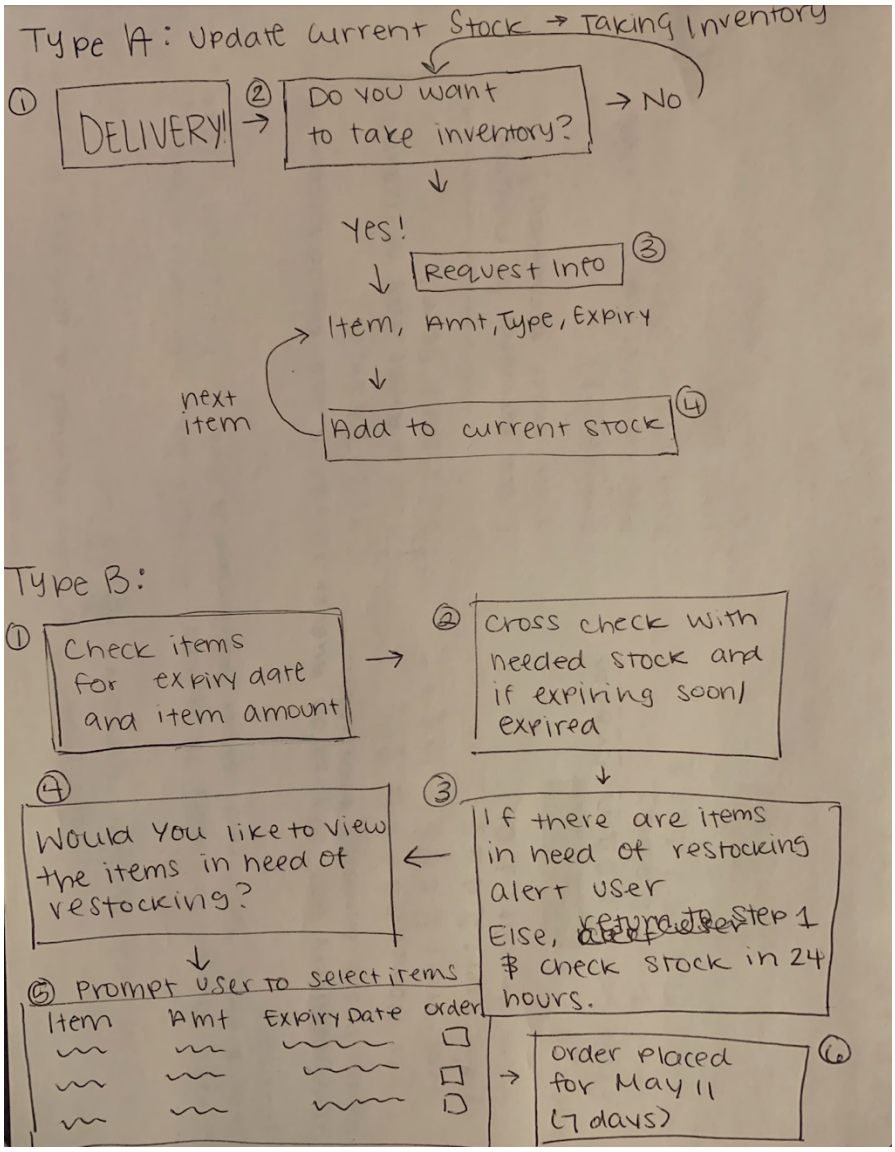
Recitation 9: Paper Prototyping
Preston Dotsey & Pourna Sengupta
CSCI 3002 – Section 106

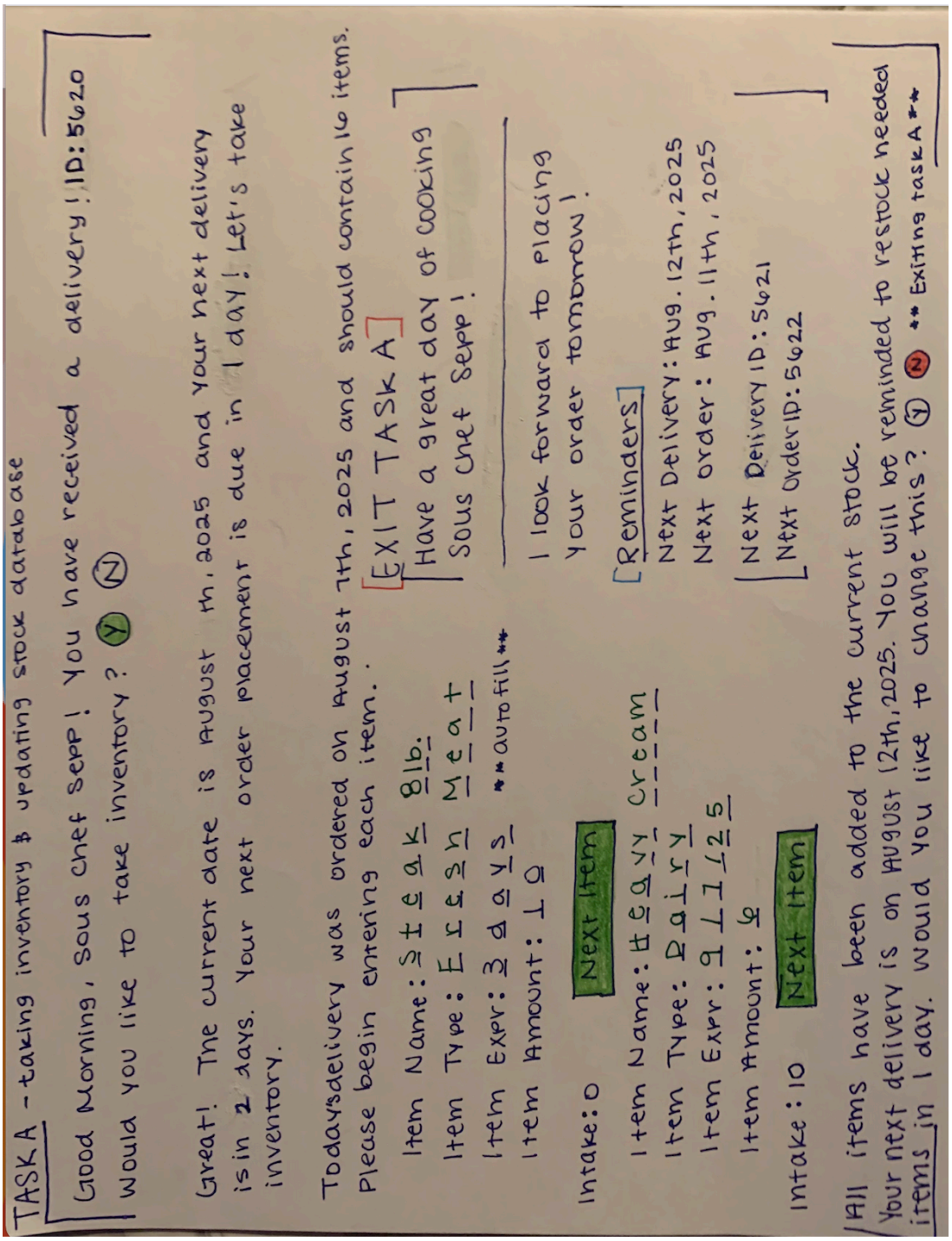
# Planning

A. Update inventory after a delivery. Track item name, amount, and type. If the item is not of type fresh, prompt user for an expiry date too. Store data in the current stock of items.
B. Tracking the amount of supply for each item. Check for sufficient stock of item and if expired/to be expired.

# Sketches



# Paper Prototype




# Testing Script

---

Task A: Update Current Stock

A. Alert: Received Delivery
    a. *dID* = 5620
        i. Delivery ID (#)
    b. *oDate* = 8/7/25
        i. Order Placement Date of *dID*
    c. *dDate* = currDate = 8/10/25
        i. Delivery date
        ii. Set to current date
    d. *lOrder* = 8/9/25
        i. Last Order Placement Date
    e. *lID* = *dID* + 1
        i. *Last Order Placement ID*
    f. *nDate* = *lOrder* + *nDelivery*
        i. Next Expected Delivery Date
    g. *nDelivery* = 3 tomorrow
        i. Delivery date is initialized to 3 days in advance.
        ii. The user can change this initialized value if they choose to. Therefore, the value is reinitialized every time in case the user previously had a stock surplus from a delivery and would like to be reminded to place the next order later than 3 days from the current date.
B. Alert: Take Inventory
    a. *nDelivered:* number of items ordered
    b. User Select: Yes
        i. Continue
    c. User Select: No
        i. Wait 2 hours before reminding user to take inventory (return to B)
C. Alert: Enter Item Name
    a. *ItemName* = User Input
D. Alert: Enter Item Type
    a. *ItemType* = User Input
        i. If(*ItemType* = Fruit/Vegetables)
            1. *ItemExpr* = 10
        ii. If(*ItemType* = Meat)
            1. *ItemExpr* = 3
        iii. If(*ItemType* = Fish/Seafood)
            1. *ItemExpr* = 2
        iv. Else
            1. Alert: Enter Item Expiry Date
            2. *ItemExpr* = User Input
    b. Continue
E. Alert: Enter Item Amount
F. Save item data in current stock database
G. *ItemCount += 1*

H. *ItemStock* = *ItemAmt* used daily : *ItemExpr*
I. Continue Taking Inventory
    a. If(*ItemCount* = *nDelivered*)
        i. End Task
    b. Else
        i. Continue (return to C)

Recitation 9: Paper Prototyping
Preston Dotsey & Pourna Sengupta
CSCI 3002 – Section 106

# Prototype Testing

User: Preston Dotsey
Task A

Feedback: The prototype was thorough and anticipated user choices. It provided the options necessary to take inventory and update the current stock of a restaurant. Overall, it wasn't too complicated and could be used functionally with other tasks added like viewing past order placements or orders in progress. Adding an option to note if an item was not received can also improve the prototype.

Summary: User testing went fairly smoothly as I focused on creating a prototype that covered any potential actions that a user may need in order to complete the task of taking inventory from a newly received delivery. One thing I did find to be a potential issue is the number of alerts that the user must receive in order to properly communicate with the prototype.

# Prototype Revision

As stated in the feedback received, the prototype could be improved by allowing the user to note if any items were missing from the delivered order. An option to alert the supplier can also be added to help simplify the process for the user. Including other tasks along with this one would also improve the larger prototype (including both Task A and B). These tasks can further simplify the responsibilities of keeping track of the current stock and taking inventory regularly. Furthermore, food shortage can be reduced even more by including a scanner that photographically scans the items in stock. If the items look to be expired (i.e., moldy, mushy, wrinkly, discolored, etc.), the user will be alerted, and an order confirmation will be requested. To reduce the number of alerts and further automate taking inventory, the prototype can be revised to simply check that the items ordered are included in the delivery and possibly even sort and store them for the user. At the beginning or end of each day, the prototype can update the current stock to account for usage and to keep track of increased or decreased daily usage in order to update *ItemStock* and make sure that enough of an item is ordered when demand increases and avoid surplus with decreases in demand.