

ERP System Analyst Portfolio

Septarina



Septa
septarina

Executive Summary & Professional Profile

•**Name:** Septarina

•**Role:** ERP System Analyst (Odoo Specialist)

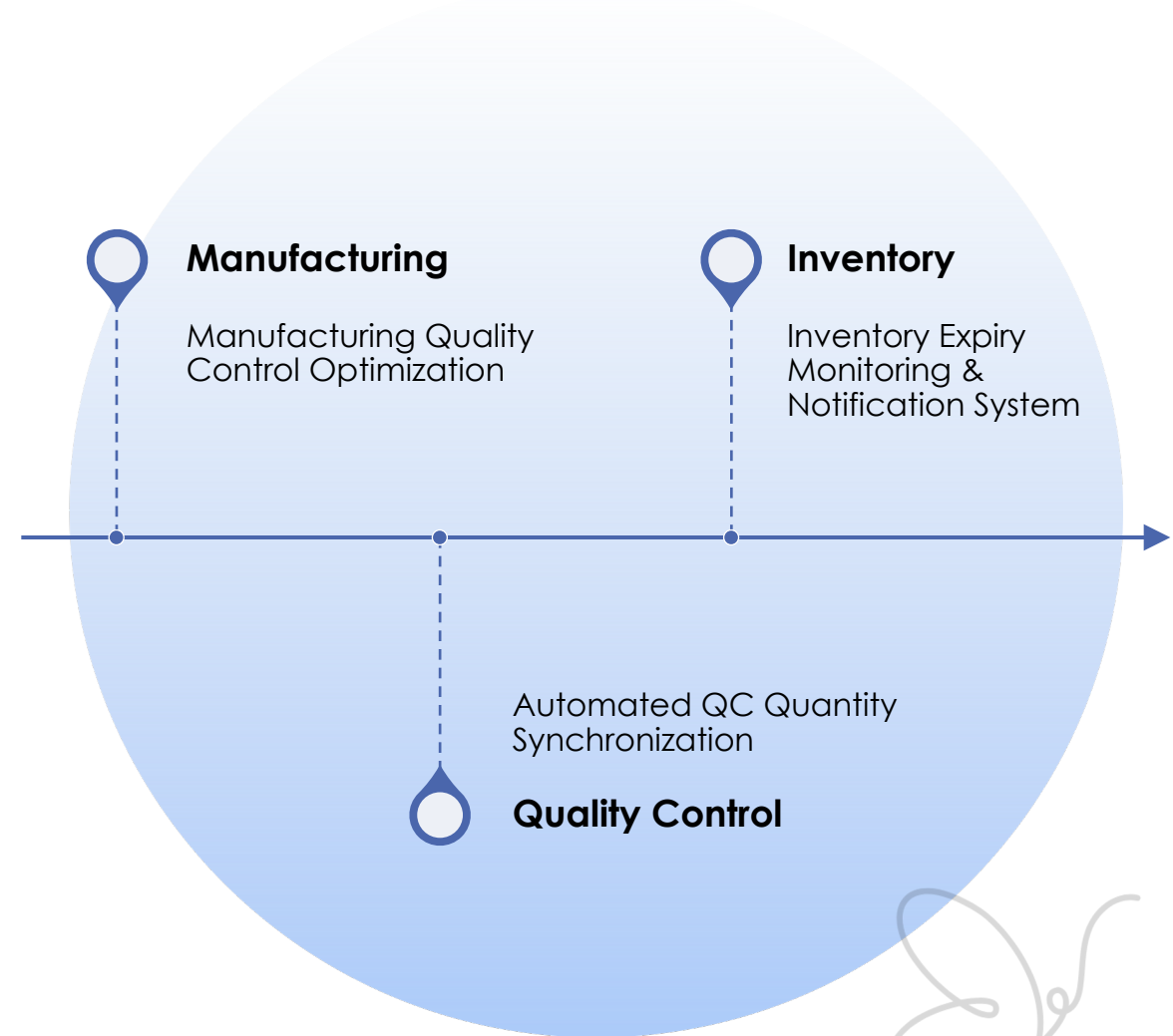
•**Core Expertise:** Manufacturing Execution Systems (MES), Inventory Traceability, and Quality Assurance Automation.

•**Value Statement:** Expert in bridging the gap between complex business operations and scalable system designs. I specialize in creating detailed Functional Design Specifications (FDS) that optimize manufacturing and inventory workflows.

Septa
septarina

Portfolio Highlights: 3 Strategic ERP Projects

1. Manufacturing Quality Control Optimization
2. Automated QC Quantity Synchronization
3. Inventory Expiry Monitoring & Notification System





PROJECT 1

Manufacturing QC Optimization

Focus: Logic Hierarchy & User Experience.

Solution: Designed a Nested Priority Logic (Variant > Product > Category) to filter QC checklists.

Impact: Eliminated operational friction by ensuring only one relevant QC list is displayed per Work Order, reducing data redundancy.

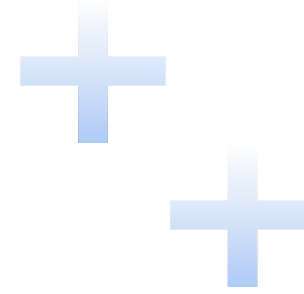
Project Overview

- **Module Name:** manufacturing_quality_control.
- **Role:** System Analyst.
- **Core Objective:** To streamline the quality inspection process by ensuring only the most relevant Quality Control (QC) list is displayed to operators, preventing data redundancy and user confusion.
- **Technical Dependency:** Integrated with the Odoo Manufacturing module

The Challenge

- **System Overload:** In standard manufacturing environments, a single Work Order may trigger multiple QC lists simultaneously.
- **Operational Friction:** Having more than one QC list active at once causes confusion for production operators and leads to inefficient data entry.
- **Requirement:** A prioritized logic was needed to filter and display a single, specific QC list based on the level of product specification

Functional Design & System Logic



1. Hierarchical Selection Logic

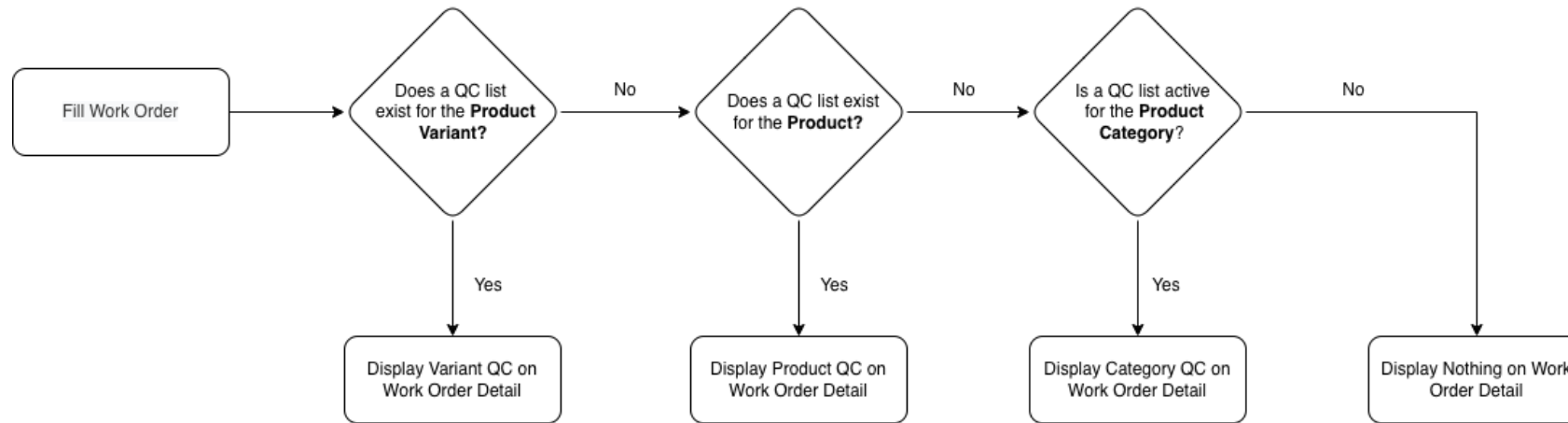
To prevent operational confusion, I designed a nested logic that prioritizes the most specific Quality Control (QC) list. The system follows this execution order:

- **Primary Priority (Product Variant):** If a QC list is assigned to the specific Product Variant, the system displays this list and ignores all others.
- **Secondary Priority (Product):** If no Variant-level QC exists, the system checks for a QC list at the Product Template level.
- **Tertiary Priority (Product Category):** If neither of the above is found, the system defaults to the broadest level—the Product Category.
- **Null State:** If no QC list is active at any of these three levels, the Work Order will display no inspection requirements to maintain data cleanliness.

2. Technical Execution

- **Trigger:** The logic is executed during the creation or confirmation of a Manufacturing Work Order.
- **Menu Integration:** Seamlessly integrated into Manufacturing > Operations > Manufacturing Orders > Work Orders.
- **Data Integrity:** By enforcing a "Single List" rule, the system prevents redundant inspection records and ensures that auditors only see the most relevant quality data

System Design: Hierarchical Quality Control



Priority Level	Placement Condition	System Action (The Logic)
Priority 1	Product Variant is active.	The system displays only the QC list for the specific Product Variant.
Priority 2	Product is active (and no Variant QC exists).	The system displays the QC list for the Product.
Priority 3	Product Category is the only one active.	The system displays the QC list for the Product Category.
Null State	No placements are active.	The system displays nothing.

Business Impact & Strategic Value

1. Operational Excellence

- **Eliminated Cognitive Load:** Production operators no longer need to decide which QC list to fill out. The system provides a single, clear "Source of Truth".
- **Error Reduction:** By automating the selection process, we eliminated the risk of operators accidentally using outdated or generic category-level checks when a specific variant-level check was required.

2. Scalability & Efficiency

- **Streamlined Onboarding:** New operators can perform inspections with minimal training, as the system simplifies the interface to show only what is necessary for the specific task.
- **Process Efficiency:** Reduces the time spent on Work Order administration and validation by 100% regarding QC list selection.

Technical Specifications & Developer Guidelines

This slide serves as evidence of high-level technical documentation, demonstrating the ability to translate business requirements into detailed, actionable instructions for the development team (including Menu Paths, Module Dependencies, and Logic Triggers).

Module Name: **manufacturing_quality_control**

Is an Existing Module: ☒ (If checked, then edit the module, if unchecked, create new module)

Dependency: **Manufacturing**

****NOTE: Please copy the Module Details below to the module's actual description so it can be viewed in Apps****

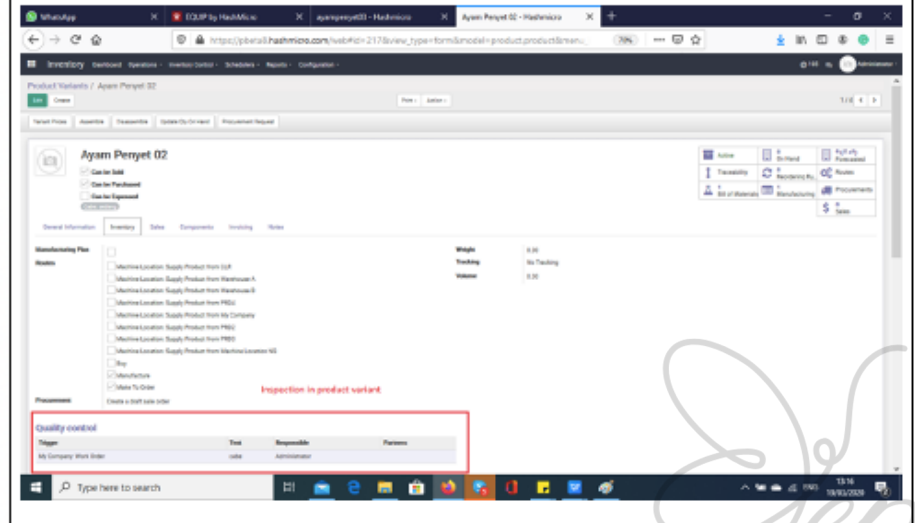
Module Details:

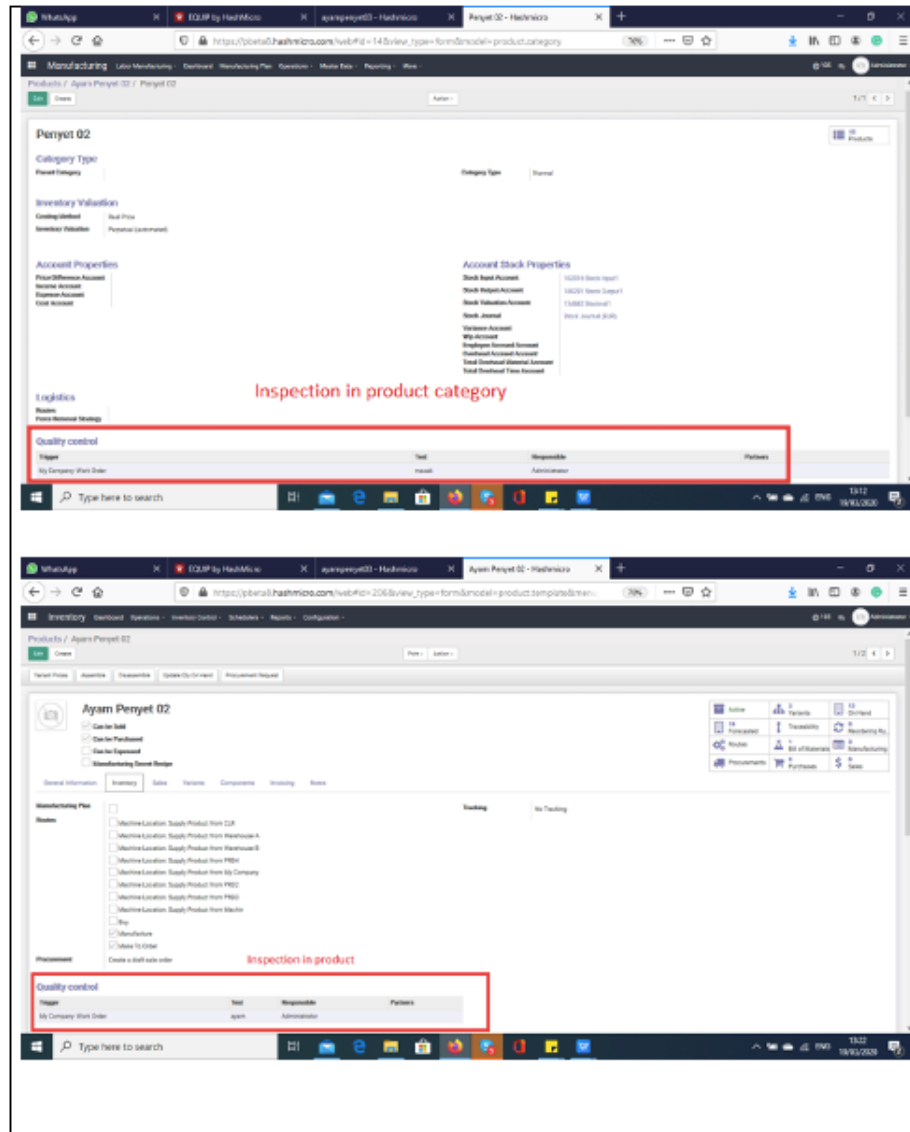
1) Only show 1 QC list in Manufacturing Work Order if there are more than 1

Menu Item	Manufacturing > Operations > Manufacturing Orders > Work Orders		
New / Existing Object	New / Existing Menu Item	Access Rights to New Object / Menu Item	Form View Fields (Add / Edit / Hide / Remove)
Existing,	Existing	Existing	
View Filters	Views Available (Form, List, Kanban, Calendar, Pivot, Graph, Gantt)	Kanban View Fields (Only if you have Kanban View)	List View Columns (Only if you have List View)
Only show 1 QC			

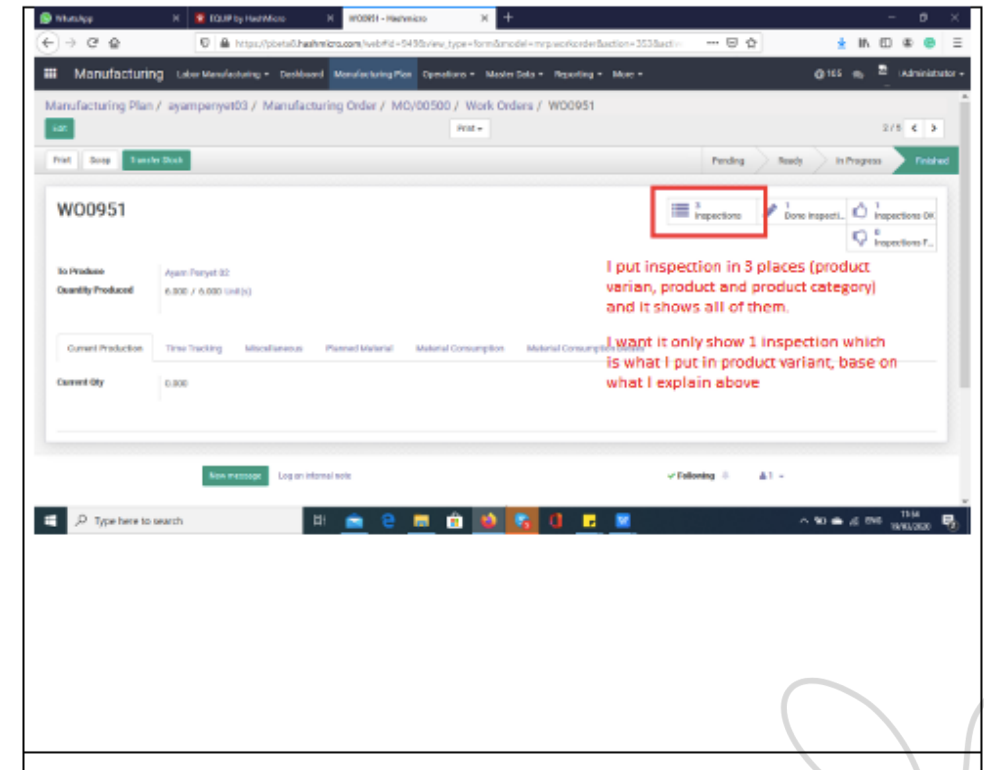
Functions, Buttons, Triggers, Schedulers, Onchange events & other remarks

- ✓ QC placement
 1. Product variants
 2. Product
 3. Product category
- ✓ QC on work order manufacturing will only show 1 QC :
 - IF placement 1,2 and 3 is active, OR 1 and 2 active, OR 1 and 3 active, OR only 1 active, THEN only show QC in placement 1 (Product Variant)
 - IF placement 2 and 3 is active, OR only 2 active, THEN only show QC in placement 2 (Product)
 - IF placement 3 is the only one active, THEN show QC in placement 3 (Product Category)
 - IF there is no placement active, THEN show nothing





Application Screenshot: Provided to assist developers in locating and implementing the required changes according to the specifications.





PROJECT 2

Automated QC Quantity Synchronization

Focus: Data Integrity & Real-Time Reporting.

Solution: Developed an automated synchronization between Finished Good Quantity and Inspection Quantity fields.

Impact: Accelerated the QA approval process by providing supervisors with instant, accurate data alignment without manual cross-checking.

Project Overview

- **Document Type:** Functional Design Specification.
- **Role:** System Analyst.
- **Tech Stack:** Odoo ERP, Python (Logic), PostgreSQL (Database structure).
- **Module Name:** manufacturing_quality_control.
- **Dependency:** Manufacturing.
- **Menu Path:** Manufacturing > Operations > Manufacturing Orders.
- **Logic Trigger:** Sync quantity field in list view with finished good value from production.



The Challenge

The client's production team found that the standard Odoo Inspection view did not provide immediate clarity on production output. They needed the **"Quantity"** field in the tree view to automatically synchronize with the **"Finished Good Quantity"** to speed up decision-making for supervisors.

My Solution (Functional Design)

As the lead analyst, I translated this operational pain point into a precise technical blueprint:

- **System Analysis**

Identified the dependency within the manufacturing_quality_control module.

- **Functional Specification**

Defined the exact menu path (Manufacturing > Operations > Manufacturing Orders) and the necessary logic triggers for the developers.

- **Bridge to Development**

Provided clear instructions on changing the QUANTITY value logic in the list view to ensure 100% data alignment with finished goods.

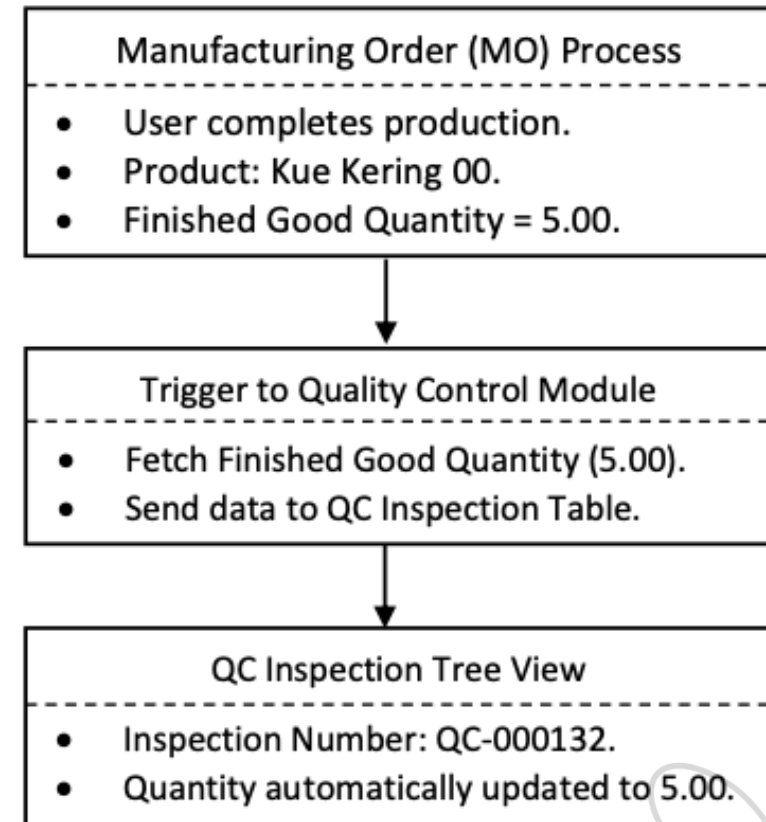


Figure 1: Functional Data Flow for Automated Quantity Sync.

Developer Guidelines

Technical documentation demonstrating the ability to translate complex business requirements into detailed, actionable instructions for development teams—including Menu Paths, Module Dependencies, and Logic Triggers. It also serves as a visual reference for developers to accurately identify and execute system changes.



Module Name: **manufacturing_quality_control**

Is an Existing Module: ☒ (If checked, then edit the module, if unchecked, create new module)

Dependency: Manufacturing

****NOTE: Please copy the Module Details below to the module's actual description so it can be viewed in Apps****

Module Details:

1) Change Quantity in Inspections into Finished Good Quantity

Menu Item	Manufacturing > Operations > Manufacturing Orders		
New / Existing Object	New / Existing Menu Item	Access Rights to New Object / Menu Item	Form View Fields (Add / Edit / Hide / Remove)
Existing	Existing	Existing	
View Filters	Views Available (Form, List, Kanban, Calendar, Pivot, Graph, Gantt)	Kanban View Fields (Only if you have Kanban View)	List View Columns (Only if you have List View)

Functions, Buttons, Triggers, Schedulers, Onchange events & other remarks

1. Change QUANTITY value in tree view Inspections into Finished Good quantity value

INSPECTION NAME	RESPONSIBLE	TEST	QUANTITY	PRODUCT	PICKING	LOT	PROCESS	STATE
QC-000130	Administrator	oqt	15.00	Kue Kering 00				Ready
QC-000131	Administrator	oqt	15.00	Kue Kering 00				Ready
QC-000132	Administrator	oqt	15.00	Kue Kering 00				Ready

Manufacturing Orders / MO/00570 / Work Orders / WO1138 / Inspections / QC-000132

1/3 < >

CREATE EDIT CONFIRM DRAFT READY PENDING SUPERVISOR APPROVAL QUALITY SUCCESS

Inspection QC-000132

Test: oqt
Responsible: Administrator
Reference: WO1138
Quantity: 15.00
Product: Kue Kering 00
Picking: Lot
Material Consumed: SAC0002

Date: 05/04/2020 07:00
Success: ☐
Auto generated: ☐
Change page: ☐
Finished Good: 5.00
Lost or scrap Good: 0.00

Questions: Notes

QUESTION	QUESTION TYPE	QUALITATIVE VALUE	QUANTITATIVE VALUE	UOM	UNIT VALUES	SUCCESS?
oqtNotes	Qualitative		0.0000	kg		

The Impact

- **Reduced Manual Error:** Eliminated the need for staff to manually verify quantities between different views.
- **Efficient Delivery:** By providing a structured functional document (including form view fields and access rights), I reduced the development turnaround time and ensured accurate system delivery.



PROJECT 3

Inventory Expiry Monitoring & Notification System



Focus: Automation & Waste Reduction.

Solution: Implemented an Automated Scheduler to monitor Lot/Serial Number expiration dates with proactive system alerts.

Impact: Significantly reduced product waste and ensured 100% compliance with FEFO (First Expired, First Out) inventory principles.

Project Overview

- **Module Name:**
product_lot_serial_number_expire_date
- **Role:** System Analyst / Functional Consultant
- **Core Objective:** To enhance inventory traceability by adding automated expiration tracking and proactive notification systems for perishable goods.
- **Technical Dependency:** Odoo Inventory Module.

The Challenge

- **Wastage Risk:** Without a visible expiration date on Lot/Serial numbers, warehouse teams struggle to prioritize stock based on the FEFO (First Expired, First Out) principle.
- **Manual Monitoring:** Checking expiration dates manually for thousands of stock items is time-consuming and prone to human error, leading to expired products being sold or wasted.

Functional Design: Smart Expiry Tracking

I designed a two-fold solution to automate the monitoring process:

- **Field Extension:** Added a dedicated Expire Date field to the stock.production.lot object, accessible via Inventory > Control > Lots/Serial Numbers.
- **Automated Scheduler (Notification Logic):**
 - **Condition:** If Expiry Date $\leq$ (Today + X Days)
 - **Interval:** Running daily.
 - **Action:** Triggers a system popup/notification: *"This product will expire in X days"*.

Business Impact

Waste Mitigation: Proactive alerts allow the warehouse team to prioritize stock sales or movements before expiration.

Regulatory Compliance: Ensures the company adheres to strict health and safety standards by maintaining a "zero-expired-stock" policy.

Enhanced Traceability: Provides a clear audit trail for every lot/serial number produced or received.



Technical Specifications & Developer Guidelines



Requirement	Detail
Object	Existing (stock.production.lot)
Access Right	Manager (CRUD), User (Read)
Views	Form View and List View
Trigger	Daily System Scheduler



Feature	Menu Path	Technical Logic
Expiry Field	Inventory > Control > Lots/Serial Numbers	Added date object to stock.production.lot.
Expiry Scheduler	Settings > Technical > Automation	Daily interval scheduler to scan upcoming expired lots.
Validation	Onchange / Scheduler	Logic to compare current date vs. user-defined threshold (X days).

Module Name: Product_lot_serial_number_expire_date

Is an Existing Module: ☒ (If checked, then edit the module, if unchecked, create new module)

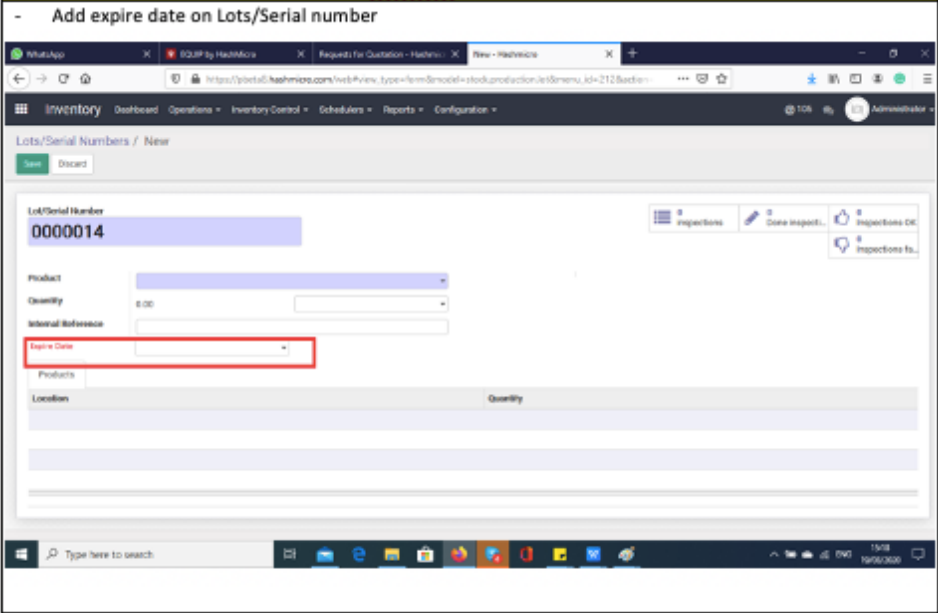
Dependency: Inventory

****NOTE: Please copy the Module Details below to the module's actual description so it can be viewed in Apps****

Module Details:

1) Add ~~Expire~~ Date on Lots/Serial Numbers

Inventory > Inventory Control > Lots/Serial Numbers			
New / Existing Object	New / Existing Menu Item	Access Rights to New Object / Menu Item	Form View Fields (Add / Edit / Hide / Remove)
Existing	Existing	Manager CRUD	Add: - Expire date: Date of stock.production.lot object
View Filters	Views Available (Form, List, Kanban, Calendar, Pivot, Graph, Gantt)	Kanban View Fields (Only if you have Kanban View)	List View Columns (Only if you have List View)
	Form, List		
Functions, Buttons, Triggers, Schedulers, Onchange events & other remarks			
- Add expire date on Lots/Serial number			



2) Add scheduler: before the products on Lots/Serial Numbers expire on x days, prompt the user "This product will expire in x days?"

Inventory > Inventory control > Lot/Serial Numbers									
New / Existing Object	New / Existing Menu Item	Access Rights to New Object / Menu Item	Form View Fields (Add / Edit / Hide / Remove)						
Existing	Existing	Manager CRUD User Read							
View Filters	Views Available (Form, List, Kanban, Calendar, Pivot, Graph, Gantt)	Kanban View Fields (Only if you have Kanban View)	List View Columns (Only if you have List View)						
			-						
Functions, Buttons, Triggers, Schedulers, Onchange events & other remarks									
- Add Scheduler: Condition: (if product on lot/serial number expire date <= (today + x days)) Interval: 1 Day Action: Show popup This product will expire in x days? <table border="1"><thead><tr><th>Lot/Serial number</th><th>Product</th><th>Expire Date</th></tr></thead><tbody><tr><td></td><td></td><td></td></tr></tbody></table> [OK]				Lot/Serial number	Product	Expire Date			
Lot/Serial number	Product	Expire Date							

Technical documentation for development teams—including Menu Paths, Module Dependencies, and Logic Triggers. It also serves as a visual reference for developers to accurately identify and execute system changes.



Thank you

Septarina

septha.github.io

Indonesia (Open to Remote/Relocation)

