

NexGate Shipping Service — Build Plan Development

How we build `docs/shipping/SHIPPING_SERVICE_DESIGN.md` (the "design", § numbers below point to it). Status: **plan agreed to start, nothing built yet**. Built on the Location Service (done: wards, shop locations, user addresses, frozen places).

Ground rules for every step

- **Talk first, then build** — each step starts with a short talk on its details, then code.
- **Code standard**: new top-level package `shipping_service/` (same level as `location_service/`), the project's controller / service / service/impl / payload / entity / repo shape, no boilerplate, comments only where needed.
- **Existing responses**: add fields; reshape only when agreed (dev phase) and always recorded.
- **Every change recorded** in `commerce_logistics_changes.md` — before / after, examples, app to-do.
- **Design doc kept true** — any decision that changes during a step is updated in the design.
- **Settings**: fixed rules in properties (`shipping.*`), per-shipper agreement settings in the database (§6.7.1).
- **Money**: every money move goes through the ledger; every step that moves money ends with the ledger balance check.
- **Tested live** locally after each step (like the location steps), using the worked scenarios (§17) as test cases.
- **Phase-0 hooks** (§15b) are built in from the start: attempt records, failure reason codes, the extra statuses, timers as data, reverse legs allowed — even though their rules come later.

The order at a glance

Step	Delivers	Real-world result after the step
0	Foundations	nothing visible; ground ready
1	Pickup, end to end (with codes and "✓ It's what I ordered")	buyers can buy and collect at the shop

Step	Delivers	Real-world result after the step
2	Native shipper registration (KYC + Service Setup basics)	delivery businesses can sign up
3	Door coverage, pricing, ETA	businesses describe where / how much
4	Engine: estimation + quotation (single door leg)	buyers see real delivery costs
5	Cart / checkout / agreement, escrow per part	buyers pay real delivery
6	Job flow: offers, codes, statuses, release	first real door-to-door deliveries
7	Reassignment, switching, cancelling, bulky mismatch	unhappy-but-common cases handled
8	Reliability, reviews, auto-pause, upcoming orders	trust and planning
9	Stations (single leg)	intercity: collect at a station
10	Multi-shipper chains	shop → bus → door
11	Group purchase and installments	shipping inside group / installment buys
12	Free shipping (seller rules + campaigns)	"Free delivery" badges and filter
13	Bolt integration	boda on demand in town
14	Admin dashboard (screens)	ops tooling — last
later	Notifications set · unhappy paths (§15a/b) · staging deploy	after the build

Step 0 — Foundations

Goal: everything later steps stand on, with no behaviour change.

- Package `shipping_service/` with sub-packages from §2 (empty shells added as each step needs them).
- `shipping.*` properties with every value from §6.7.1 (read through one settings class).
- **Shared opening hours** (§6.3): weekly ranges, special days, temporarily closed, cut-off, EAT, overnight; "is open at", "next opening", "add opening hours" helpers. Attached to **shops** now (§6.4): shop hours endpoints, "Open now / Opens Sat 09:00" on shop responses; shop **pause modes** (keep selling / stop orders, `SHOP_CLOSED`).
- **Escrow table:** `part` (ITEMS / LEG), `legId`, `orderId` filled, **provider payee** (§12.1) — new columns only, current payments keep working (they are "ITEMS" parts).

- **Product** `bulky` field (§8.1) — default false; category pre-tick later.
- **File Thunder**: new upload types *business document* (private) and *business logo* (separate repo, small).
- **Codes service**: 4 random digits, hashed, never logged, 5 tries → locked, regenerate (§11). Replaces the old `DeliveryConfirmation` code generator (and its plain-code log line).

Check: existing checkout / orders / cart-escrow tests still pass; shop hours + pause visible; a code can be generated, verified, locked after 5 wrong tries, and never appears in logs.

Step 1 — Pickup, end to end

Goal: a buyer can buy, the seller prepares, the buyer collects with a code, money is released on confirmation.

- Move Step-4 `e_commerce/fulfillment` (JSON) into `shipping_service/fulfillment` as a **table**; orders link to it (§2, §3.5). Frozen places stay inside the record.
- **Shipping agreement entity** (§4.8): `shipping_agreement` + parties + changes; states DRAFT → PENDING → AGREED → DONE / CANCELLED. For pickup the only parties are the buyer and the seller. **Escrow parts move only on agreement events** from here on.
- Order flow: seller **accepts** (24 opening hours, else auto-cancel + refund) → **Ready** → buyer told → **pickup at the counter**: buyer's code, seller enters it → *collected* (§3.1, §10).
- "✓ **It's what I ordered**" releases **all** escrow parts in one transaction; "⚠ **Report a problem**" freezes them (§12.2).
- Account setting: **code or "received" button** (§11).
- Pickup statuses `placed → accepted → ready → collected → completed` (+ cancelled / expired); optional **pickup time** (shop opening hours, inside the window; reminder that morning).
- **14-day collection** expiry with reminders (§3.1). **Cancel before the seller accepts** (§3.4a). **Seller cancel** at once (pop-up "I have already informed the buyer"; pickup before collected, delivery before Ready).
- Recent pins recorded **after payment**, not at checkout (§4.4).
- Admin endpoints needed from now: manual **release / refund** of an escrow part (§6.7.2) — `SUPER_ADMIN`, reason, audit log.

Check: scenario **A** (§17) live: pay → accept → ready → code → collected → confirm → seller paid; 14-day expiry; seller cancel refunds; cancel before accept refunds; problem report freezes; admin release works and is logged.

Step 2 — Native shipper registration

Goal: a delivery business can sign up and get approved automatically.

- **KYC** (§6.2 part 1): name, contact phone, **office location** (pin), business licence (file + number + expiry, unique number), director's NIDA (file + number), LATRA (file + number

+ expiry) → auto-approved `KYC_APPROVED`.

- **Service Setup basics** (§6.2 part 2): profile (logo, description, WhatsApp, email, badge), service type (door / station), **hours** (shared piece, default Mon-Sat 08:00–18:00), limits (max value, bulky + extra, response window, heads-up). Setup checklist endpoint.
- Statuses `KYC_APPROVED` → `LIVE` ⇌ `PAUSED`, `SUSPENDED`; documents private (owner + admin).
- "Delivery business" mode in the app (owner only, §6.6) — endpoints for the owner's own business.
- Admin endpoints: **suspend / lift** (§6.7.2).

Check: register with all documents → auto-approved; duplicate licence number refused; documents not visible to others; checklist shows what's missing; suspend / lift logged.

Step 3 — Door coverage, pricing, ETA

Goal: a business says where it works, how much, how long — and goes LIVE.

- **Location helpers** (in `location_service`): regions → districts → wards lists + search; coverage outline for the map.
- **Coverage** (§7.1): ticked areas (region / district / ward), trips `WITHIN_EACH` / `WITHIN_AND_ACROSS`, "both ends inside", fast **area-code matching** (the spec's second core function, `areaMatch`).
- **Pricing** (§8.1): distance bands (price + time each, ×1.3, last band = max distance), flat, value add-on ladder
 - max value, bulky extra, rounding up to 100.
- Goes **LIVE automatically** when service type + coverage + pricing exist.
- Delivery commission fee type (0%) in the fee table + per-shipper override with start date (§6.7.1).

Check: the §7.1 crossing table (Sinza → Mikocheni with `WITHIN_EACH` vs `ACROSS...`) as tests; band and add-on maths incl. rounding; business flips to LIVE when complete.

Step 4 — Engine: estimation + quotation (single door leg)

Goal: buyers see real delivery options and costs.

- Engine (§9): where → who (coverage) → filter (distance, value, bulky, hours / cut-off, paused) → cost + ETA → clean-up (dominated, outliers 3×) → labels $\square \leq \square$ ($\square$ = price + time at launch) → order, top 5, fairness.
- **ETA** (§8.2): from "shop ready", ranges, closed hours.
- **Estimation** `POST /api/v1/shipping/estimations` (§4.5): `cost {from,to}`, `eta {from,to}`, pickup area; Redis cache shop-area → buyer-area (1 h).

- **Quotation** (§4.1): live options for a shop's parcel and a destination (§4.2: saved / recent / pin; station recognition comes with step 9).
- Product detail: live quotation for the viewer's address.

Check: scenario **L** (cards) and the engine walk-throughs from the master guide (A, B, E) reproduce; cache hit on the second call; closed shipper shifts ETA instead of disappearing.

Step 5 — Cart, checkout, agreement, escrow per part

Goal: buyers pay the real delivery cost; money split into items + leg.

- Cart (§4.3, §4.4): one destination per cart, per-shop quotation **held 5 days**, add-to-cart `fulfillment` + `dropoff`, defaults for a new shop, unsaved pin on the cart, re-quote on changes.
- Checkout → **agreement** (§4.1, §4.8) with its **legs** (each leg escrow part points to one): editable on the checkout screen until paid (re-quote, wallet re-check), **locked once paid**. Buy-now uses the sheet's quotation.
- **Escrow per part** (§12.1): items part → seller, leg part → shipper owner; release all on confirm.
- **Seller fee on items + tax only**; delivery commission from the leg (§8.3).
- `shippingByShop` now real costs (replaces the 0 placeholder).
- What the app keeps vs the server (§4.6) honoured.

Check: scenario **B** up to payment: escrows created per part with the right payees and fees; quotation older than 5 days refreshed; edit at checkout re-quotes; ledger balance check passes.

Step 6 — Job flow, codes, delivery

Goal: the first real door-to-door delivery from offer to release.

- Seller accepts → **heads-up**; **Ready** → **offer** with response window (counted while open), **closed-shipper 4 h rule** (§10).
- **Ready pop-up**: "I have packed it well" + "label / number is on the package" (recorded).
- **Parcel number** created when the seller accepts (QR label generated by the app, or written on the box); scanned / typed at pickup → "✓ yours / ✗ wrong parcel"; shippers see the parcel, never the order (§11, §6.6).
- Shipper: accept / reject (reason), timeout; active job steps; **pickup code** (seller verifies) and **delivery code** (shipper submits) or the buyer's button (§11); optional handover photos.
- Shipper accept = **signs the agreement**; all signed → AGREED, parcel moves (§4.8).
- Statuses door ladder (§3.2) incl. **heading to the shop** + phase-0 statuses.

- Release on "✓ It's what I ordered": items → seller, leg → shipper (one transaction).
- Shipper app endpoints for offers / jobs / earnings (§6.6), live push.

Check: scenario **B** end to end; scenario **M** (a shipper's day) step by step; timeout counted; codes locked after 5 tries.

Step 7 — Reassignment, switching, cancelling, bulky mismatch

Goal: the common unhappy-but-normal cases.

- **Reassignment** (§10a): rejection / timeout → that leg only; tiers **1 km → same district → buyer**; same-or-cheaper and not slower = automatic (escrow re-pointed, difference refunded); dearer / none → buyer decides; **loop guard 3 per order; buyer silent 24 h → pickup; chain safety net 24 h**; free-shipping caps recalculated; history kept.
- **Switching** (§3.4): pickup → delivery (fresh quotation, pay, offer at once if Ready); delivery → pickup (before any shipper accepted, refund).
- **Cancelling** (§3.4a): shipping-only until pickup — free before **heading to the shop**, wasted-trip fee (50% / Bolt's charge) after; whole order when the chain failed.
- **Agreement changes** (§4.8): propose → affected parties approve → default on silence; **destination change** (accept new cost / keep old price / decline; no withdraw after pickup).
- **Bulky mismatch** (§8.4): report before pickup, bulky-only reassignment, seller pays difference + 50% wasted trip, auto-flag after 2.
- **Badly packed** refusal (§8.5): before pickup, photo, seller repacks + Ready again, seller pays 50% wasted trip.
- Admin endpoints: force **reassign** / **cancel** (§6.7.2).

Check: scenarios **D, F, G, H, N, O, P**; loop guard; silent buyer → pickup after 24 h; ledger balanced after every adjustment.

Step 8 — Reliability, reviews, auto-pause, upcoming orders

- **Reliability counters** (acceptance, on time, failed, broke binding) and **auto-pause** levels with the §14 table, grace period 10 offers, reasons shown.
- **Reviews** like shops, verified only, owner replies, moderation (§14); shown in quotations.
- **Two-way private ratings** (shipper rates buyer / seller).
- **Upcoming orders** on the shipper dashboard, area-only until accepted (§6.6).
- Real delivery times recorded (for after-launch ETA).

Check: the §14 trigger table as tests (2 / 3 missed, 3rd pause in 7 days → suspended...); new shipper not paused inside the grace period; upcoming card hides the exact pin until accepted.

Step 9 — Stations (reached by a door leg)

- Stations (§7.2): pins with counter name, phone, hours; ≥ 1 per ticked area; **lines; reach 30 km** (per business override).
- **Station-pair table** (price + time, both ways, pre-filled pairs) (§8.1).
- Handover points (§7.2a): leg 1 always starts with a collection at the shop (option A) — either the station business **collects from shops** (areas + flat fee, 0 = free; one leg) or a door shipper brings it to the station (the **two-leg chain** door → station, built here); buyer collects at the destination station (station staff enter the buyer's code). Step 10 extends it to 3 legs, editing and re-confirms.
- Station ladder (§3.3); quotation **meeting-point recognition** ("collect at Mbeya Bus Stand", §4.2).

Check: a Mbeya → Morogoro trip as boda → bus (two legs) and as a bus company collecting free (one leg, ranked cheapest); no door shipper at the shop → no station route; pin within 1 km of a station offers "collect there"; empty pair = not offered.

Step 10 — Multi-shipper chains

- Chain building (§4.7): up to 3 legs, meeting points within 1 km, always next to single-leg options.
- Quotation editing per leg: stop early, swap a leg's shipper, change a meeting point.
- **All legs offered at Ready, nothing moves until all accept; binding after accept; re-confirm** when an accepted leg's time moves > 4 h (§10a 7a).
- Shipper → shipper handover: "handed over" / "received" + that handover's code, optional photo (§11).
- One escrow part **per leg**; release all together (§12).

Check: scenarios **C, D, E** with three legs; one leg rejecting while others accepted; re-confirm triggered by a

“ 4 h move.

Step 11 — Group purchase and installments

- Group (\$5): each member's own fulfillment and fee with their payment; refunds on fail / withdraw; moves with a transferred payment; price frozen at payment.
- Installments (\$5): immediate = fee with the down payment; layaway = choose and pay after the last instalment (reuses switch-to-delivery).

Check: scenarios **J** and **K**.

Step 12 — Free shipping

- Seller rules (areas, minimum, cap, product override) and NexGate campaigns (areas, dates, cap, audience, shops, budget reserved at payment, auto-stop) (\$13).
- Stacking, the cheapest qualifying option free, faster pays the difference; campaign ledger account; seller share from the items part at release.
- Estimation `freeDelivery` + card badge + "Free delivery to my area" search filter.

Check: scenario **I**; campaign budget never overspent under parallel checkouts; refund returns budget.

Step 13 — Bolt integration

- Adapter (\$6.5): OAuth2 token cache, estimations (boda, non-bulky, in-city), book at Ready (re-estimate, held price = max + 2%), cancel, webhooks (3 IPs + token) and polling fallback, state mapping, tracking link + driver details.
- **Bolt ledger account**, monthly settlement record, receipt differences; **kill-switch** and **daily spend limit** (\$6.7); seller pays a shop no-show.
- Built and tested in Bolt's **sandbox** first.

Check: sandbox flow (estimate → create → states → completed → receipt); NO_DRIVER_FOUND → reassignment; spend limit blocks new Bolt quotations; kill-switch hides Bolt immediately.

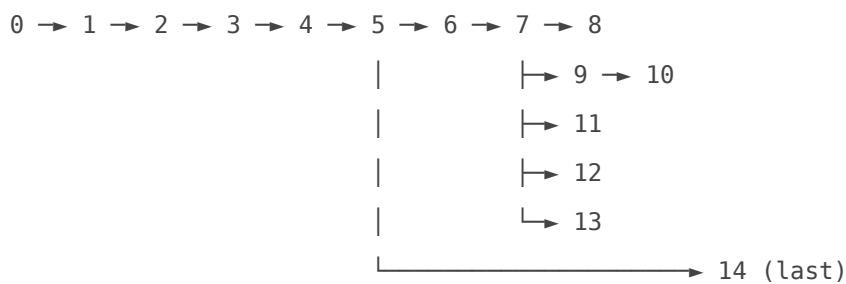
Step 14 — Admin dashboard (last)

Screens on top of the existing admin endpoints (\$6.7.3): shippers + KYC documents, Bolt status and settlement, operations board, campaigns, reports (by area), audit log, `STAFF_ADMIN` / `SUPER_ADMIN` split.

After the build

- **New dedicated shipping notifications** (one set, after the shipping service).
- **Unhappy paths** (§15a / §15b): auto-confirm window, delivery attempts + office hold, returns, disputes.
- **Staging / production:** PostGIS image + `CREATE EXTENSION`, ward file, settings — at deployment.
- After-launch items (§16).

Dependencies



Steps 9, 11, 12, 13 can be reordered after step 6 (each only needs the single-leg flow). 10 needs 9.

Revision #1

Created 13 October 2026 19:36:31 by Admin Qbit

Updated 13 October 2026 19:36:53 by Admin Qbit