Stock Management and Validation Operations:

Validations stock:

ValiderTransfert:

It validates and processes a transfer operation in the inventory. It ensures the source and destination warehouses are specified, checks stock availability, updates stock records, and finalizes the transfer if all conditions are met.

Parameters:

- `$mouvement` (Object): The transfer movement object containing details of the transfer, such as source and destination warehouses.
- `$MouvementLs` (Array): A list of movement line objects detailing each item involved in the transfer.
- `$mouvement_id` (Integer): The unique identifier for the movement being processed.
- `ReglementServices $ReglementServices` (Object): An instance of the `ReglementServices` class, used for reglement processing related to the transfer.

```
 /**
 * Validates and processes a transfer operation in the inventory system.
 *
 * @param Object $mouvement The transfer movement object.
 * @param Array $MouvementLs A list of movement line objects detailing
each item in the transfer.
 * @param Integer $mouvement_id The unique identifier for the movement.
 * @param ReglementServices $ReglementServices An instance of the
ReglementServices class for additional processing.
 * @return Array An array containing the success status and message of the
operation.
 */
 public function ValiderTransfert($mouvement, $MouvementLs, $mouvement_id,
ReglementServices $ReglementServices)
    {...}
```

WorkFlow:

https://drive.google.com/file/d/1NiAIFgzTiPCwceQGFSmwmpNq9KrJ6Vye/view?usp=drive_link

ValiderRetour:

It processes and validates a return operation in an inventory. It ensures that the quantities being returned do not exceed the remaining quantities from the original delivery, updates stock records, and handles associated financial records.

Parameters:

- `$MouvementLs` (Array): A list of movement line objects detailing each item involved in the return.
- `$mouvement_id` (Integer): The unique identifier for the movement being processed.
- `$mouvement` (Object): The return movement object containing details of the return.
- `$operation`: Operation details .
- `$mlcheck` (Boolean): A flag indicating if all movement lines are processed successfully.
- `$now` (Object): The current date and time (usually a Carbon object).

```
/**
 * Validates and processes a return operation in the inventory system.
 *
 * @param Array $MouvementLs A list of movement line objects detailing
each item in the return.
 * @param Integer $mouvement_id The unique identifier for the movement.
 * @param Object $mouvement The return movement object.
 * @param Mixed $operation Operation details.
 * @param Boolean $mlcheck A flag indicating if all movement lines are
processed successfully.
 * @param Object $now The current date and time.
 * @return Array An array containing the success status and message of the
operation.
 */
   public function ValiderRetour($MouvementLs, $mouvement_id, $mouvement,
$operation, $mlcheck, $now)
    {...}
```

WorkFlow:

https://drive.google.com/file/d/1O8yza7yo7kbwfl273UnoaLshUH1kPMMI/view?usp=drive_link

transferToDepotPrincipale:

It handles the process of transferring items from a specific depot to the main depot within an inventory. It ensures that the transfer is valid, initializes the movement, updates stock records, and finalizes the transfer.

Parameters:

- `Request $request` (Object): The HTTP request object containing the depot ID for the transfer.

```
/**
 * Transfers items from a specified depot to the main depot.
 *
 * @param Request $request The HTTP request object containing the depot ID
for the transfer.
 * @return Array An array containing the success status and message of the
operation.
 */
    public function transferToDepotPrincipale(Request $request)
   {...}
```

WorkFlow:

https://drive.google.com/file/d/1OmP8DpPChEIOl3r2cpXoIHCS2HdXfU-X/view?usp=drive_link

ValiderRemplacement:

It handles the validation process for replacing products within a movement in case there was a mistake in adding the products in the movement. It ensures that the original and replacement products are available in stock and updates the stock records accordingly.

Parameters:

- `$MouvementLs` (Array): List of movement lines.
- `$mouvement_id` (Int): ID of the movement.
- `$mouvement` (Object): Movement object containing movement details.

- `$mlcheck` (Bool): Flag to check if all movement lines are successfully validated.
- `$now` (Carbon): Current timestamp.

```
    /**
     * ValiderRemplacement
     *
     * This function handles the validation process for replacing products
within a movement. It ensures
     * that the original and replacement products are available in stock
and updates the stock records accordingly.
     *
     * @param array $MouvementLs List of movement lines.
     * @param int $mouvement_id ID of the movement.
     * @param object $mouvement Movement object containing movement
details.
     * @param string $operation Operation type.
     * @param bool $mlcheck Flag to check if all movement lines are
successfully validated.
     * @param Carbon $now Current timestamp.
     * @return array $results Results of the validation process.
     */

    public function ValiderRemplacement($MouvementLs, $mouvement_id,
$mouvement, $operation, $mlcheck, $now)
    {...}
```

WorkFlow:
https://drive.google.com/file/d/1OFW61_1u5abQfbCC0JbSJSfcx-UOcboS/view?usp=drive_link

valdiationStock:

It is designed to handle the validation process for stock movements. This process includes various checks and updates to ensure that stock quantities are correctly validated and recorded within the system. Depending on the type of movement, the function performs specific validations and actions to maintain data integrity and reflect the accurate state of stock levels.

Parameters:

- **Request $request**: The incoming request object containing necessary parameters for the function.

- **ReglementServices $ReglementServices**: A service object that handles payment regulations and related operations.
- **int $mouvement_id**: The ID of the movement that needs to be validated.

```
   /**
     * valdiationStock
     *
     * This function handles the validation process for stock movements.
It ensures that the stock quantities
     * are properly validated and updated in the system. Depending on the
type of movement, it performs
     * various checks and validations to ensure data integrity.
     *
     * @param Request $request The incoming request containing necessary
parameters.
     * @param ReglementServices $ReglementServices The service handling
payment regulations.
     * @param int $mouvement_id The ID of the movement to be validated.
     * @return array $results Results of the validation process.
     */

      public function valdiationStock(Request $request, ReglementServices
$ReglementServices, $mouvement_id)
    {...}
```

WorkFlow:

Operations stock:

ValidationMagasinier:

It is responsible for validating stock movements related to a mission. This function ensures that all necessary checks and balances are performed before finalizing the validation of the stock movements handled by the storekeeper. It is typically called when a storekeeper needs to validate the stock movements related to a mission. This ensures that all stock movements are accurately recorded, any corrections are made, and the financial records are updated accordingly.

Parameters:

- **Request $request**: This parameter contains the incoming request data, including the mission ID and other relevant information required for the validation process.

```
    /**
     * Handle the validation and processing of the magasinier (warehouse
keeper) tasks for a given mission.
     *
     * This method performs various checks and operations related to the
validation of a mission in the warehouse
     * management system. It handles:
     * - Checking if the mission has been validated for multiple
beneficiaries.
     * - Ensuring a transfer exists before proceeding with a reverse
transfer.
     * - Validating if the mission has already been processed with a
global BL.
     * - Calculating and validating product quantities and prices.
     * - Creating and updating movement records and related items.
     * - Managing stock operations and adjustments.
     * - Handling potential errors and logging them.
     *
     * @param \Illuminate\Http\Request $request The request object
containing the mission ID and other parameters.
     * @return \Illuminate\Http\JsonResponse The response in JSON format
indicating the success or failure of the operation.
     */
    public function ValidationMagasinier(Request $request)
    {...}
```

WorkFlow:

https://drive.google.com/file/d/1O2erx6S8mJtraEXnGxtQnjrlijMX7EEw/view?usp=drive_link

ValidationMagasinierCorrection:

This function validates and corrects stock and delivery information related to a specific mission and shipment. It handles various tasks including processing product quantities, prices, promotions, and managing the creation and update of return and delivery records. This function

ensures that the stock and delivery records are accurately updated and that any discrepancies are resolved.

Parameters:

- **$mission** (object): Represents the mission associated with the delivery. It contains details such as mission ID and related information.
- **$chargement** (object): Represents the shipment or loading associated with the mission. It includes details like movement ID and other relevant information.
- **$now** (string): Represents the current date and time, formatted as a string. This is used for timestamping records and operations.

```
    /**
     * Validates and processes the corrections for a warehouse
(magasinier) based on the provided mission and loading (chargement)
details.
     *
     * This function performs the following tasks:
     * 1. Calculates the delivery declaration for the given mission.
     * 2. Deletes duplicated lines from the global bill of lading.
     * 3. Retrieves and processes the updated and original lines of the
loading.
     * 4. Checks the return quantities and updates or creates return lines
accordingly.
     * 5. Updates the delivery lines with any remaining quantities and
applies any relevant promotions.
     * 6. Validates and corrects returnable items and updates the mission
sales amount.
     * 7. Finalizes the stock operations and updates the status of the
relevant documents.
     *
     * @param object $mission The mission object containing details of the
mission to process.
     * @param object $chargement The loading (chargement) object related
to the correction.
     * @param string $now The current timestamp used for date-related
operations.
     *
     * @return \Illuminate\Http\JsonResponse A JSON response containing
the results of the validation and correction process.
     */
```

```
    public function ValidationMagasinierCorrection($mission, $chargement,
$now)
    {
```

WorkFlow:

https://drive.google.com/file/d/1OmFTPKKjowhXXCDpznD4X1J2ud176Vnt/view?usp=drive_link

setMouvemenTransfereAuDepot:

This function processes a reverse transfer for a given mission. It validates the transfer, creates necessary records for a return, updates related movements and stock, and handles any errors encountered during the process.

Parameters:

- **$request** (object): The HTTP request object containing the `mission_id` which identifies the mission for which the reverse transfer is being processed.

```
/**
     * Processes a reverse transfer for a given mission.
     *
     * This function performs the following actions:
     * 1. Validates that a morning shipment exists for the mission.
     * 2. Checks if a reverse transfer has already been made.
     * 3. Ensures the necessary BL Global records are validated.
     * 4. Creates and initializes a new return movement.
     * 5. Establishes relationships between the original and return
movements.
     * 6. Validates quantities and inserts reversed movement lines.
     * 7. Handles article movements and updates stock levels.
     * 8. Validates returnable items and manages remaining shipment lines.
     *
     * @param Request $request The HTTP request object containing the
`mission_id`.
     * @return \Illuminate\Http\JsonResponse Returns a JSON response
indicating success or failure.
     */
    public function setMouvemenTransfereAuDepot(Request $request)
    {...}
```

WorkFlow:

https://drive.google.com/file/d/1O46r-aAfdwi2imn12skapSae24ZUHoLx/view?usp=drive_link

transfertChargmentToMission:

This function transfers a shipment to a new mission and records the necessary details in the database.

Parameters:

`$request`: The HTTP request object. It should contain:

- `mission_id`: The ID of the mission to which the shipment is being transferred.
- `mode_id`: The mode of transfer, though it's not used in this function.

```
 *//**
     * Handles the transfer of a shipment to a mission.
     *
     * This function performs the following actions:
     * 1. Retrieves the mission details based on the provided
`mission_id`.
     * 2. Finds the most recent mission for the same transporter that
started before the current mission.
     * 3. Fetches the morning shipment related to the mission.
     * 4. Initializes a new transfer movement with the current mission
details.
     * 5. Creates a relationship between the original and new transfer
movements.
     * 6. Inserts transfer lines for each item with a return quantity.
     *
     * @param Request $request The HTTP request object containing the
following parameters:
     * - `mission_id`: The ID of the mission to which the shipment is
being transferred.
     * - `mode_id`: The mode of transfer (not used in the current
implementation).
     * @return \Illuminate\Http\JsonResponse Returns a JSON response
indicating success or failure.
     */

    public function transfertChargmentToMission(Request $request)
```

```
    {...}
```

WorkFlow:

https://drive.google.com/file/d/1OXIZlAOPAp0OIYnVzuK-75nVHLzevj4O/view?usp=drive_link

ValiderRetournables:

This function validates and processes returnable items by comparing reported quantities to actual returns. It updates movements, creates financial records, and adjusts stock details to ensure accurate handling of returns and associated costs. The function handles both positive returns (where returned quantities exceed reported quantities) and negative returns (where reported quantities exceed actual returns), ensuring proper financial and stock adjustments.

Parameters:

- `$mission`: An object representing the mission associated with the returnable items.
- `$chargement`: An object representing the shipment or loading related to the mission.

```
/**
     * Validate and process returnable items for a given mission.
     *
     * This function handles the validation of returnable items by
comparing the reported quantities
     * with the actual returned quantities. It performs the following
operations:
     *
     * - Initializes movement records for packaging returns and credits.
     * - Retrieves and processes return shipments related to the mission.
     * - Creates or updates movement lines based on discrepancies between
reported and actual returned quantities.
     * - Generates financial records (Reglement) for positive and negative
discrepancies.
     * - Updates depot returns and creates article movement records for
stock adjustments.
     *
     * Parameters:
     *
     * @param object $mission The mission object containing details about
the mission.
     * @param object $chargement The shipment or loading object related to
the mission.
```

```
     *
     * Returns:
     *
     * @return bool Returns true upon successful validation and processing
of returnable items.
     */
    public function ValiderRetournables($mission, $chargement)
    {...}
```

WorkFlow:

https://drive.google.com/file/d/1OCtmteBYAGJZ22lVRrPx3TBNfAY30uXv/view?usp=drive_link

ValiderRetournablesCorrection:

This function is used to ensure that discrepancies in returnable items are properly managed and recorded, maintaining accurate inventory and financial records.

Parameters:

- **`$mission`**: This is an object representing the mission for which returnable items are being processed. It typically contains data like the mission ID, transporter ID, and other relevant mission details.
- **`$chargement`**: This is an object representing the shipment or loading associated with the mission. It includes information such as the destination warehouse or depot for the returnable items.

```
  /**
     * Validate and process returnable items with corrections for a given
mission.
     *
     * This function manages the validation and correction of returnable
items, handling discrepancies
     * between reported and actual return quantities. It performs the
following operations:
     *
     * - Initializes movement records for packaging returns and credits if
not already present.
```

```
     * - Retrieves and processes return shipments related to the mission.
     * - Creates or updates movement lines based on discrepancies between
reported and actual returned quantities.
     * - Generates financial records (Reglement) for positive and negative
discrepancies, ensuring no duplicates.
     * - Updates depot returns and creates article movement records for
stock adjustments, avoiding duplicate entries.
     *
     * Parameters:
     *
     * @param object $mission The mission object containing details about
the mission.
     * @param object $chargement The shipment or loading object related to
the mission.
     *
     * Returns:
     *
     * @return bool Returns true upon successful validation and processing
of returnable items.
     */
    public function ValiderRetournablesCorrection($mission, $chargement)
    {...}
```

WorkFlow:
https://drive.google.com/file/d/1OXSp2clcAGh8TuzQZ79yEI0qOl0BBxi4/view?usp=drive_link

Overview of some attributes

Here are the explanations for some attribute and term in the context of stock and validation processes:

Changement Ligne (Loading Line)

- **article_quantite**: The quantity of the item being loaded.
- **article_quantite_delivred**: The quantity of the item delivered by the transporter.
- **article_quantite_reste**: The remaining quantity, calculated as the difference between the quantity loaded and the quantity delivered (`article_quantite - article_quantite_delivred`).

- **article_quantite_retour**: The quantity of the item returned, as declared by the storekeeper.

BL Ligne (Bon livraison Line)

- **article_quantite**: The total quantity sold.
- **article_quantite_delivred**: The total quantity delivered as per the order (i.e., the quantity listed in the BL).
- **article_quantite_reste**: The remaining quantity, calculated as the difference between the total quantity sold and the total quantity returned (`total vendue - total retourne`).

BL (Delivery Note)

- **is_stocked**: Indicates if the stock has been validated by the storekeeper on BLs that have already been validated by the seller.
- **livraison_is_delived**: Validation by the seller either via mobile or back office.

Promo Ligne (by loading line)

- **montant_remise_chargement**: The discount amount applied to the loading line.
- **montant_remise_declarer_par_le_vendeur**: The total discount amount declared by the seller, which is the sum of discounts for each product from BLs validated by the seller.
- **montant_remise_reste**: The remaining discount amount, calculated as the difference between the discount amount applied to the loading line and the declared discount amount (`montant_remise_chargement - montant_remise_declarer`).
- **justifier_montant_remise_reste_sur_bl_global_ou_retour_global**: Justification of the remaining discount amount on the global BL or global return.

Init Credit Draft in Validation Magasinier

- **montant_bl - montant_reglement_de_bl**: The credit draft amount, calculated as the difference between the total amount of the BL and the payment made against the BL.

Balance (Value of Products)

- **montant_chargement**: Total amount related to the loading.
- **montant_retour**: Total amount related to returns.
- **montant_remise**: Total amount of discounts.
- **montant_credit_mobile**: Total amount of mobile credits.

Statistics and reportings:

The system provides two distinct types of statistical data, each designed to meet the needs of different users:

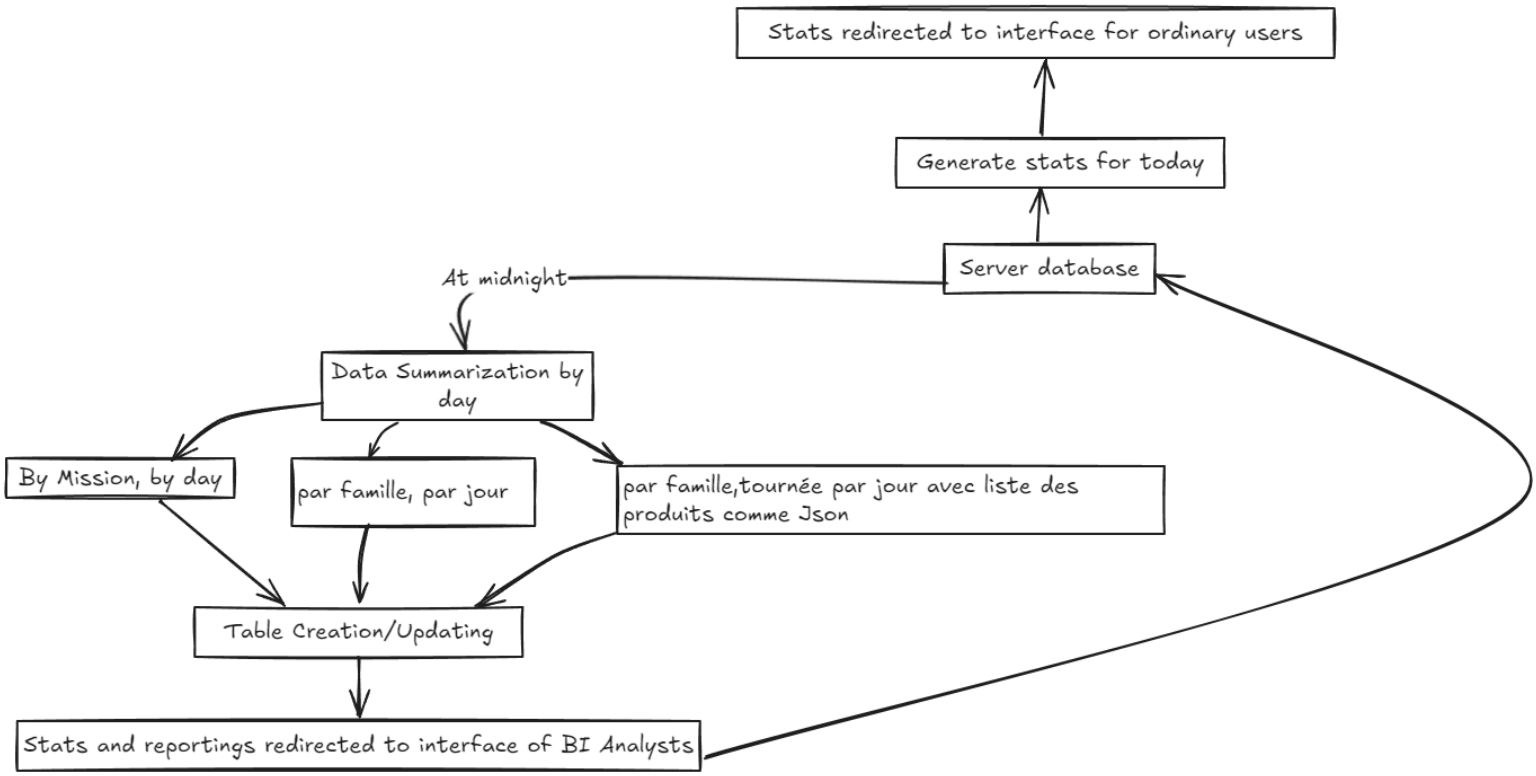
1. Regular Daily Stats:

These are real-time statistics reflecting the current day's activities. They are primarily used by regular users who need immediate insights into ongoing operations.

2. Summarized Daily Reports:

These reports offer a more in-depth analysis, summarizing data by key categories such as family, mission, and tournée. These summaries are compiled at midnight and are primarily used by Business Intelligence (BI) analysts.

By providing tailored interfaces for both regular users and BI analysts, the system ensures that each user group has access to the specific data and tools they need to perform their roles effectively.



Syncing data from mobile devices

To prevent duplicate IDs in mobile devices, the process is as follows:

1. The mobile device first reserves the required number of free, empty slots in the server database.
2. The server then compares the IDs on the mobile device with those in the database to identify any overlapping IDs.
3. Next, the server reorders the IDs and replaces the old mobile IDs with the newly assigned ones.
4. Finally, the mobile database is updated with this new set of IDs, ensuring consistency and avoiding duplication.