DIGITAL HAUTE

App Specification for Rork

A fashion wholesale buying management web app. Users register, capture products, track budgets and deliveries, collaborate with teams, and integrate with Shopify.

Version: 1.0 | **Date:** February 04, 2026

1. APP OVERVIEW

App Name: Digital Haute

Platform: Responsive Web App (PWA) — works on desktop, tablet, and mobile browsers. No app store needed.

Core Purpose: Helps fashion boutique buyers organize wholesale purchases, track budgets/deliveries, and collaborate with their team — all in one place.

User Roles

Role	Can Do
Owner	Full access: manage products, vendors, budgets, team, billing, integrations
Buyer	Add/edit products, vendors, budgets. Export data. Cannot manage team or billing.
Viewer	Read-only access to all data. Can export reports. Cannot edit anything.

2. SCREENS & FEATURES

2.1 Registration & Auth

- Email/password signup with email verification. Fields: email, password, first name, last name, business name.
- Google and Apple OAuth sign-in (P1).
- Password reset via email link.
- 14-day free trial, no credit card required.
- JWT auth: 15-min access token + 7-day refresh token (HTTP-only cookie).

2.2 Onboarding Wizard (4 steps)

- Step 1: Confirm business profile (name, logo).
- Step 2: Create first season (e.g., 'Fall 2026').
- Step 3: Set up categories (Tops, Bottoms, Dresses, Accessories, etc.).
- Step 4: Invite team members (optional, skippable).

2.3 Dashboard

- **Quick Stats Bar:** Total products on order, total vendors, next delivery date, budget utilization %.
- **Budget Summary Cards:** Total budget / spent / remaining per season or category with progress bars.
- **Product Grid:** Masonry-style image grid showing product photo, vendor name, category badge, delivery month. Filterable + searchable.
- **Delivery Calendar:** Monthly calendar with colored dots per vendor. Tap a date to see deliveries.
- **Activity Feed (P1):** Recent actions timeline.

2.4 Product Management

Feature	Details	P
Add Product	Upload photos (camera or file picker). Form: name, style #, vendor, category, subcategory, wholesale cost, retail price, qty, colors, sizes, delivery date, season, collection, notes, status.	P0
Tag Scanner (OCR)	Camera-based: photograph hang tag → OCR auto-fills style #, price, color, size.	P1

Feature	Details	P
Search & Filter	Full-text search. Filter by vendor, category, season, delivery month, color, status, price range.	P0
Status Workflow	Maybe → Ordered → Shipped → Delivered → Received. Also: any → Cancelled.	P0
Bulk Edit	Select multiple → update shared fields (delivery date, status, category).	P1
Duplicate Detection	Warn when style # or name closely matches existing product.	P2

2.5 Vendor Management

- Vendor directory: list with name, contact info, total spend, product count, last order date.
- Vendor detail page: all products from that vendor + spend summary + delivery history.
- Add/edit vendor: name, rep name, email, phone, website, payment terms, notes.

2.6 Budget & Financial Tracking

- Create budgets scoped to season + optional category or vendor.
- Real-time spend tracking as products are added. Visual progress bars. Alerts at 75%, 90%, 100%.
- Revenue projections: retail price × quantity. Margin calculation (P1).
- Interactive charts: spend by category, vendor, month, season (P1).
- Export reports as CSV, XLSX, or PDF.

2.7 Delivery Tracking

- Visual timeline of all expected deliveries, filterable by vendor/month/season.
- Email/push alerts when delivery is approaching (7 days, 1 day) or overdue (P1).
- Mark products as Received (batch or individual).

2.8 Team Collaboration

- Invite members via email. Assign role: Owner, Buyer, or Viewer.
- Real-time sync: changes by any member appear instantly (WebSocket).
- Activity log / audit trail of all actions (P1).
- Generate shareable link or PDF of curated product selections (P1).

2.9 Integrations

- **Shopify Export (P1):** Push selected products to Shopify as unpublished drafts with images, pricing, variants.
- **Wholesale Import (P1):** Import orders from JOOR, NuOrder, Faire, FashionGo, Hubventory via CSV upload.

2.10 Settings

- Profile: edit name, email, password, business name, logo, timezone.
- Billing: view plan, upgrade/downgrade, payment method, invoice history (Stripe).
- Notifications: toggle email/push/in-app per event type.
- Data export & account deletion (GDPR).

3. DATA MODEL

All tables have: **id** (UUID), **created_at**, **updated_at**, **deleted_at** (soft deletes).

3.1 users

Column	Type	Notes
email	VARCHAR(255)	Unique, lowercased
password_hash	VARCHAR(255)	bcrypt 12 rounds (nullable if OAuth-only)
first_name / last_name	VARCHAR(100)	Required
email_verified	BOOLEAN	Default false
avatar_url	TEXT	S3 URL
timezone	VARCHAR(50)	IANA timezone
organization_id	UUID	FK → organizations
role	ENUM	owner / buyer / viewer

3.2 organizations

Column	Type	Notes
name	VARCHAR(255)	Business name
logo_url	TEXT	Optional
plan	ENUM	free_trial / boutique / growth / enterprise
trial_ends_at	TIMESTAMP	14 days from creation
stripe_customer_id	VARCHAR(100)	Billing
settings	JSONB	Org preferences

3.3 products

Column	Type	Notes
organization_id	UUID	FK → organizations (multi-tenant)
vendor_id	UUID	FK → vendors
name	VARCHAR(255)	Product name
style_number	VARCHAR(100)	For duplicate detection
wholesale_price	DECIMAL(10,2)	Cost price
retail_price	DECIMAL(10,2)	Selling price (nullable)
quantity	INTEGER	Default 1
colors / sizes	TEXT[]	Arrays
category / subcategory	VARCHAR(100)	e.g., Tops, Dresses
season / collection	VARCHAR	e.g., Fall 2026
delivery_date	DATE	Expected delivery
received_date	DATE	Actual received
status	ENUM	maybe / ordered / shipped / delivered / received / cancelled
notes	TEXT	Free text
created_by	UUID	FK → users

3.4 product_images

Column	Type	Notes
product_id	UUID	FK → products
original_url / thumbnail_url / medium_url	TEXT	S3 paths (full / 200px / 600px)
sort_order	INTEGER	Display ordering

3.5 vendors

Column	Type	Notes
organization_id	UUID	FK → organizations
name	VARCHAR(255)	Brand name
contact_name / email / phone	VARCHAR	Sales rep info
website	TEXT	Optional
payment_terms / notes	TEXT	e.g., Net 30

3.6 budgets

Column	Type	Notes
organization_id	UUID	FK → organizations
season	VARCHAR(50)	Required
category	VARCHAR(100)	Optional scope filter
vendor_id	UUID	Optional scope filter
amount	DECIMAL(12,2)	Budget ceiling
spent	DECIMAL(12,2)	Cached computed sum

3.7 Other Tables

- **audit_logs:** user_id, action, resource_type, resource_id, old_values (JSONB), new_values (JSONB), timestamp.
- **notifications:** user_id, type, title, body, action_url, read_at.
- **invitations:** organization_id, email, role, token, status, expires_at.
- **integrations:** organization_id, provider, access_token (encrypted), refresh_token, metadata (JSONB).
- **shared_links:** organization_id, token, filter_config (JSONB), expires_at.

4. KEY API ENDPOINTS

Base URL: **https://api.digitalhaute.com/v1** — JSON responses, Bearer token auth.

Method	Endpoint	Purpose
POST	/auth/register	Create account (returns JWT pair)
POST	/auth/login	Login (returns JWT pair)
POST	/auth/refresh	Refresh access token
POST	/auth/forgot-password	Send password reset email
POST	/auth/oauth/{provider}	Google/Apple OAuth
GET	/products	List products (paginated, filtered, sorted)
POST	/products	Create product
PUT	/products/:id	Update product
DELETE	/products/:id	Soft-delete product
POST	/products/scan-tag	OCR tag scan → parsed fields
GET	/products/search?q=...	Full-text search
POST	/products/:id/images	Upload product images
GET/POST/PUT/DEL	/vendors[/:id]	Vendor CRUD
GET/POST/PUT/DEL	/budgets[/:id]	Budget CRUD
GET	/deliveries	List deliveries (grouped by date/vendor)
POST	/deliveries/:id/receive	Mark as received
POST	/team/invite	Invite team member
GET/PUT/DEL	/team/:userId	Manage team member
GET	/notifications	Notification feed
POST	/integrations/shopify/export	Push products to Shopify
POST	/imports/upload	Upload wholesale CSV for import
POST	/reports/generate	Generate report (async)
POST	/shares	Create shareable link

5. TECH STACK

Layer	Technology
Frontend	React 18 + TypeScript, Tailwind CSS, Vite, PWA service worker
Backend	Node.js (Fastify or Express) or Python (FastAPI)
Database	PostgreSQL 15+ (UUID PKs, soft deletes)
Cache / Queue	Redis 7+ (sessions, rate limiting, budget cache, job queue)
Search	Meilisearch or Elasticsearch
Real-Time	Socket.IO / WebSocket (rooms per organization)
Image Storage	AWS S3 / GCS + CDN (CloudFront / Cloudflare)
Email	SendGrid or AWS SES
Payments	Stripe (subscriptions, invoices, Stripe.js)
OCR	Google Cloud Vision API or Tesseract
CI/CD	GitHub Actions → Docker → Cloud Run / ECS

6. KEY USER FLOWS

6.1 Registration → First Product

Sign up → email verification → onboarding wizard (season, categories, invite team) → dashboard → add first product (upload photo, fill details) → product appears in grid, budget updates.

6.2 Market Day Buying

Open on mobile → tap '+' → photograph item → scan tag (OCR auto-fills) → confirm details, pick vendor/category → save → team sees it instantly → check budget before ordering more → end of day: convert 'Maybe' to 'Ordered' or delete.

6.3 Delivery Receiving

Get delivery alert email → open Deliveries view → check boxes against physical shipment → mark products as Received → marketing team (Viewers) sees updated status.

6.4 Shopify Export

Settings → connect Shopify (OAuth) → select products → 'Push to Shopify' → creates unpublished drafts with images and pricing → owner reviews in Shopify and publishes.

— End of Specification —