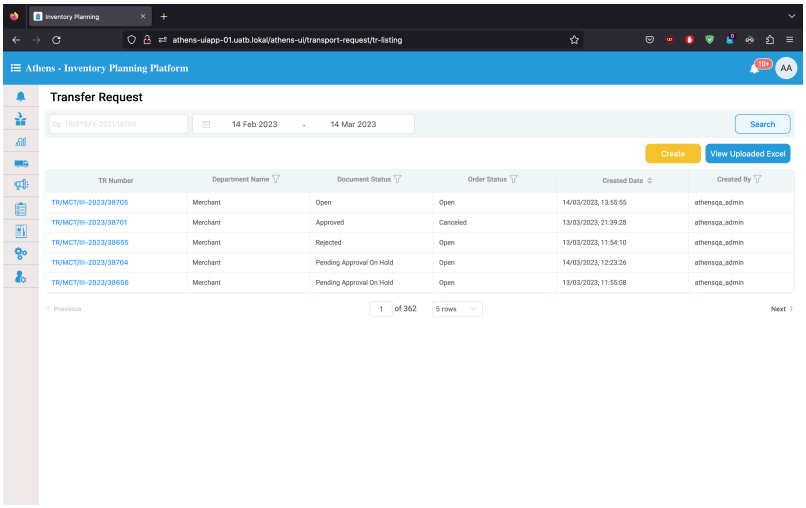
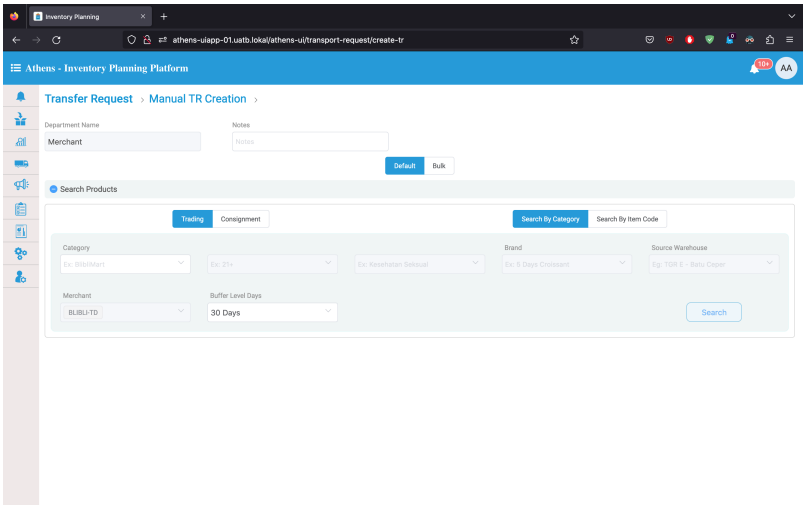
Transfer Request Flow

First Screen of TR Flow



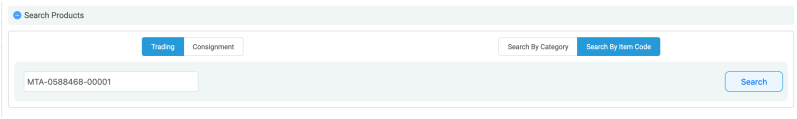
- Multiple APIs are called here to populate the filter menus i.e. department name, document status, order status etc.
- The TRs' are fetched based on the date, default is a months' duration from the current date.
- We can also filter based on TR number.
- Department Name is fetched from `CommonDataCommandController` (CDCC) whereas Document Status and Order Status are fetched from the `TransportRequestCommandController (TRCC)`
- The TR details are fetched from TRCC.
- Pagination is done for the TR details.

Create TR Flow



- The user can either upload an excel sheet (Bulk TR creation) or create one or more TRs manually (Default)
- User can search for products based on (selected in sequential order):
    1. Category 1
    2. Category 2
    3. Category 3 (Optional)
    4. Brand(Optional)
    5. Source Warehouse
    6. Buffer Days

- The filter menu options are populated through API calls after successive selection of categories 1 & 2.
- User can also search for a product through its item code.



- After selecting all the required filters / entering the item code , the search button is enabled, hitting the button now would hit an API in the `ProductCommandController`
- The same API is called in *search by category* and *search by item code* :

search products api

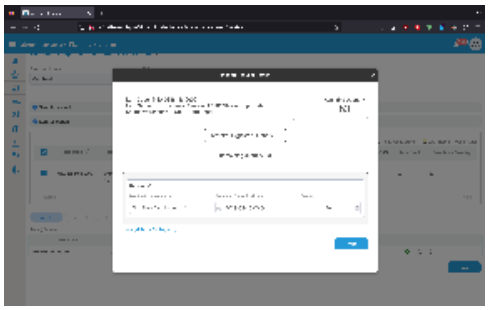
```
    public Mono<Response<List<TransportRequestProductResponse>>> transportRequestSearchProducts(
        @RequestBody TransportRequestSearchProductsRequest request,
        @RequestParam(value = "page", defaultValue = AthensCoreConstant.DEFAULT_PAGE_NUMBER) Integer
page,
        @RequestParam(value = "size", defaultValue = AthensCoreConstant.DEFAULT_PAGE_SIZE) Integer size,
        @RequestParam(value = "orderBy", required = false, defaultValue = StringUtils.EMPTY) String
orderBy,
        @RequestParam(value = "direction", required = false, defaultValue = StringUtils.EMPTY) String
direction)
```

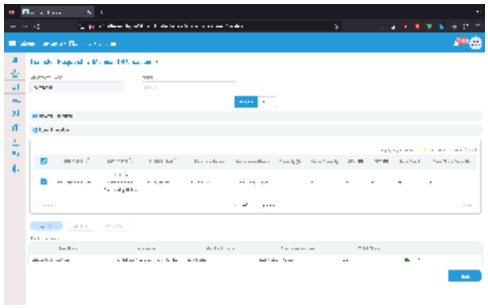
- The request body is a `TransportRequestSearchProductsRequest` object which has a boolean field, `searchByCategory`

request body

```
{
  "brands": [],
  "category1": [],
  "category2": [],
  "category3": [],
  "warehouseCodes": [],
  "merchantCodes": [],

  "itemCodes": [],

  "includeZeroQOH": ,

  "searchByCategory": ,

  "filters": [],
  "stockFilter": {
    "stockLabel": "30 Days",
    "stockStatus": null
  }
}
```

- This field is used to identify whether it is search by category of item code, if it is by item code, then the data is filtered via the `itemCodes` array.
- The user can then select the item(s), destination warehouse, estimated date of delivery and quantity, all of which are mandatory fields, and create a TR for them.
- Multiple TRs can be created for a single item with varying fields except for the item itself.





- The user can hit the save button to save a draft of their TR.
- The save API inside TRCC is called, all the validations are done and then the TR is saved into DB.
- The quota is also verified and reserved after successful verification.
- The Trade Partner (TP) user can send the TR for review or choose to send it later, they can also choose to create an Alert when sending for review.

Review Flow

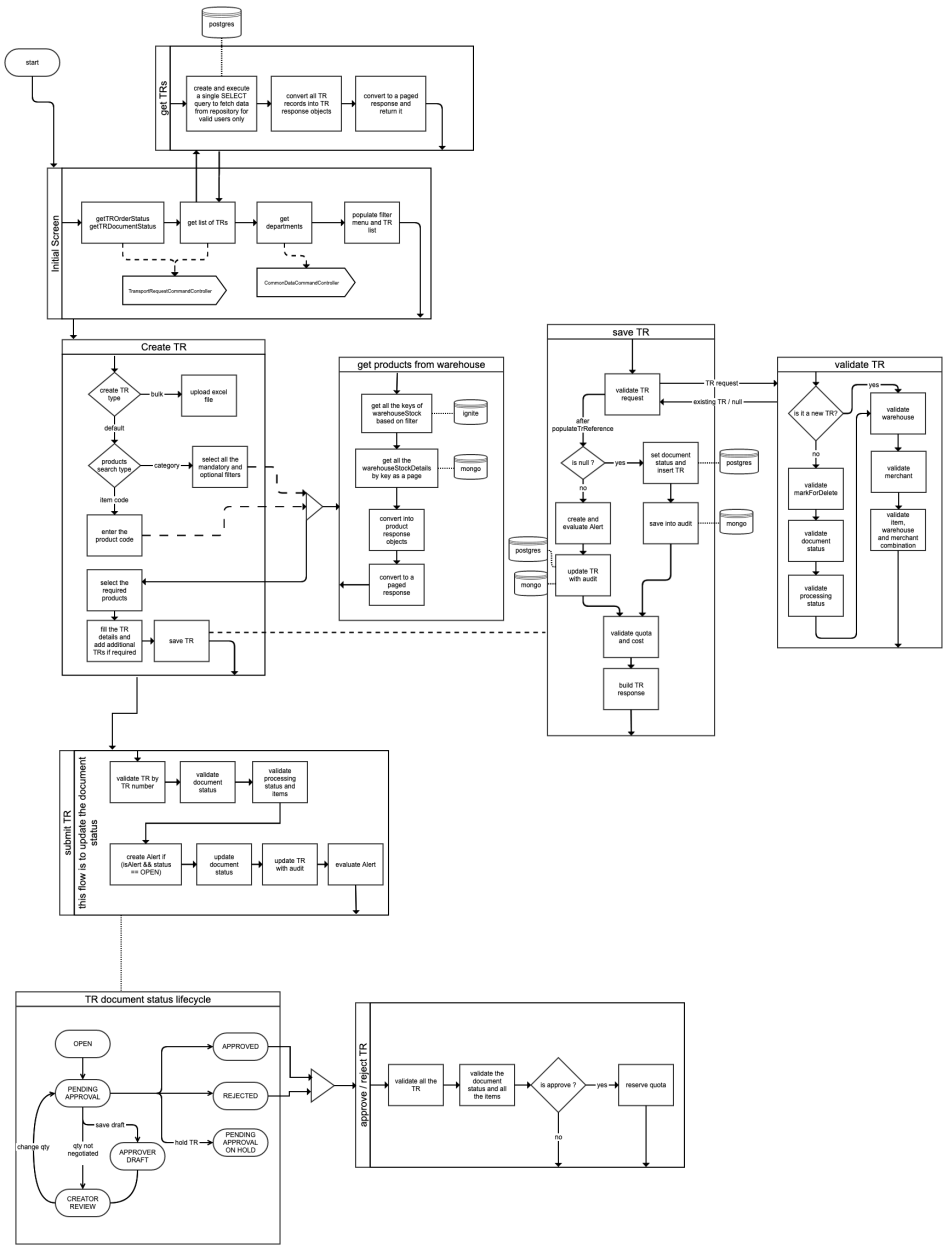
TR Number	Department Name	Document Status	Order Status	Created Date	Created By
TR/MCT/III-2023/38707	Merchant	Open	Open	14/03/2023, 15:49:41	athensqa_admin

1. A TR document can have 6 statuses

document status enum

```
public enum TRDocumentStatus {
  OPEN("Open"),
  APPROVER_DRAFT("Approver Draft"),
  CREATOR_REVIEW("Creator Review"),
  PENDING_APPROVAL("Pending Approval"),
  APPROVED("Approved"),
  REJECTED("Rejected");
.
.
.
```

2. The initial status after creating a TR and saving it would be the "Open" status.
3. After sending it for review, the status changes to "Pending approval"
4. The Approver can then choose to approve, reject or save a draft of the TR which puts the document in "Approver Draft" status.
5. If there are any changes in quantity, the approver can change it and send it back to the creator which put the document in "Creator Review" status.
6. If the quantities of approver and creator do not match it creates an infinite loop of steps 4 & 5.
7. The TR can only be approved if both the creator and approver negotiate on the quantity.
8. After Approval or Rejected, the status of the document is "Approved" or "Rejected" respectively.
9. There is another state in which a document can exist, i.e. the "Pending Approval on Hold" state, this is achieved when the Approver has received the TR and puts it on hold via their TaskManager.
10. After approval of the TR a Transport Order of the TR can be created.

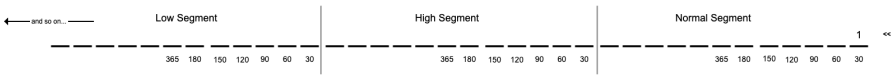


StockStatusValue

- Every WarehouseStockDetail document in mongoDB has a PeriodicDetail object that classifies the stock into high, normal, low, aging stock and out of stock for different durations i.e., 30 Days, 60 Days, 90, 120, 150, 180 and 365 days.
- High, Normal and Aging stock is called StockStatus and the 30, 60, 90 .... 365 days is called StockLabel
- While searching for WarehouseStock we also filter based on a Long value called stockStatus
- This value is calculated as a combition of both StockStatus and StockLabel

How?

- the stockStatus is variable of type Long, its size is 64bits
- we left shift the number based on the StockLabel i.e. 30, 60, 90 .... 365.
- we then left shift it based on the StockStatus (we multiply it with 12), in order to create segments
- the logic results in the creation of three Segments namely high, normal and aging, and each segment would have 12 bits to denote the StockLabel i.e. 30, 60, 90 365
- the first 7 bits would denote stock that has a StockStatus Normal Stock
- the maximum segment size is 12 bits


and so on...
Low Segment
High Segment
Normal Segment
1
<<
365 180 150 120 90 60 30
365 180 150 120 90 60 30
365 180 150 120 90 60 30