StoreFlow — A1.5 Dashboard, Storefront, UX & Accessibility Deep Dive

1. Dashboard Architecture Overview

The StoreFlow merchant dashboard is designed as a modern, fast, modal-driven SPA powered by Vue 3 + Inertia.js. The dashboard has two primary functional zones:

- Operations Zone — Real-time order management, fulfilment workflows, printing.
- Management Zone — Products, shipping config, loyalty rules, customers, audit logs.

Design Goals:

- Zero full-page reloads.
- All CRUD flows in modals.
- Real-time order board for active fulfilment.
- Clear separation between “work mode” and “admin mode.”
- Mobile-responsive, but heavily optimized for desktop usage in retail environments.

2. Dashboard Layout & Navigation Model

Layout Structure:

```
------------------------------------------
| Sidebar (vertical) | Content |
------------------------------------------
```

Sidebar Sections:

- Operations (default landing)
- Orders (history)
- Products
- Customers
- Shipping
- Loyalty
- Store Settings
- Audit Logs

Navigation:

- Built using Inertia with server-side controllers composing props.
- No frontend routing library required.
- Each section is a Vue page component with nested subcomponents.

Rendering Strategy:

- Use Inertia partial reloads to avoid fetching unchanged data.
- Heavy tables use server-driven pagination and search callbacks.

3. Operations Board – UX & Interactions

The Operations Board presents active orders grouped by status.

Columns:

• Pending

• Accepted

• In Progress / Packing

• Ready / Shipped

• Completed (collapsed or hidden by default)

• Cancelled (hidden by default)

Features:

• Real-time updates via WebSockets.

• Drag-and-drop MAY be supported in future versions.

• Clicking an order opens a modal with:

- Items

- Customer details

- Fulfilment type

- Action buttons (Accept, Ready, Ship, Complete)

- Print options

Performance:

• Only active orders are kept in memory.

• Completed orders unloaded after N minutes.

• WebSocket updates patch only changed order entries.

4. Product Management – UX Patterns

Product management requires frequent creation/editing of items with customizations.

UX Components:

• Product table with filters and search.

• Product drawer (slide-over) for editing.

• Customization groups presented as nested accordions.

• Ability to bulk-enable/disable products.

Image Handling:

• Drag-and-drop uploader.

• Accepts JPEG/PNG/WebP.

• Preview rendered instantly.

Validation:

• Client-side validation for quick feedback.

• Server-side validation ensures integrity:

- price rules

- customizations rules (min/max rules)

- shipping attributes (weight/dimensions)

5. Shipping Configuration – Interface Structure

Shipping configuration complexity requires a clean hierarchical interface.

UI Structure:

- Zones tab: List + edit zone ranges (postcode_from/to).
- Methods tab: Add/edit methods with type selector.
- Rates tab: Rates nested under each method.

Visual Aids:

- Color coding for flat/weight/price methods.
- Inline explanation text for complex fields.
- Preview summary of effective shipping rules.

Validation:

- Prevent overlapping postcode ranges if possible.
- Prevent methods without active rates from showing as available.

6. Loyalty Configuration – Merchant UX

Loyalty settings are accessible only to merchant owners and managers.

Interface:

• Points per dollar input.

• Threshold input (points required to redeem).

• Reward builder UI using selectable:

- Fixed discount (cents)

- Percentage discount

• Live preview:

“Customer receives $X off once reaching Y points.”

UX Goals:

• Zero ambiguity for merchants.

• Clear warnings if loyalty system is disabled (points_per_dollar=0).

7. Customer Management – Dashboard View

Customer management includes search, filtering, and order insights.

Information Shown:

- Name, email, mobile
- Total orders
- Loyalty points (if active)
- Last order date

Customer Detail Drawer:

- Past order list
- Loyalty adjustments (future)
- Ability to merge duplicate records (future)

Privacy:

- Sensitive PII masked in most list views (partial email/mobile).

8. Order History – UX and Data Navigation

Order history is a separate view from Operations.

Features:

- Date-range filters
- Fulfilment type filters
- Status filters
- Export to CSV (future)
- Order detail drawer

Performance:

- Paginated server-side.
- Complex queries use composite indexes.
- Avoid rendering customizations inline until drawer opened.

9. Storefront Templates – Overview

StoreFlow storefronts provide three selectable themes:

Themes:

1. Classic – business-like, neutral
2. Modern – bold typography, contemporary UI
3. Minimal – white space heavy, clean

Merchants can choose a theme from store settings.

Switching themes simply swaps one of three CSS bundles.

Design Principles:

- Mobile-first layout
- High readability
- Accessibility compliant
- Low JS usage (except cart/checkout)

10. Storefront Cart UX

Cart UX integrates dynamic updates and real-time shipping quoting.

Cart Features:

• Increment/decrement quantities

• Apply customizations visually

• Inline shipping quote (auto-refresh on postcode change)

• Guest checkout or login prompt

• Price breakdown:

- items total

- shipping

- discounts

- tax

- final total

Failure Handling:

• If shipping cannot be calculated:

- Disable checkout button

- Provide clear error message

11. Storefront Checkout Flow

Checkout Steps:

1. Customer enters contact & shipping details
2. Shipping quote calculated in real-time
3. Customer selects shipping or pickup
4. Payment section (Stripe in future)
5. Order confirmation screen with public_id

Features:

- Minimal friction for guests
- Real-time validation
- Accessibility-friendly input fields
- Optional “Save details for next time” when logged in

12. Order Tracking UX – Customer Side

Order tracking page is bound to the public_id token.

Page Features:

- Large status tracker bar (Pending → Accepted → Ready/Shipped → Completed)
- Tracking code + tracking URL (shipping)
- Real-time updates via customer channel
- Contact information and store details shown

13. Accessibility Standards (WCAG AA Focus)

Accessibility is a core requirement for StoreFlow.

Key Compliance Elements:

• Color contrast meets WCAG AA standards.

• All actionable UI elements labelled with ARIA attributes.

• Forms fully keyboard-navigable.

• Error states described both visually and audibly (via screen readers).

• Modals:

- trap focus

- restore focus on close

- ESC key support

Semantic Structure:

• Use , , , , for clear structure.

• Use for all inputs.

14. Mobile UX Considerations

Merchant Dashboard:

• While desktop-first, layout adapts for tablet usage.

• Operations board collapses columns into dropdown filter on smaller screens.

Storefront:

• Full mobile optimization:

- Large tap targets

- Vertical product list flow

- Sticky cart button

- Fast navigation

Checkout:

• Multi-step form broken into vertically stacked segments for mobile.

15. Printing & Thermal Printer Support

Thermal printing is supported for fulfilment workflows.

Print Options:

- 80mm receipt format (ESC/POS-friendly layouts)
- A4 invoice layout (browser PDF print)
- Future: Direct thermal printer API support

Receipt Format:

- Order number (public_id)
- Timestamp
- Items + customizations
- Subtotals, tax, shipping
- Pickup or shipping instructions

Technical:

- Browser print stylesheets hide UI elements.
- CSS page-break rules ensure clean output.

16. Performance & Optimization Techniques

Dashboard Performance:

- Real-time updates eliminate polling overhead.
- Inertia partial reloads minimize data transfer.
- Large tables rendered with virtual scrolling (future upgrade).

Storefront Performance:

- CSS-only theme switching.
- No heavy frameworks; Vue used only where needed.
- Lazy-load product images.

Accessibility & Performance Combined:

- Avoid animations that hinder screen readers.
- Ensure consistent focus handling for modals.