**F. Issue a new order due to shortage**

Textual Description:

When the quantity of certain product reaches to minimum point (defined in the details of any product), the system will issue new order to get more supply of that product immediately.

List of actors:

Inventory worker, Supplier.

Pre-conditions:

Product quantity gets below minimum.

Post-conditions:

For each product that its quantity got below minimum, order was issued and sent to the supplier.

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 ()