

2. Các Module Tùy chỉnh (Custom Addons)

2. Các Module Tùy chỉnh (Custom Addons)

Dự án hiện tại đã hợp nhất các tính năng vào một module lõi duy nhất là `qlv` (Quản lý Vàng).

Module `qlv`

- **Mục đích:** Quản lý toàn diện nghiệp vụ cửa hàng vàng (Giá, Mua/Bán, Trade-in, Cầm cố).
- **Các thành phần chính:**

a. Quản lý Giá & Sản phẩm (`product_price_manager` cũ)

- Sử dụng `list_price` (Bán) và `standard_price` (Mua/Vốn).
- Lưu lịch sử thay đổi giá (`product.price.history`).

b. Nghiệp vụ Mua Bán & Trade-in (`sale.order` extension)

- **Cân bằng tiền (Auto-Balance):**
 - Sử dụng **Sản phẩm Tiền mặt** (Money Product) để đại diện cho tiền.
 - Logic tự động thêm dòng Tiền mặt để đưa `amount_total` về 0 (Tiền Hàng = Tiền Khách Trả).
 - Hỗ trợ trigger cập nhật lại tiền khi sửa đơn hàng (đặc biệt sau khi Thanh lý).
- **Trade-in:**
 - Dòng hàng có `is_trade_in=True` sẽ được xử lý là Hàng Mua (Receipt) thay vì Hàng Bán.
 - Giá âm để giảm trừ tổng thanh toán.
- **Tích hợp Cầm cố:**
 - Hiển thị danh sách `pending_pawn_ids` (Cầm cố đang hiệu lực) ngay trên form bán hàng.
- **Quản lý Trạng thái & UI Fix:**
 - Field `custom_state`: Ánh xạ lại trạng thái gốc (draft, sale, done) sang ngôn ngữ nghiệp vụ (Đang lập phiếu, Hợp đồng...).
 - **Locking:** Override hàm `write()` để chặn sửa đổi khi đơn ở trạng thái `done` hoặc `invoiced`.
 - **JS Patch (`force_sort_patch.js`):** Can thiệp vào `ListRenderer`.
 - Tự động phát hiện khi User "Thêm hàng".
 - Chặn hành vi nhảy Focus khi danh sách được sắp xếp lại.
 - Cơ chế **Polling** tìm lại dòng vừa nhập để restore focus (cursor).

c. Nghiệp vụ Cầm cố (`pawn.order` - NEW)

- **Model:** `pawn.order` và `pawn.order.line`.
- **Workflow:**
 - **Confirm:** Tạo phiếu nhập kho Tài sản + Phiếu xuất kho Tiền.
 - **Redeem:** Tạo phiếu thu Tiền + Phiếu trả Tài sản.
 - **Liquidate:**
 - Tạo phiếu trả hàng ảo (Clear Stock Owner).
 - Tạo `sale.order` mới với các line là Trade-in.
 - Link `pawn.order` -> `sale.order`.
- **Trạng thái & Quyền hạn:**
 - Sử dụng `readonly` attributes ở View layer để khóa dữ liệu khi phiếu đã tắt toán (chỉ mở `note`).

6. Logic Tính Công Nợ (Debt Calculation)

Mô hình: Tiền là Hàng hóa (`product.product`). Nguyên tắc:

- Công nợ được tính dựa trên **Tổng giá trị các phiếu kho (Stock Moves)** đã hoàn thành (`state='done'`).
- **Phải thu (Debt Increases):** Giá trị hàng/tiền Shop chuyển cho Khách (Outgoing).
- **Phải trả (Debt Decreases):** Giá trị hàng/tiền Khách chuyển cho Shop (Incoming).

Công thức:

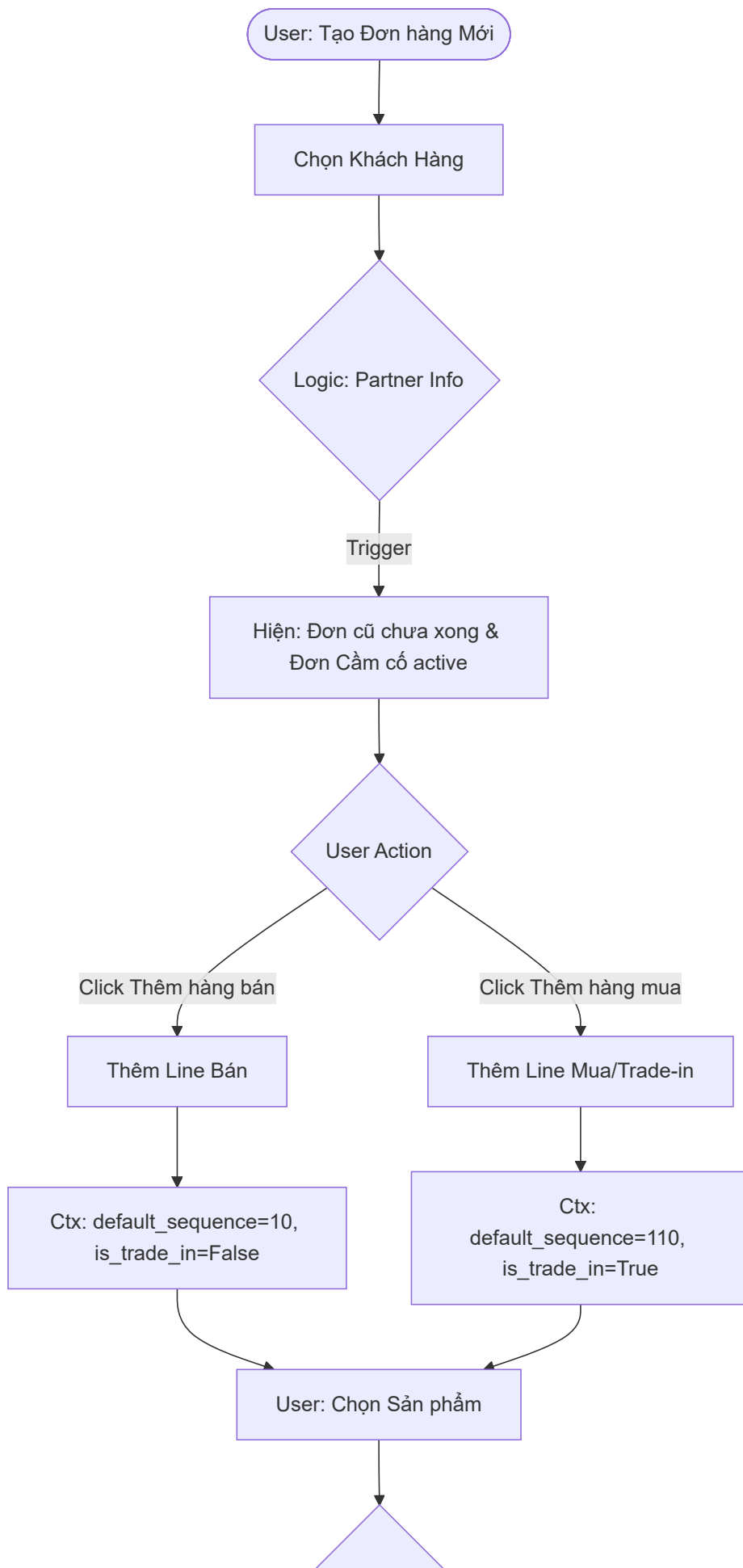
```
Net Debt = Sum(Qty * Price) [Outgoing to Customer] - Sum(Qty * Price) [Incoming from Customer]
```

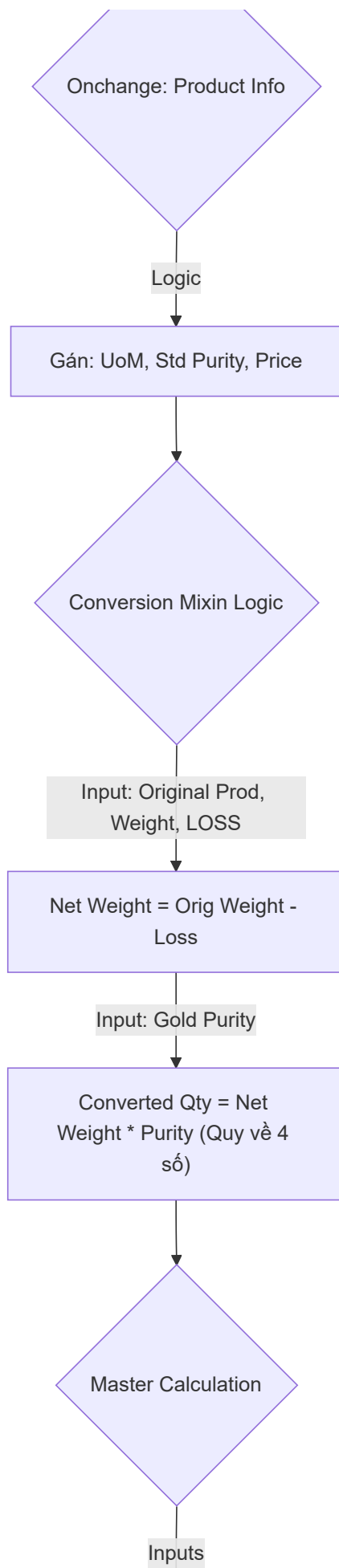
Bộ lọc quan trọng:

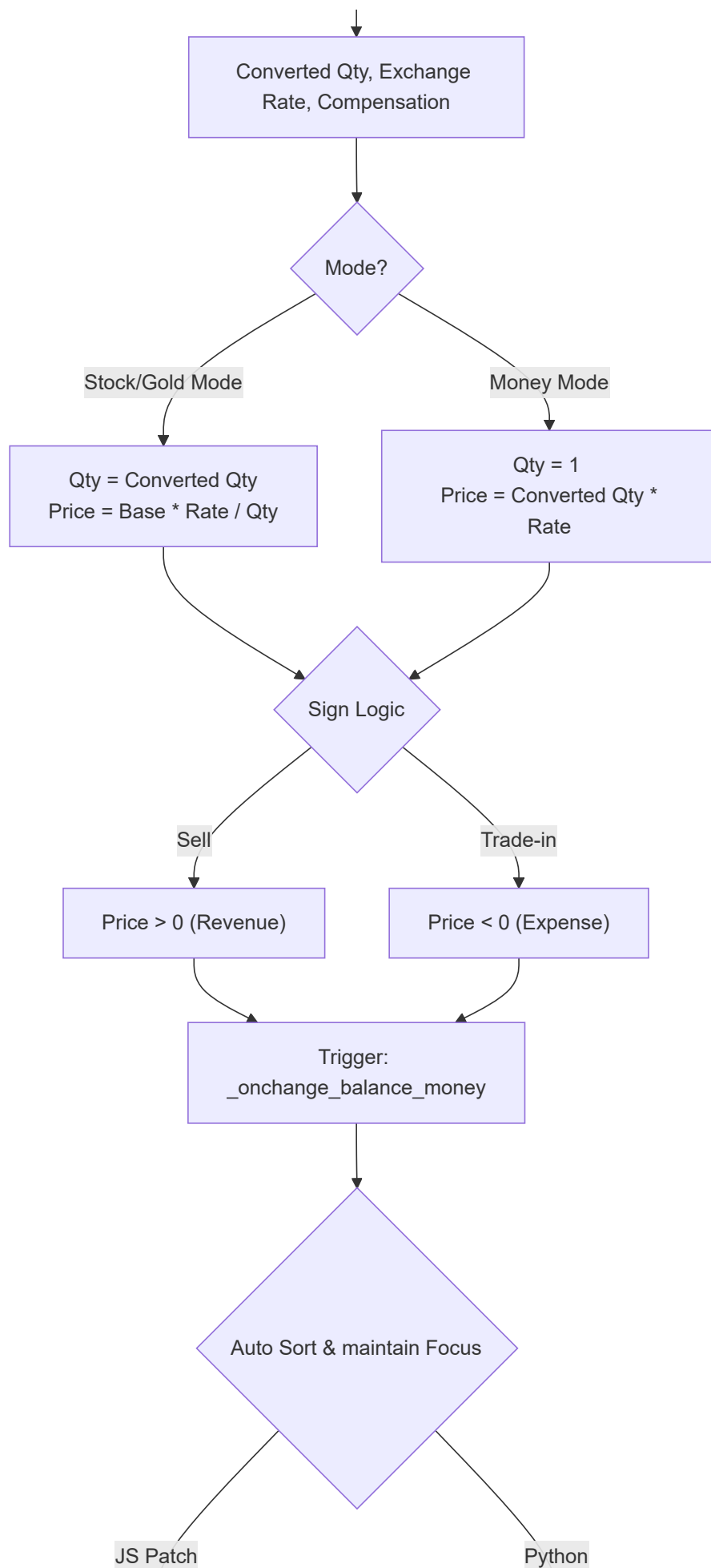
1. **Loại trừ Hàng gửi (Collateral):** Các phiếu kho có `picking_id.owner_id` được set (Hàng của khách gửi kho) **KHÔNG** tính vào công nợ.
 - Ví dụ: Khách cầm vàng (Pawn) -> Kho nhận vàng (Owner=Khách) -> Không thay đổi công nợ.
 - Shop đưa tiền (Loan) -> Kho xuất tiền (Owner=Shop) -> Tăng nợ.
2. **Giá trị (Price):** Ưu tiên sử dụng `price_unit_base` (nếu có logic trade-in) hoặc `price_unit` gốc của stock move.

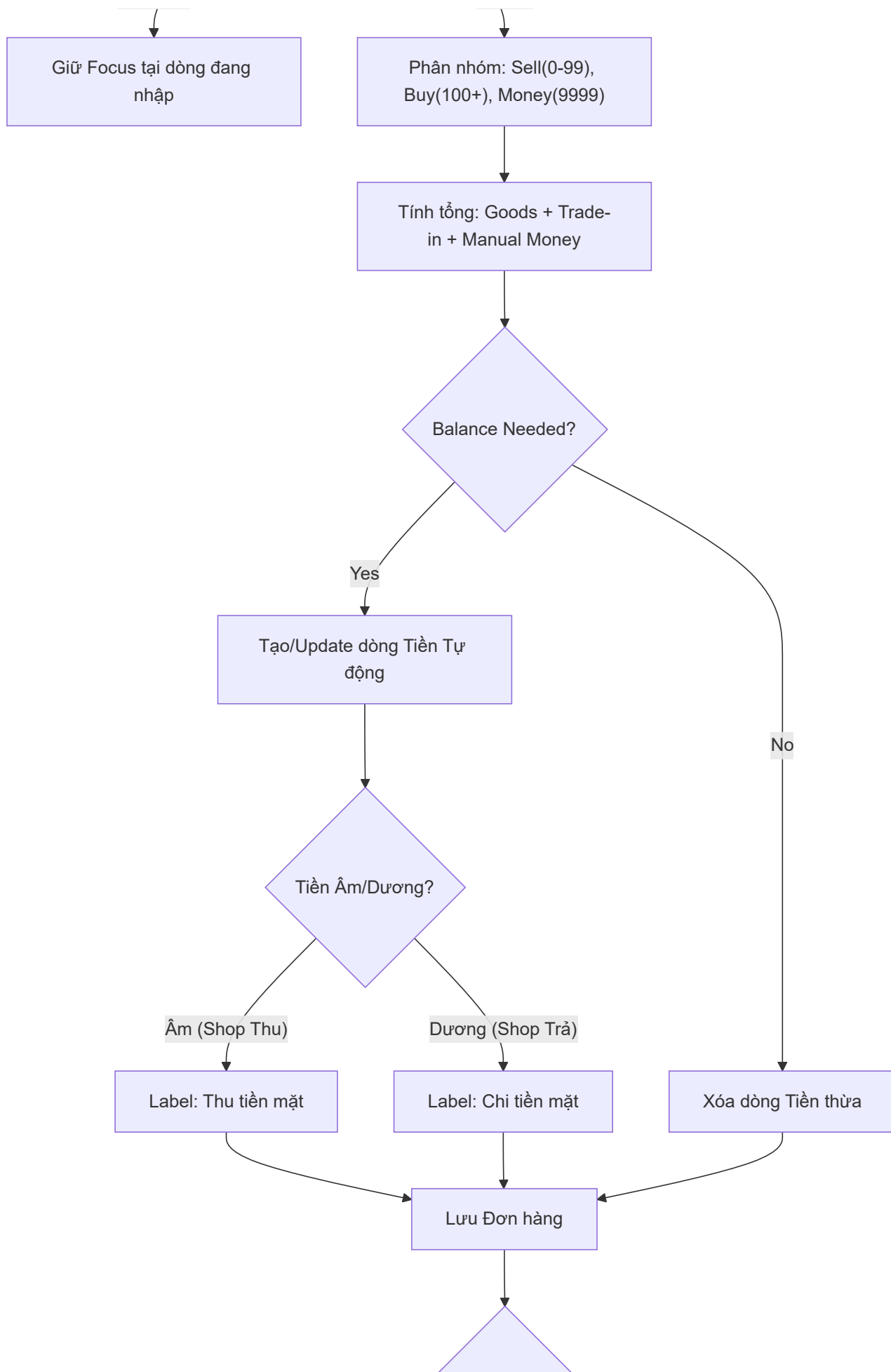
4. Quy trình Thiết lập Đơn hàng (Process Flow)

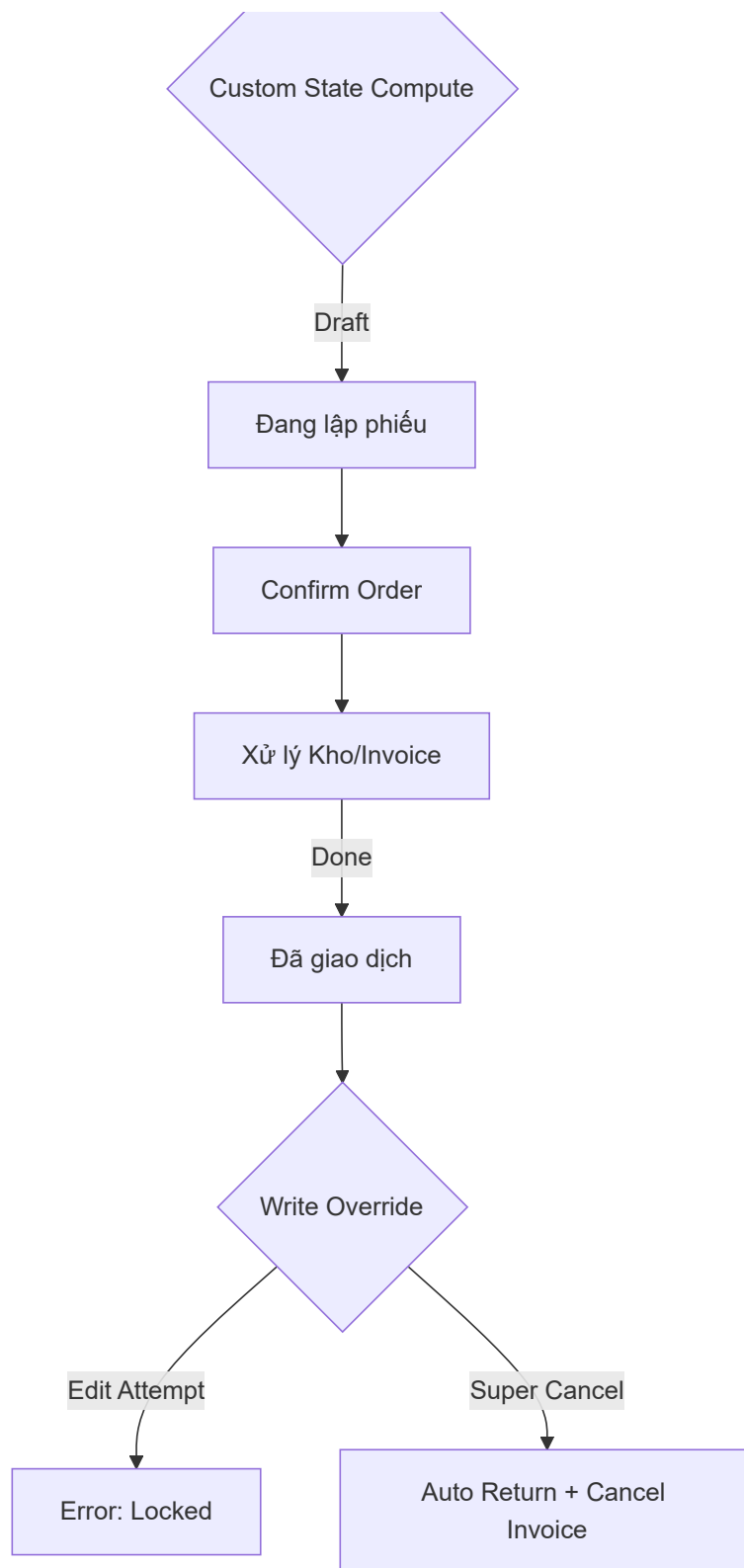
Biểu đồ dưới đây mô tả chi tiết luồng dữ liệu và logic khi thiết lập một đơn hàng trong hệ thống `qlv` (Dựa trên Implementation Plan V4 & Code Review).





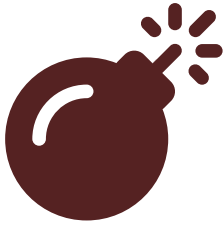






5. Quy trình Luân chuyển Hàng hóa (Inventory Flow)

Cơ chế phân tách phiếu kho tự động khi xác nhận đơn hàng (`action_confirm`).

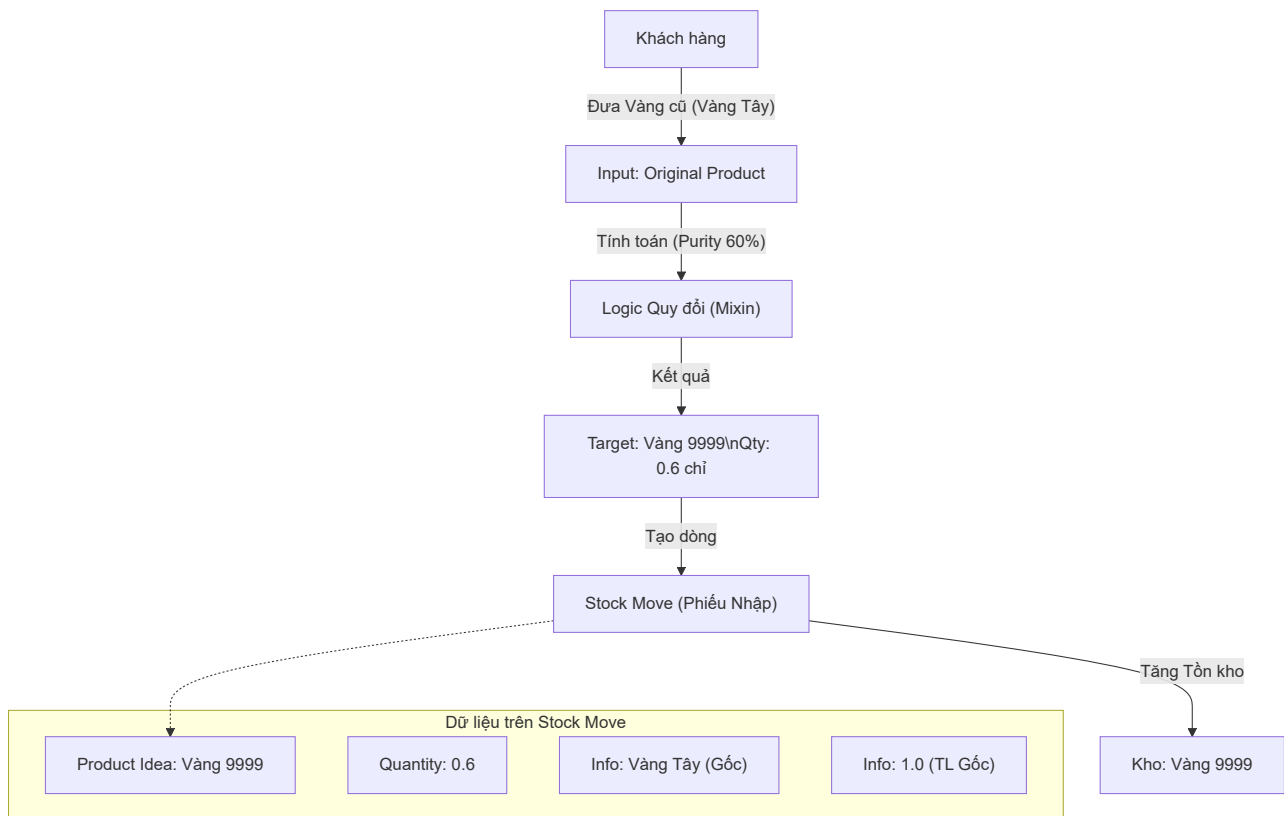


Syntax error in text

mermaid version 11.10.1

5.1. Cơ chế Quy đổi trong Kho (Stock Conversion Detail)

Với các đơn hàng có Quy đổi (Ví dụ: Khách bán vàng tây, quy về vàng 9999), hệ thống sẽ **ghi nhận nhập kho theo Sản phẩm Đích** (Target Product) nhưng vẫn lưu trữ thông tin gốc.



6. Cấu trúc Thư mục

```
e:\repo\odoo_docker\
├─ docker-compose.yml      # File cấu hình Docker
├─ custom_addons\          # Thư mục chứa module tùy chỉnh
│   └─ q1v\                # Module Quản lý Vàng (Core)
│       └─ configure_gold_shop.py # Script cấu hình ban đầu
├─ docs\                  # Tài liệu dự án
└─ README.md              # Hướng dẫn chính
```