Fabric Management Module Requirements Specification: ZYENTRA Apparel

Document Version: 1.0
Date: November 28, 2025

---

1. Executive Summary

This module is designed to manage the lifecycle of fabric purchases, from initial recording (including quantity and discount options) to production completion (tracking actual vs. expected items). It must be built around robust calculation logic and must support full CRUD operations within a real-time data environment (Firestore).

2. Scope of Work (In-Scope)

The following table outlines the key features included in the scope of this development project.

Feature Category	Description
Data Entry (CREATE)	Comprehensive form to capture all fabric details, quantity (yards/rolls), dis
Core Calculations	Server-authoritative logic for `Expected Items`, `Original Amount`, `Discounte`
Data View (READ)	Real-time, sortable, searchable, and paginated data table displaying all fab
Data Modification (UPDATE)	Full record editing (re-computation required) and a separate quick-update
Data Deletion (DELETE)	Soft delete functionality with confirmation prompt.
Authentication	Secure access control (Auth required for write operations).
Audit	Automatic tracking of `createdBy`, `updatedBy`, `createdAt`, and `updatedAt` for

3. Functional Requirements (FR)

FR 3.1. Fabric Creation (CREATE)

FR 3.1.1 **Data Capture:** The system must accept inputs for `Fabric Name`, `Fabric Height`, `Price per Yard`, and `Apparel Length (inches)`.

FR 3.1.2 **Quantity Input Toggle:** The user must select the input mode: 'Num. of Yards' or 'Num. of Rolls' (Radio buttons).

- If 'Yards' is selected, input for `numYards` is displayed.
- If 'Rolls' is selected, inputs for `numRolls` and `yardsPerRoll` are displayed.

FR 3.1.3 Discount Input Toggle: The user must toggle 'Receive Discount? (Yes/No)'.

- If Yes, the user selects one discount type (Overall, Per Yard, Per Roll) via radio buttons, and the corresponding input field is shown.

FR 3.1.4 Live Calculation Preview: The UI must display a live preview of the calculated amounts *before* submission: Original Amount, Discounted Amount, Final Total Amount, and Expected Items.

FR 3.1.5 Submission: The backend/server is responsible for final validation and computation of all derived fields (`numYards`, `expectedItems`, `originalAmount`, `discountedAmount`, `totalAmount`) before saving the record.

FR 3.2. Data Display and Search (READ)

FR 3.2.1 Table View: All fabric records must be displayed in a paginated, sortable, and searchable table.

FR 3.2.2 Required Columns: Columns must include: Fabric Name, Price/Yard, Quantity (Yards or Rolls, with mode indication), Yards Total, Apparel Length, Expected Items, Actual Produced Items, Original Amount, Discounted Amount, Total Amount, and Action Buttons.

FR 3.2.3 Filtering and Sorting: Users must be able to Search by `Fabric Name`, Sort by `Price per Yard`, `Date Created`, and `Fabric Name`, and Filter by Date Range and Discount Type.

FR 3.2.4 Row Actions: Each row must contain action buttons: `View`, `Edit`, `Update Actual`, and `Delete`.

FR 3.2.5 Status Highlighting: Rows where `Actual Produced Items` $<$ `Expected Items` must be visually highlighted.

FR 3.3. Record Modification (UPDATE)

FR 3.3.1 Full Edit: Modifications to any input field must trigger re-computation of all derived fields.

FR 3.3.2 Actual Production Update: A dedicated modal must be used for updating `Actual Produced Items`. Upon submission, the system displays Expected Items, Actual Items, Difference (`Actual - Expected`), and Efficiency Percentage (`(Actual / Expected) * 100`).

FR 3.4. Record Deletion (DELETE)

FR 3.4.1 Confirmation: Deleting a record must trigger a confirmation modal (not a browser alert).

FR 3.4.2 Soft Delete: The record must be soft-deleted by setting a boolean flag (`isDeleted: true`).

4. Business Logic & Calculation Rules

All currency values must be stored with two decimals (`round2(x)`).

4.1. Total Yards Determination

- Conversion: 1 yard = 36 inches.
- Logic: If `inputMode` is 'roll', `numYards` = `numRolls` $\times$ `yardsPerRoll`. Otherwise, `numYards` is taken as entered.

4.2. Expected Items Formula

The result must be an integer (floor).

$$\text{Expected Items} = \lfloor \frac{\text{Number of Yards} \times 36}{\text{Apparel Length (inches)}} \rfloor$$

4.3. Amount Calculations

Calculation	Logic
Original Amount	`originalAmount = round2(pricePerYard * numYards)`
Discount (None)	`discountedAmount = 0; totalAmount = originalAmount`
Discount (Overall)	`discountedAmount = overallDiscountAmount; totalAmount = max(0, originalAmo`
Discount (Per Yard)	`discountedAmount = round2(discountedPricePerYard * numYards); totalAmount`
Discount (Per Roll)	`effectiveNumRolls = inputMode === 'roll' ? numRolls : ceil(numYards / yar`
	`discountedAmount = round2(discountedPricePerRoll * effectiveNumRolls); tot`

5. Non-Functional Requirements (NFR)

- **NFR 5.1. Security:** All write operations require user authentication.
- **NFR 5.2. Data Integrity:** Validation must prevent negative numbers. `overallDiscountAmount` must not exceed `originalAmount` (warn user, set final amount to 0).
- **NFR 5.3. Performance:** The data table must load records efficiently with pagination and mandatory real-time updates via Firestore listeners.
- **NFR 5.4. Technology:** MERN Stack with Firebase Firestore as the data store.

6. Data Model (Firestore Document Structure)

The fabric document will be stored in Firestore: `/artifacts/{__app_id}/users/{userId}/fabric_inventory`

Field Name	Data Type	Source	Description
Fabric Details (User Input)			
`fabricName`	String	User Input	E.g., Cotton Blue
`pricePerYard`	Number	User Input	Cost of fabric per yard
`apparelLengthInches`	Number	User Input	Length required per item
Quantity Inputs			
`inputMode`	String	User Input	('yard' | 'roll')
`numYardsInput`	Number	Cond. User Input	Raw yards input (if mode is 'yard')
`numRollsInput`	Number	Cond. User Input	Raw rolls input (if mode is 'roll')
Discount Inputs			
`receiveDiscount`	Boolean	User Input	True if a discount was applied
`discountType`	String	Server Computed	('none' | 'overall' | 'perYard' | 'perRoll')
`overallDiscountAmount`	Number	Cond. User Input	Raw overall discount value
Computed Fields (Server Authoritative)			
`numYards`	Number	Server Computed	Total computed yards
`originalAmount`	Number	Server Computed	`pricePerYard` $\times$ `numYards` (2 decimals)
`totalAmount`	Number	Server Computed	Final payable amount (2 decimals)
`expectedItems`	Number	Server Computed	Total items expected (integer, floor)
Production Tracking & Audit			
`actualProducedItems`	Number	User Update	Items actually manufactured
`isDeleted`	Boolean	System	Soft delete flag
`createdAt`	Timestamp	System	Creation timestamp
`createdBy`	String	System	User ID who created the record