

Flux - User Stories + Acceptance Criteria Backlog

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

Date: January 31, 2026 | Tech: Next.js (App Router), TypeScript, Tailwind, Supabase, IndexedDB

How to use this backlog

This document is a single backlog for Phase 0 and Phase 1. Each row is a user story with: priority, a clear Definition of Done (DoD tags), acceptance criteria, and acceptance tests. Phase 0 proves the end-to-end stack and offline sale sync. Phase 1 delivers the trimmed MVP for real daily use.

Priority scale

MUST = required for the phase release. **SHOULD** = valuable but can slip without blocking the release.

Definition of Done (DoD tags)

DoD-STD: Built, reviewed, and merged; TypeScript types pass; basic unit tests added/updated; no critical console errors.

DoD-RLS: All server writes/reads are scoped by location_id and pass Supabase RLS tests for the role(s).

DoD-OFF: Flow works offline (no internet): data saved locally, visible in UI, and appears in sync queue.

DoD-SYNC: After reconnect, queued items sync successfully; failures show clear status; retry works.

DoD-UI: UI validation matches scope fields; required fields enforced; error messages are clear.

DoD-SEC: No secrets stored in IndexedDB; auth tokens handled safely; logout clears local sensitive data.

DoD-REP: Report matches expected totals and filters; supports branch/location filtering.

DoD-DATA: DB schema/migrations updated (if needed) and aligns with Core ERD rules (UUID PKs, audit fields).

Backlog table

Phases: P0 = Phase 0 (Walking Skeleton), P1 = Phase 1 (Trimmed MVP).

| ID | Epic | Phase | Prio | User Story | DoD tags | Acceptance Criteria | Acceptance Tests | Req refs |
|--------|-------------------|-------|------|---|----------------------------|--|--|------------------------|
| P0-001 | Platform & PWA | P0 | MUST | As a user, I want to install Flux as a PWA on desktop and mobile, so that I can use it like an app even with poor internet. | DoD-STD, DoD-UI | <ul style="list-style-type: none"> Install prompt works (or manual install) in supported browsers. App launches in standalone mode after install. App opens to login screen even when offline (after first load). | <ul style="list-style-type: none"> Given a supported browser, when I install the app, then it appears as an installed app icon. Given the app is installed, when I open it with no internet, then the login screen loads without blank errors. | FR-001 |
| P0-002 | Platform & PWA | P0 | MUST | As a user, I want the core app shell to load offline using a service worker cache, so that I can open the app and reach key pages without internet. | DoD-STD, DoD-OFF | <ul style="list-style-type: none"> Service worker caches the app shell (core routes/assets). With network disabled, navigating to key routes does not break (login, POS shell). | <ul style="list-style-type: none"> Given the app was opened once online, when I refresh offline, then core UI loads from cache. Given I open POS route offline, then I see the POS shell (even if data is limited). | FR-002 |
| P0-003 | Security & Access | P0 | MUST | As a developer, I want Supabase Auth login to protect all business screens, so that only authenticated users can access Flux. | DoD-STD, DoD-SEC | <ul style="list-style-type: none"> Unauthenticated users cannot access masters, POS, reports, or transactions. Session persists across refresh; logout ends the session. | <ul style="list-style-type: none"> Given I am logged out, when I open any protected route, then I am redirected to login. Given I logout, when I refresh, then I remain logged out. | FR-004 |
| P0-004 | Security & Access | P0 | MUST | As a developer, I want basic branch isolation using location_id + RLS for Phase 0 tables, so that one branch cannot see another branch's data. | DoD-STD, DoD-RLS, DoD-DATA | <ul style="list-style-type: none"> RLS enabled for core tables used in Phase 0. All queries include location_id. User from Location A cannot read/write Location B rows. | <ul style="list-style-type: none"> Given two locations exist, when a Location A user queries categories, then only Location A rows return. Given a Location A user tries to insert a Location B row, then Supabase rejects it. | FR-006, NFR-003 |
| P0-005 | Master Data | P0 | MUST | As a store admin, I want to create a minimal set of master data (category, unit, item), so that I can sell a real item in the walking skeleton. | DoD-STD, DoD-UI, DoD-DATA | <ul style="list-style-type: none"> Can create at least 1 category, 1 unit, 1 item (with barcode). Item appears in the POS item list/search. | <ul style="list-style-type: none"> Given I create a category/unit/item, when I open POS, then I can find the item. Given I create an item with a barcode, when I scan/type it, then the item is found. | FR-021, FR-028, FR-026 |
| P0-006 | Sales / POS | P0 | MUST | As a cashier, I want a minimal POS that lets me sell one item, so that I can complete a sale end-to-end. | DoD-STD, DoD-UI | <ul style="list-style-type: none"> POS shows the dummy item and allows add-to-cart. Cash payment can be recorded and the sale can be posted. A posted sale appears in the local sales list. | <ul style="list-style-type: none"> Given I add the item to cart, when I post as cash, then an invoice is created locally. Given I view sales list, then I can see the posted invoice with totals. | FR-035, FR-038 |

| ID | Epic | Phase | Prio | User Story | DoD tags | Acceptance Criteria | Acceptance Tests | Req refs |
|--------|-------------------|-------|------|---|-------------------------------------|--|---|--|
| P0-007 | Offline & Sync | P0 | MUST | As a cashier, I want to post sales while offline and sync later, so that sales continue without internet. | DoD-STD, DoD-OFF, DoD-SYNC | <ul style="list-style-type: none"> When offline, posting a sale stores it in IndexedDB and marks it as pending sync. When internet returns, the sale syncs to Supabase automatically. Stock on hand updates after sync. | <ul style="list-style-type: none"> Given no internet, when I post a sale, then it is saved locally and queued. Given internet returns, when sync runs, then the sale appears in Supabase and local status becomes Synced. | FR-010, FR-011, FR-015, FR-040, NFR-005 |
| P0-008 | Offline & Sync | P0 | MUST | As a user, I want to see online/offline state and pending sync count, so that I know whether data is synced. | DoD-STD, DoD-UI | <ul style="list-style-type: none"> UI shows Online/Offline indicator. UI shows Unsynced count. Each transaction shows status (Draft/Pending/Synced/Failed). | <ul style="list-style-type: none"> Given I go offline, when I view the header, then I see Offline. Given I have 2 pending items, then Unsynced count shows 2. | FR-017, FR-018 |
| P0-009 | Reporting | P0 | MUST | As a manager, I want a simple sales report/list to confirm the data flow, so that I can verify sales were captured and synced. | DoD-STD, DoD-REP | <ul style="list-style-type: none"> Sales list shows invoices created offline and after sync. Sales list can filter by status (Pending/Synced/Failed). | <ul style="list-style-type: none"> Given I posted a sale offline, when I open sales list, then I can see it with Pending status. Given it syncs, when I refresh list, then it shows Synced. | FR-018, FR-050 |
| P1-001 | Security & Access | P1 | MUST | As a super admin, I want to manage users with roles (Super Admin/Manager/Cashier), so that each staff member has correct access. | DoD-STD, DoD-RLS, DoD-UI | <ul style="list-style-type: none"> User can be assigned one of the three roles. UI adapts per role (nav and actions). RLS policies align with roles. | <ul style="list-style-type: none"> Given I assign Cashier role, when the user logs in, then inventory pages are blocked. Given I assign Super Admin, when the user opens locations, then cross-branch access works. | FR-005, FR-007, FR-008, FR-009 |
| P1-002 | Offline & Sync | P1 | MUST | As a developer, I want a durable client-side sync queue with retry safety, so that offline data is not lost and duplicates are prevented. | DoD-STD, DoD-OFF, DoD-SYNC, DoD-SEC | <ul style="list-style-type: none"> Queue records include entity_type, payload, location_id, created_at, status. App restart does not lose queued items. Retrying sync does not create duplicates in Supabase. | <ul style="list-style-type: none"> Given I create 3 items offline, when I close and reopen the app, then they are still pending. Given a sync request is retried, then only one row exists in Supabase for that UUID. | FR-012, FR-013, FR-019, NFR-005, NFR-006 |
| P1-003 | Offline & Sync | P1 | MUST | As a developer, I want client-only sync flags (no offline flags stored in Supabase), so that the DB stays clean and consistent. | DoD-STD, DoD-DATA | <ul style="list-style-type: none"> No 'synced' or 'offline_status' columns in Supabase tables. Sync state is tracked only in IndexedDB/local metadata. | <ul style="list-style-type: none"> Given a synced invoice exists in Supabase, then it has no offline-only fields. Given I clear local storage, then server data remains valid. | FR-014 |
| P1-004 | Offline & Sync | P1 | MUST | As a developer, I want a simple conflict policy (Last-Write-Wins) for baseline, so that sync conflicts resolve predictably. | DoD-STD, DoD-SYNC | <ul style="list-style-type: none"> If two updates conflict, the newest update wins based on updated_at/version. Conflicts are logged for later review (basic). | <ul style="list-style-type: none"> Given two edits happen, when both sync, then the latest updated_at is the final state. Given a conflict occurs, then the user can see a conflict note in logs/status. | FR-016 |

| ID | Epic | Phase | Prio | User Story | DoD tags | Acceptance Criteria | Acceptance Tests | Req refs |
|--------|----------------|-------|--------|---|----------------------------|--|---|-------------------------|
| P1-005 | Offline & Sync | P1 | SHOULD | As a developer, I want exponential backoff for failed sync retries, so that the system is gentle on network and server. | DoD-STD, DoD-SYNC | <ul style="list-style-type: none"> Failed sync items retry with increasing delay. User can also manually retry a failed item. | <ul style="list-style-type: none"> Given a failure, when retry happens, then next attempt is delayed longer than previous. Given a failed item, when I tap Retry, then it attempts sync immediately. | FR-020, NFR-007 |
| P1-006 | Master Data | P1 | MUST | As a manager, I want full CRUD for Categories and Sub-Categories with required fields, so that I can manage my product structure. | DoD-STD, DoD-UI, DoD-RLS | <ul style="list-style-type: none"> Create/edit/delete categories with: Category No, Name, Short Code (and Level if applicable). Sub-category must link to a valid category. Validations prevent saving missing required fields. | <ul style="list-style-type: none"> Given missing Category Name, when I save, then the form blocks with an error. Given a sub-category without parent, when I save, then it is rejected. | FR-021, FR-022 |
| P1-007 | Master Data | P1 | MUST | As a manager, I want CRUD for Units supporting decimal quantities, so that I can sell and purchase items in correct units. | DoD-STD, DoD-UI | <ul style="list-style-type: none"> Units can be created/edited. Quantities allow decimals where configured (e.g., 1.5). | <ul style="list-style-type: none"> Given unit allows decimals, when I enter 1.5, then it saves and calculations work. Given unit is whole-only, when I enter 1.2, then it is rejected. | FR-028 |
| P1-008 | Master Data | P1 | MUST | As a manager, I want CRUD for Items with barcode and unit links, so that items are searchable and consistent. | DoD-STD, DoD-UI, DoD-RLS | <ul style="list-style-type: none"> Item form includes barcode, unit, category links and required fields. Item appears in POS search and barcode lookup. | <ul style="list-style-type: none"> Given I create an item, when I search in POS, then it appears. Given I scan barcode, then the correct item loads. | FR-026, FR-036 |
| P1-009 | Master Data | P1 | MUST | As a developer, I want barcode uniqueness enforcement per location, so that duplicate scans never point to multiple items. | DoD-STD, DoD-DATA, DoD-RLS | <ul style="list-style-type: none"> Cannot create two items with same barcode in the same location. Same barcode across different locations follows policy (allowed only if specified). | <ul style="list-style-type: none"> Given an existing barcode, when I create another item with same barcode in same location, then save fails. Given different location, when policy allows, then save succeeds. | FR-027 |
| P1-010 | Master Data | P1 | MUST | As a manager, I want CRUD for Suppliers and Customers including credit terms, so that I can manage purchasing and credit sales. | DoD-STD, DoD-UI, DoD-RLS | <ul style="list-style-type: none"> Supplier and Customer records can be created/edited with required IDs and names. Customer includes Credit Limit and Credit Days. Records can be selected in GRN/POS. | <ul style="list-style-type: none"> Given I create a supplier, when I create a GRN, then the supplier can be selected. Given I create a customer, when I do a credit sale, then the customer can be selected. | FR-023, FR-024, FR-025 |
| P1-011 | Master Data | P1 | MUST | As a super admin, I want CRUD for Locations (branches) and user-to-location assignment, so that multi-branch operations are possible. | DoD-STD, DoD-RLS, DoD-DATA | <ul style="list-style-type: none"> Locations can be created/edited by Super Admin. Users are assigned to one location for baseline (manager/cashier). All transactions store location_id. | <ul style="list-style-type: none"> Given two locations, when a manager logs in, then they only see their assigned location data. Given a transaction, then location_id is stored and enforced. | FR-029, FR-006, NFR-003 |

| ID | Epic | Phase | Prio | User Story | DoD tags | Acceptance Criteria | Acceptance Tests | Req refs |
|--------|---------------------|-------|------|---|------------------------------------|--|--|--------------------------------|
| P1-012 | Inventory & Batches | P1 | MUST | As a developer, I want inventory model using stock moves + stock balances, so that stock on hand is derived reliably. | DoD-STD, DoD-DATA, DoD-RLS | <ul style="list-style-type: none"> • Posting sales/GRNs/transfers creates stock_moves rows. • stock_balances reflects summed moves per (location,item,lot). • Stock cannot drift due to duplicates. | <ul style="list-style-type: none"> • Given a GRN posts +10, when I check stock, then on-hand increases by 10. • Given a sale posts -2, then on-hand decreases by 2. | FR-030, FR-040, FR-045, FR-049 |
| P1-013 | Inventory & Batches | P1 | MUST | As a manager, I want simple stock adjustment (increase/decrease) with a reason note, so that I can correct stock mistakes safely. | DoD-STD, DoD-UI, DoD-RLS, DoD-DATA | <ul style="list-style-type: none"> • Manager can create an adjustment with: item, qty (+/-), reason, optional comment. • Cashier cannot access adjustments. • Adjustment posting creates stock_moves. | <ul style="list-style-type: none"> • Given manager role, when I post +5 adjustment, then on-hand increases by 5 and move is logged. • Given cashier role, when I try to open adjustments, then access is denied. | (baseline SRS) |
| P1-014 | Inventory & Batches | P1 | MUST | As a manager, I want batch-tracked item configuration, so that batch rules apply only when needed. | DoD-STD, DoD-UI | <ul style="list-style-type: none"> • Item can be marked Batch-Tracked or Non-Batch. • Batch setting affects GRN and POS requirements. | <ul style="list-style-type: none"> • Given Batch-Tracked item, when I create GRN line, then batch/expiry fields are required. • Given Non-Batch item, batch selection is not required. | FR-031, FR-032, FR-033 |
| P1-015 | Inventory & Batches | P1 | MUST | As a cashier, I want batch selection at POS for batch-tracked items, so that the correct batch is sold and expiry is controlled. | DoD-STD, DoD-UI, DoD-OFF | <ul style="list-style-type: none"> • POS forces batch selection for batch-tracked items. • Batch list shows available batches and quantities (if available). | <ul style="list-style-type: none"> • Given a batch-tracked item in cart, when I try to post without batch, then it blocks. • Given I select a batch, then posting succeeds. | FR-033 |
| P1-016 | Sales / POS | P1 | MUST | As a cashier, I want create, save, and resume a Draft sale, so that I can handle interruptions without losing work. | DoD-STD, DoD-OFF, DoD-UI | <ul style="list-style-type: none"> • Draft can be saved locally. • Draft can be reopened and edited. • Posting changes status and locks key totals. | <ul style="list-style-type: none"> • Given I save a draft, when I reopen, then cart items persist. • Given I post, then the invoice becomes Posted and prints. | FR-035 |
| P1-017 | Sales / POS | P1 | MUST | As a cashier, I want barcode scan and text search in POS (offline-capable), so that I can find items quickly. | DoD-STD, DoD-OFF | <ul style="list-style-type: none"> • Barcode input finds item (keyboard-wedge). • Text search finds items by name/sku. • Works offline using cached item list. | <ul style="list-style-type: none"> • Given offline, when I scan barcode, then item loads from cache. • Given I type part of name, then matching items appear. | FR-036, FR-011 |
| P1-018 | Sales / POS | P1 | MUST | As a cashier, I want edit line quantity and apply simple discount, so that I can price correctly at checkout. | DoD-STD, DoD-UI | <ul style="list-style-type: none"> • Quantity can be edited per line. • Discount can be applied (percentage or amount - baseline). • Totals recalc instantly. | <ul style="list-style-type: none"> • Given I change qty, when totals recalc, then invoice total updates correctly. • Given I apply discount, then net total reflects discount. | FR-037 |
| P1-019 | Sales / POS | P1 | MUST | As a cashier, I want support payments: cash, card, and credit, so that I can complete real transactions. | DoD-STD, DoD-UI | <ul style="list-style-type: none"> • Posting requires selecting one of the allowed payment types. • Cash and card require amount; credit requires customer. | <ul style="list-style-type: none"> • Given I select cash and enter amount, when I post, then sale posts. • Given I select credit without customer, then posting is blocked. | FR-038, FR-039 |

| ID | Epic | Phase | Prio | User Story | DoD tags | Acceptance Criteria | Acceptance Tests | Req refs |
|--------|-------------|-------|------|---|------------------------------------|---|---|------------------------|
| P1-020 | Sales / POS | P1 | MUST | As a manager, I want negative stock policy (WARN or BLOCK), so that stock rules match business needs. | DoD-STD, DoD-UI, DoD-RLS | <ul style="list-style-type: none"> Policy can be set per location (baseline). If BLOCK, posting sale/transfer fails when stock would go negative. If WARN, user sees warning but can proceed (role-based if needed). | <ul style="list-style-type: none"> Given BLOCK policy, when stock is 0 and I sell 1, then posting is blocked. Given WARN policy, when stock is 0 and I sell 1, then a warning is shown and user can proceed. | FR-042 |
| P1-021 | Sales / POS | P1 | MUST | As a cashier, I want print a receipt after posting a sale, so that customers get proof of purchase. | DoD-STD, DoD-UI | <ul style="list-style-type: none"> Receipt is printable via browser print. Receipt includes invoice no, date/time, items, totals, payment type, branch. | <ul style="list-style-type: none"> Given I post a sale, when I click Print, then print preview opens with receipt format. Given printed receipt, it contains all required fields. | FR-041 |
| P1-022 | Sales / POS | P1 | MUST | As a developer, I want unique invoice identity per location to avoid duplicates, so that offline retries do not create double invoices. | DoD-STD, DoD-DATA, DoD-SYNC | <ul style="list-style-type: none"> Invoice number/UUID uniqueness enforced per location. Sync retries do not create duplicate invoices. | <ul style="list-style-type: none"> Given same invoice UUID, when sync repeats, then only one invoice exists in Supabase. Given same invoice number, system prevents duplicates within location. | FR-034, NFR-006 |
| P1-023 | Sales / POS | P1 | MUST | As a cashier, I want void a sale before finalising and process a simple return/refund, so that mistakes can be corrected. | DoD-STD, DoD-UI, DoD-SYNC | <ul style="list-style-type: none"> Can void an unposted/draft invoice. Return/refund references an existing invoice (baseline simple flow). Stock is adjusted accordingly after sync. | <ul style="list-style-type: none"> Given a draft invoice, when I void it, then it is marked Voided and not synced as a sale. Given a posted invoice, when I create a return, then stock increases and refund is recorded. | (roadmap) |
| P1-024 | Purchasing | P1 | MUST | As a manager, I want create and manage Purchase Orders (PO) with basic statuses, so that I can track what was ordered vs received. | DoD-STD, DoD-OFF, DoD-UI, DoD-SYNC | <ul style="list-style-type: none"> Can create PO with supplier, items, qty, cost. Status shows Open/Partial/Closed based on received qty. PO works offline (draft at least). | <ul style="list-style-type: none"> Given a new PO, when I add items and save, then PO exists in list with Open status. Given I receive partial GRN, then PO becomes Partial. | (charter, roadmap) |
| P1-025 | Purchasing | P1 | MUST | As a manager, I want create a GRN draft by selecting supplier and receiving items, so that stock intake is recorded. | DoD-STD, DoD-OFF, DoD-UI | <ul style="list-style-type: none"> GRN draft can be saved and resumed offline. GRN includes supplier, date, lines with qty and unit cost. Batch/expiry required for batch-tracked items. | <ul style="list-style-type: none"> Given offline, when I save GRN draft, then it is stored locally. Given batch-tracked item, when batch is missing, then GRN cannot be posted. | FR-043, FR-044, FR-032 |
| P1-026 | Purchasing | P1 | MUST | As a manager, I want post a GRN to update stock on hand, so that inventory increases reliably. | DoD-STD, DoD-SYNC, DoD-DATA | <ul style="list-style-type: none"> Posting GRN creates stock-in moves. Stock balances increase accordingly. GRN appears as Posted and is locked from edit. | <ul style="list-style-type: none"> Given a GRN posts with +10, then stock on hand increases by 10. Given a GRN is posted, when I edit, then system blocks edits. | FR-045, FR-046 |

| ID | Epic | Phase | Prio | User Story | DoD tags | Acceptance Criteria | Acceptance Tests | Req refs |
|--------|-----------|-------|--------|---|-------------------------------------|---|--|-----------------|
| P1-027 | Transfers | P1 | MUST | As a manager, I want create a transfer-out from source location, so that I can move stock between branches. | DoD-STD, DoD-OFF, DoD-SYNC, DoD-RLS | <ul style="list-style-type: none"> Transfer-out captures from_location, to_location, and lines. Transfer-out can be saved offline and posted. Posted transfer-out reduces source stock. | <ul style="list-style-type: none"> Given stock available, when I post transfer-out, then source on-hand decreases. Given offline, when I save transfer-out, then it queues for sync. | FR-047, FR-049 |
| P1-028 | Transfers | P1 | MUST | As a manager, I want receive a transfer-in at destination location, so that destination stock increases correctly. | DoD-STD, DoD-SYNC, DoD-RLS | <ul style="list-style-type: none"> Destination can view pending inbound transfers. Receiving transfer-in increases destination stock. Transfer is marked Completed. | <ul style="list-style-type: none"> Given an outbound transfer exists, when destination receives, then destination on-hand increases. Given transfer completed, then it cannot be received twice. | FR-047, NFR-006 |
| P1-029 | Transfers | P1 | MUST | As a manager, I want batch-aware transfers for batch-tracked items, so that the correct batches move across branches. | DoD-STD, DoD-UI, DoD-DATA | <ul style="list-style-type: none"> Transfer lines require batch selection for batch-tracked items. Stock moves preserve lot/batch identity. | <ul style="list-style-type: none"> Given batch-tracked item, when batch is missing, then transfer cannot be posted. Given posted transfer, lot_id is recorded in stock moves. | FR-048 |
| P1-030 | Reporting | P1 | MUST | As a manager, I want basic sales summary report by date, user, and branch, so that I can track performance quickly. | DoD-STD, DoD-REP, DoD-RLS | <ul style="list-style-type: none"> Report shows total sales, counts, and breakdown by payment type. Filters: date range, user, location. Works with synced data; shows note for pending. | <ul style="list-style-type: none"> Given a date range, when I run report, then totals match invoice list sums. Given a location filter, then only that location data shows. | FR-050 |
| P1-031 | Reporting | P1 | MUST | As a manager, I want stock balance view per item and per batch (where applicable), so that I know current stock levels. | DoD-STD, DoD-REP, DoD-RLS | <ul style="list-style-type: none"> Stock screen shows on-hand per item for current location. For batch-tracked items, show per batch/expiry breakdown. Supports search/filter. | <ul style="list-style-type: none"> Given an item, when I open stock view, then on-hand matches transactions. Given batch item, then each batch appears with expiry date. | FR-051 |
| P1-032 | Reporting | P1 | MUST | As a manager, I want expiry warning report listing batches nearing expiry, so that I can prevent selling expired goods. | DoD-STD, DoD-REP, DoD-RLS | <ul style="list-style-type: none"> Report lists batches expiring within configurable window (baseline fixed like 30/60/90). Includes item, batch, expiry date, qty, location. | <ul style="list-style-type: none"> Given an expiry window, when I run report, then only matching batches appear. Given a batch is expired, then it is flagged clearly. | FR-052 |
| P1-033 | Reporting | P1 | SHOULD | As a manager, I want outstanding credit summary by customer, so that I can manage credit risk. | DoD-STD, DoD-REP | <ul style="list-style-type: none"> Report lists customers with outstanding balances. Shows total outstanding and ageing buckets (basic). | <ul style="list-style-type: none"> Given credit invoices exist, when I open credit report, then customers are listed with totals. Given no credit invoices, report shows empty state. | FR-053 |

| ID | Epic | Phase | Prio | User Story | DoD tags | Acceptance Criteria | Acceptance Tests | Req refs |
|-------|---------------|-------|--------|---|------------------|---|---|------------------|
| Q-001 | Quality (NFR) | P0 | MUST | As a developer, I want fast offline responsiveness for POS lookup, so that cashier experience is smooth. | DoD-STD | <ul style="list-style-type: none"> • App remains responsive offline; no UI freezes during POS lookup. • Typical barcode lookup returns quickly on mid-range device. | <ul style="list-style-type: none"> • Given offline with 1k items cached, when I scan barcode, then result appears quickly. • Given rapid scans, then UI remains usable. | NFR-001, NFR-002 |
| Q-002 | Quality (NFR) | P0 | MUST | As a developer, I want avoid storing sensitive secrets in IndexedDB, so that local storage is safe. | DoD-STD, DoD-SEC | <ul style="list-style-type: none"> • No long-lived secrets stored in IndexedDB. • Logout clears cached sensitive user/session data. | <ul style="list-style-type: none"> • Given I inspect IndexedDB, then no secrets are visible. • Given logout, then user-specific cached secrets are removed. | NFR-004 |
| Q-003 | Quality (NFR) | P0 | SHOULD | As a developer, I want typed models and consistent API contracts in TypeScript, so that changes are safer and faster. | DoD-STD | <ul style="list-style-type: none"> • Core entities share types/interfaces between UI, offline queue, and sync layer. • Breaking changes fail compilation. | <ul style="list-style-type: none"> • Given I change a field, then type errors show all impacted places. • Given a sync payload, it validates against types. | NFR-008 |

Glossary (baseline)

PWA: Installable web app (manifest + service worker).

IndexedDB: Browser database for offline storage and the sync queue.

Supabase RLS: Row Level Security in Postgres to isolate branches (location_id).

location_id: UUID that identifies the branch/location and scopes all data.

LWW: Last-Write-Wins conflict policy (newest update wins).

GRN: Goods Received Note (posting increases stock).

PO: Purchase Order (planned purchasing; receiving is done via GRN).

Stock Moves: Ledger rows representing +/- changes to stock.

Stock Balances: Current stock on hand derived from stock moves.