BZ Logistics Assets System — Technical Requirements (Updated v2)

# 1. Introduction

The BZ Logistics Assets System is designed to manage, track, and audit all assets under the logistics department of BZ. The system ensures accurate registration, bulk import, lifecycle tracking, reporting, and compliance with organizational requirements. This reflects the refined database schema, user role/location rules, and asset lifecycle history design.

# 2. Goals & Scope

Primary goals include:

* Bulk-import assets using a restricted Excel template with controlled picklists.
* Track asset lifecycle: creation, assignment, location, warranty, and status changes.
* Provide advanced search, filtering, and reporting with export functionality.
* Support compliance requirements (audit logs, approvals, backups, reporting).

# 3. Personas & Roles

* PR Admin: Manages PRs and expected PR items for one assigned location.
* Logistics Admin: Manages asset registration, transfers, status updates for one assigned location.
* Super Admin: Approves sensitive changes (info edits) and can manage any location.

# 4. Features / Functional Requirements

## 4.1 PR Management (PR Admin)

Each PR has code, date, file and total price. PR Admin selects category → sub-category → fixed item. Stored in prs and pr\_items.

## 4.2 Asset Management (Logistics Admin)

Each asset is created individually or via bulk import. Features:

* Select PR but without seeing PR’s expected items (to enable later reconciliation).
* Assign to one location and floor.
* Link to fixed item, brand, color, supplier, status, holder, acquisition details, donor/budget codes.
* Attach documents (PR, warranty, photos).
* Generate QR/Barcode for each asset.

## 4.3 Lifecycle Tracking

All changes to assets are stored in immutable history events. Event types: CREATED, UPDATED, STATUS\_CHANGED, LOCATION\_CHANGED, HOLDER\_CHANGED, WARRANTY\_UPDATED, ATTRIBUTE\_UPDATED, ATTACHMENT\_ADDED/REMOVED, BULK\_IMPORT, BULK\_TRANSFER.

## 4.4 Bulk Import

Excel template includes controlled dropdowns for category → sub-category → fixed item, plus brand, color, supplier, status, location, floor, user. Free text allowed only in description and notes. Import process validates rows, previews errors, and commits valid assets.

## 4.5 Change Requests & Approvals

Certain actions (sensitive edits) require Super Admin approval. Logistics Admin submits a request. On approval, the actual change is applied to items and a history event is written.

## 4.6 Attributes System

Attributes can describe either fixed item models or individual assets. att\_options define allowed choices. Both fixed\_item\_attributes and asset\_attributes can be used depending on type. Payloads store IDs and labels to enable frozen history snapshots.

## 4.7 Search, Filters, and Reporting

Advanced filters across all asset fields (category, fixed item, brand, supplier, location, status, donor, dates). Export results to Excel or PDF. Prebuilt reports: inventory by location, expiring warranties, donor-funded assets, reconciliation between PR expected items and actual assets.

## 4.9 Notifications

System generates alerts for expiring warranties, assets in maintenance beyond threshold, and import errors. Delivered via email digest.

## 4.10 Audit & Compliance

Audit logs track all CRUD operations with actor, timestamp, and payload diff. Critical actions require reason. Soft-deletes with recovery enabled.

# 5. Non-Functional Requirements

* Security: RBAC with single role + single location per user, hashed passwords for admins only, CSRF protection, validation, file scanning.
* Performance: Handle 200k+ assets; bulk imports up to 10k rows; indexed queries under 2s.
* Availability: 99.5% uptime target, staging/prod environments.
* Usability: WCAG AA compliance, Arabic/English support.
* - Scalability: Horizontally scalable API, job queues, object storage.
* - Observability: Centralized logs, import metrics, error tracking.

# 6. Tech Stack

Backend: Laravel + MySQL

Frontend: Angular + Tailwind CSS

Auth: Laravel Sanctum (SPA-friendly, cookie-based)

Queue: Redis + Horizon

Storage: S3-compatible for attachments

# 7. Database Design Highlights

* - Separate fixed\_items (models) vs items (physical assets).
* - pr\_items links PRs to fixed items with expected quantities.
* - user\_roles links each user to exactly one role and one location.
* - floors are scoped to locations (unique per location).
* - asset\_history\_events stores immutable lifecycle events.
* - change\_requests support approval flows.
* - attributes system supports both fixed models and individual assets.

# 8. Deliverables

* - Database schema with migrations and constraints.
* - Admin dashboard for assets, PRs, and reference data.
* - Import/Export center.
* - Reports and reconciliation tools.
* - Notifications and alerts.
* - User and admin guides.