pressure Drop:

$\Delta P = f \times (L/D) \times (V/4005)^2$

Where:

ε = Absolute roughness (ft) of material
L = Duct length (ft)
D = Duct diameter (ft)
V = Velocity (FPM)

Friction Factors by Material:

Material	Absolute Roughness (ft)
Galvanized steel	0.0005
Stainless steel	0.0002
Aluminum	0.0002
Flex duct	0.003

Note: Use ε (Absolute roughness) from the table above in the Swamee■Jain friction factor formula.

FR-CALC-008: Fitting Pressure Loss

Method: Equivalent Length

$\Delta P\_fitting = \Delta P\_per\_100ft \times (Le / 100)$

Where:

Le = Equivalent length of fitting (ft)

Industrial Mode (Exhaust/Grease):

Loss Coefficient method:

$\Delta P\_fitting = C \times (V/4005)^2$

Where:

C = Loss coefficient (dimensionless)

Common Fitting Equivalent Lengths:

Fitting Type	Equivalent Length (ft)
90° elbow (round)	5-15 (diameter dependent)
45° elbow	3-8
Tee (branch)	20-30
Reducer	2-5

FR-CALC-009: Calculation Triggers

Recalculate affected entities on property change (300ms debounce)
Recalculate all connected entities on connection change
Display “calculating…” indicator during computation
Cache results until inputs change

FR-CALC-010: Return Air Calculations

Support return air CFM calculations at room and system level
Allow user to designate ducts/equipment as Return
Validate return path continuity (room → return ductwork → return fan/AHU)
Include return airflow results in the Calculations panel

FR-CALC-011: Variable Air Volume (VAV) Systems

Support VAV terminals as equipment entities with:

Minimum airflow (CFM)

    - Maximum airflow (CFM)
    - Control setpoint mode (Cooling, Heating, Auto)
- Support supply duct sizing based on design airflow scenario:
    - Peak (max airflow)
    - Minimum (min airflow)
    - Diversity factor (system-level override)
- Include VAV sizing and validation results in the Calculations panel
- Include VAV-related line items in BOM and Quote outputs when applicable

2.7 Bill of Materials (BOM)

FR-BOM-001: Auto-Generation

- Automatically extract all entities from project
- Group by category: Rooms, Ducts, Fittings, Equipment
- Aggregate quantities (e.g., sum duct lengths by size)
- Update in real-time as canvas changes

FR-BOM-002: BOM Display Panel

- Collapsible panel in canvas view (bottom or side)
- Tree structure: Category → Subcategory → Line items
- Columns: Item, Description, Quantity, Unit

- Sortable by any column

FR-BOM-003: BOM Data Structure

```
 interface BOMLineItem {
   id: string; category: 'duct' | 'fitting' | 'equipment';      subcategory: string;
 // e.g., "Round Duct", "90° Elbow" description: string;              // e.g., "12\" Galva
 nized Round Duct" quantity: number; unit: 'LF' | 'EA' | 'SF';    // Linear feet, Ea
 ch, Square feet material?: string; size?: string; entityIds: string[];          //
 Source entity IDs}
```

2.8 File Management

FR-FILE-001: Project File Format (.sws)

- JSON-based format (human-readable)
- UTF-8 encoding
- Required root fields:

```
{ "schemaVersion": "1.0.0", "projectId": "uuid", "projectName": "string", "crea
tedAt": "ISO8601", "modifiedAt": "ISO8601", "entities": {}, "viewportState": {},
"settings": {}}
```

FR-FILE-002: Save Operations

- Manual save: Ctrl+S → immediate save
- Auto-save: Every 60 seconds if changes exist
- Save indicator: Show “Saving…” then “Saved” with timestamp
- Backup: Create .sws.bak before overwriting

FR-FILE-003: Load Operations

- Validate schema on load
- Migration support: upgrade older schemaVersions
- Error handling: show specific error if load fails
- Recovery: offer to load .sws.bak if main file corrupt

FR-FILE-004: Project Folder Configuration

- Default: ~/Documents/SizeWise Projects

- Configurable in Settings
- Create folder if doesn’t exist
- Validate write permissions

2.9 Export System

FR-EXPORT-001: JSON Export (Full Fidelity)

- Export complete project state as formatted JSON
- Include all entities, history, settings
- Use for backup and inter-system transfer
- File extension: .sws.json

FR-EXPORT-002: CSV Export (BOM & CRM)

- Export Bill of Materials as CSV

- Format optimized for CRM import (Dynamics 365, generic CRM)
- Columns: Case ID, Category, Subcategory, Description, Quantity, Unit, Size, Material, Unit Price,
- Note: Unit Price is derived from a static base price specified in each component. Total Price = Q
- UTF-8 with BOM for Excel compatibility
- Filename: {projectName}_BOM_{date}.csv

FR-EXPORT-003: PDF Export

- Generate professional PDF document
- Scope Selection Dialog: User selects components to include:
    - Drawing / Canvas
    - Bill of Materials (BOM)
    - Formal Quote / Pricing
- Contents (based on selection):
    - Cover page with project info
    - Canvas drawing (fit to page)
    - BOM table
    - Quote summary with pricing
    - Calculation summary
- Page size: Letter (8.5” × 11”) or A4 (configurable)
- Filename: {projectName}_Takeoff_{date}.pdf

FR-EXPORT-004: Export Scope Configuration

- Dialog appears before any export (PDF/CSV)
- Checkbox options:
    - Include Drawing
    - Include BOM
    - Include Quote
- Remembers last used selection

2.10 Settings

FR-SETT-001: User Preferences

- Unit System: Imperial (default) / Metric
- Auto-save interval: 30s / 60s (default) / 120s / Off
- Grid size: 1/4” / 1/2” / 1” (default)
- Theme: Light (default) / Dark

FR-SETT-002: Project Folder

- Current folder path display
- “Change Folder” button with folder picker
- Validate new folder is writable

FR-SETT-003: Keyboard Shortcuts Reference

- Display all keyboard shortcuts

Non-configurable in Phase 1

2.11 Migration & Backward Compatibility

FR-MIG-001: Auto-Migration for Legacy Projects

Automatic Upgrades: Detect and upgrade older .sws schema versions upon load.
Versioning: Store schema version in project file metadata.
Backup: Create a non-destructive backup (.sws.bak) before any migration.
Validation: Run full project validation post-migration to flag unresolved items.

FR-MIG-002: Service Mapping

Default Assignment: Assign default “General Supply” service to legacy entities lacking service co execution.
Entity Mapping: Map legacy entity properties to new baseline component definitions where possible
Fallback: Mark unmappable items as “Custom” with preserved raw properties.

FR-MIG-003: User Resolution Wizard

Trigger: Launch wizard if validation errors occur during migration.
Flow:

1. Display summary of migrated items vs. items requiring attention.
2. Allow user to map unknown materials to new Component Library definitions.
3. Allow user to assign Services to bulk groups of entities.
4. Confirm and finalize migration.

FR-MIG-004: Graceful Degradation

Legacy Rendering: Fall back to basic geometric rendering if component assets are missing.
Calculation Defaults: Use conservative defaults (e.g., standard friction factors) if specific eng missing from legacy files.

2.12 Quoting System

FR-QUOTE-001: Pricing Configuration

Input base costs for material categories (e.g., per lb, per ft)
Configure markup percentages
Set labor rates (per hour) and installation time factors

FR-QUOTE-002: Quote Generation

Calculate total project cost based on BOM quantities
Apply markups and labor calculations
Generate line-item pricing

3. Technical Requirements

3.1 Technology Stack

Component	Technology	Version	Purpose
Desktop Runtime	Tauri	2.x	Native desktop wrapper, filesystem access
Frontend Framework	Next.js	16.x	React framework with routing
UI Library	React	18.3.x	Component-based UI
State Management	Zustand	5.x	Lightweight, performant state
Schema Validation	Zod	4.x	Runtime type validation
Canvas Rendering	Pure Canvas 2D	-	No wrapper libraries; Fabric.js explicitly re
Local Cache	localStorage	-	Browser-based persistence for web; .sws file on Desktop

Component	Technology	Version	Purpose
Language	TypeScript	5.x	Type-safe development
Package Manager	pnpm	8.x	Fast, disk-efficient
Testing	Vitest	1.x	Unit/integration testing
E2E Testing	Playwright	1.x	End-to-end testing

Canvas Performance Considerations

Requirement: Without third-party wrapper libraries, implement a spatial indexing structure (e.g.,

3.2 Architecture Patterns

3.2.1 Command Pattern (State Mutations)

All state changes MUST be executed via commands:

```
interface Command {
  id: string;              // ULID for ordering                    type: string;            // e.
EATE_ROOM", "UPDATE_DUCT" payload: unknown;                           // Command-specific data ti
p: number;       // Unix timestamp}
interface ReversibleCommand extends Command {
  inverse: Command;        // For undo}
```

3.2.2 Normalized Entity State

Entities stored flat by ID, never nested:

```
interface EntityState {
  byId: Record<string, Entity>;                 allIds: string[];        selectedIds: string[];}
```

3.2.3 Hook Naming Convention

All Zustand hooks follow this naming pattern:

useXState - Read-only state access

useXActions - Write operations / mutations

useXQuery - Async data fetching / IO

useX - Composed convenience hook combining the above

3.2.4 Feature Slices

Each feature is self-contained with its own:

- Store slice
- Components
- Hooks
- Types

```
/features
  /canvas
    /components              # Canvas-specific UI
    /hooks                   # useCanvas, useSelection, etc.
     /store                  # Canvas Zustand slice
     /types                  # Canvas types
     index.ts                # Public exports
```

3.3 Data Models

3.3.1 Base Entity Schema

```
interface BaseEntity {
  id: string;                         // UUID v7
  type: EntityType;                   // 'room' | 'duct' | 'equipment' | 'fitting' | 'note' | 'gr
oup'
  transform: Transform;
```

```
    zIndex: number;
    createdAt: string;     // ISO8601
    modifiedAt: string;    // ISO8601
}
interface Transform {
  x: number;              // Pixels from origin    y: number;   rotation: number;
// Degrees scaleX: number; scaleY: number;}
```

3.3.2 Room Entity

```
interface Room extends BaseEntity {
  type: 'room'; props: {
    name: string;                    // 1-100 chars     width: number;
// inches, 1-10,000    length: number;                   // inches, 1-10,000    height:
number;                   // inches, 1-500    occupancyType: OccupancyType; airChang
esPerHour: number;        // 1-100    notes?: string; }; calculated: {
     area: number;                    // sq ft    volume: number;               // c
u ft     requiredCFM: number; };}
type OccupancyType = | 'office' | 'retail' | 'restaurant' | 'kitchen_commercial'
| 'warehouse' | 'classroom' | 'conference' | 'lobby';
```

3.3.3 Duct Entity

```
interface Duct extends BaseEntity {
   type: 'duct'; props: {
     name: string;    shape: 'round' | 'rectangular';    diameter?: number;
// round only, inches     width?: number;                  // rectangular only, inches
height?: number;                  // rectangular only, inches     length: number;
// feet     material: DuctMaterial;    airflow: number;                  // CFM (canonic
al field name)     staticPressure: number;          // in.w.g.     connectedFrom?: strin
g;           // Entity ID    connectedTo?: string;             // Entity ID }; calcula
ted: {
    area: number;                    // sq in    velocity: number;                    // F
PM     frictionLoss: number;           // in.w.g./100ft };}
type DuctMaterial = 'galvanized' | 'stainless' | 'aluminum' | 'flex' | 'carbon_steel' | 'black_iron
```

3.3.4 Equipment Entity

```
interface Equipment extends BaseEntity {
  type: 'equipment'; props: {
    name: string;    equipmentType: EquipmentType;     manufacturer?: string;       modelN
umber?: string;    capacity: number;                // CFM     staticPressure: number;
// in.w.g.    width: number;                   // inches     depth: number;
// inches    height: number;                  // inches };}
type EquipmentType = 'hood' | 'fan' | 'diffuser' | 'damper';
```

3.3.5 Fitting Entity

```
interface Fitting extends BaseEntity {
  type: 'fitting'; props: {
    fittingType: FittingType;    angle?: number;                    // degrees (for elbo
ws)    inletDuctId?: string;    outletDuctId?: string; }; calculated: {
    equivalentLength: number;        // feet    pressureLoss: number;               // i
n.w.g. };}
type FittingType = 'elbow_90' | 'elbow_45' | 'tee' | 'reducer' | 'cap';
```

3.3.6 Unified Component Definition

```
// Unified Component Definition
interface UnifiedComponentDefinition {
  // Identity
  id: string;
  name: string;
  category: 'duct' | 'fitting' | 'equipment' | 'accessory';
  type: string;
    subtype?: string;                    // e.g., "rectangular", "round", "elbow", "tee"

    // Catalog Information
    manufacturer?: string;
    model?: string;
    partNumber?: string;
    sku?: string;
    description?: string;
    thumbnail?: string;

    // Service Integration
    systemType?: 'supply' | 'return' | 'exhaust';
    pressureClass?: 'low' | 'medium' | 'high';


    // Engineering Properties
    engineeringProperties: {
      frictionFactor: number;            // Darcy friction factor
      maxVelocity: number;               // Max recommended air velocity (fpm)
      minVelocity?: number;              // Min recommended air velocity (fpm)
      pressureCoefficient?: number;      // Pressure loss coefficient
      maxPressureDrop?: number;          // in. w.g./100 ft
         roughness?: number;             // Absolute roughness (ft)
    };

    // Pricing
    pricing: {
      basePrice: number;                // Static unit price for exports
      materialCost: number;
      laborUnits: number;                // Labor hours per unit
         laborRate?: number;             // $/hour (can override project rate)
         wasteFactor: number;            // 0.0 to 1.0 (10% waste = 0.10)
         markup?: number;                // Markup percentage override
         notes?: string;
    };

    // Materials
    materials: MaterialSpec[];
    defaultDimensions?: Record<string, number>;

    // Metadata
    tags?: string[];
    customFields?: Record<string, unknown>;
    isCustom: boolean;                   // User-created vs. system component
    createdAt?: Date;
    updatedAt?: Date;
}

// Material Specification
interface MaterialSpec {
  id: string;
    name: string;
    grade?: string;                      // e.g., "24-gauge", "26-gauge"
    type: 'galvanized_steel' | 'stainless_steel' | 'aluminum' | 'fiberglass' | 'flexibl
```

```
e';
  cost: number;                        // Cost per unit
  costUnit: 'linear_foot' | 'square_foot' | 'piece';
  properties?: {
         weight?: number;                   // lb/ft² or lb/ft
         thermalConductivity?: number;
         maxTemperature?: number;           // °F
    };
}

// Hierarchical Categories
interface ComponentCategory {
  id: string;
  name: string;
  parentId: string | null;
  description?: string;
  icon?: string;
  subcategories?: ComponentCategory[];
}

// Component Templates
interface ComponentTemplate {
  id: string;
  name: string;
  description?: string;
    componentId: string;                 // Base component this template is for
    dimensions?: Record<string, number>;
    materialId?: string;                 // Preset material selection
    engineeringOverrides?: {
      airflow?: number;
      velocity?: number;
      pressureClass?: string;
    };
    createdBy?: string;
    isShared: boolean;
    createdAt?: Date;
}
```

3.3.7 Service Definition

```
// Service Definition
interface Service {
  id: string;                    // UUID v4
  name: string;                  // 1+ characters
  description?: string;          // Optional description
  // Broader enum set (fresh_air, relief_air, other) is a conscious deviation from ori
ginal plan for completeness
  systemType: 'supply' | 'return' | 'exhaust' | 'fresh_air' | 'relief_air' | 'other';
  material: 'galvanized' | 'stainless' | 'aluminum' | 'fiberglass' | 'flexible' | 'carbon_steel' |
  pressureClass: 'low' | 'medium' | 'high' | 'low-pressure' | 'medium-pressure' | 'hig
h-pressure';
```

```
    // Dimensional Constraints - Required
    dimensionalConstraints: {
      minDiameter: number;
      maxDiameter: number;
         minWidth: number;
         maxWidth: number;
         minHeight: number;
```

```
         maxHeight: number;
         allowedShapes: ('round' | 'rectangular')[];
    };

    // Fitting Rules
    fittingRules: Array<{
      angle: number;                 // e.g. 90, 45
      fittingType: string;           // e.g. "elbow_90_stamped", "elbow_90_gore"
      preference: string;            // "Preferred", "Standard", "Avoid"
    }>;

    // Preferences
    manufacturerPreferences: string[];    // Ranked list of preferred manufacturers


  // Metadata
  source: 'baseline' | 'custom';          // 'baseline' = system template, 'custom' = user
-created
  color: string;                          // Visual override color (hex), required
  createdAt?: Date;
  updatedAt?: Date;
}


// Service Template (baseline)
interface ServiceTemplate extends Omit<Service, 'id' | 'createdAt' | 'updatedAt'> {
  id: string;                         // Static string IDs like 'tmpl_low_pressure_sup
ply'
  isTemplate: true;
}
```

3.3.8 Note Entity

```
interface Note extends BaseEntity {
  type: 'note';
  props: {
     content: string;
     fontSize: number;
     color: string;
  };
}
```

3.3.9 Group Entity

```
interface Group extends BaseEntity {
  type: 'group';
  props: {
     name: string;
     childIds: string[];
```

```
   };
 }
```

3.3.10 Automation Data Structures

```
 // Validation Result
 interface ValidationResult {
   entityId: string;
   serviceId: string;
   catalogStatus: 'valid' | 'invalid' | 'unknown';
   catalogMessage?: string;
```

```
    violations: Array<{
      ruleId: string;
      severity: 'error' | 'warning' | 'info';
      message: string;
    }>;
}

// Auto-Sizing Result
interface SizeOption {
  size: {
         diameter?: number;
         width?: number;
         height?: number;
    };
    velocity: number;                     // FPM
    pressureDrop: number;                 // in.w.g./100ft
    recommendation: string;               // "Optimal", "Low velocity", etc.
}

// Parametric Update Result
interface ParametricUpdateResult {
    updatedEntities: string[];            // IDs of entities that were updated
    violations: Array<{
      entityId: string;
      message: string;
    }>;
    requiresUserAction: boolean;
    entityUpdates?: Array<{
         id: string;
         updates: Partial<Entity>;
         previous: Entity;
    }>;
    engineeringData?: DuctEngineeringData;
}

// Validation Dashboard Data (from ValidationAggregationService)
interface ValidationDashboardData {
  summary: {
         totalEntities: number;
         validEntities: number;
         entitiesWithErrors: number;
         entitiesWithWarnings: number;
         validationPercentage: number;
    };
    byCategory: {
         ducts: CategoryValidation;
         fittings: CategoryValidation;
         equipment: CategoryValidation;
    };
    topIssues: ValidationIssue[];
    recentlyValidated: Date;
}

interface CategoryValidation {
    total: number;
    valid: number;
    errors: number;
    warnings: number;
}
```

```
// Validation Issue (from ValidationAggregationService)
interface ValidationIssue {
  entityId: string;
    entityType: 'duct' | 'fitting' | 'equipment';
    severity: 'error' | 'warning' | 'info';
    type: string;
    message: string;
    count: number;                                      // How many entities have this issue
}

// Constraint Status (used within entities)
interface ConstraintStatus {
  isValid: boolean;
    violations: Array<{
      type: string;
       severity: 'error' | 'warning' | 'info';
       message: string;
       field?: string;
    }>;
}

// Bulk Edit Changes
interface BulkEditChanges {
  material?: string;
    insulation?: string;
    costMarkup?: number;
    systemAssignment?: string;
    category?: string;
}

// Update Dependency (for parametric updates)
interface UpdateDependency {
    entityId: string;
    entityType: 'duct' | 'fitting' | 'equipment';
    updateType: 'recalculate' | 'validate' | 'propagate';
}
```

3.4 State Management Architecture

3.4.1 Component Library Store

Store: componentLibraryStoreV2

Persistence: localStorage (key: sws.componentLibrary.v2)
Structure: Normalized components by ID, hierarchical category tree
Middleware: Zustand persist middleware with Immer for immutability
Key Actions:
addComponent : Add new component definition
updateComponent : Modify existing component
deleteComponent : Remove custom component
duplicateComponent : Create copy of existing component
getComponent : Retrieve component by ID
activateComponent : Set active component for tool system
deactivateComponent : Clear active component
getActiveComponent : Get currently active component
search : Fuzzy search by name, description, tags

setSearchQuery : Update search query
setFilterTags : Filter by tags
setSelectedCategory : Filter components by category/subcategory
getByCategoryTree : Get components within category hierarchy
getFilteredComponents : Get components matching all active filters
addCategory / updateCategory / deleteCategory : Manage category tree
addTemplate / updateTemplate / deleteTemplate : Manage component templates
setEnabled / setLoading / setError : UI state management
reset : Reset store to initial state

Note: Import/export functionality is handled by separate services, not directly by the store. See ComponentImportService for CSV/JSON import operations.

3.4.2 Service Store

Store: serviceStore
Persistence: Project file (.sws)
Structure: Record of custom services keyed by ID, baseline templates (read-only), active service
Middleware: Zustand with Immer middleware
Key Actions:
addService : Create new service definition
updateService : Modify custom service properties (baseline services are read-only)
removeService : Remove custom service (baseline services cannot be removed)
setActiveService : Set current working service context (can be custom service or template
cloneService : Create new custom service from template or existing service
loadBaselineTemplates : Load system service templates
hydrate : Restore store state from project file

Note: Service assignment to entities is handled through the entity’s serviceId property in the entit through a store action. Use entityStore.updateEntity(entityId, { serviceId: '...' }) to assign servi

3.4.3 Validation Store

Store: validationStore
Persistence: In-memory (derived from entity state)
Structure: Aggregated validation results grouped by severity and category
Key Features:
Real-time validation aggregation from entity.warnings
Dashboard counts by severity (error, warning, info)
Filter by severity, category, or service
Issue navigation (click to select entity)
Resolution tracking

3.4.4 Calculation Settings Store

Store: calculationSettingsStore
Persistence: Project file (.sws)

```
  Structure: Calculation templates and active settings
  Templates:
      Commercial Standard ($65/hr labor, 15% markup)
```

```
       Residential Budget ($45/hr labor, 10% markup)
       Industrial Heavy ($85/hr labor, 20% markup)
    Key Actions:
        applyTemplate   : Apply predefined template
        updateSettings   : Modify custom settings
        createCustomTemplate   : Save current settings as template
```

3.5 Project File Schema

```
 interface ProjectFile {
   schemaVersion: string;                              // Semantic version     projectId: string;
 // UUID       projectName: string;            projectNumber?: string;      clientName?: string;
 t: string;                              // ISO8601     modifiedAt: string;                  // ISO
 entities: {
        byId: Record<string, Entity>;                  allIds: string[]; }; viewportState: {
        panX: number;    panY: number;                  zoom: number; }; settings: {
        unitSystem: 'imperial' | 'metric';                   gridSize: number;      gridVisible: boo
 };     commandHistory?: {
        commands: Command[];               currentIndex: number;      };}
```

3.6 Folder Structure

```
 hvac-design-app/
 ■■■ src/
 ■   ■■■ app/                                    # Next.js routes
 ■      ■    ■■■ (main)/
 ■      ■    ■   ■■■ dashboard/                  # Dashboard page
 ■      ■    ■      ■■■ canvas/[id]/             # Canvas page with project ID
 ■      ■    ■      ■■■ settings/                # Settings page
 ■      ■    ■      ■■■ help/                    # Help page
 ■      ■    ■■■ layout.tsx
 ■      ■    ■■■ page.tsx                        # Redirect to dashboard
 ■      ■
 ■      ■■■ components/                          # Shared UI components
 ■      ■    ■■■ ui/                             # Primitives (Button, Input, etc.)
 ■      ■    ■■■ layout/                         # Layout components
 ■      ■    ■■■ ErrorBoundary.tsx
 ■      ■
 ■      ■■■ features/                            # Feature modules
 ■      ■    ■■■ dashboard/
 ■      ■    ■   ■■■ components/
 ■      ■    ■      ■■■ hooks/
 ■      ■    ■      ■■■ store/
 ■      ■    ■      ■■■ index.ts
 ■      ■    ■
 ■      ■    ■■■ canvas/
 ■      ■    ■      ■■■ components/
 ■      ■    ■      ■   ■■■ Canvas.tsx
 ■      ■    ■      ■      ■■■ Toolbar.tsx
 ■      ■    ■      ■      ■■■ Inspector/
 ■      ■    ■      ■      ■■■ BOMPanel.tsx
 ■      ■    ■      ■■■ hooks/
 ■      ■    ■      ■   ■■■ useCanvas.ts
 ■      ■    ■      ■      ■■■ useSelection.ts
```

```
■      ■    ■      ■      ■■■ useViewport.ts
■      ■    ■      ■   ■■■ useTools.ts
■      ■    ■      ■■■ store/
```

PRD

```
■   ■    ■    ■    ■■■ canvasStore.ts
■   ■    ■    ■   ■■■ selectors.ts
■   ■    ■    ■■■ tools/
■   ■    ■    ■    ■■■ SelectionTool.ts
■   ■    ■    ■    ■■■ RoomTool.ts
■   ■    ■    ■    ■■■ DuctTool.ts
■   ■    ■    ■   ■■■ EquipmentTool.ts
■   ■    ■    ■■■ calculators/
■   ■    ■    ■    ■■■ ventilation.ts
■   ■    ■    ■    ■■■ ductSizing.ts
■   ■    ■    ■   ■■■ pressureDrop.ts
■   ■    ■    ■■■ index.ts
■   ■    ■
■   ■    ■■■ export/
■   ■        ■■■ csv.ts
■   ■         ■■■ json.ts
■   ■         ■■■ pdf.ts
■   ■
■   ■■■ core/                        # Shared utilities
■   ■    ■■■ schema/                 # Zod schemas
■   ■    ■■■ store/                  # Zustand helpers
■   ■    ■■■ commands/               # Command infrastructure
■   ■    ■■■ persistence/            # File I/O
■   ■    ■■■ geometry/               # Math utilities
■   ■
■   ■■■ styles/                      # Global styles
■        ■■■ theme.ts
■
■■■ src-tauri/                       # Tauri backend
■   ■■■ src/
■   ■   ■■■ main.rs
■   ■■■ Cargo.toml
■   ■■■ tauri.conf.json
■
■■■ public/
■   ■■■ templates/                   # Project templates
■   ■■■ data/                        # ASHRAE lookup tables
■
■■■ tests/
    ■■■ unit/
    ■■■ integration/
    ■■■ e2e/
```

4. User Stories

4.1 Project Management

US-PM-001: Create New Project

As an HVAC estimator
I want to create a new blank project
So that I can start designing a new HVAC system

Flow:

1. User opens application → Dashboard displayed

2. User clicks “New Project” button
3. Dialog appears for project name input
4. User enters name and clicks “Create”
5. New .sws file created in project folder
6. Canvas view opens with empty workspace

US-PM-002: Open Existing Project
As an HVAC designer
I want to open a previously saved project
So that I can continue working on my design
Flow:

1. User views project list on Dashboard
2. User double-clicks project or clicks “Open”
3. Project file loaded and validated
4. Canvas view opens with project contents

US-PM-003: Recent Projects Quick Access
As a returning user
I want to quickly access my recent projects
So that I can resume work without searching
Flow:

1. Dashboard shows “Recent Projects” section
2. Last 10 opened projects displayed with thumbnails
3. User clicks project to open immediately

4.2 Canvas Design

US-CD-001: Draw Room
As an HVAC designer
I want to draw a room on the canvas
So that I can define the space requiring ventilation
Flow:

1. User selects Room tool (R key or toolbar)
2. User clicks to set first corner
3. User moves mouse to define dimensions
4. Room preview shows width × length
5. User clicks to confirm placement
6. Room entity created with default properties
7. Inspector panel shows room properties

US-CD-002: Draw Duct
As an HVAC designer
I want to draw ductwork connecting rooms and equipment
So that I can define the airflow path
Flow:

1. User selects Duct tool (D key)
2. User clicks start point (snaps to equipment/room)
3. User drags to end point
4. User releases to confirm
5. Duct entity created with default properties
6. Connections established to snapped entities

US-CD-003: Place Equipment
As an HVAC designer
I want to place HVAC equipment on the canvas
So that I can specify fans, hoods, and other devices
Flow:

1. User selects Equipment tool (E key)
2. Equipment palette/submenu appears

3. User selects equipment type (e.g., “Hood”)
4. User clicks canvas to place
5. Equipment entity created at click position

US-CD-004: Edit Entity Properties

As an HVAC designer

I want to modify entity properties in the inspector

So that I can specify exact dimensions and parameters

Flow:

1. User selects entity on canvas
2. Inspector panel shows entity properties
3. User modifies values (e.g., room height)
4. Validation occurs in real-time
5. Calculations update automatically
6. Canvas rendering updates

US-CD-005: Undo/Redo Actions

As a user

I want to undo and redo my actions

So that I can correct mistakes easily

Flow:

1. User performs action (e.g., delete entity)
2. User presses Ctrl+Z
3. Action is reversed (entity restored)
4. User presses Ctrl+Y
5. Action is re-applied (entity deleted again)

4.3 Calculations

US-CALC-001: View Room Ventilation Requirements

As an HVAC designer

I want to see calculated CFM requirements for rooms

So that I can size my ductwork appropriately

Flow:

1. User selects a room
2. Inspector shows room properties
3. Calculated section displays:

- Area (sq ft)
- Volume (cu ft)
- Required CFM (based on occupancy type)

4. Values update when properties change

US-CALC-002: View Duct Performance

As an HVAC designer

I want to see duct velocity and pressure drop

So that I can verify proper sizing

Flow:

1. User selects a duct
2. Inspector shows duct properties
3. Calculated section displays:

- Velocity (FPM)
- Friction loss (in.w.g./100ft)

4. Warnings shown if velocity out of range

4.4 Export

US-EXP-001: Export Bill of Materials

As an HVAC estimator

I want to export a CSV of all materials

So that I can create quotes and order materials

Flow:

1. User clicks Export → CSV (BOM)
2. Save dialog appears with default filename
3. User confirms location
4. CSV file generated with all materials
5. Success toast shown

US-EXP-002: Export PDF Documentation

As an HVAC professional
I want to export a PDF of my design
So that I can share with clients and contractors
Flow:

1. User clicks Export → PDF
2. Export options dialog (page size, include BOM)
3. User clicks “Export”
4. PDF generated with drawing and tables
5. PDF opened in default viewer

4.5 Component Library

US-LIB-001: Browse Components

As an HVAC designer
I want to browse and filter the component library
So that I can find specific parts for my design
Flow:

1. User opens “Library” panel
2. User selects category (e.g. “Ducts > Round”)
3. User types “Spiral” in search
4. List updates to visually show matching components

US-LIB-002: Import Catalog

As an HVAC estimator
I want to import my company’s part catalog from CSV
So that I can use accurate part numbers and pricing
Flow:

1. User clicks “Import” in Library
2. Selects CSV file
3. Maps columns (SKU, Cost, etc.)
4. Validates preview
5. Components added to “Custom” library

4.6 Services

US-SERV-001: Define Service

As an HVAC lead
I want to define a “High Pressure Supply” service
So that my team uses consistent specifications
Flow:

1. User opens “Services” panel
2. Clicks “Create New”
3. Sets Name=“Supply HP”, Color=Blue, System=Supply
4. Sets standard constraints (e.g. Min Velocity=1500)
5. Saves service

US-SERV-002: Visual Verification

As an HVAC designer

I want to see services color-coded on the canvas
So that I can visually verify system separation

Flow:
1. User enables “Service Coloring” toggle
2. Supply ducts turn Blue, Return ducts turn Pink
3. User identifies a Return duct connected to Supply system (mismatch)

4.7 Automation

US-AUTO-001: Auto-Size Duct Run

As an HVAC designer
I want the system to suggest duct sizes based on airflow
So that I can ensure proper velocity without manual calculation
Flow:
1. User sets airflow at terminal (e.g. 500 CFM)
2. System calculates required size for upstream ducts
3. Non-intrusive notification: “Sizing update available”
4. User clicks “Apply” to resize ducts to recommended dimensions

US-AUTO-002: Auto-Insert Fittings

As an HVAC designer
I want elbows to appear automatically when I turn corners
So that I can draw quickly without interruption
Flow:
1. User draws duct segment A
2. User continues drawing segment B at 90 degrees
3. System automatically inserts 90-degree elbow at vertex
4. Connections established A -> Elbow -> B

US-AUTO-003: Parametric Updates

As an HVAC designer
I want connected fittings to resize when I change duct dimensions
So that I don’t have to manually update every connected component
Flow:
1. User selects duct and changes diameter from 12” to 14”
2. System detects connected fittings (elbows, tees)
3. Parametric update service resizes fittings to match new diameter
4. Connected ducts inherit size changes if part of same system
5. Brief visual highlight (0.5s) indicates affected entities
6. Single undo operation reverses all changes

US-AUTO-004: Bulk Operations

As an HVAC estimator
I want to update properties on multiple selected entities at once
So that I can efficiently make project-wide changes
Flow:
1. User multi-selects 20 duct segments (Ctrl+click or drag selection)
2. User opens context menu and selects “Bulk Edit”
3. Bulk Edit dialog shows common properties that can be changed
4. User changes material from “Galvanized” to “Stainless”
5. Preview shows affected entities highlighted
6. User clicks “Apply” to update all selected entities
7. Single undo operation reverses all changes

4.8 File Management

US-FM-001: Auto-Save Project

As a user
I want my work to be automatically saved

So that I don’t lose progress if something goes wrong

Flow:
1. User makes changes to project
2. After 60 seconds of inactivity, auto-save triggers
3. “Saving…” indicator appears briefly
4. .sws.bak created before save
5. .sws file updated
6. “Saved” indicator with timestamp

US-FM-002: Recover from Backup
As a user
I want to recover from a backup file
So that I can restore my work after corruption
Flow:
1. User attempts to open corrupted .sws file
2. Error dialog: “File appears corrupted”
3. Dialog offers: “Load backup (.sws.bak)?”
4. User clicks “Yes”
5. Backup file loaded instead

5. Acceptance Criteria

5.1 Dashboard

ID	Criterion	Validation Metho
AC-DASH-001	Project list loads within 2 seconds for 100 projects	Performance test
AC-DASH-002	New project creation navigates to canvas in < 1 second	User testing
AC-DASH-003	Deleted projects move to system trash, not permanent delete	Manual test
AC-DASH-004	Recent projects list updates on project open	Automated test
AC-DASH-005	Project folder change persists across app restarts	Integration test

5.2 Canvas

ID	Criterion	Validation Method
AC-CANV-001	Canvas maintains 60fps with 500 entities	Performance benchmark
AC-CANV-002	Pan and zoom respond within 16ms (one frame)	Performance test
AC-CANV-003	Grid snapping accurate to 0.1 pixel	Unit test
AC-CANV-004	Room tool creates valid entity with all required props	Unit test
AC-CANV-005	Selection tool supports multi-select via Shift+click	E2E test
AC-CANV-006	Undo/redo stack maintains 100 commands minimum	Unit test
AC-CANV-007	Entity deletion removable via Delete and Backspace	E2E test
AC-CANV-008	Keyboard shortcuts match specification	E2E test

5.3 Inspector

ID	Criterion	Validation Method
AC-INSP-001	Inspector updates within 100ms of selection change	Performance test
AC-INSP-002	Validation errors display with red border and message	Visual test
AC-INSP-003	Calculated fields are read-only with gray background	Visual test
AC-INSP-004	Property changes save without explicit save action	Integration test
AC-INSP-005	All field validations match specification ranges	Unit test

5.4 Calculations

ID	Criterion	Validation Method
AC-CALC-001	ASHRAE 62.1 calculation within ±1% of manual	Validation test
AC-CALC-002	Duct velocity calculation matches formula exactly	Unit test

ID	Criterion	Validation Method
AC-CALC-003	Friction loss calculation uses correct roughness values	Unit test

AC-CALC-004	Calculations trigger within 300ms of input change	Performance test
AC-CALC-005	Velocity warnings appear when outside limits	Unit test

5.5 Component Library

ID	Criterion	Validation Method
AC-LIB-001	Browser handles 1000+ items with < 500ms delay	Performance test
AC-LIB-002	Search filters results in real-time (< 100ms)	Unit test
AC-LIB-003	CSV import validates schema and rejects bad rows	Integration test
AC-LIB-004	Custom components persist across app restarts	E2E test

5.6 Services

ID	Criterion	Validation Method
AC-SERV-001	New entities inherit service from connection	Unit test
AC-SERV-002	Service colors render correctly on canvas	Visual test
AC-SERV-003	Changing service triggers re-validation	Integration test

5.7 Automation

ID	Criterion	Validation Method
AC-AUTO-001	Fittings insert automatically on 90/45 degree turns	E2E test
AC-AUTO-002	Auto-sizing suggests sizes within velocity limits	Unit test
AC-AUTO-003	Bulk edit updates 50+ entities in < 1s	Performance test
AC-AUTO-004	Parametric updates propagate flow correctly	Unit test
AC-AUTO-005	Parametric changes create single undo group	Unit test
AC-AUTO-006	Visual highlight appears for 0.5s on affected entities	Visual test
AC-AUTO-007	Bulk operations show preview before apply	E2E test
AC-AUTO-008	Auto-sizing triggers within 300ms of airflow change	Performance test

5.8 File Management

ID	Criterion	Validation Method
AC-FILE-001	Save completes within 500ms for 500-entity project	Performance test
AC-FILE-002	Load completes within 1 second for 500-entity project	Performance test
AC-FILE-003	.sws.bak created before every save	Integration test
AC-FILE-004	Schema validation rejects invalid files with message	Unit test
AC-FILE-005	Auto-save triggers at configured interval	Integration test

5.9 Export

ID	Criterion	Validation Method
AC-EXP-001	CSV export contains all BOM columns	Integration test
AC-EXP-002	CSV is UTF-8 with BOM for Excel compatibility	Unit test
AC-EXP-003	JSON export is valid and re-importable	Round-trip test
AC-EXP-004	PDF contains canvas drawing and BOM table	Manual test
AC-EXP-005	Export filename follows naming convention	Unit test

6. Non-Functional Requirements

6.1 Performance

Requirement	Target	Measurement
Canvas Frame Rate	60fps sustained	Profiler at 500 entities
First Paint	< 1 second	Lighthouse
Project Load	< 1 second	Timer for 500 entities
Project Save	< 500ms	Timer for 500 entities
Calculation Response	< 300ms	Timer from input to display
Memory Usage	< 500MB	Task manager at 500 entities
Application Size	< 100MB	Installer size

6.2 Reliability

Requirement	Target
Data Loss Prevention	Zero data loss on crash (auto-save + backup)
Crash Recovery	Offer backup restore on next launch
File Corruption Detection	100% detection via schema validation
Undo Reliability	All commands reversible

6.3 Usability

Requirement	Standard
Keyboard Accessibility	All primary actions have shortcuts
Error Messages	Specific, actionable, non-technical
Loading States	Visual indicator for operations > 100ms
Responsiveness	UI responds within 100ms of input

6.4 Security (Phase 1 - Local Only)

Requirement	Implementation
Input Validation	Zod schemas on all inputs
XSS Prevention	DOMPurify for rendered text
File System Access	Scoped to project folder only
No Telemetry	No data leaves device without consent

6.5 Error Handling

Scenario	Behavior
Invalid Input	Inline validation message, prevent save
File Load Failure	Show error, offer backup recovery
Calculation Error	Display "—", show warning icon
Unhandled Exception Error	boundary catches, offers reload
Save Failure	Retry 3x with backoff, then alert user

7. Dependencies

7.1 External Libraries

Library	Purpose	License	Notes
Tauri	Desktop runtime	MIT	Core desktop wrapper
Next.js	React framework	MIT	Already in package.json
React	UI library	MIT	Already in package.json
Zustand	State management	MIT	Already in package.json
Zod	Schema validation	MIT	Already in package.json
nanoid	ID generation	MIT	Already in package.json
date-fns	Date formatting	MIT	Needed for timestamps
jspdf	PDF generation	MIT	Needed for PDF export
papaparse	CSV generation	MIT	Needed for CSV export

7.2 Standards Data (ASHRAE/SMACNA)

Required Lookup Tables:

Table	Source	Format	Purpose
Outdoor Air Rates	ASHRAE 62.1, Table 6-1	JSON	Rp and Ra values by oc
Default Occupant Density	ASHRAE 62.1, Table 6-1	JSON	People per sq ft
Duct Roughness Factors	SMACNA	JSON	Friction calculation
Fitting Loss Coefficients	ASHRAE Fundamentals	JSON	Pressure drop
Velocity Limits by Application	Industry standard	JSON	Warning thresholds

Example Lookup Data Structure:

```
{      "occupancyTypes": {                 "office": {              "Rp": 5,          "Ra": 0.06,
ity": 0.005,      "defaultACH": 4                            },      "restaurant": {
0.18,      "defaultDensity": 0.015,                              "defaultACH": 10            }
```

7.3 Assets Required

Asset	Type	Source
Application icon	SVG/PNG	Design (TBD)
Equipment icons	SVG	Design (TBD)
Tool icons	SVG	Design or icon library
Loading spinner	SVG/CSS	Standard
Error/warning icons	SVG	Icon library

7.4 Development Dependencies

Tool	Purpose	Notes
TypeScript	Type safety	Already configured
Vitest	Unit testing	Already configured
Playwright	E2E testing	Needs setup
ESLint	Linting	Needs configuration
Prettier	Formatting	Needs configuration
Husky	Git hooks	Needs setup

8. Out of Scope (Phase 2+)

8.1 Explicitly Excluded Features

The following features are NOT included in Phase 1 MVP:

Heat & Thermal Calculations

- Heat load calculations (BTU/hr)
- Cooling load calculations (tonnage)
- Psychrometric calculations (humidity, wet bulb)
- Coil selection and sizing
- Temperature-based calculations
- Weather data integration

Advanced HVAC

- Multiple zone analysis
- Energy modeling
- Equipment scheduling
- Commissioning checklists

Collaboration Features

- Cloud sync
- User authentication
- Real-time collaboration
- Team workspaces
- Version control (beyond undo/redo)
- Comments and annotations

Advanced Export

- DXF/DWG CAD export
- Revit integration
- 3D visualization
- Animated airflow simulation

Enterprise Features

- Organization management
- Role-based access control
- Audit logging
- SSO integration
- API access

8.2 Deferred Specifications (DRAFT Status)

The following specifications are marked DRAFT and excluded from Phase 1:

Document	Status	Reason

First Launch & Onboarding Flow	DRAFT - Awaiting UX Review	UX decisions pending
Inspector Panel Specification	DRAFT - Awaiting Domain Review	Field list needs validation
Validation Ranges	DRAFT - Awaiting Domain Review	Ranges need HVAC engineer sign-

8.3 Phase 2 Roadmap Preview

For planning purposes, anticipated Phase 2 features include:

1. Cloud Sync: Conflict-free backup and multi-device access
2. Template Library: Pre-built system templates
3. Advanced Calculations: Heat load and cooling
4. DXF Export: CAD-compatible geometry export
5. Material Pricing: BOM with cost estimates

Appendix A: Keyboard Shortcuts Reference

Key	Action	Context
V	Selection tool	Canvas
R	Room tool	Canvas
D	Duct tool	Canvas
E	Equipment tool	Canvas
G	Toggle grid	Canvas
Escape	Cancel / Deselect	Global
Delete	Delete selected	Canvas
Backspace	Delete selected	Canvas
Ctrl+Z	Undo	Global
Ctrl+Y	Redo	Global
Ctrl+Shift+Z	Redo (alternate)	Global
Ctrl+S	Save	Global
Ctrl+D	Duplicate selected	Canvas
Ctrl+G	Group selected	Canvas
Ctrl+Shift+G	Ungroup	Canvas
Ctrl+A	Select all	Canvas
Ctrl+0	Fit to content	Canvas
Home	Reset view	Canvas
Space+Drag	Pan canvas	Canvas
Scroll	Zoom in/out	Canvas

Appendix B: Validation Ranges Summary

Entity	Field	Min	Max	Unit
Room	width	1	10,000	inches
Room	length	1	10,000	inches
Room	height	1	500	inches
Room	airChangesPerHour	1	100	ACH
Duct	diameter	4	60	inches
Duct	width	4	96	inches
Duct	height	4	96	inches
Duct	length	0.1	1,000	feet
Duct	airflow	1	100,000	CFM
Duct	staticPressure	0	20	in.w.g.
Equipment	capacity	1	100,000	CFM
Velocity	residential	600	900	FPM
Velocity	commercial	1,000	1,500	FPM
Velocity	industrial	1,500	2,500	FPM
Velocity	kitchen exhaust	1,500	4,000	FPM

Appendix C: Glossary

Term	Definition
ACH	Air Changes per Hour - how many times the room air is replaced per hour
CFM	Cubic Feet per Minute - airflow rate
FPM	Feet Per Minute - air velocity
in.w.g.	Inches of Water Gauge - pressure measurement
BOM	Bill of Materials - list of all materials and quantities
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
SMACNA	Sheet Metal and Air Conditioning Contractors’ National Association
Rp	People outdoor air rate (CFM/person)
Ra	Area outdoor air rate (CFM/sq ft)
Vbz	Breathing zone outdoor airflow

End of Product Requirements Document