

Finance and Inventory Linkage Test Walkthrough

Generated from the Gramia QA walkthrough. Use this checklist to validate the operational link between Finance procurement lines and Inventory stock movements.

Use this walkthrough to test the new linkage between Finance procurement lines and Inventory stock management.

The goal is to prove four things:

1. A normal finance item can still be saved without affecting stock.
2. A stock-linked supplier invoice item creates a stock **IN** movement.
3. A stock-linked purchase order item creates a stock **IN** movement.
4. Once a finance line has created stock, the linkage is locked against silent edits.

Scope

This test covers the new procurement-to-stock receiving flow only.

It does not yet cover full accounting entries for inventory asset reduction, COGS, or stock valuation journals.

Preflight Checks

From the Gramia project root:

Command / SQL check

```
cd /Library/WebServer/Documents/gramia
php -r 'require "public/autoload.php"; App\Services\FinanceModuleService::ensureTables(); echo "finance schema ok\n";'
```

Expected result:

Command / SQL check

```
finance schema ok
```

Confirm the new columns exist:

Command / SQL check

```
sqlite3 public/database/app.db "PRAGMA table_info(supplierInvoiceItem);"
sqlite3 public/database/app.db "PRAGMA table_info(purchaseOrderItem);"
```

Expected columns in both tables:

- stock_item
- stock_received_quantity
- stock_movement
- stock_received_date

Test Data Needed

Before testing, make sure the selected institution has:

- at least one active inventory item in Stock Management
- at least one approved supplier
- at least one supplier invoice, or enough data to create one
- at least one purchase order, or enough data to create one

Useful inventory check:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT id, name, institution  
FROM stock_item  
WHERE status=1  
ORDER BY institution, name  
LIMIT 20;  
"
```

Test 1: Supplier Invoice Item Without Stock Link

Purpose: prove normal expense items still work without changing inventory.

Steps:

1. Open Finance.
2. Go to Expenses.
3. Open or create a supplier invoice.
4. Add an expense item.
5. Fill normal fields:
 - item name
 - description
 - quantity
 - unit cost
 - total cost
6. Leave `Receive into inventory item` blank.

7. Save.

Expected result:

- item saves successfully
- supplier invoice amount refreshes
- no stock movement is created
- item has no `stock_movement`

SQL check:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT id, name, stock_item, stock_received_quantity, stock_movement, stock_received_date  
FROM supplierInvoiceItem  
ORDER BY id DESC  
LIMIT 5;  
"
```

Expected:

- latest non-stock item has `stock_item = 0`
- `stock_movement = 0`

Test 2: Supplier Invoice Item Creates Stock In

Purpose: prove a supplier invoice item can receive inventory stock.

Steps:

1. Open Finance.
2. Go to Expenses.
3. Open or create a supplier invoice.
4. Add an expense item.
5. Fill normal fields.
6. Select `Receive into inventory item`.
7. Enter `Inventory quantity received`, or leave it as `0` to use the item quantity.
8. Save.

Expected result:

- item saves successfully
- a stock `IN` movement is created
- the movement ID is written back to `supplierInvoiceItem.stock_movement`

- `stock_received_quantity` is filled with the received quantity
- `stock_received_date` is filled

SQL check:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT sii.id,  
       sii.name,  
       sii.quantity,  
       sii.unit_cost,  
       sii.stock_item,  
       sii.stock_received_quantity,  
       sii.stock_movement,  
       sii.stock_received_date,  
       sm.movement_type,  
       sm.quantity AS movement_qty,  
       sm.unit_cost AS movement_unit_cost,  
       sm.reference,  
       sm.notes  
FROM supplierInvoiceItem sii  
LEFT JOIN stock_movement sm ON sm.id=sii.stock_movement  
ORDER BY sii.id DESC  
LIMIT 5;  
"
```

Expected:

- `stock_movement` is greater than 0
- `movement_type` = IN
- movement quantity equals `stock_received_quantity`, or line quantity if received quantity was left as 0
- movement unit cost equals supplier invoice item unit cost
- reference starts with `Supplier invoice #`

Test 3: Supplier Invoice Item Duplicate Guard

Purpose: prove saving again does not create another stock movement.

Steps:

1. Open the stock-linked supplier invoice item from Test 2.
2. Save again without changing stock item, quantity, unit cost, or status.

Expected result:

- no duplicate stock movement is created
- the item keeps the same `stock_movement`

SQL check:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT stock_movement, COUNT(*) AS item_count  
FROM supplierInvoiceItem  
WHERE stock_movement > 0  
GROUP BY stock_movement  
ORDER BY stock_movement DESC  
LIMIT 10;  
"
```

Expected:

- the same finance item still points to one movement
- no new movement appears for the same save

Test 4: Supplier Invoice Item Edit Lock

Purpose: prove linked stock cannot be silently changed after receipt.

Steps:

1. Open the stock-linked supplier invoice item from Test 2.
2. Try changing one of these fields:
 - inventory item
 - inventory quantity received
 - unit cost
 - status
3. Save.

Expected result:

- save is rejected
- message says the line has already created a stock movement and must be reversed or adjusted before changing stock fields
- the existing stock movement remains unchanged

Test 5: Purchase Order Item Without Stock Link

Purpose: prove PO items still work normally without inventory receiving.

Steps:

1. Open Finance.
2. Go to Purchase orders.
3. Open or create a purchase order.
4. Add a purchase order item.

5. Fill normal fields.
6. Leave Receive into inventory item blank.
7. Save.

Expected result:

- item saves successfully
- no stock movement is created
- item has no stock_movement

SQL check:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT id, item_name, stock_item, stock_received_quantity, stock_movement, stock_received_date  
FROM purchaseOrderItem  
ORDER BY id DESC  
LIMIT 5;  
"
```

Expected:

- latest non-stock PO item has stock_item = 0
- stock_movement = 0

Test 6: Purchase Order Item Creates Stock In

Purpose: prove a PO item can receive inventory stock.

Steps:

1. Open Finance.
2. Go to Purchase orders.
3. Open or create a purchase order.
4. Add a purchase order item.
5. Fill normal fields.
6. Select Receive into inventory item.
7. Enter Inventory quantity received, or leave it as 0 to use the item quantity.
8. Save.

Expected result:

- item saves successfully
- stock IN movement is created

- movement ID is written back to `purchaseOrderItem.stock_movement`

SQL check:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT poi.id,  
       poi.item_name,  
       poi.quantity,  
       poi.unit_price,  
       poi.stock_item,  
       poi.stock_received_quantity,  
       poi.stock_movement,  
       poi.stock_received_date,  
       sm.movement_type,  
       sm.quantity AS movement_qty,  
       sm.unit_cost AS movement_unit_cost,  
       sm.reference,  
       sm.notes  
FROM purchaseOrderItem poi  
LEFT JOIN stock_movement sm ON sm.id=poi.stock_movement  
ORDER BY poi.id DESC  
LIMIT 5;  
"
```

Expected:

- `stock_movement` is greater than 0
- `movement_type` = IN
- movement quantity equals `stock_received_quantity`, or line quantity if received quantity was left as 0
- movement unit cost equals PO item unit price
- reference starts with `Purchase order #`

Test 7: Inventory Balance Changes

Purpose: prove the stock item balance actually increased.

Before creating a stock-linked item, record the current quantity:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT stock_item,  
       SUM(CASE WHEN movement_type='IN' THEN quantity ELSE -quantity END) AS qty  
FROM stock_movement  
WHERE stock_item = REPLACE_WITH_STOCK_ITEM_ID  
  AND status=1  
GROUP BY stock_item;  
"
```

After saving the linked supplier invoice item or PO item, run the same query again.

Expected result:

- quantity increases by the received quantity

Test 8: Cross-Institution Protection

Purpose: prove users cannot receive into another institution's stock item.

Steps:

1. Find a stock item from another institution.
2. Attempt to submit a supplier invoice item or PO item using that stock item ID.

Expected result:

- save is rejected
- message says the selected inventory item was not found for this institution

Test 9: List and Detail Display

Purpose: prove operators can audit the linkage visually.

Steps:

1. Go to Expense items.
2. Confirm the linked line displays:
 - inventory item
 - quantity received
 - stock movement
3. Go to Purchase order items.
4. Confirm the linked line displays:
 - inventory item
 - quantity received
 - stock movement

Expected result:

- stock-linked lines show the inventory item and movement reference
- non-stock lines show blank inventory fields

Troubleshooting Queries

Latest supplier invoice stock receipts:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT sii.id, sii.name, sii.stock_item, sii.stock_movement, sm.movement_type, sm.quantity,  
       sm.reference  
FROM supplierInvoiceItem sii  
LEFT JOIN stock_movement sm ON sm.id=sii.stock_movement  
WHERE sii.stock_item > 0  
ORDER BY sii.id DESC  
LIMIT 20;  
"
```

Latest purchase order stock receipts:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT poi.id, poi.item_name, poi.stock_item, poi.stock_movement, sm.movement_type, sm.quantity,  
       sm.reference  
FROM purchaseOrderItem poi  
LEFT JOIN stock_movement sm ON sm.id=poi.stock_movement  
WHERE poi.stock_item > 0  
ORDER BY poi.id DESC  
LIMIT 20;  
"
```

Latest stock movements:

Command / SQL check

```
sqlite3 public/database/app.db "  
SELECT id, institution, stock_item, movement_type, quantity, unit_cost, total_cost, movement_date,  
       reference, notes  
FROM stock_movement  
ORDER BY id DESC  
LIMIT 20;  
"
```

Pass Criteria

The linkage passes if:

- non-stock finance items do not create stock movements
- stock-linked supplier invoice items create exactly one stock **IN** movement
- stock-linked purchase order items create exactly one stock **IN** movement
- finance item records store the created stock movement ID
- stock item balance increases by the received quantity
- already-received lines cannot silently change inventory item, received quantity, unit cost, or status

Known Follow-Up

This test confirms operational inventory receiving. The next accounting step is to add ledger-level inventory asset and COGS postings.