

Centralizing institutional stock movements, threshold alerts and reporting

A Laravel inventory portal built from administrative requirements to centralize products, movements, low-stock alerts and formal reporting.

STATUS	TEAM	DEMO	SOURCE
Academic delivery · Visual evidence pending	Solo internship project	Demo pending source recovery	Source not currently published

EVIDENCE STATUS

Documented project · interface captures still required

VERIFIED

CV delivery record

Stakeholder analysis, Laravel implementation, alerts and exports are documented.

DOCUMENTED

Workflow and data model

Products, entries, exits, thresholds and reporting form the stated scope.

PENDING

Visual proof

Dashboard, movement history, alert and export screenshots are not supplied.

NEXT GATE

Source recovery

Run the portal with synthetic data, capture evidence and validate the final case study.

No interface has been fabricated. The portfolio discloses the evidence gap explicitly.

Evidence status map - no interface has been fabricated.

ANASS BENAMEUR · FULL-STACK PRODUCT ENGINEER

The product, the problem and the proof

During a final internship at Faculté Chariaa in Fès, I built a stock-management portal directly from requirements gathered with administrative stakeholders — the kind of engagement where continuity and traceability matter more than visual polish, and where the requirements-gathering itself is half the engineering work.

Feb–May 2025 INTERNSHIP PERIOD	Full cycle ANALYSIS TO DOCUMENTATION	Thresholds CONFIGURABLE STOCK ALERTS	2 formats EXCEL AND PDF REPORTING
---	---	---	--

Context

A public university environment, where a stock system has to survive staff turnover and produce reports that satisfy an administrative audit trail, not just look good in a demo. Between February and May 2025, I worked directly with administrative stakeholders to define the requirements before writing a line of code.

My role and ownership

- Stakeholder requirements analysis
- Merise data modelling
- Laravel backend and Blade interface
- Functional testing
- Documentation and handoff preparation

Personal contribution

Requirements, modelling, implementation, testing and documentation.

Why the existing workflow needed a system

Manual stock checking meant nobody could answer "how much do we actually have" without walking to the shelf. Low-stock situations were caught late because nothing was watching thresholds. And there was no single, reconstructable history of what entered and left inventory — exactly the kind of gap an administrative audit finds.

Before and after

BEFORE	AFTER
• Manual or fragmented product records • Stock levels checked reactively • Movement history difficult to reconstruct • Reports assembled separately • Limited administrative visibility	• Centralized product and stock records • Entry and exit movement history • Automatic threshold-alert logic • Excel and PDF exports • Documented institutional workflow

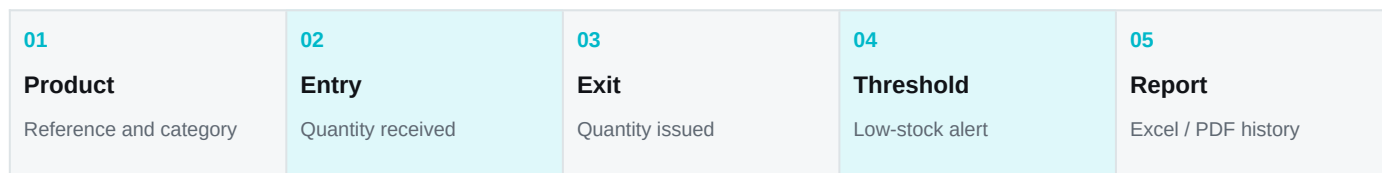
Meaningful result

Centralized movement history, threshold alerts and formal inventory reporting.

How the system works

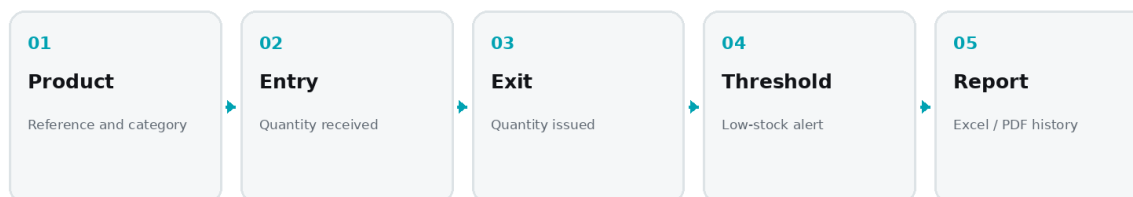
Products and movements share one relational core, modelled with Merise before a line of code was written and reviewed with the stakeholder group so the data model was agreed on, not just assumed. Configurable thresholds trigger alerts when stock drops below a set level per product, rather than relying on someone remembering to check manually.

Operational workflow



SYSTEM ARCHITECTURE

Institutional Stock Management Portal



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

Every movement — entry, exit, adjustment — writes to a history that Excel and PDF reports generate from directly, so a printed report reflects the same data an administrator would see on screen, not a separately maintained summary that can quietly drift out of sync with reality. The lifecycle, step by step:

Architecture and decisions

The documented stack is Laravel, MySQL, Blade and MVC in a straightforward structure — the standard, unglamorous choice, deliberate for an institution that needs long-term maintainability over novelty. Requirements and data relationships were modelled with Merise before implementation began.

Critical engineering decisions

DECISION 01 Model movement history explicitly Inventory must remain explainable through entries and exits rather than only a mutable quantity field.	DECISION 02 Thresholds as an operational control Administrators need early warning instead of discovering shortages during a request.
DECISION 03 Formal export formats Excel supports analysis while PDF supports stable administrative circulation.	DECISION 04 Document the complete lifecycle Institutional adoption depends on requirements, testing and documentation as much as on code.

Technology stack

Laravel	PHP	MySQL	Blade	Merise	Excel / PDF
---------	-----	-------	-------	--------	-------------

Evidence 1 of 1

EVIDENCE STATUS

Documented project · interface captures still required

VERIFIED CV delivery record Stakeholder analysis, Laravel implementation, alerts and exports are documented.	DOCUMENTED Workflow and data model Products, entries, exits, thresholds and reporting form the stated scope.	PENDING Visual proof Dashboard, movement history, alert and export screenshots are not supplied.	NEXT GATE Source recovery Run the portal with synthetic data, capture evidence and validate the final case study.
---	---	---	--

No interface has been fabricated. The portfolio discloses the evidence gap explicitly.

Evidence map explaining what is verified and what remains pending.

Value, proof and honest limitations

Operational impact

The delivered system gave the institution centralized product records, a reconstructable movement history, automatic low-stock alerts, and reports generated straight from the operational data instead of assembled separately — a complete requirements-to-documentation cycle over a February to May 2025 internship.

Verified evidence

- CV documents the internship, stack and complete delivery cycle
- Threshold alerts, movement history and Excel/PDF exports are consistently documented
- Requirements analysis and Merise modelling were part of the project
- Visual proof remains pending and is clearly disclosed

Current limitations

- No project screenshots were included in the supplied screenshot archive.
- No source archive or live runtime was supplied for independent verification.
- The project should not be presented as a final visual showcase until dashboard, movement, alert and export captures are added.
- No numerical impact is claimed without institutional measurement.

Current status

STATUS	TEAM	DEMO	SOURCE
Academic delivery · Visual evidence pending	Solo internship project	Demo pending source recovery	Source not currently published

WHY THIS PROJECT MATTERS

Centralized movement history, threshold alerts and formal inventory reporting.

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