Use Case: Manage stock

Stock table contains the data about all what is getting sold.

Stock table can be updated by either fulfilled orders or sales

Use case description: Update Stock and Fulfill Orders

Actor: Admin

**Description**: To update the stock quantities in the **Stock Table** based on the details of a fulfilled order, ensuring the stock levels reflect the products ordered and their quantities.

Preconditions:

* The user (admin) is logged into the system and has appropriate permissions.
* An order has been created and is in the pending state
* The system has the capability to either automatically or manually update the stock quantities.
* The **Stock Table must not** contain initial stock quantities for the products involved in the order.

Postconditions:

* The **Stock Table** is updated with the correct quantities for each product based on the **Order Details**.
* If the **manual update option** is used, the **Stock Table** is updated only after confirmation from the user.
* If the **auto-update option** is selected, stock is updated immediately and changes the order status to **Fulfilled**.
* The **Order Status** is set to **Fulfilled** after stock update is completed.
* Orders are marked either as **Fulfilled** or **Pending Complete Update**, depending on the number of product items in the order are pushed to stock.

Steps:

1. The user navigates to the orders page and views a list of unfulfilled orders.
2. The user clicks on an unfulfilled order to view the list of products within that order.
3. Each product item has got a button to push the product item to stock, remove the product item from stock , edit the product item
4. If product is not already `pushed to stock` ,the `remove item` from stock button is not seen
5. When Admin `pushes product item to stock`, the `push to stock` button for that product is disabled and all others are enabled including the `remove from stock button`
6. The user has two options to update stock:
   * **Per Item Update**: The user clicks on each product item, updates the stock table with the details of the fulfilled item, and updates the order price if necessary.
   * **Batch Update**: The user clicks on an "Update Order" button to update the stock table with the details of all products in the order at once.

3.1 Batch update

* The admin clicks on an "Update Order" button to update the stock table with the details of all products in the order at once.
* A prompt appears asking the admin to confirm if to automatically update the batch stock
* The admin clicks **Yes** to confirm.
* The system then automatically updates the stock table, adjusting the product quantities based on the order details.
* The **Stock Table** is updated, and the product quantities are increased by the amounts ordered.
* The "Add to Stock" button for that product becomes unavailable.
* The user can no longer attempt to update the stock for already-added items.
* The **Order Status** is updated to **Fulfilled**

3.2 Manually update

* The admin clicks on each product item, clicks on the push to stock button to add the product to stock
* Once product has been added in stock, the push to stock button, edit product item, delete product from order are disabled. Only remove the product from stock button is enabled for x number of days

3.2.1 Admin edits

* If admin clicks on edit and edits the prices, the price discrepancy of the total products against the order cost should disable the ‘push to stock’ button. Only the `edit product` and `delete` product from order should be enabled
* That should be resulted once he edits the order prices

3.2.2 Admin deletes product item from order

* If Admin deletes the product item from order, then the system should recalibrate the prices of the remaining product in that order.
* Only the `delete product from order` and `edit product in order` button should be enabled until the admin updates the order total price to fit the sum total of all the prices under that order.

* If Admin updates only some of the products are (e.g., one product added but others are left unfulfilled), the system changes the order status to **Pending Complete Update**.
* The admin should receive a system a reminder or notification about the remaining unfulfilled items, indicating that the order is not fully updated
* If all products in the order are successfully updated in the stock table, the system marks the order as **Fulfilled**.

#### 

#### Extensions:

* **Stock Update Reminder**: If the Admin tries to mark the order as fulfilled while some items are missing from the stock update, the system prompts a message reminding the Admin to update the stock for the remaining items.
* **Order Price Update**: If the user updates or deletes any product item, the system presents a menu with two options: either re-edit the order prices manually or have the system automatically adjust them.
* **Single Item Left**: If the order initially contains more than one item and the user deletes all but one, the system still marks the order as **Fulfilled** if the remaining item is added to stock. This is done after the second last Product item has been deleted

**Alternative Flows:**

* **Partial Fulfillment Option**: The user chooses to fulfill and update stock for only a portion of the order (per item update), and the system changes the order status to **Pending Complete Update** rather than **Fulfilled**.

## Data Structure

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Element name | Description | Length | Base  /Derived | Default value | Validation criteria | | |
|  |  |  |  |  | Upper limit | Lower limit | comments | |
| orderDetailID | Id of the orderDetail | 10 | Derived | Not null | Char=50 | Char >20 |  | |
| ProductSku | Unit sku for each product | derived | Derived | Not null | Derived | derived |  | |
| productName | Name of the product (e.g., "Hard Red Winter Wheat"). |  |  |  |  |  |  | |
| QuantityOfProductItems | In numeral form eg 10 | Base | Base | Not null | int<10000 | int>0 |  | |
| ReorderLeveL | Minimum stock level before triggering a reorder. |  |  |  |  |  |  | |
| Units | Eg kgs | Derived | derived | Not null | derived | derived |  | |
| priceAtTimeOfOrder | Total price of that product item | >0 | Base | Not null | >0 | Float 3 |  | |
| dateAdded | Date order details is added | Date | Base | Not null | Date | Date |  | |
| dateLastUpdated | Date order is updated | Date | Base | Not null | Date | Date |  | |