

Super Li Management System

Group S

Modules Stock, Suppliers

Stock:

- Amit Hassan – 206966004
- Gili Veltz – 318916384

Suppliers:

- Nadav Katlav – 315401448
- David Volodarsky - 208188326

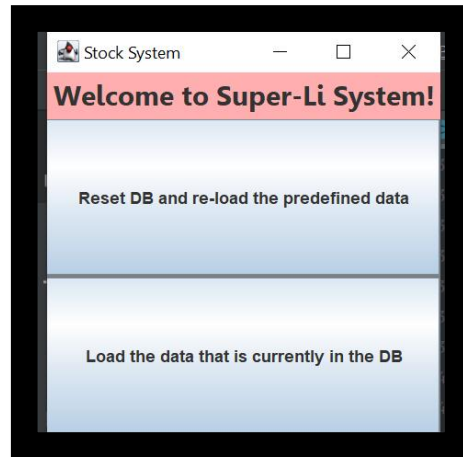
הוראות הפעלה למערכת סופר-לי

בהרצת המערכת בשורת מערכת ההפעלה יש להכניס CUI/GUI ואת התפקיד הרלוונטי ובהתאם יפתחו החלון המערכת.

עם עליית המערכת יינתנו שתי אפשרויות:

- (1) טעינת נתונים באופן ידני.
- (2) טעינת נתונים לדוגמה (מפורטים בסוף המסמך).

כך זה יראה ע"י שימוש ב-GUI:



מערכת המלאי:

לאחר הבחירה ייפתח תפריט הסניפים (מקשים את המספר המתאים):

1. יצירת סניף חדש (הקש 1)

מילוי הפרטים הבאים:

✓ שם הסניף

✓ עיר

2. בחירת סניף קיים (הקש 2)

בחירת הסניף מרשימת הסניפים הקיימים על ידי הקשה על המספר המתאים.

לאחר מכן ייפתח התפריט הראשי (מקשים את המספר המתאים):

1. קבלת סחורה (הקש 1)

הכנסת מספר המוצרים המתקבלים (בדגש על מוצרים, לא פריטים!)

הכנסת מק"ט של המוצר

- אם המוצר לא קיים במערכת ממלאים את הפרטים הבאים:

- ✓ בוחרים את הקטגוריה אליה המוצר ישתייך באופן היררכי (בוחרים קטגוריה, לאחר מכן תת קטגוריה וכן הלאה). בכל שלב ניתן להקיש 0 ולהוסיף קטגוריה חדשה עבור המוצר (אם לא קיימת קטגוריה מתאימה).

✓ שם המוצר

✓ מחיר עלות המוצר מהספק

✓ מחיר מכירת המוצר

✓ יצרן

✓ ביקוש המוצר

- ✓ זמן האספקה
 - ✓ כמות מינימלית במלאי
 - לאחר מכן מכניסים מידע אודות כמות המוצר:
 - ✓ כמות לחנות
 - ✓ כמות למחסן
 - ✓ תוקף- בפורמט dd-mm-yyyy בלבד! (למשל 16-07-2023)
- וחוזר חלילה למוצר הבא שהגיע למלאי.

2. הדפסת דוחות (הקש 2)

בחירת הדוח הרצוי:

1. דוח מלאי- בוחרים קטגוריות רצויות לדוח (בכל שלב ניתן להקיש 0 לסיום הבחירה, כלומר לא לבחור תת-קטגוריה).
2. דוח פגומים- בוחרים קטגוריות רצויות לדוח (בכל שלב ניתן להקיש 0 לסיום הבחירה, כלומר לא לבחור תת-קטגוריה). לכל קטגוריה בוחרים את סוגי הפגם להצגה בדוח.

3. עדכון פריט פריט (הקש 3)

בחירה מהתפריט הבא:

1. דיווח על פריט כפגום- הקשת מק"ט הפריט ובחירת סוג הפגם.
2. עדכון ביקוש של מוצר- הקשת מק"ט המוצר והקשת הביקוש החדש.
3. עדכון זמן אספקה של מוצר- הקשת מק"ט המוצר והקשת זמן האספקה החדש.
4. שינוי מיקום מוצר- הקשת מק"ט הפריט. אם הוא היה בחנות יועבר למחסן ולהפך.
5. עדכון מחיר עלות של מוצר- הקשת מק"ט המוצר והקשת המחיר החדש.

4. קופה (הקש 4)

מתחילים רכישה חדשה על ידי הקשה על 1 ומכניסים את המק"ט של הפריט (בדגש על פריט ולא מוצר). הנתונים במערכת מתעדכנים לפי המוצרים שנרכשו. במידה והפריט שנרכש פגום- תוקפץ התראה. במידה ובעקבות הרכישה יש מוצר שהכמות שלו במלאי הגיעה לכמות המינימלית- תוקפץ התראה. לאחר כל פריט שנזין תוצג האופציה להזין פריט נוסף, או לסיים את הרכישה. בסיום הרכישה יוצג המחיר שנגבה בפועל עבור כל פריט וסכום הרכישה הכולל.

5. ניהול הנחות (הקש 5)

יש לבצע בחירה בין הזנת הנחה חדשה לבין עריכה/מחיקה של הנחה קיימת. לאחר מכן בוחרים את סוג ההנחה:

- הנחה עבור קטגוריה - בוחרים את הקטגוריה אליה ההנחה תשתייך/ משתייכת באופן היררכי (בוחרים קטגוריה, ואז תת קטגוריה וכן הלאה). בכל שלב ניתן להקיש 0 לסיום הבחירה, כלומר לא לבחור תת-קטגוריה.
 - הנחה עבור מוצר - מזינים את המק"ט של המוצר.
 - הנחה עבור פריטי מוצר פגומים - מזינים את המק"ט של המוצר ובוחרים את סוג הפגם.
- במידה ובחרנו בהזנת הנחה חדשה ממלאים את הפרטים הבאים:
- תאריך התחלה - בפורמט dd-mm-yyyy בלבד! (למשל 16-07-2023)
 - תאריך סיום - בפורמט dd-mm-yyyy בלבד! (למשל 16-07-2023)

- יחידת מידה - אחזים / מספר שקלים
- תעריף ההנחה
- *עבור פריטי מוצר פגומים בלבד! – סוג הפגם
- במידה ובחרנו בעריכה / מחיקה של הנחה קיימת, נבחר בין השניים.
- * אם בחרנו במחיקה – ההנחה תימחק.
- * אם בחרנו בעריכה – יש לבצע בחירה בשדה שנרצה לערוך בין פרטי ההנחה שפורטו מעלה, ולאחר מכן להזין את ערכו החדש.

6. ניהול הזמנות (הקש 6)

- יש לבצע בחירה בין הוספת הזמנה חדשה לבין עריכה/מחיקה של הזמנה קיימת.
- אם בחרנו בהזנת הזמנה חדשה נבחר הזמנה תקופתית או הזמנת אקספרס.
- ✓ להזמנה תקופתית (אוטומטי) ממלאים את הפרטים הבאים:
 - בחירת ימי אספקה לסחורה מתפריט הימים
 - עבור כל מוצר שנרצה להזמין:
 - מק"ט המוצר
 - במידה והמוצר עדיין לא קיים בחנות כלל – שם המוצר
 - כמות שנרצה שתסופק
 - נבחר אם להזין מוצר נוסף או שניתן לשלוח את ההזמנה.
- ✓ להזמנת אקספרס ממלאים את הפרטים הבאים:
 - עבור כל מוצר שנרצה להזמין:
 - מק"ט המוצר
 - במידה והמוצר עדיין לא קיים בחנות כלל – שם המוצר
 - כמות שנרצה שתסופק
 - נבחר אם להזין מוצר נוסף או שניתן לשלוח את ההזמנה.
- במידה ובחרנו בעריכה / מחיקה של הזמנה קיימת:
 - ✓ נבחר הזמנה מתפריט ההזמנות
 - ✓ נבחר אם ברצוננו למחוק / לערוך את ההזמנה
 - אם בחרנו במחיקה – ההזמנה תמחק
 - אם בחרנו בעריכה – נבחר את סוג העדכון שנרצה לבצע להזמנה:
 - ימי אספקה - בחירת ימי אספקה חדשים מתפריט הימים
 - מוצרים וכמויות – נבחר אם ברצוננו להוסיף מוצר חדש להזמנה או לשנות כמות / למחוק מוצר שכבר הזמננו.
 - *להוספת מוצר – נזין מק"ט, שם (במידת הצורך) וכמות.
 - *לעדכון מוצר קיים – נבחר מוצר מהמוצרים הקיימים בהזמנה שיוצגו לנו, ונזין את הכמות המעודכנת שנרצה. במידה ונרצה למחוק את המוצר מההזמנה – נזין 0.
 - *נבחר אם ברצוננו לעדכן מוצר חדש או שסיימנו.
- ✓ במידה וניסינו לבצע את העדכון פחות מ-24 שעות לפני הגעת ההזמנה – נקבל שגיאה. במידה ולא – נקבל הודעה כי ההזמנה עודכנה בהצלחה.

7. חזרה לתפריט הסניפים (הקש 7)

הערות:

- אין התעלמות מרווחים בקריאת הקלט.
- נכתבו Unit Tests תחת התיקיה Tests.

נתונים לדוגמה בטעינת המערכת

BRANCH		
ID	Name	City
1	Branch 1	Tel Aviv
2	Branch 2	Ashdod

Branch 1

CATEGORY				
ID	Name	Sub-Categories	Products	Discounts
1	Dairy	Milk, Cheese	-	-
2	Meat	Steak	-	-
1-1	Milk	1 liter, 2 liter	-	-
1-2	Cheese	400 gr	-	-
1-1-1	1 liter	-	111, 222	1
1-1-2	2 liter	-	112	-
1-2-1	400 gr	-	333	-
2-1	Steak	300 gr	-	-
2-1-1	300 gr		444	-

PRODUCT										
ID	Name	Total amount	Damage amount	Notification amount	Purchase price	Selling price	Manufacture	Demand	Supply Time	Discount
111	Milk 3%	25	4	60	4.5	8	Tnuva	10	2	-
222	Milk 1% (Yotvata)	75	2	50	4.5	8	Yotvata	10	3	2
112	Milk 1% (Tnuva)	40	0	60	9	16	Tnuva	20	3	-
333	Emek cheese	35	0	30	8.5	20	Emek	10	3	-
444	Steak Adom	30	1	35	50	100	Adom Adom	7	5	0

DAMAGED	
ID	Damage type
111-0	EXPIRED
111-1	EXPIRED
111-2	EXPIRED
111-3	EXPIRED
222-2	OTHER
222-3	OTHER
444-2	OTHER

DISCOUNT						
ID	Type	CategoryID/ ProductID	Start Date	End Date	Value	Damage Type
0	PRODUCT	444	01-04-23	01-05-2023	20%	-
1	CATEGORY	1-1-1	01-04-23	01-05-2023	2 NIS	-
2	DAMAGED	222	01-04-23	01-05-2023	3 NIS	OTHER

Branch 2

CATEGORY				
ID	Name	Sub-Categories	Products	Discounts
1	Vegetables and Fruits	Vegetables, Fruits	-	-
2	Drinks	Cola	-	-
1-1	Vegetables	Tomato, Cucumber	-	-
1-2	Fruits	Apple	-	-
1-1-1	Tomato		111, 222	1
1-1-2	Cucumber		112	-
1-2-1	Apple		333	-
2-1	Cola	1.5 liter	-	-
2-1-1	1.5 liter		444	-

PRODUCT										
ID	Name	Total amount	Damage amount	Notification amount	Purchase price	Selling price	Manufacture	Demand	Supply Time	Discount
555	Chery	25	0	30	4.5	8	Amit	20	3	-
666	Tamar	75	2	40	4.5	8	Amit	10	3	-
552	Mini	40	0	45	4	10	Gili	20	3	-
777	Pink lady	35	0	30	8.5	20	Tali	10	3	-
888	Coca cola	30	1	35	10	12	Coca Cola	7	5	0 , 2

DAMAGED	
ID	Damage type
666-2	OTHER
666-3	OTHER
888-2	BROKEN

DISCOUNT						
ID	Type	CategoryID/ ProductID	Start Date	End Date	Value	Damage Type
0	PRODUCT	888	01-04-23	01-05-2023	20%	-
1	CATEGORY	1-1-1	01-04-23	01-05-2023	2 NIS	-
2	DAMAGED	888	01-04-23	01-05-2023	3 NIS	BROKEN

מערכת הספקים:

בעת הפעלת המערכת ע"י הממשק הגרפי GUI יפתח החלון הבא:



באמצעות לחיצה על כל כפתור, ניתן להמשיך לביצוע הפעולה הרצויה.
לחיצה על כל כפתור, תפתח את חלון חדש, הרלוונטי להמשך ביצוע הפעולה הרצויה.

בעת הפעלת המערכת באמצעות ה-CUI יעלה התפריט הבא :

```
hey, welcome to the Suppliers System
please select your next action:
1 - add new Supplier
2 - remove supplier from the system
3 - look at Last order
4 - edit existing supplier
5 - load system data
6 - add products to supplier
7 - add new discount to supplier
8 - add new contact to supplier
9 - remove contact from supplier
10 - remove product from supplier
0 - exit from the system
```

סדר הזמנה ללא מידע – הוספת סניף -> הוספת ספק -> הוספת מוצר למערכת -> הוספת מוצר לספק -> הזמנת המוצר.

יש לבחור את האופציה הרצויה בעזרת הקשת מספרים בלבד – ולאחר להקיש ENTER.

לאחר בחירת הפעולה יעלה תפריט בהתאם לפעולה שנבחרה, יש לשים לב לפרטים הבאים ולבחור אותם על פי הנדרש.

יש להזין מספרים בלבד כקלט עבור כל קלט למעט הפרטים הבאים: מספר סידורי של פריט, שם ספק, כתובת ספק, מספר טלפון איש קשר ושם איש קשר (בפריטים אלה ניתן להזין גם אותיות כקלט).

לאחר כל פעולה, אם לא הודפס שהפעולה בוצעה בהצלחה, התרחשה שגיאה, ולא קרה שום שינוי במערכת.

בבחירת אופציה מס' 5 – טעינת מידע, יטען המידע הבא למערכת:

❖ יטענו ספקים מספר (1,2,3,4):

ספק מס' 1 מסוגל לספק את המוצרים הבאים:

- מוצר 111 בכמות 100 במחיר 7.9 ליחידה.
- מוצר 222 בכמות 200 במחיר 1.9 ליחידה.
- מוצר 444 בכמות 2400 במחיר 5.3 ליחידה.
- לספק זה קיימת הנחה על מוצר 111, בקניית יותר מ-10 פריטים ישנה הנחה בגובה 15 אחוז.
- לספק זה קיימת הנחה על מוצר 222, בקניית יותר מ-5 פריטים ישנה הנחה בגובה 18.3 אחוז.

- לספק זה קיימת הנחה על מוצר 444 , בקניית יותר מ-5 פריטים ישנה הנחה בגובה 18.3 אחוז.
- ספק זה מספק בימים קבועים: ראשון-שבת כולל.
- לספק זה קיימת הנחה עבור הזמנה כללית , בהזמנה מעל ל-10 שקלים תינתן 13 אחוז הנחה.

ספק מס' 2 מסוגל לספק את המוצרים הבאים:

- מוצר 111 בכמות 50 במחיר 14.5 ליחידה.
- מוצר 333 בכמות 600 במחיר 146.9 ליחידה.
- ספק זה לא יכול לספק את ההזמנות ממנו בעצמו.

ספק מס' 3 מסוגל לספק את המוצרים הבאים:

- מוצר 111 בכמות 100 במחיר 7.8 ליחידה.
- ספק זה מספק בימים קבועים: שישי ושבת.

ספק מס' 4 מסוגל לספק את המוצרים הבאים:

- מוצר 112 בכמות 200 במחיר 5 ליחידה.
- לספק זה אין ימי הספקה קבועים, הוא מספק לפי ההזמנות שהוא מקבל.

בעת הקשת 0 בתפריט הראשי ניתן לחזור לעליית המערכת ולבחור.

הוראות שימוש וסקיצה של הממשק הגרפי

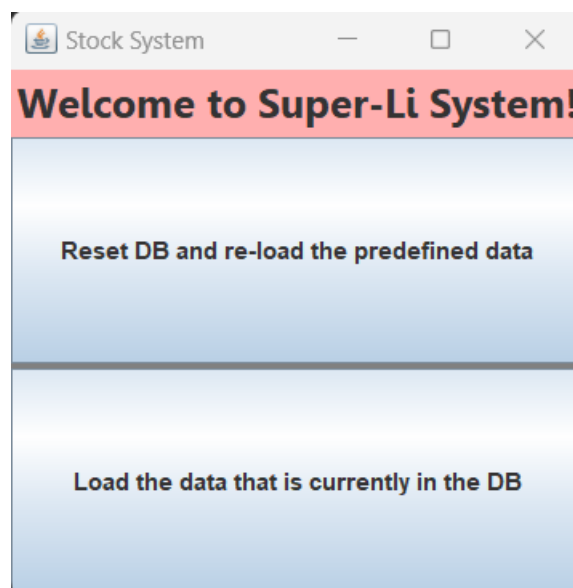
תרחיש C- עדכון מלאי והתראה על חוסרים

1. בשורת ההפעלה, יש להעביר את האחד מהארגומנטים הבאים למערכת:

`java -jar adss2023_v03.jar GUI StoreManager`

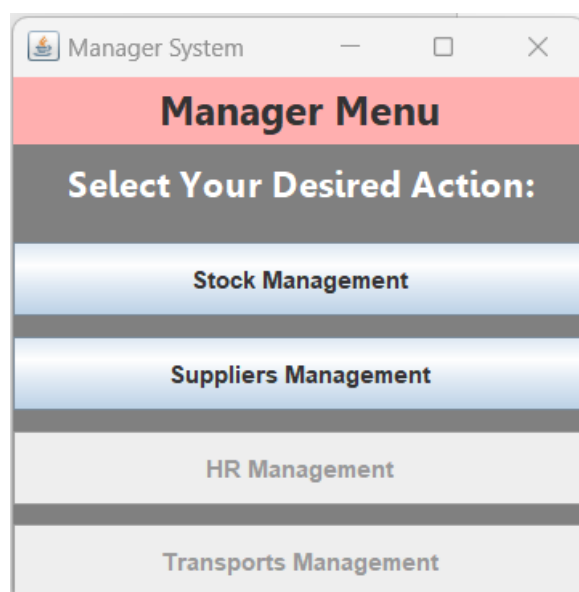
`java -jar adss2023_v03.jar GUI StockKeeper`

לאחר מכן ייפתח החלון של טעינת המידע:

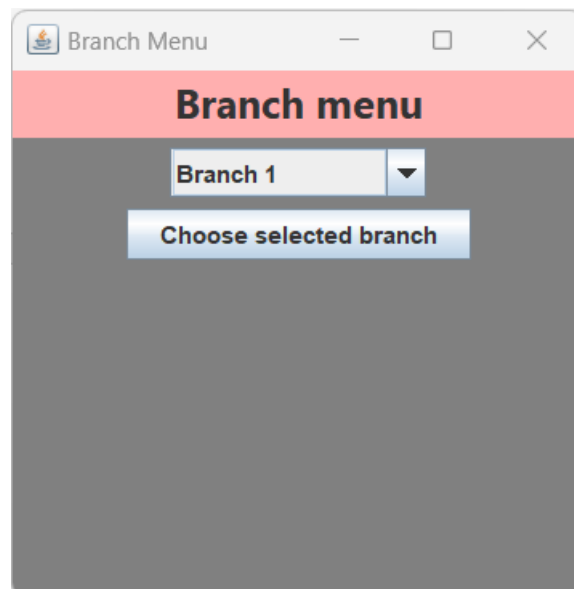


ניתן לבחור לאתחל את המידע ב-DB או לטעון את המידע הקיים בו כרגע.

2. בתפריט המנהל, יש לבחור ב- **Stock Management**



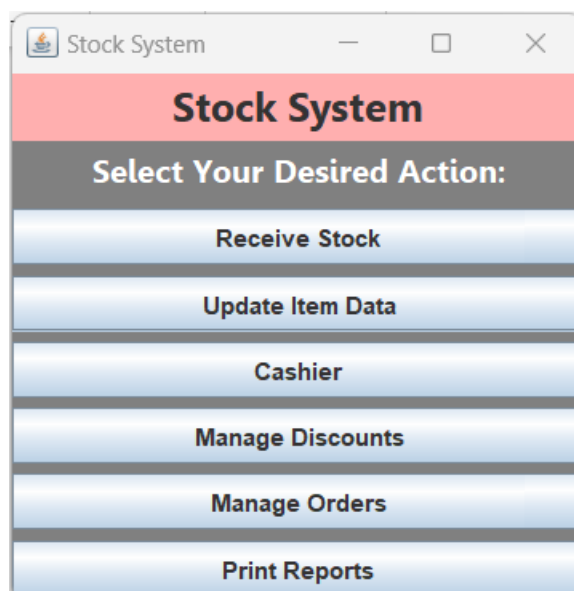
3. יש לבחור את הסניף הרצוי. לאחר הבחירה יופיע הכפתור "Choose selected branch" עליו יש ללחוץ.



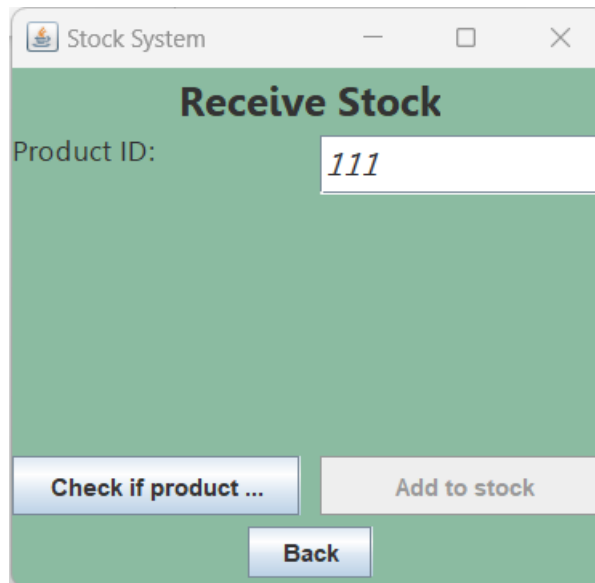
נבחין בין שני תהליכים של עדכון מלאי- רכישה והוספת סחורה.

הוספת סחורה

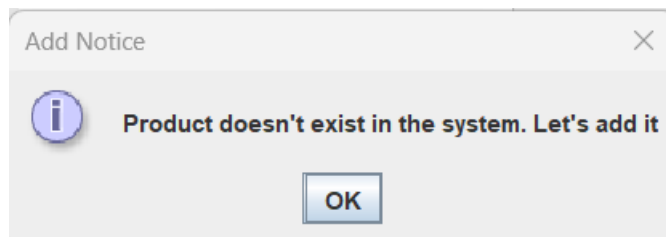
4. בתפריט המערכת, יש ללחוץ על כפתור "Receive Stock"



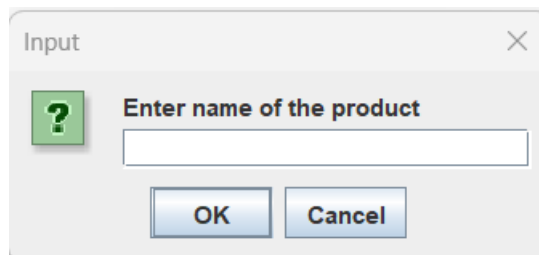
5. יש להקיש את המק"ט של המוצר וללחוץ על "Check if product exists"



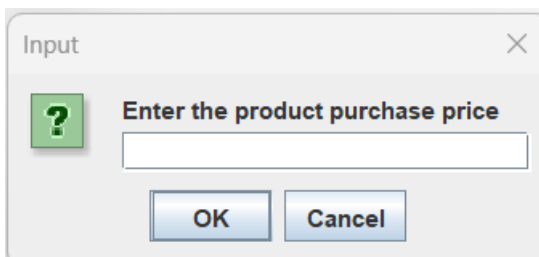
a. אם המוצר לא קיים במערכת, המערכת תתריע על כך בחלון הבא :



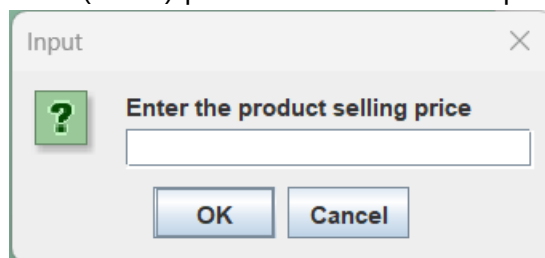
יש להוסיפו על ידי מילוי הפרטים בחלונות ולחיצה על "Ok" :
(1) הקלדת שם המוצר



(2) הקלדת מחיר הרכישה מהספק (מספר)

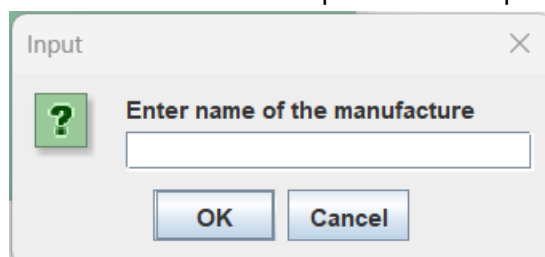


(3) הקלדת מחיר המכירה לצרכן (מספר)



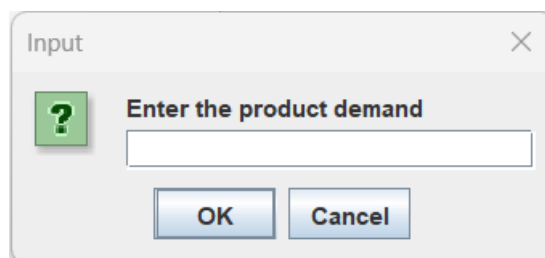
An input dialog box titled "Input" with a close button (X) in the top right corner. It contains a green square icon with a white question mark on the left. The text "Enter the product selling price" is displayed above a text input field. Below the input field are two buttons: "OK" and "Cancel".

(4) הקלדת שם היצרן



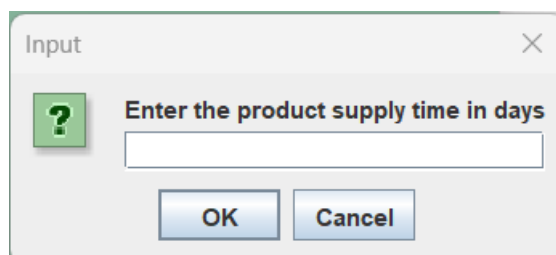
An input dialog box titled "Input" with a close button (X) in the top right corner. It contains a green square icon with a white question mark on the left. The text "Enter name of the manufacture" is displayed above a text input field. Below the input field are two buttons: "OK" and "Cancel".

(5) הקלדת הביקוש (מספר)



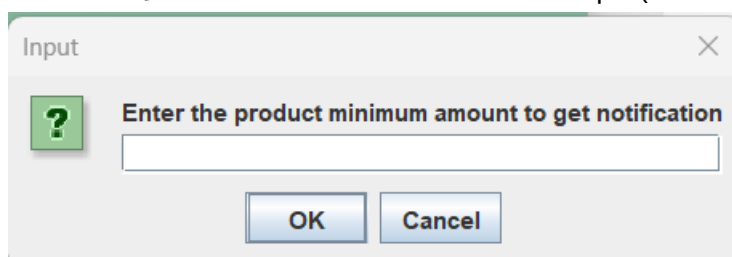
An input dialog box titled "Input" with a close button (X) in the top right corner. It contains a green square icon with a white question mark on the left. The text "Enter the product demand" is displayed above a text input field. Below the input field are two buttons: "OK" and "Cancel".

(6) הקלדת מספר ימי אספקה



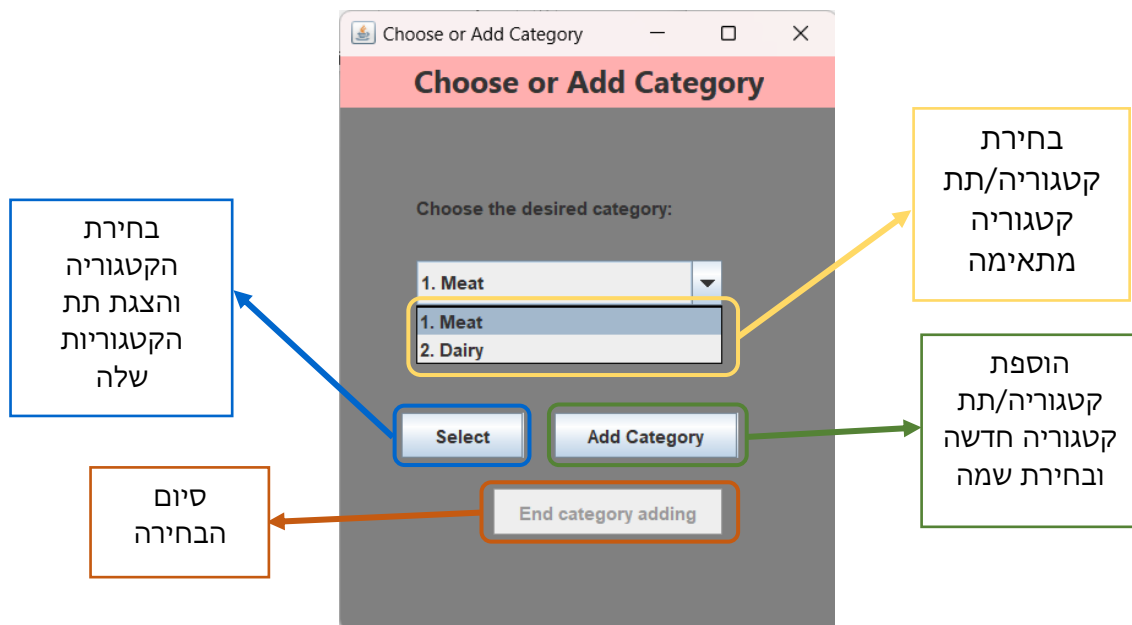
An input dialog box titled "Input" with a close button (X) in the top right corner. It contains a green square icon with a white question mark on the left. The text "Enter the product supply time in days" is displayed above a text input field. Below the input field are two buttons: "OK" and "Cancel".

(7) הקלדת מספר ימים מינימלי להתראה על חוסר

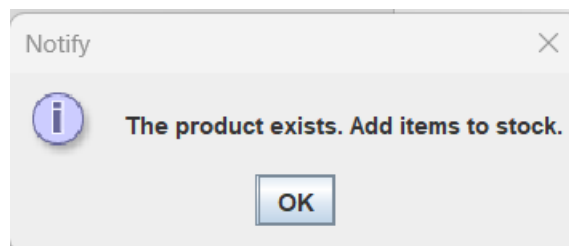


An input dialog box titled "Input" with a close button (X) in the top right corner. It contains a green square icon with a white question mark on the left. The text "Enter the product minimum amount to get notification" is displayed above a text input field. Below the input field are two buttons: "OK" and "Cancel".

(8) בחירה או הוספת קטגוריה מתאימה:



b. אם המוצר קיים במערכת/ההוספה הסתיימה, המערכת תתריע על כך :



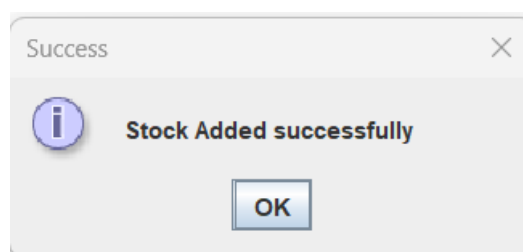
יש למלא את הפרטים על הסחורה שהגיעה- מספרים עבור הכמות לחנות ולמחסן, ותאריך בפורמט dd-mm-yyyy לתאריך התפוגה. לאחר מכן יש ללחוץ על הכפתור "Add to Stock"

The screenshot shows a window titled "Stock System" with a form titled "Receive Stock". The form has the following fields and values:

- Product ID: 111
- Store amount: 12
- Warehouse amount: 20
- Expiration date: 08-08-2024

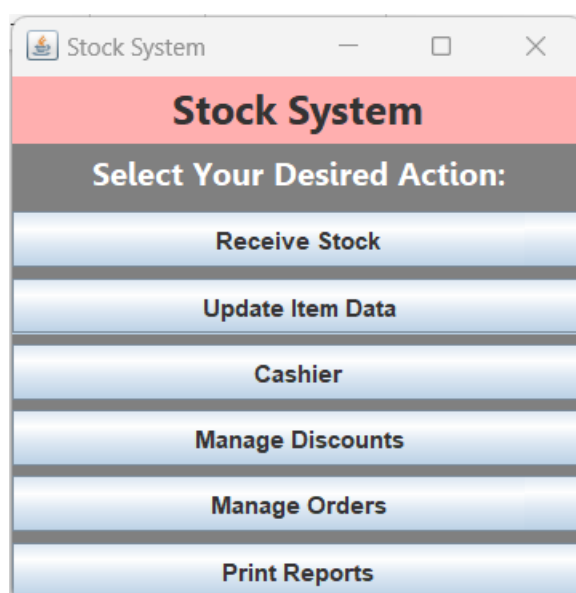
At the bottom, there are three buttons: "Check if product ...", "Add to stock", and "Back".

c. עם הצלחת קבלת הסחורה, יקפוץ החלון שיודיע על כך:

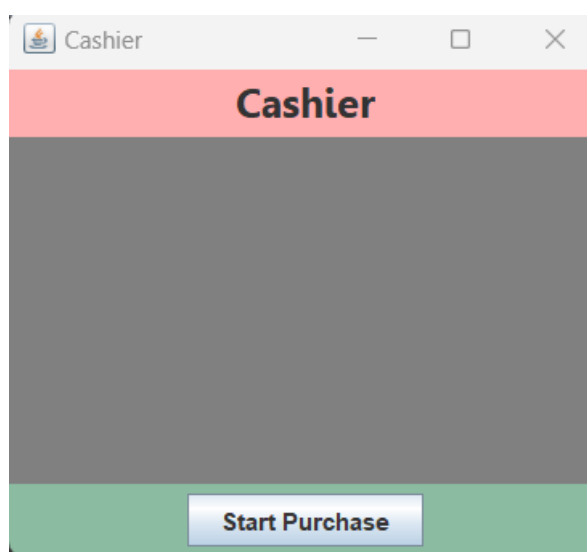


רכישה

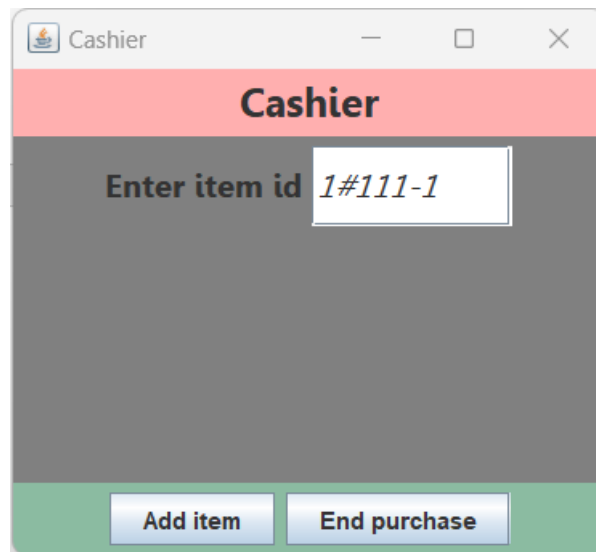
4. בתפריט המערכת, יש ללחוץ על כפתור "**Cashier**"



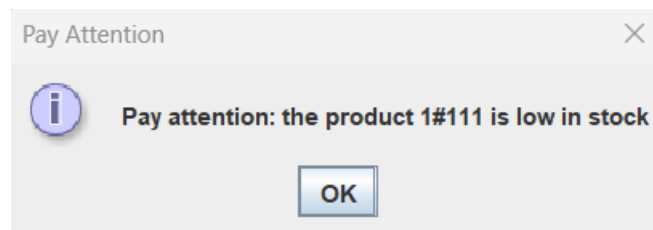
5. יש ללחוץ על "**Start Purchase**"



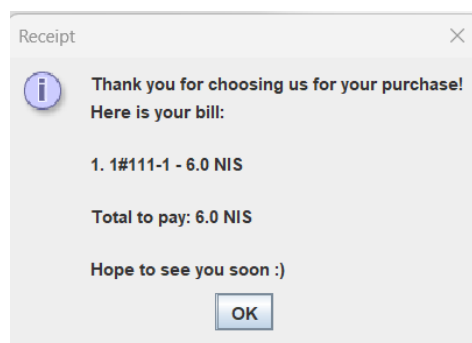
6. הקלד את המק"ט של הפריט שנרכש, הוסף לרכישה על ידי הכפתור "Add item". ניתן לחזור על פעולה זו עבור כל פריט שרוצים לרכוש תחת הקנייה הנוכחית. ניתן לסיים את הרכישה על ידי לחיצה על הכפתור "End purchase" :



אם המוצר שצורף לרכישה הגיע לכמות המינימלית שלו, קופצת ההתראה הבאה:



בסיום הרכישה תופיע קבלה :



תרחיש D – הוספת ספק חדש.

הוראות שימוש:

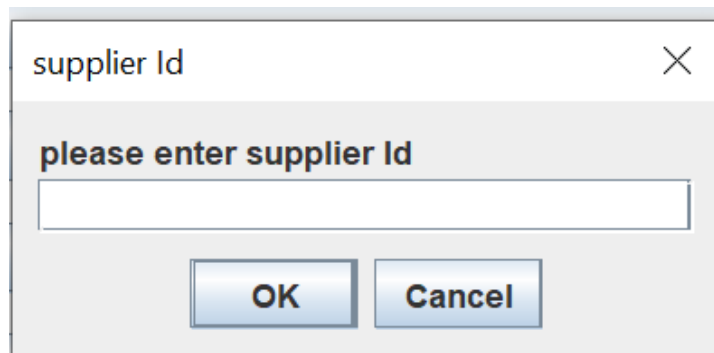
1. בשורת ההפעלה, יש להעביר את הארגומנט הבא למערכת:

```
java -jar adss2023_v03.jar GUI SupplierManager
```

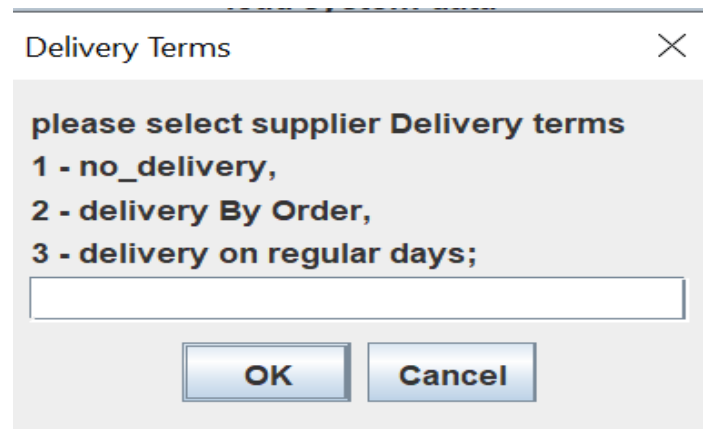
לאחר מכן יפתח החלון של המערכת לניהול ספקים:



2. יש ללחוץ על הכפתור "add new supplier", לאחר מכן יפתח החלון הבא:



3. יש להכניס את תעודת הזהות של הספק ולאחר מכן ללחוץ על הכפתור "ok".
יפתח החלון הבא:



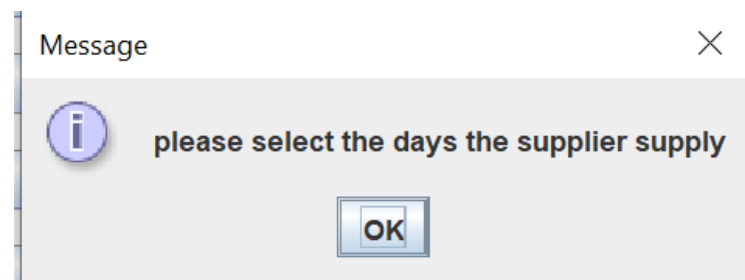
Delivery Terms

please select supplier Delivery terms

1 - no_delivery,
2 - delivery By Order,
3 - delivery on regular days;

OK Cancel

4. תבחרו את תנאי ההובלה המתאימים עבור הספק ע"י הכנסת המספר הנכון ולחצו על "ok". אם בחרתם באופציה מספר 3, יפתח החלון הבא:

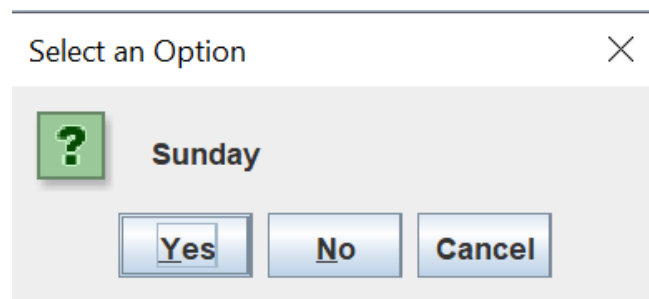


Message

please select the days the supplier supply

OK

כעת תצטרכו לבחור באיזה ימים קבועים בשבוע הספק שלכם עובד, אחרי לחיצה על "ok" יפתח:

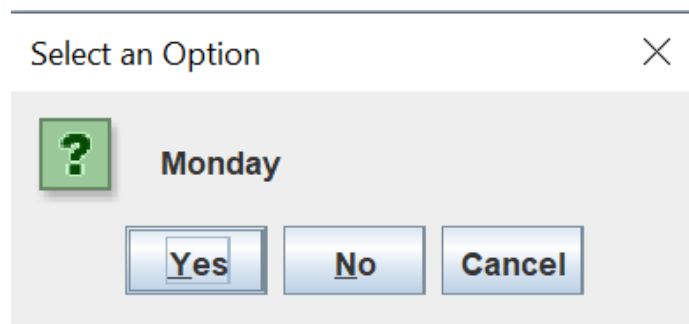


Select an Option

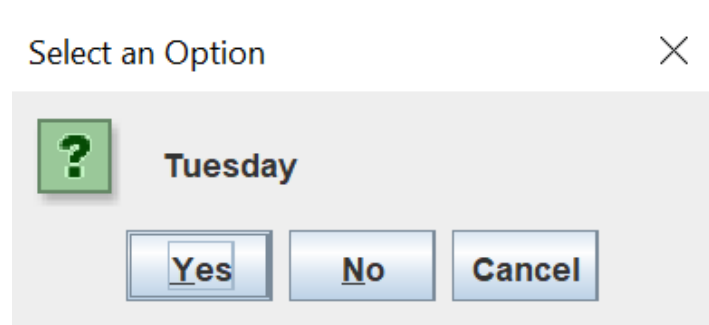
Sunday

Yes No Cancel

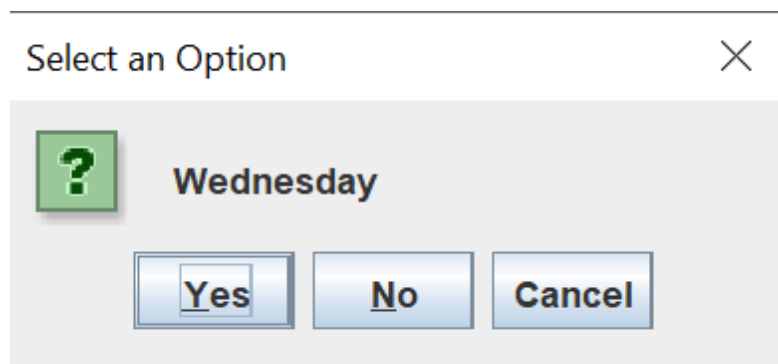
א. יש ללחוץ "yes" אם הספק יכול לספק בימי ראשון בשבוע ו"no" אחרת.
לאחר מכן יפתח החלון:



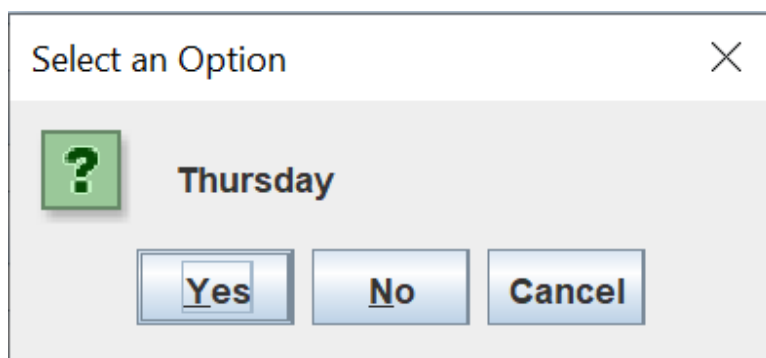
ב. יש ללחוץ "yes" אם הספק יכול לספק בימי שני בשבוע ו"no" אחרת.
לאחר מכן יפתח החלון:



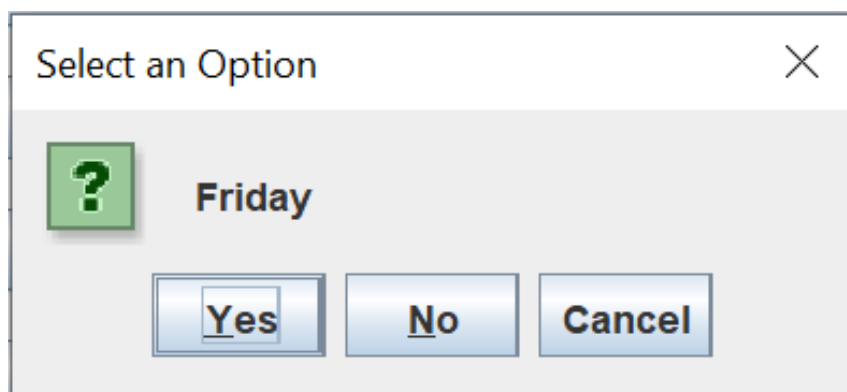
ג. יש ללחוץ "yes" אם הספק יכול לספק בימי שלישי בשבוע ו"no" אחרת.
לאחר מכן יפתח החלון:



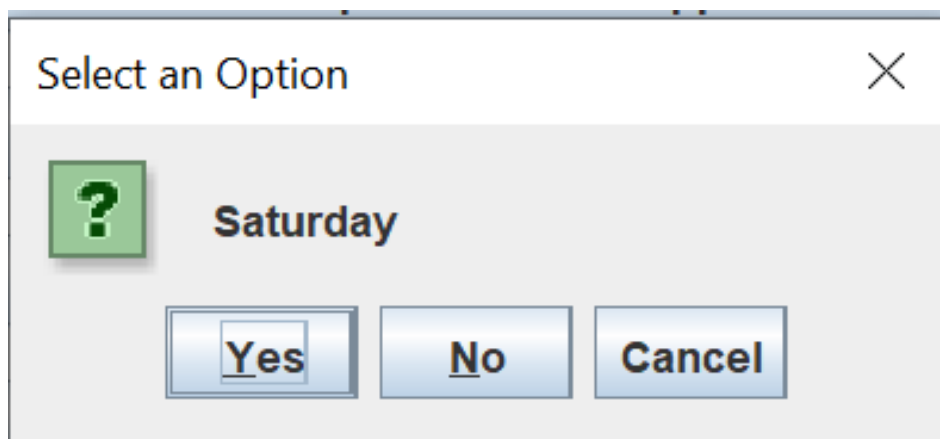
ד. יש ללחוץ "yes" אם הספק יכול לספק בימי רביעי בשבוע ו"no" אחרת.
לאחר מכן יפתח החלון:



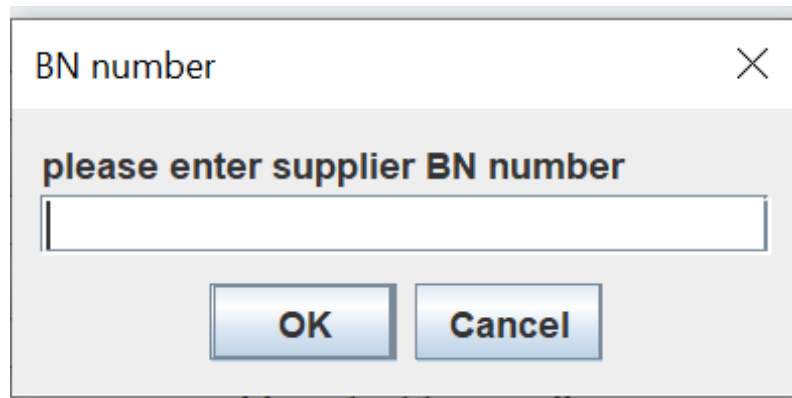
ה. יש ללחוץ "yes" אם הספק יכול לספק בימי חמישי בשבוע ו"no" אחרת.
לאחר מכן יפתח החלון:



ו. יש ללחוץ "yes" אם הספק יכול לספק בימי שישי בשבוע ו"no" אחרת.
לאחר מכן יפתח החלון:

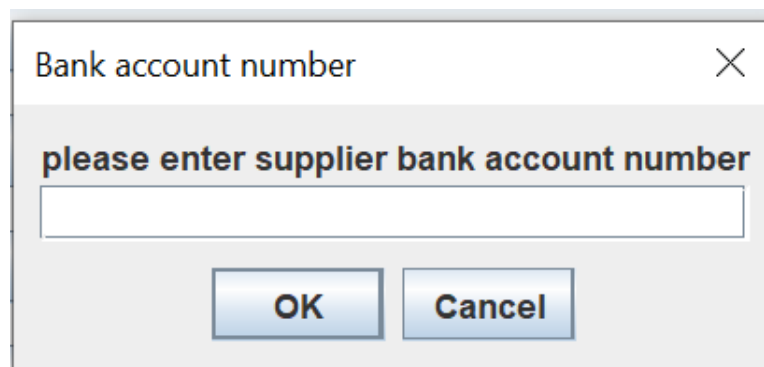


2. יש ללחוץ "yes" אם הספק יכול לספק בימי שבת בשבוע ו"no" אחרת.
לאחר מכן יפתח החלון שהיה נפתח אילו היינו באופציות 1 ו-2 בתנאי ההובלה של הספק:



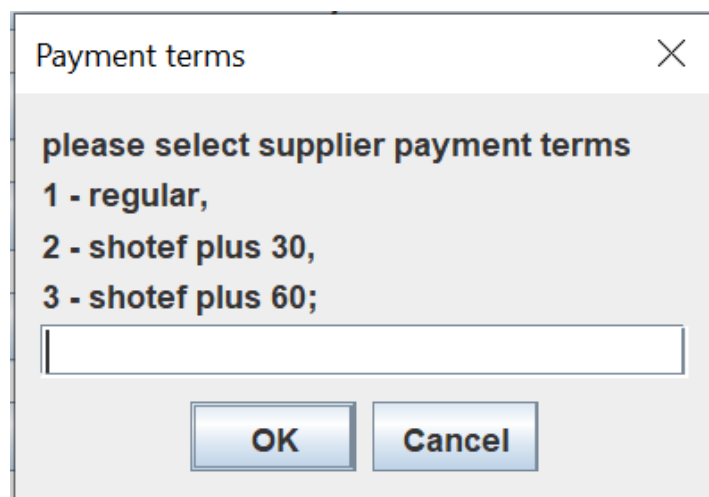
A dialog box titled "BN number" with a close button (X) in the top right corner. The main text inside the dialog is "please enter supplier BN number". Below the text is a single-line text input field. At the bottom of the dialog are two buttons: "OK" and "Cancel".

5. יש להזין מספר BN וללחוץ על הכפתור "ok", לאחר מכן יפתח החלון הבא:



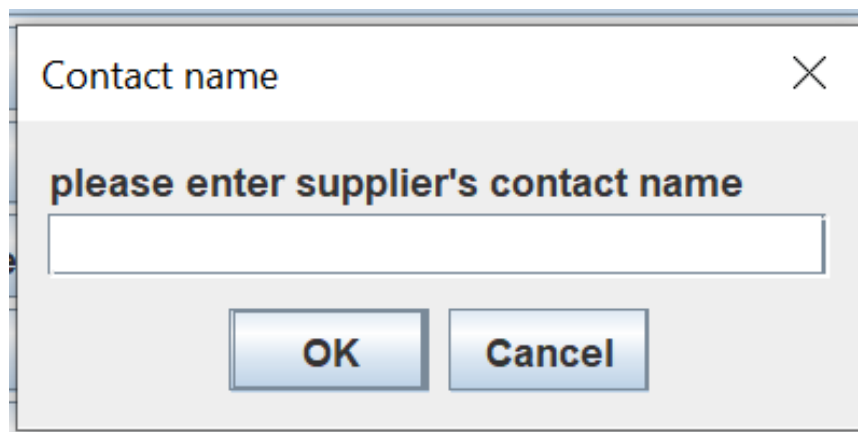
A dialog box titled "Bank account number" with a close button (X) in the top right corner. The main text inside the dialog is "please enter supplier bank account number". Below the text is a single-line text input field. At the bottom of the dialog are two buttons: "OK" and "Cancel".

6. יש להזין את מספר חשבון הבנק של הספק וללחוץ על הכפתור "ok", לאחר מכן יפתח החלון הבא:



A dialog box titled "Payment terms" with a close button (X) in the top right corner. The main text inside the dialog is "please select supplier payment terms". Below the text is a list of three options: "1 - regular,", "2 - shotef plus 30,", and "3 - shotef plus 60;". Below the list is a single-line text input field. At the bottom of the dialog are two buttons: "OK" and "Cancel".

7. יש לבחור את תנאי התשלום של הספק וללחוץ על הכפתור "ok", לאחר מכן יפתח החלון הבא:

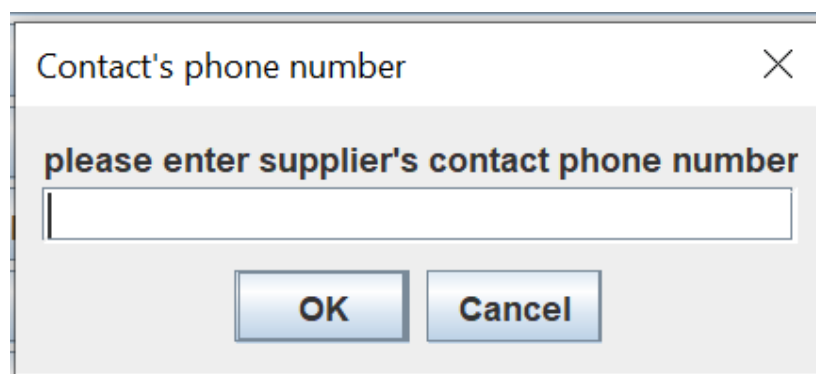


Contact name

please enter supplier's contact name

OK Cancel

8. יש להקליד את שם איש הקשר של הספק וללחוץ על הכפתור "ok", לאחר מכן יפתח החלון הבא:

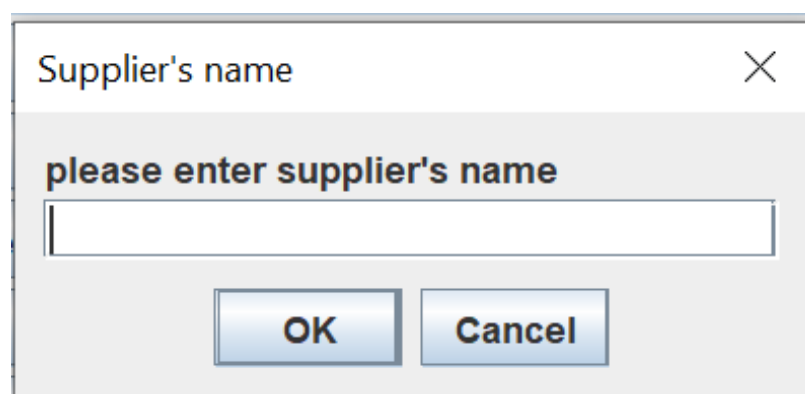


Contact's phone number

please enter supplier's contact phone number

OK Cancel

9. יש להזין את מס' הטלפון של איש הקשר וללחוץ על הכפתור "ok", לאחר מכן יפתח החלון הבא:

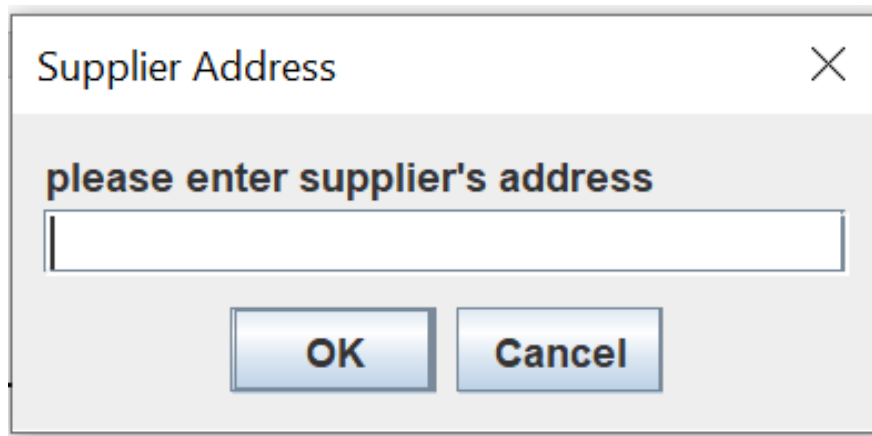


Supplier's name

please enter supplier's name

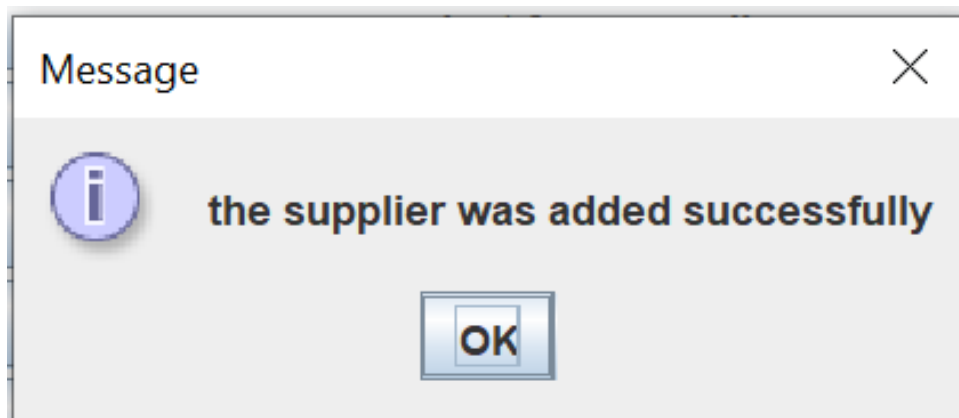
OK Cancel

10. יש להקליד את שם הספק וללחוץ על הכפתור "ok", לאחר מכן יפתח החלון הבא:



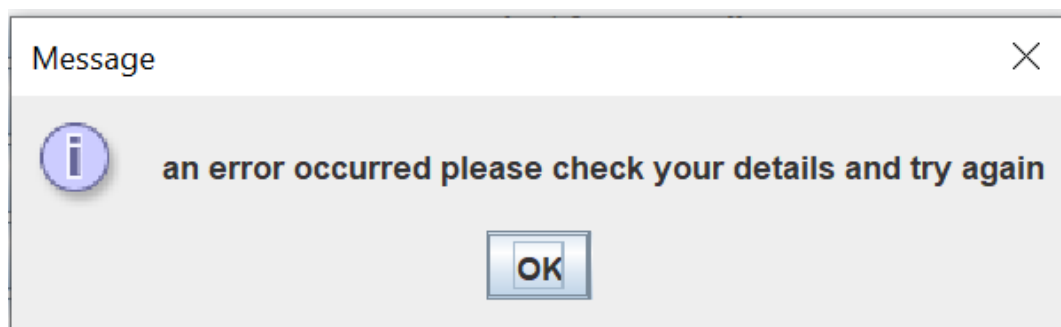
A dialog box titled "Supplier Address" with a close button (X) in the top right corner. The main area contains the text "please enter supplier's address" above a text input field. At the bottom, there are two buttons: "OK" and "Cancel".

11. יש להקליד את כתובת הספק וללחוץ על הכפתור "ok", לאחר מכן אם כל הפרטים היו תקינים והחננו בהצלחה, יפתח החלון הבא שיודיע על כך שהספק נוסף בהצלחה למערכת:



A message dialog box titled "Message" with a close button (X) in the top right corner. It features an information icon (i) on the left and the text "the supplier was added successfully" in the center. At the bottom, there is an "OK" button.

אחרת, במקרה שהייתה בעיה עם אחד הנתונים שהזן, יפתח חלק שיבשר שהייתה שגיאה במהלך הדרך ולא הייתה אפשרות להוסיף את הספק:



A message dialog box titled "Message" with a close button (X) in the top right corner. It features an information icon (i) on the left and the text "an error occurred please check your details and try again" in the center. At the bottom, there is an "OK" button.

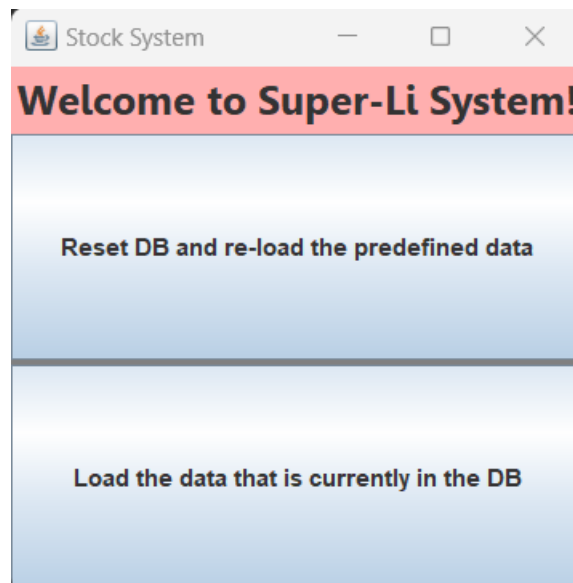
תרחיש E - הוצאת הזמנה תקופתית מספק

1. בשורת ההפעלה, יש להעביר את האחד מהארגומנטים הבאים למערכת:

java -jar adss2023_v03.jar GUI StoreManager

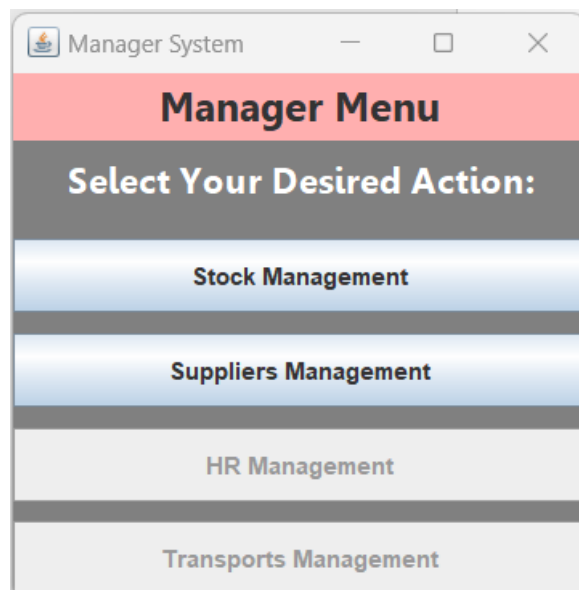
java -jar adss2023_v03.jar GUI StockKeeper

לאחר מכן ייפתח החלון של טעינת המידע:

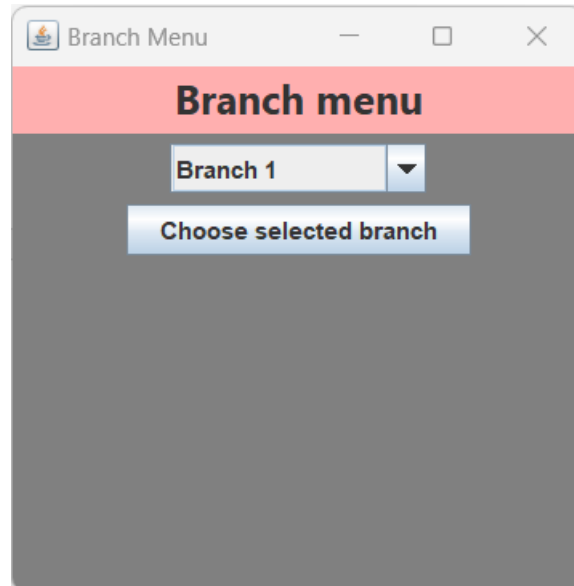


ניתן לבחור לאתחל את המידע ב-DB או לטעון את המידע הקיים בו כרגע.

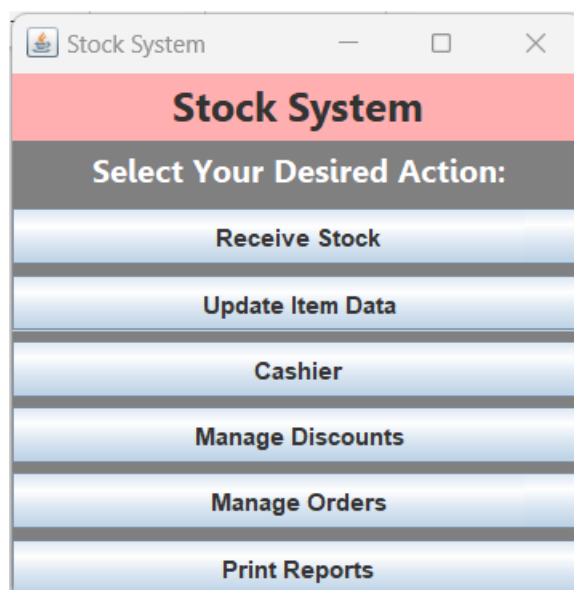
1. בתפריט המנהל, יש לבחור ב- **Stock Management**



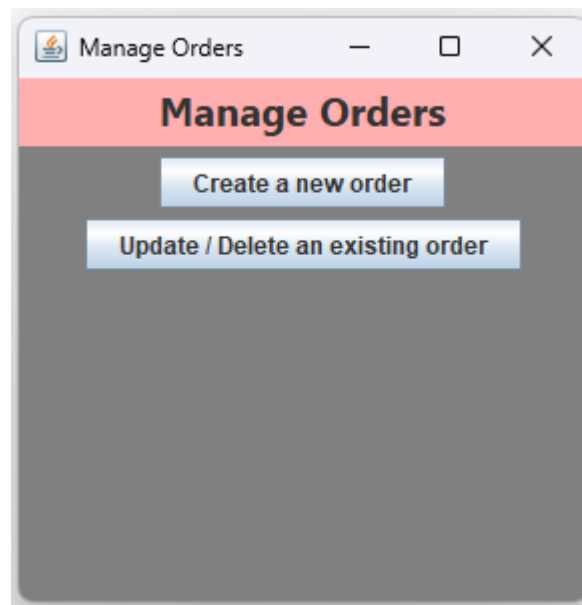
2. יש לבחור את הסניף הרצוי. לאחר הבחירה יופיע הכפתור "Choose selected branch" עליו יש ללחוץ.



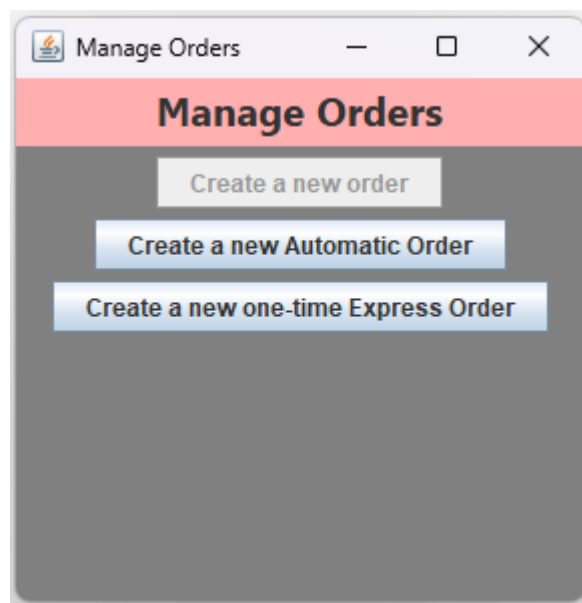
3. בתפריט המערכת, יש ללחוץ על כפתור "Manage Orders"



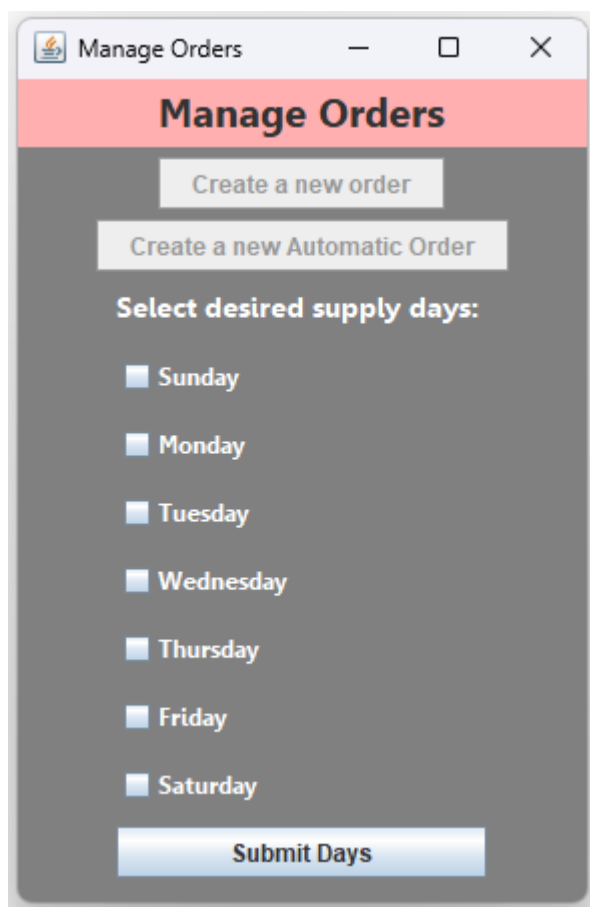
4. לאחר מכן יפתח תפריט ניהול ההזמנות, בו נלחץ על **"Create a new order"**.



5. מתוך 2 סוגי ההזמנות שהמערכת שלנו מאפשרת, נבחר ב" **Create a new Automatic Order**" על מנת ליצור הזמנה תקופתית חדשה.

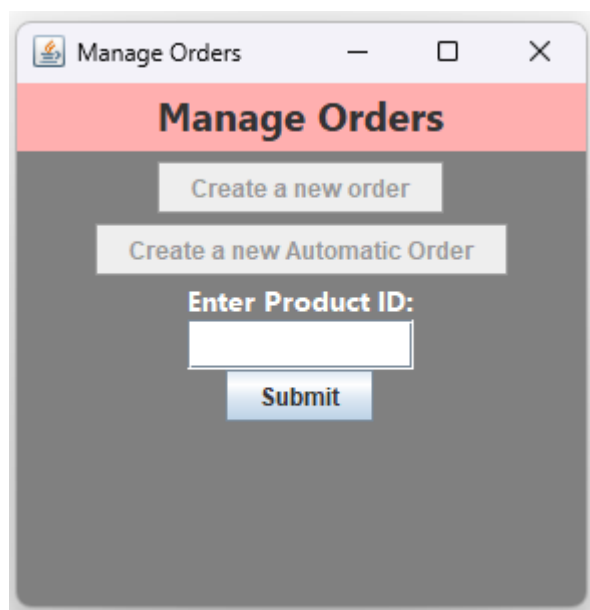


6. נסמן את ימי השבוע בהם נרצה לקבל את ההזמנה החדשה, ונלחץ על **Submit Days**.



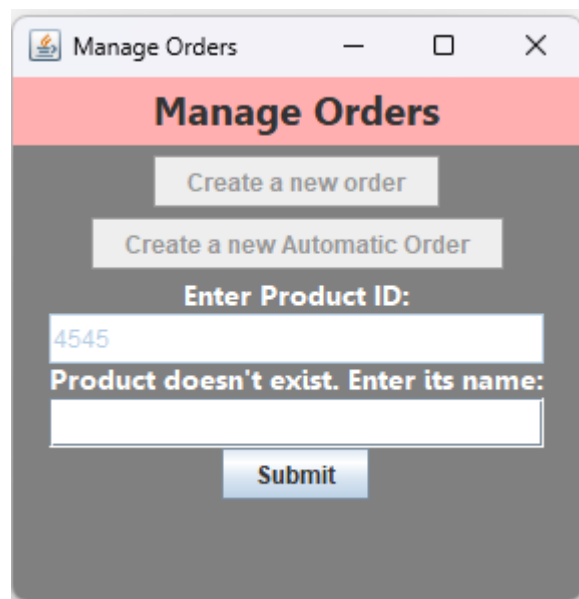
The screenshot shows a web browser window titled "Manage Orders". The page has a red header with the title "Manage Orders". Below the header, there are two buttons: "Create a new order" and "Create a new Automatic Order". Underneath these buttons, the text "Select desired supply days:" is displayed. Below this text, there is a list of days of the week, each with a checkbox: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. At the bottom of the form, there is a blue button labeled "Submit Days".

7. נכניס את המק"ט של המוצר הראשון אותו נרצה לרכוש, ונלחץ על **Submit**.



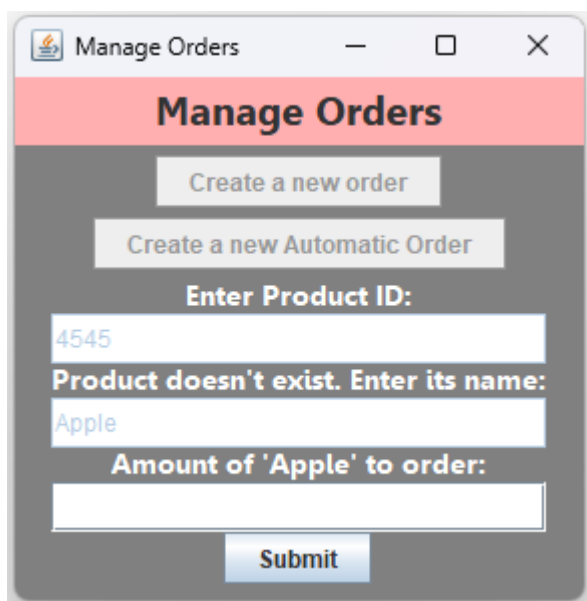
The screenshot shows the same "Manage Orders" web browser window. The buttons "Create a new order" and "Create a new Automatic Order" are still visible. Below them, the text "Enter Product ID:" is displayed. Underneath this text, there is a white text input field. Below the input field, there is a blue button labeled "Submit".

8. במידה והמערכת מזהה את המק"ט – נדלג ישירות לשלב 10. במידה ומק"ט זה אינו מזהה במערכת, משמע מוצר זה לעולם לא התקבל בסניף זה, יפתח החלון הבא. נזין את שמו של המוצר, ונלחץ על "Submit".

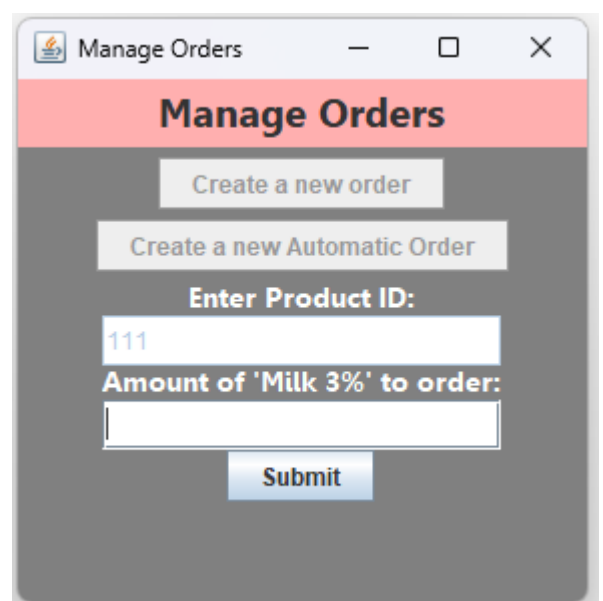


The screenshot shows a window titled "Manage Orders" with a red header. Below the header are two buttons: "Create a new order" and "Create a new Automatic Order". Under these is a label "Enter Product ID:" followed by a text input field containing "4545". Below the input field is a message: "Product doesn't exist. Enter its name:". Under this message is another empty text input field. At the bottom is a "Submit" button.

9. נזין את כמות המוצרים שנרצה לקבל בכל הזמנה שמגיעה לחנות, ונלחץ על "Submit".



The screenshot shows the same "Manage Orders" window. The "Enter Product ID:" field still contains "4545". The message "Product doesn't exist. Enter its name:" is still present. The text input field below it now contains "Apple". Below this is a new label: "Amount of 'Apple' to order:". Under this label is an empty text input field. The "Submit" button remains at the bottom.

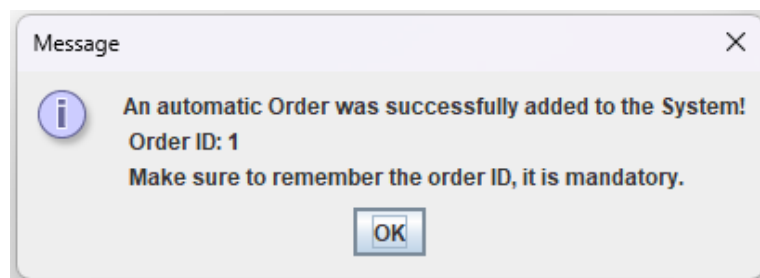


The screenshot shows the same "Manage Orders" window. The "Enter Product ID:" field now contains "111". The message "Product doesn't exist. Enter its name:" is no longer present. The text input field below it now contains "Amount of 'Milk 3%' to order:". Below this is an empty text input field. The "Submit" button remains at the bottom.

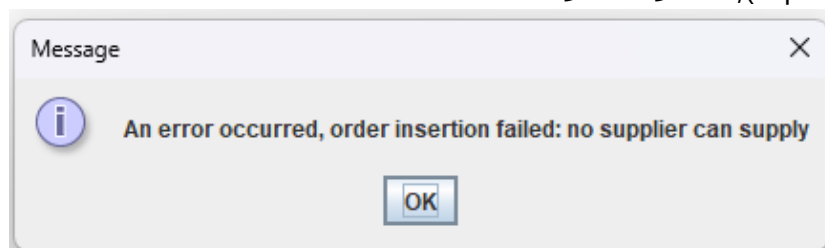
10. בשלב זה, יהיה עלינו להחליט האם סיימנו להזין את כלל המוצרים שברצוננו להוסיף להזמנה החדשה, או שברצוננו להזין מוצרים נוספים.
- במידה וסיימנו, נלחץ על **"YES – Submit Order"**, ונמשיך לשלב 12.
 - במידה וברצוננו להוסיף מוצר נוסף, נלחץ על **"NO – Add another product"**, ונחזור לשלב 8. כך יחזור חלילה עד שנחליט שסיימנו, ונבחר באופציה הראשונה.

11. לאחר שליחת ההזמנה, ההזמנה תועבר למערכת הספקים, ותתקבל ממנה אחת מ-2 תשובות אפשריות.

- אם ההזמנה התקבלה בהצלחה – תופיע ההודעה הבאה:



- אחרת, במידה ולא ניתן לספק את ההזמנה (על פי התנאים שהוגדרו בדרישות הלקוח), תופיע ההודעה הבאה:



לאחר השלמת תהליך הזנת הזמנה תקופתית חדשה על ידי אחד מעובדי המלאי בחנות, מתחיל תהליך אוטומטי במערכת הספקים שמטרתו למצוא את הספק המתאים ביותר עבור ההזמנה.

במסגרת התהליך מבוצעת בדיקה של איזה ספקים יכולים לספק את ההזמנה ביום הנדרש, ובנוסף ברשותם כלל המוצרים הנדרשים עבור השלמת ההזמנה. מתוך הספקים הללו, המערכת תבחר בספק המציע את המחיר המשתלם ביותר עבור החנות, ותשייך את ההזמנה לספק הנבחר.

במערכת הספקים מתרחש אחת ליממה תהליך אוטומטי. במסגרת תהליך זה, מתבצע מעבר על כלל ההזמנות התקופתיות במערכת, ושליחה של ההזמנות שמתחזמנות ל-24 שעות הקרובות לספקים הרלוונטיים (כלומר יצירת instance של constOrder).

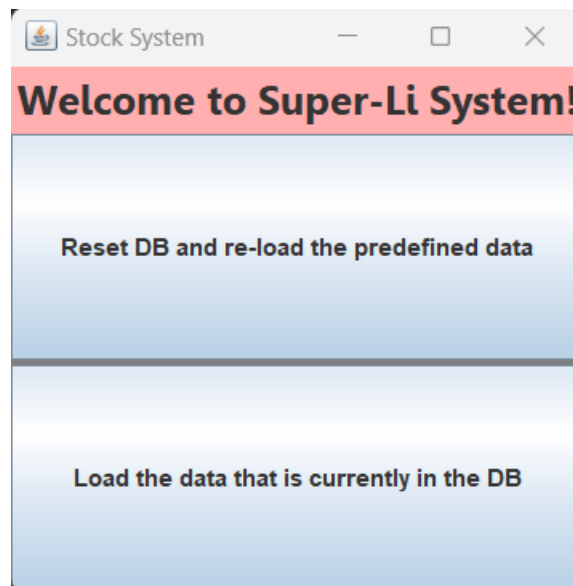
תרחיש F - הוצאת הזמנה מספק עקב חוסר.

1. בשורת ההפעלה, יש להעביר את האחד מהארגומנטים הבאים למערכת:

java -jar adss2023_v03.jar GUI StoreManager

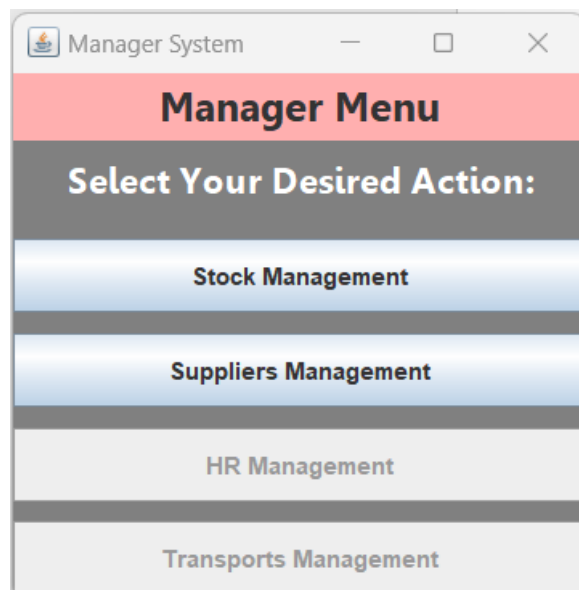
java -jar adss2023_v03.jar GUI StockKeeper

לאחר מכן ייפתח החלון של טעינת המידע:

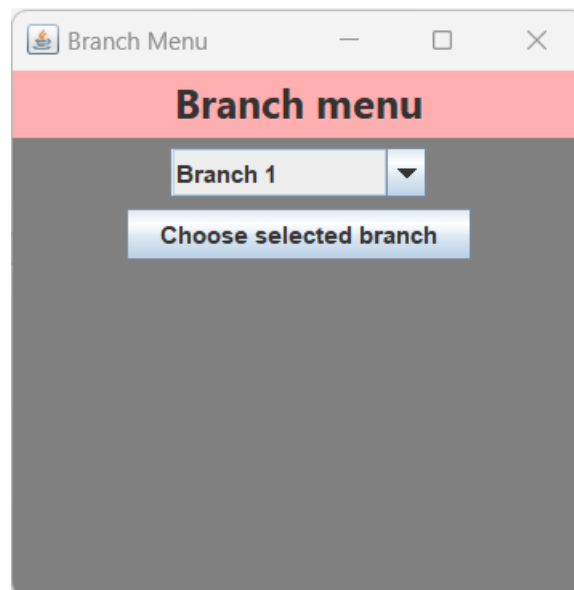


ניתן לבחור לאתחל את המידע ב-DB או לטעון את המידע הקיים בו כרגע.

2. בתפריט המנהל, יש לבחור ב- **Stock Management**

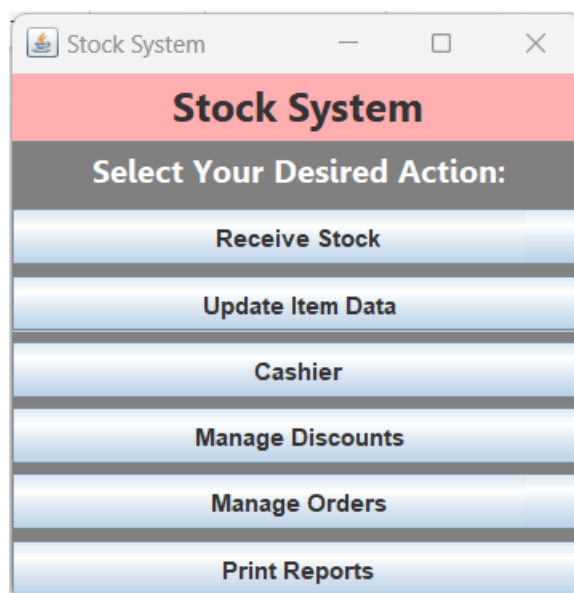


3. יש לבחור את הסניף הרצוי. לאחר הבחירה יופיע הכפתור "Choose selected branch" עליו יש ללחוץ.

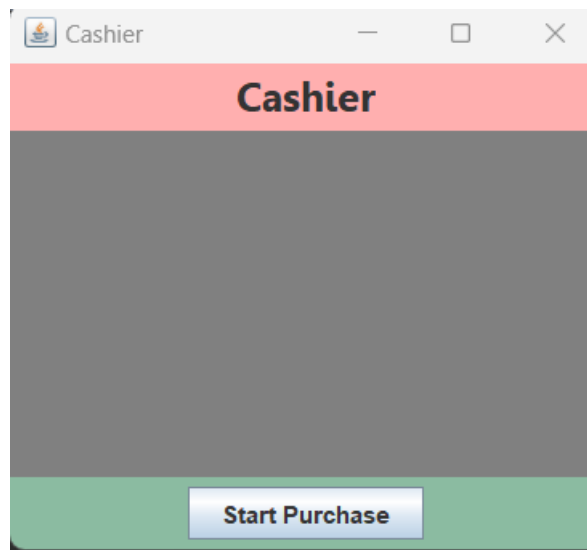


כעת נבצע רכישה שתגרום לחוסר, ובעקבותיה תתבצע שליחת הזמנה עקב חוסר לספק.

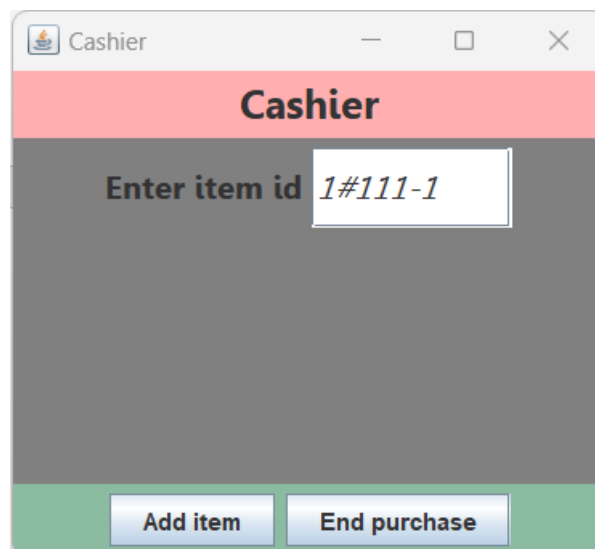
4. בתפריט המערכת, יש ללחוץ על כפתור "Cashier"



5. יש ללחוץ על "Start Purchase"



6. נקליד מק"ט של פריט שנרצה לרכוש, ונלחץ על "Add item".
נעשה זאת כל עוד נרצה להוסיף פריטים נוספים לרכישה.



כאשר מוצר שצורף לרכישה הגיע לכמות המינימלית שלו, תוצג ההתראה הבאה.



7. לסיום הרכישה - נלחץ על "End Purchase" (בתפריט שלב 6), ותופיע קבלה.

כעת, רק לאחר השלמת הרכישה, תבוצע הוצאת הזמנה מספק עקב חוסר. כל מוצר שכמותו ירדה תחת רף הכמות המינימלית (במסגרת ביצוע הרכישה הנוכחית) יתווסף להזמנה זו. ההזמנה תועבר למערכת הספקים שתדאג לשליחת ההזמנה לספק/ים הטוב/ים ביותר לצורך קבלת הסחורה מהר ככל הניתן.

מסמך דרישות קבוצה S – מלאי ספקים

מס"ד	מודל	פונקציונלי / לא פונקציונלי	תיאור	עדיפות	סיכון	סטטוס
1	מלאי	פונקציונלי	המערכת תקבע עבור כל מוצר כמות מינימלית שונה להתראה התלויה בזמן האספקה ובביקוש המוצר	MH	Low	בוצע
2	מלאי	פונקציונלי	המערכת תתריע כאשר כמות מוצר במלאי קטנה מן הכמות המינימלית שנקבעה עבורו	MH	Low	בוצע
3	מלאי	פונקציונלי	המערכת תאפשר הפקת רשימת חוסרים בהתאם למוצרים עבורם התקבלה התראה על כמות קטנה מהמינימלית	MH	High	בוצע
4	מלאי	פונקציונלי	המערכת תאפשר ניהול מידע אודות המוצרים הקיימים במלאי – שם המוצר, יצרן, כמות נוכחית, כמות במדפים, כמות במחסן וכמות מינימלית להתראה	MH	Low	בוצע
5	מלאי	פונקציונלי	המערכת תאפשר ניהול מידע אודות הפריטים הקיימים במלאי מכל מוצר – מק"ט הפריט, מיקומו, תוקף, האם תקין למכירה, סוג פגם	MH	Low	בוצע
6	מלאי	פונקציונלי	המערכת תאפשר ניהול מידע אודות ההיסטוריה של מוצרי החנות - מחיר העלות לכל פריט והמחיר בו הוא נמכר	NTH	Low	בוצע
7	מלאי	פונקציונלי	המערכת תאפשר ניהול מבצעים שונים בחנות – מוצר / קטגוריה, אחוזי הנחה, תאריכי המבצע	MH	Low	בוצע
8	מלאי	פונקציונלי	המערכת תסווג את הפריטים השונים לפי קטגוריות, תתי קטגוריות, ותתי תתי קטגוריות	MH	High	בוצע

9	מלאי	פונקציונלי	על המערכת לתמוך בהפקת דוחות מלאי שונים על פי קטגוריות לבחירת ההנהלה	MH	High	בוצע
10	מלאי	פונקציונלי	המערכת תאפשר הפקה אוטומטית של דו"חות פריטים פגומים בתדירות שתקבע על ידי ההנהלה	MH	Low	לא בוצע
11	מלאי	פונקציונלי	המערכת תאפשר הפקה של רשימות פריטים פגומים או פגי תוקף על פי בקשה	MH	High	בוצע
12	מלאי	פונקציונלי	המערכת תאפשר לבצע רכישה של מוצר/ים עבור המחיר התקף הנמוך ביותר לכל מוצר	NTH	High	בוצע
13	מלאי	פונקציונלי	המערכת תאפשר הפקת 2 סוגי הזמנות באופן אוטומטי: הזמנה תקופתית, והזמנה עקב חוסר	MH	High	בוצע
14	מלאי	פונקציונלי	המערכת תוודא הזמנת הכמות הנדרשת כך שרמת המלאי הצפויה בהגעת הזמנה תעלה על הכמות המינימלית המוגדרת עבור כל מוצר	MH	High	בוצע
15	מלאי + ספקים	פונקציונלי	המערכת תאפשר לעדכן הזמנה תקופתית עד 24 שעות לפני מועד האספקה שלה	NTH	High	בוצע
16	מלאי + ספקים	פונקציונלי	המערכת תפיק הזמנה עקב חוסר לפי: פרטי המוצר החסר, כמות נדרשת	NTH	Low	בוצע
17	ספקים	פונקציונלי	על המערכת לאפשר הוספת ספקים חדשים למערכת.	MH	Low	בוצע
18	ספקים	פונקציונלי	על המערכת לתמוך בפתיחת כרטיס ספק לכל ספק שמתווסף למערכת.	MH	Low	בוצע
19	ספקים	פונקציונלי	על המערכת לאפשר הסרת ספקים קיימים מהמערכת.	MH	Low	בוצע
20	ספקים	פונקציונלי	על המערכת לשמור את הפרטים הבאים בכל כרטיס ספק: מספר ח"פ, חשבון בנק, תנאי תשלום (שוטף, שוטף+30 או שוטף +	MH	Low	בוצע

			60) , שמות אנשי קשר ופרטי התקשרות עימם.			
21	ספקים	פונקציונלי	על המערכת לאפשר עריכת פרטי ספקים קיימים במערכת – כמו (תנאי משלוח, תנאי תשלום , חשבון בנק, שם ספק , חשבון BN)	MH	High	בוצע
22	ספקים	פונקציונלי	על המערכת לשמור עבור כל ספק- אילו פריטים הוא מסוגל מספק.	MH	Low	בוצע
23	ספקים	פונקציונלי	על המערכת להיות מסוגלת למצוא עבור כל מוצר מאיזה ספק משתלם ביותר להזמינו מתוך הספקים הקיימים במערכת.	MH	High	בוצע
25	ספקים	פונקציונלי	על המערכת לשמור עבור כל ספק האם ימי האספקה קבועים או לא.	MH	Low	בוצע
26	ספקים	פונקציונלי	על המערכת לשמור עבור כל ספק האם הוא מספק את מוצריו בעצמו או לא.	MH	Low	בוצע
27	ספקים	פונקציונלי	על המערכת לשמור לכל פריט בהסכם את מחירו.	MH	Low	בוצע
28	ספקים	פונקציונלי	המערכת תאפשר שמירת כתב כמויות , תשמור את ההנחה עבור מוצר וכמות באם קיימת הנחה כזו.	MH	Low	בוצע
29	ספקים	פונקציונלי	על המערכת לשמור עבור כל פריט וספק מהו המספר הקטלוגי של הפריט מאותו ספק.	MH	Low	בוצע
31	ספקים	פונקציונלי	על המערכת לשמור עבור כל ספק את היסטוריית ההזמנות ממנו.	MH	Low	בוצע
32	ספקים	לא פונקציונלי	על המערכת לשמור בבסיס נתונים את אודות הספקים, מוצרים, הזמנות שבמערכת.	MH	High	בוצע
34	מלאי + ספקים	פונקציונלי	על המערכת לתמוך בהוספת הזמנות תקופתיות חדשות	MH	High	בוצע
35	ספקים + מלאי	פונקציונלי	על המערכת לתמוך בהוספת הזמנות חדשות עקב מחסור בחנות.	MH	High	בוצע

36	ספקים	פונקציונלי	על המערכת לשייך להזמנה תקופתית את הספקים הרלוונטיים לפי סדר החשיבות הבא: הזמנה מספק יחיד, ההזמנה הזולה ביותר.	MH	Low	בוצע
37	ספקים	פונקציונלי	על המערכת לשייך להזמנה תקופתית את הספק הרלוונטי כך שתגיע במהירות המרבית	MH	Low	בוצע
38	ספקים	פונקציונלי	על המערכת לשמור לכל ספק את כתובתו	MH	Low	בוצע
39	מלאי + ספקים	לא פונקציונלי	על המערכת לתמוך בגישה למערכת באמצעות GUI ו-CUI	MH	Low	בוצע
40	מלאי + ספקים	לא פונקציונלי	המערכת תתמוך בהגבלת הרשאות גישה עבור בעלי תפקידים שונים.	MH	Low	בוצע

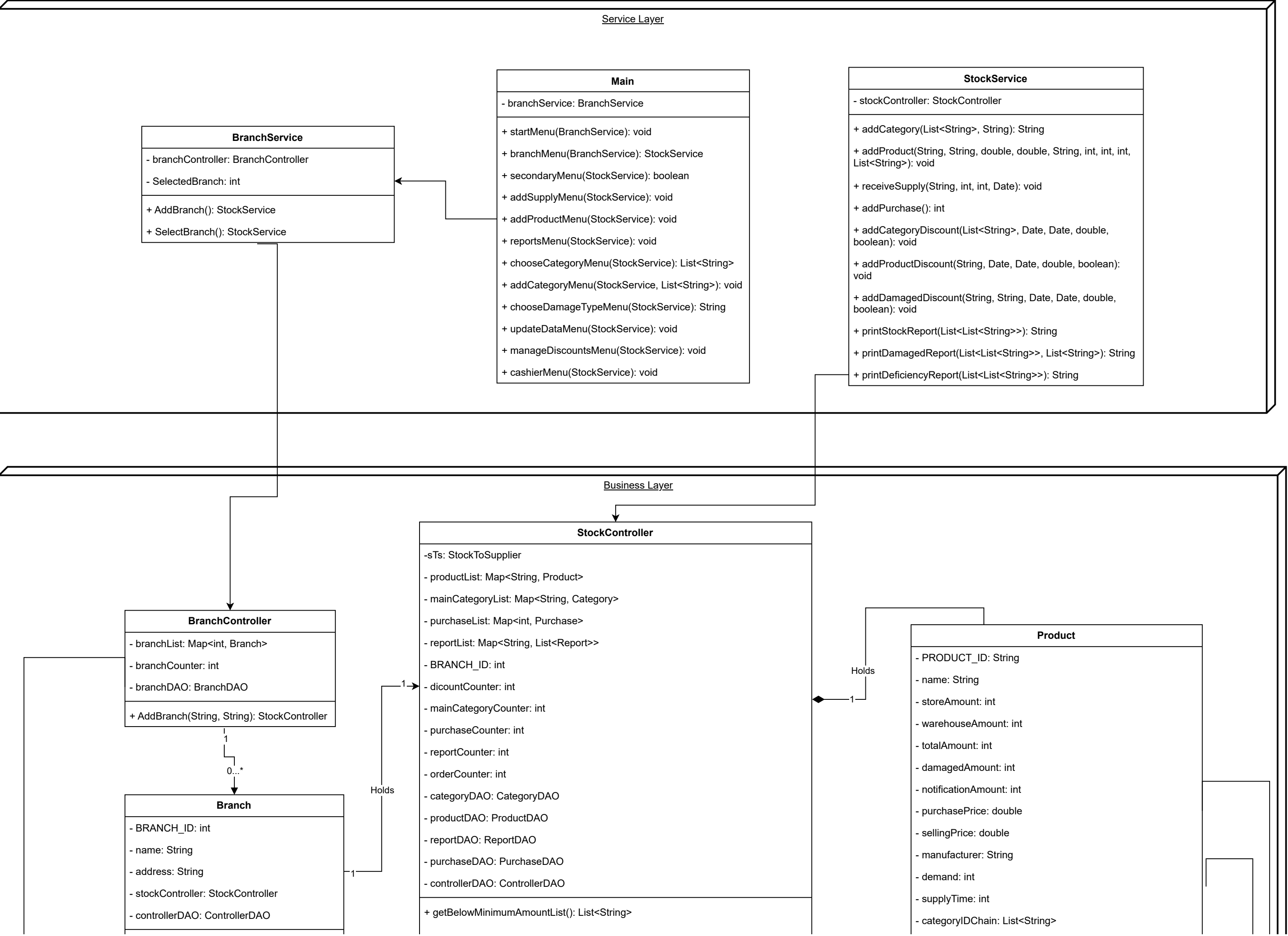
שינויים שבוצעו מהמטלה הקודמת- מלאי וספקים:

1. התווסף הממשק גרפי GUI למערכת התומך בכל פונקציונליות המערכת.
2. כתוצאה מהוספת ה-GUI, ישנה אופציה במערכת לבחור בין הרצת המערכת באמצעות GUI לבין הרצת המערכת באמצעות CUI.
3. המערכת כעת תומכת באכיפת הגבלת הרשאות הגישה, כלומר כל בעל תפקיד, יכול לגשת במערכת כעת רק לפונקציות המערכת הרלוונטיות לתפקידו.
4. בוצעו התיקונים הרלוונטיים בטעינת הנתונים מה-DATA BASE למערכת: כעת ישנן 2 אפשרויות - טעינת הנתונים הקיימים ב-DB בעת עליית המערכת, או איפוס כלל נתוני ה-DB וטעינת הנתונים לדוגמה שהוגדרו מראש.
5. בוצעו התיקונים הרלוונטיים בתרחישי השימוש: הספק כבר לא מופיע כשחקן, תנאי הקדם ותנאי הסיום הם כבר לא חלק מתסריט הצלחה ראשי ובנוסף תנאי סיום הם מצב המערכת לאחר תסריט ההצלחה הראשי.

הגדרת מושגים

- ספק - הספק הוא חיצוני למערכת, תפקידו לספק מוצרים לסופר - לי, החברה עובדת עם ספקים שונים עבור תחומים שונים. ישנם ספקים שמגיעים בימים קבועים, ספקים שמגיעים כאשר ישנה הזמנה בלבד וספקים שלא מספקים הובלה אלא איסוף עצמי. כל ספק מייצג חברות שונות בשוק. יתכן מצב ובו מספר ספקים שונים מספקים מוצרים של אותו היצרן.
- הזמנה - הזמנה מתבצעת דרך ספק מסוים והיא מתחלקת לשני סוגים:
 1. הזמנה עקב חוסר מוצרים - כאשר יש מחסור במוצר בסופר, ניתן לשלוח הזמנה לספק על מנת להשלים את המחסור. הזמנה זו תגיע באיסוף ולא בימים קבועים. כמו כן במידה והמערכת מזהה חוסר במוצר היא מוציאה הזמנה אוטומטית עקב חוסר במוצר זה.
 2. הזמנה בימים קבועים (תקופתית) - בסופר ישנם מוצרים שזקוקים לחידוש מלאי באופן קבוע, כלומר בימים בשבוע שנקבעו מראש. על כן בהזמנה תקופתית יהיו המוצרים הללו ויהיה ספק שידאג לספק הזמנה זו באופן קבוע.
- כתב כמיות - זהו הסכם מול הספק שמכיל את הפרטים שניתן לספק דרך ספק מסוים, ואת ההנחות עבור המוצרים. הנחה ניתנת ברכישה של כמות מסוימת ממוצר שהוגדרה מראש, כמות ההנחה משתנה על פי כמות הפריטים שהחברה הזמינה מהספק באותה הזמנה.
- הסכם עם ספק - הסכם עם ספק הוא מסמך המתאר את כמות ההנחה הניתנת על הזמנה של ספק כלשהו ואת הסכום המינימאלי הנדרש בשביל לקבל הנחה זו, בנוסף הסכם עם ספק מכיל את קטלוג המוצרים שהספק מספק ואת מחיריהם.
- פריט - פריט הוא אובייקט ספציפי של מוצר מסוים, וניתן לו מזהה ייחודי בקליטתו במלאי בחנות.
- מוצר - מוצר הוא מוצר במלאי ולו מק"ט ייחודי המוטבע על אריזתו.
- קטגוריה - קטגוריה הוא איגוד של מספר מוצרים ותתי קטגוריות.
- הנחת ספק - הנחת ספק הוא כמות ההנחה שניתנה על ידי ספק מסוים.
- הנחת מכירה - הנחת מכירה הוא הנחה של פריט כלשהו בסניף כלשהו.
- דוח - דוח הוא מידע על החנות ברגע מסוים. ישנם דוחות מלאי, פריטים פגומים וחוסרים.
- סניף - סניף הוא חנות ספציפית, כחלק מהרשת, שאליה מגיעות ההזמנות, ובה קונים את הפריטים.
- משמרת - לכל סניף יש 2 משמרות ביום - בוקר או ערב, בכל משמרת עובדים מספר עובדים בתפקידים שונים ועונים על צרכי הלקוחות בסניף.
- עובד - עובד הוא אדם המוכשר לבצע תפקיד בסניף ומקב על כך שכר.
- זמינות - הימים בהם עובד יכול לעבוד.
- עובד סטנדרטי - עובד שמשוּבץ למשמרת אך ניתן להגיד לו לא לבוא - הוא משובץ רק למקרה הצורך.
- אזורי שילוח - כל אזור שילוח מייצג מרחק שונה מהמרכז הלוגיסטי, בין כל 2 אזורי שילוח עוקבים יש שעה נסיעה.
- אתר - אתר זה שם כללי לספק או סניף.
- חריגה - לכל משאית יש משקל מקסימלי שהיא יכולה לשאת, לפני יציאת המשאית למשלוח, המשאית נשקלת - אם המשקל של המשאיר עם המוצרים גדול מהמשקל המותר - יש חריגה.

תרשים מחלקות Stock



+ supplyReportForSuppliers(): Map<String, Integer>

+ supplyReportForSuppliers(): Map<String, Integer>
+ addNewProduct(String, String, double, double, String, int, int, int, List<String>): void
+ addCategory(List<String>, String): String
+ purchaseItem(int, String): Map<String, boolean>
+ addDiscount(Date, Date, double, boolean, String, String, List<String>): void
+ getItemCalculatedPrice(String): double
+ setAnItemAsDamaged(String, String): void
+ createDeficiencyReport(List<List<String>>): String
+ createDamageReport(List<List<String>>, List<String>): String
+ createStockReport(List<List<String>>): String
+ getPurchaseItemsList(iint): Map<String, Double>

- itemCounter: int
- itemList: Map<String, Item>
- damagedList: Map<DamageType, List<String>>
- expirationList: Map<Date, List<String>>
- discountList: Map<DamageType, List<Discount>>
- productDAO: ProductDAO
- itemDAO: ItemDAO
- discountDAO: DiscountInStoreDAO

+ receiveSupply(int, int, Date): void
+ addProductDiscount(int, Date, Date, double, boolean, DiscountType): void
+ setNotificationAmount(): void
+ isBelowMinimumAmount(): boolean
+ getProductCalculatedPrice(double): double
+ getDamagedItemsByDamage(DamageType): List<Item>
+ getDamagedItemsByDamage(DamageType): List<Item>
+ getExpiredItemsByDate(Date): List<Item>

Report
- REPORT_ID: int
- name: String
- reportDate: Date
- categories: List<Category>
+ toString(): String

Category
- CATEGORY_ID: String
- name: String
- subCategoryList: Map<String, Category>
- productList: List<Product>
- discountList: List<Discount>
- subCounter: int
- categoryDAO: CategoryDAO
- discountInStoreDAO: DiscountInStoreDAO
+ addSubCategory(List<String>, String): String
+ addProduct(Product): void
+ addDiscount(int, Date, Date, double, boolean, DiscountType): void
+ getShortageInCategory(): Map<Product, Integer>
+ getDamagedInCategoryByType(List<DamageType>) Map<DamageType, List<Item>>

Purchase
- PURCHASE_ID: int
- date: Date
- totalPrice: double
- productItemsList: Map<String, List<PurchasedItemDocumentation>>
- purchaseDAO: PurchaseDAO
- discountDAO: DiscountInStoreDAO
+ AddItem(String, String, double, double): void
+ getItemsAndPricesMap(): Map<String, Double>

Item
- ITEM_ID: String
- expirationDate: Date
- location: Location (enum)
- inLocationPosition: String
- isDamaged: boolean
- discountPrice: double
- sellingPrice: double
- damageType: DamageType (enum)
- itemDAO: ItemDAO
+ setIsDamaged(boolean): void
+ setDamageType(DamageType): void

Discount
- DISCOUNT_ID: int
- startDate: Date
- endDate: Date
- discountValue: double
- isPercent: boolean
- discountType: DiscountType (enum)
+ calculateDiscountedPrice(double): double

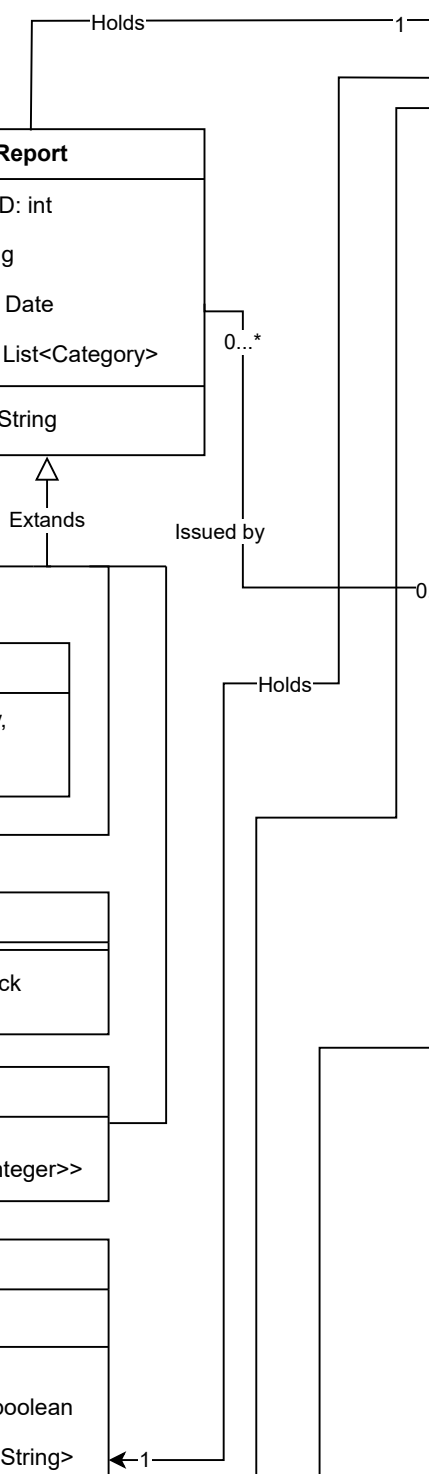
PurchasedItemDocumentation
- itemID: String
- supplierPrice: double
- sellingPrice: double

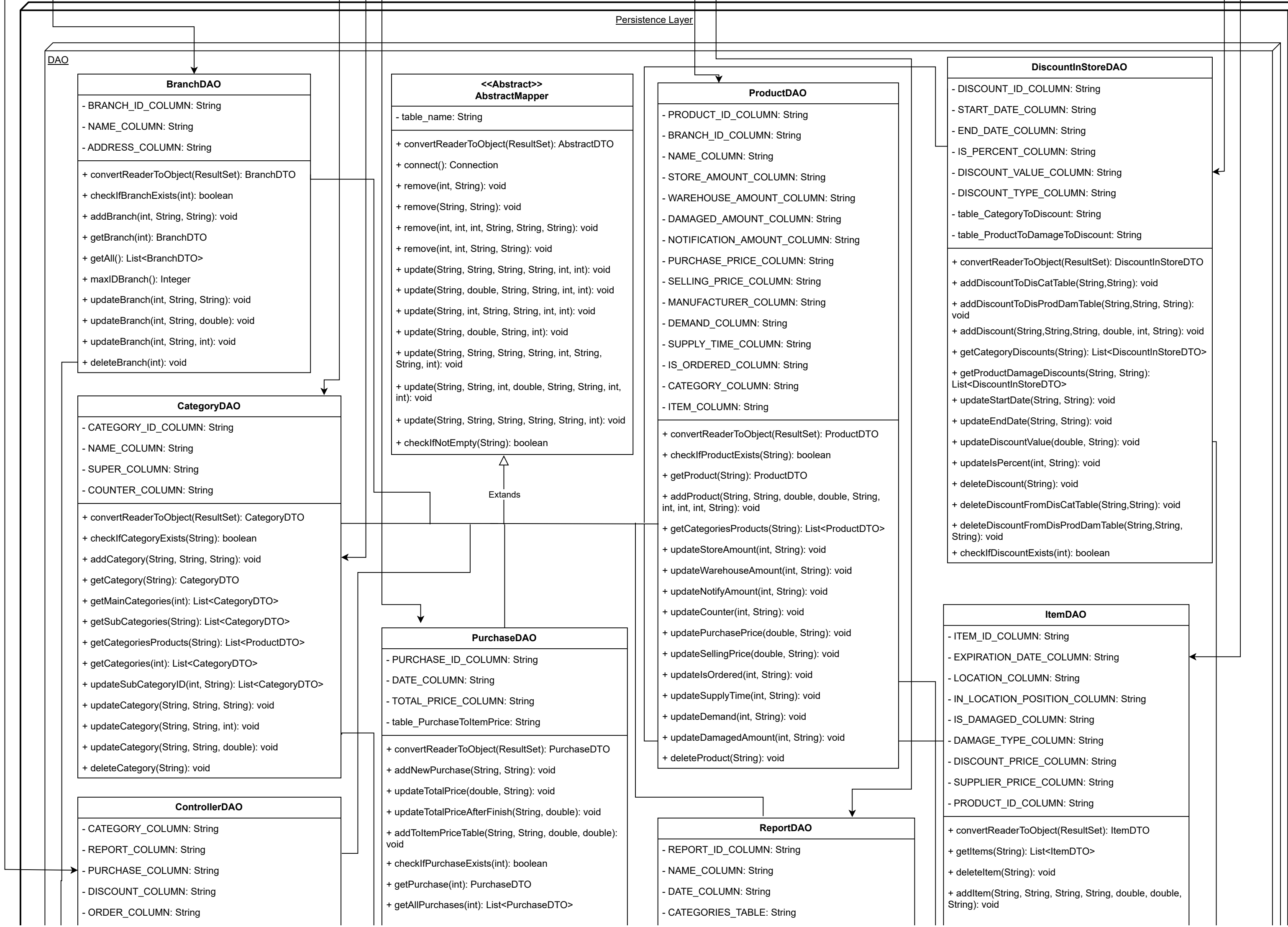
DamagedReport
-damagedItems: Map<Category, Map<Item.DamageType, List<Item>>>

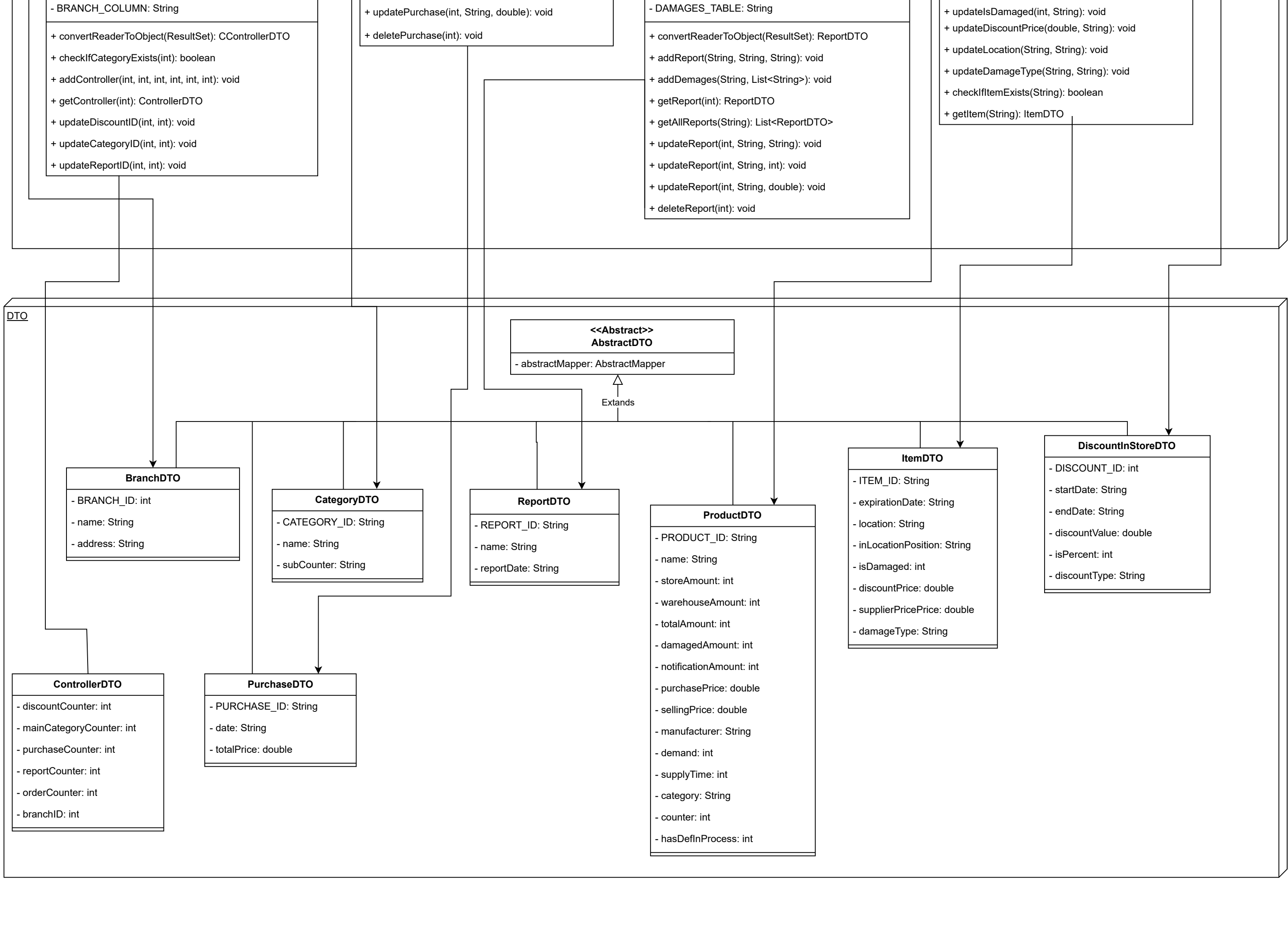
StockReport
+ stringAllCategoryProductsStock (String, Category): String

DeficiencyReport
- categoryProducts: Map<Category, Map<Product, Integer>>

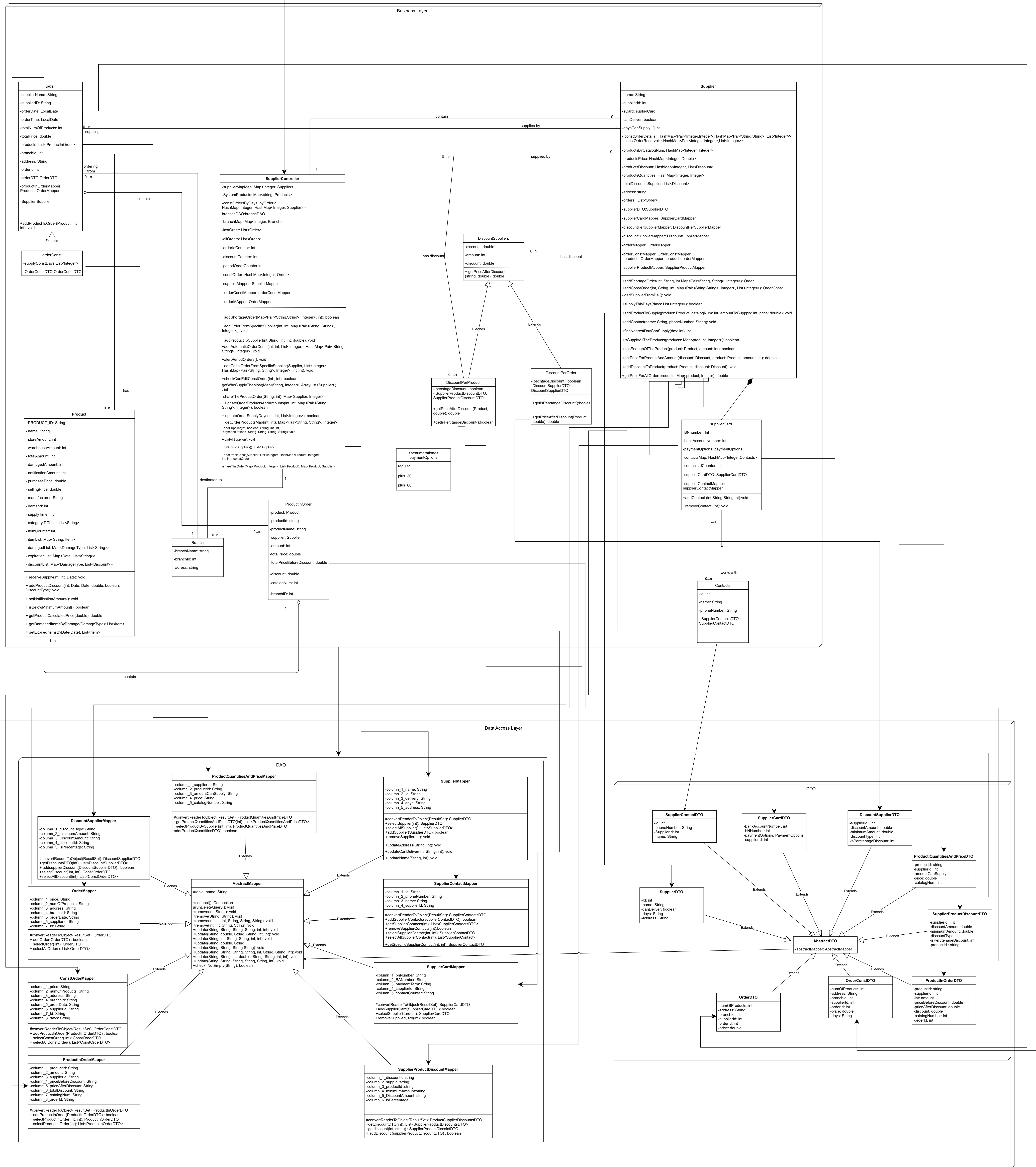
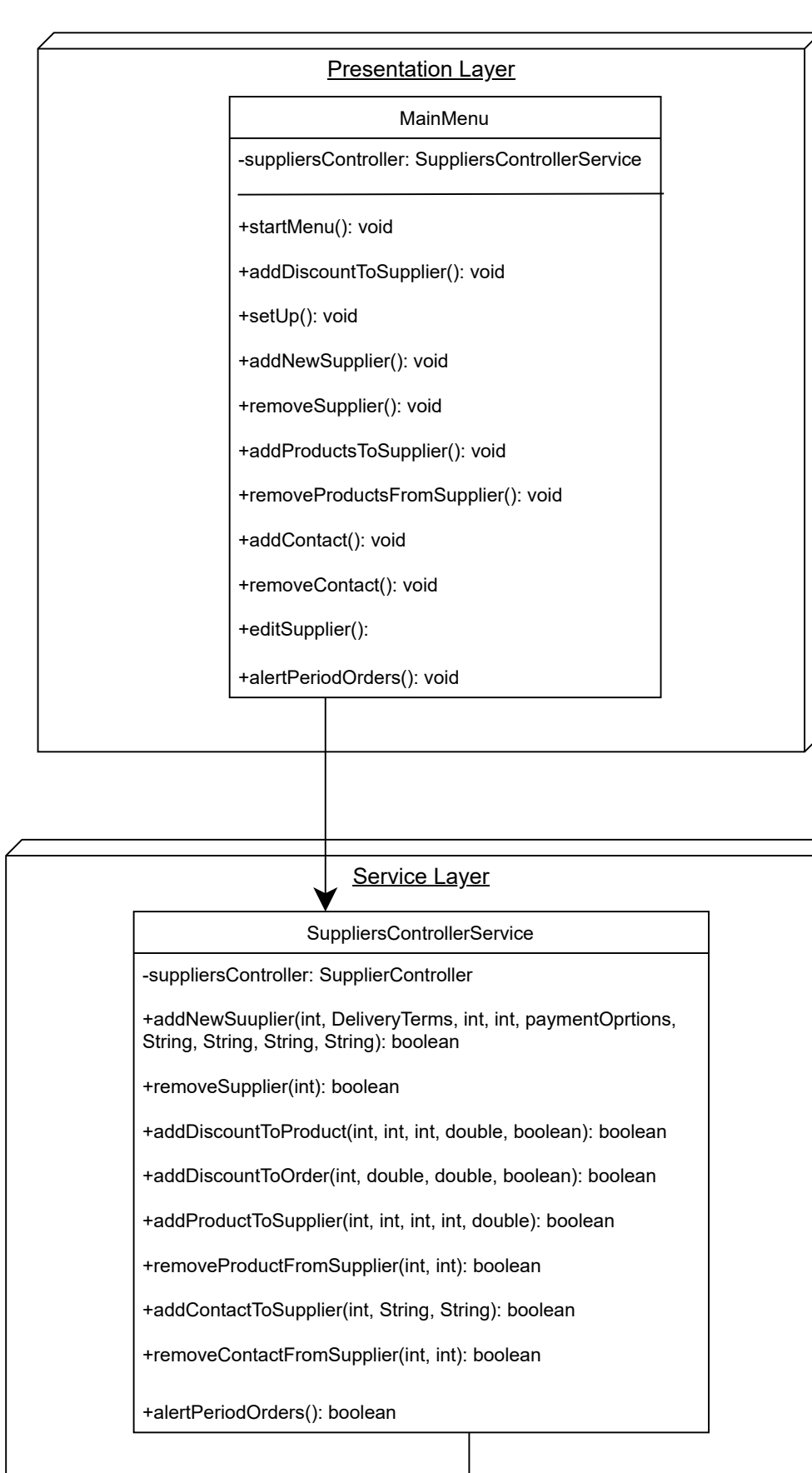
StockToSuppliers
- supController: SuppliersController
+ sendDeficiencyOrderToSuppliers(int, Map<Pair < String,String >, Integer>): boolean
+ getAutoOrdersList(int): Map<Integer, String>
+ deleteAutomaticOrder(int, int): void
+ addAutomaticOrder(int, int, List<Integer>, Map<Pair<String, String>, Integer>): void
+ addExpressOrder(int, Map<Pair<String, String>, Integer>): boolean
+ updateOrderSupplyDays(int, int, List<Integer>): boolean
+ getOrderProductsMap(int, int): Map<Pair<String, String>, Integer>
+ updateOrderProductsAndAmounts(int, int, Map<Pair<String, String>, Integer>): boolean







תרשים מחלקות Suppliers



תרשימי אובייקטים - מודול מלאי

תרחיש 1 – Highly Satisfiable

מתרחש בסניף הרשת באשדוד.

ביצוע רכישה של 2 בקבוקי קוקה קולה 1.5 ליטר – האחד פגום (שרוט) ועל כן בהנחה, והשני תקין.

התרשים מציג את האובייקטים רגע לפני סיום הרכישה.

Business Layer

branchController: BranchController

- branchList = < <1:"Branch1: Branch"> >
- branchCounter = 2

Branch1: Branch

- BRANCH_ID = 1
- name = Branch1
- City = Ashdod
- stockController = StockController

stockController: StockController

- productList = < <"1#444": "1#444: Product"> >
- mainCategoryList = < <1#2:"1#2: Category"> >
- purchaseList = < <1#0:"1#0: Purchase"> >
- reportList = < <1#0:"1#0:Supply Report">
<1#1:"1#1:Damaged Report"> >
- BRANCH_ID = 1
- dicountCounter = 1
- mainCategoryCounter = 3
- purchaseCounter = 1
- reportCounter = 2

1#444: Product

- PRODUCT_ID = 1#444
- name = Coca Cola
- storeAmount = 2
- warehouseAmount = 1
- totalAmount = 3
- damagedAmount = 1
- notificationAmount = 2
- purchasePrice = 4.5
- sellingPrice = 7.0
- manufacturer = Coca Cola
- demand = 1
- supplyTime = 1
- hasDefInProcess= true
- categoryIDChain: <2-1, 2-1-1>
- itemCounter = 4
- itemList = < <1#444-0:"1#444-0: Item">, <1#444-1:"1#444-1: Item">, <1#444-2:"1#444-2: Item"> >
- damagedList = < <SCRATCH:{1#444-1}> >
- expirationList = < <18/5/23:{1#444-0, 1#444-1}>, <24/5/23:{1#444-2}> >
- discountList = < <SCRATCH:{ "1#0: Discount"}> >

1#2: Category

- CATEGORY_ID = "1#2"
- name = Drinks
- subCategoryList = < <1#2-1:"1#2-1 :Category">, <1#2-1-1:"1#2-1-1: Category"> >
- productList = {}
- discountList = {}
- subCounter = 2
- categoryDAO = CD

1#2-1: Category

- CATEGORY_ID = "1#2-1"
- name = Cola
- subCategoryList = < <1#2-1-1:"1#2-1-1: Category"> >
- productList = {}
- discountList = {}
- subCounter = 2
- categoryDAO = CD

1#2-1-1: Category

- CATEGORY_ID = "1#2-1-1"
- name = 1.5 liter
- subCategoryList = <>
- productList = { "1#444: Product" }
- discountList = {}
- subCounter = 1
- categoryDAO = CD

1#0: Discount

- DISCOUNT_ID = "1#0"
- startDate = 15/4/23
- endDate = 18/5/23
- discountValue = 50
- isPercent = True
- discountType = DAMAGED

1#444-0: Item

- ITEM_ID = "1#444 - 0"
- expirationDate = 18/5/23
- location = STORE
- inLocationPosition = 7th ile left, 2nd shelf, section 3
- isDamaged = False
- discountPrice = 7.0
- damageType = NONE

1#444-1: Item

- ITEM_ID = "1#444 - 1"
- expirationDate = 18/5/23
- location = STORE
- inLocationPosition = 7th ile left, 2nd shelf, section 3
- isDamaged = True
- discountPrice = 3.5
- damageType = SCRATCH

1#444-2: Item

- ITEM_ID = "1#444 - 2"
- expirationDate = 24/5/23
- location = WAREHOUSE
- inLocationPosition = 223/44/3/Left
- isDamaged = False
- discountPrice = 7.0
- damageType = NONE

1#0: Purchase

- PURCHASE_ID = "1#0"
- date = 16/4/23
- totalPrice = 10.5
- productItemsList = < <1#444:{PID1, PID2}> >

PID1: PurchasedItemDocumentation

- itemID = 1#444 - 0
- supplierPrice = 4.5
- sellingPrice = 7.0

PID2: PurchasedItemDocumentation

- itemID = 1#444 - 1
- supplierPrice = 4.5
- sellingPrice = 3.5

1#0: Supply Report

- REPORT_ID = "1#0"
- name = Supply Report 1/4/23
- reportDate = 1/4/23
- categories = { "1#2-1-1: Category" }

1#1: Damaged Report

- REPORT_ID = "1#1"
- name = Damaged Report 18/4/23
- reportDate = 18/4/23
- categories = { "1#2-1: Category" }
-damages = {SCRATCH}

Data Access Layer

ProductDAO

- PRODUCT_ID_COLUMN = "PRODUCT_ID"
- NAME_COLUMN = "name"
- STORE_AMOUNT_COLUMN = "storeAmount"
- WAREHOUSE_AMOUNT_COLUMN = "warehouseAmount"
- DAMAGED_AMOUNT_COLUMN = "damagedAmount"
- NOTIFICATION_AMOUNT_COLUMN = "notificationAmount"
- PURCHASE_PRICE_COLUMN = "purchasePrice"
- SELLING_PRICE_COLUMN = "sellingPrice"
- MANUFACTURER_COLUMN = "manufacture"
- DEMAND_COLUMN = "demand"
- SUPPLY_TIME_COLUMN = "supplyTime"
- IS_ORDERED_COLUMN = 0

BranchDAO

- BRANCH_ID_COLUMN = "BRANCH_ID"
- NAME_COLUMN = "name"
- ADDRESS_COLUMN = "adress"

ItemDAO

- ITEM_ID_COLUMN = "ITEM_ID"
- EXPIRATION_DATE_COLUMN = "expirationDate"
- LOCATION_COLUMN = "location"
- IN_LOCATION_POSITION_COLUMN = "inLocationPosition"
- IS_DAMAGED_COLUMN = "isDamaged"
- DAMAGE_TYPE_COLUMN = "damageType"
- DISCOUNT_PRICE_COLUMN = "discountPrice"
- PRODUCT_ID_COLUMN = "productID"

ControllerDAO

- REPORT_COLUMN = "reportCounter"
- CATEGORY_COLUMN= "categoryCounter"
- PURCHASE_COLUMN= "purchaseCounter"
- DISCOUNT_COLUMN= "discountCounter"
- ORDER_COLUMN= "orderCounter"
- BRANCH_COLUMN= "BRANCH_ID"

CD: CategoryDAO

- CATEGORY_ID_COLUMN: "CATEGORY_ID"
- NAME_COLUMN: "name"
- SUPER_COLUMN: "superCategort"

DiscountInStoreDAO

- DISCOUNT_ID_COLUMN = "DISCOUNT_ID"
- START_DATE_COLUMN = "startDate"
- END_DATE_COLUMN = "endDate"
- IS_PERCENT_COLUMN = "isPercent"
- DISCOUNT_VALUE_COLUMN = "discountValue"
- DISCOUNT_TYPE_COLUMN = "discountType"

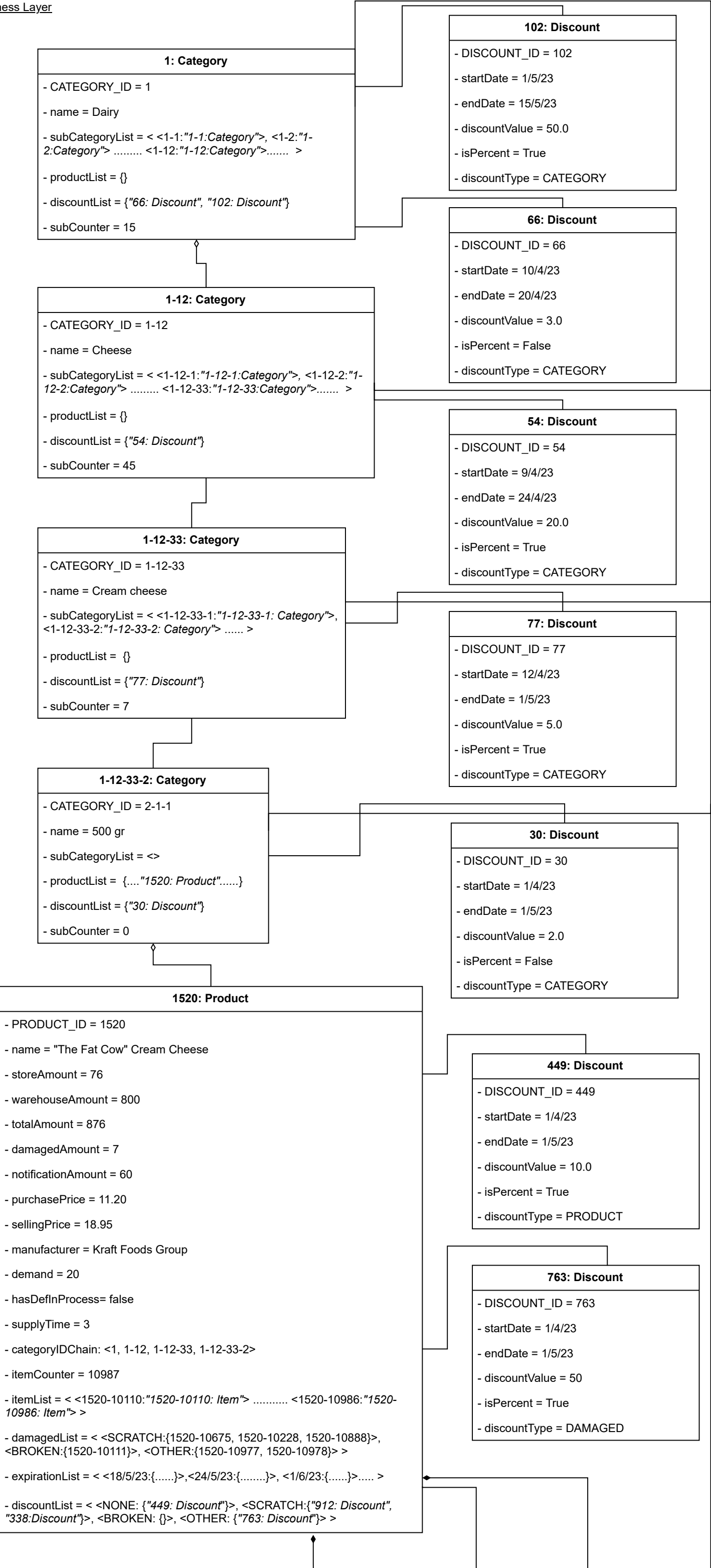
PurchaseDAO

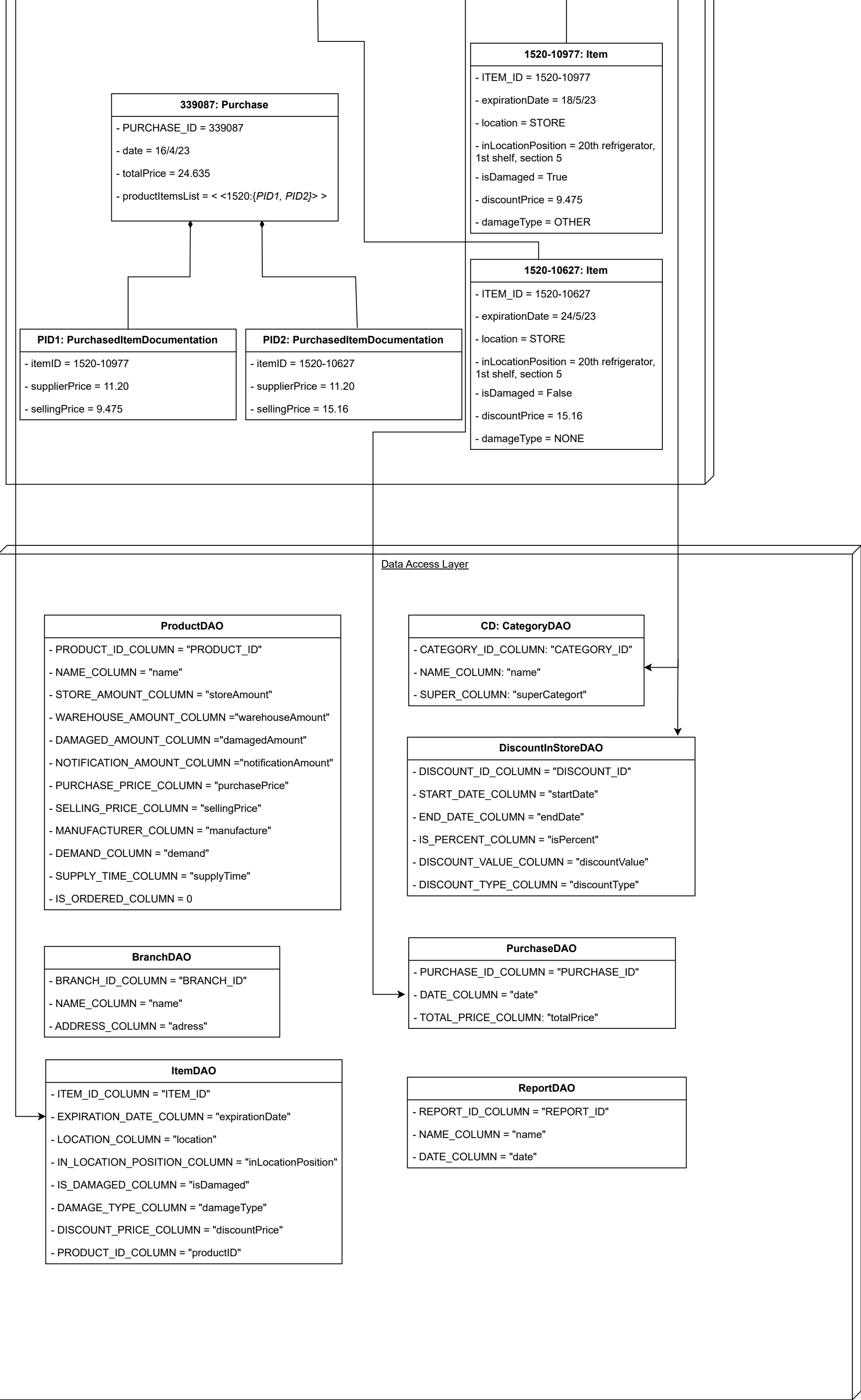
- PURCHASE_ID_COLUMN = "PURCHASE_ID"
- DATE_COLUMN = "date"
- TOTAL_PRICE_COLUMN: "totalPrice"

ReportDAO

- REPORT_ID_COLUMN = "REPORT_ID"
- NAME_COLUMN = "name"
- DATE_COLUMN = "date"

ביצוע רכישה גדולה, המתחילה ברכישת 2 יחידות שלגבינת שמנת "הפרה השמנה", משני תאריכי תוקף שונים. הרכישה מתבצעת בתקופה רוויות מבצעים עבור מוצרי חלב בחנות בשל חג השבועות הקרב. התרשים מציג את המחיר המשתלם ביותר לכל פריט, בהתחשב ב- מבצעים קיימים, תוקף המבצעים, תאריך הרכישה הנקוב, מאפייני כל פריט (תוקף, נזק, מוצר, מחלקות וכו'). התרשים מציג את האובייקטים הרלוונטיים לאחר הוספת 2 הגבינות לסל הקניות.

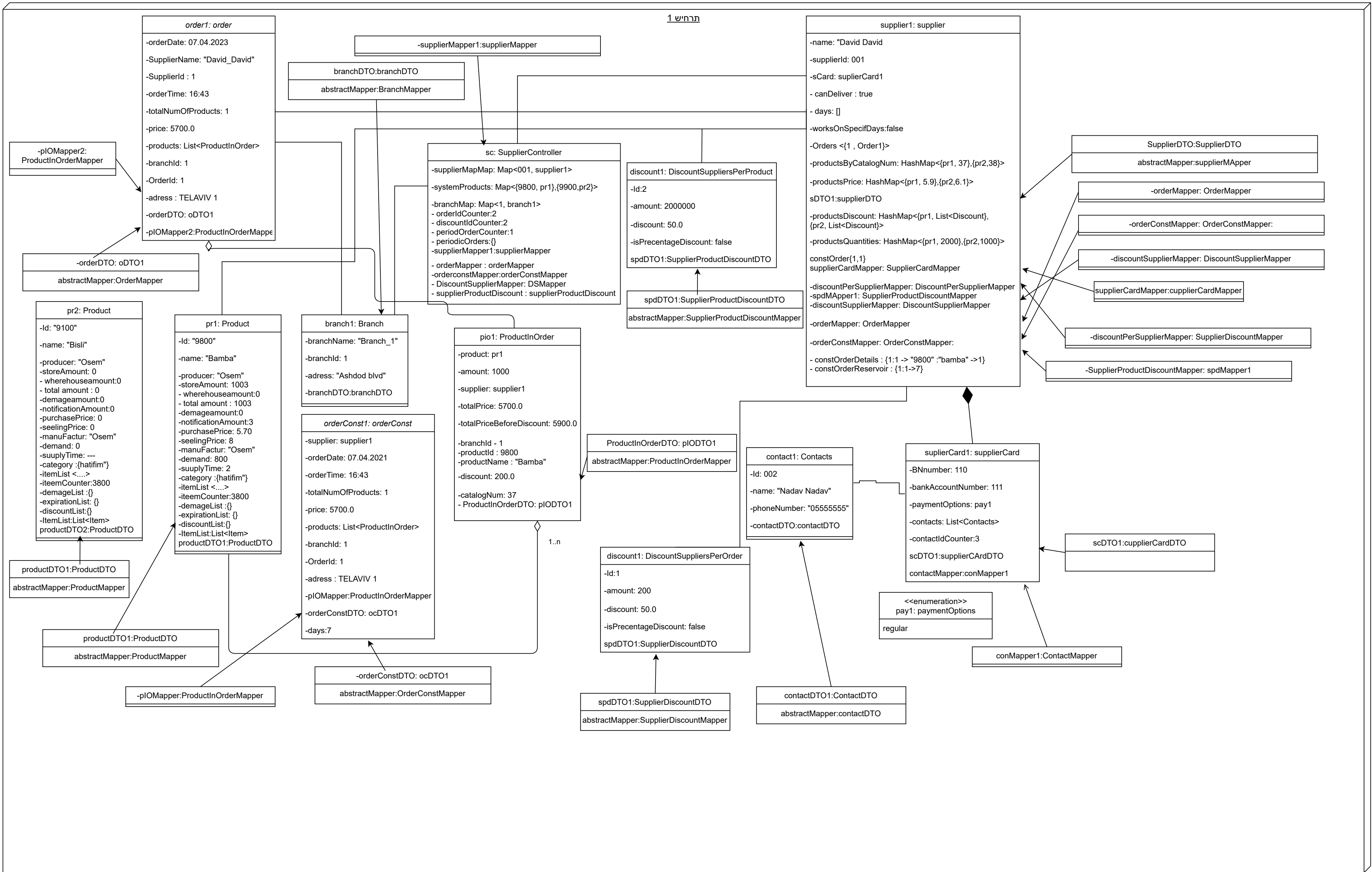




תרשימי אובייקטים - מודול ספקים

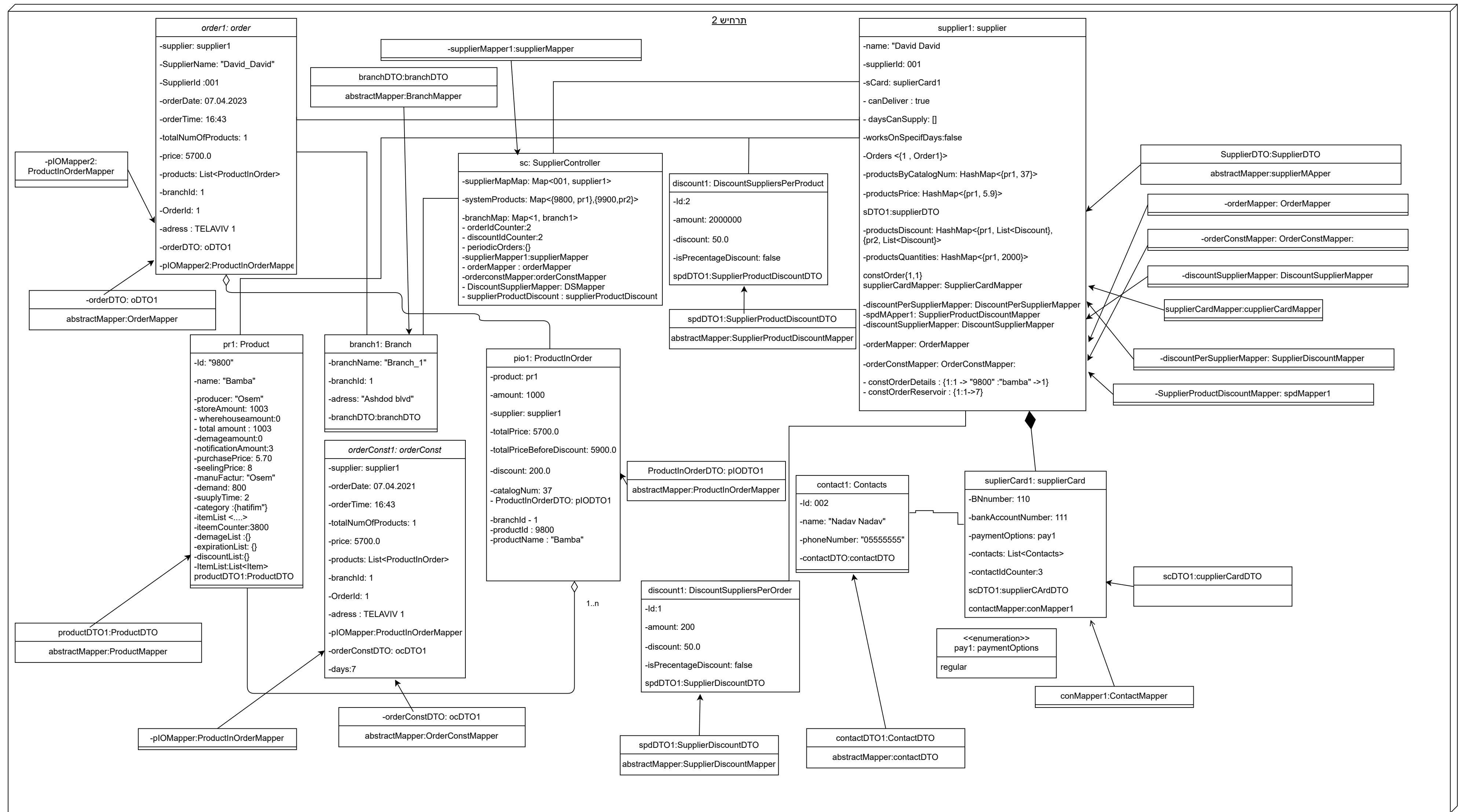
תרחיש 1:

כחלק מהכנסת הספק החדש למערכת, מתווסף גם הסכם הספק שלו, המוצרים שהוא יכול לספק. אחראי האספקה של הרשת מדווח על חוסר בבמבה בסניף 8. לבסוף, יוצאת הזמנה עקב חוסר לסניף ע"י הספק המתאים.

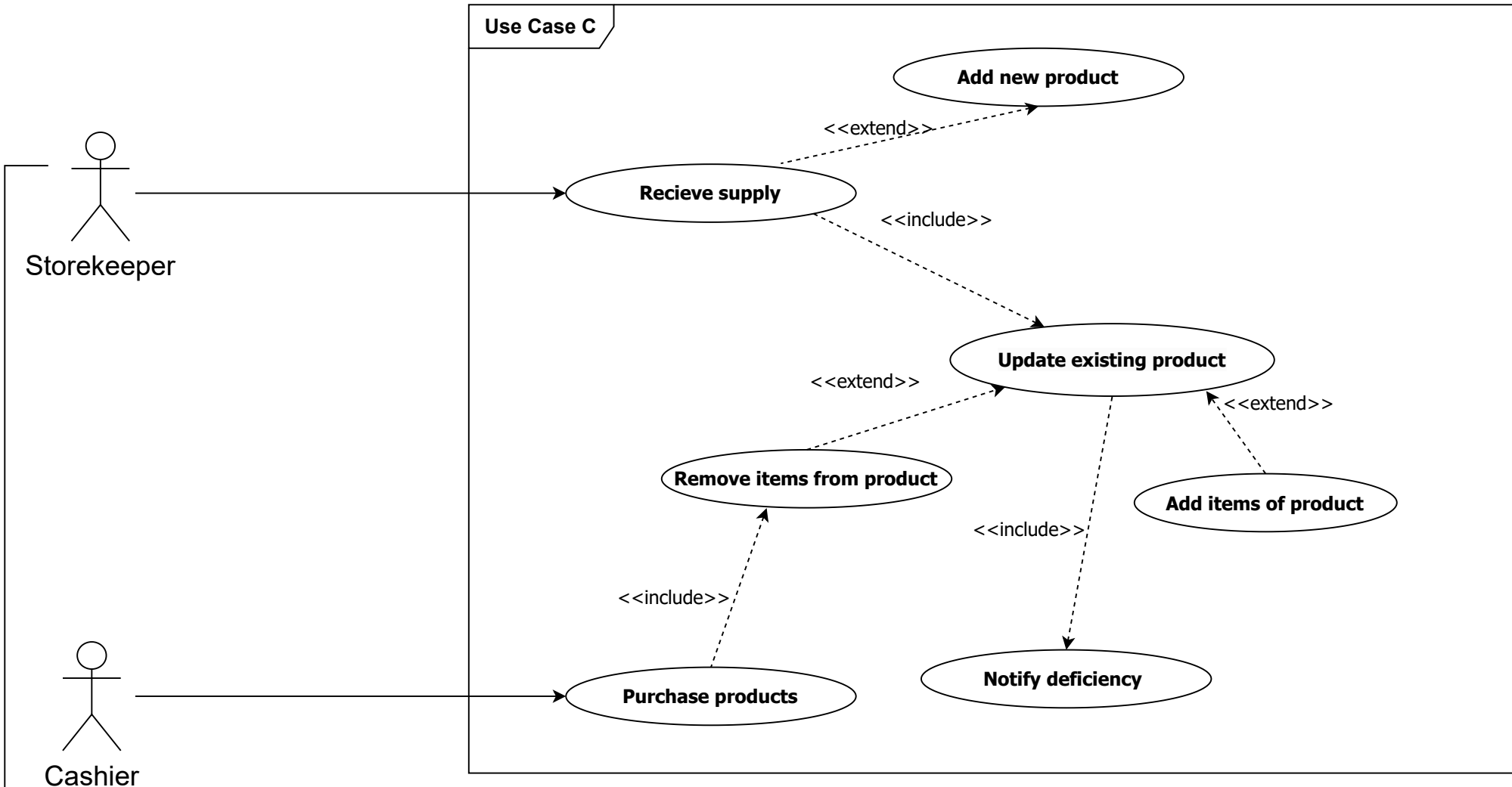


תרחיש 2:

supplier1, מסירים את המוצר "ביסלי" מההסכם עם הספק. כלומר הוא כבר לא יוכל לספק לרשת את מוצר זה.

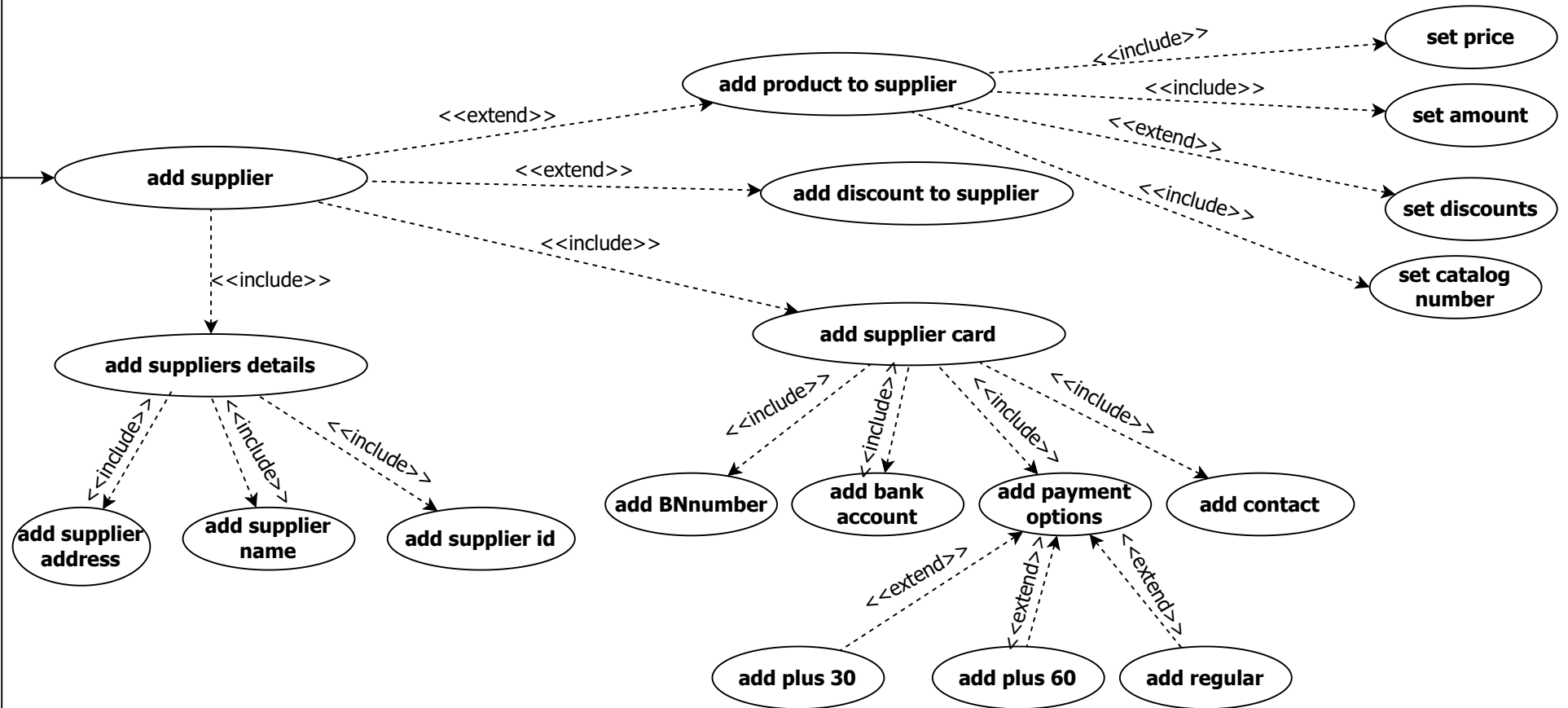


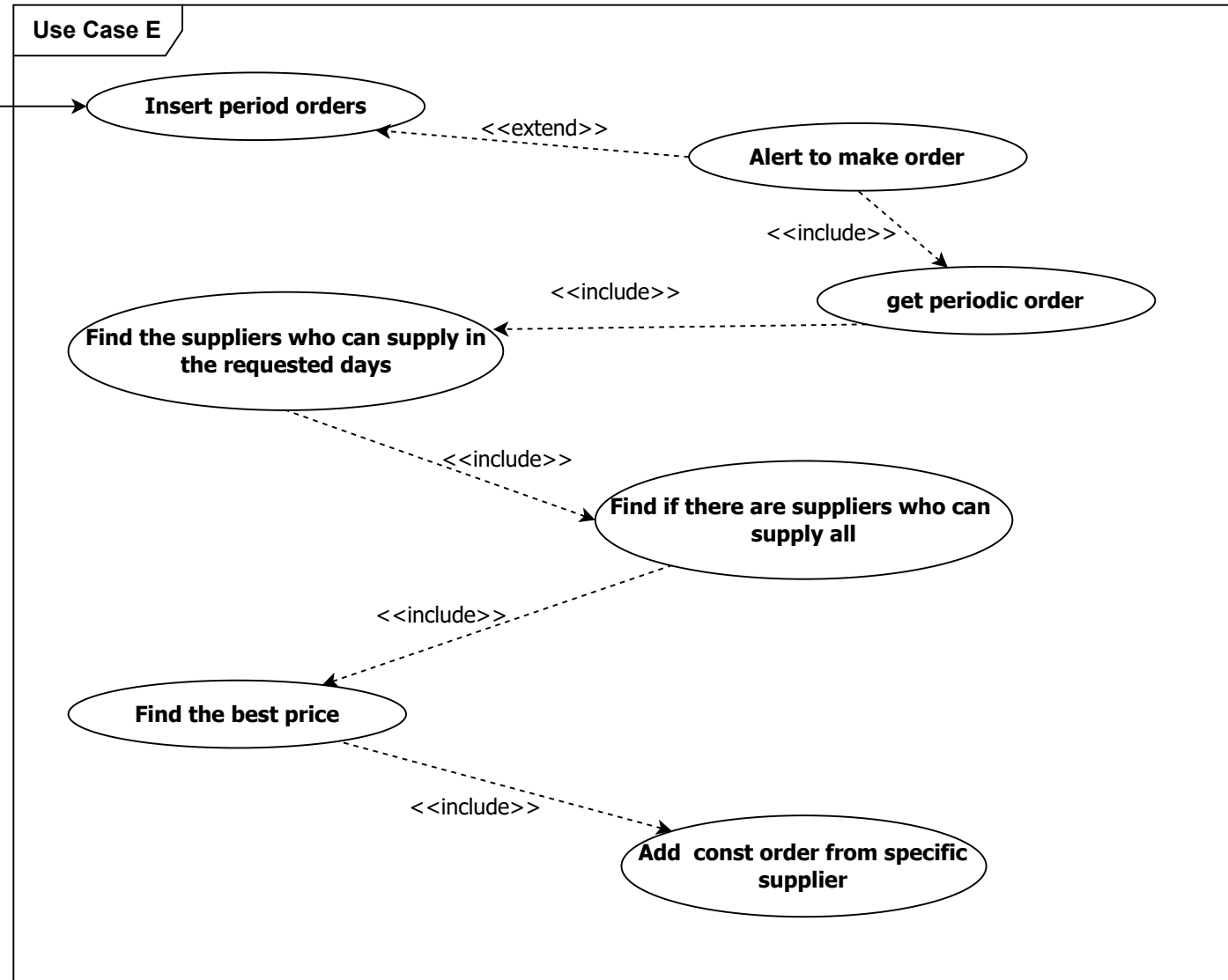
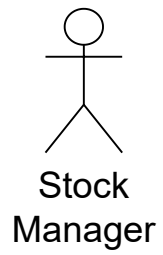
תרחישי שימוש



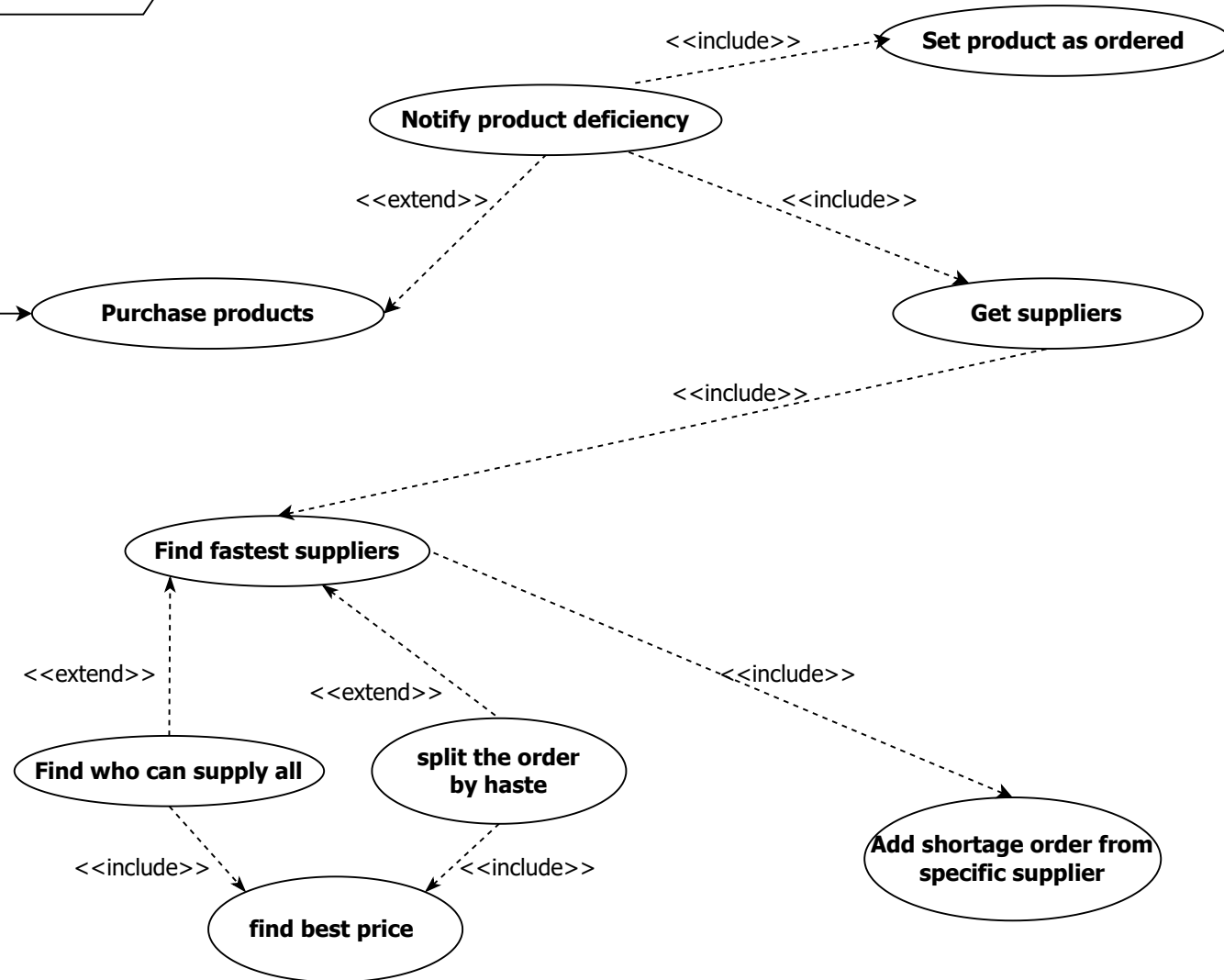
Use Case D

Suppliers Manager



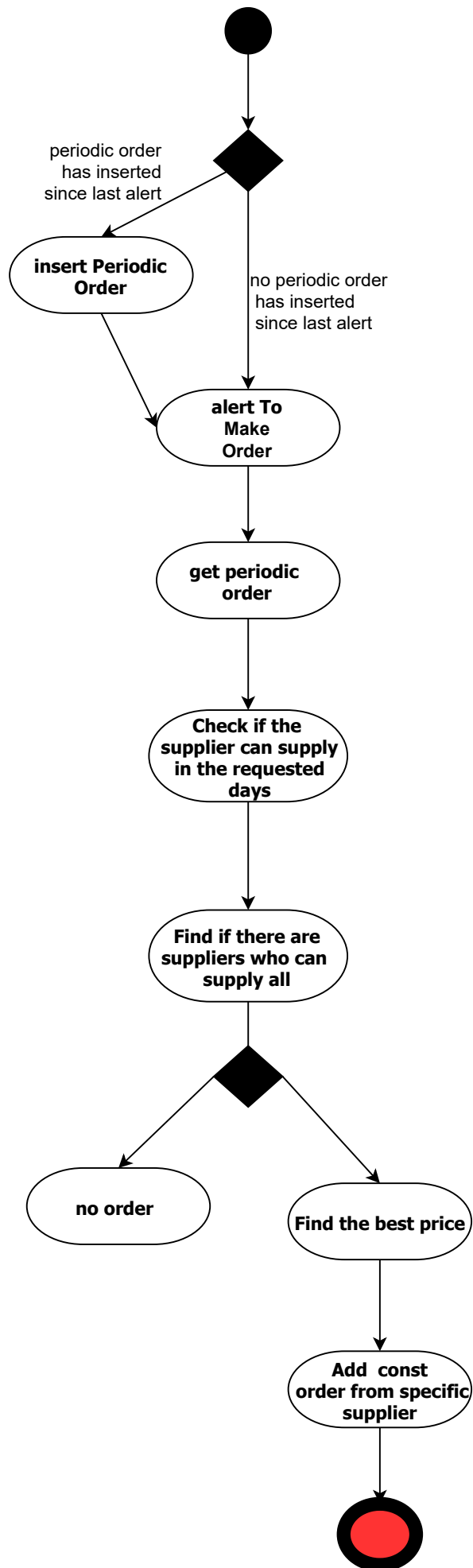


Use Case F



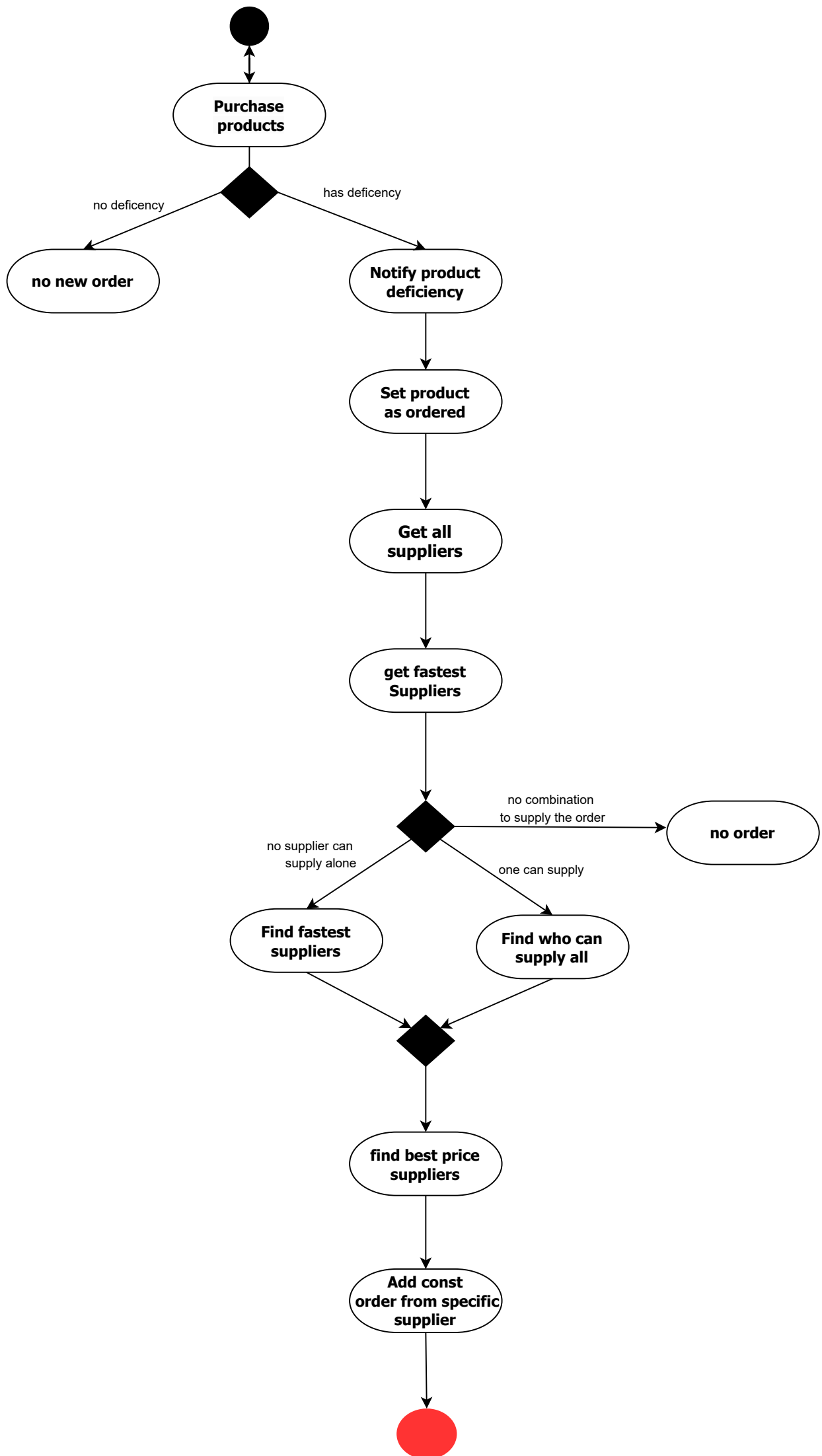
תרחיש E:

Use case name	Issuing a periodic order from a supplier
Textual Description	The scenario describes the process of inserting new periodic order to the system by the stock manger and it shows the automatic process the system operates afterwards for finding the most suitable supplier for the order.
List of Actors	Stock manager.
Pre-conditions	<input type="checkbox"/> The target branch of the order exists in the system. <input type="checkbox"/> The products that the order contains exist in the system.
Post-conditions	<input type="checkbox"/> The details of the periodic order have saved in the system.
Main success scenario	The stock manager insert new periodic order the system → the automatic process found the best supplier for the order → the supplier will provide the order on the scheduled day/ days.
Alternatives/ Extension	<input type="checkbox"/> If there isn't a supplier who can provide all the order by himself, the system won't allow to accept the order.



תרחיש F:

Use case name	Issuing an order from a supplier due to a shortage
Textual Description	The system detects a lack of at least one of the store's products, afterwards an automatic process starts by the system which leads to an order to fulfill the shortage.
List of Actors	Stock manager.
Pre-conditions	<input type="checkbox"/> A shortage at one of the store's products is discovered.
Post-conditions	<input type="checkbox"/> The system finished it's checking for trying to fulfill the shortage.
Main success scenario	The system detects a lack of at least one of the store's products → the automatic process found a supplier the suitable suppliers who can answer the shortage according to by the fastest time and the best price with priority to the time → The store is once against with no shortages.
Alternatives/ Extension	



חוזים

תרחיש E:

Contract co1: Insert period order

Operation:	Insert period order(branchID: Integer, productsMap: Map<String, Integer>, time: List<Integer>)
Cross references	Create and place periodic order
PreConditions	<ul style="list-style-type: none">The branch exists in the system.The products in productsList are existing in the system.
PostConditions	<ul style="list-style-type: none">The details of the order were sent for finding a suitable supplier (association formed).The field 'periodicOrder' was updated (attribute modification)

Contract co2: Alert to make order

Operation:	AlertToMakeOrder()
Cross references	Create and place periodic order
PreConditions	<ul style="list-style-type: none">The products in productsList are existing in the system.
PostConditions	<ul style="list-style-type: none">Returns all the orders that need to be provided in the next 24 hours(Link formed).

Contract co3: get periodic order

Operation	Get_const_order(supplier:supplier, orderId : int)
Cross references	Create and place periodic order
PreConditions	<ul style="list-style-type: none">There is a peridodic order in the system.
PostConditions	<ul style="list-style-type: none">The system has the last updated periodic order

Contract co4: Find the suppliers who can supply in the requested days

Operation	supplyThisDay(day: int)
Cross references	Create and place periodic order
PreConditions	<ul style="list-style-type: none">The products in productsList are existing in the system.An alert was created and returned relevant orders that need to be provide in 24 hours.
PostConditions	<ul style="list-style-type: none">Returns a list of the suppliers that are available at the requested days of the orders(instance creation).

Contract co5: find if there are suppliers who can supply all

Operation	isSupplyAllTheProducts(productsAndAmount: Map<String, Integer>)
Cross references	Create and place periodic order
PreConditions	<ul style="list-style-type: none">• The products in productsList are existing in the system.• An alert was created and returned relevant orders that need to be provide in 24 hours.• There are suppliers that are available at the requested days of the orders.
PostConditions	<ul style="list-style-type: none">• Returns a list of the suppliers that are available at the requested days of the orders and are able to provide all the needed products by their amount(instance creation).

Contract co6: find the best price

Operation	getPriceForAllOrder(productsAndAmount: Map<string, Integer>)
Cross references	Create and place periodic order
PreConditions	<ul style="list-style-type: none">• The products in productsList are existing in the system.• An alert was created and returned relevant orders that need to be provide in 24 hours.• There are suppliers that are available at the requested days of the orders and are able to provide all the needed products by their amount.
PostConditions	<ul style="list-style-type: none">• The function returns the supplier with the best price for this order(Link formed).

Contract co7: Add const order from specific supplier

Operation	addConstOrder(supplierID: Integer, branchID: Integer, productsAmounr: Map<String, Integer>, orderId: Integer, listOfDays: List<Integer>)
Cross references	Create and place periodic order
PreConditions	<ul style="list-style-type: none">• The products in productsList are existing in the system.• An alert was created and returned relevant orders that need to be provide in 24 hours.• There are suppliers that are available at the requested days of the orders and are able to provide all the needed products by their amount.• The suppliers with the best price for this order has been found.
PostConditions	<ul style="list-style-type: none">• An instance of constOrder has been created (instance creation).• An instance of productInOrder has been created (instance creation).• The constOrderDAO has been updated (attribute modification).• The productInOrderDAO has been updated (attribute modification).

Contract co1: Purchase products

Operation	purchaseProducts(branchID: Integer, productsMap: Map<String, Integer>)
Cross references	Create and place deficiency order
PreConditions	<ul style="list-style-type: none"> The branch exists in the system. The products in productsMap exist in the system.
PostConditions	<ul style="list-style-type: none"> All purchased items are deleted from the system (instance deletion).

Contract co2: Notify product deficiency

Operation	notifyDeficiency(productsMap: Map<String, Integer>)
Cross references	Create and place deficiency order
PreConditions	<ul style="list-style-type: none"> The branch exists in the system. The products in productsMap exist in the system.
PostConditions	<ul style="list-style-type: none"> Make deficiency alert to suppliers system (Link formed).

Contract co3: Set product as ordered

Operation	setProductAsOrdered (productid: String)
Cross references	Create and place deficiency order
PreConditions	<ul style="list-style-type: none"> The product exists in the system.
PostConditions	<ul style="list-style-type: none"> The field isOrder is updated (attribute modification).

Contract co5: Find fastest suppliers

Operation	SuppliersNextSupplyDayArray()
Cross references	Create and place deficiency order
PreConditions	<ul style="list-style-type: none"> The branch exists in the system. The products in productsList are existing in the system. List with all the suppliers in the system.
PostConditions	<ul style="list-style-type: none"> An ordered list of suppliers by the nearest day that they can supply(instance creation).

Contract co4: Get suppliers

Operation	loadAllSuppliers()
Cross references	Create and place deficiency order
PreConditions	<ul style="list-style-type: none"> The branch exists in the system. The products in productsMap exist in the system.
PostConditions	<ul style="list-style-type: none"> Returns a list of all the suppliers that are in the system (instance creation).

Contract co6: Find who can supply all

Operation	isSupplyAllTheProducts(productsAndAmount: Map<String, Integer>)
-----------	---

Cross references	Create and place deficiency order
------------------	-----------------------------------

PreConditions	<ul style="list-style-type: none">• The branch exists in the system.• The products in productList are existing in the system.• List with all the suppliers in the system.
---------------	---

PostConditions	<ul style="list-style-type: none">• The nearest day a suppliers can supply all by himself(Link formed)
----------------	--

Contract co7: split the order by haste

Operation	shareTheOrder()
-----------	-----------------

Cross references	Create and place deficiency order
------------------	-----------------------------------

PreConditions	<ul style="list-style-type: none">• The branch exists in the system.• The products in productList are existing in the system.• List with all the suppliers in the system.• The max days it will take to provide all the order.
---------------	---

PostConditions	<ul style="list-style-type: none">• Returns a list of supplier who supply the requested order, each supplier may supplier different products and amount on different day(link formed)
----------------	---

Contract co8: find Best price supplier

Operation	getPriceforAllOrder(HashMap<SupplierProduct, Integer> productsAndAmount)
-----------	--

Cross references	Create and place deficiency order
------------------	-----------------------------------

PreConditions	<ul style="list-style-type: none">• The branch exists in the system.• The products in productList are existing in the system.• List with all the suppliers exist in the system.• There are few supplliers that can supply the order in the same time
---------------	---

PostConditions	<ul style="list-style-type: none">• Between the suppliers that can supply the order in the same time , the cheapest one has selected.
----------------	---

Contract co9: Add shortage order from specific supplier

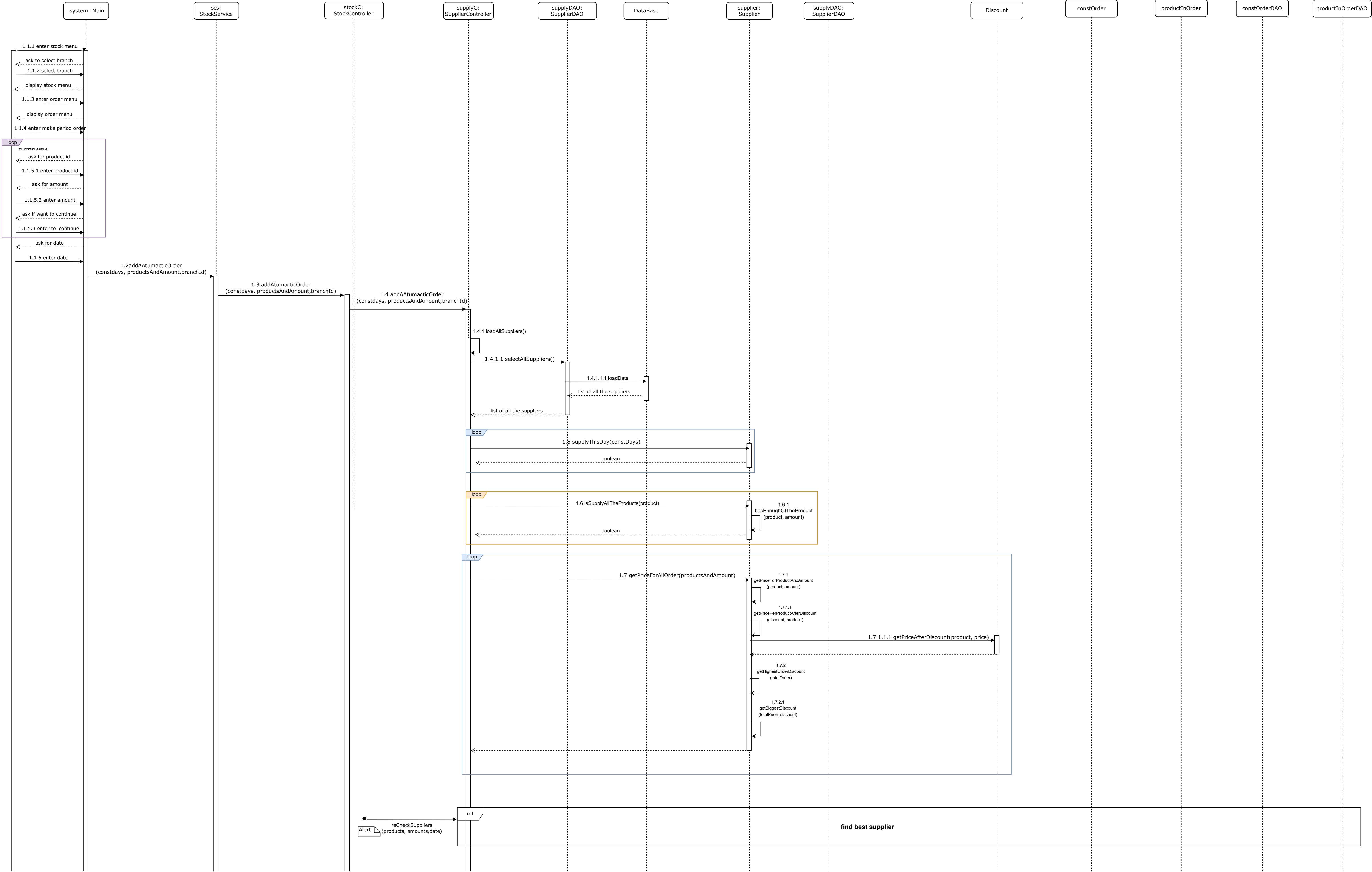
Operation	addOrderFromSpecificSupplier(supplierID: Integer, branchID: Integer, productsAmount: Map<String, Integer>, orderId: Integer)
-----------	--

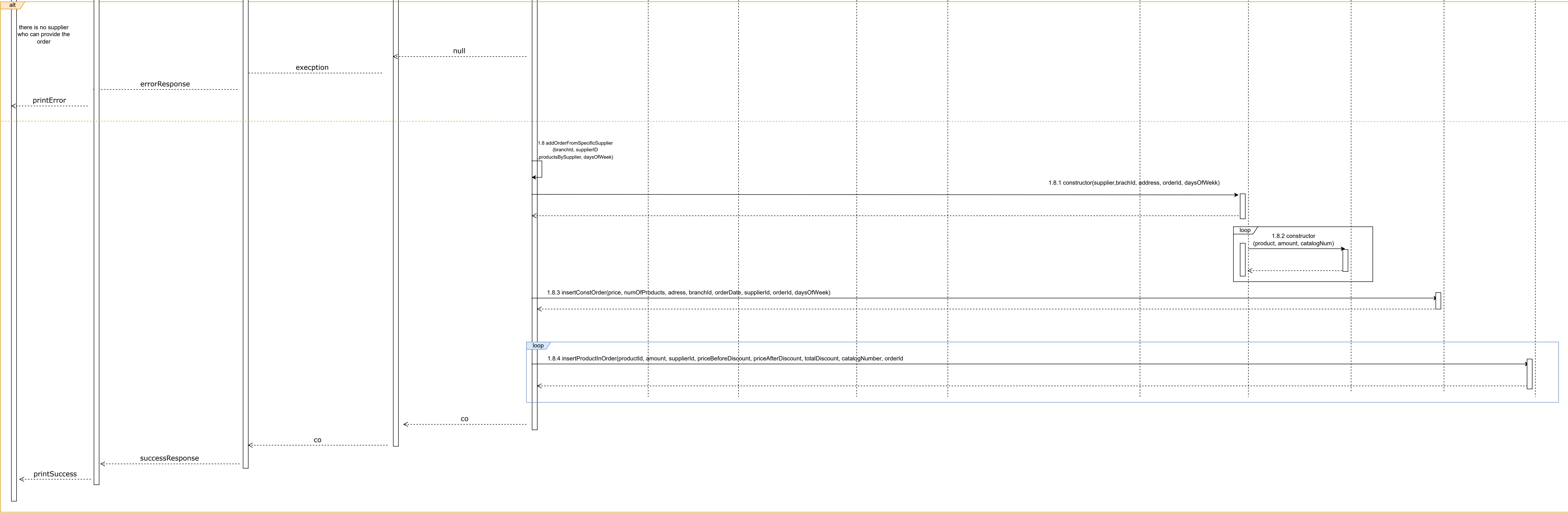
Cross references	Create and place deficiency order
------------------	-----------------------------------

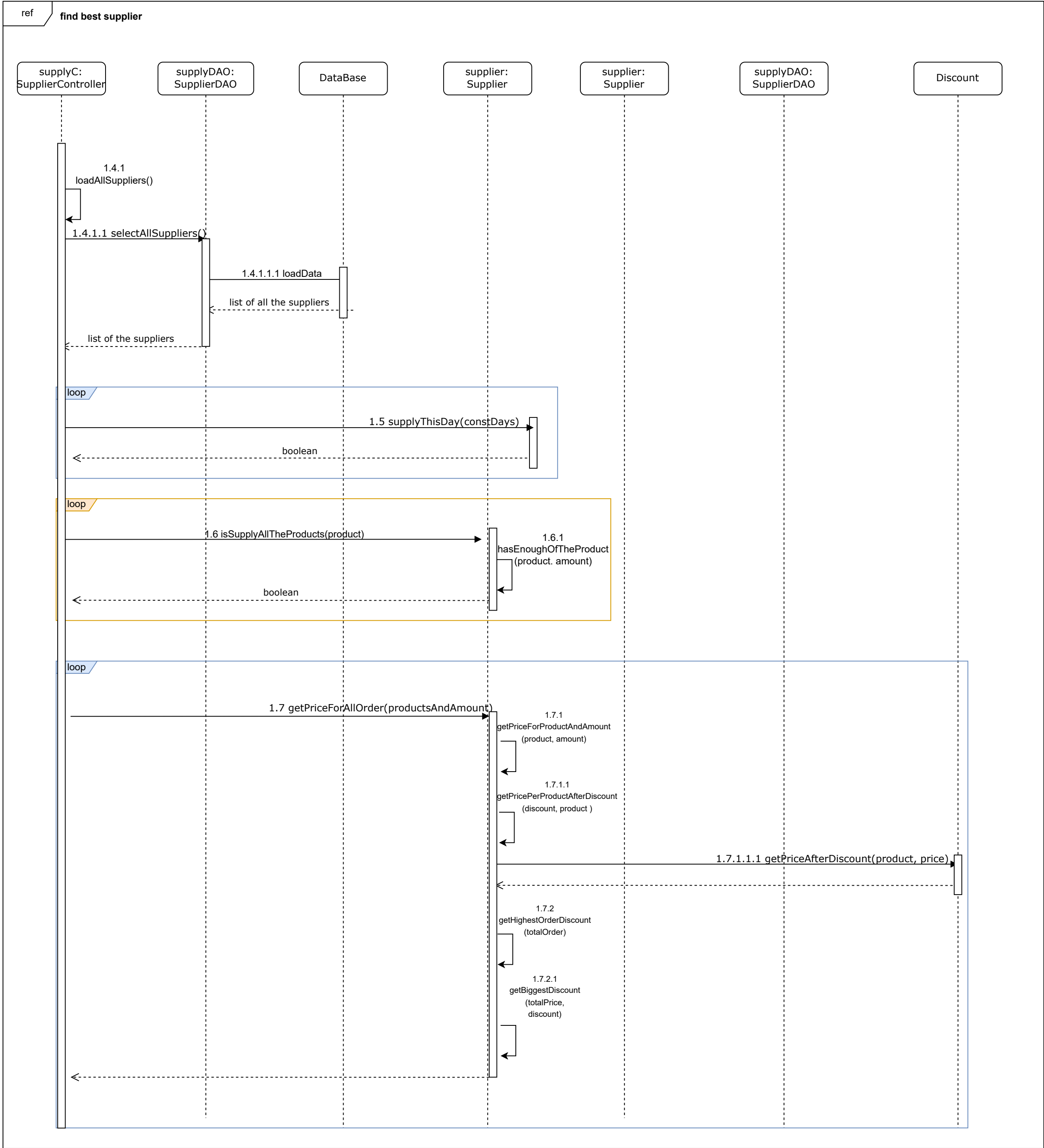
PreConditions	<ul style="list-style-type: none">• The branch exists in the system.• The products in productList are existing in the system.• List with all the suppliers in the system.• The max days it will take to provide all the order.• A list of the suppliers who are involved in providing the order.
---------------	--

PostConditions	<ul style="list-style-type: none">• For each supplier, An instance of Order has been created (instance creation).• For each supplier and his order, An instance of productInOrder has been created (instance creation).• The orderDAO has been updated (attribute modification).• The productInOrderDAO has been updated (attribute modification).
----------------	---

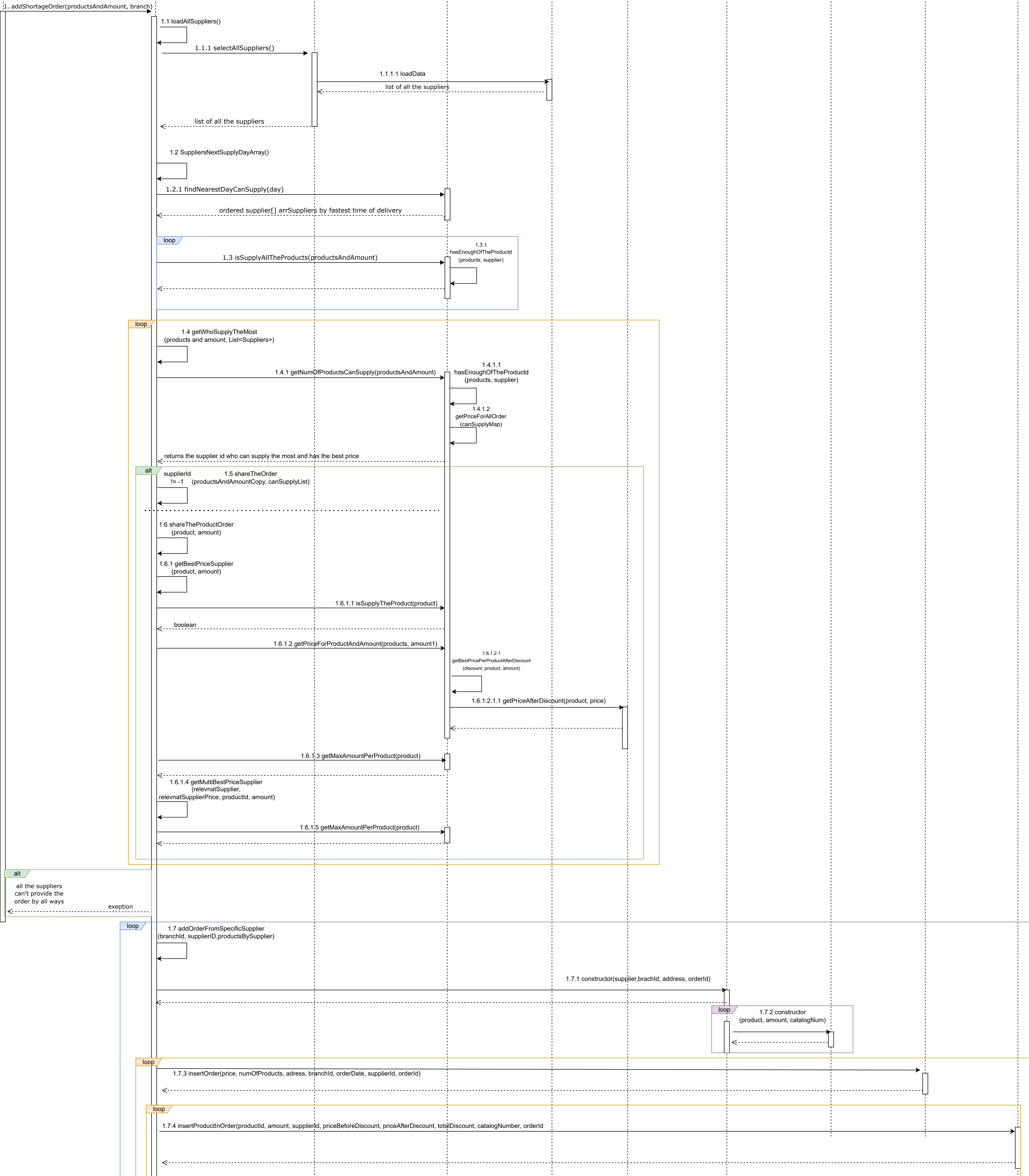
תרחיש e
הוצאת הזמנה תקופתית מספק







f תרחיש



תרשים ERD

