

## מסמך שינויים :

### 1. ERD :



#### 1. Item\_order : לבצע מעקב על כמות הפריטים שהוזמנו על ידי הזמנות מיוחדות

(דחופות). הוספנו זאת כדי למנוע הזמנה חוזרת של מוצרים שאמורים להגיע.

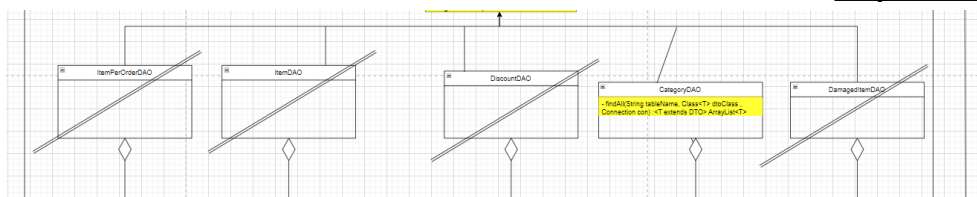
#### 2. Waiting\_list : טבלת ששומרת מוצרים שסופקו על ידי הספקים אך טרם מוקמו בחנות על

ידי העובדים

#### 3. Constants : משתנים שהיינו צריכים לטובת תפקוד תקין של המערכת. (counter – סופר

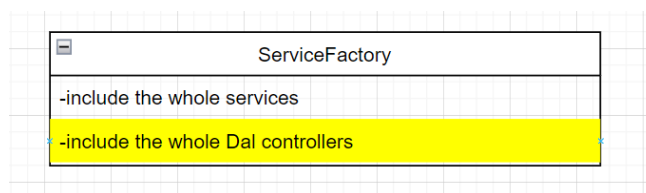
את ההנחות , date – תאריך עדכני של המערכת)

### 2. תרשים מחלקות :



1.

מבין הDAO's , מחקנו את אלו שלא רלוונטיות, כלומר את אלו שאין להם פונקציונליות מיוחדת מעבר לDAO הגנרי שסיפק את הפונקציונליות הנדרשת עבור המחלקות הללו. השארנו את המחלקה CategoryDAO מכיוון שהיה צורך לדרוס את השיטה findAll וליצור שיטה חדשה המחזירה את המידע ממיון לפי קטגוריות.



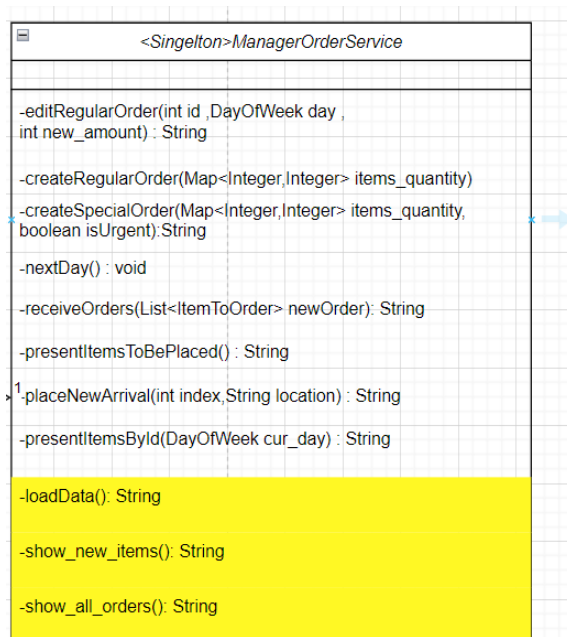
2.

היה נדרש ליצור בצורה חד פעמית את הDAL controllers , לכן ביצענו זאת בFactory.

-loadData(): String

3.

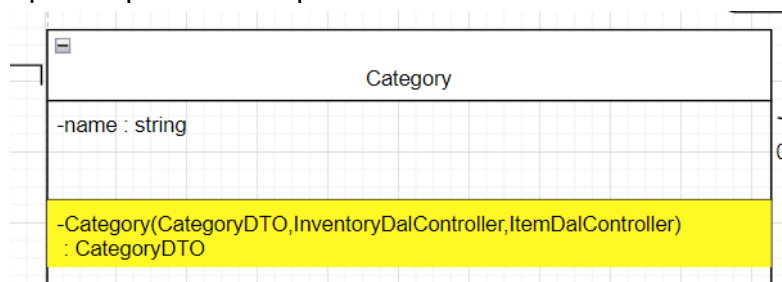
עבור מחלקות מסוימות הוספנו את הפונקציונליות של טעינת המידע מהDB לRAM.



.4

הוספנו פונקציונליות חדשה למערכת – כפי שמתואר בהמשך בטבלת הדרישות המעודכנת. (3)

5. במחלקות הBusiness הוספנו בנאים המקבלים DTO התואם לאותה המחלקה ובהתאם את הרלוונטי עבור אותה המחלקה. לדוגמא מתוך המחלקה Category :



Show\_special\_orders , show\_all\_orders  
show\_new\_items – תוספות בהתאם למסמך הדרישות.

```

<Singleton>OrderController

-items_to_place: List<ItemToOrder>

-regularOrders: Map<Integer, Integer> items_quantity : void

-regularOrders: Map<Integer, Integer> items_quantity
,boolean isUrgent : void

-makeAutomaticallyOrder(DayOfWeek curDay) : String

-amountOfReceivedItem(List<ItemToOrder> items_list ,
String manufacturer_name , String product_name): int

-editRegularOrder(int id, DayOfWeek day,
int new_amount) : void

-receiveOrders(List<ItemToOrder> newOrder): void

-placeNewArrival(int index, String location): void

-presentItemsToBePlaced(): String

-presentItemsByDay(DayOfWeek cur_day): String

-show_special_orders(): void

-show_all_orders(): void

-loadData(): void

-show_new_items(): String

```

ז. למחלקה DAO הוספנו פונקציות גנריות : findAll – מחזירה את כל הרשומות כ - DTO's מטבלה מסוימת , פונקציה המסייעת לטעינת המידע לRAM.

findAllOfCondition - מחזירה את כל הרשומות העונות על תנאי מסויים כ - DTO's מטבלה מסוימת, פונקציה המסייעת לטעינת המידע לRAM.

DeleteAll – מחיקת כל המידע מטבלה מסוימת.

```

<<DAO>>

-insert(DTO dto, Connection con) : void

-delete(DTO dto, Connection con) : void

-update(DTO dto, Connection con ) : void

- findAll(String tableName, Class<T> dtoClass,
Connection con) :<T extends DTO> ArrayList<T>

- findAllOfCondition(String tableName, String
conditionKey, Object conditionValue, Class<T>
classDTO, Connection connection)
:<T extends DTO> ArrayList<T>

- deleteAll(Connection connection,
String tableName):void

```

### 3. מסמך דרישות :

LOW	NTH	V	המערכת תאפשר הצגת המידע של המוצרים שניתן לספקם. מידע זה יהיה שם מוצר, שם יצרן, כמות אפשרית להספקה	16
LOW	NTH	V	המערכת תאפשר צפייה בפרטי ההזמנות (פרטי המוצרים : כפי שמתואר בדרישה 16) אשר מיועדות להגיע בשבוע הקרוב, מסודרות לפי ימי השבוע.	17

1. (16) עלה הצורך תוך כדי כתיבת הקוד להציג בצורה נוחה את המוצרים שניתן לספקם על ידי הספקים, זאת על מנת לתת לעובדים לבחור מתוך מידע זה מוצרים שאותם ירצה להוסיף למערכת \ להזמין לחנות.
2. (17) תוך כדי כתיבת הקוד ראינו כי עלה הצורך להציג ללקוח בצורה נוחה את ההזמנות לשבוע הקרוב, כדי לתת לעובד\מנהל את האפשרות לתכנן בצורה נכונה את ההזמנות שיירצה לבצע.