

Grocery POS — User Guide

This guide covers every feature in the app. To see which version **you** are running, open **Settings** → **Updates** — if you're on an older version, some features below may not appear yet. Keep this guide handy so staff can look up anything without needing to call for help.

Recent additions:

- **Priced extras (restaurant add-ons)** — define add-ons like cheese or double with a price, then attach them to a cart line; the cost flows into the line total, receipt, and prep ticket. See [Section 4 → Adding Extras to a Line](#) and [Section 16 → Extras](#).
- **In-app updates** — **Settings** → **Updates** shows your current version, checks for a newer one, and (on Windows) downloads + installs it after an automatic backup. See [Section 16 → Updates](#).
- **Archive vs delete products, and bulk import** — products can now be archived (hidden, history kept) or permanently deleted (only if never sold), one at a time or via multi-select, and you can import products from a CSV/JSON file. See [Section 5 → Archiving vs Deleting a Product](#) and [Importing Products in Bulk](#).
- **Multi-terminal LAN sync** — run more than one till in the same shop and replicate sales and catalog over the local network, with admin-approved peer pairing. See [Section 23 → Multi-Terminal Sync \(LAN\)](#).
- **Per-type inventory-document permissions + grouped navigation** — the single “Manage Inventory Documents” permission is split per document type, a new **Delete Products** permission exists, and the side navigation is grouped (Operations / Inventory / People / Finance / Admin). See [Section 3 → User Roles and Permissions](#).

Table of Contents

- [1. Getting Started — License Activation](#)
- [2. Login and PIN](#)
- [3. User Roles and Permissions](#)
- [4. POS — Point of Sale](#)
- [5. Products](#)
- [6. Categories](#)
- [7. Inventory Overview](#)
- [8. Inventory Documents](#)
- [9. Suppliers](#)
- [10. General Expenses](#)
- [11. Shifts](#)

- 12. Orders and Refunds
 - 13. Reports
 - 14. Cashier Activity Log
 - 15. Users
 - 16. Settings
 - 17. Backup and Restore
 - 18. Activation
 - 19. Customers
 - 20. Accounting
 - 21. Promotions & Discounts
 - 22. ZATCA E-Invoicing (Phase 2)
 - 23. Multi-Terminal Sync (LAN)
-

1. Getting Started — License Activation

Before anyone can use the app, it must be activated. Your licence is **perpetual** — one purchase, no renewals, no expiry.

First Launch

1. Open the app. If no valid licence is found, the **Activation** screen appears automatically.
2. Enter the **activation code** that was emailed to you after purchase (also shown on your order page on numu-tech.com).
3. Tap **Activate**. The app contacts the NUMU server once, registers this device to your licence, and starts.
4. On success you are taken to the login screen.

Activation needs an internet connection **once**. After that the app runs fully offline, forever — the licence is verified on your device with a cryptographic signature.

Reinstalling or moving to a new computer

- Reinstalling on the **same computer**: enter the same code — it re-activates without using another device slot.
- Moving to a **new computer**: your licence allows a set number of devices. If the slot is taken by the old machine, free it yourself from your account page (account.numu-tech.com → Devices → deactivate), or ask support on WhatsApp — then activate the new machine with the same code.

Manual activation (no internet on the till)

If the till has no internet at all, your order page provides a long **offline licence code** (a long string with a  in the middle). On the Activation screen, expand the manual option and paste the whole string, including

everything after the **.**.

2. Login and PIN

Selecting a User and Entering a PIN

1. The login screen shows a list of all active users.
2. Tap your name.
3. Enter your **4–6 digit PIN** on the numeric keypad.
4. Tap **Login** (or the confirm button).

First-Time Login

- The default admin account is:
 - **Username:** Admin
 - **PIN:** 99099
- On the very first login you are **required to change** your PIN before you can proceed. Choose something only you know.

Lockout Protection

- After **5 consecutive wrong PIN attempts** the account is locked for **5 minutes**.
- The lockout remains in effect even if the app is closed and reopened. Wait 5 minutes, then try again. An admin can also reset the user's PIN if needed.

Idle Auto-Logout

- By default, if the screen is left idle for **15 minutes** the app automatically returns to the login screen.
- The timeout is **configurable** in **Settings** → **Auto logout when idle**. Choose **Off**, **5**, **15**, **30**, or **60** minutes. Set it to **Off** to keep the cashier signed in indefinitely until they press Logout explicitly. (Useful for kiosks and small shops where the same person runs the till all day.)
- Any open cart is preserved — you will not lose a sale in progress.
- After an idle-logout you can sign back in immediately from the login screen; you do **not** need to close and reopen the app. (Previously a bug here forced an app restart; fixed in v1.2.)

Changing Your PIN

1. Go to **Settings** → **Change PIN** (or the equivalent menu item shown to you after login).
2. Enter your current PIN.
3. Enter your new PIN twice to confirm.
4. Tap **Save**.

Forgot the admin PIN?

If everyone has forgotten the admin PIN and there is **no other admin** to reset it (**Users** → **reset PIN**), you can recover access without losing any data:

1. On any device with internet, the shop owner signs in to their NUMU account at **account.numu-tech.com**, opens **this till** under their devices, and taps **Reset POS admin PIN**. A recovery code appears.
2. Back on the till's login screen, tap **Forgot admin PIN?**, paste the code, and tap **Recover**.
3. The app sets a **new one-time PIN** and shows it on screen. Sign in with it — you are immediately asked to set your own new PIN.

Good to know:

- The code only works on **the till it was generated for** and **expires after 30 minutes**. Generate a fresh one if it lapses.
- Run the recovery **on the till you want to sign in from** — the new one-time PIN is shown only on that till.
- No sales, inventory, or settings are lost; only the admin PIN changes.
- No internet or no NUMU account access? Ask your vendor — they have an on-site reset and a backup-based reset (see the Owner guide).

3. User Roles and Permissions

There are two roles in the system: **Admin** and **Cashier**.

Admin

- Full access to every screen and feature.
- Can add, edit, and deactivate users.
- Can manage categories, settings, backup, activation, and reports.

Cashier

- Access is limited to the **POS** screen by default.
- An admin can unlock optional permissions for each cashier:

Permission	What it unlocks
Add Products	The cashier can create, edit, import , and archive products from the Products screen
Delete Products	The cashier can permanently delete products that have no sales history (separate from Add Products — see §5)

Permission	What it unlocks
Edit Product Cost	The cashier can change a product's unit cost on the Products page. Without it, cost stays read-only and is set only by inventory documents. Money-sensitive — granted by default to the Manager preset (see §5)
Refund Orders	The cashier can issue full or partial refunds from the Orders screen
View Sales History	The cashier can open the Orders screen to browse past sales
Manage Purchase Documents	The cashier can create and post purchase inventory documents
Manage Restock Documents	The cashier can create and post restock inventory documents
Manage Waste Documents	The cashier can create and post waste inventory documents
Manage Adjustment Documents	The cashier can create and post adjustment inventory documents
Manage Count Documents	The cashier can create and post stock-count inventory documents
Reverse Inventory Document	The cashier can post a reversal/correction of an already-posted document
Manage General Expenses	The cashier can create and edit general expenses
Record Cash In/Out	The cashier can record cash added to or taken from the drawer of an open shift (owner withdrawals, float top-ups, bank/safe drops). Granted by default to the Manager and Shift Lead presets
View Reports	The cashier can open the Reports screen
Manage Customers	The cashier can open the Customers screen and maintain customer records
Export Accounting	The cashier can open the Accounting screen and export finance CSV files

Inventory-document permissions are now per type. What used to be a single “Manage Inventory Documents” switch is split into one switch per document type (purchase, restock, waste, adjustment, stock-count) plus a separate **Reverse** switch. Grant only the document types a cashier should handle.

The navigation rail is grouped. Menu items on the left are organised into **Operations** (POS, Orders, Shifts), **Inventory** (Inventory, Products, Categories, Documents), **People** (Customers, Suppliers), **Finance** (General Expenses, Reports, Accounting), and **Admin** (Activity Log, Settings, Users, Backup, Activation). Items only appear if your role/permissions allow them.

Changing Cashier Permissions

1. Go to **Users** (admin only).
2. Tap the cashier's name.
3. Toggle the desired permission switches.
4. Tap **Save**.

4. POS — Point of Sale

The POS screen is where all sales happen.

Top bar

The centered title in the top bar shows your **business name** (set under **Settings** → **Business Info**). If you haven't filled in a store name yet, it falls back to **NUMU**. Edit the name in Business Info and the title updates immediately — no app restart needed.

Foldable side navigation

The side menu can be **folded** to give the cashier more screen space during checkout. Click the chevron at the top of the side rail to collapse it; a small menu button appears at the top corner to bring it back. The choice is **remembered per device** so the next time you launch, the menu stays the way you left it. Recommended for busy registers.

Finding and Adding Products

There are four ways to add a product to the cart:

Method	How
Barcode scan	Point the device camera or USB scanner at the barcode. The product is added instantly.
Search	Type part of the product name or SKU in the search bar at the top. Tap a result to add it.
Category filter	Tap a category chip below the search bar to filter the grid.

Method	How
Grid tap	Tap any product card in the grid.

If a **scanned barcode is unknown**, the POS now plays a short alert tone and shows a warning toast `Barcode not found: <code>` so the cashier hears + sees the miss immediately. The scanned text stays in the search box so they can correct it or search by name instead.

Weighed produce (scale barcodes). If your store uses a weighing scale that prints its own barcodes, the cashier scans the label exactly like any other barcode — the POS reads the weight embedded in the barcode and adds the line at the correct quantity automatically (e.g. `Tomatoes 1.250 kg × 4.00 = 5.00`). No extra steps at the till. This works only after an admin turns the feature on and matches it to the scale in **Settings** → **Weighed barcodes**, and tags each weighed product with its **PLU code**. See [Selling produce by weight](#).

Multiple Sell Options (Packages vs. Singles)

If a product has more than one sell option (e.g., a single chocolate bar and a box of 12), a dialog appears asking which option you want. Tap the one the customer is buying.

Composite Products (Recipes / Kits)

If a product is flagged as **composite** in the catalog (see Section 5), selling one **automatically deducts its components' stock** instead of its own. The cashier never sees this complexity — they just sell the composite at its price, and the POS quietly reduces each component's stock by `sold qty × quantity per unit`. If any component is out of stock, the sale is blocked just like a normal stock-out.

The **stock badge** on a composite's product card shows derived availability — the number of composites you can still assemble given current component stock. A "Pizza Slice" composite whose only component is "Whole Pizza" (qty-per-unit `0.125`) reads **In stock** with 80 slices available when 10 whole pizzas are in stock, ticks down to **Stock low** as components run out, and turns red **Out of stock** the moment any one component can no longer cover one composite. For the full walk-through on configuring this, see [Section 5](#) → [Selling a product in slices or portions](#).

Cart panel

The right-hand **Current Cart** panel is laid out to give the cart-line list as much room as possible:

- **Customer selector** — now a compact chip in the **Current Cart** header, immediately to the left of the cart-count badge. When no customer is selected, the chip reads **Select customer**; tap it to open the customer picker. When a customer is selected, the chip shows their name (tap the name to switch customer, tap the × to clear).
- **General notes** — the kitchen-notes box is now **collapsible**. By default the box is collapsed to a one-line header that shows the title, an edit button, and a chevron. Tap anywhere on the header to expand it and see the note summary. A small filled dot appears in the collapsed header when a note exists. Once you

add or edit a note, the box stays expanded for the rest of the sale, so the cashier doesn't have to remember to re-open it.

Together these reclaim ~100 pixels for the scrollable cart list, so more items fit before scrolling.

Adjusting Quantity and Line Discount

- Tap a cart line to open the **line editor**.
- Change the **quantity** using the **+** / **-** buttons or by typing.
- Enter a **line discount** as a percentage (e.g., **10** for 10 % off that line).
- Tap **Done**.

Adding Extras to a Line (Cheese, Double, ...)

For restaurants, a menu item often has priced add-ons — extra cheese, a double patty, an extra sauce. These are called **extras**.

1. Tap a cart line to open the **line editor**.
2. Under **Add extras**, tap the chips for the add-ons the customer wants. Each chip shows its price (e.g. **Cheese +3.00**).
3. Tap a selected chip again to remove it.
4. Tap **Done**.

The extra's price is added to the line automatically, so the line total, VAT, and the order total all update — there is nothing extra to ring up. The chosen extras are shown:

- next to the item on the **customer receipt** (e.g. **Burger (+Cheese, +Double)**),
- and on the **kitchen prep ticket**, so the kitchen knows what to add.

Extras are defined once by an admin in **Settings** → **Extras** (see Section 16). If no extras are configured, the **Add extras** section simply doesn't appear. The same line editor is also where you switch a product's variant (e.g. Half vs. Whole) — extras and variants can be combined on one line.

Cart Discount

- Tap the **discount** icon at the bottom of the cart.
- Enter the discount percentage that applies to the whole order.

Bundle Discounts

If active **bundle promotions** match the contents of the cart, they auto-apply with no cashier action. The totals area shows:

- **Applied bundles** list — each matched bundle's name with how many times it triggered (e.g. **Family Pack × 2**).

- **Bundle savings** row — the total amount saved across all applied bundles.

The bundle savings reduce the order total and are correctly factored into VAT (no over-charging on items the customer didn't pay for). See Section 21 to configure bundles.

Category & time discounts set up by an admin also apply here automatically. When a scheduled discount is inside its active window (its days + hours), every cart item in the chosen category is reduced by the discount percentage and the saving appears on the totals. See Section 21.

Parked Sales

- Tap **New Sale** to start a fresh cart and park the current one.
- Tap the parked sale tabs (numbered) to switch between open carts.
- Each parked cart is independent.

Payment

1. When the cart is ready tap **Charge** (or the payment button).
2. Select **Cash** or **Card**.
 - For **Cash**: enter the amount tendered; the change due is calculated automatically.
 - For **Card**: confirm the amount and tap **Confirm**.
 - If a tracked product does not have enough stock, the sale is blocked before completion and the app shows a stock warning instead of creating a partial order.
3. An order confirmation appears. From here you can:
 - **Print receipt** — sends to the configured receipt printer.
 - **New Sale** — clears the cart and starts fresh.

Printing Prep Tickets

If a category is linked to a prep printer, a prep ticket is sent to that printer automatically when an order is placed. No manual action is needed.

5. Products

Who Can Add Products

- **Admin** — always.
- **Cashier** — only if the admin has enabled the *Add Products* permission for that user.

Finding products and filter chips

The **search bar** at the top matches product name (Arabic or English), barcode, or SKU as you type — no need to press Enter. Below it is a row of **filter chips**: tap one to show only the matching products, tap it

again to clear it. Each chip carries a **live count**, so at a glance you can see how many products fall into it.

Chip	Shows
Low	Stock at or below the product's low-stock threshold
Out of stock	Stock of zero or less
Negative stock	Stock below zero — items the till has oversold (sold more than was on hand). Recount and correct these.
Zero cost	Products with no cost entered (cost = 0). Their profit and food-cost figures are wrong until a cost is set — use this chip to find and fix them.
Expired	Past their expiry date
Expiring soon	Expiry within the configured window

Creating a Product

1. Go to **Products** and tap the **+** button.
2. Fill in the base product fields:

Field	Description
SKU	A unique code you assign. Can be left blank — one will be generated.
PLU code	Optional. The short item code a weighing scale prints into its barcodes (e.g. bananas = 4011). Only needed for produce sold by weight — see Selling produce by weight . Leave blank for normal products.
Arabic Name	How the product appears in Arabic UI and on receipts.
English Name	How the product appears in English UI.
Price	The default selling price in the store's configured currency.
Base Unit	The smallest inventory unit (e.g., <i>piece</i> , <i>kg</i> , <i>litre</i>).
Barcode	The main barcode for this product.
VAT / Tax	Set the product's tax treatment. Depending on your store setup, you may see a simple VAT Exempt toggle while tax rates are managed centrally in Settings → Tax Settings .
Track Inventory	Toggle on to enable stock tracking for this product.

3. **Packages section** — if the product is sold in multiple package sizes:

- Tap **Add Package**.
- Per package row, enter:
 - **Barcode** — the scannable code printed on the box / pack. Required.
 - **Display name (Arabic)** and **Display name (English)** — what shows on the cart line and the receipt. These are now **two separate fields**; whatever you type in each box is what's stored, with no Arabic-to-English fallback. Leave one empty if you don't need it.
 - **Unit** (e.g. *box*, *carton*), **Quantity per unit**, and **Package price**.
- Mark one row **Primary** by tapping the star on its header. The starred package is the canonical entry — used for shelf labels and product-list display. Removing the primary card automatically promotes the next remaining row.
- **Save-time validation:**
 - If any non-empty row is missing a barcode, save is blocked with a clear error (no silent data loss).
 - If two rows share the same barcode, save is blocked with Barcode "X" is used more than once.
 - If a barcode is already used by a **different product** in the catalog, save is blocked with Barcode "X" is already used by product "<name>".
 - Quantity-per-unit must be a positive number; the input field accepts only digits and a single decimal point.

4. **Composite product (recipe / kit)** — toggle this on if the product is *assembled* from other products rather than being purchased and stocked on its own. Examples: a sandwich made from bread + cheese + lettuce; a gift basket; a 12-pack you build internally from singles.

- When the toggle is on, the **Components** section appears.
- Add one row per ingredient: pick the **component product** from the dropdown and set the **quantity per unit** (how many base units of that component go into one composite).
- Composite products **do not maintain their own stock**. The stock field is irrelevant on them — selling one decrements each component instead, and reports the components' batch/FIFO/cost machinery as if the components were sold directly.
- As you add components, the editor shows a live **Recipe cost / unit** and **Food cost %**. You don't type a cost on a composite — it's derived from the recipe and booked as the dish's cost of goods at sale, so the dish is profit-tracked. Full details under [Option B — Composite product](#).
- You **cannot** add a composite as a component of another composite in this version. Pick leaf products only.
- A composite **must have at least one component** before you can save it; the editor blocks save otherwise.

5. Tap **Save**.

Automatic finance tracking:

When this product is sold, the app stores the unit cost at the moment of sale automatically. Staff do not

need to enter anything extra during checkout.

Selling produce by weight (scale barcodes)

For loose produce sold by weight — tomatoes, apples, onions — a weighing scale in the back prints a sticker barcode that already contains the weight. The cashier scans that sticker at the till and the line is priced automatically. Here's how to set it up.

1. Configure the scale format once (admin). Go to **Settings** → **Weighed barcodes**, turn it on, and match the fields to your scale (prefix digits, where the item code and weight sit in the barcode, and the divisor). Most scales use prefix **2**. There is a built-in **Test a barcode** box — paste a label your scale printed and confirm it decodes to the right item code and weight before saving. See [Section 16 → Weighed barcodes](#).

2. Set up each weighed product.

- **Base Unit:** set to **kg** (or **g**).
- **Price:** enter the **price per kg** (this is what the barcode's weight is multiplied by).
- **PLU code:** enter the item code your scale uses for this product (e.g. **4011**). This is how a scanned label finds the right product. The same code you program into the scale.

3. At the till, the cashier just scans the sticker — the POS reads the weight, finds the product by its PLU, and adds the line at **weight × price/kg**. Nothing else to enter.

Weight-embedded vs price-embedded. Most scales embed the **weight** in the barcode and let the POS apply your price/kg — recommended, because the price stays controlled by the product record. Some scales instead embed the **total price** they already calculated; the app supports both (set the **Embedded value** option in Settings accordingly). With price-embedded labels the till reconstructs the quantity from your price/kg, which can round to the nearest cent, so weight-embedded is the cleaner choice when your scale offers it.

Note: stock for weighed products is tracked in the same unit you sell in (kg). Receiving 50 kg and selling 1.25 kg work exactly like any other inventory — fractional quantities are fully supported.

Selling a product in slices or portions (e.g., pizza by the slice)

When a product is **stocked as a whole unit but sold in fractions** — a pizza by the slice, a watermelon by the quarter, a cake by the piece — you have two ways to model it. Pick whichever fits your workflow.

Option A — Sale option on the whole-unit product (*simplest, recommended for fixed cut counts*)

Keep **one** product (e.g. “Pizza”) stocked in whole pizzas. Add a **sale option** on it for the slice:

1. Open the product in the **Products** screen.
2. In the **Sale options** section, tap **Add sale option**.
3. Fill in:

- **Package barcode** — assign a barcode specifically for the slice (different from the whole-pizza barcode).
- **Display name (Arabic / English)** — e.g. شريحة بيتزا / Pizza slice .
- **Sell unit** — slice .
- **Units of stock per sale** — the fraction of one whole pizza this slice represents:
 - 8-slice cut → 0.125
 - 4-slice cut → 0.25
 - 6-slice cut → 0.1667
- **Package price** — the per-slice selling price.

4. Save.

Scanning the slice barcode now adds one slice to the cart. Selling **8** slices deducts exactly **1** whole pizza from stock. Selling **4** quarter-slices deducts exactly **1** whole pizza. Selling **6** sixth-slices deducts ≈ 1 whole pizza (a microscopic rounding crumb is left behind — the POS treats it as zero, so you never get a phantom “0.000...1 pizza” lingering on the screen).

Tip: prefer 0.25 (4-slice) or 0.125 (8-slice) where you can — those are mathematically exact. $1/6 \approx 0.1667$ works fine, but $1/4$ and $1/8$ are exact.

If the cut count varies from pizza to pizza (today’s pizza is cut into 6, tomorrow’s into 8), add **one sale option per cut style** — *Slice (4-cut)*, *Slice (6-cut)*, *Slice (8-cut)* — each with its own barcode, multiplier, and price. The cashier scans whichever barcode matches today’s pizza. Don’t try to use one slice option for different cut sizes; that will under- or over-deduct inventory.

Option B — Composite product (when the slice is built from raw ingredients, not sliced from a pre-stocked whole)

Use this when there is no “Whole Pizza” sitting on a shelf — the slice is **assembled to order from raw ingredients** (dough, cheese, sauce, toppings):

1. Make sure each ingredient already exists as its own product with stock tracking turned **on**.
2. Create a new product (“Pizza Slice”). Toggle **Composite product** on.
3. In the **Components** section, add one row per ingredient with the **quantity per slice** — e.g. 0.10 kg dough, 0.05 kg cheese, 0.02 L sauce.
4. Save.

Selling one slice deducts each ingredient by its component quantity. The composite never tracks its own stock — its POS card shows the **derived availability** (the minimum number of slices you could still build given current ingredient stock). Refunds restore each ingredient proportionally.

You cannot nest composites in this version — a component must be a leaf product (no composite-of-composite). And a composite **must have at least one component** before you can save it.

Recipe cost and food-cost %

As soon as you add components, the editor shows two live figures beneath the recipe:

- **Recipe cost / unit** — what one composite costs to make: the sum of each component's cost × its quantity per unit. It updates as you edit the rows.
- **Food cost %** — the recipe cost as a share of the **before-tax selling price**. Use it to price the dish — most restaurants aim for roughly **28–35%**. It turns amber as the ratio climbs and red if the dish costs more than you're charging.

Example — a bowl of vegetable soup. Each vegetable is a stock-tracked product with a cost per kg; the recipe lists how much goes into one bowl:

Ingredient	Cost	Qty per bowl	Line cost
Tomato	4.00 / kg	0.20 kg	0.80
Onion	3.00 / kg	0.10 kg	0.30
Carrot	2.50 / kg	0.08 kg	0.20
Recipe cost / unit			1.30

Priced at **5.00** before tax, the **food cost %** is $1.30 \div 5.00 = 26\%$ — at or below the 28–35% most restaurants target. Sell 10 bowls and the app removes 2 kg tomato, 1 kg onion and 0.8 kg carrot from stock and books **13.00** as cost of goods, so the Profit report shows **37.00** gross margin on those bowls — not the 50.00 it would wrongly show if the dish had no cost.

You do **not** type a cost on the composite itself — it is always **derived from the recipe**. For the figure to be accurate, each ingredient must be a **stock-tracked product with a real cost**: keep ingredient costs current by recording deliveries as **purchase documents** (Section 8). The app keeps a quantity-weighted average cost per ingredient, so the recipe cost — and the dish's margin — follow automatically, including when an ingredient's price changes. The dish's computed cost also appears on its row in the **Products** list.

When you sell the composite, this recipe cost is booked as the dish's **cost of goods**, so the **Profit report** (Section 13) shows the dish's true margin instead of the ~100% it would show with no cost. Refunds reverse it.

Tip — waste and yield. The recipe deducts exactly what you enter. If real prep loses some to trimming, peeling, or spillage, enter a slightly higher quantity per unit (e.g. 0.22 kg tomato instead of 0.20 kg) so both the stock deduction and the cost match reality. Keep every ingredient in a **consistent unit** — if you buy onions in kg, write the recipe in kg (0.10), not grams.

Which option should I use?

Situation	Use
Pizza is pre-baked and cut at the counter from a whole pie that's already on the line	Option A (sale option)
Pizza is assembled to order from raw ingredients	Option B (composite)
You want to track each ingredient's stock separately for waste / cost reports	Option B (composite)
You sell <i>both</i> whole pizzas and slices from the same line of pizzas	Option A — the whole-pizza and the slice are two sale options on the same product, sharing one inventory pool
The cut count varies per pizza	Option A with one sale option per cut style

The same patterns work for cakes by the slice, watermelons by the quarter, large meat cuts by the kilo from a whole carcass, and any other “stock one, sell fractions” scenario.

Selling the same stock under different names (e.g. roasted vs grilled chicken)

Sometimes one stock item is **sold under two or more names** that the customer should see on the receipt — the same chickens sold as **Roasted chicken** and **Grilled chicken**, the same loaves sold as **Plain bread** and **Toasted bread**. There is **one product and one stock pool**; the names are just different ways to ring it up.

This uses the same **sale option** mechanism as the pizza slice above, with one difference: each option deducts **one whole unit** (no fraction), and the only thing that changes between options is the **name** (and, if you like, the price).

Example — roasted vs grilled chicken

1. Create or open one product, e.g. **Chicken** (دجاج). Stock it normally — say **40** whole chickens. Leave its base name as the plain item (“Chicken”).
2. In the **Sale options** section, tap **Add sale option** and fill in the first style:
 - **Package barcode** — leave it **blank** for a tap-only button, or scan/assign a code if you want the cashier to be able to scan it.
 - **Display name (Arabic / English)** — دجاج مشوي / Roasted chicken.
 - **Sell unit** — chicken (or piece).
 - **Units of stock per sale** — 1 ← each sale removes one whole chicken from stock.
 - **Package price** — the roasted price, e.g. 25.00.
3. Tap **Add sale option** again for the second style:
 - **Display name** — دجاج مقلي / Grilled chicken.
 - **Units of stock per sale** — 1.
 - **Package price** — the grilled price (can be the same as roasted, or different).

4. Save.

At the register, **Roasted chicken** and **Grilled chicken** now appear as two separate buttons. Selling either one prints **its own name** on the customer receipt and deducts **1** from the shared chicken stock. Sell 3 roasted + 2 grilled and your 40 chickens drop to **35** — the receipts read “Roasted chicken” and “Grilled chicken” exactly as the customer ordered.

Make sure “Units of stock per sale” is 1. That field is also used for box-packs (a box of 12 → 12), so it does **not** default to 1. For same-as-the-whole-unit styles like roasted/grilled, set it to 1 or you will over-deduct stock.

More than two names? Add as many sale options as you like — *Roasted, Grilled, Fried, BBQ* — all on the same Chicken product, all sharing the one stock pool, each with its own name and price.

Ingredients — items you never sell directly (rice, oil, bags)

A kitchen buys things it never sells on their own: a sack of rice, a tin of oil, foil trays, carrier bags. These are **ingredients**: they must exist in the catalog so recipes can deduct them and carry their cost, but they have no business appearing on the cashier's selling screen.

Tick **Ingredient (not sold directly)** in the product editor, and the item:

- **Disappears from the POS** — not in the product grid, not in search results, not via barcode scan. A cashier cannot accidentally ring up a kilo of raw rice.
- **Stays in the Products screen**, so you can still set its **cost** — and that cost is what feeds recipe cost and dish profit.
- **Is included in the catalog export**, along with the recipes, so importing the file on another machine reproduces a complete, working catalog.
- **Hides its category from the POS tab bar when that category holds nothing else**, so the cashier never sees an empty tab. The tab returns by itself the moment you put a sellable product in that category.

Ingredient ≠ archived. Archiving means “this is finished, remove it from the catalog”: it vanishes everywhere, the export included. An ingredient is a **live part of the catalog** that is merely hidden from the cashier. Never archive your ingredients — the export would lose them and the customer would import a broken catalog.

Most stores park ingredients in one category (e.g. **Stock items**) to make them easy to review, but the setting lives on the product, not the category — an ingredient can sit in any category you like.

Which tool do I use? Sale option, separate product, extra, or kitchen note

Four features look interchangeable, and picking the wrong one quietly corrupts either your stock or your receipts. The deciding question is: **do these things genuinely share the same stock?**

Situation	Use	Why
Same stock, sold differently — a 40-bottle water box	Sale option with <i>units of stock per sale</i> = 40	The box IS the same bottles, so selling one draws 40 off the count
Genuinely different stock — a Pepsi can vs a family bottle	Separate products	Sale options share one stock row ; selling a family bottle would drain the can count
A free choice that must show on both receipts — rice kind	Sale option at the same price, or an extra priced 0	Both print to the customer and to the kitchen
A paid add-on — extra cheese	Extra with its price	It is added to the line total automatically
An instruction for the cook only — no onions, spicy	Kitchen note	Prints on the prep ticket and never on the customer receipt

Practical differences worth knowing:

- **A sale option's label replaces the item name** on both receipts. So if you use one for a size or a style, write the full name — “Pepsi Family”, not just “Family” — otherwise the kitchen gets a receipt line it cannot act on.
- **A sale option is a single choice** per line; **extras are multi-select**. If the rule is “pick exactly one”, a sale option is the closer fit.
- **Extras are global** — they offer themselves on any line, not just the product you had in mind.
- **All sale options of a composite share one recipe**. A rice kind is a printed label only; it does not change which ingredient is deducted. To draw down different rices from their own stock, they have to be separate products with separate recipes.

Why Cost and Stock are Read-Only

The **cost** and **stock quantity** on the product page cannot be edited directly. They are updated automatically when you **post an Inventory Document** (see Section 8). This ensures there is always an audit trail for every stock movement.

How cost is calculated. Each time you post a purchase or restock, the product's cost becomes a **quantity-weighted average** of the stock you already had and the new stock you received. Example: 100 units on hand at 1.00 and you receive 10 units at 1.50 → the new cost is 1.045, not 1.25 — the larger existing quantity keeps the average stable. Cost is always stored **per single (base) unit**, even when you buy by the case, so profit is measured correctly however you sell the item.

New products start with a cost of 0 until their first purchase/restock is posted. While the cost is 0, every sale of that product shows as **100% profit**, which overstates your real profit. The app surfaces this in two ways so it never goes unnoticed:

- At the register, adding a zero-cost item shows a brief warning: *“No cost set — profit will be overstated until a cost is recorded.”*
- The **Profit** report shows how many products sold with no recorded cost (see Section 13).

To fix it, post a purchase or restock document for that product with the real unit cost.

For **composite products** the two fields behave differently. **Stock** is informational — selling a composite reduces each *component’s* stock, not its own, and the POS card shows derived availability instead. **Cost**, however, is **computed from the recipe** (the sum of each component’s cost × quantity per unit) and is booked as the dish’s cost of goods on every sale, so composite dishes are profit-tracked just like ordinary products. You never type it in; keep the *ingredients’* costs current and the dish’s cost follows. See [Section 5 → Selling a product in slices or portions](#).

Editing cost manually (permission required)

Posting an inventory document is the normal way to set cost, and it keeps a full audit trail. For corrections, a user with the **Edit Product Cost** permission (Managers and Admins by default) can also change the **cost** field directly when editing an existing product:

- The **Cost** field becomes editable on the product page; everyone else still sees it as read-only.
- This is a **manual override** of the weighted-average cost. It applies from that point on and does not create an inventory document, so prefer a purchase/restock document when stock actually moved.
- The permission is enforced beyond hiding the field — a save that changes the cost is rejected for anyone without it. Saving a product **without** changing the cost never needs the permission.
- **Stock** stays read-only here regardless; only inventory documents move stock.
- New products: the cost is still set by the first purchase/restock (the field on the *Add product* screen is not used for cost).

Archiving vs Deleting a Product

There are two ways to remove a product from the catalog, and they are **not** the same:

Action	What it does	Permission
Archive	Hides the product from the POS and product list but keeps all its history . Past orders, reports, and accounting stay intact. This is the safe default.	Add Products
Delete	Permanently removes the product. Only allowed for products that have never appeared in an order .	Delete Products

Why delete is restricted: a product that was ever sold is referenced by past orders. Deleting it would break that history, so the app blocks it. If you try to delete such a product, you are offered to **archive it instead** —

"{n} product(s) appear in past orders and can't be deleted. Archive them instead?"

Per-product: open the product, tap the **:** menu (top-right) and choose **Archive** or **Delete** (Delete only shows if you have the Delete Products permission).

Multiple products at once:

1. On the **Products** screen, tap **Select** in the top bar.
2. Tick the checkboxes on the products you want.
3. Tap **Archive** or **Delete** in the action bar.
4. Confirm. Any selected products that are in past orders are kept and offered for archiving instead — the rest are deleted.
5. Tap **Done** to leave selection mode.

A ready-made restaurant catalog to try

Rather than building a full menu from scratch to learn how products work, download a ready-made 65-product restaurant catalog and import it:

1. Get it from numu-tech.com/documentation → **Restaurant sample catalog**.
2. In **Products** → **Bulk actions** → **Import products**, pick the file.

Import it on a fresh install or a test terminal — importing adds these products to your existing catalog.

It is built the same way you would build yours, and shows three ideas:

What it shows	The example in the file
A recipe that deducts its ingredients	"Mandi Whole Chicken" at 42, with a 12-ingredient recipe: a kilo of rice, a chicken, two trays, oil, salt, bags... selling one deducts every one of them
A hidden ingredient	"White Rice" and "Aluminium Plate" — in the catalog because recipes need them, but never shown on the selling screen
A sale option	"Water 300ml" at 0.50 with a "Box of 40" option at 15 that deducts 40 bottles; and "Mandi Whole Chicken" with four rice choices at the same price

How it was built: the ingredients came first (rice, oil, bags...) with **Ingredient (not sold directly)** ticked; then the dishes, with **Composite product** ticked and each ingredient added with its quantity per dish; finally the **sale options** for sizes and rice choices. Details in [Ingredients](#) and [Which tool do I use?](#).

Importing Products in Bulk

You can create or update many products at once from a file.

1. On the **Products** screen, open the **Bulk actions** menu in the top bar.
2. (First time) choose **Import format help** to read the rules, and **Download CSV template** or **Download JSON template** to start from the correct layout.
3. Choose **Import products** and pick your **CSV** or **JSON** file.
4. Review the **import preview** (how many will be created/updated and any per-row errors), then confirm.
5. A summary shows how many were **Created**, **Updated**, and **Failed**.

Rules to know:

- **SKU is the match key.** A row whose SKU already exists **updates** that product; a new SKU **creates** a new one.
- **Required fields:** SKU, Arabic name, category, and a valid price.
- Imported **barcode and package rows replace** the existing barcode/package rows for the same SKU.
- When updating an existing product, a **blank cell keeps the current value** for stock, cost, PLU, and tax — only the columns you fill in change.
- Importing requires the **Add Products** permission.

Recounting all stock with Excel (full stocktake)

When you need to correct the on-hand quantity of **many** products at once — for example after a physical recount, or after items were sold while negative stock was allowed — the fastest way is to export the catalogue, fix the quantities in Excel, and import it back. The imported `stock_quantity` **overwrites** the current stock, so a product showing `-8` or `50` simply becomes whatever number you type.

 **Back up first — this overwrites live stock and cannot be undone per row.** Before you start, go to **Backup** → **Local Backup**, set a passphrase, and use **Share backup** to keep a copy off the machine. See [Backup and Restore](#).

1. On the **Products** screen, open the **Bulk actions** menu and choose **Export CSV**. Save the file and open it in Excel (or any spreadsheet app).
2. For **every** product, type the **physically counted quantity** into the `stock_quantity` column. Enter `0` for anything that is physically gone.
3. **Do not delete or blank other columns.** Edit only `stock_quantity` and leave the exported rows otherwise untouched.
4. Save the file as **CSV**.
5. Back on the **Products** screen, open **Bulk actions** → **Import products**, pick your file, check the **import preview**, then confirm.

⚠ **A blank stock_quantity cell keeps that product's current stock** — it will not be zeroed, but it also means a wrong number silently survives the recount. For a full stocktake every row must carry a real counted number; enter 0 for anything that is physically gone. Export first and edit that file — never build a fresh sheet with only a few columns.

Excel method vs. a Count document. The Excel round-trip is best for a whole-store recount because you edit everything in one sheet. If you only need to correct a handful of items, or you want an **auditable, reversible record** of the change, use a **Count** inventory document instead — see [Inventory Documents](#). A Count document sets each line to the counted quantity too, but you add products one at a time and the change is logged as a posted document.

Coming from another POS — fixing wrong stock and missing costs

A common situation after migrating: the products file imported fine and selling works, but the quantities the old system exported were wrong and many products arrived with **no cost (cost = 0)**. Until this is fixed, every sale of a zero-cost product is recorded with **zero cost of goods**, so the Profit report shows the full sale price as profit. Here is how to take back control, **in this order**:

Step 1 — Fix costs first. Every sale saves the product's cost onto the receipt line at the moment of sale, so each extra day of selling at cost 0 writes more wrong lines. Two ways:

- **Many products:** export the catalogue (**Bulk actions** → **Export CSV**), fill in the `cost` column, and import the file back. Rows match by SKU and update the existing products; a **blank cost cell keeps the current cost**, so you can fill in only the products you know and do the rest later.
- **A few products:** edit each product and set the cost directly — this needs the **Edit Product Cost** permission (see [Editing cost manually](#)).

Because these products currently have cost 0, the first real cost you enter becomes the cost as-is — nothing is averaged against the zero.

Step 2 — Fix the quantities. Do a physical count and correct the stock:

- Whole store: the [Excel recount](#) above.
- A handful of items, or when you want an auditable, reversible record: a **Count** inventory document — see [Inventory Documents](#).

Do **not** “fix” quantities with a Purchase or Restock document unless goods really arrived — those blend their unit cost into the product's average cost. Count and Adjustment documents change stock only and never touch cost.

Step 3 — Receive properly from now on. Enter every new delivery as a **Purchase** document with the real unit cost. That keeps the weighted-average cost true automatically, and every future sale books the right profit.

What about the sales already made at cost 0? The Profit report is self-correcting: a sale line that was saved with no cost is valued at the product's **current** cost when the report runs. So once Step 1 is done, past profit reports become realistic again — no re-entry needed. One caveat: those old sales are estimated at today's cost, not the (unknown) cost on the day they were sold.

6. Categories

Admin only

Categories group products together in the POS grid and can be linked to prep printers.

Creating a Category

1. Go to **Categories** and tap **+**.
2. Enter the Arabic and English names.
3. Optionally select a **Prep Printer** — orders for products in this category will print a prep ticket on that printer.
4. Tap **Save**.

Editing a Category

Tap an existing category to open the editor. Make your changes and tap **Save**.

Routing categories to a prep printer

Creating a prep printer in **Settings** → **Printers** is only **half** the setup — kitchen tickets only print when a category is **routed** to that printer:

1. Open **Categories** and edit the category whose items should print a kitchen ticket (e.g. *Bakery*, *Hot food*, *Frozen*).
2. In the **Prep printer** dropdown, pick your prep printer.
3. Save.

Repeat for each category that needs kitchen tickets. Categories left on **None** simply don't print prep tickets — that's fine for grocery items that don't need kitchen prep.

Why this matters: if a prep printer has zero categories routed to it, finalising a sale enqueues nothing for the kitchen — silently. To make the missing step discoverable, the **Settings** → **Printers** list now shows a red  icon on any prep printer with no category routed; hover it for the explanation. The warning disappears the moment one category points at the printer.

7. Inventory Overview

The inventory screen gives you a quick health check on your stock.

Dashboard Cards

Card	Meaning
Total Products	Number of products in the system
Low Stock	Products where stock has fallen below the minimum threshold
Out of Stock	Products with zero or negative stock
Expirable	Products with an expiry date configured
Stock Value	Total value of all stock on hand (quantity × cost)

Searching

Use the **search bar** at the top to filter by:

- Product name (Arabic or English)
- Barcode
- SKU

Type a few characters and the list updates instantly. You do not need to press Enter.

8. Inventory Documents

This is the most important section for keeping stock accurate.

What is an Inventory Document?

An inventory document is a **record of a stock movement**. Instead of directly editing stock numbers, every change goes through a document. This creates a clean audit trail — you can always look back and see exactly when, how much, and why stock changed.

Document Types

Type	When to use
Purchase	Receiving new goods from a supplier
Restock	Adding stock without treating it as a normal supplier purchase

Type	When to use
Waste	Writing off damaged, expired, or lost items
Adjustment	Manual correction when a physical count reveals a discrepancy
Count	Recording a full physical stock count
Reversal	Exactly undoes a previously posted document (created from the original)

Clerk Quick Rule

Use this simple rule every time:

- **Purchase** = use when stock came in from a supplier and there is an invoice, bill, or delivery document.
- **Restock** = use when you are adding stock back into the system without a normal supplier purchase document.
- **Draft document** = safe to edit directly.
- **Posted document** = do not edit directly; create a **Correction** or **Reversal** instead.

Easy Examples

Situation	What to choose
New goods arrived from a supplier	Purchase
Supplier invoice or delivery note is available	Purchase
Free replacement items or bonus stock	Restock
Stock was found in storage and added into the system	Restock
Draft document has the wrong quantity, cost, or expiry	Edit the draft directly
Posted document has a small mistake	Create Correction
Posted document is completely wrong and should be undone	Create Reversal

Creating a New Document

1. Go to **Inventory** → **Documents** and tap **+**.
2. Select the **document type**.
3. Optionally enter a **supplier** (for purchases) — type a new name or select an existing one.
4. Optionally enter a **note**.
5. The document is saved as a **Draft** automatically as you work.

Counting? Jump to the fast window. When the type is **Count**, a **Scan to count** button appears under the header. Tap it to continue the *same* document in the barcode-driven **Scan to count** window — the draft is saved first, so nothing is lost.

Browsing and Filtering the Documents List

Each row shows the **document number** (e.g. #142) next to its type and status, so you can refer to a specific document unambiguously.

Above the list you can filter by:

- **Type** and **Status** — e.g. only posted purchases.
- **Title** and **Supplier** — match the document's own fields.
- **Contains product** — type a **product name or barcode** to show only documents that have a line for that product (handy for “which purchase did this item come in on?”).
- **Date range** — narrow to a period.

A document that another document has already **reversed** carries a red **Reversed** badge, so you can see at a glance that its stock effect has been undone.

Adding Product Lines

By barcode scan:

Scan the barcode. If there is a single match, the product line is pre-filled automatically. Fill in the quantity and cost, then tap **Add Line**.

By name or SKU:

Type in the search field. If there are multiple matches, a list appears below. Tap the product you want to select it, then fill in the quantity and cost.

On purchases and restocks, *Add Line* stays disabled until you enter a cost. The moment you type a positive **Unit Cost** or **Total Cost** the button enables; clear it and it disables again. This stops a zero-cost line being added in the first place. (Waste, count, and adjustment documents don't require a cost, so the button isn't gated on them.)

Quantity and Cost Fields

Field	Description
Quantity	How many units you are recording (in the scanned/selected unit)
Unit Cost	Cost per single base unit
Total Cost	Optional — if you know the total cost for the line, enter it here and the unit cost is calculated automatically

Live calculation: When you change *Total Cost* or *Quantity*, *Unit Cost* updates in real time.

Stock Formula

Stock added = Quantity × Barcode multiplier

Example: You scan the box barcode (multiplier = 12) and enter quantity = 5. Stock added = 60 base units.

Cost Formula

New average cost = (Old cost + Unit cost) ÷ 2

If the product has never had a cost set (old cost = 0), the unit cost you enter is used directly.

Posting a Document

When all lines are correct, tap **Post Document**.

- The document status changes to **Posted**.
- Stock quantities and average costs are updated immediately.
- Expiry batches are updated immediately for stock-in and stock-out flows.
- **A posted document cannot be edited.** This protects the audit trail.

Purchase and restock lines need a unit cost. Posting is blocked if any line on a purchase or restock document has a unit cost of 0. This guarantees the product’s cost — and therefore your profit figures — stays accurate. Enter the cost per line, or type the line total and let the app divide it by the quantity. (Waste, count, and adjustment documents are not affected.)

Choosing how a purchase was paid (required)

When you post a **purchase** or **restock** that has a value, the app asks **how it was paid before it finishes posting**. This is mandatory — you cannot post the document without making a choice — so every purchase always records its payment type. This keeps both your cash drawer and your bank balance accurate.

Choice	When to use it	What it does
Cash	You paid on the spot with cash from the drawer.	Marks the document Paid . If a shift is open, the amount is deducted from the drawer’s expected cash at close (see Section 11).
Card	You paid on the spot by card or bank transfer.	Marks the document Paid . The money settles through the bank, never the drawer, and shows up in the Bank reconciliation report (Section 13).

Choice	When to use it	What it does
On account	You'll pay the supplier later.	Leaves the document Unpaid with a due date , so it appears in the Payables aging report and the overdue reminder. Settle it later with Record payment (see below). <i>Only available when a supplier is selected — you can't owe an unnamed seller.</i>

No supplier? You must pay now. A purchase with no supplier (a “cash & carry” run to the market) is always paid on the spot, so only **Cash** and **Card** are offered — there is no on-account option.

On account needs a due date. If you choose **On account**, pick the due date in the dialog before you can confirm.

Fast recount with a barcode scanner (Scan to count)

For a full physical recount, the **Scan to count** screen is much faster than adding count lines one field at a time. Connect a **barcode scanner** (USB or Bluetooth — it behaves like a keyboard, no driver or setup needed) to a tablet or phone and walk the shelves scanning items.

Where: **Inventory** → **Documents** → **Scan to count** (the button next to *New Document*). It needs the **stock-count** permission, the same as a Count document.

Pausing a count? Tap **Save draft** and come back later — a count saved as a draft **reopens right back in this Scan to count window** when you tap it in the Documents list (other document types open in the normal line editor). You can also open this window from a Count document's **Scan to count** button.

Two ways to work — pick from the toggle at the top:

Mode	How it works	Best for
Type quantity	Scan an item, type the counted amount on the keypad, press Enter. Focus jumps straight back to the scan box for the next item.	Counting a shelf and entering the total per product.
Scan to add	Each scan adds one (or one pack) to that item's running count. No typing.	Passing every physical unit across the scanner.

As you go:

- Each product shows **System** → **Counted** with a coloured **variance** chip (e.g. +3 or -5), so a wrong count is obvious immediately.
- An unknown barcode gives a **beep** and a warning instead of adding anything.
- Scanning the same product again **edits its one line** — it never creates duplicates. Tap any item in the list to fix its number, or remove it.

No scanner? Type the name. The scan box doubles as a **search box**: type part of a **product name** or **SKU** and matching products appear below — tap one to count it. Pressing **Enter** on text that isn't an exact barcode also falls back to a name/SKU search, so you don't have to scan every item by barcode.

When the count is complete, tap **Post**. Just like a Count document, this **sets each product's stock to the number you counted** — including pulling oversold items back up from negative. Products you never scanned are left untouched, so to recount the whole store, scan every item (enter 0 for anything that's physically gone).

⚠ Counting on a second device needs its own app licence

If you want to do the recount on a **separate tablet or phone** (for example, walking the aisles while the main till keeps serving customers), that device needs **its own purchased copy of the app** — each device requires its own licence. Install and activate the app on the counting device, then connect it to the main till over **Multi-Terminal Sync (LAN)** — see [Section 23](#). Sync makes your product catalogue available on the counting device and carries the posted count back to the main till automatically.

You do **not** need a second device or a second licence if you simply run Scan to count on the till you already use.

Excel or scanner? Use **Scan to count** when you have a barcode scanner and want to walk the shelves; it leaves a proper posted Count document as an audit trail. Use the [Excel recount](#) when you'd rather edit every quantity in a spreadsheet in one go. Both set stock to the counted value.

Stock count report (variance)

A count's real value is the **variance** — the difference between what the system expected and what you actually counted. The app records each item's system stock **at the moment you count it**, so this comparison survives even after you post the count (posting overwrites the live stock to match your count).

On the document: open any Count document and every line shows **Counted**, **System** (the expected stock at count time), and a coloured **±Variance** chip — green for a surplus, red for a shortage.

Printable report: on a Count document, tap the **report icon** in the top bar (*Stock count report*) to generate a PDF you can **print or save**. It lists each item with **System / Counted / Variance**, plus a summary of how many were **Surplus**, **Shortage**, and **Matched**, and the net variance — a ready audit sheet for the recount.

Older counts. Counts posted before this feature existed didn't record the snapshot, so their variance shows as —. Every new count captures it from the first scan.

Deleting a Draft

Requires the *Manage Inventory Documents* permission. Users without it won't see the trash icon at all.

If a draft document was created by mistake (test data, wrong supplier, abandoned entry), you can delete it directly from the documents list:

- On any **draft** card a red **trash icon** appears next to the status chip.
- Tap it → a confirmation dialog opens → tap **Delete draft** to confirm.
- The draft and all its lines are removed in one transaction.

A supplier payment you already recorded is kept. If you used **Record payment** on the draft (you paid the supplier up front, then abandoned the half-entered document), deleting the draft does **not** erase that payment. The cash genuinely left the drawer, so it survives as a standalone supplier payment and stays deducted from your shift's expected cash — only the document and its lines are removed. (Without this, deleting the draft would make the till look like it's holding cash that's actually gone.)

Posted documents cannot be deleted — only drafts can. To undo a posted document, use Reversal or Correction (see below).

Supplier invoice payments

For documents of type **purchase** or **restock**, a **Supplier invoice** section appears on the editor with:

- **Invoice number** — the number from the supplier's paper invoice. Free-form text.
- **Due date** — when payment is owed. Optional. Picked from a calendar.
- **Total / Paid / Outstanding** — auto-computed read-only stats.
- **Payment status** chip — **Unpaid** / **Partially paid** / **Paid**.
- **Record payment** button — enabled when there's an outstanding balance **and** the current user has the *Manage Suppliers* permission. It works both on a **posted** invoice (settling an on-account balance over time) **and** on a **draft** purchase/restock you're still entering (paying the supplier up front — see *Paying cash before you finish a big delivery* below). Users without the permission see the button disabled.

Paid now vs. on account. If you chose **Cash** or **Card** at posting time (see *Choosing how a purchase was paid* above), the invoice is already fully paid and there's nothing more to do here. The **Record payment** flow below is for invoices left **On account** — log each payment until the balance reaches zero — or for paying a supplier up front while a purchase is still a draft.

Paying cash before you finish a big delivery. A supplier drops off a large order and wants paying on the spot, but keying in every line will take a while. **Don't wait until you post the document** — the cash has already left the drawer, and an unposted draft would otherwise leave your till looking *short* at close. While the purchase is still a **Draft**, enter the supplier name, add at least one line, then tap **Record payment** and log the **Cash** you handed over. The payout attaches to your open shift immediately, so the drawer still balances at close even though you finish and post the document later. When you do post it, the app sees it's already paid and skips the payment prompt.

To record a payment against a supplier invoice — **posted, or a draft you're still entering**:

1. Open the purchase/restock document — a **posted** invoice, or a **draft** with the supplier set and at least one line added.
2. Tap **Record payment**.
3. Enter the **payment amount**, choose **Cash** or **Card**, and add an optional reference (cheque number, transfer reference, etc).
4. Tap **Save**. The `amountPaid` on the document goes up, the **Payables aging** report updates automatically, and the document's payment-status chip flips from `Unpaid` → `Partially paid` (or to `Paid` if the outstanding hits zero).

Cash vs Card — and the till. Choose **Cash** when you hand the supplier money out of the drawer, and **Card** when it settles through the bank (card or transfer). A **Cash** payment recorded while your shift is open is automatically deducted from the drawer's expected cash at close, so the till still balances. A **Card** payment never touches the drawer. (See Section 11.)

The document list view shows a small payment-status pill on each posted purchase/restock card so you can scan for unpaid invoices at a glance.

Overdue reminder. When an on-account purchase passes its due date and is still unpaid, a **red count badge** appears on the **Documents** item in the side navigation, showing how many supplier invoices are overdue. Open the **Payables aging** report (Section 13) to see exactly which ones and settle them.

Overdue card on the Documents page. The Documents page itself shows an **Overdue** summary card that mirrors the nav badge. **Tap it to filter the list to just the overdue invoices** (the card gets a red ring while the filter is active); tap again to clear. Overdue rows also carry a red **Overdue** pill so they stand out, and when nothing is overdue the card shows an **all settled** message.

Why this is separate from General Expenses: paying a supplier for inventory you'll resell is **not** an operating expense. It's a transfer from cash → inventory (an asset), which becomes Cost of Goods Sold only when you sell each item. Recording it as a General Expense would double-count it once you sell the goods. The invoice payment flow correctly leaves COGS to do its work at sale time.

Correcting a Posted Document

If you posted a document with an error, you have two options:

Where to find these actions:

Open the posted document from **Inventory** → **Documents**. You can use the action menu on the posted document card from the list, or open the posted document itself and use the **Create Correction** / **Create Reversal** buttons inside the document screen.

Reversal

1. Open the posted document.
2. Tap **Create Reversal**.
3. A new draft is created that is the exact negative of the original.
4. Review and tap **Post** — this undoes every stock/cost movement from the original document.

A document can be reversed only once, and a reversal document itself cannot be corrected or reversed — it exists solely to undo its original. If you need to redo the change after a reversal, post a fresh document. Corrections, by contrast, can be created against any posted document.

Correction

1. Open the posted document.
2. Tap **Create Correction**.
3. A new editable draft is created pre-filled from the original lines.
4. Adjust the quantities and costs to what they *should have been*.
5. Post the correction document.

Tip: Always create a Reversal first if you need to completely undo something. Use Correction if you only need to adjust a few values.

Expiry Dates During Inventory

For **Purchase** and **Restock** documents, you can enter an expiry date on each incoming stock line.

- The expiry date is stored with that incoming batch.
- When stock is sold, wasted, counted down, refunded, or reversed, the system updates the tracked batches automatically.
- The expiry report and dashboard alerts use the active batches that are still in stock.

Clerk rule:

If the received stock has an expiry printed on the package, enter it on the line before posting. If the item does not expire, leave the field empty.

9. Suppliers

Suppliers are used for **Purchase** documents and for tracking payments made to vendors.

Adding a Supplier

When creating or editing a purchase document, tap the **Supplier** field and either:

- Select an existing supplier from the dropdown.
- Type a new name — it is saved automatically when you save or post the document.

Suppliers Screen

There is now a dedicated **Suppliers** screen in the navigation rail.

Use it to:

- review the supplier list
- open an existing supplier
- record a supplier payment

Recording a Supplier Payment

1. Go to **Suppliers**.
2. Tap the supplier you want to pay.
3. Enter the **payment amount**.
4. Choose the **payment method** — **Cash** (paid from the till) or **Card** (paid through the bank, by card or transfer).
5. Optionally enter a **reference or note**.
6. Tap **Save**.

The payment is stored immediately. If the payment is linked to a purchase document, the document's payment status is updated automatically in the background.

Cash payments and the drawer. A **Cash** payment recorded while your shift is open is treated as money leaving the till: it is automatically subtracted from your expected drawer cash when you close the shift, so the count still matches. A **Card** payment does not affect the drawer. If no shift is open when you record a cash payment, it is simply logged and does not affect any drawer.

10. General Expenses

General Expenses are used for **operating costs that do not affect stock**.

Examples:

- rent
- electricity or water bills
- salaries
- internet or phone bills
- transport
- maintenance
- petty cash purchases

Who Can Use It

- **Admin** — always.
- **Cashier** — only if the admin has enabled the **Manage General Expenses** permission for that user.

What Is Different Now

Expense categories are **admin-managed**. Staff do not type category names manually anymore.

- Admins create and maintain the list of available expense categories.
- Users only **choose** a category from that predefined list when recording an expense.
- This keeps reports consistent and prevents duplicate spellings such as **Rent** , **rent** , or **Shop Rent** .

When to Use General Expenses vs Inventory Documents

Use this rule:

- If the spending brings products into stock or changes stock quantity, use **Inventory Documents**.
- If the spending is a business operating cost and should **not** change stock, use **General Expenses**.

Situation	Where to record it
Supplier invoice for incoming stock	Inventory → Documents → Purchase
Bonus or replacement stock	Inventory → Documents → Restock
Monthly rent	General Expenses
Electricity bill	General Expenses
Salary payment	General Expenses
Shop repair cost	General Expenses

Managing Expense Categories (Admin)

1. Go to **General Expenses**.
2. Tap **Manage Categories**.
3. Add a new category or edit an existing one.
4. If a category should no longer be used, disable it.

Disabled categories stay on existing old expense records, but users should no longer choose them for new entries.

Creating a New Expense

1. Go to **General Expenses**.
2. Tap **New Expense**.
3. Fill in:
 - **Title**
 - **Expense Category** — choose from the admin-managed list
 - **Expense Amount**
 - **Payment method** — **Cash** (paid from the till) or **Card** (paid through the bank)
 - **Expense Date**
 - **Note** (optional)
4. Tap **Save**.

Cash expenses and the drawer. Just like supplier payments, a **Cash** expense recorded during an open shift is deducted from that shift's expected drawer cash at close, so petty-cash purchases out of the till don't make the drawer look short. **Card** expenses don't affect the drawer. The payment method is set when the expense is created; editing an expense later does not move it to a different shift.

Editing or Deleting an Expense

- Tap an expense row to edit it.
- Use the **Delete** action on the row to remove it permanently.

Filtering Expenses

You can filter the list by:

- search text
- category
- date range

The summary cards show:

- total number of expenses

- total spend
 - number of categories in the current filtered result
-

11. Shifts

A shift tracks cash flow during a work period.

Opening a Shift

1. Go to the **Shifts** tab.
2. Tap **Open Shift**.
3. Enter the **opening cash** amount (the float in the till).
4. Tap **Open**.

You can now process sales.

Requiring a shift before selling

By default, if a cashier rings up a sale without opening a shift, the app opens one automatically with a **zero opening float** — which leaves the drawer reconciliation wrong. To force a deliberate start instead, an admin can turn on **Settings** → **Require an open shift before selling**.

When it's on, a cashier with no open shift who tries to **sell or refund** is asked for the **opening cash** right then; entering it opens the shift and the sale continues. Cancelling the prompt cancels the sale — nothing is rung up without an open shift and a recorded float. (With the setting off, the old auto-open behaviour is unchanged.)

X Report (Mid-Day Summary)

- Tap **X Report** while a shift is open.
- Shows a live summary so far: total sales, cash sales, card sales, and order count. When any occurred this shift it also lists **on-account cash** (customer cash settlements), **cash payouts** (cash supplier/expense payments), and any **cash in / cash out** drawer movements — the same cash figures you reconcile against at close.
- The shift stays open — this is for information only.
- You can print it: tap **Receipt** (thermal receipt printer) or **A4** (PDF you can print, save, or share). See **Printing shift reports** below.

Cash In / Cash Out (Drawer Movements)

Sometimes cash enters or leaves the drawer for reasons that are **not** a sale, a supplier payment, or an expense — for example the owner takes cash to the bank, drops large notes into the safe, or adds change to top up the float. Record these so the till still balances at close.

1. While a shift is **open**, tap **Cash in/out** on the shift card.
2. Choose the direction:
 - **Cash out** — money leaving the drawer (owner withdrawal, bank/safe drop).
 - **Cash in** — money added to the drawer (float top-up, returning cash).
3. Enter the **amount** and an optional **reason**.
4. Tap **Save**.

Each movement is logged with who recorded it, and it adjusts the shift's expected cash at close: **Cash in** raises expected drawer cash, **Cash out** lowers it.

Cash movements are not expenses. Unlike a cash supplier payment or a general expense, a drawer movement never appears in expense reports or profit figures — it only moves cash in or out of the till. Use a **general expense** when you actually spent money on something; use **Cash out** when you simply removed cash from the drawer.

Permission required. This button only appears for users with the **Record Cash In/Out** permission (Managers, Shift Leads, and Admins by default). A plain cashier does not see it.

Closing a Shift

1. Tap **Close Shift**.
2. Count the physical cash in the till and enter the **counted cash** amount.
3. The system shows:
 - **Cash payouts** — total cash taken from the till during this shift: cash supplier payments, cash expenses, and cash-paid “cash & carry” purchases (a purchase with no supplier, paid in cash). Shown only when there are any.
 - **On-account cash** — cash a customer handed over during this shift to settle their account balance (see §19). It went into the drawer, so it raises expected cash. Shown only when there is any.
 - **Cash in / Cash out** — manual drawer movements recorded during the shift (shown only when there are any).
 - **Expected cash** = opening cash + cash sales + on-account cash – cash payouts + cash in – cash out
 - **Variance** = counted cash – expected cash
 - A positive variance means more cash than expected; negative means a shortfall.
4. Optionally tap **Receipt** or **A4** to print the closing slip (the figures above plus expected/counted/variance) before you finish — see **Printing shift reports** below.
5. Tap **Close** to finish the cash-count step.

Why cash payouts matter here. If you pay a supplier, a petty-cash expense, or a no-supplier cash & carry purchase in **Cash** during the shift, that money physically leaves the drawer. The close screen

subtracts those cash payouts from the expected cash so the till still balances against what you counted. Payments made by **Card** are ignored here because the money never came from the drawer.

Closing the shift records the counted cash, expected cash, and variance. The shift is closed, but it is **not yet Z-closed**.

Z Close (End-of-Day Finalization)

After a shift is closed, an additional **Z Close** step is available.

1. Go to **Shifts**.
2. Find a shift with status **Closed** that is not yet Z-closed.
3. Tap **Z Close**.
4. Confirm the action.
5. The system finalizes the shift and shows the Z summary, including:
 - total sales
 - cash sales
 - card sales
 - on-account cash (cash settlements of customer balances collected this shift)
 - cash payouts (cash supplier/expense payments taken from the till)
 - cash in / cash out (manual drawer movements)
 - the drawer reconciliation (opening balance, expected cash, counted cash, variance)
 - refunds
 - VAT collected
6. Print it if you need a copy: tap **Receipt** or **A4** (see **Printing shift reports** below).

Important:

Z Close is the final end-of-day step. Use **X Report** for an in-progress summary, **Close Shift** for the cash count, and **Z Close** when you are ready to finalize the closed shift.

Printing shift reports

The **X Report**, the **closing slip** (on the Close Shift dialog), and the **Z report** can each be printed two ways:

- **Receipt** — sends the report to the configured **receipt printer** (the same thermal printer used for sale receipts). If no receipt printer is set up, you'll see *"No receipt printer configured"* (set one in **Settings** → **Printers**, §16).
- **A4** — produces an **A4 PDF**. On desktop (Windows/Mac) a dialog lets you **print** to any system printer or **save as PDF**; on mobile it opens the **share** sheet. The PDF is laid out in your document language and includes the store name and logo.

Printing never changes or closes the shift — on the Close Shift dialog you can print first and then tap **Z / Close**.

Who Sees What

- **Admins** see all shifts from all cashiers.
- **Cashiers** see only their own shifts.

12. Orders and Refunds

Finding Past Orders

1. Go to **Orders**.
2. Use the **date range** filter (From / To) to narrow down by date.
3. Use the **status** filter to show All, Completed, Partially Refunded, or Refunded orders.

Order Statuses

Status	Meaning
Completed	Order was paid in full, no refund issued
Partially Refunded	Some items were refunded
Refunded	The entire order was refunded

Full Refund

1. Open an order with status **Completed**.
2. Tap **Refund**, then tap **Full refund** to select every remaining item.
3. Under **Refund method**, choose how the money is returned — **Cash**, **Card**, or a **split** (enter the card portion; the rest is returned as cash). This defaults to how the order was originally paid, so the usual case is a single tap.
4. Tap **Confirm**.
5. The order status changes to **Refunded**.

Partial Refund

1. Open an order.
2. Tap **Refund** (this opens the line-by-line refund dialog).
3. For each line, adjust the **quantity to refund** (default is 0 — only set quantities for the items being returned).

4. Under **Refund method**, choose how the money is returned — **Cash**, **Card**, or a **split** (enter the card portion; the rest is returned as cash). This defaults to how the order was originally paid.
5. Tap **Confirm**.
6. The order status changes to **Partially Refunded**.

How the refund method affects your drawer: a **Cash** refund takes money out of the till, so it lowers your shift's expected cash at close; a **Card** refund goes back to the card and does not touch the till. You pick the method that matches how you actually hand the money back — it does not have to match how the sale was paid. A refund of an **on-account (credit)** sale shows no method chooser: it simply reduces the customer's outstanding balance.

Correcting an Order's Payment Type

If a sale was rung up under the wrong payment method — for example **Cash** when the customer actually paid by **Card** — you can fix it on the order itself, without doing a refund-and-resell. This keeps your inventory and refund counts clean while still making the cash drawer and bank totals correct.

1. Open the order.
2. Tap **Correct payment type**.
3. Choose **Cash**, **Card**, or a **split** (enter the card portion; the rest is treated as cash).
4. Confirm. The drawer's expected cash and the card total update automatically.

Only while the shift is still open. You can correct a payment type only while the shift that originally rang up the order is still open. Once that shift is closed and reconciled its drawer is locked and the option disappears — so make corrections before you close out. An order rung up on **another terminal** must be corrected on that terminal.

A few limits. Corrections cover **Cash, Card, or a cash/card split** only. **On-account (credit)** orders can't be changed here, because that would touch the customer's ledger. The cash and card amounts you enter must add up to the order total.

Who can do it: the same staff who can issue refunds — **Admins always**, and any cashier granted the *Refund Orders* permission. Every correction is recorded in the background audit log.

Who Can Refund

- **Admin** — always.
 - **Cashier** — only if the admin has enabled the *Refund Orders* permission for that user.
-

13. Reports

Admin only

Accessing Reports

Go to **Reports** in the navigation rail.

How the Reports Screen Works

The Reports screen is a single hub for all report types.

- 1. Use the **From** and **To** filters at the top to choose the reporting period.
- 2. Tap a report chip in the **Report Catalog** row to switch between report types.
- 3. Review the summary cards and detail tables for the selected report.
- 4. Tap **Print** to export the currently selected report as a PDF that you can share or save.

Available Reports

Report	What it shows
Sales Summary	Net sales, cash sales, card sales, order count, plus breakdowns by item, category, and hour
Refund Report	Total refunded amount, number of refund transactions, affected orders, returned quantity, and refund breakdowns by item, cashier, and day
VAT Report	Net sales, taxable sales, exempt sales, collected VAT, net VAT, VAT by day, and taxable sales by category
Cashier Performance	Net sales, refunds, orders, active cashiers, and a comparison table showing shifts, orders, refunds, and average ticket per cashier
Inventory Movement	Movement count, net quantity movement, sold quantity, returned quantity, purchased quantity, waste quantity, plus breakdowns by reason and by top-moving products
Supplier Purchases	Total supplier spend, posted purchase/restock document count, supplier count, spend by supplier, and spend by document type
General Expenses	Total operating expenses, expense count, category count, plus breakdowns by category and by day

Report	What it shows
Expiry Report	Tracked products with nearest expiry dates, expired items, items expiring soon, healthy items, and stock on hand for each tracked product
Profit	Revenue, historical cost, profit, margin, general expenses, net profit, net margin, plus profit breakdowns by item and category
Bank reconciliation	What should be in the bank from card activity: card sales, card purchases, card expenses, and the resulting Expected in bank , plus breakdowns of card purchases by supplier and card expenses by category
Payables aging	Outstanding supplier purchases grouped by how overdue they are (current, 1–30, 31–60, 61–90+ days) with the total payable. A snapshot of <i>now</i> — it ignores the date range.

Date Range Filter

Use the **From** and **To** date pickers at the top to set the reporting period.

- If you leave both dates empty, the report uses **all available data**.
- If you set only **From**, the report starts from that date forward.
- If you set only **To**, the report includes all data up to that date.
- Tap the **clear filter** icon to remove both dates.

Exporting Reports

Each report has a **Print** button in the summary card.

When you tap it, the app creates a PDF for the currently selected report with:

- the report title
- the store name
- the generated date and time
- the active date range
- summary metrics
- the visible detail tables
- report notes when applicable

Important Notes

- **Expiry Report** reflects the earliest expiry among the active inventory batches still in stock for each product.

- **Profit** uses the unit cost stored on each sale line at the time of sale. Older legacy rows may still fall back to the current average product cost if they were created before historical cost tracking was added.
- **Products with no recorded cost** report as 100% margin and overstate profit. The Profit report shows a note counting how many such products sold in the period — set a cost for them via a purchase document (see Section 5).
- **General Expenses** are separate from supplier purchases and inventory documents. They do not change stock.
- **Net Profit** in the profit report is calculated as: `Profit - General Expenses` for the selected date range.
- **Bank reconciliation** computes **Expected in bank = card sales – card purchases – card expenses** for the date range. Card refunds are already netted out of card sales. “Card purchases” covers both supplier payments made by card and no-supplier purchases paid by card. Compare the *Expected in bank* figure with your real bank statement to catch a misbehaving card terminal or a missing deposit.
- **Payables aging** is a live snapshot of unpaid supplier purchases as of today, so the date-range filter does not affect it. Settle invoices from each document’s **Record payment** action (Section 8).

14. Cashier Activity Log

Admin only

This screen is the day-to-day review screen for cashier activity.

What is Tracked

Activity	Description
Product Added	A cashier added a new product
Refund	A cashier issued a refund
Waste	A cashier posted a Waste inventory document
Restock	A cashier posted a Restock inventory document
Adjustment	A cashier posted an Adjustment document
Count	A cashier posted a Count document

Filtering

Use the **date** filter to view activity for a specific period.

Finance note:

The system now also records finance-sensitive actions such as order creation, refunds, payment-type

corrections, shift open/close, and settings changes in the background for audit purposes. Staff do not need to do anything extra for this.

15. Users

Admin only

Adding a New User

1. Go to **Users** and tap **+**.
2. Fill in:
 - **Name** (displayed on the login screen)
 - **Role**: Admin or Cashier
 - **PIN**: 4–6 digits
3. If the role is **Cashier**, set the permission toggles:
 - Add Products
 - Refund Orders
 - View Sales History
 - Manage Inventory Documents
 - Manage General Expenses
 - View Reports
 - Manage Customers
 - Export Accounting
4. Tap **Save**.

Editing a User

Tap an existing user to open the editor. You can change the name, PIN, role, and permissions.

Resetting a PIN

Open the user, enter a new PIN in the PIN field, and tap **Save**.

Disabling a User

Toggle the **Active** switch off and tap **Save**. A disabled user cannot log in but their history is preserved.

16. Settings

Admin only

Business Info

Tap **Business Info** to open the store details screen.

Field	Description
Business Logo	Image printed at the top of receipts and A4 invoices. Use a square image with a white or transparent background. If no logo is set, receipts print without one.
Store Name (Arabic)	Printed on Arabic receipts and invoices
Store Name (English)	Printed on English receipts and invoices
Phone Number	Contact number printed on receipts and invoices
Email	Store email printed on receipts and invoices
Address	Store address printed on receipts and invoices

Tax Settings

Tap **Tax Settings** to configure VAT, tax rules, currency display, and compliance options.

Setting	Description
VAT Number	15-digit VAT registration number (required for ZATCA compliance)
ZATCA QR Code	Toggle to include the Phase 1 ZATCA QR code on receipts
E-invoicing (ZATCA)	Turns on ZATCA Phase 2 e-invoicing and reveals the ZATCA onboarding and E-invoice status screens. Off by default — see section 22
Currency Code	Three-letter code used by the store, such as SAR , USD , or EUR
Currency Symbol	Symbol printed and shown in the app, such as ر.س , \$, or €
Tax Rates	The list of tax rules available for products. Admins can add or edit rates and choose a default rate

Managing tax rates:

1. Open **Settings** → **Tax Settings**.
2. In the **Tax Rates** section, tap **+** to add a new rate or tap an existing rate to edit it.
3. Enter the rate name and percentage.
4. Enable **Default Tax Rate** if this should be used automatically for most products.
5. Tap **Save**.

Extras

Tap **Extras** to manage the priced add-ons cashiers can attach to a cart line (extra cheese, a double patty, an extra sauce, and so on). Each extra has Arabic and English names and a **price** that is added to the line when selected.

Managing extras:

1. Open **Settings** → **Extras**.
2. Tap **Manage extras**.
3. Tap **Add extra** to create one, or the pencil to edit an existing one.
4. Enter the **Name (Arabic)**, **Name (English)**, and **Price**.
5. Use the up/down arrows to reorder how extras appear in the line editor, or the block icon to disable one you no longer offer.
6. Tap **Save**.

Notes:

- A new store starts with a few examples (Cheese, Double, Extra sauce) that you can edit or remove. Existing stores upgrading to this version start with an empty list — nothing is added to your menu automatically.
- Changing an extra's price only affects **future** sales. Past orders keep the price that applied when they were rung up.
- Defining or changing an extra requires the **Manage Settings** permission.

Printers

The app supports four types of printers:

- **Receipt Printer** — prints customer receipts.
- **Prep Printer** — prints kitchen/prep tickets for specific categories.
- **Invoice Printer** — prints A4 tax invoices.
- **Barcode / Label Printer** — prints product shelf labels and price tags.

Each printer connects in one of two ways:

- **Network** — reached over the LAN by **IP address** and **port** (default 9100).

- **USB / System printer** — a printer installed on this computer's operating system; the app sends to it through the OS print spooler.

Adding a printer:

1. Tap **Add Printer** and choose how to add it — **Scan network**, **Enter manually**, **Scan QR code**, **Use template**, or **USB / System printer**.
2. For a network printer, enter the **IP address** and **port**; for a USB printer, pick it from the system printer list.
3. Choose the printer type — **Receipt**, **Prep**, **Invoice**, or **Barcode label**.
4. For **Receipt** and **Prep** printers, choose the **paper width** — 80 mm or 58 mm .
5. For a **Barcode label** printer, set the label hardware instead:
 - **Command language** — ESC/P0S (raster) , ZPL – Zebra , or TSPL – TSC / Godex . Match this to your printer's model.
 - **Label size** — the **width**, **height**, and **gap** of your die-cut label stock, in millimetres.
 - **Print density** — the printer's resolution in DPI (commonly 203).
6. Give it a name and tap **Save**.

A barcode/label printer is described by its label stock and command language, **not** by receipt paper width — so the paper-width field does not apply to it.

Managing an existing printer:

- Tap the printer actions menu to:
 - **Edit** the printer
 - **Set as Default** — the receipt, invoice, and barcode/label roles each have their own default
 - **Test Print** (receipt/prep/invoice) or **Print test label** (barcode/label) to confirm the hardware is set up correctly. The test page now renders at the **same printable width** (Receipt width (dots)) as real sales receipts, so the test is a true preview — no more clipping that disappears once you sell something real.
 - **Preview Receipt / Preview Invoice**
 - **Delete**

In the printer list, a barcode/label printer shows a **label (tag) icon** and a subtitle with its command language and label size — for example ESC/P0S • 40 × 30 mm — so it stands apart from receipt printers at a glance.

Prep printer with no category routed: A prep printer that has zero categories pointing at it shows a red ⚠ icon next to its actions menu — hover for the full explanation. Kitchen tickets only print when a category is routed to a prep printer (see [Section 6 → Routing categories to a prep printer](#)). The warning disappears as soon as one category is assigned.

Using previews:

- **Preview Receipt** shows a sample thermal receipt using the selected paper width.
- **Preview Invoice** shows a sample A4 invoice before exporting or printing.

Important prep-printer note:

- A prep printer cannot be deleted while categories are still assigned to it. Reassign or clear those category links first.

Printing product labels:

A barcode/label printer prints two label layouts:

- **Shelf label** — a wide shelf-edge label: product name and price stacked above a full-width barcode.
- **Price tag** — a compact tag: centred name, large price, and barcode.

Each label carries the product's name, price, unit price, and a scannable barcode (EAN-13, UPC-A, or Code 128, chosen automatically from the barcode value).

To print a label for **one product**:

1. Open **Products** and tap the **label icon** on the product's row.
2. Choose the layout (**Shelf label** or **Price tag**) and the number of **copies**.
3. Tap **Print label**. The dialog also offers **Save label commands**, which exports the raw printer commands (ZPL/TSPL/ESC-POS) to a file for testing or troubleshooting.

To print labels for **many products at once**:

1. In **Products**, open the actions menu and choose **Print labels**.
2. Select the products — use **Select all** or **Clear** to manage the selection.
3. Choose the layout and copies, then print. One label (times the copy count) is queued for each selected product.

If no barcode/label printer has been added yet, the app shows **No barcode printer configured** — add one in **Settings** → **Printers** first.

Language

Tap **Language** and choose **Arabic** or **English**. The app switches language immediately.

Arabic keyboards welcome. Numeric fields — quantities, prices, payment amounts, counts — accept both Western (0–9) and Arabic-Indic (٠–٩) digits, so you can type numbers in either script and they register the same way.

Auto logout when idle

Controls how long the app waits before signing the cashier out due to inactivity. Pick **Off**, **5**, **15**, **30**, or **60 minutes**.

- Default is **15 minutes**.
- Set to **Off** to disable auto-logout entirely — useful for kiosks and single-operator stores where the cashier should stay signed in until they explicitly tap Logout.
- The setting is per-device and survives app restarts.

Weighed barcodes

Lets the till decode the sticker barcodes a weighing scale prints for loose produce, so a scan sets the line weight automatically. **Off by default** — when off, scanning behaves exactly as before, so you only need this if your store has a barcode-printing scale.

Tap **Configure** and set:

Field	What it means
Enable weighed barcodes	Master switch. While off, nothing changes at the till.
Prefix digits	The leading digit(s) that mark a scale's barcode (comma-separated). Almost always 2 — a range reserved worldwide for in-store codes, so it never clashes with a manufacturer barcode.
Item code start / length	Which digits hold the product's item code (matched to a product's PLU code).
Value start / length	Which digits hold the embedded weight (or price).
Embedded value	Whether the barcode carries the Weight (recommended) or the total Price the scale calculated.
Divisor	Scales the value field: 1000 turns grams into kg, 100 turns cents into currency.
Verify EAN-13 check digit	Rejects mis-scanned labels whose check digit doesn't add up. Leave on.

Match it to your scale. There is no single industry standard for *which* digits hold the weight, so the layout above must match how your scale (Bizerba, CAS, Mettler-Toledo, DIGI, ...) is programmed. Use the **Test a barcode** box: paste a real label your scale printed and the screen shows the decoded item code and weight, so you can confirm the settings before saving.

To finish the setup, give each weighed product its **PLU code** in the product editor — see [Selling produce by weight](#).

Promotions

Opens the **Promotions** screen where you create and manage bundle deals. See Section 21 for the full workflow.

Category discounts

Visible only to users with the **Manage Categories** permission.

Opens the **Category discounts** screen where you set up scheduled, percentage-off-by-category deals (for example, a bakery happy hour from 5 to 7 PM). See Section 21 for the full workflow.

Diagnostics

Visible only to users with the **Manage Diagnostics** permission.

Diagnostics is the **opt-in crash and error reporting** panel. Errors that happen on the till (failed prints, sync conflicts, unhandled exceptions) can be forwarded to a Sentry project so the operator/owner can see what went wrong without being on-site.

Field	What it controls
Enable diagnostics	Master switch. Off = nothing is sent over the network.
DSN	The Sentry project URL (<code>https://<key>@<host>/<project></code>).
Environment	Tag attached to every event: <code>production</code> / <code>staging</code> / <code>development</code> . Lets you filter the Sentry dashboard by store or build.
Send test event	Generates an artificial error to verify the DSN works and events land in Sentry.
View recent log	Opens a dialog showing the last 200 lines of the local log (<code><docs>/logs/app.log</code>). The text is selectable and there's a Copy button so you can paste it into a support message. This is purely local — no network call — and useful for confirming what the app <i>would</i> send before you flip the switch.

Privacy: when diagnostics is enabled, sensitive fields are stripped before sending to Sentry — cashier names, customer names, customer VAT numbers, PINs, license codes, and 4–6-digit number strings inside free text are all redacted. The **local** log shown by *View recent log* is **not** scrubbed (it stays on this device), so be aware that copying that text into an external system may include those values.

Restart required: the Sentry initializer reads the saved settings at app startup. After changing the DSN / enable toggle / environment, **close and reopen the app** for the change to take effect. The active/inactive label below the buttons reflects the current runtime state, not the saved state.

Repair customer balances

Each customer's shown balance is a running total kept for speed; the true record is the list of their charges and payments. In rare cases — a crash mid-payment, or the same customer being served on two tills at once — the shown total can drift away from that list.

Repair customer balances (in the Diagnostics screen) fixes this safely:

1. Tap **Check balances**. The app recomputes every customer's balance from their charge/payment history and shows you only the ones that differ, as `name: shown → correct`.
2. If any are listed, review them, then tap **Repair now** to write the corrected figures. If everything already matches, it tells you so and changes nothing.

It only ever recomputes from the existing history — it never deletes a charge, a payment, or a customer — so it is safe to run any time a balance looks wrong. Requires the **Manage Customers** permission.

Updates

Open **Settings** → **Updates** to see the version you are running and check for a newer one. Anyone can open the screen and check; **installing** an update requires the **Manage Backups** permission.

What the screen shows:

- **Current version** of the app.
- A **Check for updates** button. After checking, a status panel shows one of:
 - ✓ *"You're on the latest version."*
 - ⚠ *"Update available: {version}"* with the release notes.
 - ✗ *"Couldn't check for updates."* with *"Check your connection and try again."* and a small technical error line (selectable) to help diagnose network/firewall issues.

Installing an update (Windows only): when an update is available on Windows, a **Download & install** button appears. Other platforms (macOS, Linux) show the status but **not** an install button — update those manually.

Tapping **Download & install**:

1. Asks you to confirm — *"Install version {version}? The app will back up your data, then close and reinstall. Make sure no sale is in progress."* Tap **Back up & install**.
2. The app downloads the installer and **verifies it** before running it.
3. It takes an automatic **backup snapshot** of your database first (a passphrase-free rollback point), then launches the installer and closes.
4. If a sale is in progress, the install is blocked — *"Finish the current sale before installing."* Complete or clear the cart and try again.

Keep the app open while it says “*Downloading update... do not close the app.*” The reinstall continues on its own once the app closes.

17. Backup and Restore

Admin only

Regular backups protect your data. It is recommended to back up at least once a day.

Creating a Local Backup

1. Go to **Settings** → **Backup & Restore**.
2. Tap **Create Backup**.
3. Choose how to protect the backup:
 - **With a passphrase (recommended)**: enter a passphrase of at least 8 characters. The backup is encrypted with AES-256, so the file is useless to anyone who doesn't have the passphrase. **Keep it safe — without it the backup cannot be restored.**
 - **Without a passphrase**: tick **Create an unencrypted backup** to skip encryption. The backup is a plain `.zip` that restores with no passphrase, but anyone who gets the file can read all of your data — only choose this if you will keep the file somewhere safe.
4. A `.zip` file is created containing the full database.
5. Use the **Share** button to save it to Files, AirDrop, or any other location.

Restoring from a Local Backup

1. Tap **Restore from File**.
2. Select the `.zip` backup file.
3. Confirm the warning — **this replaces all current data**.
4. After restore completes, close the app when prompted.
5. Reopen the app before doing any more work.

S3 Cloud Backup

For automatic off-device backups, configure an S3-compatible storage bucket:

Field	Description
Endpoint	The S3 endpoint URL (e.g., <code>https://s3.amazonaws.com</code> or your MinIO server)
Bucket	The bucket name

Field	Description
Access Key	Your S3 access key ID
Secret Key	Your S3 secret access key

Uploading to S3:

Tap **Upload to S3** after configuring the credentials. A timestamped backup file is uploaded.

Browsing S3 Backups:

Tap **Browse S3 Backups** to see existing backups in the bucket. Tap any file to restore from it.

Warning: Restoring from any source (local or S3) permanently replaces all current data. Make sure you are restoring the correct file. After restore, close the app and reopen it before using the system again.

18. Activation

Admin only

Where to Find It

Tap **Activation** in the navigation rail (visible to admins).

Checking License Status

The screen shows:

- The activation code currently in use.
- The **expiry date**.
- Days remaining.

Renewing the License

1. Paste the new activation code into the code field.
2. Tap **Activate**.
3. On success the new expiry date is displayed.

Expiry Warning

A warning banner appears on the home screen when fewer than 14 days remain on the license. Renew as soon as possible to avoid service interruption.

19. Customers

Access

- **Admin** — always.
- **Cashier** — only if the admin has enabled the **Manage Customers** permission.

What the Customers Screen Is For

Use **Customers** to maintain customer master records for finance and receivables workflows.

The screen shows:

- customer name
- phone or email
- current balance

Adding or Editing a Customer

1. Go to **Customers**.
2. Tap **+** to add a customer, or tap an existing customer to edit.
3. Fill in:
 - **Name**
 - **Phone** (optional)
 - **Email** (optional)
 - **Credit Limit**
 - **Notes** (optional)
4. Tap **Save**.

Recording a Payment

When a customer pays down what they owe, record it so their balance drops:

1. Open the customer and tap **Record payment**.
2. Enter the **amount**.
3. Choose how they paid:
 - **Cash** — the money goes into the till. If you have a shift open, this payment is added to that shift's drawer and shows as **On-account cash** on the close/Z report, so the till still balances (see §11).
 - **Card** — settled by card/bank transfer. It does **not** touch the cash drawer.
4. Add an optional **note** and tap **Save**.

The payment reduces the customer's balance immediately and is listed in their transaction history.

Permission required. Recording a payment needs the **Settle Customer Payment** permission. Choosing Cash vs Card matters for the drawer: only **Cash** affects the shift's expected cash.

Understanding the Balance

- A **positive balance** means the customer owes the store.
- A **zero balance** means nothing is currently owed.

If your store is not yet using customer credit workflows, many customers may remain at zero balance.

If a balance ever looks wrong, an admin can recompute it from the customer's charge/payment history with **Settings** → **Diagnostics** → **Repair customer balances** (Section 16). It never deletes anything.

20. Accounting

Access

- **Admin** — always.
- **Cashier** — only if the admin has enabled the **Export Accounting** permission.

What the Accounting Screen Does

The **Accounting** screen exports finance data in CSV format for import into external accounting software.

Exporting Journal Entries

1. Go to **Accounting**.
2. Choose the **From** and **To** dates.
3. Tap **Export Journal Entries (CSV)**.
4. The app generates a CSV file and opens the device share sheet so you can save or send it.

Exporting a Trial Balance

1. Go to **Accounting**.
2. Choose the **From** and **To** dates.
3. Tap **Export Trial Balance (CSV)**.
4. The app generates a CSV file and opens the device share sheet.

What Is Included

The accounting export is built from the transactions already recorded in the app, including:

- sales

- refunds
- exchange orders
- tax
- cost of goods sold
- purchase and restock inventory documents
- general expenses
- supplier payments

Important accounting notes:

- Refund exports reverse the related revenue, tax, and cost-of-goods impact.
- Exchange exports include both the returned items and the newly sold items.
- Posted supplier purchase/restock documents can appear as inventory/account payable style entries in the journal export.
- **Payment type drives the cash vs. bank side of each entry.** Anything paid by **card** — card sales, card-paid supplier payments, card-paid expenses, and card-paid no-supplier purchases — posts against the **Card Receivable** (bank) account. Anything paid by **cash** posts against the **Cash** (till) account. Supplier purchases left **on account** post to **Accounts Payable** and only move to Cash/Card Receivable when you record the payment. This is what keeps the *Bank reconciliation* report and your real bank balance in agreement.

Quick Reference — Common Tasks

Task	Where to go
Process a sale	POS
Add a product	Products → +
Receive new stock	Inventory → Documents → + (Purchase)
Record rent or utility bills	General Expenses → New Expense
Write off damaged items	Inventory → Documents → + (Waste)
Open / close a shift	Shifts
Finalize end of day	Shifts → Close Shift, then Z Close
Issue a refund	Orders → select order → Refund
View today's sales	Reports
Add or update a customer	Customers
Record a supplier payment	Suppliers

Task	Where to go
Export accounting CSV files	Accounting
Add a new staff member	Users → +
Change store name or contact info	Settings → Business Info
Change currency or tax rates	Settings → Tax Settings
Back up data	Settings → Backup & Restore
Activate / renew license	Activation

Frequently Asked Questions

Q: I entered my PIN wrong too many times and I'm locked out. What do I do?

A: Wait 5 minutes for the lockout to expire. An admin can also reset your PIN if needed.

Q: Can I edit stock quantities directly?

A: No. Stock is updated only through Inventory Documents. This protects the audit trail. Create an Adjustment document to correct stock levels.

Q: I posted a document with the wrong cost. How do I fix it?

A: Open the posted document and tap **Create Reversal** to undo it completely, then create a new document with the correct values. Or tap **Create Correction** to make an adjusted version.

Q: When should I use Purchase vs Restock?

A: Use **Purchase** when stock came from a supplier invoice or delivery. Use **Restock** when you are adding stock without a normal supplier purchase document, such as replacement stock, bonus stock, or stock found in storage.

Q: When should I use General Expenses instead of a Purchase document?

A: Use **General Expenses** for operating costs that do not change stock, such as rent, utilities, salaries, transport, maintenance, or petty cash. Use **Purchase** only when the spending brings products into inventory.

Q: Can staff type any expense category they want?

A: No. Expense categories are created by an admin. Staff choose from the available category list when creating or editing an expense.

Q: Can I enter expiry dates while receiving stock?

A: Yes. In **Purchase** and **Restock** documents, enter the expiry date on each incoming line before posting. The system tracks expiry by batch and keeps the report updated as stock moves.

Q: Why is the product cost different from what I entered?

A: The system blends the new cost with your existing stock using a **quantity-weighted average**: new cost

$$= (\text{old cost} \times \text{stock on hand} + \text{entered cost} \times \text{quantity received}) \div (\text{total quantity})$$
. A large existing stock keeps the cost stable when a small new batch arrives at a different price. If the product had no previous stock or cost, the entered cost is used as-is. Cost is always stored per single (base) unit, even when you buy by the case.

Q: I paid a supplier in cash from the till and now my drawer looks short at close. Why?

A: That's expected — and handled. Record the payment with the **Cash** method while your shift is open (from the Suppliers screen or the supplier-invoice payment button). The app counts it as money taken from the drawer and subtracts it from the expected cash at close, so your count matches. If you paid by bank/card, choose **Card** instead and it won't affect the drawer. The same applies to cash General Expenses.

Q: The Profit report says some products were sold with no recorded cost. What does that mean?

A: Those products still have a cost of 0 (usually new products that were sold before their first purchase was posted), so every sale shows as 100% profit and overstates your real profit. Post a Purchase or Restock document with the correct unit cost for each one to fix it.

Q: What is the difference between Close Shift and Z Close?

A: **Close Shift** records the counted cash, expected cash, and variance. **Z Close** is the final end-of-day step that locks in the closed shift and produces the final Z summary.

Q: Where do I change the store currency or tax rates?

A: Go to **Settings** → **Tax Settings**. You can update the currency code, currency symbol, VAT number, and the list of available tax rates there.

Q: A barcode scan found multiple products. Which one do I pick?

A: A list of matching products appears. Tap the correct one to select it.

Q: How do I add a product that comes in a box of 12?

A: In the product editor, add the single item barcode as the main barcode (multiplier = 1). Then under **Packages**, add the box barcode with multiplier = 12 and the box price. When you scan the box barcode at the POS, the customer is charged the box price. When you scan the box barcode in an Inventory Document with quantity = 1, the system adds 12 units to stock.

Q: The app logged me out on its own. Is that normal?

A: Yes — by default after 15 minutes idle. The owner can change the timeout (or disable it) under **Settings** → **Auto logout when idle**. Your open cart is preserved across logouts.

21. Promotions & Discounts

The POS supports two kinds of automatic deal, both configured here and both applied at the till with no cashier action:

- **Bundle promotions** — re-price a set of products to a flat bundle price (e.g. *"2 milk + 1 bread for 10 SAR"*).

- **Category & time discounts** — take a percentage off everything in a category, on a daily/weekly schedule (e.g. “20 % off Bakery, 5–7 PM”).

Bundle promotions

Requires the **Manage Settings** permission to access. The Settings → Promotions entry is hidden from users without it.

A **bundle promotion** is a deal like “Buy 2 milk + 1 bread for 10 SAR”. When the cart contains everything the bundle requires, the savings auto-apply — the cashier doesn’t have to do anything, and there’s no way for them to forget.

What a bundle is

A bundle is defined by:

- A **name** (Arabic + English). Shown in the cart’s “Applied bundles” list.
- A **bundle price** — the flat amount the matched items should sum to, **before tax**. Whatever the difference is between the items’ regular prices and this number becomes the customer’s saving.
- A list of **required items**, each with a product and a required quantity. All of them must be present in the cart (at or above the required qty) for the bundle to fire.
- An **Active** toggle. Inactive bundles are kept for future re-use but ignored at the POS.

Creating a bundle

1. Go to **Settings** → **Promotions**.
2. Tap **New promotion** (the green floating button).
3. Fill in:
 - **Name** (and **Name (Arabic)** if you want a localised display).
 - **Bundle price** (positive number, before-tax).
 - Toggle **Active** on/off as needed.
4. Under **Required items**, tap **Add item** and pick a product + the qty required.
5. Add more items if the bundle requires multiple products.
6. Tap **Save**.

Editing or deleting a bundle

- Tap a bundle row to edit. The same editor opens.
- The switch on each row toggles **Active** without opening the editor — useful for turning a holiday promo on and off without re-typing it.
- The trash icon deletes the bundle. Past sales that used it are **not** affected.

How bundles apply at the POS

- The POS evaluates every active bundle against the cart on every change. If the cart satisfies a bundle, the savings show under the totals as an **Applied bundles** list with **× N** indicating how many times the bundle was satisfied.
- A bundle of “**2 × Milk + 1 × Bread**” triggers **twice** if the cart has 4 Milk + 2 Bread, **once** if 2 Milk + 1 Bread, and **zero times** if 2 Milk + 0 Bread.
- Each cart item can only count toward **one** bundle. Bundles are tried in the order they were created (lowest ID first).
- Bundle savings are correctly factored into VAT — the customer is **not** taxed on amounts they didn’t pay for.
- A bundle whose total list price is **lower** than the bundle price (e.g. items already at sale prices) is skipped — bundles never *increase* the price.

What bundles cannot do (current limits)

- **No time windows.** Bundles are either Active or not. To run a *percentage* deal on a daily or weekly schedule, use **Category & time discounts** (below) instead; for a bundle, toggle it on Friday morning and off Sunday night.
- **No customer-group targeting.** All cart contents are eligible; you can’t restrict a bundle to a specific customer.
- **No automatic stacking rules.** If you create two bundles that share a product, the FIRST bundle (lowest ID) claims those units; the SECOND skips them.
- **No “buy N get one free” syntax.** Express the same intent as a bundle of N+1 items priced at the cost of N items.
- **Promotions don’t sync between terminals.** If you run multiple LAN-paired tills, configure the same bundles on each.

Worked example

You want to run “**Lunch combo: 1 burger + 1 chips + 1 cola for 25 SAR**” when the individual prices are $18 + 6 + 5 = 29$ SAR.

1. Name: `Lunch combo` . Bundle price: `25` . Active: on.
2. Required items: Burger × 1, Chips × 1, Cola × 1.
3. Save.

At the POS, when the cashier scans all three items, the cart automatically shows:

```
Applied bundles
• Lunch combo × 1
Bundle savings: -4.00
```

The customer pays the 25 SAR bundle price (before tax), VAT is computed on the discounted amount, and the receipt shows the saving on the totals row.

Category & time discounts

A **category discount** takes a percentage off every item in a chosen category, on a schedule you set — specific days of the week and/or a daily time window (a “happy hour”). Like bundles, it applies automatically at the POS; the cashier does nothing.

Requires the *Manage Categories* permission to access. The Settings → Category discounts entry is hidden from users without it.

What a category discount is

- A **name** (Arabic + English). Shown on the cart’s applied-discounts line.
- A **category** — the discount applies to every product in this category.
- A **discount %** (1–100) off each matching line, before tax.
- **Active days** — pick any combination of weekdays; leave them all selected for every day.
- **Active hours** — either **All day**, or a **From / To** time window. A window may run *past midnight* (e.g. 22:00 → 02:00) for late-night deals.
- An **Active** toggle. Inactive discounts are kept but ignored at the POS.

Creating a category discount

1. Go to **Settings** → **Category discounts**.
2. Tap **New category discount** (the green floating button).
3. Enter the **name**, pick the **Category**, and enter the **Discount %**.
4. Choose the **Active days** chips, then either keep **All day** on or turn it off and pick a **From** and **To** time.
5. Toggle **Active** as needed and tap **Save**.

The switch on each row in the list toggles **Active** without opening the editor; the trash icon deletes the discount (past sales are **not** affected).

How they apply at the POS

- The POS checks the clock and calendar against every active discount whenever the cart changes. If a discount is inside its window, each cart line in its category is reduced by the percentage and the saving shows on the totals.
- **Best discount wins per line.** If two category discounts could apply to the same category, only the larger percentage is used — they do **not** stack.
- Savings are factored into VAT correctly — the customer is not taxed on amounts they didn’t pay.
- The schedule is evaluated using the **till’s local clock at each cart change**. A window that opens at 17:00 therefore takes effect the next time the cashier adds or changes an item, not retroactively on a cart that was left untouched.

Worked example — bakery happy hour

Run **20 % off the Bakery category, every day from 17:00 to 19:00**:

1. Name: Bakery happy hour . Category: Bakery . Discount %: 20 .
2. Active days: all selected. **All day**: off → From 17:00 , To 19:00 . Active: on.
3. Save.

Between 5 and 7 PM, any bakery item in the cart is automatically discounted 20 %, the saving appears on the totals, and VAT is charged on the reduced amount. Outside those hours the price is normal.

Current limits

- **Percentage only** — no fixed-amount-off in this version.
- **One category discount per line** — the best-matching one; no stacking with another category discount. (A line can still also be part of a bundle.)
- **Category discounts don't sync between terminals** — configure the same ones on each LAN-paired till, like bundles.

22. ZATCA E-Invoicing (Phase 2)

ZATCA Phase 2 (the “Integration Phase”) is the Saudi e-invoicing standard where every sale becomes a **signed electronic invoice** that is reported to ZATCA. The POS does all of this automatically once it is set up — there is nothing extra to do on each sale.

Turning it on

Phase 2 is **off by default**. To enable it:

1. Open **Settings** → **Tax Settings**.
2. Turn on **E-invoicing (ZATCA)** and tap **Save**.
3. Two new entries appear under **Settings**: **ZATCA onboarding** and **E-invoice status**.

The till must then be **onboarded** with ZATCA — a one-time registration per till. That procedure (portal OTP, EGS registration, activation) is in the separate **ZATCA_ONBOARDING.md** guide. Until a till is onboarded, the POS keeps using the Phase 1 QR code and issues no electronic invoices.

What happens after a till is onboarded

Event	What the POS does automatically
You complete a sale	Issues a signed ZATCA invoice and queues it for reporting
You refund a sale	Issues a signed credit note linked to the original invoice

Event	What the POS does automatically
A receipt prints	Carries the Phase 2 QR code (nine data tags)
The till is offline	Invoices queue and report automatically once the connection returns

You never sign, report, or send anything by hand — issuing and reporting run in the background.

Checking invoice status

Open **Settings** → **E-invoice status** to see every ZATCA invoice and its state:

Status	Meaning
Pending	Issued, waiting to be sent to ZATCA
Queued	Sending was deferred (offline or a temporary error) — will retry
Reported	Accepted by ZATCA (simplified / B2C invoice)
Cleared	Accepted and stamped by ZATCA (standard / B2B invoice)
Failed	ZATCA rejected it — open the row to see why

Send pending reports and **Send pending clearances** force an immediate send; normally the background task handles this on its own.

23. Multi-Terminal Sync (LAN)

Requires the Manage Sync permission. Optional — most stores run a single till and never need this.

If you run **more than one till in the same shop**, the app can replicate sales and catalog data between them over the **local network (LAN)** — no internet and no cloud server involved. Each till keeps its own database; sync copies new activity from one to another.

This is a same-shop, same-network feature. It is **not** a central cloud system for linking branches over the internet. Every till still works fully on its own if the network drops.

What replicates between tills

- **Orders** — sales, refunds, and exchanges, with line items and the matching **stock change** applied on every till, plus an audit movement row.

- **Products** and **Categories** — admin edits propagate; the most recent edit wins if two tills change the same item.
- **Inventory documents** — purchases, restocks, waste, counts, and reversals apply their stock change on every till, **including the supplier and payment status** (who you owe, how much is paid, when it's due).
- **Supplier payments** — a settlement recorded on one till updates the purchase's paid amount and status on all of them.
- **Users** and **Customers** — a cashier or customer added on one till can be used on the others; customer balances are recomputed locally so concurrent payments on different tills add up correctly.
- **Promotions (bundles)** and **general expenses** — a bundle or an expense entered on one till shows up on all of them (an expense counts in every till's store-wide reports). **Deleting** one on any till also removes it everywhere, so you never have to clean up the same stale bundle or mistaken expense twice.

What stays on each till by design: a till's **shift / cash-drawer session** and its **cash in/out movements** are local to that drawer — they are how *that* till reconciles its cash, not store-wide data, so they are never sent to other tills. (A sale, supplier payment, or expense made at a drawer still replicates as a store-wide fact; only the *drawer attribution* stays local.)

Turning it on (do this on every till)

1. Go to **Settings** → **Multi-terminal sync**.
2. Turn on **Enable LAN sync**.
3. Give the till a **Terminal name** (e.g. *Front counter*, *Drive-thru*). Its read-only **Terminal ID** is shown beneath the field.
4. Set the **Shared LAN secret** — the **same secret on every till**. This is what authorises tills to talk to each other, so treat it like a password. On a fresh till the field is **pre-filled with a strong random secret** — tap the **copy** icon and paste it onto your other tills. Use the **eye** icon to reveal it, or **Generate new secret** to make a fresh one. (You can still type your own, but it must be at least 16 characters.)
5. Tap **Save**, then **close and reopen the app** so sync starts.

One secret for the whole shop. Generate it once on the first till, copy it, and Save the *same* value on every other till. A till with a different secret simply won't sync.

The status line shows “**Sync is active. Discovered peers: {n}**” when running, or “**Sync is inactive.**” when off.

Approving tills (Peer pairing)

For security, a till only syncs with tills you have **explicitly approved**. The **Peer pairing** panel handles this:

1. On **both** tills, open **Settings** → **Multi-terminal sync** and tap **Open 5 min** in the Peer pairing panel. The panel shows “**Pairing open — closes in m:ss**” with a countdown.
2. While both windows are open, the tills discover each other on the LAN and **auto-approve (pin)** each other on first contact.

- Each till then lists the other under **Approved peers** with a ✓ and its Terminal ID. *“This terminal: ...”* shows your own ID for reference.

To stop syncing with a till later, tap the **trash icon (Revoke)** next to its ID. To close the pairing window early, tap **Close**.

Only open a pairing window when you are actually adding a till — it’s a brief, deliberate step, not something to leave open.

Direct connection (when tills can’t find each other)

Tills normally **find each other automatically** on the LAN. On some networks that automatic discovery is blocked — **guest/public Wi-Fi**, networks with “**client isolation**” or “**AP isolation**” turned on, or tills on **separate Wi-Fi access points**. If the other till never appears even with pairing open, use the **Direct connection** panel to point one till straight at the other by its address:

- On the till you want others to reach, look under **Direct connection** → **This device**. It lists this till’s address as **IP:port** (e.g. `192.168.1.42:46101`). Tap the **copy** icon next to it.
- On the other till, in **Direct connection** → **Manual peers**, type or paste that address into **IP address or host** and tap **Add**. (The port is optional — leave it off and it defaults to **46101**.)
- The till you added now appears in the **Sync status** panel and starts syncing within a few seconds.

Security is unchanged. A manual address only tells one till *where* to look — the **shared secret** and **peer approval** still apply on both sides, exactly as with automatic discovery. Add the address on whichever till(s) you want to do the reaching; one direction is enough since sync is two-way.

Still need the same network. Direct connection works around blocked *discovery*, not a missing network — both tills must still be able to reach each other (same router/LAN, no firewall blocking the port between them).

Sync status panel

While sync is running, a **Sync status** card shows:

- Last pull** time and a **Pull now** button to fetch immediately instead of waiting for the automatic cycle.
- Each **peer** — discovered automatically *or* added under Direct connection — with a green ✓ (healthy) or red △ (error), its address, when it was last **seen** and last **pulled**, and its app version.
- A **version-mismatch warning** if a peer runs a different app version — “*update all terminals to the same version.*” Keep every till on the **same version** to avoid sync problems (see §16 → Updates).
- A **Recent conflicts** box if a simultaneous edit on two tills caused one side’s change to be dropped (most-recent-edit-wins), so an admin can redo it.

Good to know / limits

- **No internet exposure.** Sync runs only on your LAN. Do not forward its port to the public internet.
- **Brief overcommit is possible during a network outage.** If two tills are disconnected, both might sell the last unit of a low-stock item; the numbers reconcile once they reconnect, and the inventory movement log shows it.
- **Same shared secret + approval on every till** — a till with the wrong secret, or one you haven't approved, simply won't sync.

Features evolve between releases — check the “Recent additions” note at the top of this file, and confirm your installed version in Settings → Updates. For technical support, contact your system administrator.