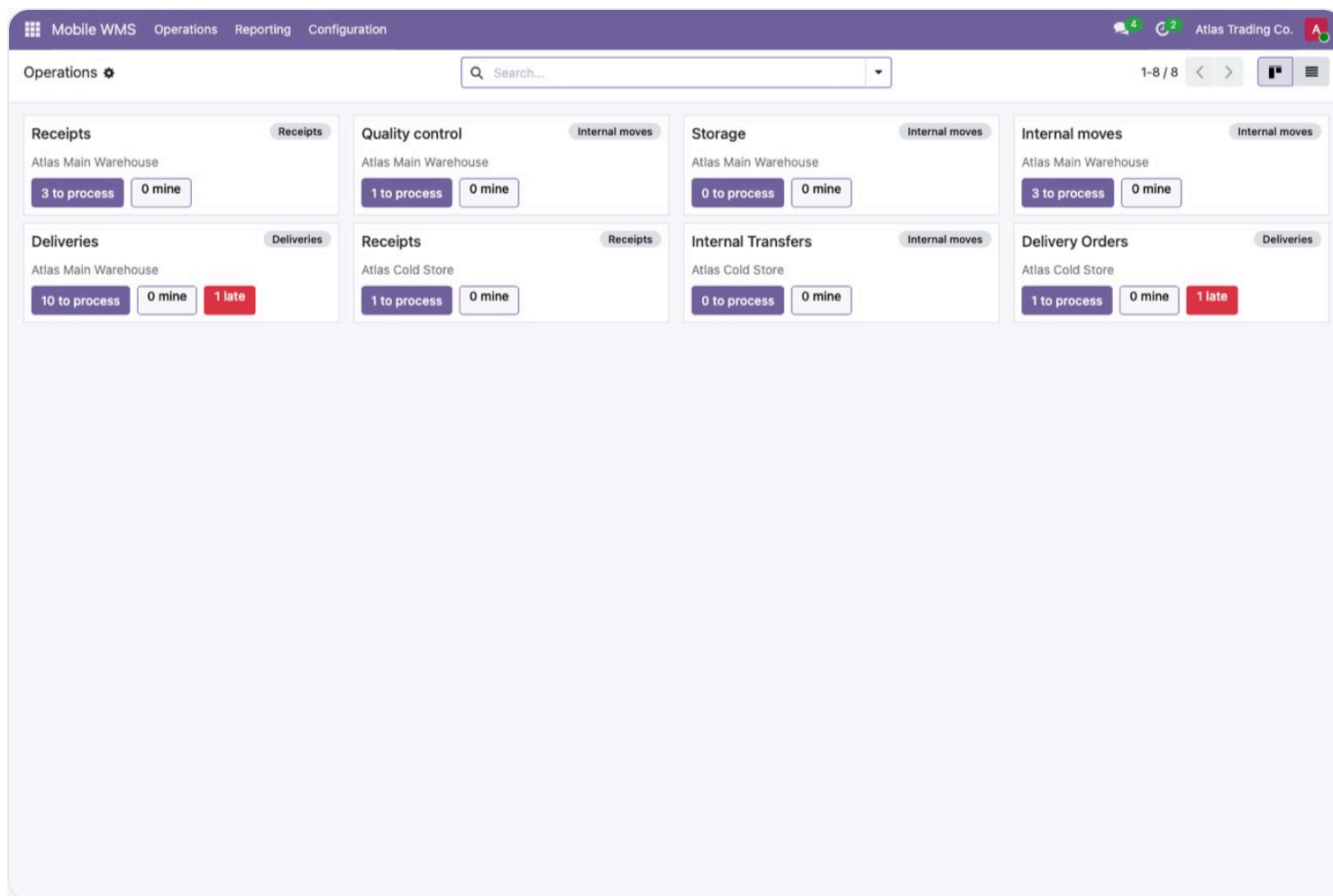


Odoo Mobile WMS & Barcode Scanner

A guide to Odoo Mobile WMS & Barcode Scanner: installing the module and the app, setting up Zebra, Honeywell and Datalogic scanners, bins and check digits, GS1 and barcodes, receipts, deliveries, moves, waves, lots and expiry, cycle counts, offline sync, labels, reports, the API, limits and troubleshooting.

1. Start here

In plain words: This package gives you a **paperless warehouse** on **Odoo Community**: an **Odoo module** (`taqnix_wms_barcode`) with the barcode engine, the operations, the waves, the cycle counts and the audit log, and the **Flutter source code** of a **rugged-scanner app** for Zebra, Honeywell and Datalogic devices, with the camera of a normal phone as a fallback (346 Dart tests and 53 Kotlin tests, run end to end on the iOS simulator against this module; not run on a real rugged scanner, see chapter 31). Operators receive, pick, move and count with the trigger of the scanner. Nobody types in a text field.



The Mobile WMS dashboard: one card per operation type, with what is ready, what is mine and what is late.

Tip: No Odoo Enterprise app is needed. The module replaces the Enterprise Barcode app with `stock` , `barcodes` and the GS1 nomenclature that Odoo Community already ships.

2. What is in the package

Item	What it is
taqnix_wms_barcode.zip	The Odoo 19 module: the barcode resolver (GS1, EAN, UPC, packaging, weight and price barcodes), the scan engine for receipts, deliveries and internal moves, waves along an optimized path, cycle counts, bin check digits and pick sequence, labels, the scan log, the API of the app, Arabic translation, automated tests.
flutter-app-source.zip	The Flutter project of the scanner app for Android (hardware scanners and camera) and iOS (camera), with its tests, README, changelog and docs/ARCHITECTURE.md . See chapters 6 to 8 and 26.
documentation/	This guide in English (index.html) and Arabic (index-ar.html), with PDFs, and API.md for developers.
licensing/	The licence (OPL-1: one purchase = one company, updates included).
preview-images/	Screenshots of the Odoo screens and of the app.
changelog.txt	History of the versions.

3. How it works

Moment	Who	What happens on the scanner	In Odoo
Receiving	Operator	Opens the receipt, scans the staging bin, scans the products (each scan adds one, a case barcode adds the whole case). Lots, serials and expiry dates are asked only when the product needs them.	The scans are kept on the lines of the transfer. Nothing moves yet.
Receiving	Operator	Presses Validate. If something is missing the scanner asks: backorder or not.	The receipt is validated, lots are created, a backorder is created if asked.
Put-away	Operator	Opens the storage transfer, scans the pallet (one scan moves the whole package) and the destination bin, types the two check digits printed on the bin label.	An internal transfer is validated. The quants of the pallet move exactly.
Picking	Operator	Opens a delivery or a wave, follows the path bin by bin (the next bin is shown in large amber letters), scans the bin, the product and the lot.	Move lines get the scanned quantity; a wrong item, an expired lot or too much is refused with a buzzer.
Counting	Operator	Scans the shelf, scans what is on it. A blind count never shows the expected quantity.	Counts keep the expected quantity frozen at the start, the live stock and the drift.
Approval	Manager	Reviews the differences in Odoo and approves.	One inventory adjustment per line, once.

In plain words: The warehouse wins. A movement that already happened on the floor is never refused for a soft rule: if the operator took the goods from another bin, or slightly more than ordered within your tolerance, or syncs two hours late, the server accepts it and raises a *flag* and a to-do for the manager (chapters 12 to 18). Only hard errors are refused: closed transfer, tracked product without lot, serial already in stock, expired lot, over-pick beyond the tolerance.

4. What you need

Item	Details
Odoo	19.0 Community or Enterprise. The module depends on Inventory (<code>stock</code>), <code>product</code> , <code>barcodes</code> , the GS1 nomenclature (<code>barcodes_gs1_nomenclature</code>) and <code>mail</code> . Purchase and Sales are not required; the receipts of a purchase order are used when Purchase is installed.
Locations	Bins as stock locations. Multi-step locations, several warehouses, lots and serial numbers, packages and units of measure with barcodes all work; switch on the Inventory settings you use (the module activates the internal transfer operation type).
Address	A public https address for your Odoo (or your VPN) that the scanners can reach. A self-signed certificate does not work in the app.
Scanners	Android 7.0 or newer. Rugged: Zebra (DataWedge), Honeywell (Data Intent), Datalogic (DecodeWedge), or any scanner that sends an intent or keys. Any Android phone or iPhone (iOS 15.5 or newer) works with the camera. To build the app: Flutter 3.41.
PDF engine	<code>wkhtmltopdf</code> , as for any Odoo, to print the bin and package labels.

5. Install the Odoo module

- 1 Unzip the package. Inside is `taqnix_wms_barcode.zip` ; unzip it to get the folder `taqnix_wms_barcode` .
- 2 Copy the folder into one of the `addons_path` folders of your Odoo server and restart Odoo.
- 3 Log in as an administrator, activate **Developer mode**, open **Apps** and press **Update Apps List**.
- 4 Remove the "Apps" filter, search **Mobile WMS** and press **Install**.

```
# or from the command line
./odoo-bin -c odoo.conf -d YOUR_DATABASE -i taqnix_wms_barcode --stop-after-init
```

In plain words: After the installation the module activates the *Internal Transfers* operation type of every warehouse (Odoo keeps it archived until multi-location is switched on) and selects the default GS1 nomenclature in the settings of each company. The *Mobile WMS* menu appears for users of the group **Mobile WMS / Manager**; the administrator gets it automatically.

6. Install the app on the scanners

In plain words: The app is a standard **Flutter** project (Flutter 3.41 with Dart 3.11, the Android SDK for API 24 or newer which is Android 7.0, and Xcode for iPhone). Package `taqnix_wms_scanner`, application id and bundle id `com.taqnix.wmsscanner`, display name *Taqnix WMS Scanner*. It uses no Firebase and no push service: the only thing to configure is the address of your Odoo.

- 1 Unzip `flutter-app-source.zip` and open a terminal in the folder.
- 2 Fetch the packages: `flutter pub get`. The database layer (Drift) is already generated and included; run `dart run build_runner build --delete-conflicting-outputs` only if you change the tables. After editing the texts of the app, `python3 tool/l10n_src.py && flutter gen-l10n` regenerates the translations.
- 3 Check the project: `flutter analyze` and `flutter test` (the suite is described in chapter 26).
- 4 Run it on a connected Android device or an emulator: `flutter run --dart-define=SITE_URL=https://odoo.example.com`. `SITE_URL` is optional: without it the sign-in screen shows a *Server address* field.
- 5 Build the Android app for the rugged devices: `flutter build apk --release --dart-define=SITE_URL=https://odoo.example.com`. Install the APK by USB, by your MDM (Zebra StageNow, Honeywell Enterprise Provisioner, SOTI...) or from a download link. The release APK is signed with the debug key until you create `android/key.properties` as in the standard Flutter signing guide: do that before you distribute it.
- 6 Build the iPhone app on a Mac with Xcode: `flutter build ios --release --dart-define=SITE_URL=https://odoo.example.com` (iPhones use the camera, see chapter 8).

- 7 Open the app. Sign in with the login and password of a user of the group *Mobile WMS / Operator*. The app downloads the master data (products, packaging barcodes, bins, pallets, lots) and is ready to scan, even without network from then on.
- 8 In *Settings > Scanner* check the scanner profile (chapters 7 and 8).

Permission	Why
<code>INTERNET</code> , <code>ACCESS_NETWORK_STATE</code>	Talk to Odoo and know when the network is back.
<code>CAMERA</code> (optional)	Camera scanning when there is no hardware scanner.
<code>VIBRATE</code>	The vibration of the scan feedback.
<code>WAKE_LOCK</code>	The screen stays on while the app is open.
<code><queries></code>	Declared for the DataWedge, Honeywell and Datalogic packages, so that Android 11 or newer lets the app see them.

Tip: Without a server or a device you can try the app against the small test server of the source: `python3 tool/fake_server.py --port 8765` (Python standard library only, 60 demo products, receipts, a delivery, a wave and two counts, logins `demo / demo` and `manager / manager`), then `flutter run --dart-define=SITE_URL=http://127.0.0.1:8765` . Debug builds can also show a thin *Type a barcode to inject* bar above the scan area (it is visible in the screenshots of this guide); release builds never show it. *Settings > Scanner test* always has one.

Tip: Every device that signs in is listed in *Mobile WMS > Configuration > Devices* with its model and the scanner it uses. A manager can sign a device out from that list.

Mobile WMS Operations Reporting Configuration							Atlas Trading Co.
Devices							1-5 / 5
User	Device	Model	Platform	Scanner	App version	Last Seen	
L Layla Nasser	TC57-0312	Zebra TC57	Android	Zebra DataWedge	1.0.0	Oct 28, 6:09 AM	Sign out
M Mariam Saleh	MEMOR20-0005	Datalogic Memor 20	Android	Datalogic	1.0.0	Oct 28, 5:58 AM	Sign out
O Omar Haddad	TC52-0147	Zebra TC52	Android	Zebra DataWedge	1.0.0	Oct 28, 5:44 AM	Sign out
Y Yusuf Karim	EDA52-0021	Honeywell EDA52	Android	Honeywell	1.0.0	Oct 28, 4:45 AM	Sign out
T Tariq Aziz	PIXEL8-CAM	Pixel 8 (camera)	Android	Camera	1.0.0	Oct 28, 4:13 AM	Sign out

The devices that are signed in: user, device, model, scanner, app version and the last time it was seen.

7. Scanner setup: Zebra DataWedge

In plain words: The app listens to the **broadcast intent** that DataWedge sends after each scan. It does not use the soft keyboard and does not need a text field to be focused. All the scanners go through one *scan bus*: only the screen on top receives a scan, so a scan never lands on a hidden screen.

In plain words: The app sets DataWedge up by itself. With *Settings > Scanner > Manage DataWedge automatically on*, the app creates and maintains a profile called `TaqnixWMS` at start-up through the DataWedge API (`CREATE_PROFILE` then one `SET_CONFIG`); the button *Create DataWedge profile* does the same on demand. The profile has barcode input on, **intent output on** (action `com.taqnix.wms.SCAN` , category `android.intent.category.DEFAULT` ,

delivery *broadcast*), **keystroke output off**, and is associated with `com.taqnix.wmsscanner` for all its activities. The scanner plugin is switched off when the app goes to the background and on again when it comes back, so the trigger does not scan for another app. The action can be changed in *DataWedge output action*.

In plain words: By hand (DataWedge 8 or newer; the way to go when your MDM pushes the profile):

- 1 Open the **DataWedge** app and tap **New profile**; call it `TaqnixWMS` .
- 2 Under **Associated apps** add `com.taqnix.wmsscanner` with activity `*` .
- 3 Under **Barcode input** keep *Enabled* and tick the symbologies you use: EAN-13, EAN-8, UPC-A, UPC-E, Code 128, GS1-128, DataMatrix and QR if you label with them.
- 4 Under **Keystroke output** switch **Enabled off**: the scan must not be typed anywhere.
- 5 Under **Intent output** switch **Enabled on**. Intent action `com.taqnix.wms.SCAN` (or the one you typed in the app), category `android.intent.category.DEFAULT` , **Intent delivery: Broadcast intent**.
- 6 Test: open *Settings > Scanner test* in the app and pull the trigger. The code, the symbology and the raw intent appear on the screen.

In plain words: The app reads the standard DataWedge extras (`com.symbol.datawedge.data_string` and the label type, with the older `com.motorolasolutions.emdk.datawedge` names too). With multi-barcode scanning on, every code of the same read arrives as a scan of its own.

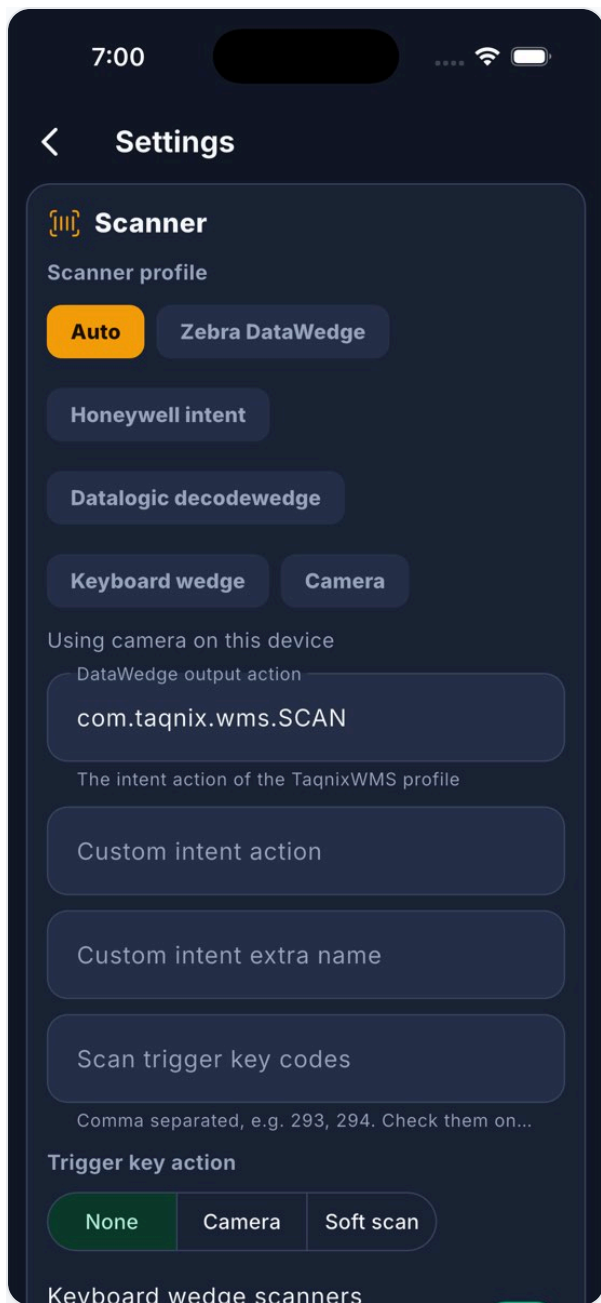
Important: The DataWedge profile, the intents and the `SET_CONFIG` bundle follow the Zebra documentation and are unit tested on the parsing side. They have **not** been run on a real Zebra device by the author (chapter 31). Check the first trigger pull on your own device, and if DataWedge refuses the automatic profile (DataWedge 6.x and 8.x differ), use the manual steps.

8. Scanner setup: Honeywell, Datalogic and others

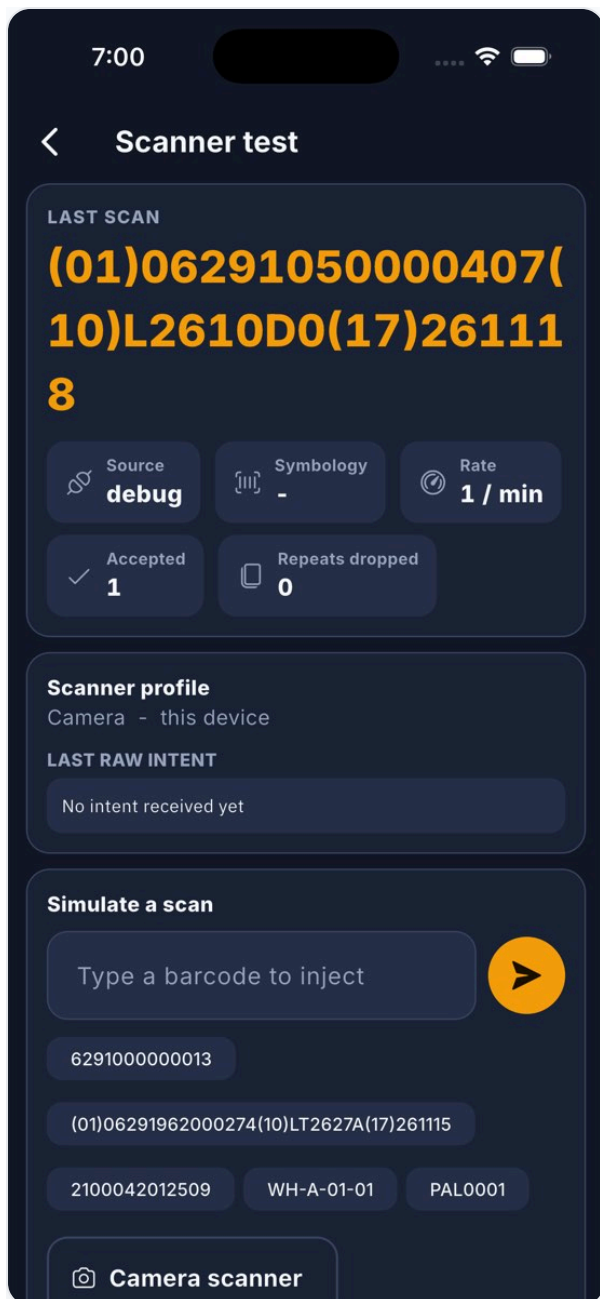
In plain words: In *Settings > Scanner > Scanner profile* you choose *Auto* or a fixed profile. **Auto** reads the manufacturer of the device: Zebra, Motorola Solutions or Symbol use DataWedge; Honeywell or Intermec use the Honeywell intent; Datalogic uses DecodeWedge; anything else uses the camera. Choose a profile by hand if Auto picks the wrong one.

Device	Where	Settings
Honeywell (EDA, CT, CK, Dolphin)	Settings app > Honeywell Settings > Scanning > Internal scanner > Default profile > Data processing settings	Switch Data Intent on (the <i>Intent API</i> ; older images call it <i>Data Collection Service</i>). The app listens to <code>com.honeywell.decode.intent.action.EDIT_DATA</code> and <code>com.honeywell.sample.action.BARCODE_DATA</code> and reads the extras <code>data</code> (or <code>dataBytes</code>), <code>aimId</code> , <code>codeId</code> and <code>charset</code> . Switch <i>Keyboard output</i> off, or leave the wedge on: the key-burst assembler below also works.
Datalogic (Memor, Skorpio)	Datalogic scanner settings > DecodeWedge / Intent Wedge	Switch <i>Intent output</i> on, delivery <i>broadcast</i> , action <code>com.datalogic.decodededge.decode_action</code> , category <code>com.datalogic.decodededge.decode_category</code> . The app reads the string extra <code>com.datalogic.decode.intentwedge.barcode_string</code> (the type is in <code>...barcode_type</code>).
Any other Android scanner	The vendor tool, then <i>Settings</i> > <i>Scanner</i>	Choose <i>Custom intent</i> : type the Custom intent action and the Custom intent extra name that carries the barcode. Any broadcast with a string extra (or a byte array) works.
Keyboard wedge scanners	<i>Settings</i> > <i>Scanner</i> > <i>Keyboard wedge scanners</i>	The scanner types the code and presses Enter. The app collects keys that arrive less than 50 ms apart and end with Enter as one scan, and ignores such bursts inside a text field. It needs no other setting, but it is slower than an intent.
Phones and iPhones	<i>Camera</i>	The camera scanner (1D and 2D codes, continuous mode, torch, a scan window that reads only inside the frame). It is the only scanner on iOS.

In plain words: The physical scan button. Type its Android key codes in *Scan trigger key codes* (comma separated, for example `293, 294`; some Zebra devices use 103 and 104, some Datalogic 291 and 292) and choose what the button does: nothing (the hardware does the scan), open the camera, or start a soft scan through DataWedge. The codes depend on the model: **check them on your device** with *Settings* > *Scanner test*, which logs the trigger key events.



Settings > Scanner: the profile (Auto, Zebra DataWedge, Honeywell intent, Datalogic DecodeWedge, keyboard wedge, camera), the DataWedge action, the custom intent and the trigger keys.



Scanner test: the last scan, its source and symbology, the rate, the repeats that were dropped, the last raw intent, and a bar to simulate a scan.

In plain words: Repeats. A code that arrives twice inside 400 ms is counted once (the *repeats dropped* counter of the test screen shows it).

Important: The Honeywell and Datalogic settings and the trigger key codes follow the vendor documentation. They have **not** been run on a real device by the author, and the Honeywell `codeId` letters are mapped only for a few common symbologies (the AIM id is the reliable source). See chapter 31.

9. First setup in Odoo

- 1 Open *Inventory > Configuration > Operation Types*, open Receipts, Delivery Orders and Internal Transfers and tick **Mobile WMS** on the new tab of the same name (the options are in the table below).
- 2 Give the bins their barcode, aisle, bay, level and check digits (chapter 10) and print the bin labels (chapter 24).
- 3 Create the users of the warehouse team and give them the group *Mobile WMS / Operator*; the supervisors get *Mobile WMS / Manager* (chapter 22).
- 4 Open *Settings > Mobile WMS* and look at the tolerance, the expiry rules, the count thresholds and the barcode prefixes (chapter 23).
- 5 For products with a lot or a serial number, make sure *Tracking* is set; tick *Ask for the expiry date when receiving* on products with a shelf life.
- 6 Print the barcodes of the products and sign in with the app.

Option of the operation type	What it does
Mobile WMS	The operation type appears in the app and the scan engine works on its transfers.
Tile in the app, order	Which tile of the menu shows these transfers (Receipts, Deliveries, Internal moves, Pick, Pack) and in which order.
Scan the source bin / destination bin	Whether the app asks for the bin before the product.
Confirm the bin with its check digits	The operator types the digits printed on the bin label (when the bin has some).
Lot / serial prompt	<i>Ask at the first scan:</i> the scanner asks for the lot as soon as the product is scanned. <i>Ask when validating:</i> the quantity is accepted and the lot is required when the transfer is validated.
Extra tolerance (%)	Quantity above the demand accepted for this type; the larger of this value and the company tolerance applies.
Jump to the next line	The app jumps to the next line when a line is complete.
Backorder policy in the app	<i>Ask, Create a backorder or No backorder, cancel the rest.</i>

Mobile WMS

Operations

Reporting

Configuration

Atlas Trading Co.

New

Operations

Atlas Main Warehouse: Receipts

1 / 1

Operation Type

Receipts

General

Hardware

Mobile WMS

IN THE APP

Mobile WMS ?

Tile in the app ?

Order in the app

Jump to the next line when a line is complete

Backorder policy in the app

☒

Receipts

1

☒

Ask the operator

SCAN FLOW

Scan the destination bin

Confirm the bin with its check digits ?

Lot / serial prompt

Extra tolerance for this type (%) ?

☒

☒

Ask at the first scan

0.00

The Mobile WMS tab of an operation type.

10. Bins, check digits and the pick path

In plain words: Each storage bin is a stock location. Open *Mobile WMS > Configuration > Bins and addresses*: the list is editable, so you can type the barcode, the check digits, the zone, the aisle, the bay and the level of many bins quickly.

Mobile WMS					Operations		Reporting	Configuration	Atlas Trading Co.	
Bins and addresses					Search...		1-70 / 70			
☐	Full Location Name	Warehouse	Barcode	Check digits	Zone	Aisle	Bay	Level	Pick seq...	
☐	WH/Stock/A/A-01-1	Atlas Main Warehouse	WH-A-01-1	97	Fast movers	A	01	1	10	
☐	WH/Stock/A/A-01-2	Atlas Main Warehouse	WH-A-01-2	16	Fast movers	A	01	2	20	
☐	WH/Stock/A/A-01-3	Atlas Main Warehouse	WH-A-01-3	77	Fast movers	A	01	3	30	
☐	WH/Stock/A/A-02-1	Atlas Main Warehouse	WH-A-02-1	26	Fast movers	A	02	1	40	
☐	WH/Stock/A/A-02-2	Atlas Main Warehouse	WH-A-02-2	76	Fast movers	A	02	2	50	
☐	WH/Stock/A/A-02-3	Atlas Main Warehouse	WH-A-02-3	36	Fast movers	A	02	3	60	
☐	WH/Stock/A/A-03-1	Atlas Main Warehouse	WH-A-03-1	61	Fast movers	A	03	1	70	
☐	WH/Stock/A/A-03-2	Atlas Main Warehouse	WH-A-03-2	55	Fast movers	A	03	2	80	
☐	WH/Stock/A/A-03-3	Atlas Main Warehouse	WH-A-03-3	78	Fast movers	A	03	3	90	
☐	WH/Stock/A/A-04-1	Atlas Main Warehouse	WH-A-04-1	48	Fast movers	A	04	1	100	
☐	WH/Stock/A/A-04-2	Atlas Main Warehouse	WH-A-04-2	85	Fast movers	A	04	2	110	
☐	WH/Stock/A/A-04-3	Atlas Main Warehouse	WH-A-04-3	29	Fast movers	A	04	3	120	
☐	WH/Stock/B/B-04-1	Atlas Main Warehouse	WH-B-04-1	90	Fast movers	B	04	1	130	
☐	WH/Stock/B/B-04-2	Atlas Main Warehouse	WH-B-04-2	20	Fast movers	B	04	2	140	
☐	WH/Stock/B/B-04-3	Atlas Main Warehouse	WH-B-04-3	25	Fast movers	B	04	3	150	
☐	WH/Stock/B/B-03-1	Atlas Main Warehouse	WH-B-03-1	50	Fast movers	B	03	1	160	
☐	WH/Stock/B/B-03-2	Atlas Main Warehouse	WH-B-03-2	13	Fast movers	B	03	2	170	
☐	WH/Stock/B/B-03-3	Atlas Main Warehouse	WH-B-03-3	83	Fast movers	B	03	3	180	
☐	WH/Stock/B/B-02-1	Atlas Main Warehouse	WH-B-02-1	63	Fast movers	B	02	1	190	
☐	WH/Stock/B/B-02-2	Atlas Main Warehouse	WH-B-02-2	52	Fast movers	B	02	2	200	
☐	WH/Stock/B/B-02-3	Atlas Main Warehouse	WH-B-02-3	45	Fast movers	B	02	3	210	

Bins and addresses: barcode, check digits, zone, aisle, bay, level and pick sequence in one list.

Field	Use
Barcode	What the label shows (Code 128). Unique per company. Scanned to confirm the bin.
Check digits	2 to 4 digits printed in a box on the label. The operator types them to prove that he or she is standing at the right bin. Generate them with the bulk action <i>Generate check digits</i> or the <i>Generate</i> link on the form (random, never repeated inside the aisle).
Zone	A picking zone such as Cold, Fast movers, Mezzanine. Waves and counts can be limited to a zone.
Aisle, bay, level	The address of the bin. Aisles and bays sort naturally (2 before 10).
Pick sequence	The position of the bin on the path. It is generated from aisle, bay and level in snake order : the first aisle goes from bay 01 up, the next aisle comes back from the last bay, and so on, so the picker never walks back empty. Tick <i>Keep my pick sequence</i> to protect a number you typed.

Mobile WMS Operations Reporting Configuration

Bins and addresses WH/Stock/A-A-01-1

Putaway Rules Products

1/1

Location Name
A-01-1

Parent Location ?
WH/Stock/A

MOBILE WMS

Barcode WH-A-01-1

Check digits ? 97

Zone ? Fast movers

Aisle A

Bay 01

Level 1

Pick sequence ? 10 ☐ Keep my pick sequence ?

ADDITIONAL INFORMATION

Location Type ? Internal

Storage Category

Replenishments ? ☐

CYCLIC COUNTING

Inventory Frequency ? 0 days

Last Inventory ?

Next Expected ?

LOGISTICS

Removal Strategy ? ☐ First In First Out (FIFO)
☐ Last In First Out (LIFO)
☐ Closest Location
☐ Least Packages

A bin with its check digits, zone, address and pick sequence.

Tip: The pick sequence is renumbered when you create a bin or change its aisle, bay or level. Use the bulk action *Generate pick sequence* on a selection to renumber a whole warehouse. Switch the snake off in the settings if you prefer every aisle in the same direction.

11. Barcodes: what the scanner understands

In plain words: Every scan goes through one engine on the server (and the same rules are sent to the app so that it can read offline). The engine answers in a few milliseconds (chapter 27) and tells the app what the barcode is.

Barcode	What the engine does
EAN-13, EAN-8, UPC-A, UPC-E, GTIN-14	Matches the product barcode with the leading zeros of the same GTIN (a UPC-A scanned as EAN-13 is the same product), checks the check digit and expands UPC-E. A wrong check digit is reported as such instead of an unknown product.
Packaging (case, pack)	The barcode of a unit of measure of the product (Odoo 19 packaging) adds the quantity of the pack. A GTIN-14 case that is not registered falls back to the unit product with a warning.
GS1-128 (AI 01, 02, 10, 21, 13, 15, 17, 30, 37, 310x, 320x, 00, 91)	Reads the GTIN, the lot (10) or the serial (21), the expiry (17, day 00 means the end of the month), the best-before date (15), the count (30, 37), the net weight (310x kg, 320x lb converted to the unit of the product) and the SSCC of a pallet (00). Identifiers not in the nomenclature (11, 20, 240, 400, 420, 92 to 99...) are skipped.
Separators	The FNC1 character, # , Alt029 , the]C1 prefix of the scanner, the human readable form (01)...(10)... and spaces between the elements are all accepted. A variable-length lot followed by another element needs a separator, as the GS1 standard requires.
Weight and price barcodes	EAN-13 made of a two digit prefix (21 for weight, 23 for price by default), a 5 digit item code, 5 digits of value and the check digit. The product is found by its barcode with the value digits at zero; the quantity is the weight in the unit of the product, or the price divided by the sales price. A weight (GS1 310x, 320x or a weight barcode) sets a quantity only for a product counted in a weight unit (kg, g, lb...); for a product in units it stays information.
Bins, packages, lots, documents	Bin barcode, package name (a pallet SSCC), lot name, the name or the source document of a transfer (a purchase order number opens its receipt), the name of a wave or of a count.

In plain words: What comes first? When a code can be several things, the context decides. On a receipt a code that is both a product barcode and a lot name is the product. On a delivery, before any bin was scanned, a code that is a bin is the bin. In the menu, documents come first.

Important: A number that is just a number is never read as a GS1 lot: GS1 is only tried when nothing matched exactly, or when the string carries separators, or when it starts with an SSCC or a GTIN.

12. Receipts

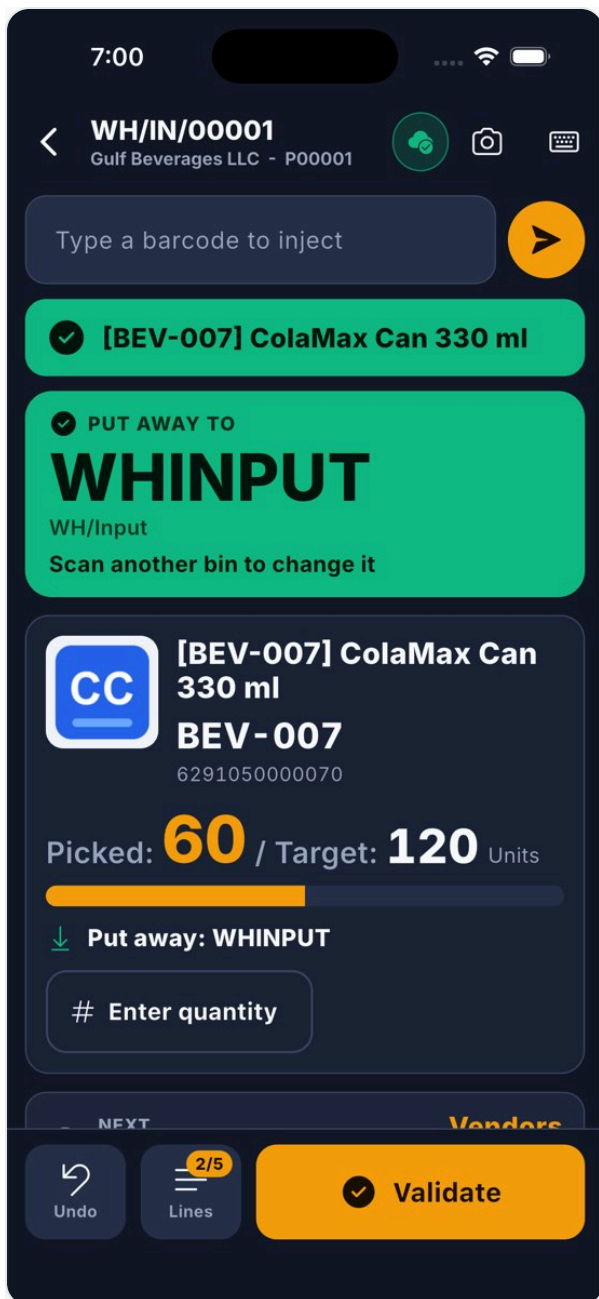
- 1 Open the Receipts tile and choose the transfer (or scan the purchase order number: it opens its receipt).
- 2 Scan the staging bin if your operation type asks for it. For a 3-step reception this is the input location.
- 3 Scan the products. Each scan adds one unit; the quantity keypad of the app changes the amount; a case barcode adds its pack quantity.
- 4 For a lot product the scanner asks for the lot (or reads it from the GS1 barcode). If the product needs an expiry date the scanner asks for it unless AI 17 gave it.
- 5 Press **Validate**. A transfer that is not complete asks whether to create a backorder.

Situation	Answer of the scanner
Product not on the receipt	Error <code>wrong_product</code> , buzzer; nothing is changed.
More than ordered	Refused with <code>over_pick</code> unless the tolerance allows it; within the tolerance it is accepted and flagged.
Serial already received or in stock	<code>serial_duplicate</code> .
Expired lot	<code>lot_expired</code> when <i>Block expired lots</i> is on.
Lot or expiry missing	A prompt (<code>needs_lot</code> , <code>needs_serial</code> , <code>needs_expiry</code>): the scanner opens the entry sheet.

Mobile WMS Operations Reporting Configuration										
Atlas Trading Co.										
New Transfers on the terminal										
<input type="text" value="Ready"/> <input type="text" value="Search..."/>										
1-19 / 19										
Priority	Reference	Contact	Source Document	Operation Type	Scheduled Date	Mobile operator	Scan state	Wave	Status	
☐ ★	WH/OUT/00006	Sunrise Hotel	SO01045	Atlas Main Warehouse: Deliveries	Yesterday		Not started	WAVE/2026/0002	Ready	
☐ ★	WH/OUT/00002	Creek Grocery	SO01041	Atlas Main Warehouse: Deliveries	Today	L Layla Nasser	All scanned	WAVE/2026/0001	Ready	
☐ ☆	CS/OUT/00001	Blue Bay Supermarket	SO01048	Atlas Cold Store: Delivery Orders	Yesterday		Not started		Ready	
☐ ☆	WH/OUT/00004	Gulf Pharmacy	SO01043	Atlas Main Warehouse: Deliveries	Today	L Layla Nasser	Not started	WAVE/2026/0001	Ready	
☐ ☆	WH/IN/00001	Gulf Beverages LLC	P00001	Atlas Main Warehouse: Receipts	Today	O Omar Haddad	Scanning		Ready	
☐ ☆	WH/QC/00001		P00004	Atlas Main Warehouse: Quality control	Today		Not started		Ready	
☐ ☆	WH/INT/00003		Replenishment	Atlas Main Warehouse: Internal moves	Today		Not started		Ready	
☐ ☆	WH/INT/00002		Replenishment	Atlas Main Warehouse: Internal moves	Today		Not started		Ready	
☐ ☆	WH/INT/00001		Replenishment	Atlas Main Warehouse: Internal moves	Today		Not started		Ready	
☐ ☆	WH/OUT/00010	Falcon Office Supplies	SO01099	Atlas Main Warehouse: Deliveries	Today		Not started		Ready	
☐ ☆	WH/OUT/00009	Falcon Office Supplies	SO01049	Atlas Main Warehouse: Deliveries	Today		Not started	WAVE/2026/0002	Ready	
☐ ☆	WH/OUT/00007	Desert Rose Catering	SO01046	Atlas Main Warehouse: Deliveries	Today		Not started	WAVE/2026/0002	Ready	
☐ ☆	WH/OUT/00005	Palm Cafe	SO01044	Atlas Main Warehouse: Deliveries	Tomorrow	L Layla Nasser	Not started	WAVE/2026/0001	Ready	
☐ ☆	WH/OUT/00001	Al Noor Market	SO01040	Atlas Main Warehouse: Deliveries	Tomorrow	L Layla Nasser	Scanning	WAVE/2026/0001	Ready	
☐ ☆	WH/IN/00002	Orient Electronics Trading	P00003	Atlas Main Warehouse: Receipts	Tomorrow	T Tariq Aziz	Scanning		Ready	
☐ ☆	CS/IN/00001	Emirates Dairy Farms	P00002	Atlas Cold Store: Receipts	Tomorrow	Y Yusuf Karim	Scanning		Ready	
☐ ☆	WH/OUT/00008	City Mart Hypermarket	SO01047	Atlas Main Warehouse: Deliveries	Tomorrow		Not started	WAVE/2026/0002	Ready	
☐ ☆	WH/OUT/00003	Marina Mini Mart	SO01042	Atlas Main Warehouse: Deliveries	Tomorrow	L Layla Nasser	Not started	WAVE/2026/0001	Ready	
☐ ☆	WH/IN/00004	Gulf Beverages LLC	P00005	Atlas Main Warehouse: Receipts	In 3 days		Not started		Ready	

The transfers that the terminal can work on, with the operator, the scan state and the wave.

In plain words: In the app a receipt starts with the staging bin (*Put away to* in large letters; scan another bin to change it), then the items; a destination bin can be set per line.



A receipt on the scanner: the bin to put away to, the product with its picture, the quantity received and the target.

13. Deliveries and picking

In plain words: The lines of a delivery come sorted along the pick path (the pick sequence of the bin of each line). The scanner shows the next bin in large amber letters, then the product with its picture and a counter such as *Picked: 12 / Target: 20*.

- 1 Open the delivery. It is claimed by you: nobody else can scan on it (a manager can).

- 2 Scan the bin. If the bin has check digits the app asks for them. Scanning another bin than the suggested one is accepted and flagged `wrong_location`.
- 3 Scan the product (and the lot). The reserved lot is suggested; another lot is accepted and flagged `substitute_lot`.
- 4 When all lines are done the app shows the completion chime. Press Validate.

Mobile WMS

Operations

Reporting

Configuration

Atlas Trading Co.

New

Transfers on the terminal

WH/OUT/00001

Packages

Moves

Scans 76

1/1

Validate

Print

Return

Cancel

Open mobile terminal URL

Draft

Waiting

Ready

Done

Send message

Log note

Activity

☆ WH/OUT/00001

Delivery Address

Al Noor Market

Scheduled Date

Oct 29, 5:16 AM

Operation Type

Atlas Main Warehouse: Deliveries

Product Availability

Available

Source Location

WH/Stock

Source Document

SO01040

Mobile operator

L Layla Nasser

Scan state

Scanning

Wave

WAVE/2026/0001

Last scan

Oct 28, 7:16 AM

Operations

Additional Info

Note

Product	Packages	Demand	Quantity	Unit	
[BEV-002] Aqua Pure Water ...	Available	48.00	48.00	Units	Details
[BEV-007] ColaMax Can 33...	Available	24.00	24.00	Units	Details
[SNK-024] Salted Crisps 15...	Available	24.00	24.00	Units	Details

Add a Product

Move a Pack

Put in Pack

The conversation is empty.

A delivery in progress: the mobile operator, the scan state, the wave and the number of scans in the log.

In plain words: The quantities you scan are kept on the lines (*scanned quantity*) while you work; the reservation of Odoo is not touched. Validation writes the real done quantities, drops what was not picked and releases its reservation.

7:00

<

WH/OUT/00009

Falcon Office Supplies - SO01049

Type a barcode to inject

[BEV-008] ColaMax 1.25 L

BIN CONFIRMED

WH-A-04-1

WH/Stock/A/A-04-1

C1

[BEV-008] ColaMax 1.25 L

BEV-008

6291050000087

Picked: **14** / Target: **24** Units

Enter quantity

NEXT

WH-B-04-2

[BEV-019] Sunny Appl...

0/12

Undo

0/3

Lines

Validate

Picking a delivery: the bin to go to, the product and Picked / Target, then Undo, Lines and Validate.

7:00

<

CS/OUT/00001

Blue Bay Supermarket - SO01048

Type a barcode to inject

Lot or serial needed for
[DRY-040] Fresh Milk 1 L

BIN CONFIRMED

CS-E-03-1

CS/Stock/E/E-03-1

[DRY-040] Fresh Milk 1 L

Lot number

[DRY-040] Fresh Milk 1 L

Lot number

L2610D0

Suggested lots

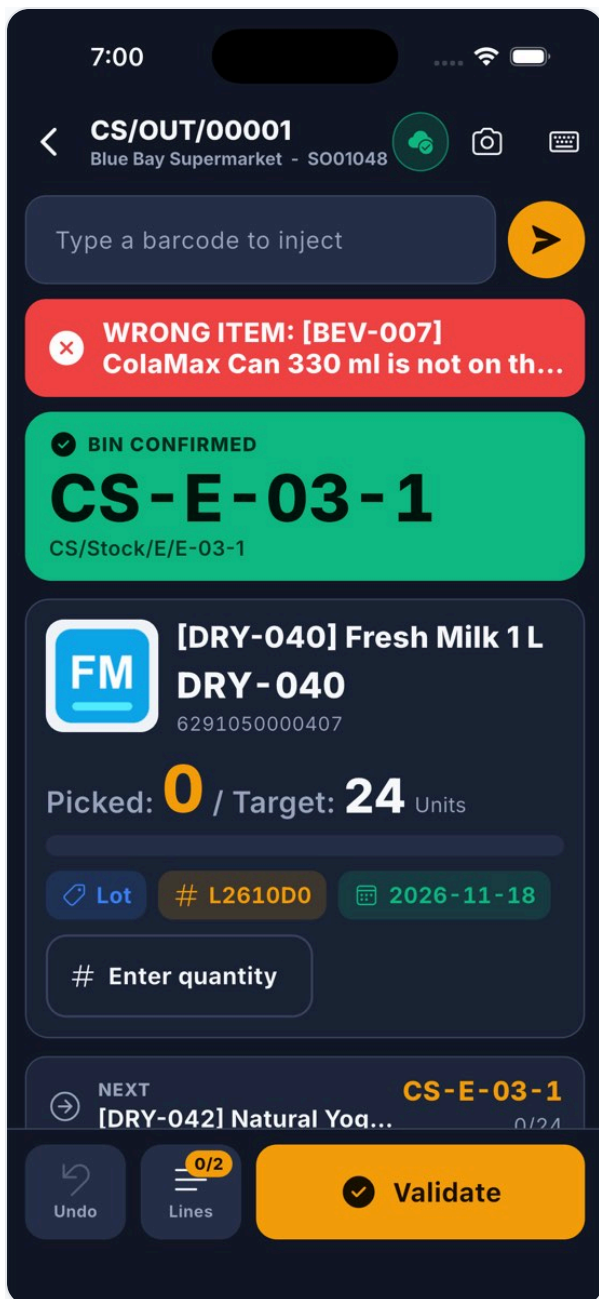
L2610D0 2026-11-18

Expiry date: 2026-11-18

Cancel

Confirm

A lot-tracked product: the lot sheet lists the lots in the bin, the earliest expiry first; a GS1 barcode fills it by itself.



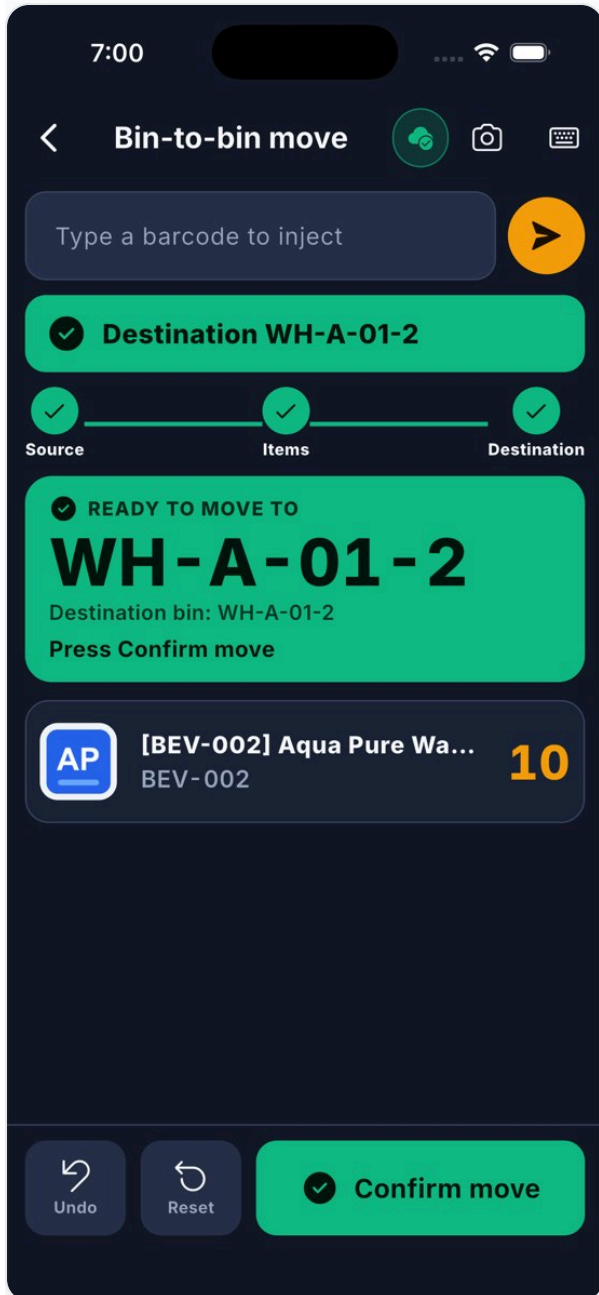
A product that is not on the transfer: a red banner and the buzzer, and nothing is changed.

14. Internal moves and pallets

- 1 Open an internal transfer, or use *Free move* in the app to move stock without an order.
- 2 Scan the source bin, then the product (or the pallet), then the destination bin and its check digits.
- 3 Scanning a pallet (a package) moves its whole content in one scan: every product, lot and quantity of the package. A pallet with a product that is not on the transfer is refused as a whole.

4 Validate. The quants move exactly and the pallet keeps its identity in the destination bin.

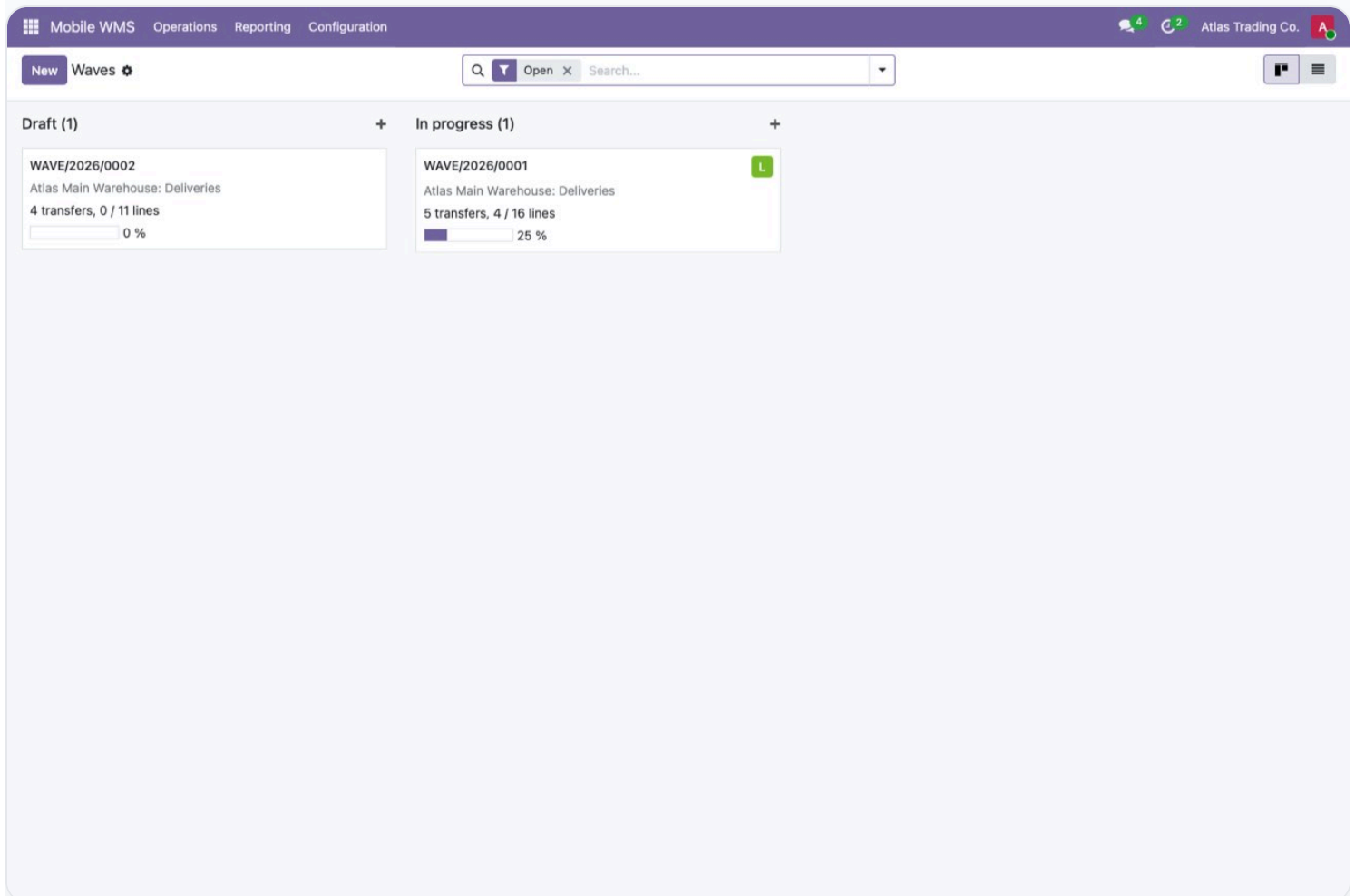
In plain words: A free move creates and validates an internal transfer for you. If you move more than the books show, the move is accepted (it happened), the stock of the bin becomes negative and the manager gets a to-do.



A bin-to-bin move: source, items and destination are done, and the move waits for Confirm move.

15. Wave picking along the path

In plain words: A wave groups several deliveries (or internal transfers of the same operation type) so that one picker serves them in one walk, along the optimized path, and the orders are packed afterwards.



Waves as a Kanban board by state.

- 1 Open *Mobile WMS* > *Transfers*, select the ready transfers of one operation type and choose **Action > Create a wave**. Or create a wave and press **Add transfers**: the wizard proposes the ready transfers of the same type that are not in another wave.
- 2 Press **Start**. The picker (or you, if you start it) claims the wave and all its transfers.
- 3 The picker opens the wave in the app. The stops are the bins sorted by pick sequence, each with the lines of all the orders; the lines of the same product in the same bin are merged for display.
- 4 A scan of a unit goes to the first order, by priority, deadline and date, that still needs the product; a larger quantity is spread over the orders in the same order.

5

Press **Validate**: the wizard asks what to do with unfinished transfers (backorder or cancel the rest). Transfers without any scan leave the wave and stay open.

Mobile WMS
Operations
Reporting
Configuration
Atlas Trading Co.

New
Waves
WAVE/2026/0001
1/1

Add transfers
Validate
Cancel
Draft
In progress
Done
Send message
Log note
Activity

WAVE/2026/0001

Operation type Atlas Main Warehouse: Deliveries

Scheduled Date Oct 28, 7:16 AM

Picker L Layla Nasser

Lines picked 4

Zone ? Fast movers

Lines 16

Progress (%) 25 %

Started At Oct 28, 7:16 AM

Transfers
Pick path
Notes

Seq.	Bin	Product	Lot/Serial Num...	Qty	Scanned	Transfer	
20	A-01-2	[SNK-030] Chocolate Wafer Bar		24.00	0.00	WH/OUT/00...	
30	A-01-3	[PCR-072] Shampoo 250 ml	L2610D0	24.00	0.00	WH/OUT/00...	
30	A-01-3	[PCR-075] Toothpaste 75 ml	L2610C0	24.00	0.00	WH/OUT/00...	
30	A-01-3	[BEV-007] ColaMax Can 330 ml		24.00	0.00	WH/OUT/00...	
50	A-02-2	[SNK-023] Salted Crisps 40 g		48.00	0.00	WH/OUT/00...	
70	A-03-1	[BEV-014] Energy Spark 250 ml		24.00	0.00	WH/OUT/00...	
100	A-04-1	[BEV-003] Aqua Pure Water 1.5 L		36.00	36.00	WH/OUT/00...	
120	A-04-3	[HHL-063] Dishwash Liquid 1.5 L		12.00	12.00	WH/OUT/00...	
130	B-04-1	[BEV-002] Aqua Pure Water 500 ml		48.00	48.00	WH/OUT/00...	
130	B-04-1	[STA-085] Sticky Notes Pack of 12		12.00	0.00	WH/OUT/00...	
130	B-04-1	[SNK-026] Cheese Puffs 40 g		36.00	0.00	WH/OUT/00...	
140	B-04-2	[STA-081] Ballpoint Pens Blue Box of 50		10.00	0.00	WH/OUT/00...	
160	B-03-1	[HHL-065] Laundry Powder 5 kg		6.00	6.00	WH/OUT/00...	
200	B-02-2	[PCR-074] Hand Soap 300 ml	L2610B0	12.00	0.00	WH/OUT/00...	
200	B-02-2	[SNK-024] Salted Crisps 150 g		24.00	0.00	WH/OUT/00...	
230	B-01-2	[STA-080] A4 Copy Paper Ream 500		40.00	0.00	WH/OUT/00...	

The conversation is empty.

The pick path of a wave: bins in snake order with the lines to pick and the scanned quantities.

Mobile WMS Operations Reporting Configuration Atlas Trading Co.

New Waves WAVE/2026/0001 1/1

Add transfers Validate Cancel Draft In progress Done Send message Log note Activity

WAVE/2026/0001

Operation type Atlas Main Warehouse: Deliveries Scheduled Date Oct 28, 7:16 AM

Picker L Layla Nasser Lines picked 4

Zone ? Fast movers Lines 16

Transfers Pick path Not

Seq.	Bin	Product	Qty	Unit	Status
20	A-01-2	[SNK-03]			
30	A-01-3	[PCR-07]			
30	A-01-3	[PCR-07]			
30	A-01-3	[BEV-00]			
50	A-02-2	[SNK-02]			
70	A-03-1	[BEV-014] Energy Spark 250 ml	24.00	0.00	WH/OUT/00...
100	A-04-1	[BEV-003] Aqua Pure Water 1.5 L	36.00	36.00	WH/OUT/00...
120	A-04-3	[HHL-063] Dishwash Liquid 1.5 L	12.00	12.00	WH/OUT/00...
130	B-04-1	[BEV-002] Aqua Pure Water 500 ml	48.00	48.00	WH/OUT/00...
130	B-04-1	[STA-085] Sticky Notes Pack of 12	12.00	0.00	WH/OUT/00...
130	B-04-1	[SNK-026] Cheese Puffs 40 g	36.00	0.00	WH/OUT/00...
140	B-04-2	[STA-081] Ballpoint Pens Blue Box of 50	10.00	0.00	WH/OUT/00...
160	B-03-1	[HHL-065] Laundry Powder 5 kg	6.00	6.00	WH/OUT/00...

Validate the wave

Picked Count 2

Open Count 3

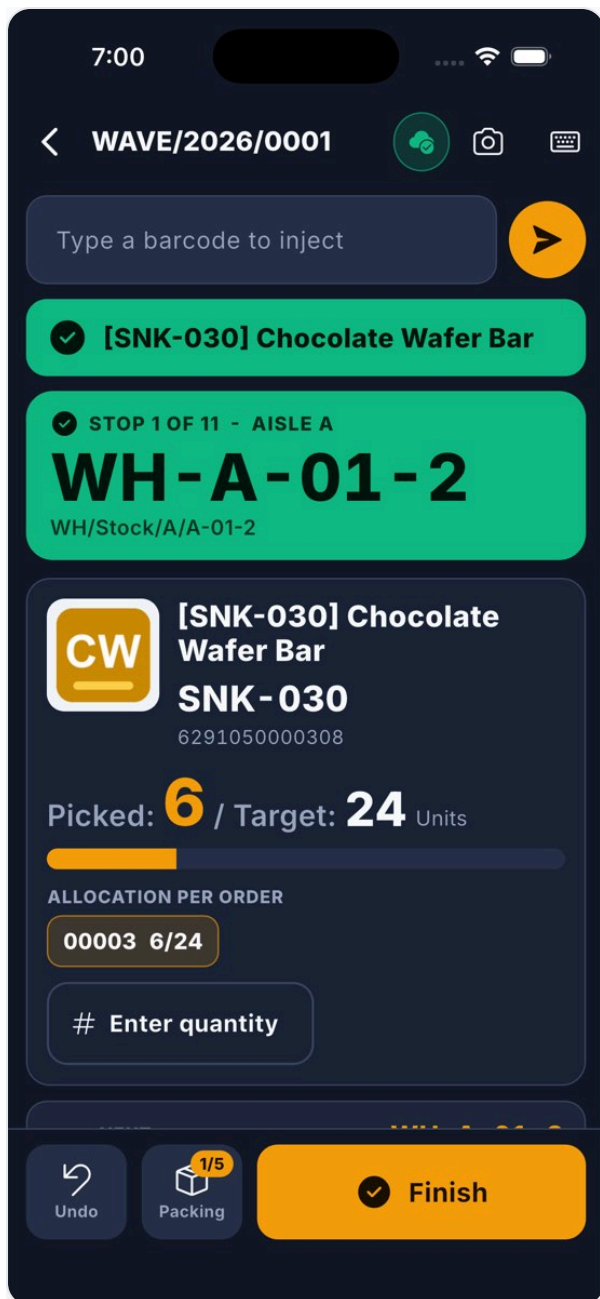
Unfinished transfers ☒ Create a backorder for what was not picked
☐ No backorder: cancel what was not picked

Transfers without any scanned quantity leave the wave and stay open. All the other transfers are validated together: if one of them fails, none is validated.

Validate the wave Cancel

Validating a wave: how many transfers are picked, and what to do with the unfinished ones.

Important: The validation of a wave is all or nothing. If one transfer cannot be validated (for example a lot is missing) none is, and the message names the transfer.



A wave on the scanner: stop 1 of 11, the bin in large letters, the product and its allocation per order, and the packing counter.

16. Lots, serial numbers and expiry

Rule	Detail
Lots	A lot is created when the receipt is validated (name from the scan, expiry from the scan). An existing lot is reused. A lot of a delivery must exist; an unknown lot is refused (<code>unknown_lot</code>).
Serial numbers	One unit per serial. The same serial twice on a transfer, or a serial that is already in stock when it is received, is refused (<code>serial_duplicate</code>).
Expiry date	Stored on the lot. When the Odoo <i>Expiration Dates</i> feature is installed its date is used and kept in step; otherwise the module keeps the date in the field <i>Expiry date (WMS)</i> of the lot.
Ask for the expiry on receipt	Tick it on the product (lot tracked products). The scanner then asks for a date for every new lot unless the GS1 barcode has it.
Expired	With <i>Block expired lots</i> a delivery or a receipt of an expired lot is refused (<code>lot_expired</code>); an internal move of an expired lot is accepted and flagged <code>expired_lot</code> , so that expired goods can go to quarantine.
Expiring soon	Lots that expire within the number of days of the settings are accepted with the flag <code>expiring_soon</code> .
Suggested lots	The app shows up to five lots in the bin, the earliest expiry first.

17. Packages and pallets

In plain words: A package is an Odoo *package* (in Odoo 19 the model is `stock.package`); its name is what the label shows. A pallet label with a valid SSCC can be scanned as `00` + 18 digits or as the plain name.

Where	What a package scan does
Receipt, package unknown	The package is created with the SSCC as its name and becomes the destination package of the products scanned next.
Receipt, package already in stock	Refused (<code>package_in_stock</code>): the same pallet cannot arrive twice.
Delivery or internal move	All the content of the package is scanned at once; the package leaves with its goods (deliveries) or moves to the destination bin keeping its name (internal moves).



Package labels with the barcode of the SSCC, the content and the location.

18. Backorders and tolerance

Policy	What happens to what was not picked
Create a backorder	A new transfer for the remaining quantity, with its reservation. The scanned quantities of the first transfer are validated.
No backorder	The remaining quantity is cancelled and its reservation released.
Ask	The server answers <code>needs_backorder_decision</code> with the remaining quantities and changes nothing (the scans are kept). The operator chooses and validates again.

In plain words: The policy comes from the operation type, and the app can send another one. Validation is atomic: if anything fails (tracking, destination, tolerance) nothing is written. Quantities above the demand are accepted up to the larger of the tolerance of the company and the one of the operation type, with the flag `over_pick` ; above that, `over_pick` is an error.

19. Cycle counts

In plain words: Count a bin, an aisle, a zone, or whatever the operator finds (ad hoc). A **blind** count never shows the expected quantity to the counter, in the app or in Odoo. A guided count does.

Mobile WMS

Operations

Reporting

Configuration

4

2

Atlas Trading Co.

New

Cycle counts

Q

Search...

1-4 / 4

<

>

☐	Name	Scope	Location	Zone	Blind count	Counter	Lines	Lines co...	Differen...	State	
☐	CNT/2026/0004	A bin or an aisle	WH/Stock/C		☐	M Mariam Saleh	38	38	2	Done	
☐	CNT/2026/0003	A bin or an aisle	WH/Stock/B		☒	Y Yusuf Karim	87	87	8	Review	
☐	CNT/2026/0002	A bin or an aisle	WH/Stock/A		☒	L Layla Nasser	65	39	0	Counting	
☐	CNT/2026/0001	A zone		Bulk	☒		0	0	0	Draft	

The cycle counts and their state.

- 1 Create a count (*Mobile WMS > Cycle counts*) with its scope and press **Start counting**, or let the operator create one in the app. At the start the expected quantity of every product, lot and package in the scope is frozen.
- 2 The operator scans a shelf, then the items, and types or scans the quantity. An item that was not expected is added with the quantity the books had at that moment.
- 3 **Submit for review.** The items that the books expected in a bin that was counted, but that nobody scanned, are counted as zero (*missing*). Bins that were never touched are not counted at all.
- 4 A manager reviews the differences. The *Drift* column shows what entered or left the bin since the start, so a receipt that arrived during the count is not mistaken for a loss. The manager can correct a quantity or ask for a recount.
- 5 **Approve and book.** One inventory adjustment per line, of the difference found by the counter (counted minus expected), so the movements made in between are kept. The count becomes Done and cannot be applied twice.

Mobile WMS

Operations

Reporting

Configuration

4

2

Atlas Trading Co.

New

Cycle counts

CNT/2026/0003

1 / 1

Approve and book

Ask for a recount

Cancel

Draft

Counting

Review

Done

Send message

Log note

Activity

CNT/2026/0003

ScopeA bin or an aisle

LocationWH/Stock/B

Blind count

CounterYusuf Karim

Submitted byYusuf Karim

Number of lines87

Lines counted87

Differences8

Discrepancies

All lines

Notes

Location	Product	Lot	Expected	Counted	Difference	Drift	M...
WH/Stock/B/B-0...	[BEV-010] LemonUp Can 330 ml		300.00	302.00	2.00	0.00	
WH/Stock/B/B-0...	[BEV-004] Aqua Pure Water 5 L		180.00	177.00	-3.00	0.00	
WH/Stock/B/B-0...	[BEV-008] ColaMax 1.25 L		96.00	0.00	-96.00	0.00	
WH/Stock/B/B-0...	[SNK-029] Roasted Almonds 400 g		240.00	0.00	-240.00	0.00	
WH/Stock/B/B-0...	[BEV-018] Sunny Apple Juice 200 ml		240.00	239.00	-1.00	0.00	
WH/Stock/B/B-0...	[HHL-065] Laundry Powder 5 kg		60.00	61.00	1.00	0.00	
WH/Stock/B/B-0...	[SNK-024] Salted Crisps 150 g		120.00	125.00	5.00	0.00	
WH/Stock/B/B-0...	[SNK-032] Date Bar 45 g		60.00	58.00	-2.00	0.00	
			-334.00				

Planned Activities

Today: "Review the count CNT/2026/0003" for Atlas Admin

Mark Done

Edit

Cancel

The conversation is empty.

A count under review: expected, counted, difference and drift per line; red lines are above the thresholds.

Mobile WMS Operations Reporting Configuration Atlas Trading Co.									
Count lines									
Differences									
Location	Product	Lot	Expected	Counted	Difference	Drift	Missing	Applied	
WH/Stock/B/B-01-2	[BEV-010] LemonUp Can 330 ml		300.00	302.00	2.00	0.00			
WH/Stock/B/B-01-3	[BEV-004] Aqua Pure Water 5 L		180.00	177.00	-3.00	0.00			
WH/Stock/B/B-01-3	[BEV-008] ColaMax 1.25 L		96.00	0.00	-96.00	0.00			
WH/Stock/B/B-01-3	[SNK-029] Roasted Almonds 400 g		240.00	0.00	-240.00	0.00			
WH/Stock/B/B-02-2	[BEV-018] Sunny Apple Juice 200 ml		240.00	239.00	-1.00	0.00			
WH/Stock/B/B-03-1	[HHL-065] Laundry Powder 5 kg		60.00	61.00	1.00	0.00			
WH/Stock/B/B-03-1	[SNK-024] Salted Crisps 150 g		120.00	125.00	5.00	0.00			
WH/Stock/B/B-04-3	[SNK-032] Date Bar 45 g		60.00	58.00	-2.00	0.00			
WH/Stock/C/C-01-2	[BEV-016] Sunny Orange Juice 200 ml		120.00	124.00	4.00	4.00			
WH/Stock/C/C-01-3	[BEV-007] ColaMax Can 330 ml		180.00	179.00	-1.00	-1.00			
					-331.00				

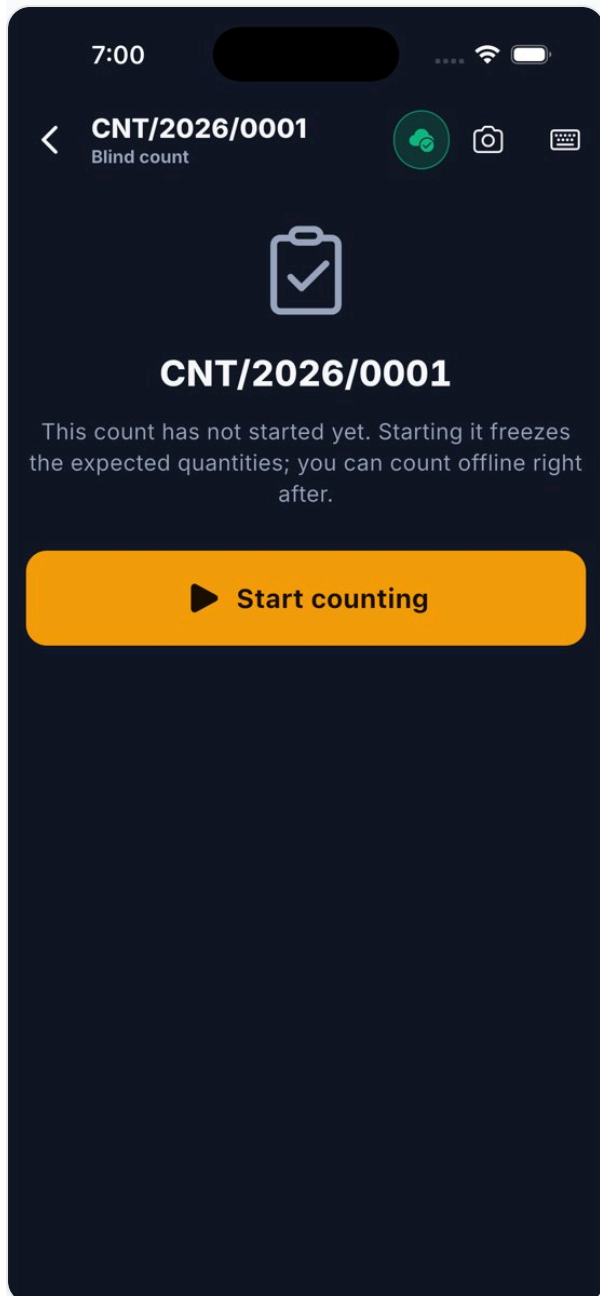
All the count differences in one list, for the analysis.

Important: The manager who approves must not be the person who counted, submitted or corrected the count, unless you allow it in the settings; the counter of a started count cannot be changed. Operators cannot approve, and cannot cancel a count that is under review.

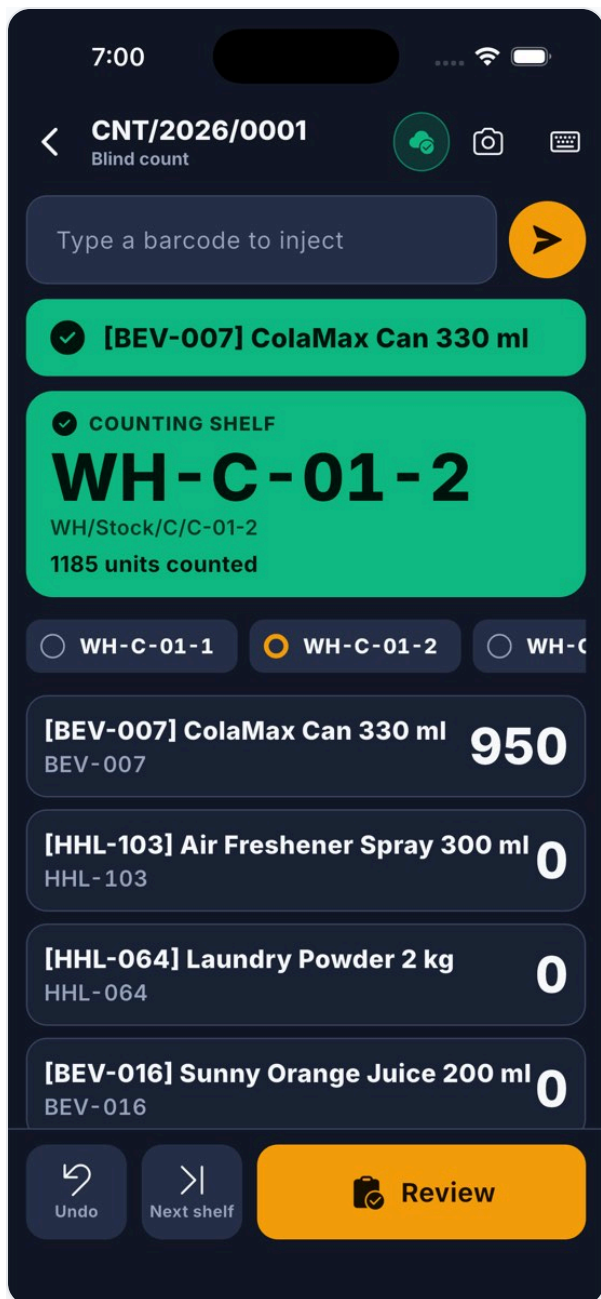
In plain words: Blind means blind. With the company setting *Operators must count blind* (on by default) every count an operator creates in the app is blind, whatever the app sends. The operator receives no expected quantity and no differences when he or she submits (only the number of lines counted): the differences go to the managers. While an operator has a blind count open, the lot and pallet quantities of the stock are also hidden from the scan answers of that operator.

In plain words: In the app. A count that is still a draft shows a big *Start counting* button: starting it freezes the expected quantities (it works offline too: the start is queued, and scans are refused until it has started). The operator scans a shelf (the shelves of the count are chips at the top),

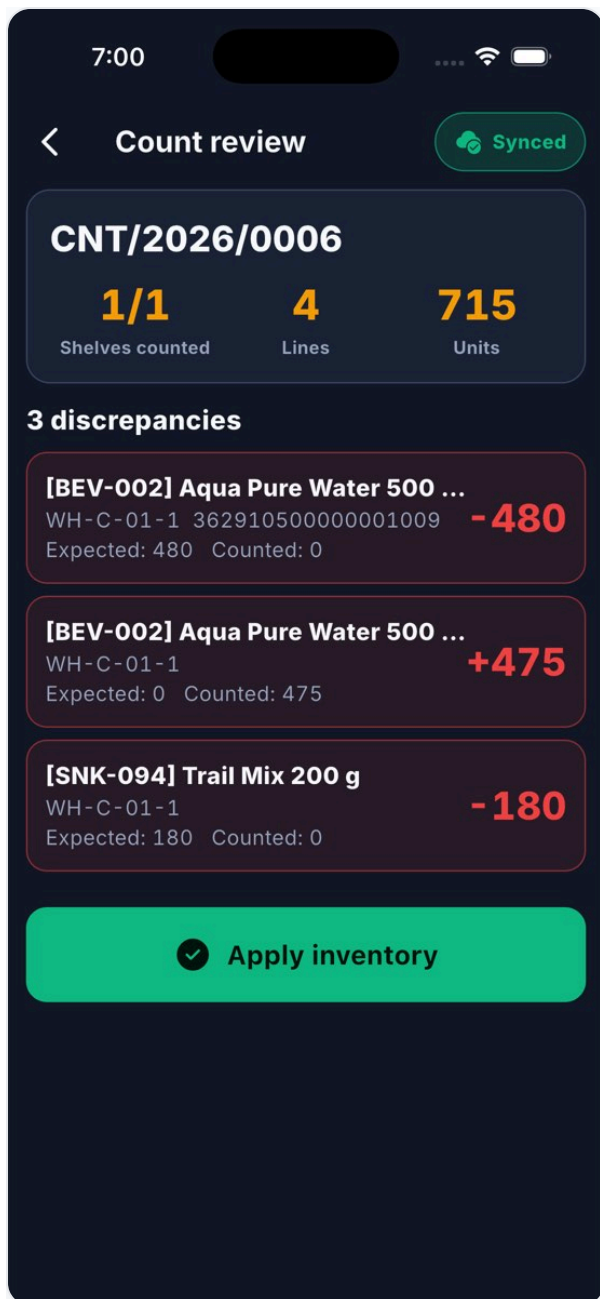
then the items; every scan adds one unit and the keypad changes the quantity. An operator always counts blind: he or she sees no expected quantity, only the counted quantities. *Next shelf* moves on and *Review* ends the count. The operator then waits for the supervisor. A supervisor sees the *Count review*: shelves counted, lines and units, and the discrepancies with expected and counted quantities, and presses *Apply inventory* (the books and the discrepancies come from the server, and only for managers).



A count that has not started: Start counting freezes the expected quantities.



Counting a shelf blind: the shelf, the units counted, and the items scanned so far with their quantities; no expected quantity.



The review of a count for a manager: the discrepancies with expected and counted quantities, and Apply inventory. The first two lines are the same product: the pallet line counted as zero and the packs scanned on a separate line (the limit of chapter 31).

20. Offline behaviour and synchronization

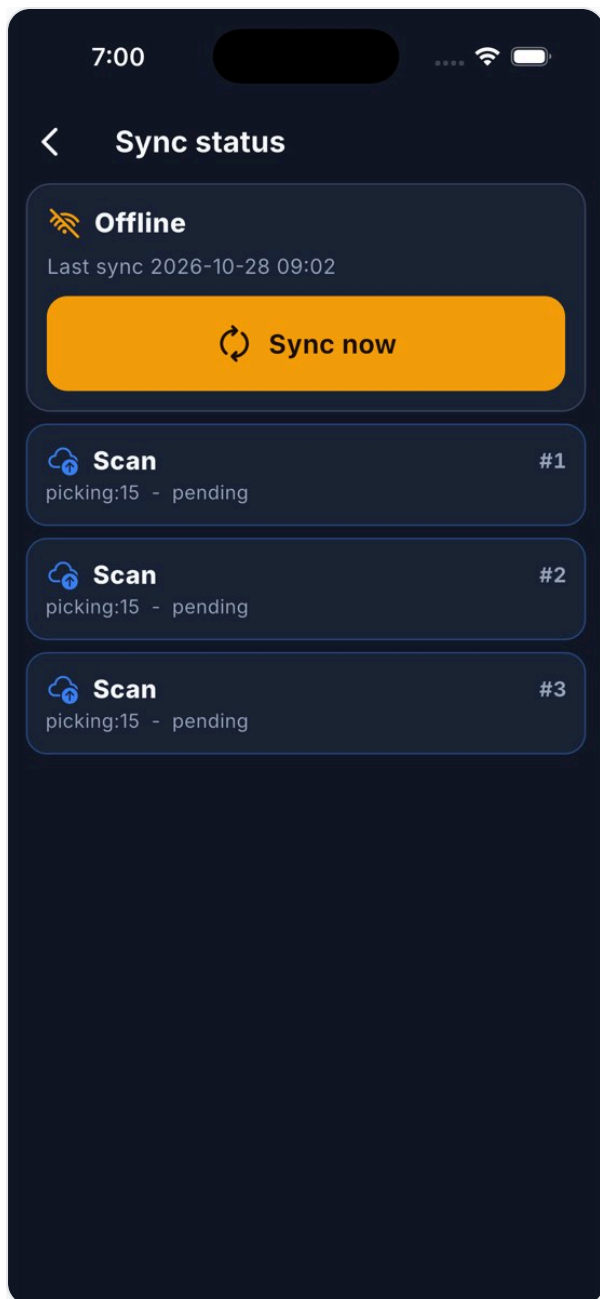
In plain words: The app keeps what it needs on the device: the GS1 rules, the products, the bins with their check digits, the packages and the lots in stock (the *master data*, loaded in pages and

refreshed incrementally), and the open transfers you opened. Scans are checked locally first, so the beep is immediate.

Rule	Detail
Operations queue	Every change (scan, line update, validation, free move, wave scan, count) is stored with a unique <code>uuid</code> and a <code>local_seq</code> that never goes back, and sent with <i>POST /sync</i> in batches of up to 100.
Idempotent	The server stores each applied operation. Sending it again answers <code>duplicate</code> and changes nothing, so a lost answer never double-counts.
Rejected operations	A refused operation is answered <code>rejected</code> with the code. If the data is fixed and the operation is sent again, it is run again.
Late operations	An operation older than 15 minutes when it arrives is applied (it happened) and flagged <code>late_sync</code> with a to-do for the manager.
Several devices	An operation id that belongs to another user is refused.
Changes	The app asks <i>GET /changes</i> for the transfers, waves and counts that changed since its last visit.

In plain words: On the device. The master data (products and their packaging barcodes, bins, pallets and lots) is cached in a local SQLite database and indexed in memory, and the GS1 rules come from your server. Every scan is resolved on the phone by a pure Dart resolver and GS1 parser, the screen repaints in the same frame and the beep is immediate. The server *POST /resolve* is only the fallback for a code the device does not know, and its answer is learned by the index. Each scan then writes the cached document *and* an operation of the outbox in one database transaction, so an app that is killed at that moment loses nothing.

The outbox of the app	Detail
Order	Operations are sent in the order of an unchangeable <code>local_seq</code> (it never goes back, a deleted row never gives its number again). An operation that waits for its retry time blocks the later ones.
Sending	A debounce of 2 seconds after a scan, batches of at most 100 operations through <i>POST /sync</i> . After a refusal of the batch (HTTP 400 or 413) the app narrows the batch down to find the operation that is refused.
Answers	<code>applied</code> and <code>duplicate</code> remove the operation; <code>rejected</code> parks it as an issue with its code; a network error retries with a back-off and jitter. An operation that was in flight when the app crashed goes back to pending.
Restart	The queue survives the restart of the app and of the phone.
Documents	While operations are unsynced, a refresh from the server never overwrites the counts of the operator. A transfer you validated disappears from the open lists at once and comes back if the server rejects the validation.
Counts	Count quantities are absolute, so a resend can never count twice.



The Sync status screen: queued operations, Sync now, and the retry and discard of issues.

In plain words: The *Sync status* screen is reached from the cloud icon at the top of every screen. A refused scan (for example a wrong item) is not counted as sent: it is shown as an issue that the operator can retry or discard.

21. Sounds and vibration

Verdict	Sound	Vibration
Good scan (<code>ok</code>)	High chime	One short pulse
Error (<code>error</code>): wrong product or bin, expired lot, over-pick	Deep buzzer	Two long pulses
Operation complete (<code>complete</code> : line, stop or document done)	Ascending double tone	Short, short, long
Information (<code>info</code>)	Soft tick	None

In plain words: The sounds are short original synthesized files of the app (`python3 tool/make_sounds.py` regenerates them), preloaded at start and played in low-latency mode (Android `SoundPool` , iOS `AVAudioPlayer`). In *Settings* each operator can switch the sound, the vibration, and a **loud mode** (full volume and longer vibration for noisy docks). The company settings *Scan sounds on the devices* and *Vibration on the devices* are only the defaults that the server sends.

Important: Audio latency, the strength of the vibration and the battery drain of a screen kept on were not measured on a real device (chapter 31).

22. Roles and access

Group	Can do
Mobile WMS / Operator	Signs in to the app, receives, picks, moves and counts on the operation types with Mobile WMS enabled, works on the transfers it claimed, sees the free waves and counts and its own. Has no access to the Odoo stock screens through this group.
Mobile WMS / Manager	Everything of the operator, plus: create and start waves and counts, approve counts, take over a transfer, read the scan log, the devices and the synchronized operations, and change the bins and the operation types. Includes the Inventory administrator rights, multi-location, lots and packages.

Rule	Detail
Claim	The first operator who scans a transfer claims it. Others (not managers) get claimed . A manager who helps does not take the transfer over. Release it from the app or set another operator on the form.
Companies	Every model has a company rule; the app works in the company of the user.
Field protection	The expected quantity of a count is readable by managers only; the state of waves and counts, and who approved, cannot be written through RPC: only the buttons change them.

23. Settings

Mobile WMSOperationsReportingConfiguration

Atlas Trading Co.

SaveDiscardSettings

Search...

General Settings

Sales

Purchase

Inventory

Invoicing

Mobile WMS

Barcodes

GS1 and embedded barcodes

GS1-128 strings (GTIN, lot, serial, expiry, weight, SSCC) are read with the rules of the nomenclature. Weight and price barcodes are EAN-13 codes with a two digit prefix, a 5 digit item code, 5 digits of value and the check digit.

Read GS1 barcodes

GS1 nomenclature

Default GS1 Nomen... →

Weight barcode prefix

21

Price barcode prefix

23

Picking and receiving

Tolerance and expiry

Quantity above the demand that the terminal accepts and flags for the manager (the larger of this value and the one of the operation type applies). Expired lots can be blocked on deliveries and receipts.

Over-pick / over-receipt tolerance (%)

2.00

Block expired lots

Warn when a lot expires within (days)

30

Pick path

Number the bins by aisle, bay and level. With the snake path every other aisle is walked in the opposite direction.

Snake pick path (alternate the direction in every aisle)

Cycle counts

Thresholds and approval

Differences above one of the thresholds are shown in red. The manager who approves a count must not be the person who counted it, unless you allow it here.

Count difference flagged above (%)

5.00

Count difference flagged above (units)

10.00

Operators must count blind

A manager may approve the count he or she made

Settings > Mobile WMS.

Setting	Meaning
Read GS1 barcodes, GS1 nomenclature	Switch GS1 on or off; choose another GS1 nomenclature.
Weight and price barcode prefix	Two digits each, or empty to switch off.
Over-pick / over-receipt tolerance (%)	Quantity above the demand that is accepted and flagged. 0 means none.
Block expired lots, warn when a lot expires within (days)	Expiry rules of chapter 16.
Snake pick path	Alternate the direction in every aisle.
Count difference flagged above (% and units)	A difference is red when it exceeds one of them.
Operators must count blind	On by default: an operator cannot create a guided count, and receives no expected quantity or difference.
A manager may approve the count he or she made	Off by default.
Scan sounds, vibration	Defaults for the devices.
App refresh interval, operations per sync call	How often the app asks for changes (10 s to one day) and the size of a batch (1 to 100).
Keep the scan log for (days)	Old rows are deleted every night; 0 keeps everything.

24. Labels

In plain words: Select bins in *Bins and addresses*, choose **Print > Bin labels (with check digits)** and print on A4 paper (two labels per row, 54 mm high). Each label has the name, the

address, the zone, a Code 128 barcode of the bin barcode and the check digits in a box. *Package labels* (SSCC) print from a package.

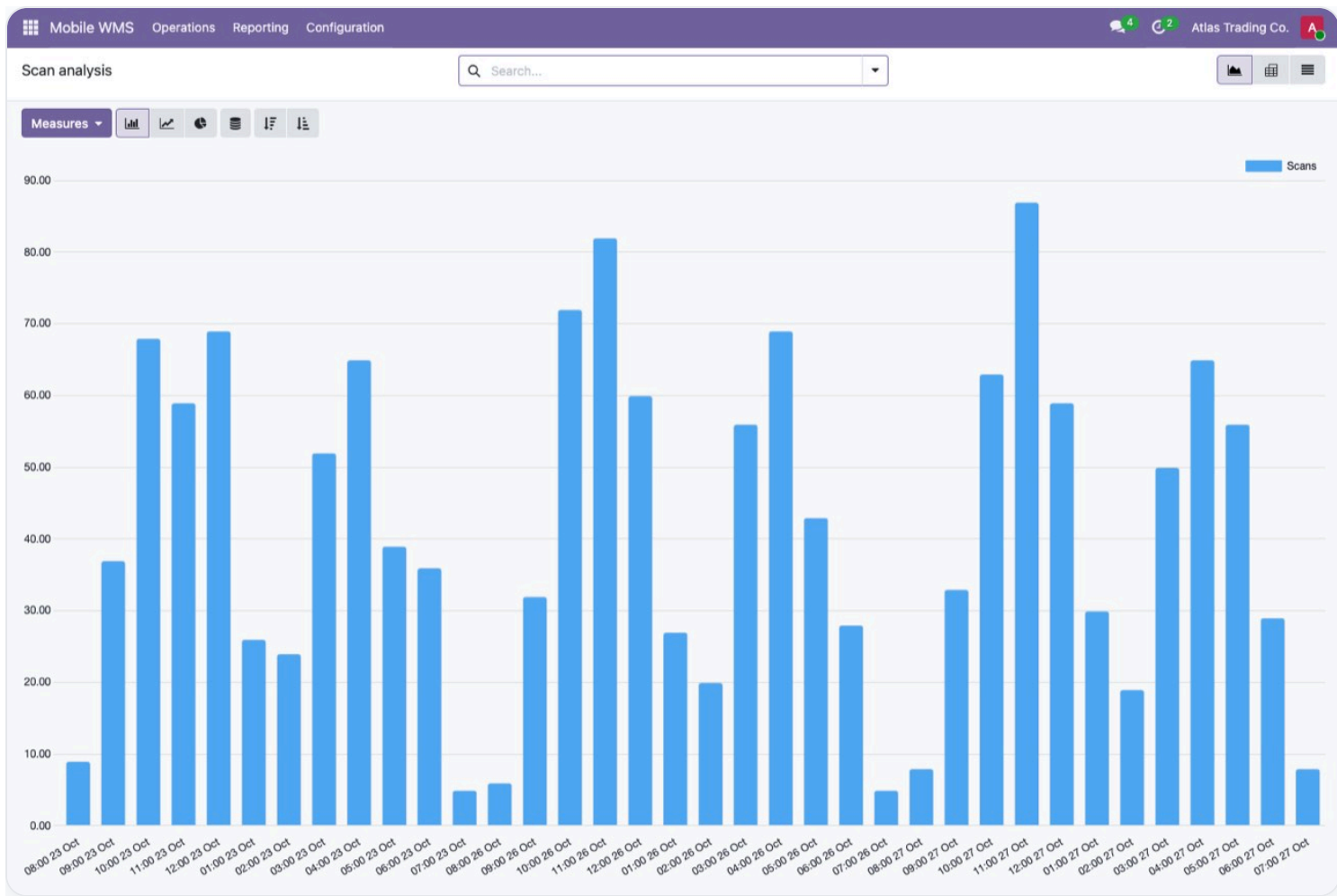


Bin labels with the Code 128 barcode and the check digits.

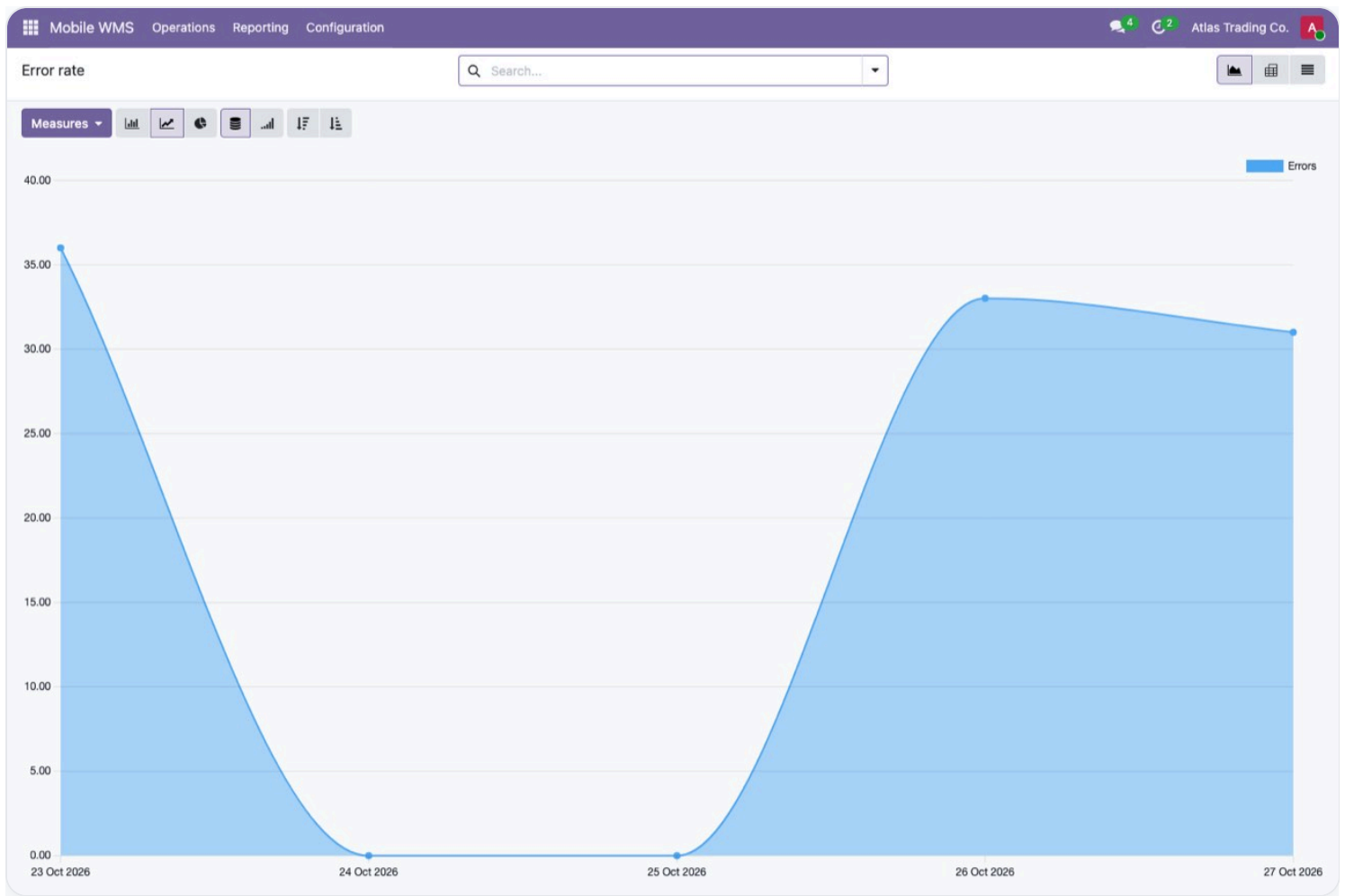
Tip: The barcode images are generated by the server and embedded in the PDF, so printing needs no call back to Odoo.

25. Reports and analysis

In plain words: Every resolve, scan and operation of the app is written to the **scan log** with the operator, the device, the transfer, the barcode, the verdict (ok, line complete, needs input, error, unknown barcode, rejected), the code and the answer time of the server in milliseconds.



Scans and errors per hour.



Errors per day: a rising line means labels or training need attention.

Mobile WMS Operations Reporting Configuration									Atlas Trading Co.	
Scan log									1-80 / 1496	
Q Search...										
☐	Date	User	Kind	Barcode	Picking	Barcode type	Verdict	Code	Answer tim...	
☐	Oct 27, 6:36 PM	Tariq Aziz	Resolve	6291050000285		product	OK		3.02	
☐	Oct 27, 7:19 PM	Layla Nasser	Line update	6291050001008	WH/OUT/00007		OK		5.08	
☐	Oct 27, 6:45 PM	Tariq Aziz	Scan	6291050000070	WH/IN/00001	package	OK		3.14	
☐	Oct 27, 6:56 PM	Layla Nasser	Scan	6291050000483	CS/OUT/00001	package	OK		2.08	
☐	Oct 27, 6:32 PM	Tariq Aziz	Scan	2100123000005	WH/OUT/00009	lot	OK		3.63	
☐	Oct 27, 7:13 PM	Layla Nasser	Scan	6291050000025	WH/IN/00003	package	OK		2.48	
☐	Oct 27, 6:39 PM	Tariq Aziz	Scan	6291050000605	WH/OUT/00004	location	OK		4.52	
☐	Oct 27, 7:05 PM	Yusuf Karim	Scan	6291050000650	WH/OUT/00006	product	OK		3.06	
☐	Oct 27, 7:28 PM	Layla Nasser	Scan	6291050001145	WH/IN/00003	product	OK		3.99	
☐	Oct 27, 6:36 PM	Layla Nasser	Scan	036000291452	CS/OUT/00001	lot	OK		4.57	
☐	Oct 27, 6:40 PM	Yusuf Karim	Scan	6291050000612	WH/OUT/00004	location	OK		4.29	
☐	Oct 27, 6:32 PM	Layla Nasser	Scan	6291050000360	WH/OUT/00007	product	OK		5.63	
☐	Oct 27, 7:06 PM	Tariq Aziz	Scan	6291050000728	WH/OUT/00001	location	OK		3.57	
☐	Oct 27, 7:01 PM	Tariq Aziz	Resolve	6291050001084		package	OK		4.26	
☐	Oct 27, 6:58 PM	Tariq Aziz	Scan	6291050000933	WH/OUT/00001	location	OK		3.23	
☐	Oct 27, 6:57 PM	Mariam Saleh	Scan	6291050000025	WH/OUT/00003	product	OK		4.77	
☐	Oct 27, 7:26 PM	Omar Haddad	Scan	6291050000650	WH/IN/00001	location	OK		3.89	
☐	Oct 27, 7:12 PM	Omar Haddad	Scan	6291050001220	WH/OUT/00003	package	OK		4.87	
☐	Oct 27, 5:48 PM	Omar Haddad	Scan	6291050000100	WH/IN/00004	product	OK		2.75	
☐	Oct 27, 5:51 PM	Mariam Saleh	Scan	6291050000339	WH/OUT/00007	lot	OK		2.98	
☐	Oct 27, 5:39 PM	Yusuf Karim	Scan	6291050000032	WH/OUT/00002	product	OK		2.95	

The scan log: filter by errors, by operator, by slow answers.

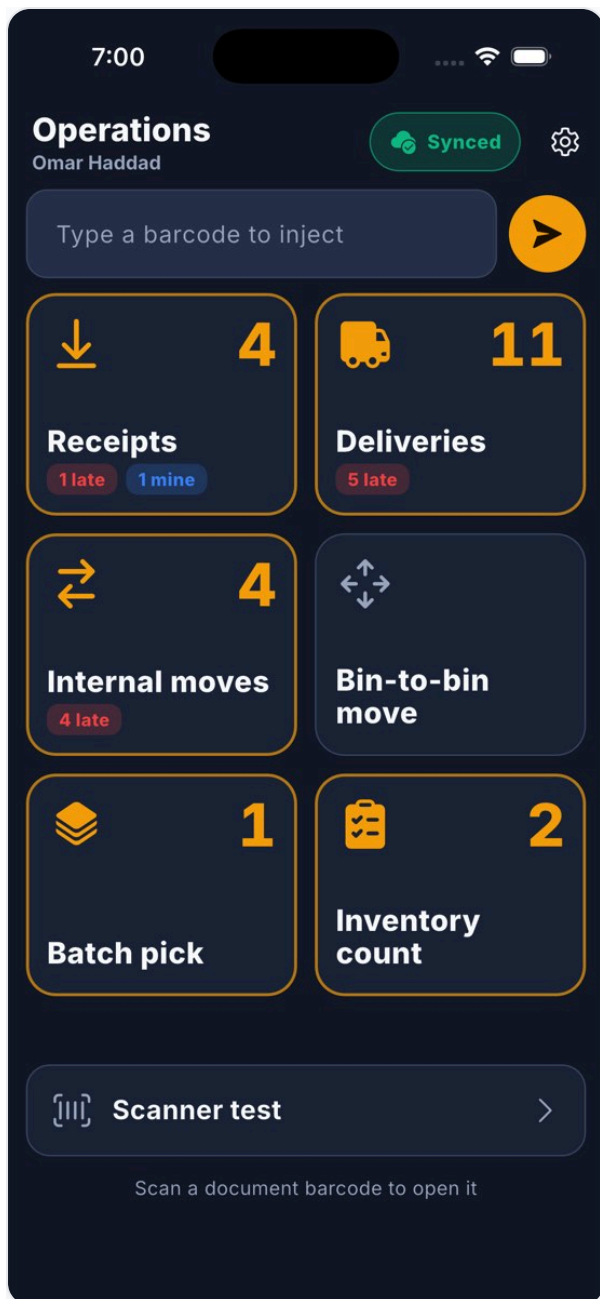
In plain words: The pivot view gives the error rate per operator. *Synchronized operations* lists what the devices sent, with the status, the code and the flags. The activity feed of a transfer shows the flags as notes, and the manager receives one to-do per flagged transfer.

26. The mobile app

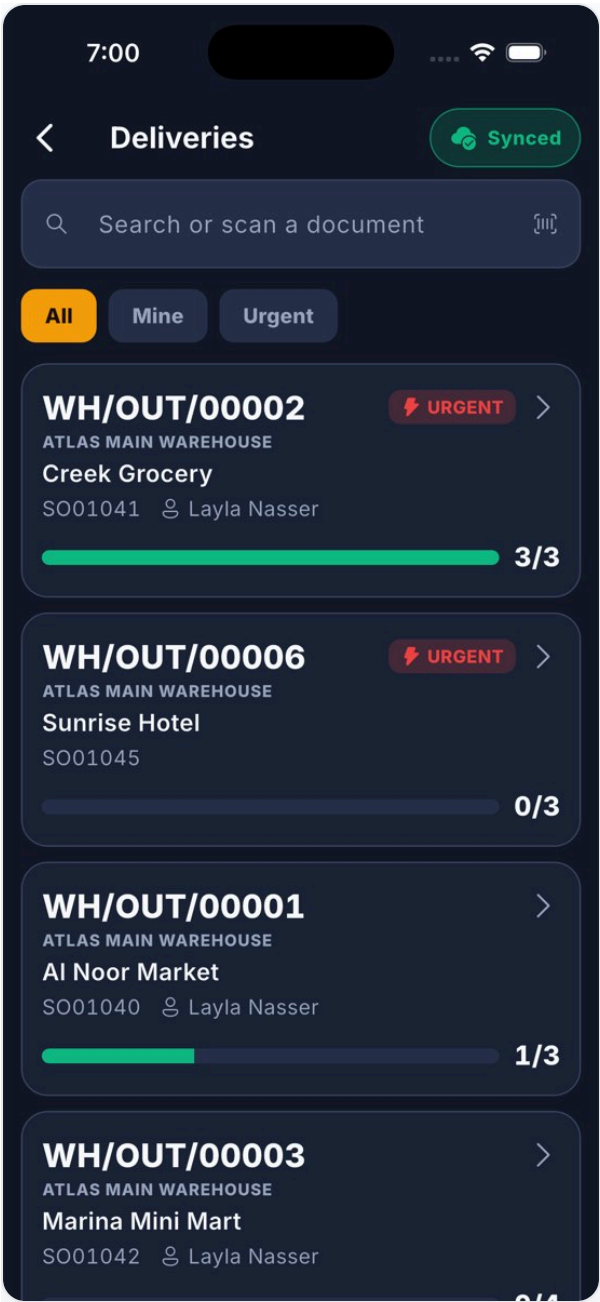
In plain words: The app is built for gloves and bright warehouses: extreme contrast, large touch targets, bin and product codes in very large letters. Colours: deep slate background (a bright theme is a setting too), **amber** for the bin to go to, **green** for a match, **red** for a discrepancy or an over-pick. It speaks English and Arabic (right to left), with the Inter and IBM Plex Sans Arabic fonts bundled, and every icon comes from the **Fluent UI System Icons** set (`fluentui_system_icons`).

In plain words: Nobody types in a text field. A scan goes to the screen on top, is resolved in memory on the device, repaints in the same frame and plays its sound; the document and the outbox operation are written to the local database right after (chapter 20).

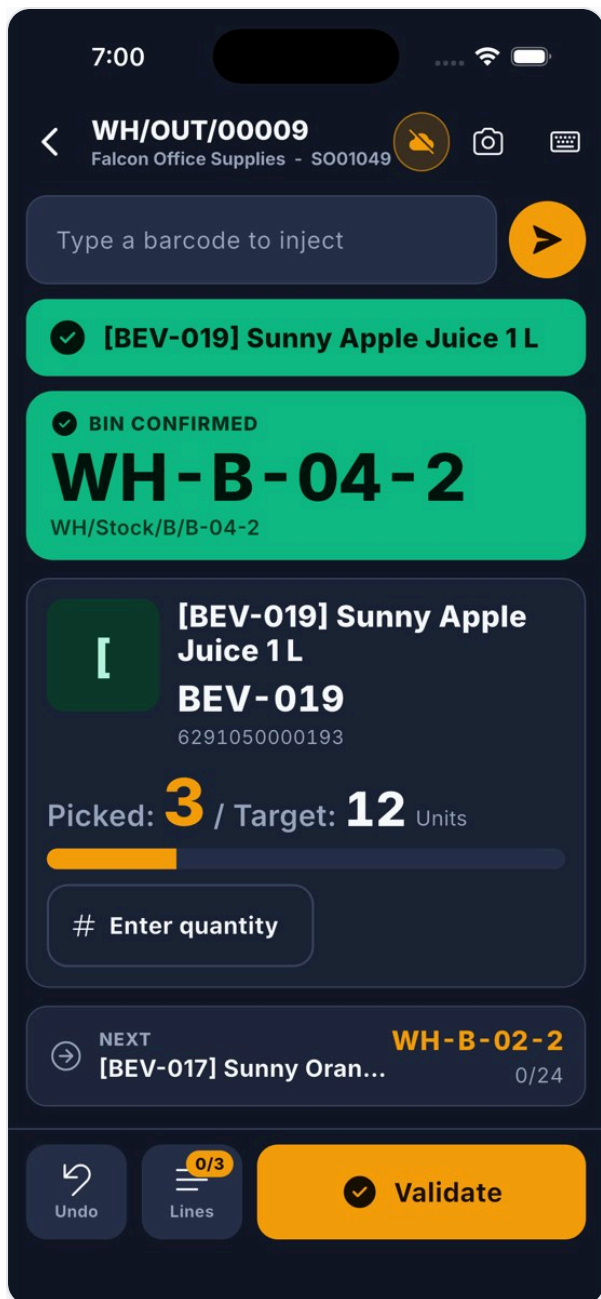
Screen	What it shows
Operations menu	Tiles for Receipts, Deliveries, Internal transfers, Bin-to-bin move, Batch picking and Inventory count, each with the number of transfers that are ready and badges for those that are mine or late. A bar at the top opens a document by scanning its barcode (a transfer, a wave, a count, a purchase order). Settings, the sync status, and a Scanner test row complete the menu.
List of transfers	Mine first, then by priority and deadline; chips All, Mine and Urgent; search by name, origin or customer, or scan a document; a progress bar per transfer.
Scan terminal	Receipts, deliveries, pick and pack. A giant target at the top (the bin to go to, <i>Bin confirmed</i> , <i>Put away to</i>), the card of the product with its picture, <i>Picked / Target</i> and a progress bar, the allocation per order for waves, the next line, a sheet for lot, serial and expiry, a keypad to type a quantity, and the buttons Undo, Lines (or Packing) and Validate.
Bin-to-bin move	A free move in three steps shown as a stepper: Source, Items, Destination. Pallets move with one scan. Buttons Undo, Reset and Confirm move.
Batch picking	The wave along its pick path: <i>Stop 1 of 11</i> with the aisle and the bin in large letters, the product card with the quantity allocated to each order, packing.
Inventory count	A draft count with a <i>Start counting</i> button; the shelves as chips; the counted items per shelf; Undo, Next shelf and Review; the count review for supervisors with the discrepancies and <i>Apply inventory</i> .
Sync status	Online or offline, the last sync, <i>Sync now</i> , the queue of operations with retry and discard.
Settings, Scanner test	Scanner profile, DataWedge, custom intent, trigger keys, camera options, sound, vibration, loud mode, theme, language, master data, sign-out.



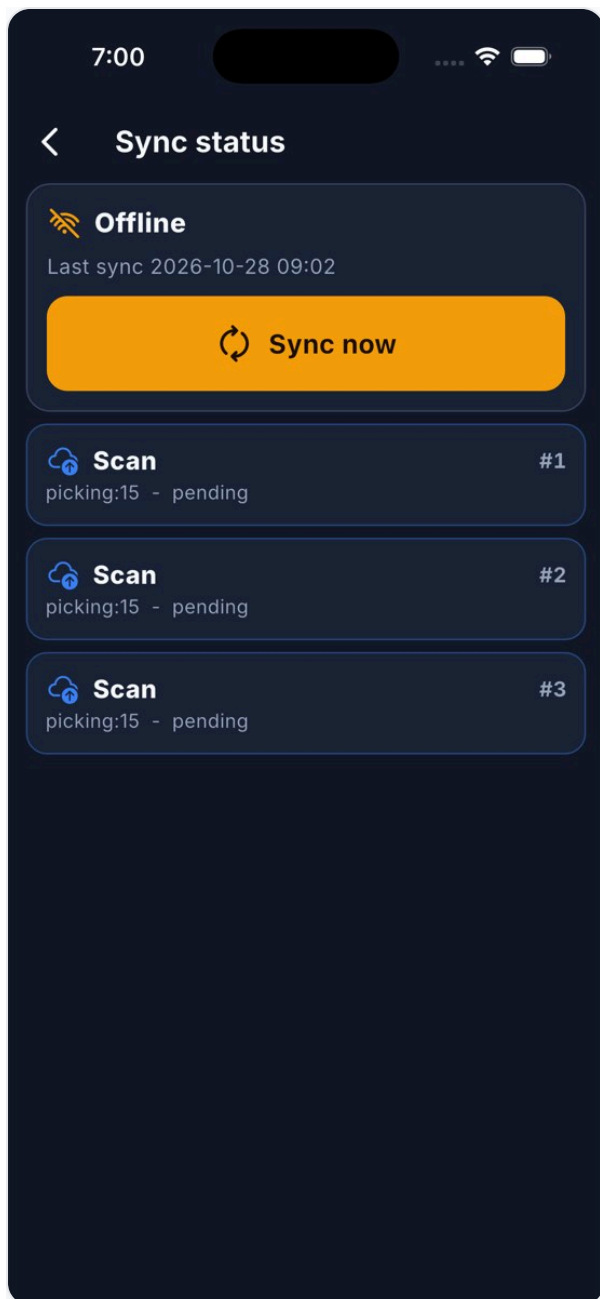
The operations menu: ready, mine and late per operation, the sync state and the Scanner test row.



The deliveries: urgent ones first, the customer, the sales order and a progress bar per transfer.



A delivery being picked with no network: the cloud icon turns amber, scans carry on, and the operations wait in the outbox.



Sync status while offline: the queued scans, each with its number and its state, and Sync now.

In plain words: Counts and tests of the app. The project has **346 Dart tests** (parsers, the GS1 parser, the resolver and the engines, scanner intent shapes of the four vendors, key bursts, the native channels with fakes, feedback patterns, the outbox and sync engine, the repository over a real socket, widget flows of the terminal, the wave, the count, the move, the menu and the settings, English and Arabic key parity, and a resolve benchmark) and **53 Kotlin JVM tests** of the native layer (extras flattening, the DataWedge configuration, vendor detection, trigger keys, intent filters, the event buffer). `flutter analyze` is clean. The app was built as a debug APK for Android, built for the iOS simulator, and run end to end on the iOS simulator against a real Odoo with this module: receipts with lot, serial and expiry from GS1 barcodes, deliveries with a

partial pick and a backorder, bin moves, a wave, a blind count by an operator approved by a manager, error paths, and an offline run (server stopped, scans queued, outbox synced after the restart). What was *not* verified is listed in chapter 31.

In plain words: The screenshots of this guide were taken in the iOS simulator against a demo database. The source has architecture notes in [docs/ARCHITECTURE.md](#) (layers, the scan path, the rules, the outbox, the native layer, the not-verified list).

27. For developers: the API and the resolver

In plain words: Skip this chapter if you only use the product. The complete contract is in [documentation/API.md](#).

```
# is the server reachable? (no sign-in)
curl https://odoo.example.com/api/wms/v1/ping
# {"ok": true, "api_version": 1, "server_time": "2026-10-28T08:00:00Z"}

# sign in as a warehouse operator (returns a token, the configuration and the GS1 rules)
curl -X POST https://odoo.example.com/api/wms/v1/auth/login -H 'Content-Type: application
  -d '{"login": "omar", "password": "...", "device": {"id": "TC52-0147", "name": "TC52 do

# what is this barcode? (GS1-128 with GTIN, lot and expiry; the answer says which product
curl -X POST https://odoo.example.com/api/wms/v1/resolve -H "Authorization: Bearer $TOKEN
  -d '{"barcode": "010629105000040710L2610A\u001d17261118", "context": {"picking_id": 12}

# push the queued operations: every operation has a uuid and a local_seq, a repeat change
curl -X POST https://odoo.example.com/api/wms/v1/sync -H "Authorization: Bearer $TOKEN" -
  -d '{"device_id": "TC52-0147", "ops": [{"uuid": "9d1c...", "type": "scan", "local_seq":
    "payload": {"picking_id": 12, "barcode": "6291050000407", "qty": 6, "location_id"
```

Area	Endpoints under /api/wms/v1
Session	GET /ping, /config; POST /auth/login, /auth/logout
Resolver	POST /resolve (barcode and context); GET /locations/<barcode>/check?digits=NN
Lists	GET /menu, /pickings, /pickings/<id>, /batches, /batches/<id>, /counts, /counts/<id>, /changes; POST /pickings/<id>/claim, /release, /batches/<id>/start, /claim, /counts/<id>/start
Operations	POST /sync (scan, line_update, validate, move, batch_pick_line, batch_validate, count_line, count_start, count_submit, count_create) or POST /ops/<type> with the same payload
Offline cache	POST /master (cursor, scope: products, locations, packages, lots), paged

In plain words: Calls send `Authorization: Bearer <token>` and optionally `Accept-Language`. Errors are `{"error": {"code", "message"}}` with the HTTP status 400, 401, 403, 404, 409 or 422. Sign-in is limited to 8 wrong attempts per 5 minutes for the same address and login.

```
# the engine of the module, in a shell or in your own code
env["wms.barcode.resolver"].resolve("010629105000040710L2610A\x1d17261118", {"picking_id"
# {"hits": [{"type": "product", "id": 9209, "lot_name": "L2610A", "expiration_date": "202
# "primary": {...}, "gs1": {"parsed": [{"ai": "01", "value": "..."}, ...], "raw": "..."]}
```

In plain words: Measured speed. The automated test resolves 200 mixed barcodes (products, bins, lots, pallets, GS1 strings, purchase order numbers, weight barcodes and unknown codes) on a database with about 2,200 products, 2,200 bins, 300 lots and 300 packages and requires a mean below 50 ms. In seven runs on a laptop the mean was between 3.3 and 9.3 ms and the 95th percentile between 4.8 and 13.3 ms. These are server-side times inside Odoo (the `ms` field of the answer); the network and the HTTP stack add their own time. Larger databases were not measured. Each kind of record is looked up with one indexed query.

In plain words: Python entry points on `stock.picking` : `process_barcode_scan(barcode, location_id, package_id, ...)` and `action_mobile_validate(picking_id, lines_data, backorder_policy)` ; on `wms.batch.pick` : `wms_scan` , `wms_validate` ; on `wms.count.session` : `wms_count_line` , `wms_submit` , `action_apply` . They check the group of the user themselves and do their work with elevated rights after the checks.

28. Security

Topic	What the module does
Tokens	A device signs in with login and password and receives a random token; only its SHA-256 hash is stored. The token ends when the user is archived, loses the group or is signed out; tokens not used for 180 days are deleted.
Authorization	Every public method checks the group, the company, the operation type and the claim. Nothing is decided from a context key or from a field that a client can write.
Workflow fields	State, approved by, applied and the scanned quantities of a count can only be changed by the buttons or the API, never directly.
Input	Quantities must be finite numbers in range, dates are validated, barcodes are limited to 300 characters, payloads to 100 operations per call; NaN and infinity are rejected.
Count integrity	Expected quantities are frozen and hidden from operators; approval is serialized with a row lock, so a count is never applied twice; whoever counted, submitted or corrected a count cannot approve it, and the counter of a started count cannot be changed.
Companies	A scan answer only shows data of the companies of the signed-in user: asking for another company in the request context is refused.
Audit	The scan log and the synchronized operations keep who, when, which device and what the answer was.

29. Update

Replace the folder with the new version, restart Odoo and run `-u taqnix_wms_barcode` . The bins, the sequences, the settings, the waves, the counts and the scan log are kept. Update the app on the devices with the new build; the app and the module talk API version 1.

30. Troubleshooting

Problem	What to do
The app cannot connect	Open <code>https://your-odoo/api/wms/v1/ping</code> on the device: it must answer <code>{"ok": true}</code> . Check https, the certificate, the address without a trailing slash and the database (if you have several, use the <code>dbfilter</code> setting or send the header <code>X-0doo-Database</code>).
Sign-in says the account is not an operator	Give the user the group <i>Mobile WMS / Operator</i> (or Manager).
The transfer is not in the list	Its operation type must have Mobile WMS ticked, its state must be Ready (or Waiting), and it must belong to the company of the user.
The scanner beeps but nothing happens in the app	The intent action of the scanner profile and the one in the app settings must be identical, delivery must be <i>Broadcast</i> , and keystroke output must be off.
The scan types into a field	Keystroke output (DataWedge) or keyboard wedge is still on.
Unknown barcode	Check the barcode of the product (Inventory > Products), the packaging barcode, and whether the code is GS1 (the engine tells <i>bad check digit</i> when the number is wrong).
Check digit refused	The digits are on the label and on the bin form; a bin without digits accepts anything.
Serial or lot refused	Read the code in the answer: <code>serial_duplicate</code> , <code>unknown_lot</code> , <code>lot_expired</code> , <code>tracking_required</code> .
Over-pick	Raise the tolerance in the settings or in the operation type, or validate with a backorder.
The wave cannot be validated	The message names the transfer that fails; fix it (lot, destination) and validate again: nothing was written.
A count cannot be approved	The count must be in Review; the approver must not be the counter (settings); someone else may already have approved it.

Labels print without the barcode	Check that <code>wkhtmltopdf</code> works and that the bin has a barcode.
Operations stay in the queue	Open <i>Synchronized operations</i> to see the code and message of the rejected ones.
Scanner problem	What to do
Nothing happens when I pull the trigger	Open <i>Settings > Scanner test</i> and pull it. If <i>Last raw intent</i> stays empty, the intent never arrived: compare the action in the scanner profile with the app, check that delivery is <i>Broadcast</i> , and that the profile is associated with <code>com.taqnix.wmsscanner</code> .
The trigger button does not scan	The key codes of the button depend on the model: read them in <i>Scanner test</i> and type them in <i>Scan trigger key codes</i> .
The scan is counted twice	Repeats inside 400 ms are dropped. If your scanner sends the same read with two different profiles (for example intent and keyboard wedge at the same time), switch one off.
The camera does not open	Allow the camera permission to the app in the Android or iOS settings.

31. Limits and what has not been verified

In plain words: Read this before you decide. It is the honest list.

Topic	Detail
Real scanners	No Zebra, Honeywell or Datalogic device was available to the author. The DataWedge profile (<code>SET_CONFIG</code>), the intents and extras of the three vendors, the soft scan trigger and the default trigger key codes follow the vendors' documentation and are unit tested on the parsing side only. The delivery of a real broadcast, the behaviour of the receiver on a given Android image, and the differences between DataWedge 6.x and 8.x must be confirmed on your hardware. Test one device before you buy fifty.
Honeywell symbologies	The Honeywell <code>codeId</code> letters are mapped for a few common symbologies only; the AIM id is the reliable source.
Camera, audio, battery	Camera scanning (torch, scan window, continuous mode) was not run on a device: the iOS simulator has no camera. The latency of the sounds, the strength of the vibration, the battery drain with the screen kept on and the behaviour of the receiver when the screen is off or in doze were not measured.
Android emulator	The debug APK builds, but the app was not run on an Android emulator or phone by the author. It was run on the iOS simulator against the real module.
iOS	iOS has no hardware scanner plugin: iPhones scan with the camera.
Offline weight barcodes	A weight-embedded EAN-13 resolves on the device only when the product record carries its code (the <i>plu</i> field) or its internal reference contains it; otherwise the server resolves it, so it needs the network.
Serial numbers	One scan is one unit. A lot change in the middle of a line replaces the lot of that line on the device; the server splits the line when it receives it.
Units of measure	A unit of measure that is not the product's own is shown, but quantities are not converted on the device.
Pack barcode in a count	Scanning the barcode of a pack of a product during a count creates a separate count line instead of landing on the line of the package that holds it. The review then shows the package line as missing (negative) and the separate line as found (positive): read the two together and correct them before you apply.
Manufacturing	Manufacturing orders (picking components, producing) are not handled.

Wave planning	Waves are made by hand or from a selection. There is no automatic wave planning by weight, volume, carrier cut-off or zone.
Put-away	The destination bin of a line is the one that Odoo's putaway rules give; the module does not add its own put-away strategy.
GS1 without separators	A GS1 string with a variable-length element followed by another element and no separator is ambiguous, as in the GS1 standard; the scanner must send FNC1.
Expiry	Without Odoo's Expiration Dates feature the expiry is kept in a field of this module. The behaviour with that feature installed is covered by the same code path but was not tested in a database that has it.
Offline	The app works on what it cached; creating a lot unknown to the server is possible on receipts only. Two devices working on the same transfer are prevented by the claim, not merged.
Scale	The resolver was measured on a database with about 2,200 products and 2,200 bins (chapter 27). Larger databases were not measured.
Odoo version	Built for and tested on Odoo 19.0.

32. Support

Installation help and bug fixes are included for 6 months, with lifetime updates. Contact us through the marketplace page where you bought the module or write to **support@taqnix.com** with your Odoo version and a screenshot.

Changelog

1.0.0 – Initial release

- Odoo module: server barcode resolver (GS1-128, EAN-13, EAN-8, UPC-A, UPC-E, GTIN-14, packaging, weight and price barcodes, SSCC), scan engine for receipts, deliveries and internal moves with lots, serials, expiry, packages and tolerance, atomic validation with backorder create, none or ask, claim of transfers, waves with a snake pick path and all-or-nothing validation, blind and guided cycle counts with drift, missing items and manager approval, bin check digits, aisle, bay, level and pick sequence, bin and package labels, scan log with answer times, idempotent sync API with master data paging, devices
- Flutter scanner app: scanner layer for Zebra DataWedge (profile TaqnixWMS created through SET_CONFIG), Honeywell, Datalogic, custom intents, keyboard wedge bursts, trigger keys and a camera fallback, one scan bus, offline-first engines with a local database and an outbox, sounds and haptics, screens for receipts, deliveries, moves, waves and blind counts with start and manager review, a scanner test screen, English and Arabic, 346 Dart tests and 53 Kotlin tests
- Arabic translation and documentation, 241 automated tests of the module