**הוצאת הזמנה מספק עקב חוסר**

שם תרחיש- Use Case Name:

ייצור הזמנה חדשה עקב מחסור.

תיאור התרחיש- Textual description:

כאשר הכמות של מוצר מסוים מגיעה לנקודת המינימום (המוגדרת בפרטי מוצר כלשהו), המערכת תנפיק הזמנה חדשה על מנת לקבל אספקה נוספת עבור אותו המוצר.

רשימת שחקנים- List of actors:

עובדי המחסן, ספקים

תנאים מקדימים- Pre-conditions:

כמות המוצר חוצה את נקודת המינימום המוגדרת לו.

תנאי סיום- Post-conditions:

עבור כל מוצר שהכמות שלו חצתה את נקודת המינימום המוגדרת לו, נוציא הזמנה ונשלח לספק.

תרחיש הצלחה ראשי - Main success scenario

1. Inventory worker choose option "update inventory" after stocktaking in the shop.
2. For each product:
   1. Worker update quantities.
   2. If the product got below minimum amount:
      1. Mark the product as missing. (Issue 'itemOrder' object)
      2. Add this product to shortage order.
3. The system passing immediately the shortage order to suppliers.
4. Suppliers issue an order.

**Pseudo code**

1. User type "update inventory" 🡪 UpdateInvWorker ()
2. While (item\_id != 0)
   1. item.updateMyQuantities (quantityMissStock, quantityMissShop, '-')
   2. if(item.checkMinimumQuant ())
      1. itemOrder 🡨 issueOrderForShortageItem ()
      2. shortageOrder 🡨 addItemToShortageOrder(itemOrder)
3. Inv2supCtrl.placeNewShortageOrder(shortageOrder)
4. myOrderAndProductManagment.createRegularOrder (order)

Alternatives and extensions:

At any alternatives or extensions point the user will get appropriate message that describe the error, and go back to the update products loop:

1. Wrong input.
2. Item id isn’t exist.
3. Place shortage order didn’t succeed.

**Pseudo code**

1. If input != 'id quantityStock quantityShop'
   1. Print "wrong input! Type again"
2. If id in input is not exist
   1. Print "Item isn’t exist in the inventor! Type again"
3. Result res = placeNewShoratgeOrder ()
   1. If(res.isFailure ())
      1. Print res.getMessage ()