

User Stories Baseline

Source: `~/ALAI/products/Plock/docs/demo-readiness/01-user-stories-baseline.md`

PLOCK — User Stories Baseline

Date: 2026-03-14

Purpose: Establish a first canonical user-story baseline for PLOCK using real repository evidence and authoritative product documents. This is not yet the final polished story set; it is the operational baseline for documentation, QA, and backlog slicing.

1. Scope and Method

This baseline is derived from:

- `docs/PRD.md`
- actual backend route/service structure
- actual frontend MFE structure
- integration modules present in the repository
- operational and regulatory documentation where relevant

This document intentionally avoids inventing features not supported by either:

- current product docs, or
 - current repository structure
-

2. Story Status Model

Each story is tagged as one of:

- **exists** — strongly supported by real code/routes/docs
- **partial** — visible in docs and/or code structure, but maturity unclear
- **planned** — described in docs, but implementation evidence is limited or phase-dependent

3. Receiving / Inbound

US-INB-001 — Receive goods via barcode

Status: exists

Evidence: `PRD.md`, `ReceivingRoutes.kt`, `ReceivingService.kt`, barcode-related backend structure

As a warehouse worker

I want to scan a barcode and identify the purchase order it belongs to

So that I can receive goods correctly without manual lookup

US-INB-002 — View open purchase orders

Status: exists

Evidence: `PRD.md`, `ReceivingRoutes.kt`, supplier/receiving route structure

As a warehouse manager

I want to see all open purchase orders and expected arrival dates

So that I can plan staffing and dock capacity

US-INB-003 — Guided putaway after receiving

Status: partial

Evidence: `PRD.md`, `LocationRoutes.kt`, `PutAwayService.kt`

As a warehouse worker

I want to be guided to the correct bin location after receiving

So that I do not need to memorize where products belong

US-INB-004 — Detect receiving discrepancies

Status: exists

Evidence: `PRD.md`, `ReceivingDiscrepancyTest.kt`, discrepancy migration/table evidence

As a warehouse manager

I want to be notified immediately when received quantities do not match the purchase order

So that I can resolve supplier issues before inventory is corrupted

4. Inventory Management

US-INV-001 — Real-time inventory visibility

Status: exists

Evidence: `PRD.md`, `InventoryRoutes.kt`, `InventoryService.kt`

As a warehouse manager

I want to see the exact quantity of a SKU by location in real time

So that I can answer operational questions without walking the warehouse

US-INV-002 — Reorder point alerting

Status: partial

Evidence: `PRD.md`, inventory domain present; implementation maturity unclear

As a warehouse manager

I want to receive an alert when inventory falls below reorder point

So that I can prevent avoidable stockouts

US-INV-003 — Zone-based cycle counts

Status: exists

Evidence: `PRD.md`, `CycleCountRoutes.kt`, `CycleCountService.kt`, tests

As a warehouse manager

I want to run a cycle count for a specific zone without stopping all operations

So that I can maintain inventory accuracy continuously

US-INV-004 — Mobile-assisted physical inventory count

Status: partial

Evidence: `PRD.md`, V7 mobile support migrations, runbook/mobile references

As a warehouse worker

I want to perform physical inventory counts on a mobile device

So that counting is faster and less error-prone than paper workflows

5. Orders / Outbound

US-OUT-001 — Release urgent orders quickly

Status: exists

Evidence: `PRD.md`, `OrderRoutes.kt`, order/picking domain structure

As a warehouse manager

I want to release urgent orders with a single action

So that I can respond quickly to escalations and deadlines

US-OUT-002 — Work from structured picking flow

Status: exists

Evidence: `PickingRoutes.kt`, `PickingService.kt`, `mfe-picking`

As a picker

I want to receive a structured picking workflow

So that I can pick efficiently and consistently

US-OUT-003 — Verify packing with product photo

Status: partial

Evidence: `PRD.md`, frontend/product/packing intent visible in docs

As a packing worker

I want to see a product photo during packing

So that I can confirm I am sealing the correct item

US-OUT-004 — Unified shipment tracking view

Status: partial

Evidence: `PRD.md`, `ShipmentRoutes.kt`, carrier integration structure

As a warehouse manager

I want to see outbound shipment status in one place

So that I can proactively handle delays and customer issues

6. Shipping / Carrier Operations

US-SHP-001 — Print carrier labels directly from Plock

Status: exists

Evidence: `PRD.md`, PostNord/DHL/Instabee integration code, shipment/carrier routes

As a warehouse worker

I want to print shipping labels directly from Plock

So that I do not need to log into separate carrier portals

US-SHP-002 — Use multiple carriers in one operational flow

Status: exists

Evidence: `CarrierRoutes.kt`, `CarrierIntegrationRoutes.kt`, multiple carrier adapters/services

As a warehouse manager

I want to work with PostNord, DHL, and Instabee in one system

So that outbound shipping does not depend on multiple manual carrier workflows

US-SHP-003 — Track carrier sync and integration health

Status: partial

Evidence: `integration_sync_log` schema, runbook sections for Fortnox/carrier failures

As a system operator

I want to inspect carrier/integration sync behavior

So that I can detect and troubleshoot failures quickly

7. AI / Smart Warehouse Operations

US-AI-001 — Ask warehouse questions in Swedish

Status: partial-to-exists

Evidence: `PRD.md`, `AI-STRATEGY.md`, `mfe-ai/`, product positioning

As a warehouse manager

I want to ask warehouse questions in Swedish and get immediate answers

So that I can make decisions without SQL, exports, or manual searching

US-AI-002 — Execute warehouse commands via AI with confirmation

Status: planned / partial

Evidence: `PRD.md`, `AI-STRATEGY.md`

As a warehouse manager

I want to trigger operational actions through natural language with explicit confirmation

So that I can manage exceptions faster than through deep UI navigation

US-AI-003 — Optimize pick routes

Status: partial-to-exists

Evidence: `PRD.md`, `AI-STRATEGY.md`, picking domain, runbook references to Smart Picking

As a warehouse manager

I want to use smart pick route optimization

So that workers spend less time walking and more time picking

8. Dashboard / Monitoring

US-DASH-001 — Operational dashboard

Status: exists

Evidence: `DashboardRoutes.kt`, `PRD.md`, dashboard domain

As a warehouse manager

I want to see a real-time dashboard with today's operational KPIs

So that I can spot bottlenecks before they become incidents

US-DASH-002 — Monitor shipment and fulfillment progress

Status: partial

Evidence: dashboard/shipment routes, PRD

As a operations lead

I want to monitor fulfillment progress and pending shipments

So that I can intervene before SLAs are missed

9. Users / Roles / Multi-Tenancy

US-SEC-001 — Manage users and roles

Status: exists

Evidence: `UserRoutes.kt`, `RoleRoutes.kt`, `RbacService.kt`

As an admin

I want to manage users and roles

So that only the right people can perform sensitive warehouse actions

US-SEC-002 — Restrict data by warehouse

Status: exists

Evidence: `TenantPlugin.kt`, `RLS-AUDIT.md`, RLS migrations

As a warehouse operator

I want to only see data for my warehouse

So that tenant/customer separation is enforced safely

US-SEC-003 — Keep tenant isolation enforced at the DB layer

Status: exists

Evidence: `RLS-AUDIT.md`, migration `V6__rls_tenant_isolation.sql`

As the platform

I want to enforce tenant isolation in PostgreSQL using RLS

So that application bugs do not automatically become cross-tenant data leaks

10. Integrations / Accounting

US-INT-001 — Sync warehouse/accounting data with Fortnox

Status: exists

Evidence: Fortnox integration modules, docs, secrets, runbook

As a Swedish SMB operator

I want to integrate Plock with Fortnox

So that warehouse operations and accounting stay aligned

US-INT-002 — Re-authorize or recover from Fortnox sync failure

Status: partial-to-exists

Evidence: Fortnox OAuth/sync routes, runbook incident section

As a tenant admin

I want to reconnect or recover the Fortnox integration when it fails

So that business operations are not blocked by stale tokens or sync errors

11. 3PL / Client Isolation (Phase-Dependent)

US-3PL-001 — Separate client inventory views

Status: planned

Evidence: `PRD.md`, `GTM-STRATEGY.md`, regulatory references

As a 3PL operations lead

I want to see inventory separated by client

So that I can manage a multi-client warehouse safely

US-3PL-002 — Support 3PL billing workflows

Status: planned

Evidence: REGULATORY.md, GTM and pricing logic

As a 3PL operator

I want to support warehousing-service billing workflows

So that customer charging is structured and scalable

12. Business / Sales / Market Layer

US-BIZ-001 — Position as Fortnox-next-step WMS

Status: planned/business

Evidence: GTM-STRATEGY.md

As the business

I want to position Plock as the natural next step after Fortnox Lager

So that market entry is focused and credible

US-BIZ-002 — Support Swedish SMB self-serve onboarding motion

Status: planned/business

Evidence: GTM and product docs

As the product business

I want to enable a low-friction onboarding and trial motion

So that acquisition cost stays low and distribution scales

13. Summary

Strongly supported by current code/docs

- receiving
- inventory
- orders

- picking
- shipments
- carriers
- RBAC
- tenant isolation
- Fortnox integration
- dashboard
- barcode
- audit/webhook/SSE infrastructure

Present but maturity unclear

- AI chat execution depth
- smart picking maturity
- mobile/PWA operational depth
- packing/photo verification
- full shipment monitoring UX
- reorder alerting maturity

More Phase 2 / strategic than current MVP

- 3PL client isolation workflows
 - 3PL billing
 - full commercial/demo/support packaging
 - polished sales enablement materials
-

14. Working Rule

This user-story baseline is the first canonical operational extraction.

Any future:

- feature catalog
- QA gate
- implementation backlog
- demo script
- support playbook

must map back to these stories and mark each item as:

- exists
- partial
- planned

Revision #3

Created 2026-03-14 12:03:42 UTC by John

Updated 2026-05-31 20:05:17 UTC by John