**ניתוח ועיצוב מערכות תוכנה**

**עבודה 2**

**מלאי ת"ז ושמות : עידן חפץ 213633332 ניר אהרוני 213161953**

**ספקים ת"ז ושמות : דנה פרידמן 213967599 אייל שפירו 213958804**

**Changes from Assignment 2**

1. Service Layer was added to enable accessing system information from sources that cannot directly interact with the program, such as Web applications or external GUI applications.
2. The "Cashier" role was removed as the 3 roles in the system, Warehouse Worker, Store Manager and Supplier Connections Manager were defined in the requirements.
3. The SupplierController and StockController are now singletons, to make sure the cached information is synced between the different services.

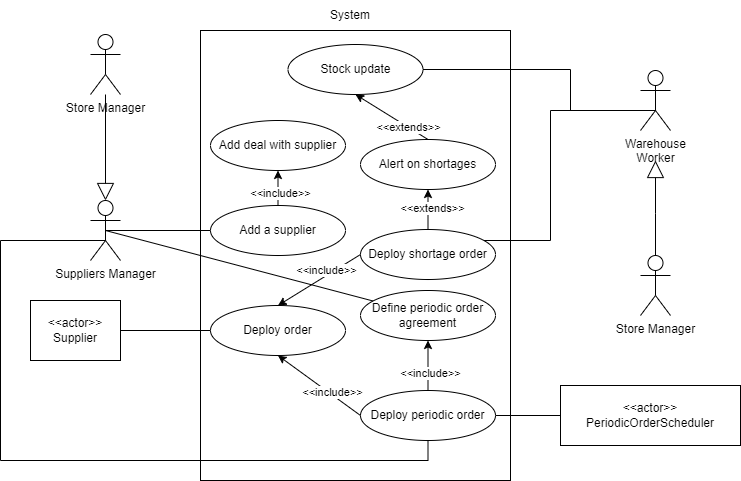
The Class Diagrams, Object Diagrams, Flowcharts, Use Case diagram and ERD were updated to match these changes.

**חלק א': הוראות משתמש**

**הוראות משתמש עבור 4 תרחישי השימוש הנדרשים מצורפים בקובץ Instructions.pdf.**

**חלק ב': עדכון תרשימים קודמים**

1. **ניתוח התנהגותי: תרחישי שימוש**
2. **תרשים תרחישי שימוש לכל המערכת**

****

1. **פירוש על תרחישי השימוש**

**Use case name:** **Send order to supplier due to shortage**

**Textual Description:** When the manager looks at a shortage report, he can order the products in shortage, by choosing the extra amount for each product. Then the system chooses the optimal division of the products to different suppliers. The manager can choose a contact person for each supplier (if there are any). The manager can also decide whether to notify the contact about the new order via email.

**List of Actors:** Supplier, Warehouse Worker, Supplier Contact, Stock Manager

**Pre-conditions:** A shortages report that contains all the products in shortage including their amount.

**Post-conditions:** The order has been created. All of the relevant supplier’s contacts have been informed via email.

**Main success scenario\Alternatives: Described using a flow chart in attached file ShortageOrderUseCase.pdf**

**Use case name: Send order to supplier periodically.**

**Textual Description:** The suppliers manager adds a periodic order by choosing a supplier, products and amounts of products. If the supplier has fixed delivery days, they would be the order’s fixed days. If not, the manager will choose them manually. The Periodic Order Scheduler will deploy the order every week, on the specified days.

**List of Actors:** Supplier, Stock Manager, Periodic Order Scheduler, Supplier Contact

**Pre-conditions:** The supplier has a defined periodic order that needs to be deployed today and the supplier has the required amount of every product in the periodic order.

**Post-conditions:** The order has been created and the assigned contact has been informed via email.

**Main success scenario\Alternatives: Described using a flow chart in attached file PeriodicOrderUseCase.pdf**

1. **ניתוח התנהגותי: Sequence Diagrams**

**חוזים ותרשימי רצף לכל פעולה בתרחישי השימוש**

**Send order to supplier due to shortage:**

**Throughout the shift, Warehouse Workers report missing or bought products**

**Pre-condition**: A product instance with the specified product instance id exists in the system. The product that product instance id represents exists in the system.

**Post-condition**: product instance doesn’t exist in the system

**Sequence Diagram: Described in appended file ReportMissing.pdf**

**Warehouse Worker generates a shortage report**

**Pre-condition**: Every product that was sold or reported as missing was removed from the system. Every product that was supplied was added to the system.

**Post-condition**: The amounts of each product are collected from the system.

**Sequence Diagram: Described in appended file GenerateShortageReport.pdf**

**"Empty shortage report" message is presented**

**Pre-condition**: Every product in the branch is above its minimum required amount.

**Post-condition**: "Empty shortage report" message is presented in the user interface.

**Sequence Diagram: Described in appended file EmptyShortageReport.pdf**

**Warehouse Worker chooses an extra amount of each product to order, and an order is made**

**Pre-condition**: At least one product in the branch is below its minimum required amount.

**Post-condition**: An order has been generated as a result of the stock manager’s input. The order includes the products ordered from each supplier and the amount of each product.

**Sequence Diagram: Described in appended file MakeShortageOrder.pdf**

**For every Supplier that the order was divided to, the Warehouse Worker chooses a contact of the supplier to assign to the order (if they have any)**

**Pre-condition:** An order has been generated as a result of the stock manager’s input. The order includes the products ordered from each supplier and the amount of each product.

**Post-condition:** Every supplier in the order that has at least one contact recorded in the system is assigned a single contact for this order.

**Sequence Diagram: Described in appended file AssignContactToOrder.pdf**

**Warehouse Worker chooses whether to send an order email to the contacts, and an email is sent to the relevant contacts**

**Pre-condition:** Every supplier that has at least one contact recorded in the system is assigned a single contact for this order.

**Post-condition**: An email is sent to each contact assigned to a supplier that the stock manager requested an email to be sent to.

**Sequence Diagram: Described in appended file SendEmailToContact.pdf**

**Send order to supplier periodically:**

**Suppliers or Stock Manager chooses products and amount for each product for periodic order with a supplier and a branch to supply the products to**

**Pre-condition**: The supplier exists in the system.

**Post-condition**: The products of the periodic order and their amounts are recorded for this supplier in the system (If all of the products are supplied by the supplier in a sufficient amount).

**Sequence Diagram: Described in appended file CreatePeriodicOrder.pdf**

**Error message is shown to Suppliers Manager**

**Pre-condition**: Supplier is unable to supply one of the products selected by the Suppliers Manager

**Post-condition**: An “Supplier is unable to supply <name of product>” error message is shown to Suppliers Manager on the user interface.

Note: Since the procedure described above is an alternative to the procedure before it, we created one sequence diagram for both of them.

**Suppliers or Stock Manager chooses the days of the week at which the periodic order will be deployed.**

**Pre-condition**: Supplier is able to supply all products required in the periodic order and the supplier doesn’t have any fixed supply dates

**Post-condition**: The periodic order information now includes the days of the week when the periodic order will be supplied (The days the supplier manager just chose or the supplier’s fixed delivery day).

**Sequence Diagram: Described in appended file SetPeriodicOrderDays.pdf**

**Suppliers or Stock Manager assigns one of the supplier's contacts to the periodic order**

**Pre-condition**: The supplier has at least one contact.

**Post-condition**: The chosen contact is assigned as a contact to the periodic order.

**Sequence Diagram: Described in appended file SetPeriodicOrderContact.pdf**

**Every 24 hours, the Periodic Order Scheduler iterates over all suppliers and checks if a periodic order needs to be deployed today. If so, the order is deployed, and**

**an email is sent to suppliers who have contacts assigned to the periodic order**

**Pre-condition:** 24 hours have passed since the last time the Periodic Order Scheduler ran this check or since the program started running.

**Post-condition**: For each supplier that has a periodic order that needs to be deployed on the day of the check, an order including the products of the periodic order is recorded in the database. If a contact is assigned to the periodic order, the contact is assigned to the newly created order and an email is sent to them detailing the order.

**Sequence Diagram: Described in appended file DeployPeriodicOrders.pdf**

1. **ERD**

**An ERD diagram is described in the attached file EntityRelationsDiagram.pdf**

1. **תרשימי מחלקות ואובייקטים**

**A class diagram for the Suppliers module is described in the attached file SuppliersClassDiagram.pdf**

**A class diagram for the Stock module is described in the attached file StockClassDiagram.pdf**

**A class diagram of the Service Layer, which is shared between the two modules, is described in the attached file ServicesDiagram.pdf**

**The requirements and assumptions for the suppliers and stock module are described in the attached file Requirements.pdf**

**Scenarios for Object Diagram:**

**תרשים האובייקטים בקובץ המצורף BambaOrderObjectDiagram.pdf מייצג את מצב המערכת של ספקים תרחיש 1 (לאחר התרחיש).**

**תרשים האובייקטים בקובץ המצורף PeriodicOrdersScenario.pdf מייצג את מצב המערכת של ספקים תרחיש 2 (לאחר התרחיש).**

**תרשים האובייקטים בקובץ המצורף**

**StronglySatisfiable\_ObjectDiagram\_Stock.pdf מייצג את מצב המערכת של מלאי תרחיש 1 (לאחר התרחיש).**

**second\_ObjectDiagram\_Stock.pdf מייצג את מצב המערכת של מלאי תרחיש 2 (לאחר התרחיש).**

**Stock**

**Object Diagram 1 (strongly satisfiable) :**

Miss. Anya a Supplier Manager entered three new products into the system

milkTnuva5% with the properties {id:2, name : "milkTnuva5%", manufacturer : "Tnuva" }

milkTnuva3% with the properties {id:1, name : "milkTnuva3%", manufacturer : "Tnuva" }

redMeat3% with the properties {id:3, name : "redMeat”, manufacturer : "adom adom"}

Mr. Platypus, a warehouse worker, is working today on branch number 1.

He insertsProduct into the branch, milkTnuva5% with the properties {id:2, priceSold: 5, demand : 8, minimum : 3, placeInStore : fridges, path : superLee/dairy/Milk/1L}

He insertsProduct into the branch, redMeat with the properties {id:3, priceSold: 5, demand : 8, minimum : 3, - placeInStore : butcherShop, - path : superLee/Meat}

He inserts Product into the branch, milkTnuva3% with the properties {id:1, priceSold: 5, demand : 8, minimum : 3, placeInStore : fridges, path : superLee/dairy/Milk/1L}

Then, he inserts Instances of products into the branch.

he inserts milkTnuva3%\_P1 with the properties { instanceId : 1, productId : 1, boughtPrice: 4.3, expireDate: 20/5/2023}

he inserts milkTnuva5%\_P4 with the properties { instanceId : 4, productId : 2,

boughtPrice: 4.1, expireDate: 20/5/2023}

Then he sees that milkTnuva5%\_P4 is open, so he reports defectivity on it with the description “package is damaged”:

Miss. Perry, a supplier manager, is working today on branch number 1.

She creates a new discount on milkTnuva5%, dicount3, with the properties { beginingDate: 2022-03-02, expiredDate: 2022-03-08, discountNumericValue: 10}

She creates a new discount on category - superLee/Meat, dicount2, with the properties { beginingDate: 2022-03-02, expiredDate: 2022-03-08, discountNumericValue: 10}

She creates a new discount on category - superLee/dairy/Milk/1L, dicount1, with the properties { beginingDate: 2023-04-02, expiredDate: 2023-04-08, discountNumericValue: 5}

**Object Diagram 2 :**

Miss. Anya a Supplier Manager entered three new products into the system

milkTnuva3% with the properties {id:1, name : "milkTnuva3%", manufacturer : "Tnuva" }

chocolate500g with the properties {id:2, name : "chocolate500g", manufacturer : "Elit" }

Mr. Platypus, a warehouse worker, is working today on branch number 1.

He inserts Product into the branch, milkTnuva3% with the properties {id:1, priceSold: 5, demand : 8, minimum : 3, placeInStore : fridges, path : superLee/dairy/Milk/1L}

He inserts Product into the branch, chocolate500g with the properties {id:2, priceSold: 5, demand : 8, minimum : 3, - placeInStore : area 12, - path : superLee/chocolate}

he inserts chocolate\_P1 with the properties { instanceId: 3, productid : 2, boughtPrice: 4.3, expireDate: 20/5/2023}

he inserts chocolate\_P2 with the properties { instanceId: 4, productId : 2, boughtPrice: 4.6, expireDate: 20/5/2023}

**Suppliers**

1. נתאר תרחיש אפשרי במערכת. נניח שמנהלת סניף מספר 0 מעוניינת לרכוש 200 שקיות במבה. לחברה יש הסכם עם 2 ספקים המספקים שקיות במבה: חברת "אסם" ו"פריגת". שניהם מוכרים כל שקית במבה ב 6 שקלים, אך לפי ההסכם עם חברת "אסם", בקנייה של מעל 100 שקיות במבה תינתן הנחה של 50%. לשתי החברות קיים איש קשר ושמו "ביל". נניח שמנהל הספקים נכנס לתפריט ההזמנות, וביקש מהמערכת לחשב אוטומטית הזמנה אופטימלית ל200 שקיות במבה, המערכת הציגה לו את ההזמנה האופטימלית הכוללת הזמנה של 200 שקיות במבה מחברת "אסם", מנהל הספקים בחר את "ביל" עבור איש הקשר של חברת אסם, והוא הוחזר לתפריט ההזמנות.
2. המשתמש רוצה להזמין מוצר שהמספר המזהה שלו הוא 0 (productId = 0) בכל יום שלישי וחמישי.

תחילה הוא מוסיף את פרטי הספק "ShapiroLtd", שבינהם ימי האספקה הקבועים שלו - שלישי וחמישי. לאחר מכן, הוא מוסיף את כל המוצרים אשר הוא מספק. בפרט, הוא מוסיף להסכם עם הספק את המוצר שהproductId שלו הוא 0, כך שהמספר הקטלוגי שלו הוא 123.

כדי ליצור הזמנות תקופתיות, הוא יוצר PeriodicOrderAgreement עם הספק. בהסכם זה מצויין בין היתר שהספק יספק 100 יחידות של המוצר 123, ושהימים שבהם ההזמנה התקופתית צריכה לצאת הם שלישי וחמישי (הימים האלו נקבעים אוטומטית בהסכם משום שלספק יש ימי אספקה קבועים).

ההזמנה o1 נוצרה אוטומטית ביום שלישי.

ביום רביעי, המשתמש מוסיף הנחות על המוצרים וגם על ההספק עם הספק "ShapiroLtd".

כעת, ההזמנה o2 נוצרת גם אוטומטית ביום חמישי, בהתאם למחירים יחד עם ההנחות שנוספו.

בנוסף, המשתמש גילה שלקראת יום שישי, יש צורך בערך 30 יחידות של המוצר הנ"ל. לכן הוא יכול להוסיף את ההזמנה, o3, באופן ידני דרך החלון "createOrderView".