

# Flux

## UX / Screen Documents

### User Flow Diagrams

Phase 0 (Walking Skeleton) + Phase 1 (Trimmed MVP)

Generated: 2026-01-31

Derived from: Project Charter, MVP Definition, Baseline SRS, Requirements Catalogue, Latest Scope, Core ERD, Roadmap, and Release Documentation checklist.

## Contents

1. How to read these flows (lanes, symbols, offline markers)
2. Global rules used in every flow (offline-first, RLS, UUIDs, audit fields)
3. Phase 0 flows (Walking Skeleton)
4. Phase 1 flows (Trimmed MVP)
5. End-to-end example scenario (single shop -> multi-branch)
6. Flow-to-requirements mapping (FR IDs)

## Flow list

- UF-00 (Phase 0): Install and First Login (Online bootstrap)
- UF-01 (Phase 0): Initial Sync and Offline Readiness
- UF-02 (Phase 0): Masters Setup (Category, Unit, Item)
- UF-03 (Phase 0): POS Sale (Cash/Card) - Offline capable
- UF-04 (Phase 1): POS Sale (Credit Customer) - Credit limit check
- UF-05 (Phase 0): Offline Sync Queue, Retry, and Conflict Handling (LWW)
- UF-10 (Phase 1): GRN (Goods Received Note) - Stock-in with Batch/Expiry
- UF-11 (Phase 1): Stock Transfer (2-step) - Transfer Out then Receive
- UF-12 (Phase 1): Batch Selection in POS (Batch-tracked sale line)
- UF-13 (Phase 1): Operational Reports (Sales Summary, Stock Balances, Expiry, Outstanding Credit)

## How to read these user flows

- Each flow is drawn as swimlanes. The lane header tells you who is doing the step (User, App, Server).
- Steps are numbered in order. Arrows show the normal path. When a step moves to a different lane, it means a hand-off (e.g., App -> Server).
- Offline-first idea: the App can create transactions locally (IndexedDB) first, then sync later.
- RLS (Row Level Security): the Server will only accept/read rows for the active location\_id.

## Global rules used in every flow

### Offline-first + sync

- Transactions (sales, GRN, transfers) must work with no internet: create locally, then sync automatically when internet returns.
- Sync queue lives only on the client. The database does not store 'pending-synced' flags.
- Sync is retry-safe: restarting the app must not lose pending records.

### Branch isolation (multi-location)

- Every master and transaction row carries location\_id (except true system-wide configs).
- Server-side RLS enforces that a user can only read/write rows for their allowed location(s).
- Transfers are a 2-step workflow to keep source and destination stock correct.

### Data integrity

- All primary keys are UUIDs (safe for offline generation).
- Barcode uniqueness and stock\_balance uniqueness are enforced in SQL to protect against offline retry duplicates.
- Batch-tracked items require batch and expiry on GRN, and require batch selection in POS.

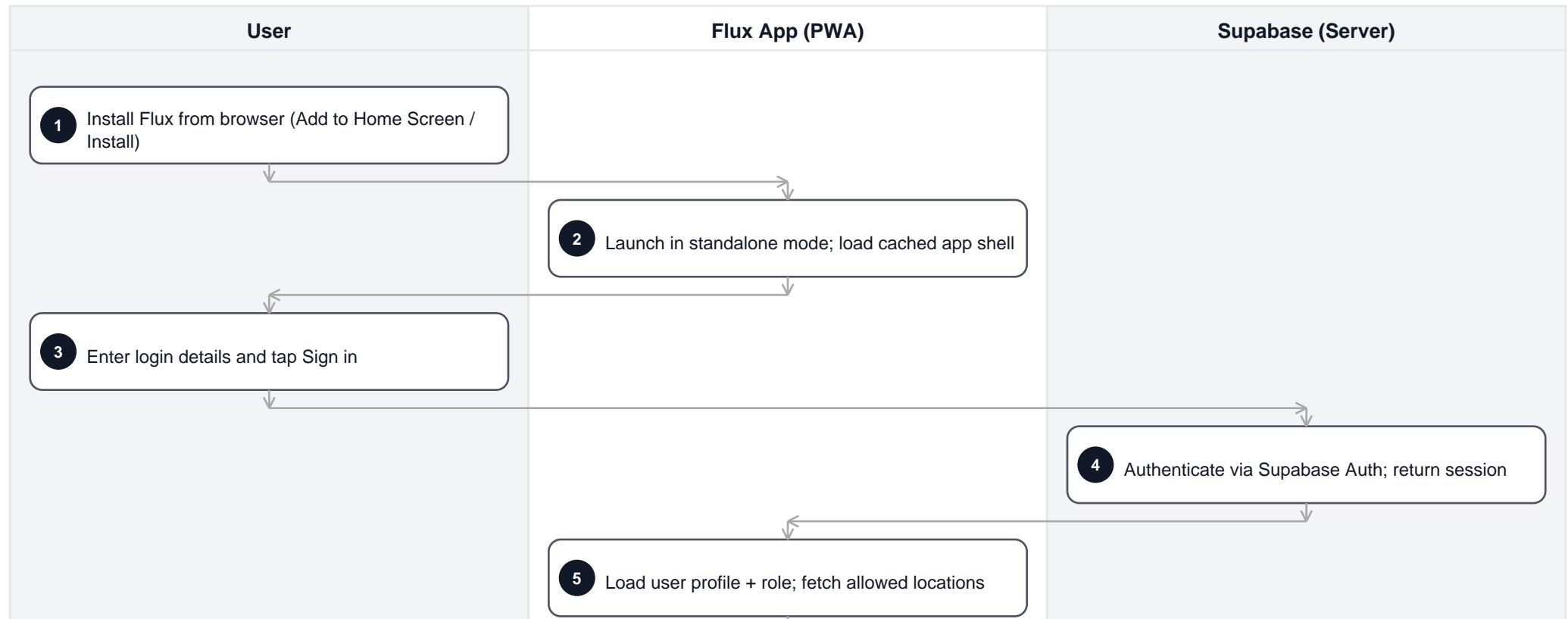
### Reference documents used

- Flux - Project Charter New (PDF)
- MVP Definition for Flux (New) (PDF)
- Flux Baseline SRS (PDF)
- Flux Baseline SRS - Requirements Catalogue (Phase 0 + Phase 1) (PDF)
- Flux Latest Scope (PDF)
- Database Architecture Design (Core ERD + Sync Strategy) (PDF)
- Roadmap for Flux Development (PDF)
- Documentations Required in Each Release - Flux (DOCX)

## UF-00: Install and First Login (Online bootstrap)

Phase: Phase 0 | Primary actor(s): Any user (Cashier / Store Manager / Super Admin)

Key FR IDs: FR-001, FR-002, FR-004, FR-005, FR-006, FR-017

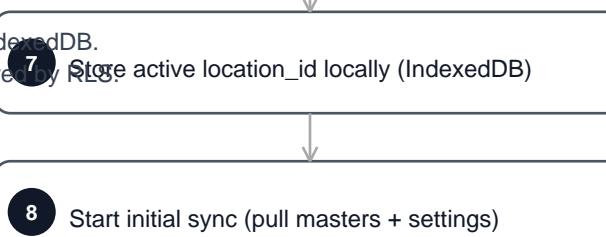


### Key rules

- First login requires internet. After at least one successful login, the app can keep a cached session to allow offline work until the session expires.
- Select active location (branch) if prompted.
- The app shell must load even with no network (service worker cache).

### Data impact (what gets created/updated)

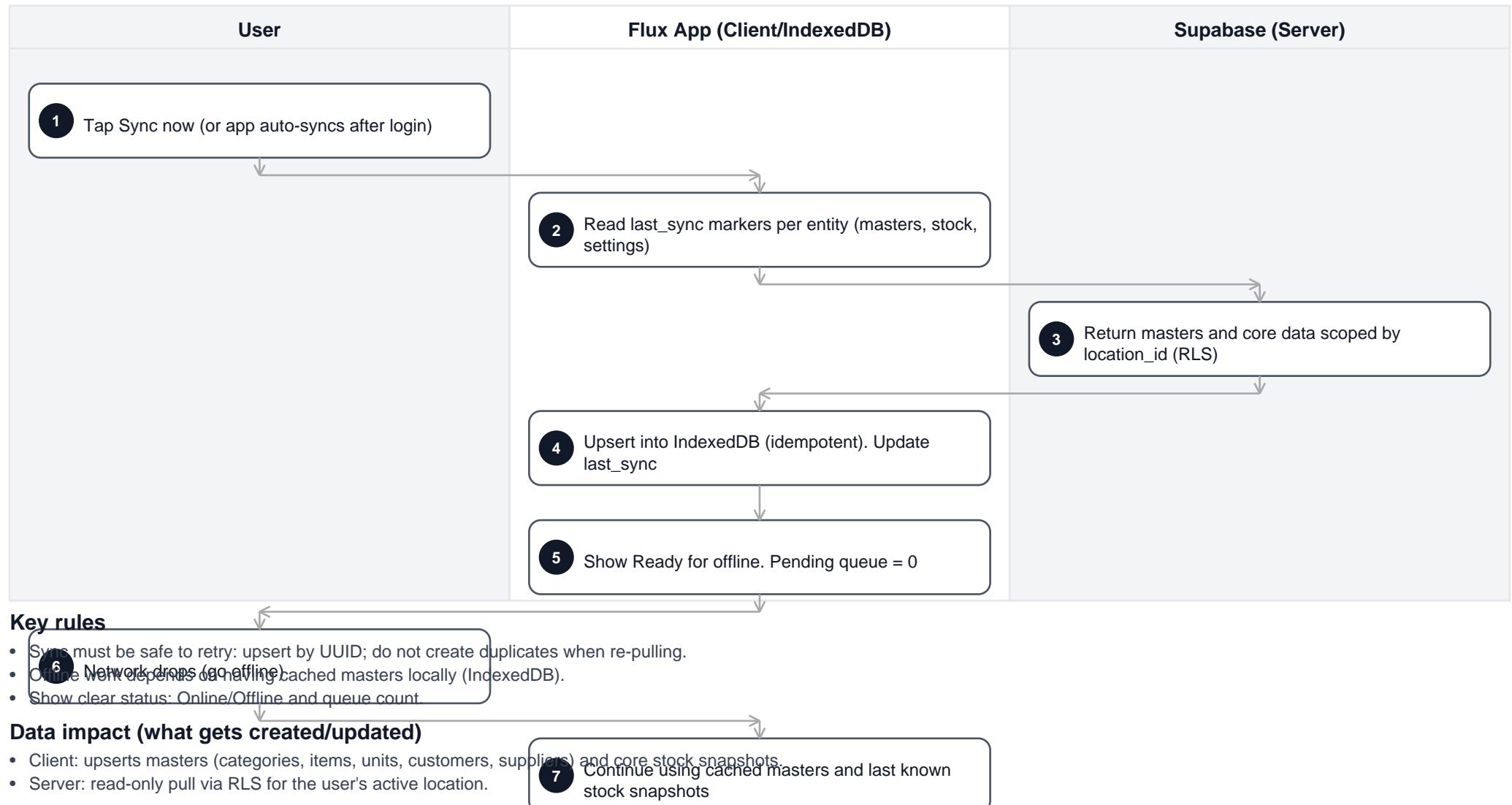
- Client: caches app shell; stores session + active location\_id in IndexedDB.
- Server: issues auth session; all subsequent reads/writes are filtered by RLS.



## UF-01: Initial Sync and Offline Readiness

Phase: Phase 0 | Primary actor(s): Any user after login

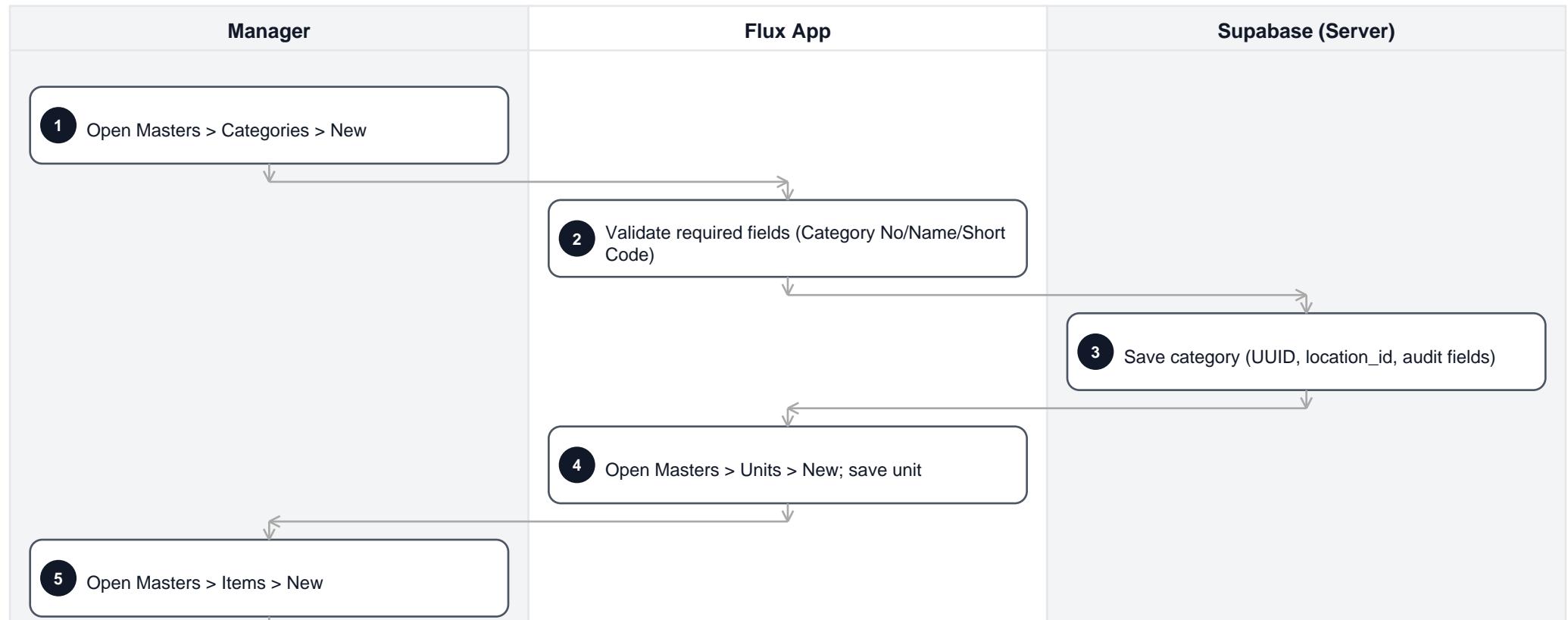
Key FR IDs: FR-011, FR-015, FR-017, FR-006



## UF-02: Masters Setup (Category, Unit, Item)

Phase: Phase 0 | Primary actor(s): Store Manager / Super Admin

Key FR IDs: FR-021, FR-026, FR-028, FR-027, FR-006

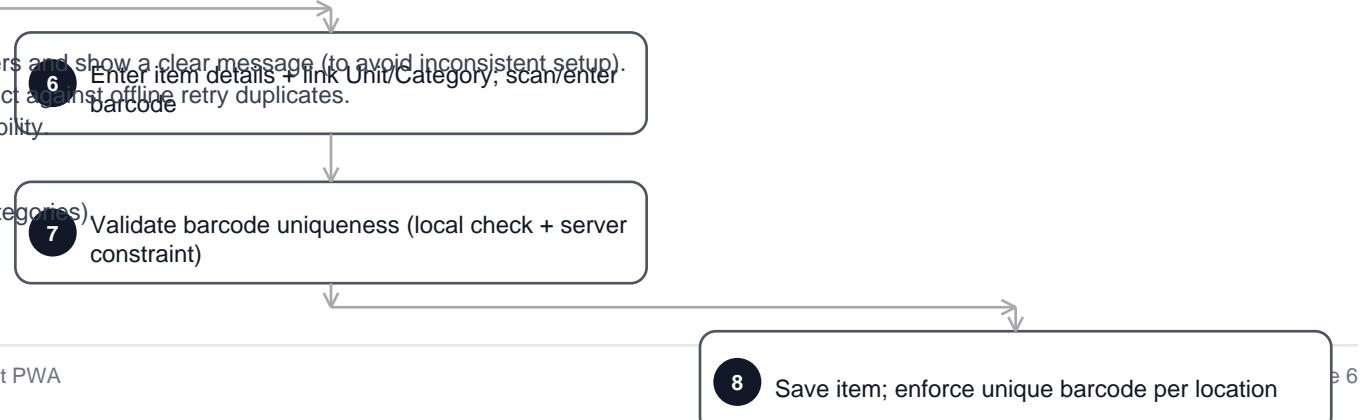


### Key rules

- Masters CRUD is safest online. If offline, allow read-only masters and show a clear message (to avoid inconsistent setup).
- Barcode uniqueness must be enforced in the database to protect against offline retry duplicates.
- All masters carry location\_id for RLS and audit fields for traceability

### Data impact (what gets created/updated)

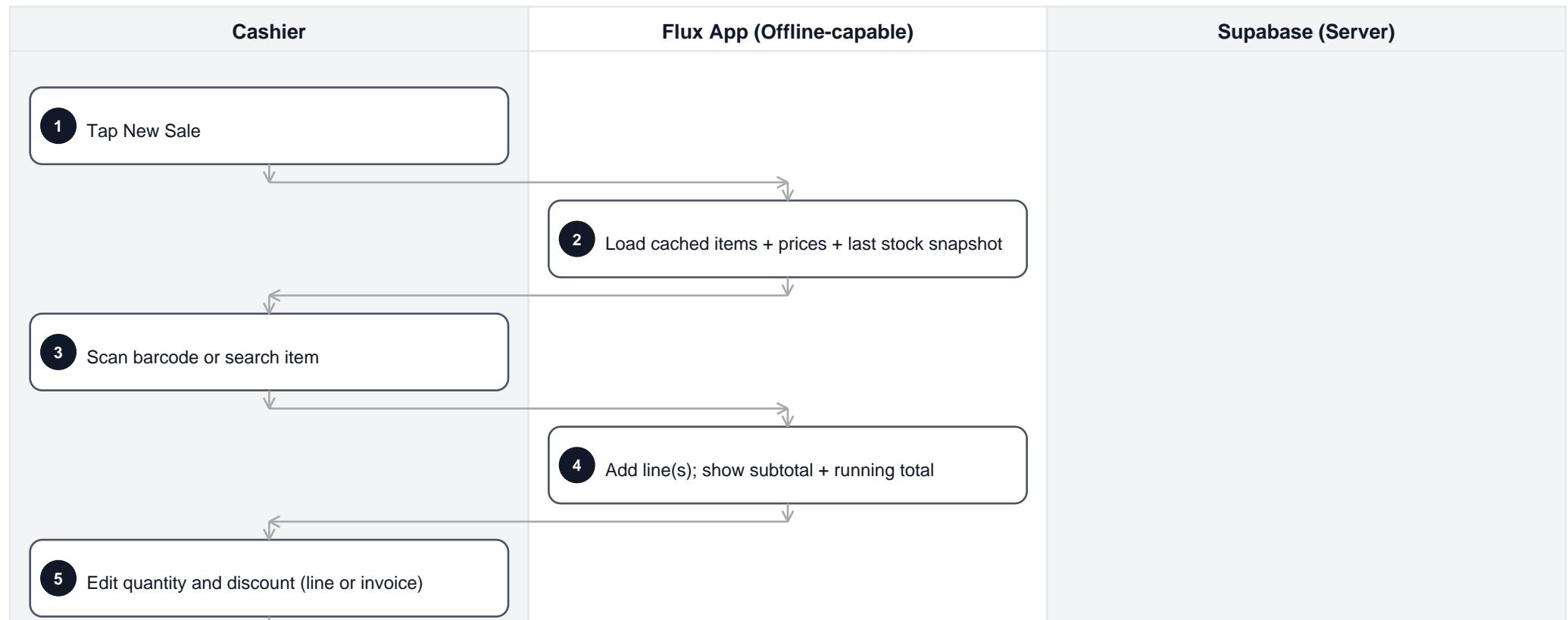
- Tables impacted: categories, units, items (and optional sub\_categories)
- Client: refresh local cache after save (on next sync).



## UF-03: POS Sale (Cash/Card) - Offline capable

Phase: Phase 0 | Primary actor(s): Cashier

Key FR IDs: FR-010, FR-035, FR-036, FR-037, FR-038, FR-040, FR-041, FR-042, FR-018



### Key rules

- Works offline: posting creates a pending sync record in IndexedDB (not in the DB schema).
- On Post, system reduces stock (stock-out) and updates stock balances (after sync success).
- Receipt printing happens after posting (even if offline).

### Data impact (what gets created/updated)

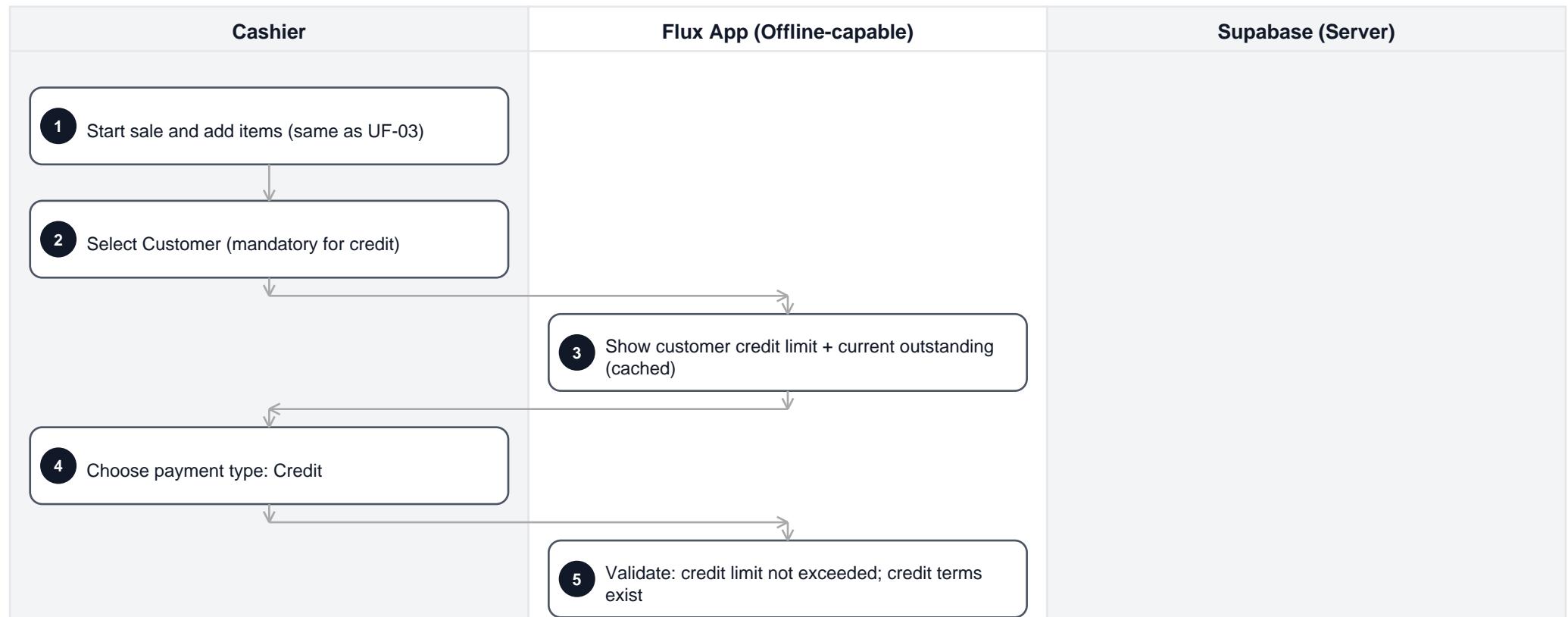
- Tables impacted on sync: sales\_invoices, sales\_invoice\_lines, payments, stock\_moves, stock\_balances.
- Client-only sync queue item with status pending-synced/failed.

8 Validate negative stock policy and required fields

## UF-04: POS Sale (Credit Customer) - Credit limit check

Phase: Phase 1 | Primary actor(s): Cashier

Key FR IDs: FR-024, FR-025, FR-038, FR-039, FR-053, FR-018



### Key rules

- Credit requires Customer selection (hard block).
- Credit limit check uses cached outstanding totals while offline, then validate on sync if needed.
- Outstanding credit is operational (summary report), not full accounting ageing in Phase 0+1

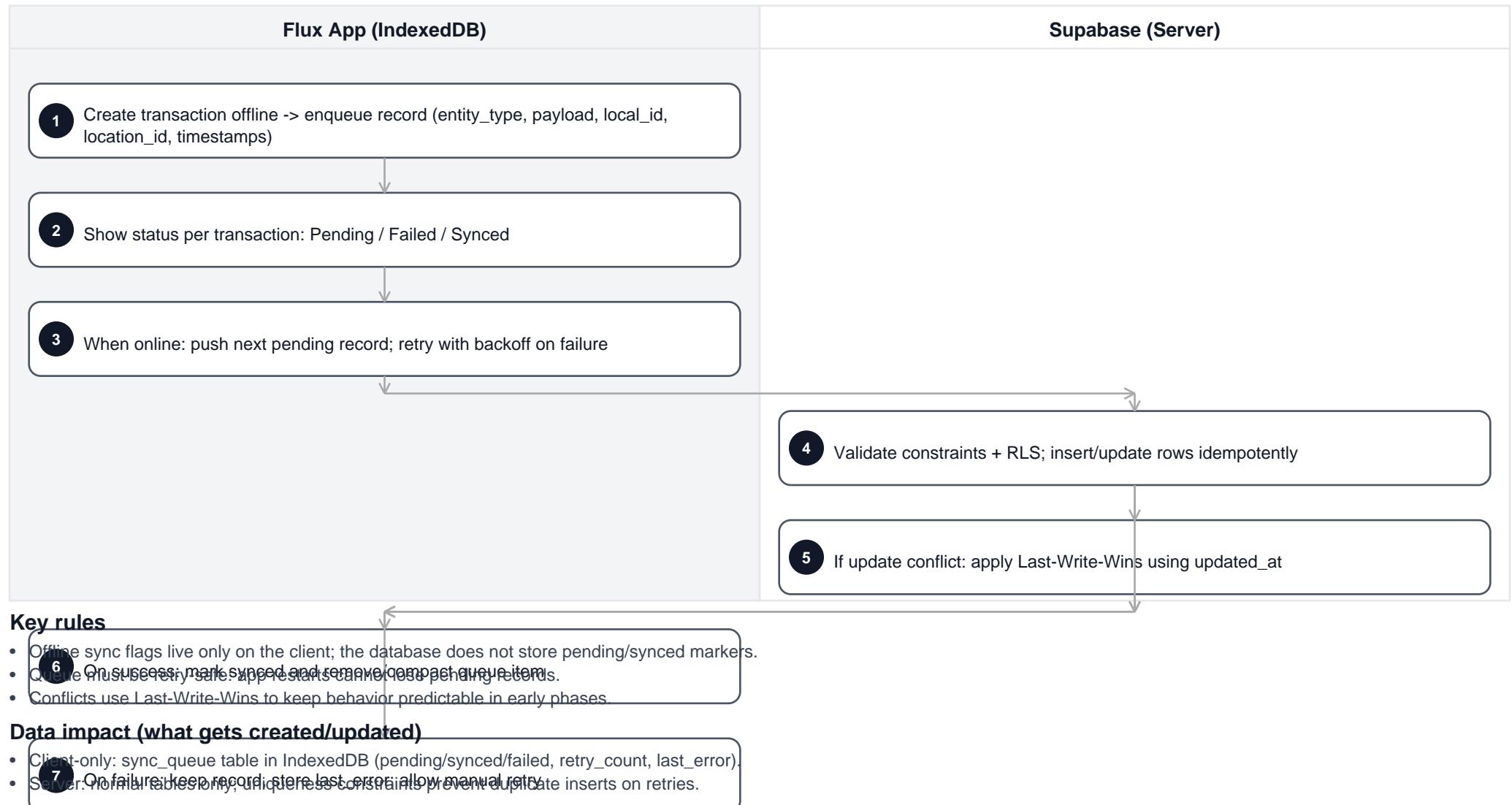
### Data impact (what gets created/updated)

- Tables impacted: sales\_invoices (customer\_id set), sales\_invoice\_lines, payments (credit), stock\_moves.
- Reports: outstanding credit summary reads from credit invoices.

## UF-05: Offline Sync Queue, Retry, and Conflict Handling (LWW)

Phase: Phase 0 | Primary actor(s): System (background) + user visibility

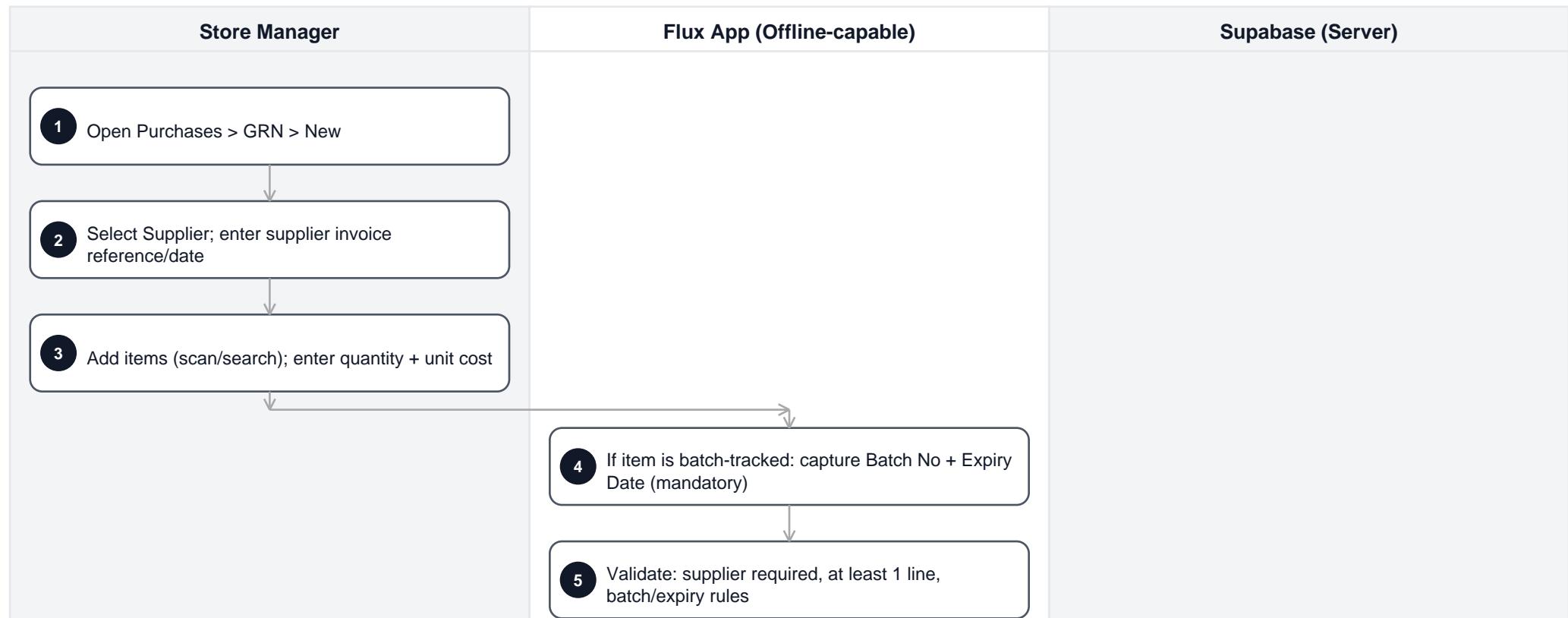
Key FR IDs: FR-012, FR-013, FR-014, FR-015, FR-016, FR-018, FR-019, FR-020



## UF-10: GRN (Goods Received Note) - Stock-in with Batch/Expiry

Phase: Phase 1 | Primary actor(s): Store Manager

Key FR IDs: FR-023, FR-031, FR-032, FR-043, FR-044, FR-045, FR-046



### Key rules

- Batch-tracked items cannot be received without batch and expiry (hard limit)
- Posted GRNs are not editable in Phase 1 (corrections are deferred to later phases)
- Supplier balance update is operational: computed from posted GRNs (no sync AP ledger in Phase 0+1)

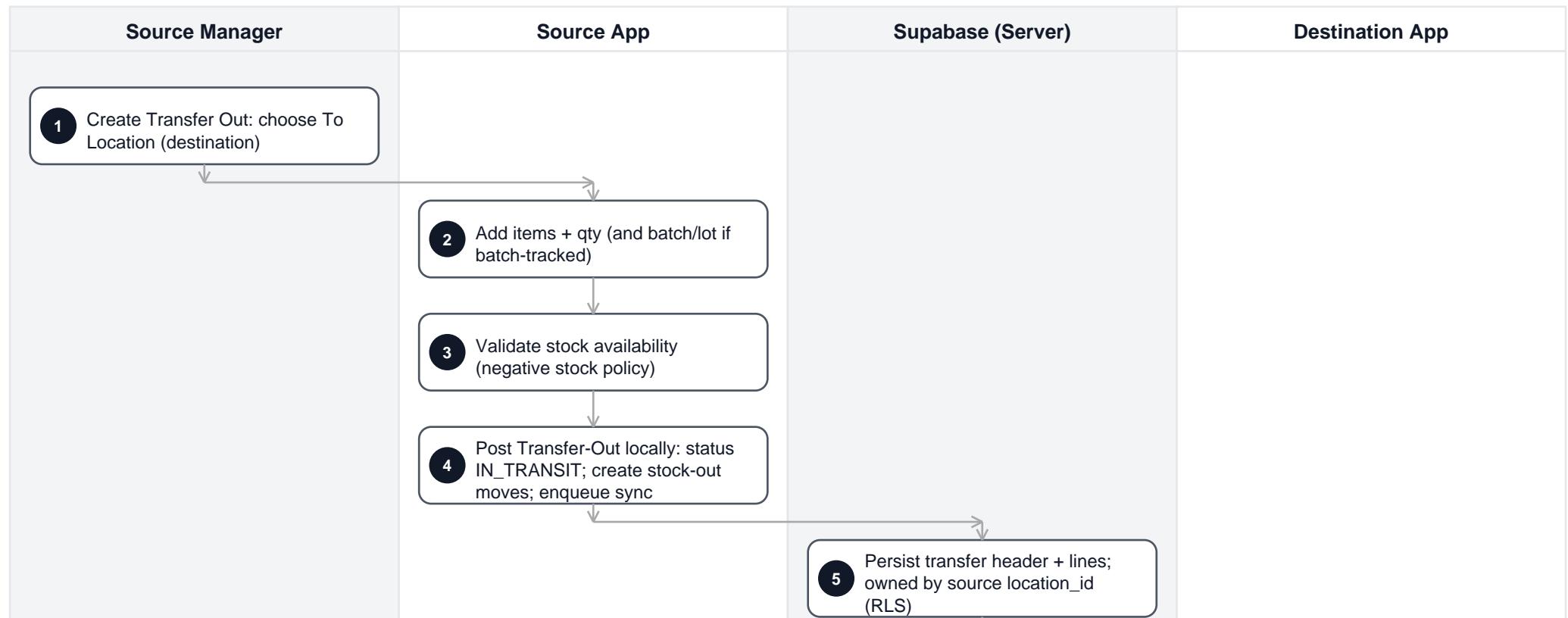
### Data impact (what gets created/updated)

- Tables impacted on sync: grns, grn\_lines, stock\_lots (for batches), stock\_moves, stock\_balances.
- Stock is increased at the receiving location\_id.

## UF-11: Stock Transfer (2-step) - Transfer Out then Receive

Phase: Phase 1 | Primary actor(s): Source Manager + Destination Manager

Key FR IDs: FR-047, FR-048, FR-049, FR-006



### Key rules

- Two-step transfer avoids double counting: stock leaves source on Transfer-Out, enters destination on Receive.
- Transfer lines are batch-aware for batch-tracked items.
- Transfer header carries a location\_id (source) to keep RLS policies clean.

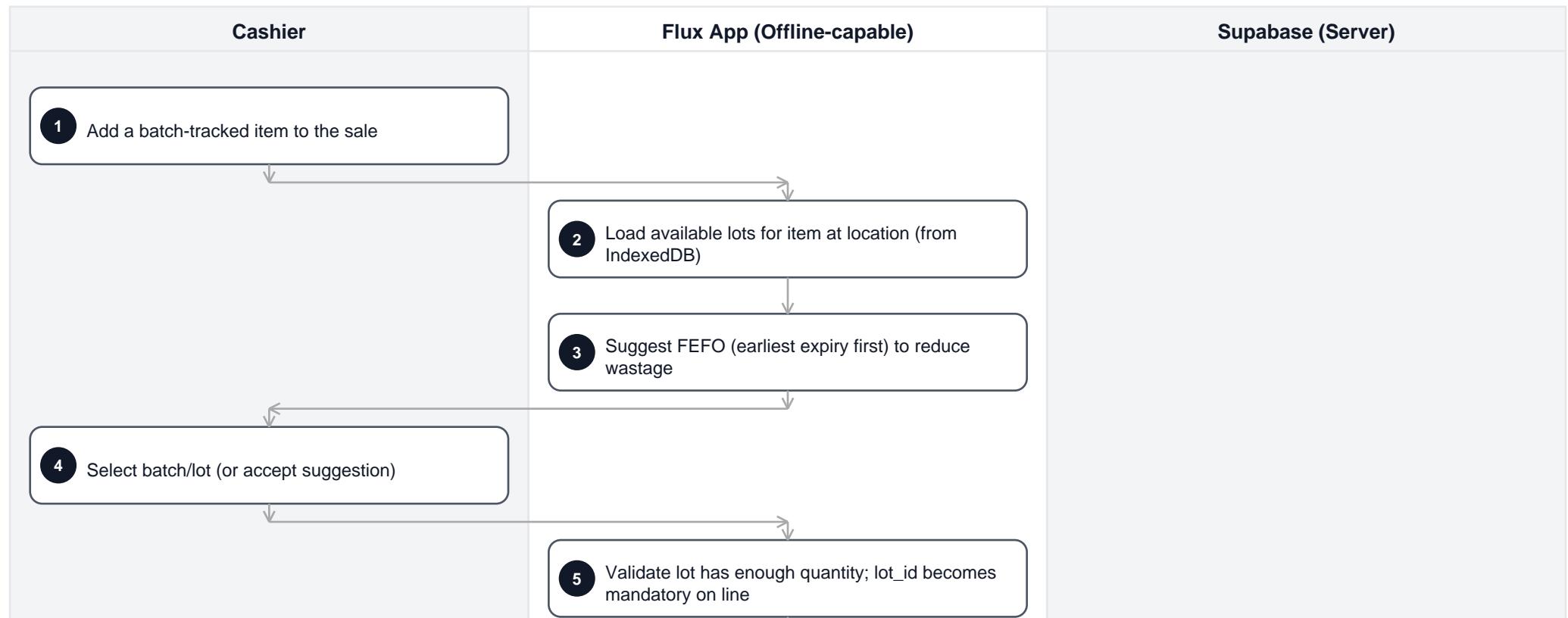
### Data impact (what gets created/updated)

- Tables impacted: stock\_transfers, stock\_transfer\_lines, stock\_moves, stock\_balances (source and destination).
- Optional: stock\_lots references on transfer lines when batch-tracked.

## UF-12: Batch Selection in POS (Batch-tracked sale line)

Phase: Phase 1 | Primary actor(s): Cashier

Key FR IDs: FR-031, FR-033, FR-040



### Key rules

- Batch-tracked items cannot be posted without selecting a batch/lot
- Lot availability is checked locally first, then confirmed on sync.
- Expiry-aware selection supports pharmacy-like workflows without full manufacturing complexity

### Data impact (what gets created/updated)

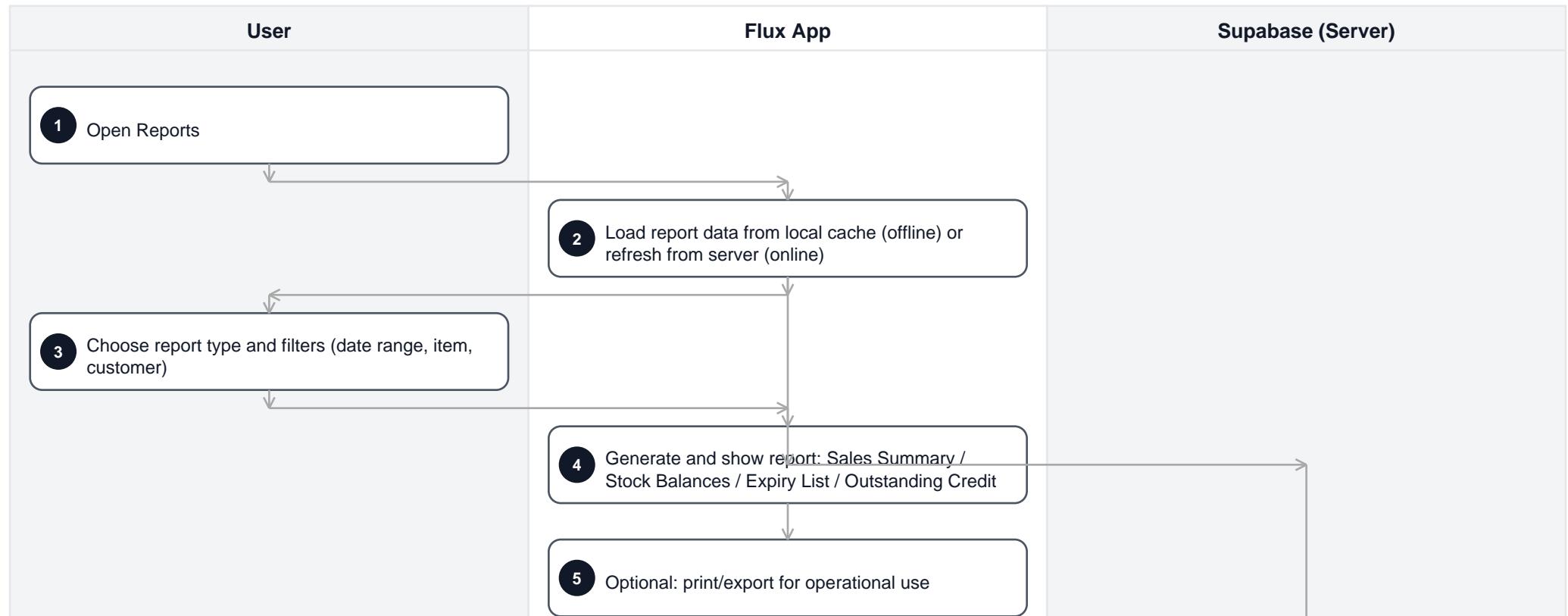
- Tables impacted: sales\_invoice\_lines (lot\_id), stock\_moves (lot\_id), stock\_balances (location\_id+item\_id+lot\_id).
- Lots are created in GRN and consumed in POS.

7 Persist sale + stock-out moves against lot\_id

## UF-13: Operational Reports (Sales Summary, Stock Balances, Expiry, Outstanding Credit)

Phase: Phase 1 | Primary actor(s): Store Manager / Super Admin

Key FR IDs: FR-050, FR-051, FR-052, FR-053, FR-006



### Key rules

- Reports must work with cached data offline; online refresh should be one tap.
- Phase 0+1 reports are operational summaries, not full accounting packs.
- Expiry report reads from stock lots (batch/expiry). Outstanding credit reads from credit invoices.

### Data impact (what gets created/updated)

- Read-only: sales, payments, stock\_balances, stock\_lots, customers.
- No writes - only refresh pulls from server.

## End-to-end example scenario (Phase 0 -> Phase 1)

This scenario shows how the Phase 0 + Phase 1 flows connect in real life. Assume a retail shop with two branches (Location A and Location B).

### Scenario steps

- Bootstrap: Install Flux and login online (UF-00). Select Location A and run the first sync (UF-01).
- Setup: Store Manager creates Category/Unit/Item and confirms barcodes (UF-02). Cashier syncs the new item list (UF-01).
- Offline sale: Internet drops. Cashier sells items using scan/search, discount and cash/card payment, prints the receipt, and the sale is queued for sync (UF-03 + UF-05).
- Reconnection: When internet returns, the sync queue pushes pending sales to Supabase, then the app marks them as Synced (UF-05).
- Purchasing: Store Manager receives new stock via GRN. For batch-tracked items, batch and expiry are captured. Stock increases after posting (UF-10).
- Multi-branch: Location A transfers stock to Location B using a 2-step transfer-out then receive workflow (UF-11).
- Expiry control: Cashier must select a batch when selling a batch-tracked item. App suggests earliest-expiry batch first (UF-12).
- Operations: Managers check Sales Summary, Stock Balances, Expiry List, and Outstanding Credit Summary using cached data (offline) or refreshed data (online) (UF-13).

### What Phase 0+1 intentionally does NOT do

- Full accounting (GL, journal postings, trial balance, financial statements).
- Complex correction workflows like posted invoice/GRN post-adjustments (kept for later phases).
- Advanced promotions, loyalty, and deep analytics.

## Flow-to-requirements mapping (Functional Requirements)

This table links each user flow to the key FR IDs from the Requirements Catalogue.

Flow	Phase	Primary actor(s)	Key FR IDs
UF-00	P0	Any user (Cashier / Store Manager / Super Admin)	FR-001, FR-002, FR-004, FR-005, FR-006, FR-017
UF-01	P0	Any user after login	FR-011, FR-015, FR-017, FR-006
UF-02	P0	Store Manager / Super Admin	FR-021, FR-026, FR-028, FR-027, FR-006
UF-03	P0	Cashier	FR-010, FR-035, FR-036, FR-037, FR-038, FR-040, FR-041, FR-042, FR-018
UF-04	P1	Cashier	FR-024, FR-025, FR-038, FR-039, FR-053, FR-018
UF-05	P0	System (background) + user visibility	FR-012, FR-013, FR-014, FR-015, FR-016, FR-018, FR-019, FR-020
UF-10	P1	Store Manager	FR-023, FR-031, FR-032, FR-043, FR-044, FR-045, FR-046
UF-11	P1	Source Manager + Destination Manager	FR-047, FR-048, FR-049, FR-006
UF-12	P1	Cashier	FR-031, FR-033, FR-040
UF-13	P1	Store Manager / Super Admin	FR-050, FR-051, FR-052, FR-053, FR-006