# Flux

## Scope & Release Plan

Baseline Release: Phase 0 (Walking Skeleton) + Phase 1 (Trimmed MVP)

Version: v1.0 (Baseline Release)

Date: 31 January 2026

This document defines what Flux will deliver in Phase 0 and Phase 1, what is explicitly out of scope for the baseline release, and what is planned for later phases.

# Document control

| Document | Flux - Scope & Release Plan |
|---|---|
| Baseline | Phase 0 (Walking Skeleton) + Phase 1 (Trimmed MVP) |
| Owner | Product + Engineering (Flux) |
| Primary use | Release scope agreement, sprint planning, and stakeholder alignment |
| Inputs (source files) | Project Charter; MVP Definition; Baseline SRS; Requirements Catalogue; Core ERD; Latest S |

## How to read this document

Phase 0 is an internal milestone that proves the full offline-first pipe works end-to-end. Phase 1 is the first sellable release (Trimmed MVP). Anything listed as “Out of scope” is intentionally deferred to keep Phase 1 achievable and stable.

## Contents

# 1. Release strategy and principles

**Goal:** deliver a reliable offline-first POS + inventory system in small, safe steps. Phase 0 proves the technical foundation. Phase 1 delivers the minimum set of day-to-day retail flows.

## Key principles

- **Offline-first:** every critical transaction can be created offline and synced later.
- **Branch isolation:** data is scoped by *location_id* and enforced with Supabase RLS.
- **Small MVP:** only the features needed to run a real shop are in Phase 1; accounting-heavy and “nice-to-have” features are pushed out.
- **Single source of truth:** once a record reaches Supabase, it is considered synced; “sync flags” live only on the client (IndexedDB).
- **Testable releases:** each phase has clear exit criteria based on the Charter’s MVP Definition of Done and the Requirements Catalogue (FR/NFR).

## Release milestones

- **Phase 0 (Walking Skeleton):** architecture proof, minimal vertical slice (internal milestone).
- **Phase 1 (Trimmed MVP / Release 1):** first usable release for a single shop or small chain.
- **Phase 2 (Post-MVP):** operational enhancements (printing, cycle counts, advanced flows).
- **Phase 3:** financial and accounting expansion.
- **Phase 4:** advanced analytics and full reporting pack.

## 2. Scope summary

This table gives a quick view of what is included in each phase. Details are in later sections.

| Capability area | Phase 0 | Phase 1 (Release 1) | Later phases |
|---|---|---|---|
| PWA install + offline shell | IN | IN | IN |
| Offline queue + auto sync | IN (sales only) | IN (sales + stock docs) | IN + enhancements |
| Auth + RLS branch isolation | IN (basic) | IN (roles) | IN (enhanced) |
| Master data | IN (min) | IN (core) | IN (expanded) |
| Inventory (stock on hand) | IN (basic) | IN (with batches/expiry) | IN (cycle counts etc) |
| Sales/POS | IN (1 item) | IN (usable POS) | IN (advanced flows) |
| Purchasing | OUT | IN (PO+GRN min) | IN (returns/adjustments) |
| Stock Transfers | OUT | IN (min 2-step) | IN (advanced logistics) |
| Reporting | IN (sales list) | IN (basic ops) | IN (advanced + financial) |
| Accounting/Financials | OUT | OUT | IN (Phase 3+) |
| Barcode printing | OUT | OUT | IN (Phase 2+) |

Note: “IN” means the capability is included as part of the deliverable for that phase. “IN (minimum)” or “IN (basic)” means the simplest usable version only. Anything not listed is assumed OUT for the baseline release.

# 3. Phase 0 scope - Walking Skeleton

Phase 0 is a proof milestone. The purpose is to validate the full stack (Next.js PWA, IndexedDB offline storage, Supabase RLS, and sync) with the smallest vertical slice.

## 3.1 In scope (Phase 0)

- Installable PWA shell with service-worker caching so the app opens without internet.
- Supabase project setup with Row Level Security (RLS) enabled and a single branch/location for initial testing.
- Login using Supabase Auth (one user is enough for Phase 0).
- CRUD for minimal master data: Category, Unit, Item (a “dummy item” is enough to prove the flow).
- Minimal POS: add the item to a cart, take a cash payment, create a sale.
- Offline-first sale: store the invoice in IndexedDB with a “pending sync” state, then sync to Supabase when online.
- Auto-update stock on hand after sync and show a basic sales list/report to confirm end-to-end flow.

## 3.2 Exit criteria (Phase 0)

Phase 0 is “done” when all items below are demonstrated on at least one device:

- App installs as a PWA and opens to the main screen even with no internet.
- User can log in and is restricted to their branch data (RLS is proven).
- User can create minimal master data (category/unit/item).
- User can create a sale while offline; sale is queued locally.
- When internet returns, the sale syncs to Supabase and appears in a sales list.
- Stock on hand updates correctly after sync (no duplicate balance rows).

## 3.3 Out of scope (Phase 0)

- Multi-user roles and permissions beyond a single login.
- Purchasing (PO/GRN), stock transfers, batch/expiry tracking, credit customers.
- All advanced reports, printing, accounting, and payment modules.

# 4. Phase 1 scope - Trimmed MVP (Release 1)

Phase 1 is the first usable release. It is aligned to the “In MVP” column of the MVP Definition and the Baseline SRS. The focus is day-to-day retail operations with reliable offline behaviour.

## 4.1 In scope (Phase 1)

- PWA + offline framework upgrades: durable queue for multiple document types; auto-sync; visible “pending sync” counter.
- User roles (super-admin, manager, cashier) with RLS policies and role-based UI access.
- Core masters CRUD: Branch/Location, Category, Unit, Item (including batch/expiry fields), Supplier, Customer (credit limit/days).
- Inventory control (minimum): stock on hand per location; stock adjustments (increase/decrease) with a reason note; batch/expiry tracking.
- POS/Sales (offline-capable): barcode scan (keyboard wedge), search, quantity edit, simple discounts, payments (cash/card/credit), receipt print.
- Returns (minimal): void before posting + simple return/refund against an invoice (no full accounting credit note yet).
- Purchasing (minimum): Purchase Orders + GRN; posting a GRN updates stock; draft edit allowed; no posted cancellations.
- Multi-location transfers (minimum): transfer-out + transfer-in receipt (two-step) and correct stock balances per location.
- Basic reporting (trimmed): sales summary; stock balance + valuation; expiry warning list; outstanding credit summary.

## 4.2 What “minimum” means in Phase 1

Phase 1 intentionally avoids complex finance and heavy correction workflows. The rules below keep the system stable:

- **Draft vs Posted:** documents can be edited while in draft; after posting, the baseline release avoids cancellation and post-adjustments.
- **Returns:** only simple return/refund against an invoice (no accounting-grade credit note workflow).
- **Offline conflicts:** baseline uses a simple “last-write-wins” approach; deeper conflict handling is planned later.
- **Printing:** receipt printing is allowed (browser print). Barcode/label printing is deferred.

## 4.3 Screens included in Phase 1 (mapped from Full Scope)

Phase 1 uses a subset of the full menu/screen list. The following screen groups are included:

- **Master data:** Category (1.1), Supplier (1.2), Item (1.4), Location (1.6), Customer (1.7), Units (1.8).
- **Transactions:** Sales Invoice (2.1.1); Return/Refund (simplified subset); Purchase Order (2.4.5); GRN (2.2.1); Stock Transfer Note + Stock Receive Note (2.4.1, 2.4.2); Stock Adjustments (2.4.3).
- **Reports:** Sales summary, Stock balance/valuation, Expiry warnings, Outstanding credit summary.

### 4.4 Phase 1 exit criteria (Release 1)

Phase 1 is ready to ship when the following are met (UAT-ready):

- All MUST requirements in the Requirements Catalogue for Phase 1 are implemented and pass acceptance tests.
- Offline creation + later sync works for: sales invoice, stock adjustment, PO/GRN, stock transfer (out + in).
- RLS policies prevent cross-branch data access for all in-scope tables.
- Basic reports load with correct totals for at least one realistic sample dataset (items, batches, sales, GRNs).
- No data duplication on sync retries (uniqueness constraints and idempotent sync are verified).
- PWA installability and baseline performance goals are met (Lighthouse “green range” target from the Charter).

## 5. Out-of-scope for the baseline release (Phase 0 + Phase 1)

The items below are intentionally excluded to keep the baseline release small, buildable, and stable. Most of these appear in the Full Scope document and/or the Charter as longer-term goals, but they are deferred by the Trimmed MVP filter.

### 5.1 Explicit exclusions (deferred to Phase 2+)

- Invoice post-adjustments, temp invoices, formal credit notes / credit invoices (accounting-grade documents).
- GRN post-adjustments, posted GRN cancellation, and a full purchase return module.
- Customer receipts and supplier payments (and their adjustments/cancellations).
- Cheque workflows, bank/branch masters, ledger accounts, journals, and financial statements (full accounting).
- Inventory reconciliation/cycle counts, advanced adjustments, serial number tracking.
- Barcode/label printing (scanning is in; printing is pushed to Phase 2).
- Advanced promotions/loyalty/complex pricing engine.
- Manufacturing/production/BOM and advanced analytics dashboards beyond the basic operational reports.

### 5.2 Full-scope modules that are NOT included in Phase 1

From the full scope menu, the following transaction families are deferred:

- **Sales:** Credit Note (2.1.2), Invoice Post Adjustment (2.1.3), Invoice Cancellation (2.1.4), Credit Invoice (2.1.5), Temp Invoice (2.1.6).
- **Purchases:** Purchase Return Note (2.2.2), GRN Post Adjustment (2.2.3), GRN Cancellation (2.2.4).
- **Payments:** Customer Receipt / Supplier Payment and all adjustments/cancellations (2.3.x).
- **Stock:** Inventory Reconciliation (2.4.4), Barcode Printing (2.4.6).
- **Cheques and related finance modules:** Cheque Deposit / Return / Fixed Deposit (2.5.x and related).
- **Accounting masters:** Banks and Branches, Ledger Accounts and related financial masters.

# 6. What comes later (high-level roadmap)

This section summarises later phases so stakeholders understand what is planned, without expanding the baseline scope.

## 6.1 Phase 2 - Operational Enhancements (Post-MVP)

- Barcode and label printing; manufacturer master and extra masters (banks/ledger accounts).
- Inventory reconciliation/cycle counts, advanced adjustments (reasons), serialised assets tracking.
- More advanced sales flows (temporary invoices, post-adjustments, formal credit notes).
- Purchase enhancements (GRN post-adjustments/cancellation, purchase returns) and payment modules (customer receipts, supplier payments).
- Enhanced user management (MFA, bulk import/export) and background sync notifications.

## 6.2 Phase 3 - Financial & Accounting Expansion

- Cash management (cash in/out, shift/day-end workflows).
- General ledger and journal entries; bank reconciliation and closing periods.
- Deep integration of payments, cheques, and financial reporting.

## 6.3 Phase 4 - Advanced Analytics & Optimisation

- Comprehensive reporting pack (full stock reports, invoice listings, movement/valuation as-at dates).
- Advanced analytics, dashboards, forecasting, and system utilities/polish.

# 7. Deliverables and acceptance gates

Each release increment should update the core architecture docs (Core ERD and Baseline SRS) and add the supporting product documents. The list below follows the project’s “Documentation Required in Each Release” standard.

## 7.1 Required artifacts for the baseline release

- **Core ERD + sync strategy** (incremental updates as schema grows).
- **Baseline SRS** (Phase 0+1 scope, functional + non-functional requirements).
- **Requirements Catalogue** (numbered FR/NFR used for sprint acceptance and UAT).
- **PRD** (users, goals, key features, priorities).
- **Scope & Release Plan** (this document).
- **User Stories + Acceptance Criteria** (backlog; to be produced next).

## 7.2 Acceptance approach

Acceptance is based on: (1) the Charter’s Definition of Done, (2) the Baseline SRS scope boundaries, and (3) the Requirements Catalogue acceptance criteria. A feature is not “done” until it works offline, syncs correctly, and passes RLS checks for the correct branch.

## 8. Assumptions, dependencies, and risks

### 8.1 Assumptions

- Supabase project is provisioned and stable (Auth, Postgres, RLS policies, and required extensions).
- Each device can store offline data in IndexedDB and can run a modern browser (Chrome/Edge/Android WebView).
- Barcode scanning is via a keyboard-wedge scanner or camera-based scanning library (Phase 1 uses the simplest approach).
- Receipt printing uses the browser print flow (hardware-specific integrations are deferred).
- The first release targets small shops; enterprise-grade customisations are deferred.

### 8.2 Key risks and mitigations

- **Offline sync duplicates:** mitigate with UUID keys, uniqueness constraints (e.g., stock balances), and idempotent upserts.
- **Conflicting edits across devices:** baseline uses last-write-wins; later phases can add conflict UI and merge rules.
- **RLS complexity:** keep every transactional table scoped by location_id and test each policy with real roles.
- **Scope creep:** enforce this document as the baseline scope contract; any new feature is planned for Phase 2+ unless it is critical.
- **Performance on low-end devices:** keep UI light, cache masters, and avoid heavy reports in Phase 1.

**End of document.** Next document to produce: **User Stories + Acceptance Criteria** (backlog).