People's Prepaid — Load Test Findings & Suggested Fixes

lan89onbox · 5 finding(s) from 11 scenarios · 2026-11-25 11:51

High	Medium	Low	Cause not yet known
3	1	1	0

Summary

ID	Severity	Effort	Area	Finding
F-1	High	Low	POS API — error handling	Overload returns a bare HTTP 500 with no message for the operator
F-3	High	Medium	POS API — overload behaviour	The system falls off a cliff instead of degrading gradually
F-6	High	Low	Voucher integrity	Vouchers left stock that the selling side never received
F-5	Medium	Low	Transaction switch	A small share of sales hang until the 30-second client timeout
F-8	Low	Low	Test environment	Electricity cannot be load-tested — the provider route is down

Every finding below was detected from the stored run data, not assumed — each one reproduced in this campaign. Effort is a rough engineering estimate, not a commitment.

Findings in detail

F-1 Overload returns a bare HTTP 500 with no message for the operator

High · Low effort

POS API — error handling

What we measured

68 response(s) came back as HTTP 500 with an empty body during voucher-sales-fleet-2min — on POST /api/voucher/buy.

Why it matters

Everywhere else this API reports business failures as HTTP 200 with errorState and a readable message, and the till's error handling expects that shape. A bare 500 gives the operator a blank failure they cannot act on, and gives support nothing to diagnose from.

Suggested fix

In VoucherController.BuyVoucher the switch client is built BEFORE the try block: `var client = await _switchFactory.CreateForUserAsync(username);` then `try { ... }`. That call reads the membership database (LegacyProfileService.GetProfileAsync), so under database pressure the exception escapes the controller entirely. Move the call inside the try so it degrades like every other failure there, and sweep the other switch-backed controllers (Balance, Electricity, Report, Info) for the same pattern.

How to confirm it worked

Re-run the scenario that produced the 500s and confirm the run's error breakdown contains no HttpError/500 — business refusals should all appear as bodyError with a message.

F-3 The system falls off a cliff instead of degrading gradually

High · Medium effort

POS API — overload behaviour

What we measured

In voucher-sales-fleet-2min, going from 700 to 900 tills took the share of sales served from 99.7% to 5.2%, and the time to complete a sale from a p95 of 2,376 ms to 25,297 ms — one step.

Why it matters

There is almost no warning band between healthy and broken, so alerting on latency creep will not give useful notice. Past the limit, work queues until it times out, which converts a capacity problem into a much longer outage and wastes switch capacity on requests nobody is waiting for any more.

Suggested fix

Add admission control to the switch-backed endpoints: a bounded concurrency limit with a short queue, returning a fast, explicit 'busy, retry' business response once full instead of accepting work that will time out. Failing fast keeps the tills that ARE being served fast, and turns a cliff into a plateau.

How to confirm it worked

Re-run the ramp one level past the limit and confirm the served share falls off gently while the time to sell for accepted sales stays inside its target.

People's Prepaid — Load Test Findings & Suggested Fixes

lan89onbox · 5 finding(s) from 11 scenarios · 2026-11-25 11:51

F-6 Vouchers left stock that the selling side never received

High · Low effort

Voucher integrity

What we measured

till-day-in-life: warehouse recorded 1,814 leaving stock, 1,807 received — 7 unaccounted. voucher-sales-fleet-2min: warehouse recorded 1,885 leaving stock, 1,642 received — 243 unaccounted. voucher-sales-fleet: warehouse recorded 813 leaving stock, 737 received — 76 unaccounted. voucher-sales-capacity: warehouse recorded 1,307 leaving stock, 1,302 received — 5 unaccounted.

Why it matters

Each unaccounted voucher is stock spent with nothing handed to a customer. At load-test volumes it is small; at production volumes the same mechanism is a recurring, silent shrinkage, and it is only visible here because the test reconciles against the warehouse.

Suggested fix

Make the recall-on-timeout path automatic (see F-5) so that any sale whose response is lost is reconciled rather than abandoned, and add a periodic reconciliation job comparing issued stock against delivered vouchers so the gap is alarmed rather than discovered.

How to confirm it worked

The stock reconciliation on a later selling run should show no gap.

F-5 A small share of sales hang until the 30-second client timeout

Medium · Low effort

Transaction switch

What we measured

366 sale(s) across 4 scenario(s) ran past the 30-second limit and were abandoned. These appear at every load level, including the smallest, so they are not purely a capacity effect.

Why it matters

An operator waits half a minute at the till and then sees a failure. Worse, the switch may still complete the sale after the till gives up, which spends a voucher nobody receives.

Suggested fix

Two changes. (a) Shorten the client timeout on the buy path to something an operator will tolerate and return a retryable message, rather than holding the till for 30 seconds. (b) On timeout, automatically recall the reserved request id — the switch is idempotent on it and the recall path already exists for offline-stock recovery — so a completed-but-unreturned sale is recovered instead of lost. Separately, ask the switch vendor why a small fraction of requests stall at all load levels.

How to confirm it worked

Confirm on a later run that the stock consumed by the warehouse matches the vouchers the test received — the gap between them is exactly this problem.

F-8 Electricity cannot be load-tested — the provider route is down

Low · Low effort

Test environment

What we measured

163 call(s) to POST /api/electricity/prevend returned 'No active route found on the third-party system'.

Why it matters

Electricity is a second, more expensive transaction shape (a pre-vend followed by a buy, so two switch round-trips). Until it can be exercised, every capacity figure in this report describes a voucher-only mix and therefore overstates a real mixed workload.

Suggested fix

Restore the electricity provider route on the dev switch, then add a mixed-transaction scenario weighted to the real product mix.

How to confirm it worked

The read-only switch probe should complete its electricity pre-vend calls without business errors.

Decisions

Record which findings are accepted for work, deferred, or rejected, and by whom.

ID	Finding	Decision	Owner / date

People's Prepaid — Load Test Findings & Suggested Fixes

lan89onbox · 5 finding(s) from 11 scenarios · 2026-11-25 11:51

F-1	Overload returns a bare HTTP 500 with no message for the operator	
F-3	The system falls off a cliff instead of degrading gradually	
F-6	Vouchers left stock that the selling side never received	
F-5	A small share of sales hang until the 30-second client timeout	
F-8	Electricity cannot be load-tested — the provider route is down	

Reviewed by | Approved by | Date