

Inventory Module

ADSS\_Group\_B

Use Case contracts and diagrams

Ori Adika - 209200559

Segev Olpak - 325176188

1. Add Product

|  |  |
| --- | --- |
| **Section** | **Purpose** |
| **Name** | Add Product |
| **Description** | Storage manager wants to add product to the Inventory system. |
| **Actors** | Storage manager. |
| **Precondition** | The product he is about to enter is not in the system.  The makat code of the product was never used. |
| **Post-conditions** | The product was successfully entering as a new product in the system.  The System printing success message. |
| **Basic course of action** | 1. Storage manager enters '3' on menu, asking to add a new product to the system. 2. Storage manager enters the following information (product code, minimal amount, product name, location, manufacturer, cost price, current price, enters the categories and the popularity in a scale from 1-3[LOW - HIGH]). 3. The system save the information and constructs a new instance of the new product. 4. The system sends a success message back to the storage manager. |
| **Alternate courses** | The product code is already use, the system sends an error message with this information to the user and back to menu screen. |

Contract:

1. **Operation** : Cross addProduct(makat : int, minimalAmount : int, name : String, place : String, manufacturer : String, costPrice : double, currentPrice : double, categories : String[], popularity : int)
2. **References** :Use Case : AddProduct
3. **Preconditions** : There is not product with the same makat/Product code.
4. **Postconditions**:
5. A Product instance prod was created(instance creation)
6. A category instance cat was created (in case it was not in the system in the first place).
7. The product instance prod is enters into cat.categoryProducts Map with product code as key.
8. The product instance prod is enters into inventoryfacade Map with product code as key.

A connection is made between product and inventrory.

1. Inventory update and notification of shortages

|  |  |
| --- | --- |
| **Section** | **Purpose** |
| **Name** | Inventory update and notification of shortages |
| **Description** | Storage worker wants to remove one item from Inventory system. |
| **Actors** | Storage worker. |
| **Precondition** | There is item with serial number of the product that the worker wants to remove.  The product we want to remove its item is already in the system. |
| **Post-conditions** | The item was successfully removed from the system.  A success message was sent back to the worker.  The inventory was updated. |
| **Basic course of action** | 1. Storage worker enters '2' on menu, asking to remove an item from certain product that is already in the system. 2. Storage worker enters the following information (product code of the product, serial number of the item he wants to remove). 3. The system look for this item and removes it from the system updating on its way the system. 4. The system sends a success message back to the storage manager. |
| **Alternate courses** | Depends on the product popularity, if there is shortage in items from this specific product the system will sends a related message back to the storage worker along side a success message informing the item was successfully removed. |

Contract:

1. **Operation** : Cross removeItem(makat : int, sn: int)
2. **References** :Use Case : Inventory update and notification of shortages
3. **Preconditions** : There is product instance with product.makat = makat in the system.

There is item instance with sn = item.sn in the system.

1. **Postconditions**:
2. Item instance item was removes from product.items.
3. Item is removed from data base.
4. A report is made in case there is too little item instances.

Contract:

1. **Operation** : Cross addItem(makat : int, sn: int, expirdDate : Date, inStorage : boolean)
2. **References** :Use Case : Inventory update and notification of shortages
3. **Preconditions** : There is product instance with product.makat = makat in the system.
4. **Postconditions**:
5. Item instance item was added from product.items and a connection is made between the instances.
6. Item data is added to data base.

Version History

Table 3: Version History

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Remarks** |
| 16/06/2024 | Segev Olpak | full draft |
| 16/06/2024 | Ori Adika | full draft |