Solutions I was not able to implement

You know the robot buyer knows the car models he is able to buy, we want to optimize the sales, as we have lag between the factory stock and the store stock, it may happen that we don’t have stock at the store stock but we actually have brand new cars ready to be sold in the factory stock. How can we optimize the stock management? (sadly we can’t centralize the stock)

Proposed Solution:

My proposed solution for this problem would be to create a script that looks up Factory stock in the background everytime an order is placed and tries to match order with the factory stock.

After you finish to code this challenge, imagine you'll receive a text message from the robot buyer, it says that several orders need to be changed because they want to change the cars models. First, you receive an order\_id and the car model, but an hour later you'll start to receive several changes of this kind request per hour. What is your proposal to solve this need? Also please implement the solution.

My proposed solution to this problem would be to run a script that receives these changes from the robot\_buyer and modifies the orders with the required model numbers,updating the store stock according to the old and new model numbers before updating the order table in the database,thereby ensuring an order is only modified when there is existing stock for the new model change requested.