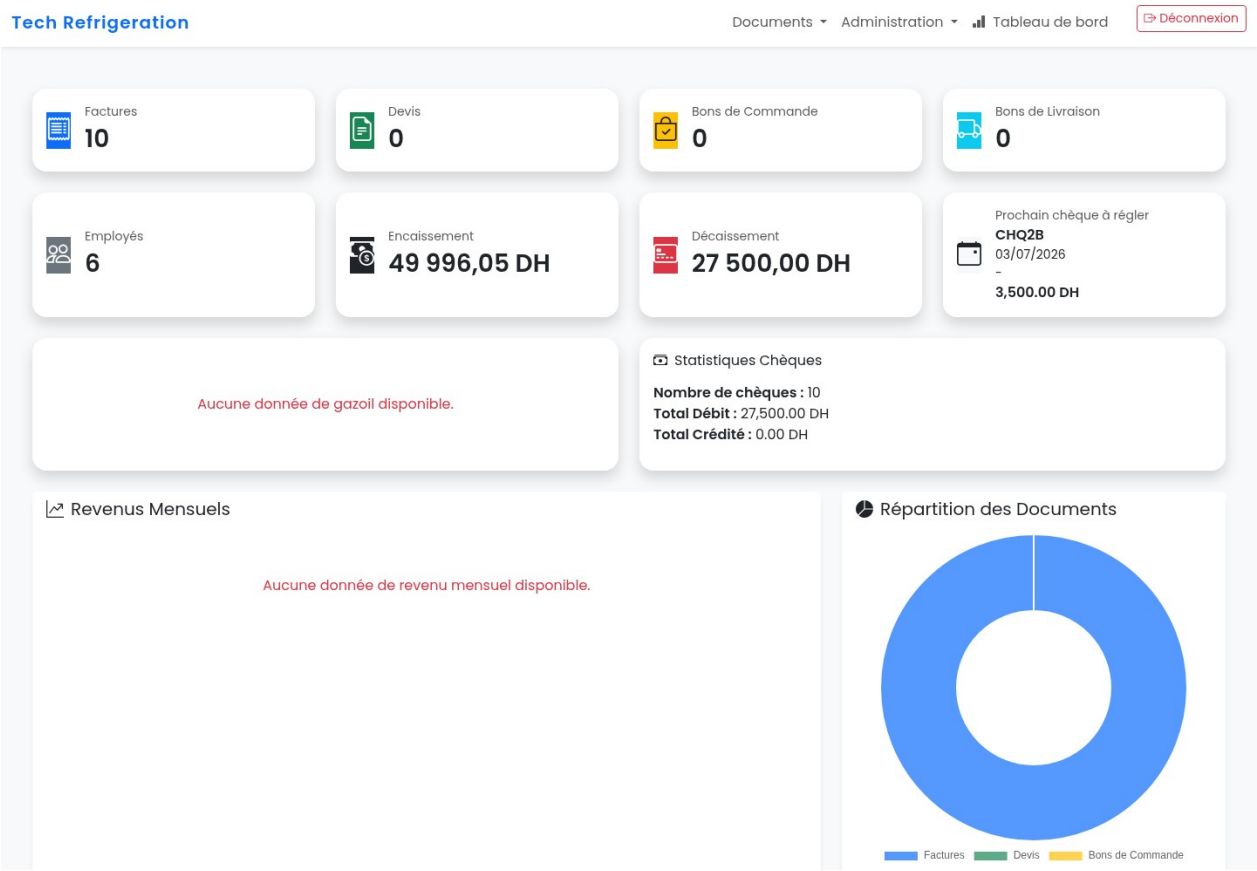
Centralizing field-service, commercial and financial operations in one Laravel platform

A real SME operations platform connecting customers, quotations, delivery documents, invoices, stock, interventions, workforce tasks and operating expenses.

STATUS	TEAM	DEMO	SOURCE
Real client delivery · Demo hardening	**Solo delivery**	**Protected demo planned**	**Private repository**



Primary visual evidence

ANASS BENAMEUR · FULL-STACK PRODUCT ENGINEER

The product, the problem and the proof

A refrigeration-services company was running customers, quotations, invoices, stock and technician work across paper, memory and disconnected files. I was brought in for four weeks, solo, to fix that. What shipped is a Laravel platform where a quotation, the invoice it becomes, the stock movement it triggers and the technician who executed the job are the same connected record, not four separate ones someone has to reconcile by hand.

4 weeks	Solo	18+	Desktop + mobile
ORIGINAL DELIVERY WINDOW	END-TO-END OWNERSHIP	CAPTURED OPERATIONAL MODULES	RESPONSIVE ACCESS

Context

Fractal Tech brought me in to build the operating system for a real refrigeration-services SME between July and August 2025: a company that sells, quotes, delivers and invoices refrigeration equipment and interventions, and had been doing all of it without a shared system. Four weeks, one developer, one client who needed something that actually worked on day one — every data-modelling decision had to be right the first time, because there wasn't slack in the schedule to redo the relational core halfway through.

My role and ownership

- Requirements analysis and workflow mapping
- Merise/UML relational data modelling
- Laravel backend and Blade interface
- Authentication and role-oriented access
- Deployment, documentation and user handoff

Personal contribution

Requirements, architecture, data model, backend, frontend, deployment, documentation and handoff.

Why the existing workflow needed a system

None of this was a technology failure so much as the predictable result of a business that grew its operations faster than its record-keeping, which is most SMEs at this exact stage. Quotations were typed fresh each time because nothing linked back to a customer's history. Invoices existed with no reliable trace to the job or the stock they consumed. Technicians' fieldwork wasn't visible to the office, and the office's paperwork wasn't visible to technicians. A stock count meant someone walking to the shelf, because nothing tracked what a completed job had actually consumed.

Before and after

BEFORE

- Scattered customer and supplier records
- Repeated manual preparation of business documents
- Weak traceability between commercial and field activity
- Disconnected stock, cash, fuel and repair records
- No unified workspace for administration and technicians

AFTER

- Centralized master data and operational records
- Structured quotation-to-invoice workspaces
- Searchable history across documents and interventions
- Inventory, cash and operating-cost modules in one platform
- Responsive access for office and field use

Meaningful result

One system replaced disconnected customer, document and operational records.

How the system works

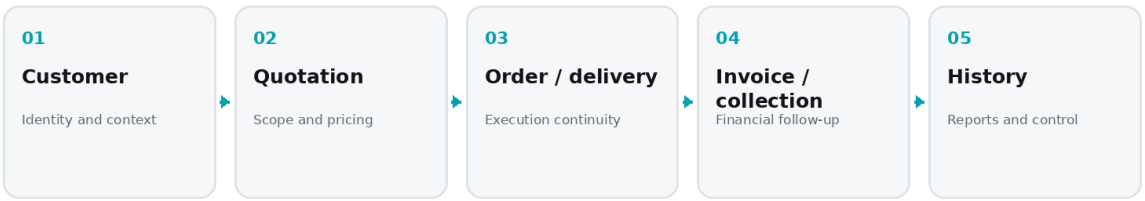
Everything hangs off five connected states rather than five independent modules that happen to share a login page: a customer record establishes identity and history; a quotation captures scope and price against that customer; an order/delivery step preserves execution continuity once the quotation is accepted; an invoice closes the loop financially and stays permanently linked back to the job that generated it; and a history layer makes every one of those states searchable across customers, documents and interventions.

Operational workflow

01	02	03	04	05
Customer	**Quotation**	**Order / delivery**	**Invoice / collection**	**History**
Identity and context	Scope and pricing	Execution continuity	Financial follow-up	Reports and control

SYSTEM ARCHITECTURE

Integrated Field Service & Operations Platform



Evidence-led architecture diagram · generated from the verified project workflow

Stock movements attach to the same relational core: an intervention that consumes a part writes a stock-ledger entry, not a manual adjustment someone has to remember to make later. Technician task assignments and operating expenses follow the same pattern. The result is that a manager can open one customer and see their full quotation-to-invoice history, the stock consumed against it, and which technician handled it, in one place, without cross-referencing five different documents. The sequence below is exactly what ships in the delivered system:

Architecture and decisions

Laravel 10 and PHP 8 over anything more exotic — a four-week SME engagement needs a framework where the distance from data model to shipped feature is short, not a chance to experiment with something unproven. MySQL for transactional integrity on financial documents. Blade instead of a separate SPA: it kept delivery cohesive, ran comfortably on the ordinary office hardware the client actually has, and still supported distinct views per role without the overhead of a second deployment target under a four-week clock.

Critical engineering decisions

DECISION 01

Connected records over isolated CRUD

Customers, documents, stock movements, interventions and financial events share relational context, reducing duplicate entry and preserving operational history.

DECISION 02

Server-rendered operational interface

Blade kept delivery cohesive and pragmatic for ordinary business hardware while supporting role-specific views.

DECISION 03

Documents as workflows

Quotations, orders, delivery notes, invoices and collections were treated as business states rather than disconnected files.

DECISION 04

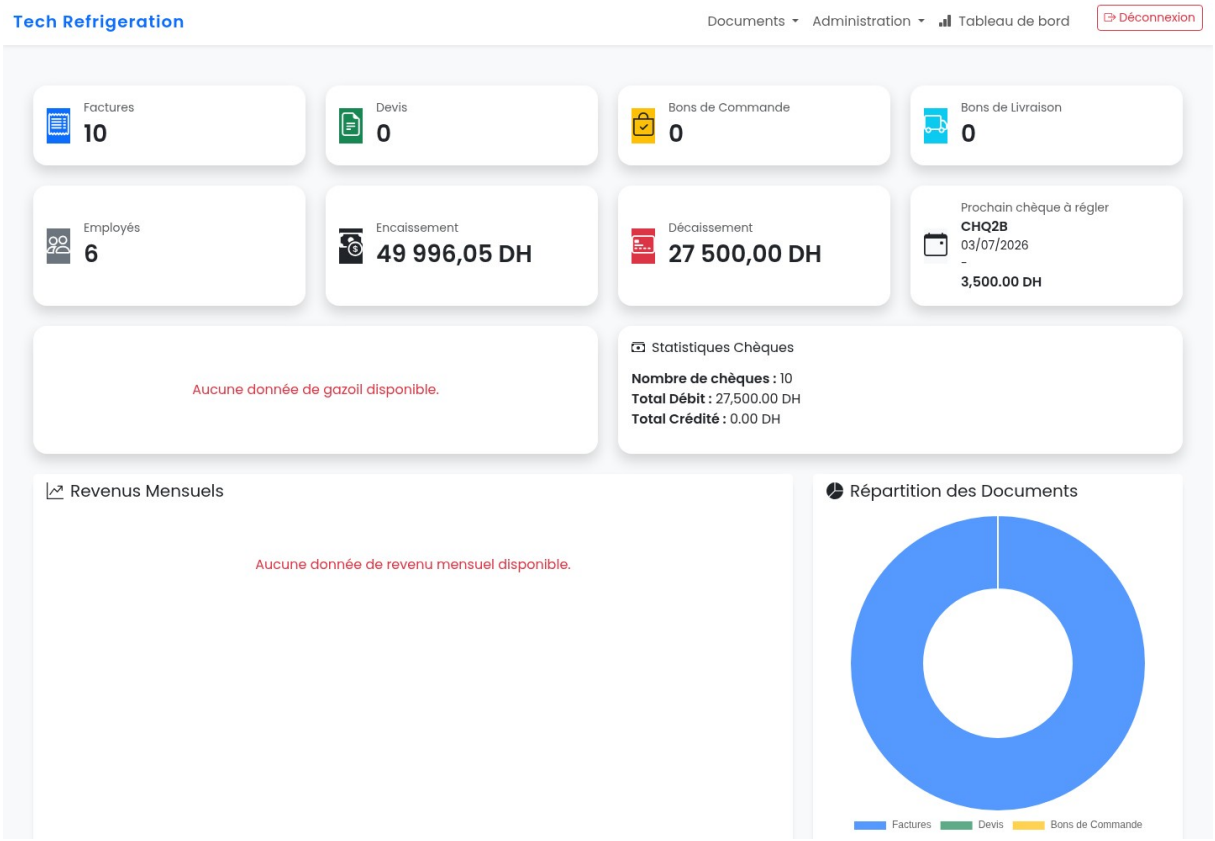
Safe proof environment

The portfolio environment uses synthetic data and an isolated database, proving breadth without exposing client information.

Technology stack

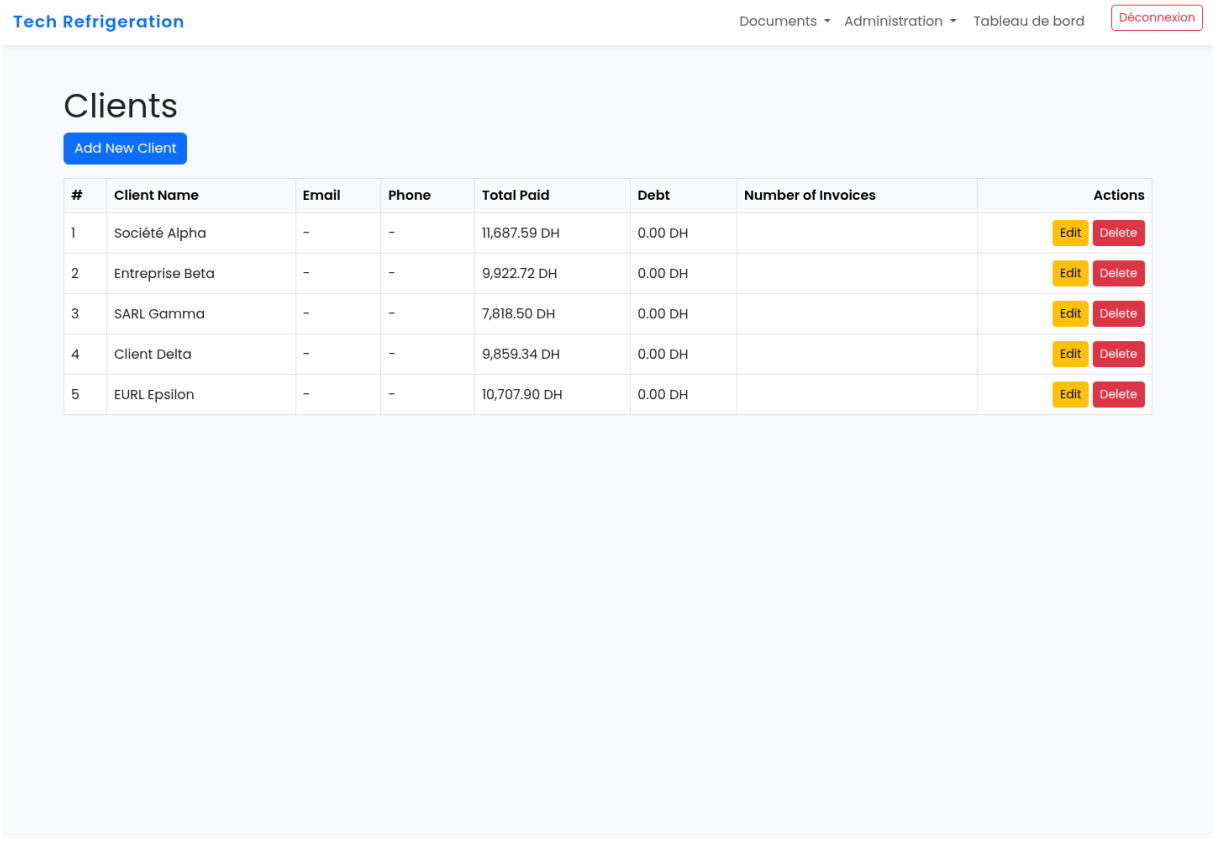
Laravel 10	PHP 8	MySQL	Blade	MVC	Role-based access

Evidence 1 of 4



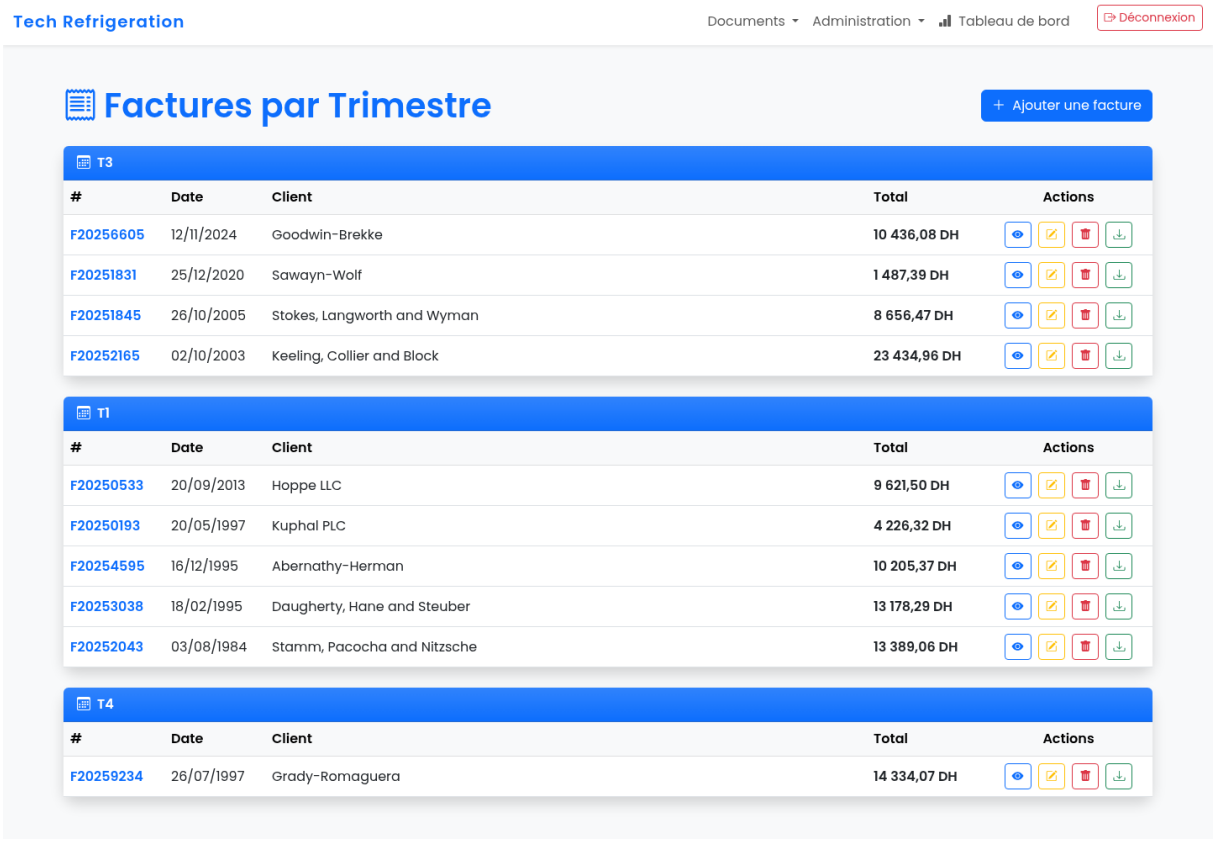
Operational cockpit combining document, cash and activity indicators.

Evidence 2 of 4



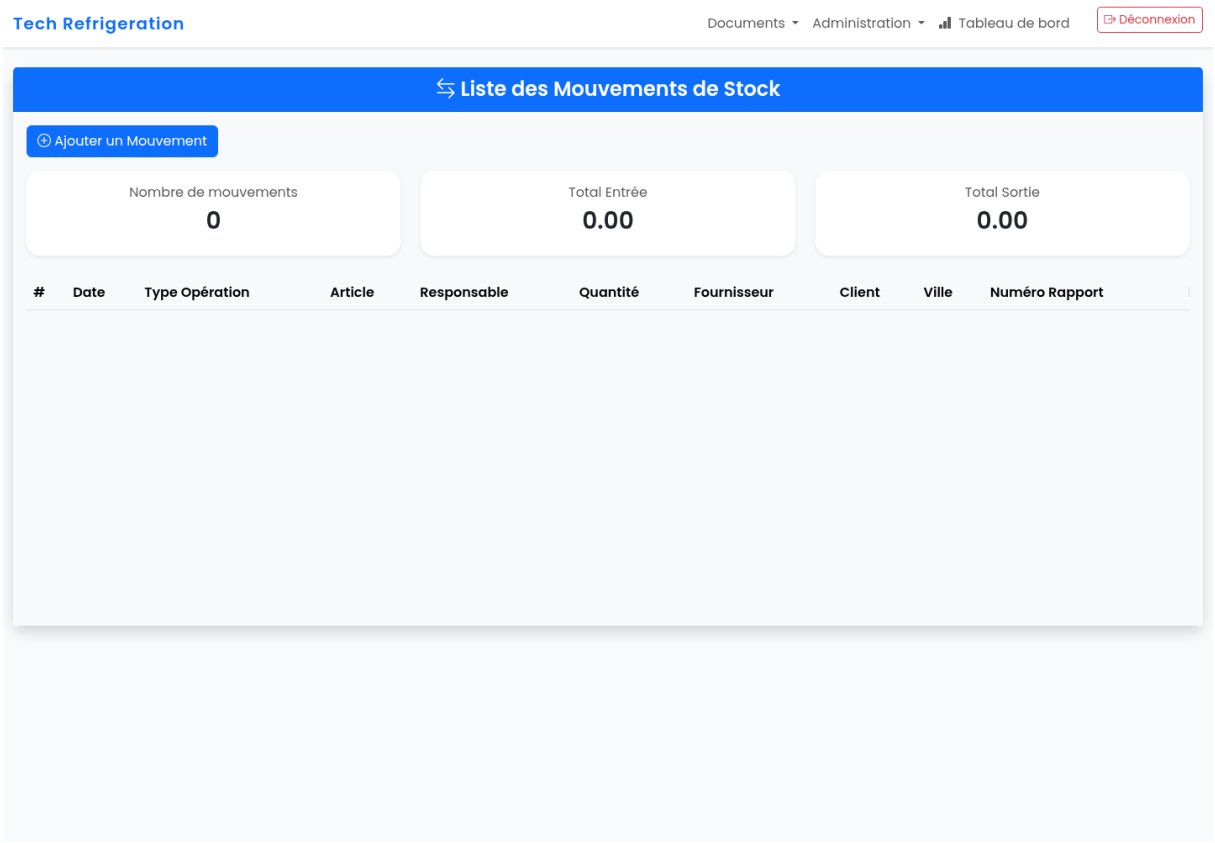
Central customer directory feeding commercial and intervention workflows.

Evidence 3 of 4



Invoice workspace preserving financial document history.

Evidence 4 of 4



Inventory movement ledger for inbound, outbound and adjustment events.

Value, proof and honest limitations

Operational impact

The client went from five disconnected record types to one searchable operational history, with management able to see documents, stock and field activity in the same place for the first time. That is the actual deliverable — not a UI, a reconciliation problem that stopped existing. The expanded source shows 33 migrations, 32 controllers and 28 models across 120 Blade views, which is the shape of a real relational business system, not a demo CRUD app with a login screen bolted on.

Verified evidence

- Real SME delivery recorded in the CV
- 18+ captured operational modules
- Desktop and mobile responsive captures
- Expanded source inventory includes 33 migrations, 32 controllers, 28 models and 120 Blade views
- Original handoff included technical documentation and a user guide

Current limitations

- The later expanded source requires a dedicated hardening release before public deployment.
- Automated tests are insufficient for document transitions, permissions and stock reconciliation.
- Several screenshots use sparse synthetic data and prove interface scope more than production volume.
- The public case study distinguishes the original delivery from later experimental expansion.

Current status

STATUS	TEAM	DEMO	SOURCE
Real client delivery · Demo hardening	**Solo delivery**	**Protected demo planned**	**Private repository**

WHY THIS PROJECT MATTERS

One system replaced disconnected customer, document and operational records.

This case study is designed to be inspectable: strong claims are connected to screenshots, source evidence or explicitly stated limitations.