# Recapped of Project Work Meeting on 2 Mar 2019 (ISY5001)

**Machine Reasoning**

**Venue:** Medical Library at NUS (10am to 7:30pm)

**Persons involved:**

Chad Ng, Sun Hang, Jin Xing, Dong Bin,Li Xin

## Points Covered:

**1** We started on trying to get everyone on the portal prototype so that we can all have a good view of the DoReMi order management system. We faced some obstacles deploying the ASP.NET Core solution. Some of us tried different ways to resolve this. At last, Sun Hang managed to get it deployed it into Microsoft Azure hosting environment as an APP Service. Sun Hang also created a database for the DoReMe and all of us were able to view the first prototype on Azure. All of us learned how this was done. It was a good practice session.

**2** We brainstorm quite a bit on the Sales order work flow. Jin Xin, Dong Bin and Li Xin contributed on ideals on sales order status setup for various workflow conditions. Sun Hang and Chad also gave thoughts on the suggestions and we agreed on the following:

**New Sales Order Status Types:**
Draft
Confirmed
Approved
Cancelled
Packing
Shipping
Closed

**Below is the explanation without the discount and forecasting Rules** (Will be covered separately)**:**

The rationale for this changes is the need to conform as much as possible to the required workflow of DoReMe.
When a sales person starts a new sales order, the first status is "**Draft**". Once Sales and Customer service finalised all discussions with customer, the sales order is updated by the sales person to "**Confirmed**". But, the product's stock qty should not be deducted yet as the order is not yet approved.

At this point, the system's Rules engine is fired and auto-approval process is handled. The sales order's status is changed to "**Approved**" if the order is good to go. **(A)** The systems writes a transaction record into **InventoryMovemnent** table and deduct the Product table's stock qty for each product in the sales order. *(See next page for **InventoryMovemnent** table**)***
However, if the Rules engine determines the order needs to be reviewed by the Accounts department, an email notification would be sent to Accounts department.
*(Sun Hang will check if he can put up the web based notification. For users to see the notifications when they signs in. - Good to have only, But the email notification is compulsory. Use smtp2go.com. It is very simple to setup.) If a web based notice is used, we will need a new table to store the notifications.*
Accounts Rep will sign in to system and check the customer's credit info (Is it a Gold member, Silver member or premium member?) their membership determines their credit facility or limits – depends on how we set it up. Accounts Rep then decide base on their credit risks and makes the update on the system. If credit risk is high, the sales order would be cancelled. (Selects the "**Cancelled**" status and update the system. Maybe a cancellation remarks can be added??) The system will send out an email notification to customer service to notify the customer service handling this account. If the credit risk is low, the sales order status would be updated by the Accounts Rep to "**Approved**"
*Execute* **(A)** *see above.*

Whenever, any sales order is **approved**, an email is sent to the warehouse department personal and the sales order status is changed to "**Packing**". Warehouse team then proceed to handle the physical product stock out from the system. When the product required in prepared and packed for shipping, the warehouse team then updates the sales order status to "**shipping**".
The last status "**Closed**" for each sales order is updated when a customer upon receipt of goods, triggers an API updates back to the system.

**Free SMTP Email Server:** https://www.smtp2go.com/

3 Dong Bin suggested we should consider a Order Review UI – This can be based on the same UI as Sales order. But the records shown would be filtered to show only sales order that have a status of "Confirmed" Our Assumption would be that any Accounts Rep may see any confirmed sales orders.

4 *Chad suggested that we need to add the field columns StockInQty, StockOutQty, BalanceQty to the **Products master table**.* After considerations, Chad decided this would not be wise or a good design choice. It is better we create a OrderTransactionLog table instead. I will explain this clearly in the next few paragraphs. *Sun Hang and Jin Xin, please let me know if this would be an issue to implement.*

These stock movement information is important because it would ensure correct stock balance for each product and this is required for tracking product demand forecasting for stock re-order triggering. This would be setup in NRules Engine.

**Assumption: We will handle in the first phase of implementing DoReMe's the system for 1 branch in one of the cities in US.**
**We need to keep it this way as the stock transfer module is out of scope.**
Also, we are assuming that the physical warehouse is highly automated. When sales orders are approved the warehouse backend automated delivery systems will auto-pick SKUs and ready the products for packing. No manual picking would be required.

**Recommended Table:**
**OrderTransactionLog**

**Database column names:**

| Nr. | Fieldname | Example |
|---|---|---|
| 1 | AutoNumber | 1 (This is a auto sequential numbering data type) |
| 2 | TransactionDate | Eg. 11-Mar-2019 08:00 |
| 3 | ProductCode | |
| 4 | ActionType | Eg. "Stock In" , "Stock Out" or "Cancelled", "Approved" |
| 5 | Qty | 100 |
| 6 | ReferenceCode | Eg. "PO1200329" (This is the publisher's PO) |
| 7 | ReferenceType | "PO" |
| 8 | CustomerCode | "Customer_K" |

**Note:** The ReferenceType can be either "PO", "SO" or "TO".
PO = Purchase Order
SO = Sales Order
TO = Stock transfer

**Let's consider an example case:**
**Inital Product master Table's SKU: A1234 has a stock Qty = 200**

***See Next Page***

Example transaction records:

| AutoNumber | TransactionDate | ProductCode | StockActionType | Qty | ReferenceCode | ReferenceType | CustomerCode |
|---|---|---|---|---|---|---|---|
| 1 | 12-Mar-2019 08:00 | A1234 | Stock In | 300 | PO1200329 | PO | |
| 2 | 13-Mar-2019 13:00 | A1234 | Stock Out | 100 | SO1230848 | SO | Customer_A |

Whenever, there is a “stock In”, it can be one of these cases :

(1) Triggered by a standard Rule to replenish a product stock qty, [Jin Xin, please set up some standard rules to do this, eg. Stock balance must not fall below the minimum allowed qty for all active products.)

(2) Triggered by a inventory the demand forecast Rules (Eg. Triggered for reordering products if product stock qty is below expected stock qty. The expected product stock qty would be 120% or the average qty ordered over the last 3 months for that product. The Minimum allowed qty is 10% of the average qty ordered over the last 3 months for that product.

**A1234 initial Qty = 200**
After **(1)** stock in, Qty = 200 + 300 = 500,
Then **(2)** Stock Out, Qty = 500 – 100 = 400, (100 was auto-approved for Cutomer_A's order)

**Scenario A**
**Current balance of product A1234 = 400.**
If Customer_B place an order on 13-Mar-2019 14:00 for 300 pcs of product A1234 with Sales. (order status: “DRAFT”)
If Customer_C place an orders on 13-Mar-2019 14:10 for 200 pcs product A1234 with Sales. (order status: “DRAFT”)

If both orders are confirmed at the about same timing, we have no issues as these actions will still be sequential handled by the Rules engine. If Customer_B's order is ran first, and it gets auto-approved, the system will deduct 300 from the current stock qty of 400. **The final product A1234 stock qty = 100.**
Then, Customer_C's order should not get approved as the current stock qty (100 pcs) of product A1234 is below the ordered qty of 200. This should show a proper message to the Sales person using the system. (Eg. Insufficient Stock Qty)

**Scenario B**
**Current balance of product A1234 = 400.**
If Customer_B place an order on 13-Mar-2019 14:00 for 400 pcs of product A1234 with Sales. (order status: “DRAFT”)
If Customer_C place an orders on 13-Mar-2019 14:10 for 500 pcs product A1234 with Sales. (order status: “DRAFT”)

However, if assuming that both orders cannot be process via the auto-approval process (ordered qty out of auto-approval limit), then both orders would be handled over to Accounts to review manually.

Assuming the Accounts team decides to approve both orders manually about the same time, we will assume that the system will process the order againts the rules engine, the result depends on which is processed first.

If Customer_B's order is processed first, the order gets processed and the final stock qty = 0. **Customer_C's order cannot be processed and the approval needs to be reversed.** Insufficient stock balance**.**

If Customer_C's order is processed first, the order will be rejected, ordered 500 > Balance 400.
But, Customer_B's order would still be processed successfully since its ordered qty is 400.

The case study did not specify how we should handle such a case.
We just need to take it as it is.

So, you can see that many different scenarios can create complex processing. I think we should better record the transactions for approved orders in my recommendation for an **OrderTransactionLog** table. This also fulfill the tracability requirements.

Also, please consider to log all approved or cancelled orders in **OrderTransactionLog** table.

**Note**: Need to make some assumptions for auto-approval limits. By product or by customer???

**6** Sun Hang: A global discount table is required with fields - start date and end date, discount percentage. Discount is applied at sales order level (Total amount). Sun Hang has makde changes to the order mgt web's naming conventings. Both Jin Xin and Sun Hang agreed to published the solution once a day. (TBA)

**7** Jin Xin - Forcasting, 3 mths projection, maybe create 2 or 3 type rules by products.
Jin Xin to prepare Rules in pseudo code for Dong Bin to add into project report.
Dong Bin suggested to set up based on demand discount rule. He will provide some inputs on this to Jin Xin.

**8** Chad mentioned for tracking of ordering, this is phase 2 and how it is done is explained in th project the report.
However, please see Item 4. Maybe, we may have to have some tracking by the order transaction logs

**9** Li Xin to prepare Book products and music score and publisher in excel format.
Li Xin to help with revised DoReMe's Flow Chart

**10** Chad to prepare process diagram for new system

**11** Dashboard – Initially, Chad suggest to research it. But, after looking at the source code, thinks that Sun Hang can easily handle this once the code system is in place.

**12** Chad to set up Zoom call on 5 Mar to touch base on action items. **Chad will also touch based with Sun Hang and Jin Xin on end Sunday or end Monday on item 4**.

**13** By 6 Mar, the system's UI should be ready, Chad can start preparing user guide starting 6 or 7 March.

**Other Notes**

**Added by Jin Xin:**
1. When closing order, top up 0.01*order amount to the user account

2. Every day run a windows task, which will compare balance and the average monthly sales of the last 3 months,
if balance<average monthly sales
then auto trigger a PO

3. When confirming order, if the confirming DateTime is in between the discount period. then apply the overall discount to the total amount

4. When the order amount < 1000 AND Order status==Confirmed then auto approve the order, set the order status=Approved

5. Do a windows app to simulate the logistics tracking system.